IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2020, No. 4, pp. 88–103. DOI:10.46586/tosc.v2020.i4.88-103

Fake Near Collisions Attacks*

Patrick Derbez†, Pierre-Alain Fouque and Victor Mollimard

Univ Rennes, Centre National de la Recherche Scientifique (CNRS), Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), Rennes, France
{patrick.derbez,pierre-alain.fouque,victor.mollimard}@irisa.fr

Abstract. Fast Near collision attacks on the stream ciphers `Grain v1` and `A5/1` were presented at Eurocrypt 2018 and Asiacrypt 2019 respectively. They use the fact that the entire internal state can be split into two parts so that the second part can be recovered from the first one which can be found using the keystream prefix and some guesses of the key materials.
In this paper we reevaluate the complexity of these attacks and show that actually they are inferior to previously known results. Basically, we show that their complexity is actually much higher and we point out the main problems of these papers based on information theoretic ideas. We also check that some distributions do not have the predicted entropy loss claimed by the authors. Checking cryptographic attacks with galactic complexity is difficult in general. In particular, as these attacks involve many steps it is hard to identify precisely where the attacks are flawed. But for the attack against `A5/1`, it could have been avoided if the author had provided a full experiment of its attack since the overall claimed complexity was lower than 2^{32} in both time and memory.

Keywords: Fast near collision technique · Reproducibility · Stream cipher

1 Introduction

Checking results is in some sciences such as experimental physics as important as the result itself. In these research domains, results have to be validated by two separate and independent teams before being published. In some computer sciences areas where results can depend on the input dataset, it is also highly important to give access to these data and to the code. In data mining for example the reproducibility of results has been acknowledged as mandatory before publishing work in order to ease the checking and/or comparison of this work with further research works.

In symmetric cryptography, where usually the complexities of attacks and distinguishers can be out of reach with experiments, a well-known method consists in experimentally checking only some parts of the attack and/or by targeting a toy cipher. Indeed attacks can usually be split in two parts: the adversary has to guess some bits and then he evaluates some distinguishers. The evaluation of the distinguisher cannot be exhaustive since it would have been tested for all guess bits. If we checked that the distinguisher is working for random guess, we declare that the attack is validated. However, it is the authors accountability to check carefully the experiments and reviewers usually verified the fact that the authors seem to have correctly performed their results. Nevertheless, sometimes

*The ToSC editorial board has contacted Bin Zhang and he stands by his claims regarding the correctness of the results of [ZXM18] and [Zha19]. Nevertheless, the editorial board considers the new results of this paper (that identifies errors in [ZXM18] and [Zha19]) as correct.

†Patrick Derbez, Pierre-Alain Fouque and Victor Mollimard were supported by the French Agence Nationale de la Recherche through the CryptAudit project under Contract ANR-17-CE39-0003.

Received: 2020-09-01 Accepted: 2020-11-01 Published: 2020-12-10

it is not sufficient to ensure the correctness of some proposed attacks and it is up to the community to revisit and discuss previous works to offer new insights on their contributions. For instance in [Gra01] Granboulan showed that differential attacks on `SKIPJACK` proposed in [KRW99] were flawed because the probabilities of some differential characteristics were not correctly evaluated. In 2007, Wang, Keller and Dunkelman [WKD07] caught a similar error for an impossible differential used in several attacks on `SHACAL-1`. Such errors may also come from hypothesis which do not hold for all ciphers as exemplified by Murphy [Mur11] with boomerangs on both `DES` and `AES`.

Symmetric cryptography is not the only place where mistakes can be made. In public-key cryptography and provable cryptography, it is also possible to discover errors as the famous bug in the OAEP paper [BR94], which has been corrected in [Sho01, FOPS01]. The same kind of problems appeared in proofs in symmetric cryptography for the equivalence between the random oracle model and ideal cipher model [CPS08] corrected in [HKT11] and more recently in the security proof of the OCB-2 mode of operation [IIMP19]. Consequently, Barthe *et al.* have developed tools to verify these proofs as in [BGHB11, BGLB11] and even on the corrected proofs they have been able to spot some errors or imprecisions since these tools do not accept unclear arguments or logical flaws. As a consequence, they design the EasyCrypt tool to help the verification of cryptographic proofs to reason about code-based proofs as these tools were first developed to verify programs. There is no such tool to check symmetric-key cryptanalysis. The verification of these attacks boils down to checking the complexity analysis of the cryptanalytic algorithm. The main difficulty is that some parts are heuristic and the verification of these heuristics are not easy to automatize and to perform rigorously. Moreover, understanding the problems is not always an easy task since it requires to reverse engineer the experiments performed which are subject to statistical effects and it is less easy than reading a proof.

Contributions. In this paper, we look at the recent fast near collision attacks proposed by Zhang, Xu and Meier against the `Grain v1` [ZXM18] stream cipher and by Zhang against `A5/1` [Zha19]. The main idea behind fast near collision attack consists in a divide-and-conquer partition of the full internal state into the crucial part (CP) and the rest part (RP). The latter part can be efficiently recovered using only the CP, while the former one is retrieved using a near collision attack based on a small number of bits of the keystream.

Our first goal was to implement the attack on the `A5/1` stream cipher since the time and memory complexities seem within our reach and practical. However, during this process we discovered several issues in the claimed probabilities, leading to an overall complexity much worst than expected. In fact, we came up to implement a slower version of the attack proposed by Golić at Eurocrypt'97 [Gol97]. Consequently, we scrutinized this article and decided to reevaluate the time complexity to 2^{28} calls to (the end of) Golić's attack, for an overall complexity around 2^{42}. Since this attack is a bit difficult as it is flooded with the details of the stream cipher under attack, we decided to present its basic ideas in a self-contained manner. Finally, we decided to also verify the previous attack on `Grain v1` as proposed at Eurocrypt'18 and we discovered similar problems in the analysis. In particular, the correct overall complexity is 2^{113} and so the attack is less efficient than the naive exhaustive search in $2^{87.4}$ ticks on `Grain v1`.

