Cryptanalysis of Ake98

Jorge Nakahara Júnior[1] and Daniel Santana de Freitas[2]

[1] jorge_nakahara@yahoo.com.br
[2] LabSEC, Laboratório de Segurança em Computação, UFSC, Brazil.
santana@inf.ufsc.br

Abstract. This paper describes a linear attack on the Ake98 block cipher, an updated version of the Akelarre cipher presented by Alvarez *et al.* at the SAC'96 Workshop. The new attacks require the assumption of weak keys. It is demonstrated that Ake98 does not introduce enough security measures to counter cryptanalytic attacks, both in a known-plaintext and in a ciphertext-only setting. A key-recovery attack on 4.5-round Ake98, for instance, is applicable to a weak-key class of size 2^{108}, and requires only 71 known plaintexts, with an effort of $71 \cdot 2^{70}$ half-round decryptions. Moreover, the existence of weak keys precludes the use of Ake98 as a building block for other cryptographic primitives, such as in Davies-Meyer Hash mode. Attacks using weak keys can be applied up to 11.5 rounds of Ake98 with less effort than an exhaustive key search. But, Ake98 with 8.5 rounds is already slower than IDEA, RC6 or AES, which implies that this updated version of the Akelarre cipher does not seem to provide significant advantages (security or efficiency) compared to the former, more established ciphers.

Keywords: cryptanalysis, Akelarre, Ake98, IDEA, RC5, RC6, AES.

1 Introduction

Akelarre is a block cipher designed by Alvarez *et al.* [4] and presented at SAC'96 Workshop. Akelarre combines design features from the IDEA [9] and RC5 [11] ciphers, and processes 128-bit text blocks, uses a 128-bit key, and iterates 4 rounds plus an output transformation (OT). The operations of modular addition, $\boxplus$, and exclusive-or, $\oplus$, were inherited from IDEA, while bitwise rotation, $\lll$, came from RC5. In [8], Knudsen and Rijmen presented known-plaintext and ciphertext-only attacks on Akelarre for any number of rounds, and that are independent of the key schedule algorithm. Further attacks were also presented by Ferguson and Schneier in [6], but using chosen plaintext.

Subsequently, the designers of Akelarre presented Ake98 [3] that is claimed to avoid the previous attacks on Akelarre.

This paper is organized as follows: Sect. 2 describes briefly the Akelarre block cipher; Sect. 3 describes Ake98 and the main differences with Akelarre. Sect. 4 explains the attack on Ake98, its similarity to the attack of Knudsen-Rijmen, the attack requirements and its complexity. Subsect. 4.2 describes a ciphertext-only attack on Ake98. Sect. 5 compares the software performance of Ake98 with that of AES, IDEA and RC6. Sect. 6 concludes the paper.

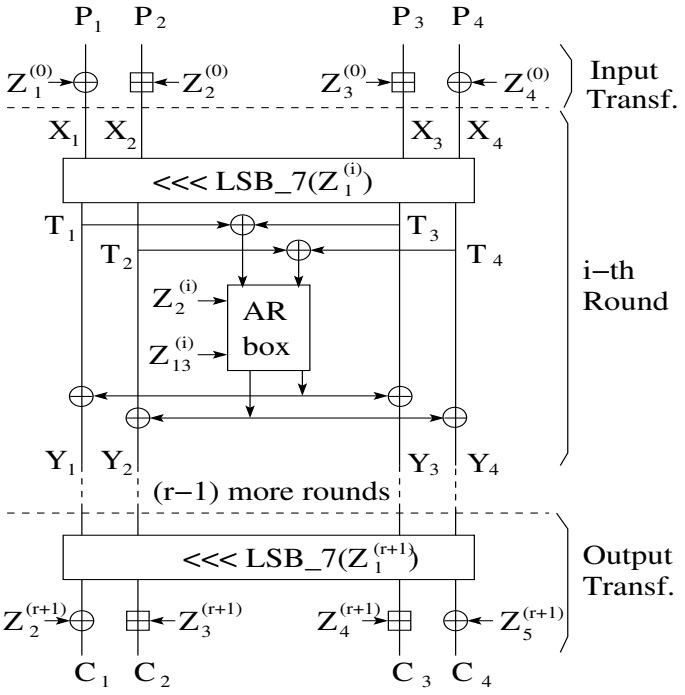

Fig. 1. Computational graph of the Akelarre block cipher.

2 The Akelarre Cipher

The Akelarre block cipher was presented at the SAC'96 workshop, and its design combines features from the IDEA and RC5 ciphers. Akelarre uses three operations on w-bit words: bitwise exclusive-or, denoted $\oplus$, addition modulo 2^w, denoted $\boxplus$, and bitwise rotation, denoted $\lll$. The multiplication operation of IDEA is absent. A note on terminology: the notation $\mathrm{lsb}_i(X)$ (lower case) will denote the i-th least significant bit(s) of

X, while $\mathrm{LSB}_j(X)$ (upper case) will denote the ensemble of i consecutive least significant bits[3].

