On the Sliding Property of SNOW 3G and SNOW 2.0

Aleksandar Kircanski and Amr M. Youssef
Concordia University, Institute for Information Systems Engineering,
1515 St. Catherine Street West, S-EV 007.638,
Montreal, Quebec, CANADA, H3G 2W1
email: {a_kircan,youssef}@ciise.concordia.ca

Abstract

SNOW 3G is a stream cipher chosen by the 3rd Generation Partnership Project (3GPP) as a crypto-primitive to substitute KASUMI in case its security is compromised. SNOW 2.0 is one of the stream ciphers chosen for the ISO/IEC standard IS 18033-4. In this paper, we show that the initialization procedure of the two ciphers admits a sliding property, resulting in several sets of related-key pairs. In case of SNOW 3G, a set of 2^{32} related key pairs is presented, whereas in case of SNOW 2.0, several such sets are found, out of which the largest are of size 2^{64} and 2^{192} for the 128-bit and 256-bit variant of the cipher, respectively. In addition to allowing related-key key recovery attacks against SNOW 2.0 with 256-bit keys, the presented properties reveal non-random behavior which yields related-key distinguishers and also questions the validity of the security proofs of protocols that are based on the assumption that SNOW 3G and SNOW 2.0 behave like perfect random functions of the key-IV.

Index Terms

cryptography, cryptanalysis, stream cipher, 3GPP, ISO/IEC, SNOW 3G, SNOW 2.0, wireless networks security

I. Introduction

In response to concerns about the security of the 3GPP encryption primitive KASUMI [1], [2] (see also [3]), the Security Algorithms Group of Experts (SAGE) proposed a possible replacement for KASUMI which is currently used in 3G systems as a component of the UEA1 confidentiality algorithm. The core primitive of the new confidentiality algorithm, UEA2, is the SNOW 3G stream cipher [4]. The design of SNOW 3G is based on SNOW 2.0 [5], a stream cipher which is chosen for the ISO/IEC standard IS 18033-4 along with Decim [6], MUGI [7] and Rabbit [8]. SNOW 3G passed extensive internal cryptanalytic efforts, surveyed in [9], but the full evaluation has not been released to public. Externally, SNOW 3G was analyzed in [10].

Biham *et al.* [2] showed that KASUMI does not behave randomly when examined in the related-key model. As stated in [2], this renders the previous security proofs based on the assumption that KASUMI behaves like a perfect random function [11] as invalid and puts into question the security of the whole 3GPP system. In [12], [13] the sliding properties of stream ciphers were used to find sets of related keys where it was shown that a stream cipher may be slidable, in the sense that there exist key-IV values such that the inner state of the cipher at some time $t > 0$ corresponds to another key-IV value. Such key-IV pairs produce equal keystreams up to a slide by some number of positions and represent related keys.

In this paper, we show that a similar strategy is also applicable to SNOW 3G and SNOW 2.0 due to the way the key and the IV are written to the inner state before the first initialization step. More precisely, we show that it is possible to find key-IV pairs such that after iterating the cipher for several initialization steps, the inner state represents a *starting* inner state for some other key-IV value. Due to the nature of the of the initialization processes of SNOW 3G and SNOW 2.0, such related keys do not generate slid keystreams, but only keystreams that have several equal words. However, this still allows distinguishing

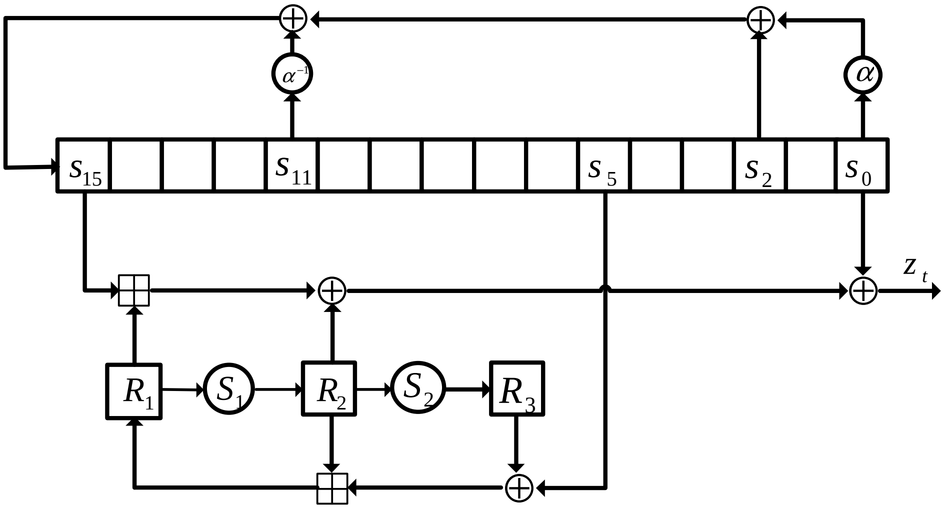

Fig. 1. The SNOW 3G stream cipher

the produced keystream from random keystreams. Table I shows the number of related key pairs, when IV is fixed, for different SNOW variants.

A feature of the related keys presented in this paper is that, given a key for which a corresponding key pair exists, it is straightforward to derive this related key, as opposed to related keys from [13] where the relation was non-obvious and the keys corresponded to a solution of a complex system of equations. Furthermore, we show that in the case of SNOW 2.0 with 256-bit key, the presented properties allow related-key attacks with complexity smaller than the exhaustive search. Finally, by using a property of the related keys by which given a (K, IV) value, the related key K' depends on the value of IV, we present a simple time-memory trade-off for the case where the attackers position is weakened with respect to the assumptions on the two related keys. The rest of the paper is organized as follows. In Section II, we briefly review the specifications of SNOW 3G and SNOW 2.0. The sets of related-keys are specified in Sections III and IV. The attacks against SNOW 2.0 with 256-bit key are examined in Section V and the conclusion is given in Section VI.

Snow Variant	Source	Related key pairs set size	# of slide steps	Key recovery attack
SNOW 3G	Th. 1	2^{32}	3	-
SNOW 2.0 (128-bit key)	Th. 2	2^{32}	2	-
SNOW 2.0 (128-bit key)	Th. 3	2^{64}	3	-
SNOW 2.0 (256-bit key)	Th. 4	2^{160}	2	$\surd$
SNOW 2.0 (256-bit key)	Th. 5	2^{192}	3	$\surd$
SNOW 2.0 (256-bit key)	Th. 6	2^{192}	4	$\surd$

TABLE I
SUMMARY OF RESULTS

II. Specifications of SNOW 3G and SNOW 2.0

Both SNOW 3G and SNOW 2.0 contain two main components: a Linear Feedback Shift Register (LFSR) and a Finite State Machine (FSM). The inner state of SNOW 3G (see Fig. 1) can be represented by $(s_0^t, \ldots s_{15}^t, R_1^t, R_2^t, R_3^t)$, where the s values represent 32-bit LFSR registers, the R values represent the 32-bit FSM registers and t denotes the number of iterations that have been executed so far. In SNOW 2.0, the FSM contains only two 32-bit registers and the inner state can be represented by $(s_0^t, \ldots s_{15}^t, R_1^t, R_2^t)$.