More importantly, we show in Section 2 that fast near collision attacks, as described in both [ZXM18] and [Zha19], are intrinsically erroneous. Replacing the refined self-contained method, which is the core of those attacks and the only algorithm relying on near collisions, by an algorithm outputting a random set (of fixed size) of pre-images would lead to the exact same complexities. Thus such attacks are illusive.

2 Fast Near Collision

At Eurocrypt'18, Zhang *et al.* described a new powerful cryptanalysis technique called *fast near collision attack*. This technique was specially designed to analyze stream ciphers and was successfully applied to both `Grain v1` [HJMM06] and `A5/1` [BGW99]. It combines both a divide-and-conquer approach and near collisions. The core idea is to use near collisions to restrict the possible values of some bits of the internal state.

2.1 The refined self-contained method

Let f be a public function from n to m bits, $\mathbf{x}_s$ be a secret n-bit word and $\mathbf{k}_s$ the output of $f(\mathbf{x}_s)$. A classical objective is to retrieve $\mathbf{x}_s$ from the knowledge of both f and $\mathbf{k}_s$. In the following we will explain how the fast near collision technique claims to restrict the search space for $\mathbf{x}_s$.

The process is composed of 3 procedures which aim at computing a set X containing $\mathbf{x}_s$ with a high enough probability.

Precomputation. The first step in a fast near collision attack is to construct a differential table T_d mapping each pair $(\Delta k, k)$ to all possible Δx such that:

- $|\Delta x| \leq d$
- there exists x such that $f(x) = k$ and $f(x \oplus \Delta x) = k \oplus \Delta k$.

In other words, the table T_d is a variant of the classical *differential distribution table* associated to an Sbox. The number of times Δx is solution for $(\Delta k, k)$ is also stored as extra information. This allows for each value of k to select Δk to maximize the probability of $f(x \oplus \Delta x) = k \oplus \Delta k$ knowing both $f(x) = k$ and $\Delta x \in T_d\,[\Delta k, k]$.

Note that in case it would be too costly to fully compute T_d, x and Δx can be sampled.

Online. The second step of the procedure uses the precomputed table to generate a set X containing $\mathbf{x}_s$ with a good probability. The process is described in Algorithm 1. The idea is to randomly generate x, compute $k = f(x)$, look into $T_d\,[k \oplus \mathbf{k}_s, \mathbf{k}_s]$ for possible Δx's and check whether $f(x \oplus \Delta x) = \mathbf{k}_s$. If the last equality holds then $x \oplus \Delta x$ is added to the set X as a possible value for $\mathbf{x}_s$.

Algorithm 1 The refined self-contained method

```
1: Data: keystream k_s, difference Δk, table T_d,
2: Result: a set X such that x_s ∈ X has high probability
3: X ← ∅
4: for i = 0 to N do
5:     randomly generate x such that f(x) = k_s ⊕ Δk
6:     for all Δx ∈ T_d[Δk, k_s] do
7:         if f(x ⊕ Δx) = k_s then
8:             X ← X ∪ {x ⊕ Δx}
9:         end if
10:    end for
11: end for
12: return X
```

Amplifying phase. In order to increase the probability that X contains $\mathbf{x}_s$, Zhang *et al.* propose to run $N \times M$ times Algorithm 1 , each random invocation outputting a set

denoted $X_{i,j}$ ($i = 1$ to N and $j = 1$ to M). Then a new set is outputted by computing

$$X = \bigcup_{i=1}^{N} \left(\bigcap_{j=1}^{M} X_{i,j} \right).$$

2.2 About probabilities

While we could discuss on the interest of this construction, we are only interested by the probability that $\mathbf{x}_s$ belongs to the constructed set X.

`Grain v1`. In [ZXM18], Zhang et al. used the fast near collision technique to mount an attack against `Grain v1`. They applied the refined self-contained method to a function f such that $n = 12$ and $m = 2$. They obtained a set X of size 848 and claimed the probability for $\mathbf{x}_s$ to belong to X is around 89.64% which is a bit higher than the $848/1024 = 82.81\%$ expected. Note that here the function f is such that $z = f(x)$ can be rewritten as $z = x_1 \oplus h(x_2)$ and thus, the refined self-contained method was applied on $h(x_2) = 0$. In particular this means that the search space is restricted without the knowledge of any bit of keystream.

`A5/1`. In [Zha19], the function f is such that $n = 15$ and $m = 2$. Zhang obtained a set X of size 7835 and claimed the probability for $\mathbf{x}_s$ to belong to X is around 99.09% which is higher than the $7835/8192 = 95.64\%$ expected.

We claim all those claimed probabilities are wrong or, more precisely, cannot be true without a big enough bias in the initialization phases of both `A5/1` and `Grain v1`. This is supported by the following theorem:

Theorem 1. *Let $\mathcal{A}$ be an algorithm which takes as input a function f and an element $\mathbf{k}_s$ and outputs a subset X of $f^{-1}(\mathbf{k}_s)$. Let $\mathbf{x}_s$ be an element of $f^{-1}(\mathbf{k}_s)$ drawn uniformly at random. The probability that $\mathbf{x}_s$ belongs to X is exactly*

$$|X|/|f^{-1}(\mathbf{k}_s)|.$$

The refined self-contained method fulfils the requirements of Theorem 1 but Zhang *et al.* claim the set X output by the algorithm contains the secret $\mathbf{x}_s$ which generated $\mathbf{k}_s$ with a good probability. Note that the algorithm can be run *before* the secret was actually generated and thus Zhang *et al.* claim can be invalidated by the following experiment:

1. randomly generate $\mathbf{k}_s$
2. run the refined self-contained method on f and $\mathbf{k}_s$ and obtain the subset X
3. draw $\mathbf{x}_s$ uniformly at random in $f^{-1}(\mathbf{k}_s)$
4. check whether $\mathbf{x}_s$ belongs to X

Hence, the probabilities given in both [ZXM18] and [Zha19], and by extension the complexity of corresponding attacks, are quite suspicious. Actually, they would hold if and only if it is not possible to draw $\mathbf{x}_s$ uniformly at random in $f^{-1}(\mathbf{k}_s)$ which would imply bias in the initialization process.