All of the internal operations in Akelarre are on w-bit words. Akelarre operates on variable-length words, text blocks and keys, and uses a variable number of rounds. The suggested parameter values in [4] are: 128-bit blocks, 32-bit words, 128-bit key and 4 rounds. Fig. 1 depicts the computational graph of Akelarre. The MA-box of IDEA becomes an AR-box (Addition-Rotation box). Details of the AR-box are given in the Appendix.

The key schedule algorithm of Akelarre will not be described in this paper but the interested reader can find further information in [4].

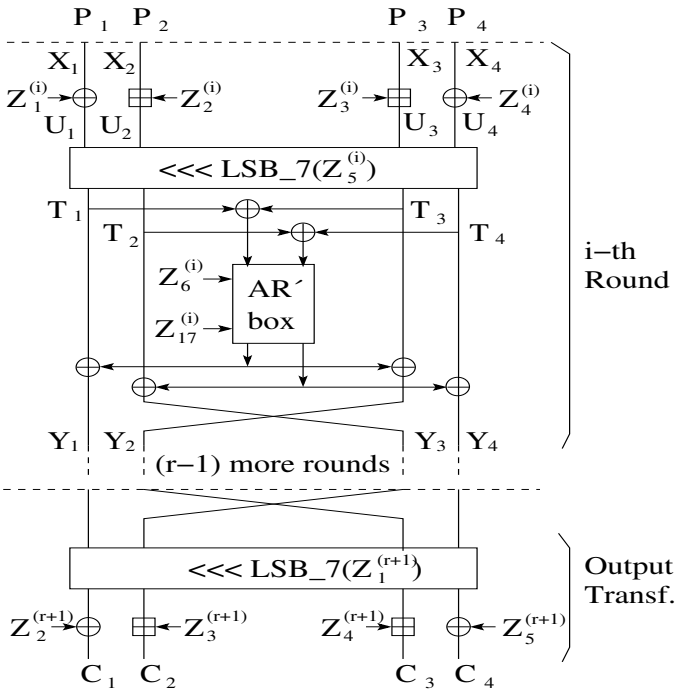

Fig. 2. Computational graph of the Ake98 cipher.

[3] For example, if $X = 01101110_2$ in binary, then $\mathrm{lsb}_1(X) = 0$, $\mathrm{lsb}_2(X) = 1$, but $\mathrm{LSB}_2(X) = 2$, and $\mathrm{LSB}_3(X) = 6$.

3 The Ake98 Cipher

In [3], an updated version of Akelarre, called Ake98, was presented. It is claimed that Ake98 resists the attacks made formerly on Akelarre [6, 8]. Ake98 differs from Akelarre in the new AR-box (Addition-Rotation box), in the swapping of words at the end of a round, and the addition of subkeys in the beginning of each round. Fig. 2 depicts the computational graph of Ake98. Details of the AR-box of Ake98 are provided in the Appendix (Sect. 7).

The block and key sizes, the number of rounds, and the internal word sizes in Ake98 are variable but no minimum value is set by the authors for any parameter. For comparison purposes, the same parameter values for Akelarre will also be assumed for Ake98.

The key schedule of Ake98 will not be described here. The only property assumed for the key schedule of Ake98 is that it behaves as a pseudo-random number generator. Further details of the subkey generation in Ake98 can be found in [3].

4 A Known-Plaintext Attack on Ake98

The Knudsen-Rijmen attack [8] on Akelarre exploited the fact that the leftmost input to the AR-box can be computed from just two input and output words in a round. From Fig. 1, $T_1 \oplus T_3 = Y_1 \oplus Y_3$ for the i-th round. Similarly, $T_2 \oplus T_4 = Y_2 \oplus Y_4$. These relations can be extended across the key-dependent rotation as[4]

$$(Y_1 \oplus Y_3)|(Y_2 \oplus Y_4) = ((X_1 \oplus X_3)|(X_2 \oplus X_4)) \lll Z_1^{(i)} . \qquad (1)$$

Relation (1) always holds, independent of the round subkeys and of the AR-box. Moreover, this is an iterative relation, namely it can be combined with itself. The attack of [8] uses (1) as an invariant for the full Akelarre, except for the input and output transformations (Fig. 1). Notice that this attack applies to any number of rounds.

Notice that (1) does not hold for the IDEA and PES [9] ciphers because of the addition and the multiplication operations (Akelarre and Ake98 do not use multiplication).

For Ake98 similar relations to (1) can be obtained, under weak subkey assumptions. Observe that in Fig. 2, $T_1 \oplus T_3 = Y_1 \oplus Y_2$, and $T_2 \oplus T_4 =$

[4] The vertical bar '|' stands for concatenation.