Unlike SNOW 3G which supports only 128-bit keys, SNOW 2.0 can be used with 128-bit and 256-bit keys. The size of the IV in both ciphers is 128 bits. In what follows, we briefly review the FSM and the LFSR update steps for the two ciphers.

SNOW 3G: The FSM update step is given by

$$\begin{aligned} &R_3^{t+1} = S_2(R_2^t), \;\; R_2^{t+1} = S_1(R_1^t) \\ &R_1^{t+1} = R_2^t \boxplus (R_3^t \oplus s_5^t) \end{aligned} \tag{1}$$

where S_1 and S_2 are two different 32×32 S-boxes, made of four parallel 8-bit S-boxes followed by a multiplication by a 4×4 matrix over GF(2^8) and $\boxplus$ denotes addition modulo 2^{32}.

The LFSR update is given by

$$s_{15}^{t+1} = \begin{cases} \alpha^{-1} \cdot s_{11}^t \oplus s_2^t \oplus \alpha \cdot s_0^t \oplus F^t, & t < 32 \\ \alpha^{-1} \cdot s_{11}^t \oplus s_2^t \oplus \alpha \cdot s_0^t & t \geq 32 \end{cases} \tag{2}$$

where α is a root of the $GF(2^8)[x]$ polynomial $x^4 + \beta^{23}x^3 + \beta^{245}x^2 + \beta^{48}x + \beta^{239}$, β is a root of the $GF(2)[x]$ polynomial $x^8 + x^7 + x^5 + x^3 + 1$, α^{-1} is the multiplicative inverse of α and F_t is the FSM output which is given by

$$F^t = (s_{15}^t \boxplus R_1^t) \oplus R_2^t.$$

Let $\mathbf{1}$ denote the all-one 32-bit word. The cipher operates as follows: the secret inner state is populated by $K = (K_0, \ldots K_4)$ and $IV = (IV_0, \ldots IV_4)$ according to

$$\begin{aligned} &s_{15}^0 = K_3 \oplus IV_0, \;\; s_{14}^0 = K_2, \;\; s_{13}^0 = K_1, \;\; s_{12}^0 = K_0 \oplus IV_1 \\ &s_{11}^0 = K_3 \oplus \mathbf{1}, \;\; s_{10}^0 = K_2 \oplus \mathbf{1} \oplus IV_2, \\ &s_9^0 = K_1 \oplus \mathbf{1} \oplus IV_3, \;\; s_8^0 = K_0 \oplus \mathbf{1} \\ &s_7^0 = K_3, \;\; s_6^0 = K_2, \;\; s_5^0 = K_1, \;\; s_4^0 = K_0 \\ &s_3^0 = K_3 \oplus \mathbf{1}, \;\; s_2^0 = K_2 \oplus \mathbf{1}, \;\; s_1^0 = K_1 \oplus \mathbf{1}, \;\; s_0^0 = K_0 \oplus \mathbf{1} \end{aligned} \tag{3}$$

and the FSM registers are reset to zero, i.e., $R_1^0 = R_2^0 = R_3^0 = 0$. The cipher is then iterated by executing (1) and (2) for 33 times without generating any output. Note that for $t < 32$, according to (2), the FSM output F_t participates in the LFSR update, contrary to step $t = 32$. Finally, the keystream words $(z^0, z^1, \ldots)$ are produced by

$$z^{t-33} = s_0^t \oplus F^t, t \geq 33. \tag{4}$$

In each such step, after generating the keystream word, the FSM and subsequently the LFSR are updated by (1) and (2).

SNOW 2.0: The FSM update function is defined by

$$R_1^{t+1} = s_5 \boxplus R_2^t, \quad R_2^{t+1} = S(R_1^t) \tag{5}$$

where S is a permutation of $\mathbb{Z}_{2^{32}}$ based on the round function of Rijndael [15]. The LFSR update function, and the FSM output F_t are defined in the same way as for SNOW 3G.

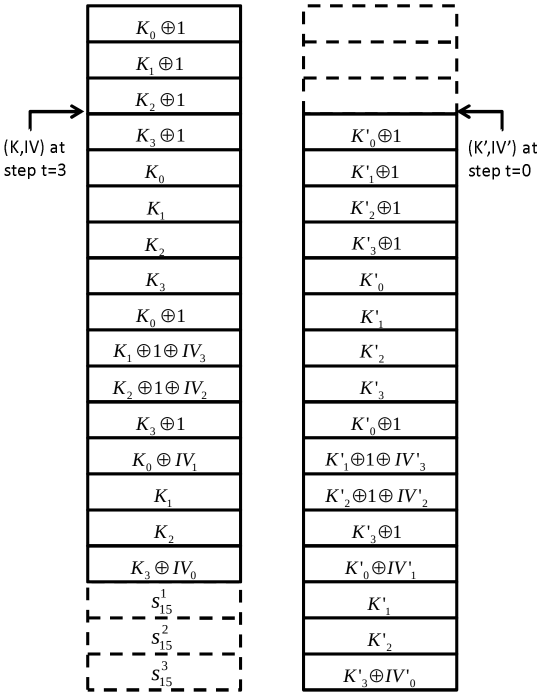

Fig. 2. (K, IV) and (K', IV') LFSR at times 3 and 0, respectively. For example, row 4 contains $K_3 \oplus \mathbf{1} = s_0^3$ and $K_0' \oplus \mathbf{1} = s_0^{'0}$

For the 128-bit version of SNOW 2.0 with $K = (K_3, K_2, K_1, K_0)$ and $IV = (IV_3, IV_2, IV_1, IV_0)$, the starting inner state is populated according to (3). For SNOW 2.0 with 256-bit key, $K = (K_7, \ldots K_0)$, the LFSR is populated by

$$\begin{array}{l} s_{15} = K_7 \oplus IV_0,\ s_{14} = K_6,\ s_{13} = K_5,\ s_{12} = K_4 \oplus IV_1 \\ s_{11} = K_3,\ s_{10} = K_2 \oplus IV_2,\ s_9 = K_1 \oplus IV_3,\ s_8 = K_0 \\ s_7 = K_7 \oplus \mathbf{1},\ s_6 = K_6 \oplus \mathbf{1}, \qquad \ldots, \qquad s_0 = K_0 \oplus \mathbf{1} \end{array} \tag{6}$$

The initialization process and the keystream generation are done the same way as in SNOW 3G.

The following notation will be used throughout the rest of the paper. For both SNOW 3G and SNOW 2.0, two instances of the cipher will be considered: one is initialized by (K, IV) and the other one is initialized by (K', IV'). Adding "′" as a suffix to the word will distinguish whether it relates to the (K, IV) or the (K', IV') instance of the cipher. For example, z_i', $s_j^{'t}$, $R_k^{'t}$ denote the keystream and the inner state of the (K', IV') instance of the cipher. Let IS_t denote the complete inner state of the (K, IV) instance of cipher at time $t \geq 0$. For example IS_0' represents the inner state of the cipher initialized by (K', IV'), after applying equations (3) and before executing any initialization steps.