2.3 Several issues

We found several issues and unreproducible results in both [ZXM18] and [Zha19]. The first and most important one is about the set outputted by Algorithm 1 and, more precisely, about its average size and the average probability for the *right value* to belong to this

set. For both `Grain v1` and `A5/1`, they were obtained experimentally from unspecified procedures and do not satisfy Theorem 1. Since Zhang *et al.* state to have conducted *extensive experiments*, either the whole experiments were flawed or the pseudo-random generators they used were biased.

Another issue lies in the amplifying phase. First the computations are all based on the wrong results regarding Algorithm 1 and so are unlikely to be correct. But there is another issue with this phase. Authors used two independent theorems to exhibit the claimed special behavior of the set X constructed in the amplifying phase: one to compute the size of X and one to compute the probability for the right value to belong to X. While using two different avenues to prove two properties on the same set is not important in regards to the truth of the statement, the theorem they used to compute the size of X (Statement 1 in this paper) is flawed. As a consequence, there is a decorrelation between the computation of the probability that X contains the correct value and the computation of the size X, explaining again the incorrect complexities they found for their attacks.

Statement 1 (Theorem 3 of [ZXM18])**.** *Let V be a set and let draw uniformly at random a collection (U_i) of subsets of V. Let $F_i = \bigcup_{k \leq i} U_k$. Then on average the following relation holds:*

$$|F_{i+1}| = |F_i| + |U_{i+1}| - \sum_{j=0}^{|U_{i+1}|} \frac{\binom{|F_i|}{j} \cdot \binom{|F_{i+1}|-|F_i|}{|U_{i+1}|-j}}{\binom{|F_{i+1}|}{|U_{i+1}|}} \cdot j$$

The sum in the formula is expected to compute the average size of the intersection between both the sets F_i and U_{i+1} and this is where the error lies. The main idea is correct as they count the number of configurations such that j elements of U_{i+1} belong to F_i and $|U_{i+1}| - j$ elements do not. But actually, at this point, *not in* F_i does not mean *in* $F_{i+1} - F_i$ but means *in* $V - F_i$. Indeed, U_{i+1} is drawn as a subset of V not as a subset of F_{i+1}. Hence the corrected version of the Statement 1 is proposed in Theorem 2.

Theorem 2 (Corrected version)**.** *Let V be a set and let draw uniformly at random a collection (U_i) of subsets of V. Let $F_i = \bigcup_{k \leq i} U_k$. Then on average the following relation holds:*

$$|F_{i+1}| = |F_i| + |U_{i+1}| - \sum_{j=0}^{|U_{i+1}|} \frac{\binom{|F_i|}{j} \cdot \binom{|V|-|F_i|}{|U_{i+1}|-j}}{\binom{|V|}{|U_{i+1}|}} \cdot j$$

In particular, the formula used by Zhang *et al.* would always underestimate the average size of set F_i. This fully supports our claiming: to reach the probabilities announced in both [ZXM18] and [Zha19] the size of the set output by the refined self-contained method has to be bigger than they expected.

Finally there is a wrong assumption about the *right value*. More precisely, in both papers authors assume there is only one right value that will behave differently than the wrong ones. With enough keystream bits this is true that there is only one internal state solution. But the fast near collision only uses a *small* part of the known keystream bits and so the assumption of only one right value does not hold. For instance, for the attack against `A5/1`, the fast near collision technique is applied to only 5 keystream bits and we show Section 3.3 there are many more right values than only one.

In the next sections, we will show for both `Grain v1` and `A5/1`, the observed deviation in the probabilities is wrong and will give the corrected complexities of the corresponding attacks.

3 Fast Near Collisions on A5/1

In this Section we carefully study the attack presented in [Zha19]. We first briefly recall the design of `A5/1` and Golić attack. Then we describe Zhang attack and explain why its complexity was underestimated.

3.1 Description of `A5/1`

`A5/1` is a stream cipher underlined by a 64-bit internal state. The internal state is composed of three short linear feedback shift registers (LFSR) of length 19, 22 and 23 bits respectively. In the rest of the paper we will refer to them as $R1$, $R2$, $R3$. As illustrated in Figure 1, the feedback taps for each LFSR are positions 13, 16, 17 and 18 for $R1$, 20 and 21 for $R2$ and 7, 20, 21 and 22 for $R3$. Furthermore, each LFSR also possesses a clocking tap at position 8, 10, 10 for respectively $R1$, $R2$ and $R3$, represented with the red arrows in the figure.

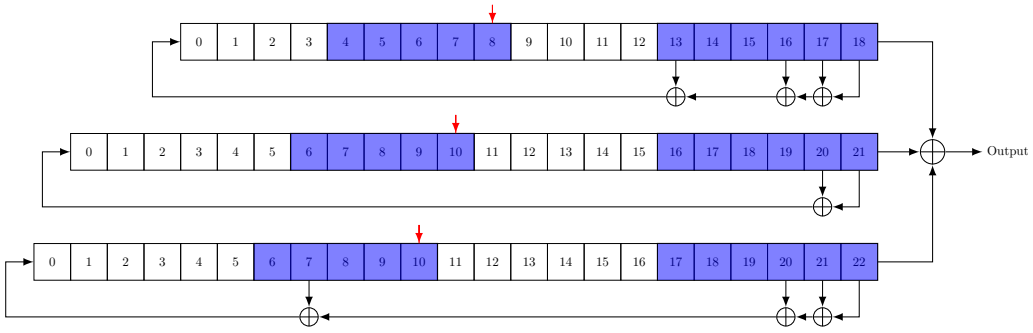

Figure 1: Description of `A5/1` (source: [Jea16]). The 33 blue bits are the one required to compute the first 5 keystream bits.

`A5/1` uses an asynchroneous clocking regime for the LFSRs: at each clock tick, a LFSR is clocked if its clock tap value is the majority value between the three clocking taps.