$Y_3 \oplus Y_4$. To achieve a similar relation to (1), both of them are combined, resulting in

$$T_1 \oplus T_2 \oplus T_3 \oplus T_4 = Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4 \,. \tag{2}$$

Furthermore, across the key-dependent rotation:

$$Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4 = (U_1 \oplus U_2 \oplus U_3 \oplus U_4) \lll Z_5^{(i)} \,, \tag{3}$$

where it is implicitly assumed that only the least significant seven bits of $Z_5^{(i)}$ are used as the rotation amount.

Relation (3) does not hold in general across the modular addition with subkeys at the beginning of a round, but it still holds with certainty for the least significant bit, because of the absence of a carry bit[5]:

$$\mathrm{lsb}_1(Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4) = \mathrm{lsb}_{-Z_5^{(i)} \bmod 32+1}(X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus Z_1^{(i)} \oplus Z_2^{(i)} \oplus Z_3^{(i)} \oplus Z_4^{(i)}) \,. \tag{4}$$

Relation (4) is not iterative, but under the assumption that $\mathrm{LSB}_7(Z_5^{(i)}) \in \{0, 32, 64, 96\}$, that is, a rotation amount that is a multiple of the word size of Ake98, this relation can be rewritten as:

$$\mathrm{lsb}_1(Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4) = \mathrm{lsb}_1(X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus Z_1^{(i)} \oplus Z_2^{(i)} \oplus Z_3^{(i)} \oplus Z_3^{(i)}) \,, \tag{5}$$

which is iterative, and independent of the new AR-box. Iterating relation (5) four times, results in a probabilistic distinguisher, under the weak subkey assumptions: $\mathrm{LSB}_7(Z_5^{(1)}), \mathrm{LSB}_7(Z_5^{(2)}), \mathrm{LSB}_7(Z_5^{(3)}), \mathrm{LSB}_7(Z_5^{(4)}) \in \{0, 32, 64, 96\}$. Assuming that the key schedule algorithm of Ake98 can be modeled as a pseudo-random number generator, each of the weak subkey assumptions will be taken independently. Therefore, the probability that these assumptions hold for four consecutive rounds is approximated as $(4/2^7)^4 = 2^{-20}$ since there are 2^7 possible rotation amounts. Similarly, under the assumption of a random behavior of the key schedule of Ake98, the weak subkey assumptions are expected to hold for a class of $2^{128} \cdot 2^{-20} = 2^{108}$ (weak) user keys.

For 4-round Ake98, a 1-bit invariant, using relation (5), can be constructed:

$$\begin{aligned} &\mathrm{lsb}_1(P_1 \oplus P_2 \oplus P_3 \oplus P_4) \oplus \mathrm{lsb}_1(C_1 \oplus C_2 \oplus C_3 \oplus C_4) = \\ &\mathrm{lsb}_1(Z_1^{(1)} \oplus Z_2^{(1)} \oplus Z_3^{(1)} \oplus Z_4^{(1)}) \oplus \mathrm{lsb}_1(Z_1^{(2)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus Z_4^{(2)}) \oplus \\ &\mathrm{lsb}_1(Z_1^{(3)} \oplus Z_2^{(3)} \oplus Z_3^{(3)} \oplus Z_4^{(3)}) \oplus \mathrm{lsb}_1(Z_1^{(4)} \oplus Z_2^{(4)} \oplus Z_3^{(4)} \oplus Z_4^{(4)}) \,. \end{aligned} \tag{6}$$

Notice in (6) that for a fixed key, one bit of information on the key, namely, $\mathrm{lsb}_1(\oplus_{i,j=1}^{4} Z_i^{(j)})$, can be recovered given one bit $\mathrm{lsb}_1(C_1 \oplus C_2 \oplus C_3 \oplus C_4)$

[5] Parameters of the 'lsb' function are counted from 1 up to 32.

of ciphertext information and of the plaintext, $\mathrm{lsb}_1(P_1 \oplus P_2 \oplus P_3 \oplus P_4)$; or alternatively, given the plaintext, and an unknown key, one bit of information on the ciphertext can be obtained with certainty.

Relation (6) alone can be used to distinguish 4-round Ake98 (under weak key assumptions and without the OT) from a random permutation, using only known plaintext/ciphertext pairs.

Moreover, (6) can be used in a key-recovery attack, to discover the subkeys of the OT. If we call the output transformation a half-round, this is a 0.5R attack. The corresponding 1-bit distinguisher is:

$$\begin{aligned}&\mathrm{lsb}_1(P_1 \oplus P_2 \oplus P_3 \oplus P_4) \oplus \mathrm{lsb}_1(Z_1^{(1)} \oplus Z_2^{(1)} \oplus Z_3^{(1)} \oplus Z_4^{(1)}) \oplus \\ &\mathrm{lsb}_1(Z_1^{(2)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus Z_4^{(2)}) \oplus \mathrm{lsb}_1(Z_1^{(3)} \oplus Z_2^{(3)} \oplus Z_3^{(3)} \oplus Z_4^{(3)}) \oplus \\ &\mathrm{lsb}_1(Z_1^{(4)} \oplus Z_2^{(4)} \oplus Z_3^{(4)} \oplus Z_4^{(4)}) = \\ &\mathrm{lsb}_1((C_1 \oplus Z_2^{(5)} \oplus C_2 \boxminus Z_3^{(5)} \oplus (C_3 \boxminus Z_4^{(5)}) \oplus (C_4 \oplus Z_5^{(5)})) \ggg \mathrm{LSB}_5(Z_1^{(5)}))\,.\end{aligned} \tag{7}$$

