

# Attacks on Simplified Versions of K2

Deike Priemuth-Schmid

LACS, University of Luxembourg  
deike.priemuth-schmid@uni.lu

**Abstract.** In 2007, S. Kiyomoto, T. Tanaka and K. Sakurai presented the stream cipher K2 at SECURE. In this paper, we present two attacks on simplified versions of K2. We show a differential chosen IV attack with key recovery on a simplified version with 5 initialization clocks with time complexity of  $2^{8.1}$  clocks. For a simplified version with 7 initialization clocks, we show a distinguishing attack with time complexity of  $2^{34.8}$  clocks.

**Keywords:** cipher K2, stream ciphers, cryptanalysis.

## 1 Introduction

The stream cipher K2 was proposed by S. Kiyomoto, T. Tanaka and K. Sakurai at SECURE 2007 [4].

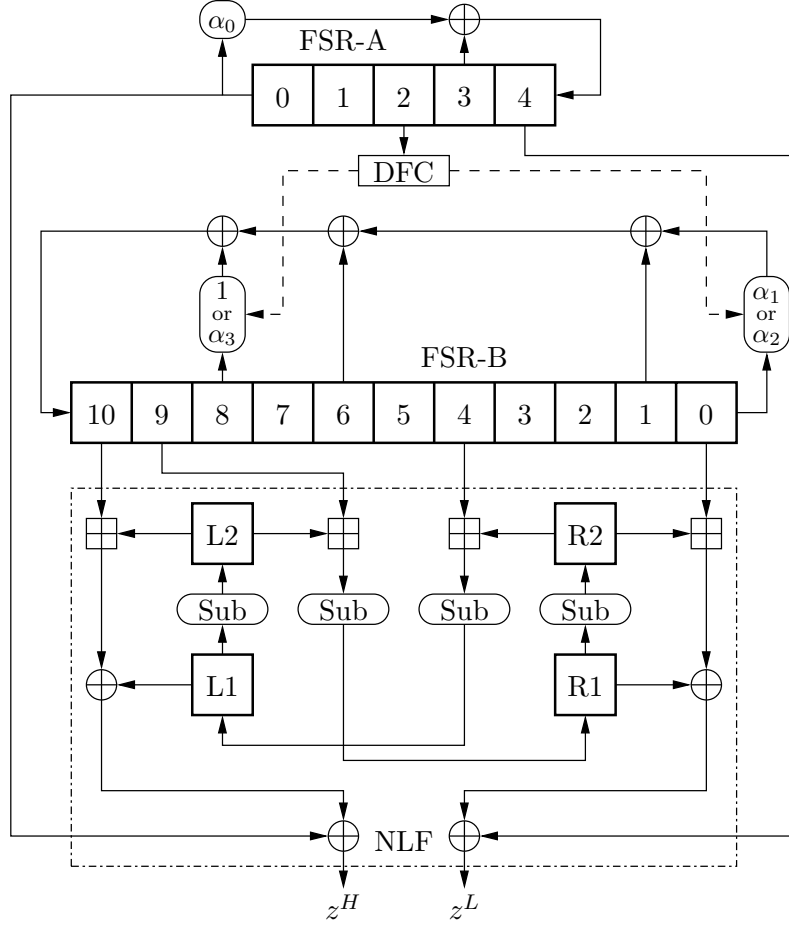
A security evaluation is given in [1] with the conclusion that no weaknesses were found. Some side-channel attacks are applied on K2 in [3] showing that K2 offers reasonable resistance to side-channel attacks even without countermeasures.

In this paper, we present two attacks on simplified versions of K2. We show a differential chosen IV attack with key recovery on a simplified version with 5 initialization clocks with time complexity of  $2^{8.1}$  clocks and needed keystream of 28 words. For a simplified version with 7 initialization clocks, we show a distinguishing attack with time complexity of  $2^{34.8}$  clocks and needed keystream of  $2^{32}$  words. Both attacks have negligible memory requirements.

This paper is organized as follows. We give a description of the cipher K2 and its simplification  $K2^\oplus$  in Section 2. The differential chosen IV attack with key recovery on  $K2^\oplus$  with 5 initialization clocks is presented in Section 3. In Section 4, we describe the distinguishing attack on  $K2^\oplus$  with 7 initialization clocks. Some conclusions are given in Section 5.