Finally, we review the utilisation of the `A5/1` stream cipher during a GSM conversation session with Algorithm 3, the pseudo code for the generation of the 228 bits of keystream of one GSM session.

3.2 An attack from Golić

In [Gol97], Golić introduces a clever memory-less attack against `A5/1`. It is a basic divide-and-conquer attack recovering the unknown initial state from a known keystream sequence.

The main idea is quite simple. If, for each of the three LFSRs, one guesses the clock bit for n (asynchronous) clocks of the LFSRs, we can obtain $3n$ linear/affine equations. For instance, for $n = 10$ it means guessing on the initial state $R1[0..8]$, $R2[1..10]$, $R3[1..10]$ as well as $R1[13] \oplus R1[16] \oplus R1[17] \oplus R1[18]$. Furthermore, from those $3n$ guesses we know the beginning of the clocking sequence and obtains on average $1 + 4n/3$ affine equations from the knowledge of the keystream bits. Indeed, at each step the probability for a register to be clocked is $3/4$ and as a consequence from the $3n$ guesses we know on average the clocking sequence for $4n/3$ rounds, leading to the equations $R1^i[18] \oplus R2^i[21] \oplus R3^i[22] = z_i$ for $0 \leq i \leq 4n/3$. Hence, a naive solution would be to accumulate enough equations to solve the system by inverting a matrix. This would require n to be such that $1 + 4n/3 + 3n \geq 64$, so $n \geq 14.6$. But actually, for $n \geq 10$, the equations are not linearly independent, and we need to increase the number of guesses to make.

To overcome this issue, Golić proposed a better algorithm close to the *early abort technique* [LKKD08]. At each step the adversary guesses/computes the majority bit, gets the corresponding equation from the corresponding keystream bit and checks whether it is consistent with the previously obtained equations. If the equation is consistent, the equation is added to the system, the missing clocking bits are guessed/computed from the majority bit and the already known clocking bits and the whole state is clocked. This process is repeated until the system uniquely determines the 64-bit state. Golić showed that the average complexity of the procedure is around $2^{41.16}$ simple operations.

3.3 Fast near collisions attack against `A5/1`

At Asiacrypt'19, Zhang proposed an improved memory-less attack against `A5/1`, claiming a time complexity around 2^{31} clocks [Zha19]. Given a sufficiently long sequence of keystream bits (around 64), he proposed a 2-step procedure to recover the full internal state.

1. The main observation is that 2 consecutive bits of keystream only depend on 15 variables of the internal states. Using the technique described in Section 2.1, Zhang constructs a set containing approximately 7835 values for the 15 variables and claims that the probability the value we want is in it to be around 99.09%. Four such sets are constructed, one for each pair (z_i, z_{i+1}) of keystream bits, for i from 0 to 3. Then a sophisticated merge procedure is applied to construct a set of $2^{16.6}$ values for the 33 bits of the internal state leading to $z_0z_1z_2z_3z_4$. Furthermore, Zhang claims that the probability for the set to contain the right value is round $(0.9909)^4 = 96.41\%$. Note that $2^{16.6}$ possibilities is much lower than $2^{33-5} = 2^{28}$, which is what we would intuitively expect.

2. The 31 remaining state bits are recovered using the procedure of Golić described Section 3.2 with few refinements.

3.4 Complexity correction

In this section, we show the time complexity of the attack presented by Zhang at Asiacrypt'19 is actually much higher than announced in [Zha19]. More precisely, we show it is impossible to restrict the number of possible values for the 33 bits of the crucial part (CP) from 2^{33} to $2^{16.6}$ using only the 5 first keystream bits without drastically decreasing the probability of success of the attack. Hence, it turns out Zhang's attack has the same complexity than the one of Golić.

Theoretical analysis. As explained in Section 3.3, the attack proposed by Zhang begins by the recovery of the crucial part (CP) corresponding to 33 bits of the internal state of `A5/1`. Those bits are coloured in blue on Figure 1. The only information used in the procedure to do so is the first five bits of keystream generated from the internal state.

Let x be a randomly chosen value for the CP part and k its corresponding 5-bit keystream output. In his attack, Zhang claims that from k he can extract a set of $2^{16.6}$ CP configurations containing x with a very high probability. To invalidate this result we first make the following proposition:

Proposition 1. *Given a 5-bit keystream output k, there are* **exactly** 2^{28} *values for the 33 bits of the CP part leading to k.*

Proof. The 33 bits of the CP part can be divided into two groups: one composed of 15 bits ($R1[4...8]$, $R2[6..10]$ and $R3[6...10]$) used only to determine the clocking sequence and one composed of 18 bits ($R1[13..18]$, $R2[16...21]$, $R3[17..22]$) used to generate the keystream bits. Hence, once the 15 bits of the first group are fixed, the clocking sequence is known

and so each of the five first keystream bits is computed as a linear combination of the 18 bits of the second groups. Furthermore, those 5 linear equations are independent since each of them depends on at least one bit that does not appear in the other ones (because at least two registers are clocked each round). Thus, for each possible value of k we have exactly $2^{18-5} = 2^{13}$ possible values for the 18 bits of the second groups. □

According to Proposition 1, the claim of Zhang would imply that over the 2^{33} possible values of the 33 bits of the CP part, only a subset of $2^{16.6+5} = 2^{21.6}$ values (a set of $2^{16.6}$ for each of the 2^5 possible keystream values) can be actually reached after A5/1 initialization, the remaining ones being reached with marginal probability. While it seems quite obvious that such a big bias would have already been observed, we ran several experiments to refute the claim made by Zhang.

Experimental results. We first experimentally verified Proposition 1. We count for each of the 2^5 5-bit keystream prefix the number of CP values that generate it. As expected, we found that for 5 given bits of keystream prefix, there are exactly 2^{28} CP combinations that generate it.

The second hypothesis we studied was a potential bias in reaching every CP configuration from the initialization phases of a GSM session. To test this hypothesis, we ran two experiments, sampling at random the 33-bit CP part after an `A5/1` initialization.