In a known-plaintext setting, the unknowns in (7) are[6] $\mathrm{LSB}_5(Z_1^{(5)})$, $Z_2^{(5)} \oplus Z_5^{(5)}$, $Z_4^{(5)}$, $Z_3^{(5)}$ and $\mathrm{lsb}_1(\oplus_{i,j=1}^{4} Z_i^{(j)})$. Actually, only the $\mathrm{LSB}_5(Z_1^{(5)})$-th bit of $Z_2^{(5)} \oplus Z_5^{(5)}$ is required. In total, $5+1+32+32 = 70$ subkey bits can be recovered, and the effort for each of the 2^{70} subkey candidates is equivalent to decrypting the OT, or a half-round computation.

The amount of known plaintext (KP) needed for the attack is computed as follows. Once the 70 subkey bits are guessed correctly in (7), the combined value of plaintext, ciphertext and guessed subkey bits must match the 1-bit key-dependent invariant:

$$\mathrm{lsb}_1(\oplus_{i,j=1}^{4} Z_i^{(j)})\,. \tag{8}$$

The value of (8) is unknown, but is constant for a fixed key. Therefore, the correct 70 subkey bits must always give a constant value, whatever the plaintext, while the wrong 70 subkey bits will only match (8) with a probability of 1/2. This reasoning is based on the fact that the correct subkey value actually decrypts the OT, reducing the 4.5 rounds to four rounds, where the distinguisher can be checked; but, the wrong subkey will not decrypt the OT correctly, rather, it will add a further 0.5 rounds on top of the 4.5-round Ake98, and its 1-bit result shall be (more) random. Thus, the expected number of false alarms (subkeys) surviving this filtering after 71 known plaintext/ciphertext pairs are used, is $2^{70} \cdot (\frac{1}{2})^{71} < 1$.

The attack using the 4-round distinguisher (7) was applied to a 4.5-round Ake98 and not to 5.5 rounds, because the latter would require too

[6] Even though the least significant seven bits of $Z_1^{(5)}$ are used in a block, the invariant involves (the xor of) 32-bit words, thus only the five least significant bits of $Z_1^{(5)}$ are relevant.

many subkey bits to recover simultaneously, namely the subkeys of one round plus the OT. The distinguisher for 5.5-round Ake98 would be:

$$\begin{aligned}
&\mathrm{lsb}_1(P_1 \oplus P_2 \oplus P_3 \oplus P_4 \oplus Z_1^{(1)} \oplus Z_2^{(1)} \oplus Z_3^{(1)} \oplus Z_4^{(1)}) \oplus \\
&\mathrm{lsb}_1(Z_1^{(2)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus Z_4^{(2)}) \oplus \mathrm{lsb}_1(Z_1^{(3)} \oplus Z_2^{(3)} \oplus Z_3^{(3)} \oplus Z_4^{(3)}) \oplus \\
&\mathrm{lsb}_1(Z_1^{(4)} \oplus Z_2^{(4)} \oplus Z_3^{(4)} \oplus Z_4^{(4)}) = \\
&\mathrm{lsb}_1((((C_1 \oplus Z_2^{(6)}) \ggg \mathrm{LSB}_5(Z_1^{(6)}) \oplus Z_1^{(5)}) \oplus \\
&((C_2 \boxminus Z_3^{(6)}) \ggg \mathrm{LSB}_5(Z_1^{(6)}) \boxminus Z_2^{(5)}) \oplus \\
&((C_3 \boxminus Z_4^{(6)}) \ggg \mathrm{LSB}_5(Z_1^{(6)}) \boxminus Z_3^{(5)}) \oplus \\
&((C_4 \oplus Z_5^{(6)}) \ggg \mathrm{LSB}_5(Z_1^{(6)}) \oplus Z_4^{(5)})) \ggg \mathrm{LSB}_5(Z_5^{(5)}))\,.
\end{aligned} \tag{9}$$

Note that a 1.5R-attack on 5.5-round Ake98 using (9) would require guessing one bit of $Z_1^{(5)} \oplus Z_4^{(5)}$, $Z_2^{(6)}$, and $Z_5^{(6)}$; the full 32 bits of $Z_2^{(5)}$, $Z_3^{(5)}$, $Z_3^{(6)}$, $Z_4^{(6)}$, and $\mathrm{LSB}_5(Z_1^{(6)})$, $\mathrm{LSB}_5(Z_5^{(5)})$, or 141 subkey bits simultaneously.