## 2 Description of K2 and $K2^\oplus$

S. Kiyomoto, T. Tanaka and K. Sakurai proposed the stream cipher K2 at SECURE 2007 [4]. It consists of two FSRs, a dynamic feedback controller and a nonlinear function as shown in Fig. 1



**Fig. 1.** keystream generation of K2

The first FSR, called FSR-A, has 5 registers ( $a_4, \dots, a_0$ ) each of size one word (32 bit). The feedback function is

$$a_4^t = \alpha_0 a_0^{t-1} \oplus a_3^{t-1}$$

where the multiplier  $\alpha_0$  is a constant, chosen as the root of an irreducible polynomial of degree four in  $GF(2^8)[x]$ . The second FSR, called FSR-B, has 11 registers ( $b_{10}, \dots, b_0$ ) each of size one word. The feedback function of FSR-B is selected by the dynamic feedback controller. This controller has two clock control bits  $cl1$  and  $cl2$  which are described as

$$cl1_t = a_2^t[30] \quad \text{and} \quad cl2_t = a_2^t[31]$$

with  $a_2^t[30]$  being the second most significant bit of  $a_2^t$  and  $a_2^t[31]$  being the most significant bit (abbr. msb) of  $a_2^t$ . Then the feedback function of the FSR-B is

$$b_{10}^t = (\alpha_1^{cl_{1t-1}} + \alpha_2^{1-cl_{1t-1}-1})b_0^{t-1} \oplus b_1^{t-1} \oplus b_6^{t-1} \oplus \alpha_3^{cl_{2t-1}}b_8^{t-1}$$

where the multipliers  $\alpha_{(1,2,3)}$  are constants, each one chosen as the root of a different irreducible polynomial of degree four in  $GF(2^8)[x]$ . The nonlinear function (abbr. NLF) has four words memory ( $L1, L2, R1, R2$ ) and four times a *Sub* function. This *Sub* function operates on a word and uses the 8-bit AES S-box and the AES Mix-Column operation [2]. The exact work flow is: divide the word into four bytes, apply on each byte the 8-bit AES S-box, mix the resulting bytes using the AES Mix-Column operation yielding a word again. To update the memory words of the NLF, compute

$$\begin{aligned} L1^t &= Sub(R2^{t-1} \boxplus b_4^{t-1}) & L2^t &= Sub(L1^{t-1}) \\ R1^t &= Sub(L2^{t-1} \boxplus b_9^{t-1}) & R2^t &= Sub(R1^{t-1}) . \end{aligned}$$

With each clock, the output of the NLF is the keystream of two words ( $z_t^H, z_t^L$ ) computed as

$$\begin{aligned} z_t^H &= (b_{10}^t \boxplus L2^t) \oplus L1^t \oplus a_0^t \\ z_t^L &= (b_0^t \boxplus R2^t) \oplus R1^t \oplus a_4^t . \end{aligned}$$

The symbol ' $\oplus$ ' denotes the bit-wise xor and the symbol ' $\boxplus$ ' denotes addition modulo  $2^{32}$ .

The K2 cipher uses a four word key  $K = [K_0, K_1, K_2, K_3]$  and a four word IV  $IV = [IV_0, IV_1, IV_2, IV_3]$ . For the loading step, two intermediate results are computed using the *Sub* function

$$\begin{aligned} S1 &= Sub[(K_3 \ll 8) \oplus (K_3 \gg 24)] \oplus 0x01000000 \\ S2 &= Sub[((K_0 \oplus K_1 \oplus K_2 \oplus K_3 \oplus S1) \ll 8) \\ &\quad \oplus ((K_0 \oplus K_1 \oplus K_2 \oplus K_3 \oplus S1) \gg 24)] \oplus 0x02000000 , \end{aligned}$$

the constants at the end are given in hexadecimal numbers. Then the FSRs are loaded

$$\begin{aligned} a_0 &= K_0 \oplus S1 & b_3 &= IV_1 \\ a_1 &= K_3 & b_4 &= K_0 \oplus S1 \oplus S2 \\ a_2 &= K_2 & b_5 &= K_1 \oplus S2 \\ a_3 &= K_1 & b_6 &= IV_2 \\ a_4 &= K_0 & b_7 &= IV_3 \\ b_0 &= K_0 \oplus K_2 \oplus S1 \oplus S2 & b_8 &= K_0 \oplus K_1 \oplus K_2 \oplus K_3 \oplus S1 \\ b_1 &= K_1 \oplus K_3 \oplus S2 & b_9 &= K_0 \oplus K_1 \oplus S1 \\ b_2 &= IV_0 & b_{10} &= K_0 \oplus K_1 \oplus K_2 \oplus S1 . \end{aligned}$$

The four memory words of the NLF are initialized with zero. Then, during the initialization the cipher is clocked 24 times doing

1. get the output from the NLF  $(z_t^H, z_t^L)$ ,
2. update the NLF,
3. update FSR-A with  $z_t^L$  is xored to the new word of FSR-A,
4. update FSR-B with  $z_t^H$  is xored to the new word of FSR-B.

After this initialization, the K2 cipher produces the keystream and the FSRs are updated without feedback from the NLF.

For the rest of the paper, we consider a simplified version of K2 where all additions modulo  $2^{32}$  are replaced with xor and denote this version with  $K2^\oplus$ .

In our attacks, we only need to compute equations similar to the equations computed for the keystream and the update of the FSRs. In each clock of  $K2^\oplus$ , four such equations are computed (two equations for the keystream and two equations for the FSRs update). We measure the time complexity for our attacks in  $K2^\oplus$  clocks.