Algorithm 2 Experiment

```
sample a 128-bit word KEY from /dev/rand                          ▷ Setup
initialize a 128-bit word COUNTER at 0
initialize a 2^33 array called CONFIGURATION;

for i = 0 to 2^36 do                                              ▷ Experiment
    RANDOM = AES-CTR(KEY, COUNTER) and increase COUNTER
    extract from RANDOM one A5/1-key, KEYEXP and one A5/1-frame, FRAMEEXP
    do an A5/1 initialization with KEYEXP and FRAMEEXP
    select the 33-bit of the CP part of the obtained internal state
    increment the corresponding field in CONFIGURATION
end for

Output CONFIGURATION
```

For the first one, we simply drawn uniformly at random 2^{36} 64-bit keys and 22-bit frame counters. For each of them we performed the initialization process of `A5/1` as detailed in Algorithm 3 and computed the corresponding value of the 33 bits of interest. To avoid any bias in the experiment we used `AES` in CTR mode as source of randomness. For the sake of clarity and to give more details about the random sampling, we give a pseudo-code description of the experiments in Algorithm 2.

The second experiment is exactly the same as the first one but the 64-bit key is composed of 54 random bits and 10 zeroes for its rightmost bits as it was traditionally done in some system, like comp128v2.

We present in Figure 2 the distribution of occurrences of the 2^{33} possible CP values for respectively GSM session key of 54 random bits and 64 random bits and for randomly selected internal states of 64-bit in the form of histograms mapping a value n to the number of CP configurations (in log scale) that are sampled n times.

No bias as strong as the one presented in the complexity announced by Zhang can be observed on those representative histograms.

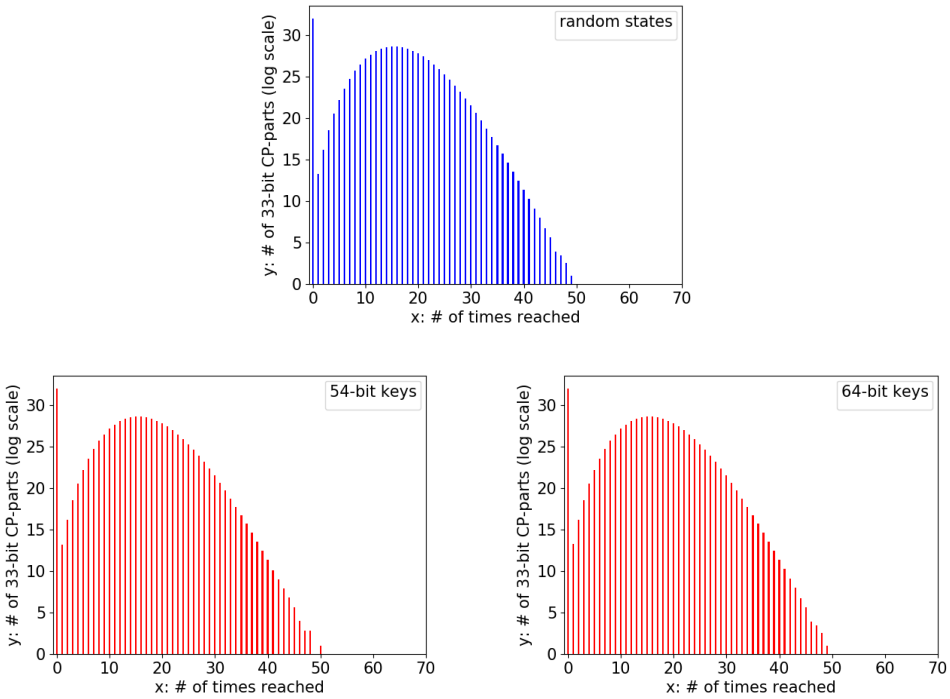

Figure 2: One diagram represents the number of 33-bit CP part values (in log scale) from experimental data in ordinates that are reached exactly x times on the horizontal axe. The experimental data for the two red diagrams are generated as described in Algorithm 2 with 54-bit keys for the left one and 64-bit keys for the right one. The experimental data of the blue diagram comes from directly randomly generating the 33-bit CP part of the A5/1 internal state.

Finally, we provide a last experiment definitely showing the attack presented in [Zha19] is flawed.

1. Randomly generate a 5-bit word $\mathbf{k}_s$
2. Run the refined self-contained method to obtain a set X of size $2^{16.6}$. According to [Zha19], this set should contain the secret which generated $\mathbf{k}_s$ with probability 0.9641.
3. Do N times:
   (a) Randomly generate a key and a frame counter and run the initialization process
   (b) Check whether the first five keystream bits match $\mathbf{k}_s$. If not repeat the previous step.
   (c) Check whether the value of the CP part belongs to X.
4. Check whether the experimental probability matches the expected one.

We ran this experiment[1] 10 times with $N = 2^{28}$ and found the experimental probability to be very close to $2^{-11.4}$, confirming the probability of 0.9641 claimed by Zhang to be far from the reality.

[1] Actually at each trial we took for X the set of the $2^{16.6}$ values reached the most. This highlights the refined self-contained method is irrelevant and can be replaced by any algorithm outputting a set of size $2^{16.6}$.

Corrected complexity. With the probability of success corrected, Zhang's attack becomes very similar to the one of Golić. The difference is that he would guess 18 extra bits while Golić would have 5 linear/affine equations between those 18 bits and the keystream. Hence in Zhang attack one would proceed less keystream bits before obtaining an invertible system of equations and thus more keystream bits should be checked *a posteriori*, leading to an attack which cannot be better than Golić one.

4 Fast Near Collisions on Grain v1

In this section, we study the attack proposed at Eurocrypt'18 [ZXM18] in the same way we did in the previous section for `A5/1`.

4.1 Description of `Grain v1`

`Grain` is a family of stream ciphers that was retained in the eSTREAM portfolio [est09]. In this paper, we focus on `Grain v1` as specified in [HJM07]. This stream cipher is composed of one LFSR of 80 bits chained with a non-linear feedback shift register (NFSR) of 80 bits.