In general, the more rounds are attacked the smaller the weak key class. Table 1 lists the number of weak keys for attacks on a different number of rounds of Ake98. All the attacks require about 71 known plaintext/ciphertext pairs, and effort equivalent to $71 \cdot 2^{70}$ decryptions of the OT. Assuming the OT is about half a round, it represents $\frac{1}{9}$ of a 4.5-round encryption. Thus, the attack complexity is equivalent to $\frac{1}{9} \cdot 71 \cdot 2^{70} \approx 8 \cdot 2^{70} = 2^{73}$ 4.5-round encryptions.

Table 1. Estimated effort and weak key class size, $|WKC|$, in attacks on Ake98.

# Rounds	$\lvert WKC\rvert$	Attack Effort
4.5	2^{108}	2^{72}
6.5	2^{98}	$\frac{1}{13} \cdot 71 \cdot 2^{70} \approx 2^{72}$
8.5	2^{88}	$\frac{1}{17} \cdot 71 \cdot 2^{70} \approx 2^{72}$
10.5	2^{78}	$\frac{1}{21} \cdot 71 \cdot 2^{70} \approx 2^{72}$
11.5	2^{73}	$\frac{1}{23} \cdot 71 \cdot 2^{70} \approx 2^{71.5}$
12.5	2^{68}	$\frac{1}{25} \cdot 71 \cdot 2^{70} \approx 2^{71.5}$
25.5	2^{3}	$\frac{1}{51} \cdot 71 \cdot 2^{70} \approx 2^{70}$

From Table 1, in order to avoid attacks based on weak keys, Ake98 should have more than 25.5 rounds. Nonetheless, our attack is more efficient than exhaustive key search (in a weak-key class) up to 11.5 rounds. For 12.5-round Ake98, the exhaustive key search effort for a weak-key class of size 2^{68} is less than the $2^{71.5}$ encryptions of our attack.

As the last remark, the attack described in this section, although explained for a 32-bit word version of Ake98, applies similarly to other word sizes.

4.1 New Weak Key Classes

The rationale for choosing subkeys that cause rotations by multiples of 32 bits can be further extended to weak subkeys whose values are of the form $16 + 32t$, $0 \leq t \leq 3$. In this case, a one-round relation with input (X_1, X_2, X_3, X_4) and output $(Y_1^{(1)}, Y_2^{(1)}, Y_3^{(1)}, Y_4^{(1)})$ becomes:

$$\begin{aligned} \mathrm{lsb}_1(X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus Z_1^{(1)} \oplus Z_2^{(1)} \oplus Z_3^{(1)} \oplus Z_4^{(1)}) = \\ \mathrm{lsb}_1((Y_1^{(1)} \oplus Y_2^{(1)} \oplus Y_3^{(1)} \oplus Y_4^{(1)}) \ggg 16)\,, \end{aligned} \tag{10}$$

which is not iterative. But, if two consecutive rounds have block rotations by amounts of the form $16 + 32t$, $0 \leq t \leq 3$, then for the next round:

$$\begin{aligned} \mathrm{lsb}_1(Y_1^{(1)} \oplus Y_2^{(1)} \oplus Y_3^{(1)} \oplus Y_4^{(1)} \oplus Z_1^{(2)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus Z_4^{(2)}) = \\ \mathrm{lsb}_1((Y_1^{(2)} \oplus Y_2^{(2)} \oplus Y_3^{(2)} \oplus Y_4^{(2)}) \ggg 16)\,, \end{aligned} \tag{11}$$

where $Y_i^{(2)}$, $1 \leq i \leq 4$ are the output words after two rounds. Combining (10) and (11) results in:

$$\begin{aligned} \mathrm{lsb}_1(X_1 \oplus X_2 \oplus X_3 \oplus X_4) = \mathrm{lsb}_1(Y_1^{(2)} \oplus Y_2^{(2)} \oplus Y_3^{(2)} \oplus Y_4^{(2)}) \oplus \\ \mathrm{lsb}_1(Z_1^{(1)} \oplus Z_2^{(1)} \oplus Z_3^{(1)} \oplus Z_4^{(1)}) \oplus \mathrm{lsb}_1((Z_1^{(2)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus Z_4^{(2)}) \ggg 16)\,, \end{aligned} \tag{12}$$

which has the following properties:

- it is a 2-round iterative linear relation, in contrast to (5) which is 1-round iterative;
- it holds with a probability that depends on the carry bit of addition with the two subkeys in the middle of a block between rounds, $Z_2^{(i)}$ and $Z_3^{(i)}$. The probability that there is no carry from the 15-th to the 16-th bit is $p = 1/2 + 1/2^{17}$. Assuming the subkey values are independent, the probability of (12) holding is approximated as $p^2 \approx 2^{-2}$.

Thus, this new weak-subkey assumption implies another, new weak-key class, namely the one which generates rotations of the form $16 + 32t$, $0 \leq t \leq 3$, distinct from the original weak-key class that generates rotations of the form $32t$, $0 \leq t \leq 3$. The former weak-key class, though, is less effective since it holds with a lower probability, 2^{-2} every two rounds, than the latter.