### 3 Differential Chosen IV Attack with Key Recovery

Considering a differential chosen IV scenario, we choose two different IVs  $IV_a$  and  $IV_b$ . We know the keystream from both pairs  $(K, IV_a)$  and  $(K, IV_b)$  with unknown key  $K$ . Both IVs only differ in IV word  $IV_1$  which takes the longest until it enters the NLF. The dynamic feedback controller always takes the most and second most significant bit of  $a_2$ . Thus, we do not want to have a difference there. Accordingly, we choose the starting difference  $\Delta d$  with most and second most significant bit equal to zero. Our goal is to recover the whole internal state right after the loading step which means we get the unknown key  $K$ .

From the differences in the keystream and the partially known differences in the FSRs, we compute the differences in the words of the NLF. We then need to know how the differences in the NLF words propagate through the *Sub* function. Let  $v_a^{t-1}$  and  $v_b^{t-1}$  be two arbitrary words at time  $t-1$  with the following equations

$$\begin{aligned} \Delta v^{t-1} &= v_a^{t-1} \oplus v_b^{t-1}, & w_a^t &= \text{Sub}(v_a^{t-1}), & w_b^t &= \text{Sub}(v_b^{t-1}), \\ \Delta w^t &= w_a^t \oplus w_b^t = \text{Sub}(v_a^{t-1}) \oplus \text{Sub}(v_b^{t-1}). \end{aligned}$$

We define a new notation

$$\overset{\text{out}}{\Delta} \text{Sub}(\Delta v^{t-1}) \stackrel{\text{def.}}{=} \text{Sub}(v_a^{t-1}) \oplus \text{Sub}(v_b^{t-1}).$$

During the keystream generation, we have the following equations for the differences at clock  $t$

$$\begin{aligned} \Delta z_t^H &= \Delta b_{10}^t \oplus \Delta L1^t \oplus \Delta L2^t \oplus \Delta a_0^t & \Delta z_t^L &= \Delta b_0^t \oplus \Delta R1^t \oplus \Delta R2^t \oplus \Delta a_4^t \\ \Delta L1^t &= \overset{\text{out}}{\Delta} \text{Sub}(\Delta R2^{t-1} \oplus \Delta b_4^{t-1}) & \Delta R1^t &= \overset{\text{out}}{\Delta} \text{Sub}(\Delta L2^{t-1} \oplus \Delta b_9^{t-1}) \\ \Delta L2^t &= \overset{\text{out}}{\Delta} \text{Sub}(\Delta L1^{t-1}) & \Delta R2^t &= \overset{\text{out}}{\Delta} \text{Sub}(\Delta R1^{t-1}). \end{aligned}$$

Any fixed input difference  $(\Delta R2^{t-1} \oplus \Delta b_4^{t-1}) \neq 0$  results in nearly  $2^{28}$  possible output differences  $\Delta L1^t$ , because any input difference in the small 8-bit AES S-box results in 127 different output differences. If we know or fix the input-output

difference of  $Sub$  at clock  $t - 1$  and  $t$  meaning  $(\Delta R2^{t-1} \oplus \Delta b_4^{t-1}) \xrightarrow{Sub} \Delta L1^t$ , we can recover on average  $(2 \cdot \frac{126}{127} + 4 \cdot \frac{1}{127})^4 = 16.51$  sorted pairs of individual values for  $Sub$ . This means that we have  $\frac{16.51}{2} \approx 8$  possible values for  $\Delta L2^{t+1}$ . If we know or fix this difference as well, then we have only one sorted pair of individual values left which satisfies the sequence  $(\Delta R2^{t-1} \oplus \Delta b_4^{t-1}) \xrightarrow{Sub} \Delta L1^t \xrightarrow{Sub} \Delta L2^{t+1}$ .

The same holds for the sequence  $(\Delta L2^{t-1} \oplus \Delta b_9^{t-1}) \xrightarrow{Sub} \Delta R1^t \xrightarrow{Sub} \Delta R2^{t+1}$ . If we have collected enough individual values for the NLF, we can derive some words for the FSR-B from the update equations  $L1^t = Sub(R2^{t-1} \oplus b_4^{t-1})$  and  $R1^t = Sub(L2^{t-1} \oplus b_9^{t-1})$  of the NLF. The insertion of the individual values of the NLF together with some words of FSR-B in the keystream equations yields some words for FSR-A. At the end, we know enough words of the NLF, FSR-B and FSR-A to clock backwards and reveal the secret key.

We reduce the number of initialization clocks of  $K2^\oplus$  to 5. After these 5 clocks of initialization, the starting difference  $\Delta d$  enters the word  $R1$  of the NLF. During the keystream computation, there is no more feedback from the NLF to the FSRs anymore. Therefore, we know all differences of FSR-A as  $\Delta d$  enters it in clock 4 and then propagates linearly. For FSR-B, the computation of the differences only depends on the unknown bits of the dynamic feedback controller which select the multipliers for the FSR-B feedback.