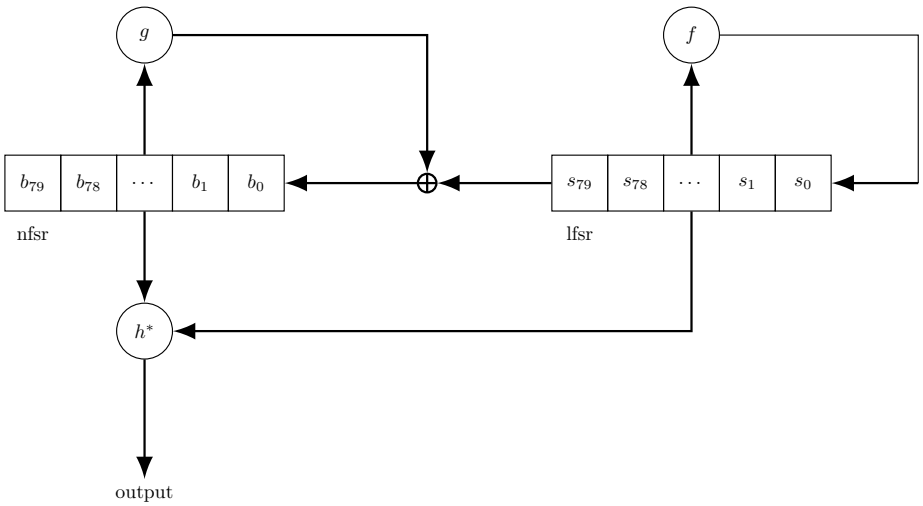

Figure 3: Simple representation of the grain v1 cipher

At step i, the content of the LFSR is denoted by $s_i, s_{i+1}, \ldots, s_{i+79}$ and the content of the NFSR is denoted by $b_i, b_{i+1}, \ldots, b_{i+79}$.

The update function of the LFSR is defined as

$$s_{i+80} = s_{i+62} \oplus s_{i+51} \oplus s_{i+38} \oplus s_{i+23} \oplus s_{i+13} \oplus s_i,$$

and the one of the NFSR as

$$\begin{aligned} b_{i+80} \;=\; & s_i \oplus b_{i+62} \oplus b_{i+60} \oplus b_{i+52} \oplus b_{i+45} \oplus b_{i+37} \oplus b_{i+33} \oplus b_{i+28} \oplus b_{i+21} \oplus b_{i+14} \\ & \oplus b_{i+9} \oplus b_i \oplus b_{i+63}b_{i+60} \oplus b_{i+37}b_{i+33} \oplus b_{i+15}b_{i+9} \oplus b_{i+60}b_{i+52}b_{i+45} \\ & \oplus b_{i+33}b_{i+28}b_{i+21} \oplus b_{i+63}b_{i+45}b_{i+28}b_{i+9} \oplus b_{i+60}b_{i+52}b_{i+37}b_{i+33} \\ & \oplus b_{i+63}b_{i+60}b_{i+21}b_{i+15} \oplus b_{i+63}b_{i+60}b_{i+52}b_{i+45}b_{i+37} \oplus b_{i+33}b_{i+28}b_{i+21}b_{i+15}b_{i+9} \\ & \oplus b_{i+52}b_{i+45}b_{i+37}b_{i+33}b_{i+28}b_{i+21}. \end{aligned}$$

At each step, the output bit is computed from 8 bits of the NFSR and 4 bits of the LFSR as

$$z_i = h(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63}) \oplus \bigoplus_{k \in \mathcal{A}} b_{i+k},$$

where h is a boolean function of degree 3 and $\mathcal{A} = \{1, 2, 4, 10, 31, 43, 56\}$.

The initialization of `Grain v1` is described in Algorithm 4. First, the 80-bit key is loaded into the NFSR and the 64-bit IV into the 64 first bits of the LFSR. Remaining bits of the LFSR are set to 1. Then, the internal state is clocked 160 times with a re-injection of the output bits.

4.2 Zhang *et al.* attack

At Eurocrypt'18, Zhang *et al.* presented a fast near collisions attack against `Grain v1`, claiming a time complexity around $2^{75.7}$ ticks. Let x_i be $\bigoplus_{k \in \mathcal{A}} b_{i+k}$ so that $z_i = x_i \oplus h(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63})$. For each $0 \leq i < j \leq 19$, they applied the refined self-contained method together with the amplified phase to (z_i, z_j) and obtained a subset of the possible pre-images $X_{i,j}$ containing the right value with probability p. As x_i can be directly computed from the value keystream bit z_i and $s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}$ and b_{i+63}, they did not store x_i nor x_j in the $X_{i,j}$. As a result they claim that $X_{i,j}$ contains on average 848 elements (over 2^{10}) and $p = 89.64\%$.

The next step of the attack is to merge all those 190 sets to get a set X containing only values leading to the rightful first 20 bits of keystream. They claim that X would contain on average $2^{6.67}$ elements and the probability for the right value of the internal state to be in X would be around $p^{343} = (0.8964)^{343} = 2^{-54.09}$. The time complexity of the whole attack is then proportional to $|X| \times p^{-343}$.

4.3 Complexity correction

Experimental result on initialization. As for `A5/1`, we checked whether there are bias in the initialization phase of `Grain v1` which could explain the probability given by Zhang *et al.* in [ZXM18]. We have drawn uniformly at random 2^{30} keys and IVs, and for each them ran the initialization phase. We then looked at the 10 bits going through the function h to generate the 2 first keystream bits. As expected, we did not notice any bias in the distribution (see Figure 4).