The inner state at $t = 0$, i.e., the state right after applying (3) or (6) will be referred to as the *starting* inner state. The iteration in which the cipher goes from time t to time $t + 1$ is denoted by *step* t. Step t will be referred to as an *initialization step* if $0 \leq t \leq 31$. If $t \geq 32$, the step will be called a *keystream generation* step. The operators $\boxplus$ and $\boxminus$ denote addition and subtraction modulo 2^{32}, respectively.

III. Related-key pairs for SNOW 3G

In this section, we show that it is possible to initialize SNOW 3G by (K, IV) so that its inner state at time $t = 3$ represents a valid *starting* inner state corresponding to another (K', IV'). More precisely, we show that there exists a set of 2^{32} (K, IV) values such that for each such value, a unique (K', IV') exists so that $IS_3 = IS'_0$.

The initial equality $IS_3 = IS'_0$ is preserved until step 32, i.e., $IS_t = IS'_{t-3}$, $3 \leq t \leq 32$. At that point, a difference occurs due to the fact that in the (K, IV) instance an initialization step is applied to update the inner state whereas in the (K', IV') instance, a keystream generation step is applied according to (2). Nevertheless, due to the high degree of similarity among the corresponding inner states at the point where the keystream words are produced, several such words will be equal, contrary to how a perfect stream cipher should behave.

Define $C_1 = S_1^{-1}(S_2^{-1}(0))$, $C_2 = (S_1^{-1}(0) \boxminus S_1(0)) \oplus S_2(0)$ and $C_3 = (\boxminus S_1(S_1^{-1}(S_2^{-1}(0)))) \oplus S_2(S_1(0))$ and let a_0, b_0, b_1, b'_0 be 32-bit words. The following theorem specifies a set of 2^{32} related key pairs for SNOW 3G.

Theorem 1: Let $K = (a_0, C_1, C_2, C_3)$ and $K' = (C_3, a_0 \oplus \mathbf{1}, C_1 \oplus \mathbf{1}, C_2 \oplus \mathbf{1})$. Then, there exist unique $IV = (b_0, b_1, 0, 0)$ and $IV' = (b'_0, b_0, 0, b_1)$ such that $IS_t = IS'_{t-3}$, $3 \leq t \leq 32$ and

$$z_3 = z'_0, z_4 = z'_1, z_8 = z'_5, z_9 = z'_6. \tag{7}$$

Proof: First, we show that there exist unique IV and IV' of the form above so that K and K' satisfy $IS_3 = IS'_0$, i.e.,

$$(s_0^3, \ldots s_{15}^3, R_1^3, R_2^3, R_3^3) = (s_0^{\prime 0}, \ldots s_{15}^{\prime 0}, R_1^{\prime 0}, R_2^{\prime 0}, R_3^{\prime 0}) \tag{8}$$

Unfolding the FSM registers at $t = 3$ yields

$$\begin{aligned} R_1^3 &= s_7^0 \oplus S_2(S_1(0)) \boxplus S_1(s_5^0), \\ R_2^3 &= S_1(s_6^0 \oplus S_2(0) \boxplus S_1(0)) \ , \ R_3^3 = S_2(S_1(s_5^0)). \end{aligned}$$

Substituting the values s_5^0, s_6^0 and s_7^0 according to $s_5^0 = K_1 = C_1$, $s_6^0 = K_2 = C_2$ and $s_7^0 = K_3 = C_3$ (which follows by (2) and by the theorem formulation) shows that $R_1^3 = 0$, $R_2^3 = 0$ and $R_3^3 = 0$. Since $R_1^{\prime 0} = 0$, $R_2^{\prime 0} = 0$ and $R_3^{\prime 0} = 0$ by the SNOW 3G specification, the equality of the FSM words is established.

As for the LFSR values of equality (8), the problem is depicted in Fig. 2. It suffices to equate the expressions shown inside the rows using the keys specified by the theorem, skipping the first 3 rows. For example, row 4 corresponds to $s_0^3 = s_0^{\prime 0}$. This is trivially satisfied by the K and K' specified by the theorem by setting $K_3 \oplus \mathbf{1} = C_3 \oplus \mathbf{1} = K'_0 \oplus \mathbf{1}$, without imposing any constraint on IV, IV'. It is straightforward to verify that the same holds for rows 5, 6, 7, 8, 9, 12, 15. However, equating rows 10, 11, 13, 14 and 16 yields

$$IV_3 = IV_2 = 0, \ IV_1 = IV'_3, \ IV'_2 = 0, \ IV_0 = IV'_1 \tag{9}$$

Finally, equating rows 17, 18, 19 and substituting values for s_{15}^1, s_{15}^2, s_{15}^3 we have

$$\alpha(K_0 \oplus \mathbf{1}) \oplus K_2 \oplus \mathbf{1} \oplus \alpha^{-1}(K_3 \oplus \mathbf{1}) \oplus K_3 \oplus IV_0 = K_0 \oplus \mathbf{1} \tag{10}$$

$$\begin{aligned} &\alpha(K_1 \oplus \mathbf{1}) \oplus K_3 \oplus \mathbf{1} \oplus \alpha^{-1}(K_0 \oplus IV_1) \oplus \\ &((K_0 \oplus \mathbf{1}) \boxplus K_1) \oplus S_1(0) = K_1 \oplus \mathbf{1} \end{aligned} \tag{11}$$

$$\begin{aligned} &\alpha(K_2 \oplus \mathbf{1}) \oplus K_0 \oplus \alpha^{-1}(K_1) \oplus \\ &((K_1 \oplus \mathbf{1}) \boxplus (K_2 \oplus S_2(0) \boxplus S_1(0)) \oplus \\ &S_1(K_1) = K_2 \oplus \mathbf{1} \oplus IV'_0. \end{aligned} \tag{12}$$

It is clear that equations (10)-(12) can be solved explicitly in IV_0, IV_1 and IV'_0. In other words, by letting $K = (a_0, C_1, C_2, C_3)$ and $K' = (C_3, a_0 \oplus \mathbf{1}, C_1 \oplus \mathbf{1}, C_2 \oplus \mathbf{1})$ as specified by the theorem and fixing a_0, these three equations yield a unique IV_0, IV_1 and IV'_0, which take the place of b_0, b_1 and b'_0, respectively, showing that for K, K', there exist unique IV, IV' of the form specified by the theorem, satisfying (8).