The work flow of K2 has two steps, first displaying the keystream words, then updating the internal state. Thus, the differences of the keystream words for clock 0 are

$$\begin{aligned}\Delta z_0^H &= \Delta b_{10}^0 \oplus \Delta L1^0 \oplus \Delta L2^0 \oplus \Delta a_0^0 \\ \Delta z_0^L &= \Delta b_0^0 \oplus \Delta R1^0 \oplus \Delta R2^0 \oplus \Delta a_4^0 ,\end{aligned}$$

where the differences in  $\Delta L1^0, \Delta L2^0, \Delta a_0^0, \Delta b_0^0, \Delta R2^0$  are zero and  $\Delta a_4^0 = \Delta d$ . Hence, we know  $\Delta b_{10}^0$  and  $\Delta R1^0$ . The update equation of FSR-B implies  $\Delta b_{10}^0 = \Delta b_{10}^{-1} = \Delta b_9^0$  and  $\Delta b_{10}^0 = \Delta b_9^1$ . With  $\Delta R1^0$ , we know the input-output sequence  $(\Delta L2^{-1} \oplus \Delta b_9^{-1}) \xrightarrow{Sub} R1^0$  and as explained above on average we can recover 16.51 sorted pairs of individual values for  $Sub$  resulting in 8 possible values for  $R2^1$ .

Clock 1 gives us  $\Delta z_1^H = \Delta b_{10}^1 = \Delta b_9^2$  because all other differences are zero. For  $\Delta z_1^L$ , we can rewrite the keystream equation in the following way

$$\begin{aligned}\Leftrightarrow \quad & \Delta z_1^L = \Delta b_0^1 \oplus \Delta R1^1 \oplus \Delta R2^1 \oplus \Delta a_4^1 \\ & \Delta R1^1 = \Delta R2^1 \oplus \Delta b_0^1 \oplus \Delta a_4^1 \oplus \Delta z_1^L \\ \Leftrightarrow \quad & \overset{\text{out}}{\Delta} Sub(\Delta L2^0 \oplus \Delta b_9^0) = \Delta R2^1 \oplus \Delta b_0^1 \oplus \Delta a_4^1 \oplus \Delta z_1^L ,\end{aligned}$$

that we know all differences at the right side. Here we can insert all 8 possibilities of  $\Delta R2^1$ , then undo the Mix Column operation by multiplying with its inverse and check byte by byte whether the computed value on the right side is a valid difference for the left side. The time complexity for this check is 2 clocks and only one pair  $(\Delta R1^1, \Delta R2^1)$  will remain due to  $\frac{2^{28} \cdot 8}{2^{32}} < 1$ . The known difference  $\Delta R2^1$  leaves only one sorted pair which fulfills the sequence

$(\Delta L2^{-1} \oplus \Delta b_9^{-1}) \xrightarrow{Sub} \Delta R1^0 \xrightarrow{Sub} \Delta R2^1$ . The known difference  $\Delta R1^1$  fixes the sequence  $(\Delta L2^0 \oplus \Delta b_9^0) \xrightarrow{Sub} \Delta R1^1$  which results in nearly 16.51 sorted pairs of individual values following 8 possible differences for  $R2^2$ .

In clock 2,  $\Delta b_{10}^2$  has two possibilities because in its update equation

$$\Delta b_{10}^2 = (\alpha_1^{cl1_1} + \alpha_2^{1-cl1_1-1})\Delta b_0^1 \oplus \Delta b_1^1 \oplus \Delta b_6^1 \oplus \alpha_3^{cl2_1} \Delta b_8^1$$

all differences are equal to zero except for  $\Delta b_8^1$ . Hence, we have  $\Delta b_{10}^2 = \Delta b_8^1$  or  $\Delta b_{10}^2 = \alpha_3 \Delta b_8^1$  because we have a zero in the msb of  $\Delta a_2^1 = \Delta d$ . We can rewrite the keystream equation in the following way

$$\begin{aligned} \Delta z_2^H &= \Delta b_{10}^2 \oplus \Delta L1^2 \oplus \Delta L2^2 \oplus \Delta a_0^2 \\ \Leftrightarrow \Delta L1^2 &= \Delta b_{10}^2 \oplus \Delta L2^2 \oplus \Delta a_0^2 \oplus \Delta z_2^H \\ \Leftrightarrow \overset{\text{out}}{\Delta} Sub(\Delta R2^1 \oplus \Delta b_4^1) &= \Delta b_{10}^2 \oplus \Delta L2^2 \oplus \Delta a_0^2 \oplus \Delta z_2^H, \end{aligned}$$

that we know all differences at the right side. We insert both possibilities for  $\Delta b_{10}^2$ , undo the Mix Column operation and check byte by byte whether the computed value on the right side is a valid difference for the left side. The time complexity for this check is  $\frac{1}{2}$  clock and only one pair  $(\Delta b_{10}^2, \Delta L1^2)$  will remain. We know  $\Delta b_{10}^2 = \Delta b_9^3$  and  $\Delta L1^2$  fixes the sequence  $(\Delta R2^1 \oplus \Delta b_4^1) \xrightarrow{Sub} \Delta L1^2$  which results in nearly 16.51 sorted pairs of individual values following 8 possibilities for  $\Delta L2^3$ . For  $\Delta z_2^L$ , we do exactly the same as described for  $\Delta z_1^L$  at clock 1.