It is straightforward to deduce other rotation amounts, for example, $8 + 32t$, $0 \leq t \leq 3$, which lead to further new weak-key classes, with exponentially lower probability compared to (5) and (12) due to the carry bits of addition with subkeys. These linear relations become iterative for 4, 8 or more rounds. The rotation amounts do not need to be powers of 2 plus a multiple of 32. More generally, the rotation amounts can be of the form $i + 32t$, $0 \leq t \leq 3$, $0 \leq i \leq 31$. The main point for deducing any of these linear relations is to track the exact position of the least significant bit of each 32-bit word in a block, because, once this bit is correctly located, the xor of 32-bit words in (5) can be applied.

Consequently, the key space can be split into several sets of disjoint weak-key classes, one for each possible set of rotation amounts, and several of them correspond to keys which are susceptible to an attack similar to that in Sect. 4.

4.2 Ciphertext-Only Attack on Ake98

The attack on 4.5-round Ake98 presented in Sect. 4 can also be adapted to recover subkeys at the top (plaintext) end of Ake98. The fact that only a few bits of the plaintext blocks are needed for the attack motivates a ciphertext-only (CO) approach to attack Ake98. We assume that ciphertext is always known by any adversary. Further, assume that the plaintext is known to be ASCII text, and some probable phrases (16 bytes long, that is, the block size of Ake98) are suspected to occur regularly in the plaintext, for instance, "replyimmediately" or "tocommandergeneral".

The ciphertext-only attack on 4.5-round Ake98 assumes that the last four block rotations, including the one in the OT are a multiple of the word size (32 bits), instead of the first four block rotations. The distinguisher is similar to (7):

$$\begin{aligned} &\mathrm{lsb}_1(C_1 \oplus C_2 \oplus C_3 \oplus C_4) \oplus \mathrm{lsb}_1(Z_1^{(2)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus Z_4^{(2)}) \oplus \\ &\mathrm{lsb}_1(Z_1^{(3)} \oplus Z_2^{(3)} \oplus Z_3^{(3)} \oplus Z_4^{(3)}) \oplus \mathrm{lsb}_1(Z_1^{(4)} \oplus Z_2^{(4)} \oplus Z_3^{(4)} \oplus Z_4^{(4)}) \oplus \\ &\mathrm{lsb}_1(Z_1^{(5)} \oplus Z_2^{(5)} \oplus Z_3^{(5)} \oplus Z_4^{(5)}) = \\ &\mathrm{lsb}_1((P_1 \oplus Z_1^{(1)} \oplus (P_2 \boxplus Z_2^{(1)}) \oplus (P_3 \boxplus Z_3^{(1)}) \oplus P_4 \oplus Z_4^{(1)}) \ggg \mathrm{LSB}_5(Z_5^{(1)}))\,. \end{aligned} \tag{13}$$

Thus, (13) only requires the xor of some least significant bits of the plaintext blocks, namely, only some small statistical information. The

time complexity and amount of probable texts for this attack are the same as for the attack in Sect. 4, requiring about 71 (probable) 16-byte long plaintexts encrypted under a fixed key. Similar attacks apply to more rounds of Ake98, under the appropriate weak subkey assumptions.

5 Software Performance of Ake98

Table 2 lists the main parameters and the performance in software of Ake98, AES, IDEA and RC6 block ciphers, for comparison. Performance estimates for encryption and key schedule were measured in CPU cycles per byte encrypted on an AMD Duron 1.2 GHz, 512 MB RAM and 128 MB cache memory, under Linux, and using the gcc compiler ver. 3.2.2 with optimization option -O3. Measurements were obtained from 2^{16} up to 2^{26} blocks encrypted under each cipher.

Table 2. Software performance and main parameters of some block ciphers.

Cipher	Ake98	AES	IDEA	RC6-w/r/b
Operations	$\oplus, \boxplus, \lll$	$\oplus$,xtime,S-box	$\oplus, \boxplus, \odot$	$\oplus$, *, $\lll$
Block Size (bits)	variable	128	64	$4w$
Key Size (bits)	$64t$	$128; 192; 256$	128	$8b$
#Rounds	variable♯	10; 12; 14	8.5	r, $r = 20$ (AES)
Origin	Alvarez *et al.*	Daemen, Rijmen	Lai, Massey, Murphy	Rivest *et al.*
Year	2000	1998	1991	1998
Word Size (bits)	variable	8	16	w, $w = 32$ (AES)
Cipher Structure	IDEA+RC5	SPN	own	Feistel
Key Schedule Oper.	$\boxplus$, modular squaring	$\oplus$, S-box	bit permutation	byte permutation
Reference	[3]	[5]	[9]	[12]
Encryption Speed	73	55	93	30

♯: 4.5 rounds, 128-bit block, 128-bit key.

From Table 2, the performance figures indicate that 4.5-round Ake98 is faster than 8.5-round IDEA, but slower than 10-round AES and 20-round RC6 (standard parameters), under the same test conditions.