Theoretical analysis. As explained Section 2.2, assuming all the 2^{160} possible internal states are equiprobable, and since the output function is balanced, the probability p for the right value to belong to $X_{i,j}$ has to be corrected to $848/1024 = 82.81\%$. In particular, the final probability becomes $(0.8281)^{343} = 2^{-93.32}$. But actually the whole attack is flawed. Indeed, since we merge 190 *independent* sets $X_{i,j}$ the probability for the right value to belong to X is p^{190} and not p^{343}. The mistake made by Zhang *et al.* lies in the merging process. First they construct the set $X_{0,1,2}$ by merging $X_{0,1}$, $X_{0,2}$ and $X_{1,2}$ claiming a probability of p^3 which is correct. Then they construct the set $X_{1,2,3}$ by merging $X_{1,2}$, $X_{1,3}$ and $X_{2,3}$ claiming a probability of p^3 which is also correct. But then they construct the set $X_{0,1,2,3}$ by merging $X_{0,1,2}$, $X_{1,2,3}$ and $X_{0,3}$ and claim a probability of $p^3 \times p^3 \times p = p^7$. This is wrong because $X_{0,1,2,3}$ is actually the merge of only 6 sets, $X_{1,2}$ being used twice, and the right probability is p^6. Thus the corrected probability for the right value to belong to X is $p^{190} = (0.8281)^{190} = 2^{-51.7}$. Surprisingly this is not so far from the $2^{-54.09}$ claimed by Zhang *et al.*. But the 20 keystream bits $z_0 \dots z_{19}$ depend on 118 (linear combinations of) state bits (see Table 5 in [ZXM18]). Thus, according to Theorem 1, to reach such probability the set X has to contain $2^{118-20} \times 2^{-51.7} = 2^{47.3}$ elements and not only $2^{6.67}$. As a consequence, the overall complexity of the attack is

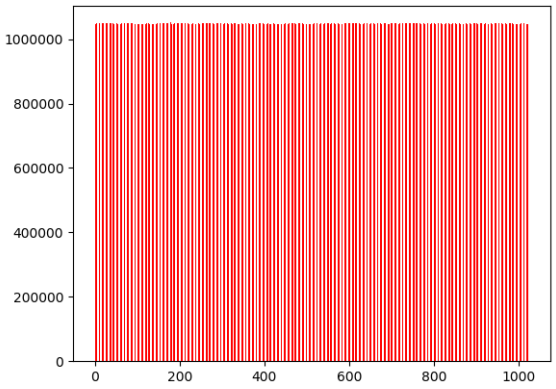

Figure 4: Histogram plotting how many times each of the 2^{10} possible two successive input of the function h were reached after 2^{30} `Grain v1` initialization using random key and iv values.

increased by a factor $2^{47.3+51.7-6.67-54.09} = 2^{37.24}$, making it slower than an exhaustive search.

Experimental result on p. Finally, as for `A5/1` we provide the following experiment to support our claim regarding the correct complexity of the attack presented in [ZXM18].

1. Randomly generate a 3-bit word $\mathbf{k}_s$
2. Run the refined self-contained method to obtain a set X of size $2^{14.2}$ which should contain the secret which generated $\mathbf{k}_s$ with probability $(0.8964)^3 = 0.7203$ according to Zhang *et al.*.
3. Do N times:
   (a) Randomly generate a key and an IV and run the initialization process.
   (b) Check whether the first three keystream bits from the current internal state match $\mathbf{k}_s$. If not repeat the previous step.
   (c) Check whether the value of the internal state part belongs to X.
4. Check whether the experimental probability matches the expected one.

We ran this experiment[2] 10 times with $N = 2^{26}$ and found the experimental probability to be very close to $(0.8281)^3 = 0.5679$, confirming the inaccuracy of the probability 0.7203 claimed by Zhang *et al.*. To ensure that the initialization process does not introduce and/or remove any biases, we repeated 10 times the experiment for random states during the keystream generation phase too. In more details, we sampled $R < 2^{10}$ and before generating the three keystream bits, we updated by R rounds the internal state. As expected, none of those experiments supported Zhang *et al.* claimed probability.

5 Conclusion

In this paper, we have shown the fast near collision attacks on both `A5/1` and `Grain v1` were flawed. More precisely, we have shown that the refined self-contained method cannot

[2] As for `A5/1`, we simply took for X the set of size $2^{14.2}$ composed of the values reached the most during the experiment.

magically restrict the search space and that all the probabilities related to the technique were overestimated. Once corrected, both those attacks do not improve on any previously known attacks. In the case of `Grain v1`, the overall complexity of the attack becomes much higher than for an exhaustive search of the key. Regarding `A5/1`, the attack becomes worst than Golić's one.

There are still mysteries about both [ZXM18] and [Zha19]. In particular, authors seem to have experimentally verified their claimed probabilities. They wrote they *have done a large number of experiments ... and almost all the experimental results conform to our theoretical predictions*. This statement is quite unlikely. Indeed, it was enough to add a loop in the publicly available C codes of their works to observe the deviation in the claimed probabilities.

Finally, we think it is important and crucial to evaluate and correct previous scientific works.

References

[BGHB11] Gilles Barthe, Benjamin Grégoire, Sylvain Heraud, and Santiago Zanella Béguelin. Computer-Aided Security Proofs for the Working Cryptographer. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 71–90. Springer, 2011.

[BGLB11] Gilles Barthe, Benjamin Grégoire, Yassine Lakhnech, and Santiago Zanella Béguelin. Beyond Provable Security Verifiable IND-CCA Security of OAEP. In Aggelos Kiayias, editor, *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, volume 6558 of *Lecture Notes in Computer Science*, pages 180–196. Springer, 2011.

[BGW99] Marc Briceno, Ian Goldberg, and David Wagner. A pedagogical implementation of A5/1. Technical report, May 1999. Available at http://www.scard.org.

[BR94] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer, 1994.

[CPS08] Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The Random Oracle Model and the Ideal Cipher Model Are Equivalent. In David A. Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2008.

[est09] The eSTREAM project. https://www.ecrypt.eu.org/stream/project.html, 2009.

[FOPS01] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA-OAEP is secure under the RSA assumption. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 260–274. Springer, 2001.