For clock 3, we have two possibilities for  $\Delta b_{10}^3$  and 8 possibilities for  $\Delta L2^3$ . Thus, we rewrite the equation

$$\begin{aligned} \Delta z_3^H &= \Delta b_{10}^3 \oplus \Delta L1^3 \oplus \Delta L2^3 \oplus \Delta a_0^3 \\ \Leftrightarrow \overset{\text{out}}{\Delta} Sub(\Delta R2^2 \oplus \Delta b_4^2) &= \Delta b_{10}^3 \oplus \Delta L2^3 \oplus \Delta a_0^3 \oplus \Delta z_3^H, \end{aligned}$$

insert all listed possibilities, undo the Mix Column operation and check byte by byte. The time complexity for this check is 4 clocks and only one triple  $(\Delta b_{10}^3, \Delta L1^3, \Delta L2^3)$  will remain due to  $\frac{2^{28} \cdot 2 \cdot 8}{2^{32}} = 1$ . We know  $\Delta b_{10}^3 = \Delta b_9^4$  and  $\Delta L1^3$  fixes the sequence  $(\Delta R2^2 \oplus \Delta b_4^2) \xrightarrow{Sub} \Delta L1^3$  which results in 8 possibilities for  $\Delta L2^4$ . The known difference  $\Delta L2^3$  fixes the pair of individual values for the sequence  $(\Delta R2^1 \oplus \Delta b_4^1) \xrightarrow{Sub} \Delta L1^2 \xrightarrow{Sub} \Delta L2^3$ . For  $\Delta z_3^L$ , we do exactly the same as described for  $\Delta z_1^L$  at clock 1.

In clock 4, 5 and 6 we do exactly the same as described for clock 3.

Now we have collected enough individual values for the NLF. So far, the time complexity is  $2 + \frac{1}{2} + 2 + 4 \cdot 4 = 20.5$  clocks.

We have to insert these individual values in the keystream and update equations. Since we collected 10 sorted pairs of individual values and two keystreams with the corresponding system of equations, we need to decide for each value to be filled in into which equation system. Wrong allocations will have contradictions somewhere in the equations and are dispelled this way. This would result a time complexity of  $2^8$  clocks. For the right allocation at clock 6, we can clock backwards and receive the secret key with time complexity of 6 clocks.

The overall time complexity in  $K2^\oplus$  clocks is

$$20.5 + 2^8 + 6 = 2^{8.1} .$$

The needed keystream amounts to 2 words per clock for 7 clocks for each pair  $(K, IV_a)$  and  $(K, IV_b)$  which yields 28 keystream words. The memory requirements are negligible.

## 4 Distinguishing Attack

We now consider  $K2^\oplus$  with the number of initialization clocks reduced to 7. To distinguish the cipher from a random function, we build a multiset over all possible  $2^{32}$  values of one word using  $2^{32}$  different key-IV-pairs and check whether the xored sum over all first keystream words  $z_0^H$  is equal to zero. For a random function the probability is  $2^{-32}$  that this sum is zero.

For all  $2^{32}$  key-IV-pairs, we take the same unknown key. The four words of the  $IV = [IV_0, IV_1, IV_2, IV_3]$  are loaded in the FSR-B; where the word  $IV_1$  loaded in  $b_3$  takes the longest time until it enters the nonlinear function. For this word  $IV_1$ , we make a multiset in a way that all values  $[0, 2^{32} - 1]$  occur exactly once. We emphasize that we need the multiset in ascending order starting with zero. Then we know that all  $IV_1$  values in the first half have msb equal to zero whereas all  $IV_1$  values in the second half have msb equal to one. We will use this fact about the msb later. The multiset in  $IV_1$  yields  $2^{32}$  different IVs  $IV^i = [IV_0, i, IV_2, IV_3]$ ,  $i = 0, \dots, 2^{32} - 1$  with arbitrary words  $IV_{(0,2,3)}$ . For each pair (key,  $IV^i$ ), we run the  $K2^\oplus$  cipher with 7 initialization clocks and get the first keystream word  $^i z_0^H$ .