To complete the proof it suffices to show that (8) implies (7). From (8), using (2), it follows that $IS_t = IS'_{t-3}$ for $3 < t \leq 32$. Again, according to (2), it follows that the difference in times $t = 33, 34, 35$ is present in registers $\{s_{15}\}$, $\{s_{15}, s_{14}\}$, $\{s_{15}, s_{14}, s_{13}\}$, respectively. As for times $t = 36, 37$, the difference in the inner states stays only in $\{s_{14}, s_{13}, s_{12}\}$, $\{s_{13}, s_{12}, s_{11}\}$, respectively. Then, using (4), it follows that $z_3 = z'_0$, $z_4 = z'_1$. By following the difference propagation, it is straightforward to see that at $t = 41, 42$ the active registers are $\{s_{14}, s_{13}, s_{12}, s_9, s_8, s_7\}$ and $\{s_{13}, s_{12}, s_{11}, s_8, s_7, s_6\}$, respectively, which, using (4), completes the proof of (7). ■

In the previous Theorem, related keys due to the slide of 3 steps are described. An attempt to change the number of sliding steps is unlikely to yield new interesting sets of related keys for SNOW 3G. Namely, slide pairs on the distance of 2 steps do not exist due to the fact that $R_3^2 = S_2(S_1(0))$ is a constant different than zero, which means that the inner state after 2 initialization steps cannot represent a starting inner state of another slided instance of the cipher. As for the slide by 4 steps, the FSM constraint restricts the key K to 2^{32} possible values. Then, the K candidates are restricted by an additional 64-bit filter due to the LFSR constraint, i.e., by equations $s_{13}^4 = s_{13}^{'0}$ and $s_{14}^4 = s_{14}^{'0}$. The two constraints together render the related-keys highly unlikely to exist. Finally, an eventual slide by more than 4 steps does not produce related keys since the difference between the initialization and the keystream generation steps for longer than 4 steps destroys the equivalence between the inner states which is needed to have some equal words in the corresponding output sequences.

IV. Related-key pairs for SNOW 2.0

In this section, we show that the strategy from Section III is also applicable against SNOW 2.0. In particular, for SNOW 2.0 with 128-bit keys, we show that two different related key sets exist due to the slide by 2 and by 3 steps. As for the 256-bit key version of SNOW 2.0, each of the slides by 2, 3 and 4 steps yield related key sets.

A. SNOW 2.0 with 128-bit keys

The following theorem reveals a set of 2^{32} related key pairs for the 128-bit version of SNOW 2.0, due to the slide by 2 steps. Let $C_1 = S^{-1}(0)$ and $C_2 = \boxminus S(0)$ and let a_0, a_3, b_1 and b'_0 be 32-bit words. Note that, according to the SNOW 2.0 specification, K and IV are are indexed in reverse order.

Theorem 2: Let a_0 and a_3 satisfy

$$\alpha(a_0 \oplus \mathbf{1}) \oplus C_2 \oplus \alpha^{-1}(a_3 \oplus \mathbf{1}) \oplus a_3 = a_0 \tag{13}$$

Let $K = (a_3, C_2, C_1, a_0)$ and $K' = (C_1 \oplus \mathbf{1}, a_0 \oplus \mathbf{1}, a_3, C_2)$. Then, for any $IV = (0, 0, b_1, 0)$, there exists a unique $IV' = (0, b_1, 0, b'_0)$ so that for SNOW 2.0 with 128-bit key, we have $IS_t = IS'_{t-2}$ for $2 \leq t \leq 32$ and

$$\begin{gathered} z_2 = z'_0, z_3 = z'_1, z_4 = z'_2 \\ z_7 = z'_5, z_8 = z'_6, z_9 = z'_7 \end{gathered} \tag{14}$$

Proof: Similar to the proof of Theorem 1, it will be shown that the K, K', IV and IV' values obeying the conditions of the theorem imply $IS_2 = IS'_0$. Since $R_1^2 = s_6 \boxplus S(0) = K_2 \boxplus S(0) = \boxminus S(0) \boxplus S(0) = 0 = R_1^{'0}$ and $R_2^2 = S(s_5) = S(K_1) = S(S^{-1}(0)) = 0 = R_2^{'0}$, the equality of the FSM registers is established. The LFSR constraint amounts to showing that

$$s_i^2 = s_i^{'0}, \ \ 0 \leq i \leq 15 \tag{15}$$

By substituting $s_i^2 = s_{i+2}^0$ for $i \leq 13$ and substituting s_{i+2}^0 and $s_i^{'0}$ according to (3), it easy to verify that for $i \in \{0,1,2,3,4,5,6,11\}$, (15) is satisfied without imposing any constraints on IV and IV'. On the other hand, using the same substitutions, due to (15) for $i \in \{7,8,9,10,12,13\}$, it follows that

$$\begin{gathered} IV_3 = 0,\ IV_2 = 0,\ IV_3' = 0, \\ IV_1 = IV_2',\ IV_1' = 0,\ IV_0 = 0 \end{gathered} \tag{16}$$

which leaves both $IV_1 = IV_2'$ and IV_0' unspecified. As for (15) for $i = 14$, it is satisfied due to (13). From (15), for $i = 15$, it follows that for any $IV_1 = b_1$, $IV_0' = b_0'$ is uniquely determined.

From $IS_2 = IS_0'$, by (2), it follows that $IS_t = IS_{t-2}'$ for $2 < t \leq 32$. In times $t = 33, 34$, the difference is present in $\{s_{15}\}$, $\{s_{15}, s_{14}\}$ registers, respectively. In times $t = 35, 36, 37$ the difference in the inner states is present only in $\{s_{14}, s_{13}\}$, $\{s_{13}, s_{12}\}$, $\{s_{12}, s_{11}\}$, respectively. Following the propagation further reveals that in times $t = 40, 41, 42$, the difference is only in $\{s_{14}, s_{13}, s_9, s_8\}$, $\{s_{13}, s_{12}, s_8, s_7\}$ and $\{s_{12}, s_{11}, s_7, s_6\}$, respectively. Taking into account (4), (14) follows. ■

The number of K values for which related K' exist is equal to the number of possible a_0, a_3 that satisfy the linear equation (13), i.e. 2^{32} values.

The next theorem reveals a larger set of 2^{64} related key pairs for 128-bit keyed SNOW 2.0, due to the slide by 3 steps. Let a_0, a_1 be arbitrary 32-bit words and let $A_3 = \boxminus S(a_1)$. Define the constant $C_1 = S^{-1}(0) \boxminus S(0)$.

Theorem 3: Let $K = (A_3, C_1, a_1, a_0)$ and $K' = (C_1 \oplus \mathbf{1}, a_1 \oplus \mathbf{1}, a_0 \oplus \mathbf{1}, A_3)$. Then, there exist unique $IV = (0, 0, b_1, b_0)$ and $IV' = (b_1, 0, b_0, b_0')$, for SNOW 2.0 with 128-bit key, such that $IS_t = IS_{t-3}'$ for $3 \leq t \leq 32$ and that

$$z_3 = z_0', z_4 = z_1', z_8 = z_5', z_9 = z_6'$$

As for the sliding by 4 steps, the FSM constraint imposes a 64-bit constraint on the key K and the equations $s_{13}^4 = s_{13}^{'0}$ and $s_{14}^4 = s_{14}^{'0}$ provide another 64-bit constraint. Since the expected number of such related key pairs is 1, they are less relevant and their treatment is omitted. As for sliding by more than 4 steps, the difference between the initialization and the keystream generation steps for longer than 4 steps destroys the equivalence between the inner states which consequently prevents having equal words in the corresponding output sequences.