Moreover, Table 3 shows that the software performance of Ake98 for increasing number of rounds degrades sharply. For 8 rounds, Ake98 is not faster than any of the three previously mentioned ciphers. For more than 25.5 rounds, Ake98 is not expected to have any weak key, but then it becomes about four times slower than IDEA, eight times slower than AES,

Table 3. Software performance of variable-round Ake98.

# Rounds Ake98	4.5	8.5	12.5	16.5	20.5	24.5	28.5
# CPU cycles/byte	73	142	212	283	354	427	499

and more than 14 times slower than RC6. Moreover, with more than 427 cycles/byte, the performance of Ake98 becames worse than that of all NESSIE block cipher candidates, except GrandCru [1, p. 53–55].

6 Conclusions

This report presented the first[7] known-plaintext and ciphertext-only attacks on Ake98. In a key-recovery attack, the subkeys of the OT can be recovered with only 71 known plaintext/ciphertext pairs. The attacks are independent of the redesigned AR-box, and can be applied up to 11.5 rounds with less effort than an exhaustive key search. To avoid weak keys, Ake98 would need more than 25.5 rounds, but then its performance degrades sharply.

The attacks in Sect. 4 exploited two main weaknesses of Ake98: the key schedule algorithm did not make any provision to avoid the key-dependent rotation amounts to be multiples of 32 (the word size), even for consecutive rounds; moreover, the subkey mixing operations at the beginning of a round allows invariants involving only the least significant text bits, similar to the attack of [8]. These attacks perhaps could be avoided, for example, if the key schedule algorithm had guaranteed that the rotation amounts were both text and key dependent, such as in RC6 [12].

Another important observation is that even if the rotation amounts were properly generated, the encryption and decryption structures of Ake98 would still not be reciprocal, that is, the computational graphs for encryption and decryption of Ake98 are different, because the modular addition and bit rotation operations do not commute. Thus, the computational graph does not become an involution by simply transforming the subkeys, as in IDEA. The existence of weak subkeys for Ake98 are far reaching. Even though the class of 2^{108} weak keys represent only a fraction of 2^{-20} of the key space, it implies for instance, that Ake98 might not be used as a building block of other cryptographic primitives, such as in Davies-Meyer or Matyas-Meyer-Oseas hash function constructions

[7] The authors are not aware of any other attack on Ake98, under any assumption.

[10, p. 340, Cap. 9] because the key input depends on the input message string or intermediate hash values, and they can be manipulated to cause weak rotations as in the attacks of Sect. 4. Further the weak-key class size ($|WKC|$) and type of attacks on IDEA and Ake98 are compared in Table 4. Notice that Hawkes' attacks on IDEA [7] require chosen plaintext (CP), Boomerang attacks on IDEA [2] require chosen plaintext adaptively-chosen ciphertext (CPACC), while the attacks on Ake98, in this paper, require known plaintext (KP) or ciphertext only (CO). It can be noticed additionally in Table 4 that the weak-key class sizes for Ake98 are bigger than for IDEA. Therefore, the attacks on Ake98 apply not only to larger weak-key classes but also work under much more realistic assumptions than on IDEA.

Table 4. Comparison of weak-key class sizes for IDEA and Ake98.

			# Rounds					
Attack	Cipher	Type	4	4.5	5	5.5	6	8.5
Hawkes	IDEA	CP	2^{99}	2^{97}	2^{84}	2^{82}	2^{82}	2^{63}
Boomerang	IDEA	CPACC	2^{104}	2^{103}	2^{97}	2^{97}	2^{83}	2^{64}
this paper	Ake98	KP/CO	2^{108}	2^{103}	2^{103}	2^{98}	2^{98}	2^{83}

Additionally, if Ake98 were used in (full 128-bit) OFB and CFB modes of operation [10], then the use of weak keys at the beginning of every round would result in the exclusive-or of the LSBs of the four input words to match the xor of the LSBs of the four output words. This invariant would propagate to the ciphertext, actually revealing information on the plaintext. Since Ake98 with only 4.5 rounds is already slower than the AES and RC6, even its practical usefulness for confidentiality purposes becomes jeopardized.

As the last comment, it is not straightforward to determine which 128-bit user key(s) lead to subkeys that cause weak rotations at the beginning of each round, but the key schedule algorithm does not have any provision to avoid such weak subkeys. It is left as an open problem to discover which 128-bit Ake98 key(s) can lead to weak subkeys.

Acknowledgements

Many thanks to Prof. S.W. Song of the CS Dept. of the Institute of Mathematics and Statistics of the University of São Paulo, Brazil, for the

kind logistical support for this research, and to the anonymous referees for the many useful comments.