Now we explain why the xored sum over all keystream words is equal to zero. Our goal is to prove the multiset propagation through  $K2^\oplus$  as shown in Table 1. In this table, we only show the multiset and its behavior. All values which are the same in all  $2^{32}$  key-IV-pairs are omitted (empty in the table). We put the '?' for those sets we do not know anything about. With the symbol 'M<sup>s</sup>', we denote the multiset in the starting order. The symbols 'M<sup>1</sup>', 'M<sup>2</sup>' and 'M<sup>3</sup>' denote different multisets. In each of them, each value  $[0, 2^{32} - 1]$  occurs exactly once, but we do not know in which order. Thus, also the xored sums over these multisets are zero. The symbol 'S0' denotes a multiset, where the characteristic that each value  $[0, 2^{32} - 1]$  occurs exactly once is lost, but the feature that it sums up (xored) to zero still remains. To prove this feature for the 'S0' multisets, it is sufficient to consider the sets in  $b_{10}^t$ . We will check them in reverse order. The update equation for the xored sum over all  $^i b_{10}^t$  is

$$\begin{aligned} \sum_{i=0}^{2^{32}-1} ^i b_{10}^t = \sum_{i=0}^{2^{32}-1} \left( ^i m_{msb}^{t-1} ^i b_8^{t-1} \oplus ^i b_6^{t-1} \oplus ^i b_1^{t-1} \oplus ^i m_{msb}^{t-1} ^i b_0^{t-1} \right. \\ \left. \oplus ^i b_{10}^{t-1} \oplus ^i L2^{t-1} \oplus ^i L1^{t-1} \oplus ^i a_0^{t-1} \right) . \end{aligned} \quad (1)$$

Here, the variable  $^i m_{msb}^{t-1}$  denotes the multiplier depending on the msb of  $^i a_2^{t-1}$ . In particular,  $^i m_{msb}^{t-1} = \alpha_3$  if the msb of  $^i a_2^{t-1}$  is equal to one and  $^i m_{msb}^{t-1} = 1$  if

msb=0. Likewise, the variable  ${}^i m_{msb}^{t-1}$  denotes the multiplier depending on the second most significant bit of  ${}^i a_2^{t-1}$ . In particular,  ${}^i m_{msb}^{t-1} = \alpha_1$  if the second most significant bit of  ${}^i a_2^{t-1}$  is equal to one and  ${}^i m_{msb}^{t-1} = \alpha_2$  otherwise.

**Table 1.** Propagation of the multiset through K2<sup>⊕</sup> during Initialization

clock	FSR-B										FSR-A					NLF				
ini	10	9	8	7	6	5	4	3	2	1	0	4	3	2	1	0	L1	L2	R1	R2
0	M <sup>s</sup>																			
1	M <sup>s</sup>																			
2	M <sup>s</sup>																			
3	M <sup>s</sup>	M <sup>s</sup>																		
4	S0	M <sup>s</sup>										M <sup>s</sup>								
5	S0	S0	M <sup>s</sup>								M <sup>s</sup>	M <sup>s</sup>				M <sup>1</sup>				
6	S0	S0	S0	M <sup>s</sup>							S0	M <sup>s</sup>	M <sup>s</sup>				?		M <sup>2</sup>	
7	S0	S0	S0	S0	M <sup>s</sup>						?	S0	M <sup>s</sup>	M <sup>s</sup>			M <sup>3</sup>	?		?

From now on, we mean always xored sum when we write sum or the sigma sign. The keystream is produced at clock zero after the initialization. For the internal states, we write the number of the initialization clock as superscript, because we need to go backwards through the initialization clocks to prove the multiset propagation. We know about the sum of all words  ${}^i z_0^H$

$$\sum_{i=0}^{2^{32}-1} {}^i z_0^H = \sum_{i=0}^{2^{32}-1} \left( {}^i b_{10}^7 \oplus {}^i L2^7 \oplus {}^i L1^7 \oplus {}^i a_0^7 \right).$$

For all  $2^{32}$  key-IV-pairs, the values for  $L2^7$  and  $a_0^7$  remain constant for all pairs meaning the sum over them is zero. The values  ${}^i L1^7$  form a multiset in which each value  $[0, 2^{32} - 1]$  occurs exactly once, but in an unknown order. The *Sub* function preserves this property that each value  $[0, 2^{32} - 1]$  occurs exactly once, but destroys the known starting order. Thus we have

$$\sum_{i=0}^{2^{32}-1} {}^i z_0^H = \sum_{i=0}^{2^{32}-1} {}^i b_{10}^7. \quad (2)$$

Now we will prove that the sum over all  ${}^i b_{10}^7$  is zero. We look up the values for (I) at clock 7 in Table I and see which values remain constant meaning the sum over them is zero. We omit those zero sums and get (II) for clock 7

$$\sum_{i=0}^{2^{32}-1} {}^i b_{10}^7 = \sum_{i=0}^{2^{32}-1} {}^i m_{msb}^6 {}^i b_8^6 \oplus \sum_{i=0}^{2^{32}-1} {}^i m_{msb}^6 {}^i b_0^6 \oplus \sum_{i=0}^{2^{32}-1} {}^i b_{10}^6. \quad (3)$$

We take a closer look on the sum over  ${}^i b_{10}^6$ . After omitting all zero sums (empty values in Table I), (II) is

$$\sum_{i=0}^{2^{32}-1} {}^i b_{10}^6 = \sum_{i=0}^{2^{32}-1} {}^i m_{msb}^5 {}^i b_8^5 \oplus \sum_{i=0}^{2^{32}-1} {}^i m_{msb}^5 {}^i b_0^5 \oplus \sum_{i=0}^{2^{32}-1} {}^i b_{10}^5.$$