B. SNOW 2.0 with 256-bit keys

The theorem that follows uses sliding by 2 steps to describe sets of 2^{160} related key pairs of SNOW 2.0 with 256-bit keys. Define the constants $C_1 = S^{-1}(0) \oplus \mathbf{1}$, $C_2 = (\boxminus S(0)) \oplus \mathbf{1}$ and let a_0, a_1, a_2, a_3, a_4, a_7, b_1, b_3 and b_0' be 32-bit words.

Theorem 4: Assume that

$$\alpha(a_0 \oplus \mathbf{1}) \oplus a_2 \oplus \alpha^{-1}(a_3) \oplus a_7 = a_0 \tag{17}$$

Let $K = (a_7, C_2, C_1, a_4, a_3, a_2, a_1, a_0)$ and $K' = (a_1 \oplus b_3 \oplus \mathbf{1}, a_0 \oplus \mathbf{1}, a_7, C_2, C_1, a_4, a_3, a_2)$. If $IV = (b_3, 0, b_1, 0)$, there exists a unique $IV' = (0, b_1, 0, b_0')$ such that for SNOW 2.0 with a 256-bit key, we have $IS_t = IS_{t-2}'$ for $2 \leq t \leq 32$ and

$$\begin{gathered} z_2 = z_0', z_3 = z_1', z_4 = z_2' \\ z_7 = z_5', z_8 = z_6', z_9 = z_7' \end{gathered} \tag{18}$$

Due to (17), the number of K values for which related K' exist is 2^{160}.

Next, sets of 2^{192} related-key pairs are derived by using a slide by 3 steps. Let a_0, a_1, a_2, a_3, a_4, a_5, b_2 and b_3 be arbitrary 32-bit words. Let $A_7 = (\boxminus S(a_5 \oplus \mathbf{1})) \oplus \mathbf{1}$ and define the constant $C_1 = (S^{-1}(0) \boxminus S(0)) \oplus \mathbf{1}$.

Theorem 5: Let $K = (A_7, C_1, a_5, a_4, a_3, a_2, a_1, a_0)$ and $K' = (a_2 \oplus b_2 \oplus \mathbf{1}, a_1 \oplus b_3 \oplus \mathbf{1}, a_0 \oplus \mathbf{1}, A_7, C_1, a_5, a_4, a_3)$. Then, there exist unique b_1, b_0 and b_0' such that with $IV = (b_3, b_2, b_1, b_0)$ and $IV' = (b_1, 0, b_0, b_0')$, for SNOW 2.0 with a 256-key, we have $IS_t = IS_{t-3}'$ for $3 \leq t \leq 32$ and

$$z_3 = z_0', z_4 = z_1', z_8 = z_5', z_9 = z_6' \tag{19}$$

Finally, sets of 2^{192} related-key pairs are described by using a slide of 4 steps for SNOW 2.0 with 256-bit keys. Let a_1, a_2, a_3, a_4, a_5 and a_6 be arbitrary 32-bit values. Define $A_7 = (S^{-1}(0) \boxminus S(a_5 \oplus \mathbf{1})) \oplus \mathbf{1}$ and $A_0 = \boxminus S((a_6 \oplus \mathbf{1}) \boxplus S(0))$.

Theorem 6: Let $K = (A_7, a_6, a_5, a_4, a_3, a_2, a_1, A_0)$. Then, there exist unique b_3, b_2, b'_1 and b'_0 such that for $K' = (a_3 \oplus \mathbf{1}, a_2 \oplus b_2 \oplus \mathbf{1}, a_1 \oplus b_3 \oplus \mathbf{1}, A_0 \oplus \mathbf{1}, A_7, a_6, a_5, a_4)$, $IV = (b_3, b_2, 0, 0)$ and $IV' = (0, 0, b'_1, b'_0)$ for SNOW 2.0 with 256-bit key, we have $IS_t = IS'_{t-4}$ for $4 \leq t \leq 32$ and

$$z_4 = z'_0, z_9 = z'_5$$

As in the case of SNOW 2.0 with 128-bit keys, attempts to slide by more than 4 steps do not yield related key pair sets since the equivalence between the inner states is destroyed due to the difference in the initialization and keystream generation steps.

V. Related-key attacks

Slide properties of stream ciphers may allow key-recovery attacks, as demonstrated in [12], [13], [14]. In [12], a slide property of Grain was exploited, by which for a fraction of 2^{-2n} key-IV values, there exists a related key-IV which produces identical but n-bit shifted keystream. Due to the simplicity of the relation between slided key-IV pairs and also due to the large number of such pairs, an attack in a single-key model was possible. As a result, a reduction of the Grain key space by a factor of 2 was achieved. In [13], a related-key attack against Salsa20 using a slide property of the cipher was given, where it was shown that the inner state can be recovered given keystream words of two instances of the cipher initialized by certain type of pairs of key-nonce-counter values. Finally, in [14], the LEX stream cipher was shown to be susceptible to a slide key-recovery attack requiring around 20000 keystream bytes produced under around $2^{60.8}$ random IVs.

In this section, we provide key-recovery attacks against SNOW 2.0 with 256-bit keys using the slide properties specified above. Firstly, we state a generic attack strategy that is straightforward and naturally follows from these properties. Then, we exploit the fact that in some of the related key-IV pairs specified by the Theorems above, the key K' depends on the IV value corresponding to its related key K. In particular, we show that the latter property gives the attacker some more freedom, without a comparable increase in the attack complexity. On the other hand, the sets of related keys for SNOW 3G and SNOW 2.0 with 128-bit keys appear to be too small to yield meaningful key-recovery attacks.

As for generic attacks against SNOW 2.0 with 256-bit key due to slide properties above, consider Theorems 4, 5 and 6 above. Given the two instances of the cipher initialized by unknown K and K' as specified by the corresponding theorem, the attacker queries the two instances until the IV and IV' that give slide pairs are found. The fact that the slide has been detected ensures that the starting inner states of the two instances are slided for the given IVs. Writing down the equation that equates corresponding registers of the two starting LFSRs and plugging the found IV and IV' in the equations gives a simple relation in the key bits and consequently restricts the key space.

As for the variation of the generic attack above, observe that in Theorems 4, 5 and 6, K' depends on the IV, the initialization vector of key K. It follows that, by varying IV in (K, IV), the key K is related to different related keys K', which in turn indicates that, given a cipher instance initialized by K, it is not necessary for the attacker to have access to a single K' cipher instance, but rather to a set of possible K' values. Consider for instance the relation between the two keys specified by Theorem 6:

$$\begin{aligned} K &= (A_7, a_6, a_5, a_4, a_3, a_2, a_1, A_0), \\ K' &= (a_3 \oplus \mathbf{1}, a_2 \oplus b_2 \oplus \mathbf{1}, a_1 \oplus b_3 \oplus \mathbf{1}, A_0 \oplus \mathbf{1}, A_7, a_6, a_5, a_4) \end{aligned}$$

The difference $K[0] \oplus K'[4]$ in the two keys is restricted to $\mathbf{1}$. Such a scenario between the two key portions is common for the usual related key model. On the other hand, if $K'[j]$ depends on IV value of its related key, as in the case of $K[1]$ and $K'[5]$ in the two keys above, more attack scenarios are possible. The attack may work for any difference between the two key words, where the attacker may not be even required to know the difference between these two words. As shown below, due to the possibility of applying time-memory tradeoffs, such extended attack scenarios do not necessarily lead to a proportional increase in the attack complexities.