7 Appendix

This appendix shows the AR-boxes (Addition-Rotation boxes) of Akelarre and Ake98 (Fig. 3). For the left-rotation operation, $\lll$, the rotation amounts are 4- or 5-bit values from parts of P_1 and Q_2. In Fig.3(a), rotations affect 32-bit operands. For example, the first rotation of P_2 to the left is by an amount represented by the five bits $P_1[1 \ldots 5]$. In Fig.3(b), the rotations affect operands 31 bits wide, namely excluding the most or the least significant bits (darkened in the pictures). These pictures are described for illustrative purposes only, because the attacks in this paper are independent of the AR-boxes.

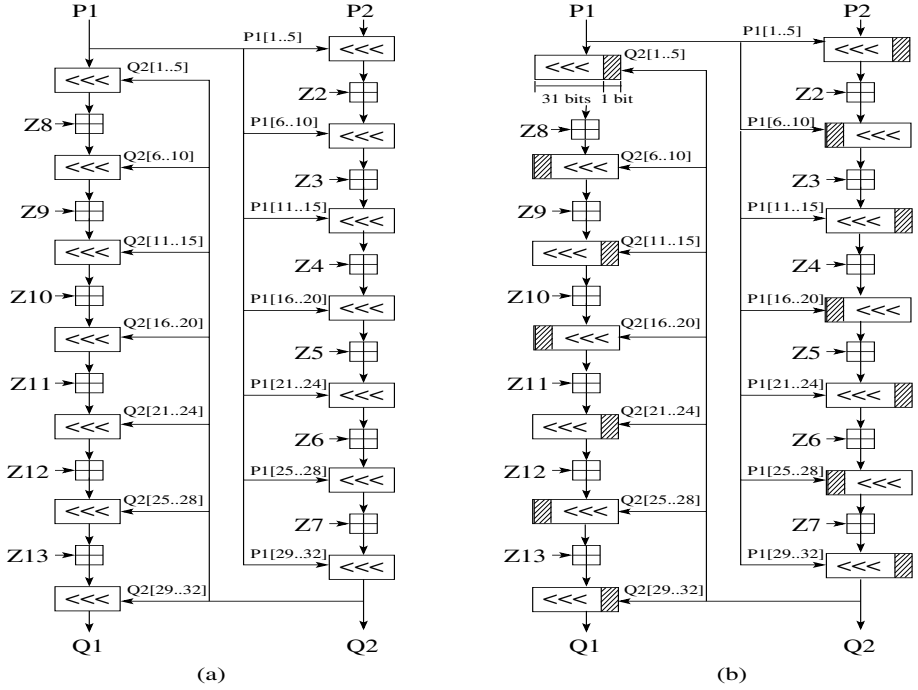

Fig. 3. AR-box of Ake98 (a), and of Akelarre (b).

References

1. E. Biham, "Performance of Optimized Implementations of the NESSIE Primitives," http://cryptonessie.org, Oct. 2002.
2. A. Biryukov, J. Nakahara,Jr., B. Preneel, J. Vandewalle, "New Weak-Key Classes of IDEA," ICICS 2002, R. Deng, S. Qing, F. Bao, J. Zhou, Eds., Springer-Verlag, LNCS 2513, Dec. 2002, 315–326.
3. G. Álvarez Marañón, "Contribución al estudio de la estructura interna del conjunto de Mandelbrot y aplicaciones en criptografía," Facultad de Informática, Universidad Politécnica de Madrid, Sep. 2000, PhD Dissertation, http://www.iec.csic.es/∼gonzalo.
4. G. Alvarez, D. de la Guia, F. Montoya, A. Peinado, "Akelarre: a new Block Cipher Algorithm," 3rd Selected Areas in Cryptography (SAC) Workshop, 1996, 1–14.
5. J. Daemen, V. Rijmen, "AES Proposal: Rijndael," First AES Conference, California, USA, 1998, http://www.nist.gov/aes.
6. N. Ferguson, B. Schneier, "Cryptanalysis of Akelarre," 4th Selected Areas in Cryptography (SAC) Workshop, 1997, 201–212.
7. P.M. Hawkes, "Asymptotic Bounds on Differential Probabilities and an Analysis of the Block Cipher IDEA," PhD Dissertation, The University of Queensland, St. Lucia, Australia, Dec. 1998.
8. L.R. Knudsen and V. Rijmen, "Ciphertext-Only Attack on Akelarre," Cryptologia, vol. XXIV, no. 2, Apr. 2000, 135–147.
9. X. Lai, "On the Design and Security of Block Ciphers," ETH Series in Information Processing, J.L. Massey, Ed., Vol. 1, 1995, Hartung-Gorre Verlag, Konstanz.
10. A.J. Menezes, P.C. van Oorschot, S. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1997.
11. R.L. Rivest, "The RC5 Encryption Algorithm," B. Preneel, Ed., 2nd Fast Software Encryption Workshop, 1995, Springer-Verlag, LNCS 1008, 86–96.
12. R.L. Rivest, M.J.B. Robshaw, R. Sidney, Y.L. Yin, "The RC6 Block Cipher," First AES Conference, California, USA, 1998, http://csrc.nist.gov/encryption/aes/.