The choice for the multipliers depends on the value of  $a_2^5$  which remains constant. Thus, we do not know which multiplier is chosen but we know it is the same for all  $2^{32}$  key-IV-pairs. The values for  $b_0^5$  remain constant meaning the sum is zero. Therefore, our equation reduces to

$$\begin{aligned} \sum_{i=0}^{2^{32}-1} {}^i b_{10}^6 &= m_{msb}^5 \sum_{i=0}^{2^{32}-1} {}^i b_8^5 \oplus \sum_{i=0}^{2^{32}-1} {}^i b_{10}^5 \\ &= m_{msb}^5 \sum_{i=0}^{2^{32}-1} {}^i b_{10}^3 \oplus \sum_{i=0}^{2^{32}-1} {}^i b_{10}^5 . \end{aligned} \quad (4)$$

For the sum over all  ${}^i b_{10}^5$ , (1) reduces to

$$\sum_{i=0}^{2^{32}-1} {}^i b_{10}^5 = \sum_{i=0}^{2^{32}-1} {}^i b_{10}^4 \quad (5)$$

because the choice for the multipliers is constant due to the constant value of  $a_2^4$ . For clock 4, the choice for the multipliers in (1) is constant as well yielding

$$\sum_{i=0}^{2^{32}-1} {}^i b_{10}^4 = m_{msb}^3 \sum_{i=0}^{2^{32}-1} {}^i b_0^3 \oplus \sum_{i=0}^{2^{32}-1} {}^i b_{10}^3 . \quad (6)$$

With the constant choice for the multipliers at clock 3, (1) reduces to

$$\sum_{i=0}^{2^{32}-1} {}^i b_{10}^3 = \sum_{i=0}^{2^{32}-1} {}^i b_1^2 . \quad (7)$$

Now we take (5), (6) and (7) and include them in (4) which yields

$$\begin{aligned} \sum_{i=0}^{2^{32}-1} {}^i b_{10}^6 &= m_{msb}^5 \sum_{i=0}^{2^{32}-1} {}^i b_1^2 \oplus m_{msb}^3 \sum_{i=0}^{2^{32}-1} {}^i b_0^3 \oplus \sum_{i=0}^{2^{32}-1} {}^i b_1^2 \\ &= m_{msb}^5 \sum_{i=0}^{2^{32}-1} {}^i b_3^0 \oplus m_{msb}^3 \sum_{i=0}^{2^{32}-1} {}^i b_3^0 \oplus \sum_{i=0}^{2^{32}-1} {}^i b_3^0 \\ &= m_{msb}^5 \sum_{i=0}^{2^{32}-1} i \oplus m_{msb}^3 \sum_{i=0}^{2^{32}-1} i \oplus \sum_{i=0}^{2^{32}-1} i = 0 . \end{aligned} \quad (8)$$

Here, we can directly see our multiset and know that the sum is zero.

Equation (3) now has only two summands left both depending on the value of  $a_2^6$ . From the update equation for  $a_2^6 = a_4^4$ , we know

$$\sum_{i=0}^{2^{32}-1} {}^i a_4^4 = \sum_{i=0}^{2^{32}-1} \left( {}^i a_3^3 \oplus \alpha_0 {}^i a_0^3 \oplus {}^i b_0^3 \oplus {}^i R2^3 \oplus {}^i R1^3 \oplus {}^i a_4^3 \right) .$$

The values for  $a_3^3$ ,  $a_0^3$ ,  $R2^3$ ,  $R1^3$  and  $a_4^3$  remain constant for all pairs. We denote the sum of them with  $C = a_3^3 \oplus \alpha_0 a_0^3 \oplus R2^3 \oplus R1^3 \oplus a_4^3$  which does not depend on  $i$  and is unknown to us. With this simplification, we obtain

$$\sum_{i=0}^{2^{32}-1} i a_4^4 = \sum_{i=0}^{2^{32}-1} (i b_0^3 \oplus C) = \sum_{i=0}^{2^{32}-1} (i \oplus C) . \quad (9)$$

As a result of the multiset, we know that in each bit of  $i$  the number of ones and zeros occurring is exactly  $2^{31}$  which is an even number. This means that the second most significant bit of  $a_4^4$  has  $2^{31}$  ones and  $2^{31}$  zeros. Taking this and the fact that the value of  $b_0^6$  remains constant, we obtain for the second summand of (3)

$$\sum_{i=0}^{2^{32}-1} i m_{msb}^6 i b_0^6 = \sum_{i=0}^{2^{31}-1} \alpha_1 b_0^6 \oplus \sum_{i=0}^{2^{31}-1} \alpha_2 b_0^6 = 0 . \quad (10)$$