In general, given the portions of the key and its related key, in this case subwords $K[i]$ and $K'[j]$, we distinguish the following scenarios:

(a) $K[i] \oplus K'[j]$ is an arbitrary value known to the attacker
(b) $K[i] \oplus K'[j]$ is an arbitrary value unknown to the attacker

Clearly, scenario (b) is less favorable for the attacker than the scenario (a). In what follow, we examine possible attacks when scenarios (a) and (b) are assumed for the key subwords in question. It should be noted that the number of unknown key bits in the two related keys can be taken to be the smaller of the numbers of unknown bits in the two keys. Since, in what follows, every two related keys have the same number of unknown bits, the number of bits in the key is equal to the number of unknown bits in one (any) of the two keys.

Let the attacker have access to two instances of the cipher initialized by unknown keys, but related as specified by Theorem 4. Along with the IVs, the initialization that results in a slide is specified by

$$\begin{aligned} K &= (a_7, C_2, C_1, a_4, a_3, a_2, a_1, a_0),\ \ IV = (b_3, 0, b_1, 0) \\ K' &= (a_1 \oplus b_3 \oplus \mathbf{1}, a_0 \oplus \mathbf{1}, a_7, C_2, C_1, a_4, a_3, a_2),\ \ IV' = (0, b_1, 0, b_0') \end{aligned}$$

where b_0' is unique once b_3 and b_1 are fixed. Let $K[1]$ and $K'[7]$ be related by scenario (a). In other words, any $K[1] \oplus K'[7]$ value is valid for the attack to succeed. Since due to the assumed scenario (a) the difference in question is known, so is the value b_3, i.e., the IV subword for which the slide can happen. Now, to find the IV and IV' such that (K, IV) and (K', IV') yield a slide pair, the attacker lets $b_1 = 0$, queries the K instance with $IV = (b_3, 0, 0, 0)$ once and the K' instance around 2^{32} times by varying b_0' in $IV' = (0, 0, 0, b_0')$, i.e., until (18) is satisfied. Then, due to (15) for $i = 15$, after simplifying the equation and substituting $s_{15}^1 = a_0 \oplus \mathbf{1}$, we have

$$\alpha(a_1) \oplus a_1 \oplus a_3 \oplus \alpha^{-1}(a_4) \oplus ((a_0 \oplus \mathbf{1}) \boxplus C_1) \oplus S(0) \oplus \alpha(\mathbf{1}) = \alpha^{-1}(b_1) \oplus b_3 \oplus b_0' \tag{20}$$

Since $b_1 = 0$, b_3 and b_0' are known, the equation above introduces a 32-bit constraint on key bits, reducing the unknown key bits number from 160 to 128. Consider now how the attack changes when instead of (a), scenario (b) is assumed between $K[1]$ and $K'[7]$. Now the attacker has access to two instances of the cipher instantiated by keys in Theorem 4, but the relation between $K[1]$ and $K'[7]$ is unknown and arbitrary. Then, the following process can be applied:

- For each b_3, query the K instance of the cipher using $IV = (b_3, 0, 0, 0)$. Save each $(z_2, z_3, z_4, z_7, z_8, z_9)$ as a row of table T.
- Sort table T
- For each b_0', query the K' instance of the cipher using $IV' = (0, 0, 0, b_0')$ and search for $(z_0', z_1', z_2', z_5', z_6', z_7')$ value in table T. If found, return the corresponding (b_3, b_0')

The advantage of the latter attack is that it does not assume any relation between $K[1]$ and $K'[7]$. It requires 2^{32} chosen-IV queries to each of the two oracles, storage of size 2^{32} and the computational effort dominated by a key search over the space of 2^{128} keys.

As for the attack based on Theorem 5, the key-IV pair that results in a slide pair is

$$\begin{aligned} K &= (A_7, C_1, a_5, a_4, a_3, a_2, a_1, a_0),\ \ IV = (b_3, b_2, b_1, b_0) \\ K' &= (a_2 \oplus b_2 \oplus \mathbf{1}, a_1 \oplus b_3 \oplus \mathbf{1}, a_0 \oplus \mathbf{1}, A_7, C_1, a_5, a_4, a_3),\ \ IV' = (b_1, 0, b_0, b_0') \end{aligned}$$

where b_1, b_0 and b_0' are uniquely determined once b_3 and b_2 are fixed, i.e. once K and K' are fixed. Assume a relation of type (a) between $K[1]$ and $K'[6]$ and also between $K[2]$ and $K'[7]$. It follows that the values b_3 and b_2 that can yield slid instances are known. Given that the IV values are of form $IV = (b_3, b_2, b_1, b_0)$ and $IV' = (b_1, 0, b_0, b_0')$, it suffices to try all possible guesses for b_1, b_0 and b_0' and find the IV, IV' pair that corresponds to the slided inner states. Again, relation (19) is used as a criterion to determine whether the slide happened or not. The cost of such a procedure is 2^{64} queries to the K instance of the cipher and 2^{96} queries to the K' oracle. Out of 2^{96} (b_1, b_0, b'_0) values, only the triplet that produces slide inner states is expected to pass, since (19) represents a 128-bit constraint. Once the b_1, b_0 and b'_0 have been found, equations $s^3_{13} = s'^0_{13}$, $s^3_{14} = s'^0_{14}$ and $s^3_{15} = s'^0_{15}$ that hold for slide pairs can be expanded. After simplifying the equations and substituting $s^1_{15} = a_0 \oplus \mathbf{1}$ and $s^2_{15} = a_1 \oplus b_3 \oplus \mathbf{1}$, we have

$$\alpha(a_0) \oplus a_2 \oplus \alpha^{-1}(a_3) \oplus (\boxminus S(a_5 \oplus \mathbf{1})) \oplus a_0 \oplus \alpha(\mathbf{1}) \oplus \mathbf{1} = b_0 \tag{21}$$

$$\alpha(a_1) \oplus a_3 \oplus \alpha^{-1}(a_4) \oplus ((a_0 \oplus \mathbf{1}) \boxplus (a_5 \oplus \mathbf{1})) \oplus a_1 \oplus \alpha(\mathbf{1}) \oplus S(0) = \alpha^{-1}(b_1) \oplus b_3 \tag{22}$$

$$\begin{aligned} &\alpha(a_2) \oplus a_4 \oplus \alpha^{-1}(a_5) \oplus ((a_1 \oplus b_3 \oplus \mathbf{1}) \boxplus (C_1 \oplus \mathbf{1}) \boxplus S(0)) \oplus \\ &\qquad\qquad \oplus S(a_5 \oplus \mathbf{1}) \oplus a_2 \oplus \alpha(\mathbf{1}) = b'_0 \oplus b_2 \end{aligned} \tag{23}$$