[Gol97] Jovan Dj Golić. Cryptanalysis of alleged A5 stream cipher. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 239–255. Springer, 1997.

[Gra01] Louis Granboulan. Flaws in differential cryptanalysis of skipjack. In *International Workshop on Fast Software Encryption*, pages 328–335. Springer, 2001.

[HJM07] Martin Hell, Thomas Johansson, and Willi Meier. Grain: a stream cipher for constrained environments. *IJWMC*, 2(1):86–93, 2007.

[HJMM06] Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. A Stream Cipher Proposal: Grain-128. In *Proceedings 2006 IEEE International Symposium on Information Theory, ISIT 2006, The Westin Seattle, Seattle, Washington, USA, July 9-14, 2006*, pages 1614–1618. IEEE, 2006.

[HKT11] Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 89–98. ACM, 2011.

[IIMP19] Akiko Inoue, Tetsu Iwata, Kazuhiko Minematsu, and Bertram Poettering. Cryptanalysis of OCB2: Attacks on Authenticity and Confidentiality. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 3–31. Springer, 2019.

[Jea16] Jérémy Jean. TikZ for Cryptographers. https://www.iacr.org/authors/tikz/, 2016.

[KRW99] Lars R Knudsen, Matthew JB Robshaw, and David Wagner. Truncated differentials and skipjack. In *Annual International Cryptology Conference*, pages 165–180. Springer, 1999.

[LKKD08] Jiqiang Lu, Jongsung Kim, Nathan Keller, and Orr Dunkelman. Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1. In Tal Malkin, editor, *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings*, volume 4964 of *Lecture Notes in Computer Science*, pages 370–386. Springer, 2008.

[Mur11] Sean Murphy. The return of the cryptographic boomerang. *IEEE Transactions on Information Theory*, 57(4):2517–2521, 2011.

[Sho01] Victor Shoup. OAEP reconsidered. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 239–259. Springer, 2001.

[WKD07] Gaoli Wang, Nathan Keller, and Orr Dunkelman. The delicate issues of addition with respect to xor differences. In *International Workshop on Selected Areas in Cryptography*, pages 212–231. Springer, 2007.

[Zha19] Bin Zhang. Cryptanalysis of GSM Encryption in 2G/3G Networks Without Rainbow Tables. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 428–456. Springer, 2019.

[ZXM18] Bin Zhang, Chao Xu, and Willi Meier. Fast Near Collision Attack on the Grain v1 Stream Cipher. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 771–802. Springer, 2018.

A Attached codes

The source codes of all our experiments is available in the public git repository:

https://gitlab.inria.fr/vmollima/a5-1sac.git

B Algorithms

Algorithm 3 The procedure used during a GSM session to generate 128 bits of keystream

$P = (F, K) = (P_{85}, \ldots, P_0)$ an 86-bit word composed of a 64-bit key K and a 22-bit frame F

for $i = 0$ to 85 **do** ▷ Initialization
  $R1[0],\ R2[0],\ R3[0] \leftarrow R1[0] \oplus P_i,\ R2[0] \oplus P_i,\ R3[0] \oplus P_i$
  clock $R1$, $R2$, $R3$ simultaneously
end for
for $i = 0$ to 99 **do**
  clock $R1$, $R2$, $R3$ asynchronously
end for

for $i = 0$ to 227 **do** ▷ Keystream generation
  $z_i = R1[18] \oplus R2[21] \oplus R3[22]$
  clock $R1$, $R2$, $R3$ asynchronously
end for

Output $z = (z_{227}, \ldots, z_0)$ ▷ the keystream

Algorithm 4 The initialization procedure and keystream generation of `Grain v1`

KEY a 80-bit key ▷ Parameters
IV a 64-bit initial value

Load KEY in ▷ Initialization
Load IV in
for $i = 0$ to 15 **do**
    $s_i \leftarrow 1$
end for
for $i = 0$ to 159 **do**
    compute $o = h^*(s_{79}, s_{54}, s_{33}, s_{15}, b_{16}, b_{78}, b_{77}, b_{75}, b_{69}, b_{48}, b_{36}, b_{23})$
    clock the LFSR and the NFSR with the following respective feedback bits:
    $f(s_{79}, s_{66}, s_{56}, s_{41}, s_{28}, s_{17}) \oplus o$
    $g(b_{16}, b_{17}, b_{19}, b_{27}, b_{34}, b_{42}, b_{46}, b_{51}, b_{58}, b_{64}, b_{65}, b_{70}, b_{79}) \oplus s_{79} \oplus o$
end for

for i **do** ▷ Keystream generation
    compute $z_i = h^*(s_{79}, s_{54}, s_{33}, s_{15}, b_{16}, b_{78}, b_{77}, b_{75}, b_{69}, b_{48}, b_{36}, b_{23})$
    clock the LFSR and the NFSR with the following respective feedback bits:
    $f(s_{79}, s_{66}, s_{56}, s_{41}, s_{28}, s_{17})$
    $g(b_{16}, b_{17}, b_{19}, b_{27}, b_{34}, b_{42}, b_{46}, b_{51}, b_{58}, b_{64}, b_{65}, b_{70}, b_{79}) \oplus s_{79}$
end for

Output $\{z_i\}_i$ ▷ the keystream

Algorithm 5 Observation of distribution after `Grain v1` initialization

Sample a 128-bit word KEY from /dev/rand ▷ Setup
Initialize a 128-bit word COUNTER at 0
Initialize a 2^{10} array called CONFIGURATION

for $i = 0$ to 2^{30} **do** ▷ Experiment
    Use AES-CTR(KEY, textsccounter) two 64-bit word called IVExp and KEYExp
    Sample a random integer RANDOM from /dev/rand
    Use AES-CTR(KEY, CTR + RANDOM) to add 16-bit to KEYExp
    Do a `Grain v1` initialization with KEYExp and IVExp
    Collect the 5 input bits of h in a variable RESULT
    Clock one more time the internal state
    Add the 5 new input bit of h to RESULT
    Increment CONFIGURATION[RESULT]
end for

Output: CONFIGURATION