Altogether (3) simplifies with (8) and (10) to

$$\sum_{i=0}^{2^{32}-1} i b_{10}^7 = \sum_{i=0}^{2^{32}-1} i m_{msb}^6 i b_8^6 = \sum_{i=0}^{2^{32}-1} i m_{msb}^6 i b_{10}^4 \quad (11)$$

where the choice of the multiplier depends on the msb of the value  $a_2^6 = a_4^4$  and the value of  $b_{10}^4$  does not remain constant. We emphasized at the beginning the ascending order of our multiset. This means that the first half of our multiset has msb zero and the second half has msb one. Accordingly, the msb of all  $i a_4^4$  with  $i = 0, \dots, 2^{31} - 1$  is the msb of unknown constant  $C$ , see (9), whereas the msb of all  $i a_4^4$  with  $i = 2^{31}, \dots, 2^{32} - 1$  is the opposite of the msb of unknown constant  $C$ . Thus, we divided the sum into a first and a second half. We need to check whether we can also divide the set of  $i b_{10}^4$ . From (6) with (7) we know

$$\sum_{i=0}^{2^{32}-1} i b_{10}^4 = m_{msb}^3 \sum_{i=0}^{2^{32}-1} i b_0^3 \oplus \sum_{i=0}^{2^{32}-1} i b_1^2 = m_{msb}^3 \sum_{i=0}^{2^{32}-1} i \oplus \sum_{i=0}^{2^{32}-1} i .$$

The choice of the multipliers is constant but unknown to us. We can divide both sums into a first and a second half.

Now with the two possibilities of the msb of  $a_4^4$  (first half zero and second half one, or the other way around), for (11) we can compute two values

$$\begin{aligned} X_1 &= \left( m_{msb}^3 \sum_{i=0}^{2^{31}-1} i \oplus \sum_{i=0}^{2^{31}-1} i \right) \oplus \alpha_3 \left( m_{msb}^3 \sum_{i=2^{31}}^{2^{32}-1} i \oplus \sum_{i=2^{31}}^{2^{32}-1} i \right) \\ X_2 &= \alpha_3 \left( m_{msb}^3 \sum_{i=0}^{2^{31}-1} i \oplus \sum_{i=0}^{2^{31}-1} i \right) \oplus \left( m_{msb}^3 \sum_{i=2^{31}}^{2^{32}-1} i \oplus \sum_{i=2^{31}}^{2^{32}-1} i \right) . \end{aligned}$$

For the sums over exactly one ordered half of the multiset the property of summing up to zero is preserved. Accordingly, the values  $X_1$  and  $X_2$  are both zero.

Thus, we have proven the assumed multiset propagation through  $K2^\oplus$  which results in

$$\sum_{i=0}^{2^{32}-1} i z_0^H = \sum_{i=0}^{2^{32}-1} i b_{10}^7 = 0 .$$

We have shown that we can distinguish the  $K2^\oplus$  from a random function with probability  $1 - 2^{-32}$ . The time complexity is  $2^{32} \cdot 7 \approx 2^{34.8}$  clocks of  $K2^\oplus$ . We need the first keystream word for all  $2^{32}$  pairs (key,  $IV^i$ ) with ( $i = 0, \dots, 2^{32}-1$ ). The memory requirements are negligible.

## 5 Conclusions

We have shown a differential chosen IV attack with key recovery on  $K2^\oplus$  with 5 initialization clocks. The complexity for this attack is  $2^{8.1}$  clocks of  $K2^\oplus$  with needed keystream of 28 words and negligible memory requirements. The extension of this attack to 6 or 7 initialization clocks is the topic of ongoing research.

A distinguishing attack on  $K2^\oplus$  with 7 initialization clocks is also presented. With a multiset and its predictable propagation through these 7 clocks, we can distinguish  $K2^\oplus$  from a random function with probability  $1 - 2^{-32}$ . The complexity for this attack is  $2^{34.8}$  clocks of  $K2^\oplus$  with needed keystream of  $2^{32}$  words and negligible memory requirements. We can not extend this attack to 8 initialization clocks because in  $L1^8$  we will get a set we do not know anything about and therefore resulting in a random sum.

**Acknowledgements.** I would like to thank Alex Biryukov and Ralf-Philipp Weinmann for helpful comments.

## References

1. Bogdanov, A., Preneel, B., Rijmen, V.: Security Evaluation of the K2 Stream Cipher (March 2011), [http://www.cryptrec.go.jp/estimation/techrep\\_id2010\\_2.pdf](http://www.cryptrec.go.jp/estimation/techrep_id2010_2.pdf)
2. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Heidelberg (2002)
3. Henriksen, M., Yap, W.S., Yian, C.H., Kiyomoto, S., Tanaka, T.: Side-Channel Analysis of the K2 Stream Cipher. In: Steinfeld, R., Hawkes, P. (eds.) ACISP 2010. LNCS, vol. 6168, pp. 53–73. Springer, Heidelberg (2010)
4. Kiyomoto, S., Tanaka, T., Sakurai, K.: K2: A stream cipher algorithm using dynamic feedback control. In: Hernando, J., Fernández-Medina, E., Malek, M. (eds.) SECRYPT, pp. 204–213. INSTICC Press (2007)