By guessing a_0, a_1 and a_5 the system is linearized in $GF(2^{32})$ and can be rewritten as

$$a_2 \oplus \alpha^{-1}(a_3) = L_1, \;\; a_3 \oplus \alpha^{-1}(a_4) = L_2, \;\; (\alpha \oplus 1)(a_2) \oplus a_4 = L_3$$

where L_1, L_2 and L_3 are known constants. These three equations above are independent and easy to solve in a_2, a_3, a_4. Consequently, the number of unknown key bits is reduced from 192 to 96. To summarize, to attack 192 bits of the secret key in the related key scenario, we require 2^{64} chosen-IV queries to the first instance and 2^{96} chosen-IV queries to the second instance of the cipher and finally a brute force over 2^{96} values to find the two secret keys. Note that given the key of form K, it is sufficient for the attacker to have access to any of the 2^{64} possible keys related to K', as long as the difference between $K[1]$ and $K'[6]$ and also between $K[2]$ and $K'[7]$ is known, i.e. scenario (a) is assumed for both pairs for key subwords. If instead of (a), scenario (b) is assumed for one of the two key subword pairs in question, say for $K[1]$ and $K'[6]$, the attack proceeds as follows:

- For each b_1, b_0
    - Create a table T with rows containing (z_3, z_4, z_8, z_9) generated by (K, IV) where b_3 is varying in $IV = (b_3, b_2, b_1, b_0)$ and b_2 is known and fixed
    - Sort table T
    - For each b'_0 search (z'_0, z'_1, z'_5, z'_6) generated using K' and $IV' = (b_1, 0, b_0, b'_0)$ in T. If found, return values for b_3, b_1, b_0 and b'_1
    - Otherwise: delete table T

On average one incorrect candidate for b_3, b_1, b_0 and b'_1 will be returned by the procedure above since (19) is a 128-bit constraint. The procedure requires sorting 2^{64} tables, each table containing 2^{32} rows, storage size of 2^{32}, 2^{96} chosen-IV queries to both instances of the cipher and finally, an exhaustive search over 2^{96} possible key values. If both of the key subwords pairs in question are assumed to follow relation (b), around 2^{32} false candidates for b_3, b_2, b_1, b_0 and b'_1 out of possible $2^{32\times 5}$ values are expected to pass the 128-bit constraint (19), which augments the computational effort of exhaustive search to $2^{96} \times 2^{32}$. Since for each b_3, b_2 the value for (z_3, z_4, z_8, z_9) is stored, there is an additional cost of sorting 2^{64} tables, each table containing 2^{64} rows and a storage requirement of 2^{64}. The number of the chosen IV queries is 2^{128} and 2^{96} to the K and K' instances of the cipher, respectively.

Compared to Theorems 4 and 5, Theorem 6 is less favorable for attacks. Consider the key-IV pair configuration specified by the theorem:

$$\begin{aligned} K &= (A_7, a_6, a_5, a_4, a_3, a_2, a_1, A_0), \;\; IV = (b_3, b_2, 0, 0) \\ K' &= (a_3 \oplus \mathbf{1}, a_2 \oplus b_2 \oplus \mathbf{1}, a_1 \oplus b_3 \oplus \mathbf{1}, A_0 \oplus \mathbf{1}, A_7, a_6, a_5, a_4), \;\; IV' = (0, 0, b'_1, b'_0) \end{aligned}$$

where, once key K is fixed, b_3, b_2, b'_1 and b'_0 are uniquely determined. Observe that the b_3 and b_2 words participate in the expressions for the key subwords $K'[5]$ and $K'[6]$, respectively. Therefore, given an instance with a key K, there exists no simple transformation to obtain a valid K'. In other words, given an instance with an unknown key K, the attacker does not know which transformation has to be applied on K to obtain K'. Instead of assuming that, nevertheless, the attacker has access to two instances with related K and K', we present the attack in the following more relevant scenario. Let the attacker know the left-hand side values in equations $s^4_{13} = s'^0_{13}$ and $s^4_{14} = s'^0_{14}$ that determine the correct b_3 and b_2:

$$\alpha(a_1) \oplus a_3 \oplus \alpha^{-1}(a_4) \oplus (((\boxminus S((a_6 \oplus \mathbf{1}) \boxplus S(0))) \oplus \mathbf{1} \oplus b'_1) \boxplus (a_5 \oplus \mathbf{1})) \oplus S(0) \oplus a_1 \oplus \alpha(\mathbf{1}) = b_3$$
$$\alpha(a_2) \oplus a_4 \oplus \alpha^{-1}(a_5) \oplus ((a_1 \oplus \mathbf{1} \oplus b_3) \boxplus (a_6 \oplus \mathbf{1}) \boxplus S(0)) \oplus S(a_5 \oplus \mathbf{1}) \oplus a_2 \oplus \alpha(\mathbf{1}) = b_2$$

The assumption lowers the number of starting unknown key bits from 192 to 128. For a perfect stream cipher, recovering 128 unknown bits of the keys should not be possible in less than 2^{128} operations. By having the knowledge about the key, the attacker also has the values of correct b_2 and b_3. Now the b'_1 and b'_0 values that produce a slide pair are found by applying 2^{64} queries to the K' oracle and comparing with the corresponding output with the output of the K instance of the cipher, used with the $IV = (b_3, b_2, 0, 0)$. After the correct b'_1 and b'_0 have been found, the equations $s^4_{12} = s'^0_{12}$ and $s^4_{15} = s'^0_{15}$ can be used to restrict the key space:

$$\begin{gathered}\alpha(\boxminus S((a_6 \oplus \mathbf{1}) \boxplus S(0))) \oplus a_2 \oplus \alpha^{-1}(a_3) \oplus (S^{-1}(0) \boxminus S(a_5 \oplus \mathbf{1})) \oplus \\ \oplus(\boxminus S((a_6 \oplus \mathbf{1}) \boxplus S(0))) \oplus \mathbf{1} \oplus \alpha(\mathbf{1}) = b'_1 \\ \alpha(a_3) \oplus a_5 \oplus \alpha^{-1}(a_6) \oplus (a_2 \oplus b_2 \oplus \mathbf{1} \boxplus (S^{-1}(0) \boxminus S(a_5 \oplus \mathbf{1})) \boxplus S(a_5 \oplus \mathbf{1})) \oplus \\ \oplus S((a_6 \oplus \mathbf{1}) \boxplus S(0)) \oplus a_3 \oplus \alpha(\mathbf{1}) = b'_0\end{gathered}$$

The key space is reduced to $128 - 64 = 64$ bits. Since it is expected that one false b'_0 and b'_1 will pass the test, the exhaustive search over 2^{65} keys and 2^{64} queries to the second oracle suffice to attack 128 unknown key.

In the case of related key sets due to Theorem 1 and 2 for SNOW 3G and SNOW 2.0 with 128-bits, the attacks are irrelevant since the number of initial unknown key bits is only 2^{32}. The attack against keys specified by Theorem 3 is also less relevant since the exhaustive search over the initial unknown 64 bits is more effective than the attack, since it would require around 2^{96} chosen-IV queries.

Finally, it should be noted that equations that reduce the key space considered in this section contain operations tat are not linear in $GF(2^{32})$. For example, (20) contains operation $\boxplus$ and (21) contains an S-box S application. So, in the attack based on Theorem 5, another key K'' equal to K' on all subwords except on a_5 would allow another equation of the form (21), with a'_5 instead of a_5, which would in turn reveal $(\boxminus S(a_5 \oplus \mathbf{1})) \oplus (\boxminus S(a'_5 \oplus \mathbf{1}))$. However, in each case above, exploiting the non-linearity for obtaining more key bit information requires introducing more related keys. For example, changing b_3 in (23) requires new related key K', since K' depends on b_3. Moreover, introducing more related keys does not lower the number of required chosen-IV queries. Since in this section the focus has been on extending the flexibility of the related key attack, adding more related keys without improving the practicality of the related key attack scenarios has been omitted.

VI. Discussion and Conclusions

We presented related key pair sets for SNOW 3G and SNOW 2.0 cipher by using a sliding technique. For several of the presented related key sets, the transformation from the key K to its related key K' is simple and amounts to rotation and bit inversion.

Using the derived related key sets, related-key key recovery attacks against SNOW 2.0 with 256-bit in complexity smaller than the exhaustive search can be mounted. Moreover, the fact that the K' depends on the IV of its related key was used to mount attacks under different assumptions on the related keys. Furthermore, the existence of the related keys exhibits non-random behavior of the ciphers, which questions the validity of the security proofs of protocols (such as the ones used in the 3GPP networks [11]) that are based on the assumption that SNOW 3G and SNOW 2.0 behave like ideal random functions when regarded as functions of the key-IV. For a more detailed discussion on related-key and *known-key* distinguishers, attacks, their security models and notions, the reader is referred to [16], [17].

It should be noted that an attack against the initialization procedure of ZUC [18] was announced in the rump session of Asiacrypt 2010 by Wu *et al.* [19]. This attack has some similarities with our work. Namely, it has been shown that for different IV values, identical inner states can be achieved only after 1 initialization step which results in identical keystream. By varying the IV value, the attacker finds the two identical keystreams and forms the simple equation required for the equal inner states to happen which significantly reduces the entropy of the secret key.

References

[1] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, 3G security, V3.1.1: ‘Specification of the 3GPP Confidentiality and Integrity Algorithms: Document 2: KASUMI Specification’, 2001

[2] Biham, E., Dunkelman, O., and Keller, N.: ‘A Related-Key Rectangle Attack on the Full KASUMI’. Proc. ASIACRYPT, Chennai, India, 2005, LNCS-3788, Springer, pp. 443-461

[3] Dunkelman, O., Keller, N., and Shamir, A.: ‘A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony’. Proc. CRYPTO 2010, Santa Barbara, California, 2010, LNCS-6223, pp. 393-410

[4] ETSI/SAGE: ‘Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2&UIA2. Document 2: SNOW 3G Specification, version 1.1’ (September 2006) `http://www.3gpp.org/ftp`

[5] Ekdahl, P., and Johansson, T.: ‘A New Version of the Stream Cipher SNOW’. Proc. SAC, St. Johns, Canada, 2002, LNCS-2595, Springer-Verlag, pp. 47-61

[6] Berbain, C., Billet, O., Canteaut, A., Courtois, N., Debraize, B., Gilbert, H., Goubin, L., Gouget, A., Granboulan, L., Lauradoux, C., Minier, M., Pornin, T. and Siber H.: ‘Decim^{v2}, The eSTREAM Finalists’. (2008), LNCS-4986, Springer, pp. 140-151

[7] Watanabe, D., Furuya, S., Takaragi, K. and Preneel, B: ‘A New Keystream Generator MUGI’, Proc. FSE 2002, LNCS-2259, Springer-Verlag, pp. 179-194

[8] Boesgaard, M., Vesterager, M., Pedersen, T., Christiansen, J. and Scavenius, O.: ‘Rabbit: A High-Performance Stream Cipher’, Proc. FSE 2003, LNCS-2887, Springer, pp. 307-329

[9] ETSI/SAGE: ‘Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2&UIA2. Document 5: Design and Evaluation Report, version 1.1’ (September 2006), `http://www.3gpp.org/ftp`

[10] Biryukov, A., Priemuth-Schmid D. and Zhang B.: ‘Multiset Collision Attacks on Reduced-Round SNOW 3G and SNOW $3G^{\oplus}$’, ACNS 2010, LNCS, Vol. 6123, pp. 139-153, Springer-Verlag, 2010

[11] Iwata, T., and Kohno, T.: ‘New Security Proofs for the 3GPP Confidentiality and Integrity Algorithms’, Proc. FSE, New Delhi, India, 2004, LNCS-3017, Springer-Verlag, pp. 427-445

[12] De Cannière C., Özgül Küçük and Preneel B.: ‘Analysis of Grain’s Initialization Algorithm’, Proc. AFRICACRYPT, Casablanca, Marocco, 2008, LNCS-4047, Springer-Verlag, pp. 276-289

[13] Priemuth-Schmid, D., and Biryukov, A.: ‘Slid Pairs in Salsa20 and Trivium’, Proc. INDOCRYPT, Khargpur, India, 2008, Springer-Verlag, LNCS-5365, pp. 1-14

[14] Wu, H. and Preneel, B.: ‘Resynchronization Attacks on WG and LEX’, Proc. FSE, Graz, Austria, 2006, LNCS-4047, Springer, pp. 422-432

[15] Daemen, J., and Rijmen, V.: ‘The Design of Rijndael: AES - The Advanced Encryption Standard (Information Security and Cryptography’, (Springer, 2002, 1st edition)

[16] Knudsen, L., and Rijmen, V.: ‘Known-Key Distinguishers for Some Block Ciphers’, Proc. ASIACRYPT, Kuching, Sarawak, Malasia, 2007, LNCS-4833, pp. 315-324

[17] Biryukov, A., Khovratovich, D., and Nikolic, I.: ‘Distinguisher and related-key attack on the full AES-256’, Proc. CRYPTO, Santa Barbara, California, 2009, LNCS-5677, pp. 231-249

[18] ETSI/SAGE: ‘Document 2: Specification of the 3GPP Confidentiality and Integrity Algorthithms 128-EEA3 & 128-EUA3: ZUC specification’, Version 1.4, 2010. Available at: `http://gsmworld.com/our-work/programmes-and-initiatives/fraud-and-security/gsm_security_algorithms.htm`

[19] Wu, H., Nguyen, P., Wang, H., Ling, S.: Cryptanalysis of Stream Cipher ZUC in the 3GPP Confidentiality & Integrity Algorithms 128-EEA3 & 128-EIA3’, Asiacrypt 2010 Rump Session talk. Available at: `http://www.spms.ntu.edu.sg/Asiacrypt2010/Common/rumpsession.html`