# Decryption oracle slide attacks on T-310

Nicolas T. Courtois

University College London, Gower Street, London, UK

**Abstract.** T-310 is an important Cold War cipher [21]. It was the principal encryption algorithm used to protect various state communication lines in Eastern Germany throughout the 1980s. The cipher seems to be quite robust and until now no cryptography researcher has proposed an attack on T-310. In this article we study decryption oracle and slide attacks on T-310.

**Key Words:** Cold War, block ciphers, T-310, linear cryptanalysis, correlation attacks, software algebraic attacks, slide attacks, self-similarity attacks.

## 1 Introduction

T-310 is an important historical cipher which was used in Eastern Germany during the last period of the Cold War. According to [12, 21], in 1989 there were some 3,800 cipher machines in active service across all sorts of government, party and internal security services.

T-310 is a synchronous stream cipher which derives its keystream from the iteration of a relatively complex block cipher. The block cipher can be classified as "Unbalanced Feistel cipher" of so-called contracting type with 4 branches, cf. [18]. An important historical example of exactly such a cipher is the RC2 cipher by Rivest which was designed in 1989 cf. [17] with an (alleged) collaboration with the NSA. RC2 have been very widely used worldwide for real-life communications security, first in Lotus Notes software and later also in the S/MIME encrypted email standard of 1997. Another more academic example of (exactly) a compressing cipher with 4 branches is the McGuffin cipher proposed and cryptanalysed at FSE'94 [19]. The earlier RC2 has remained a trade secret for a longer time and only in 1997-1998 was it re-discovered and analysed (without great success) in the crypto community [17]. Another important historical cipher with a very large real-life footprint is the SHA-1 hash function which is a "Contracting Unbalanced Feistel" with 5 branches. It was developed by the well-known US-government funded Capstone project which also produced the Skipjack algorithm cf. [2]. Skipjack is unique type of cipher with 4 branches which are neither contracting nor expanding [18] with a lot of extra irregular structure [2, 16]. It is clearly stated in [2] that Skipjack was designed earlier in the 1980s, which would make this closer to being a contemporary of T-310. Research on the security of these ciphers is scarce. Even though Lotus Notes software has been an object of a number of controversies regarding deliberate weakening by the NSA, no convincing attack has been published to date against the RC2 cipher [17]. Similarly, to this day there is no attack on the full Skipjack cipher cf. [16]. Finally, until now, no attack of any sort whatsoever have been published on the T-310. In this article we provide a first non-trivial attack on T-310 cipher.

### 1.1 Basic Description of T-310

The main component of T-310 is a keyed permutation which also takes an IV which we will call "the T-310 block cipher". In this article we study an attack based on high-level properties which will require only a simplified description of the cipher. A full description of T-310 was published in Cryptologia in 2006 cf. [21].

The block size in T-310 is 36 bits only, the secret key has 240 bits and the IV has 61 bits. The block cipher is not used directly to encrypt, but it is iterated a large number of times. Some $13 \cdot 127 = 1651$ block cipher rounds are performed in order to extract as few as 10 bits called $(B_j, r_j)$ from the cipher's internal state, which will then be used to encrypt just one 5-bit character of the plaintext by a sort of double one-time pad cf. Fig. 1.

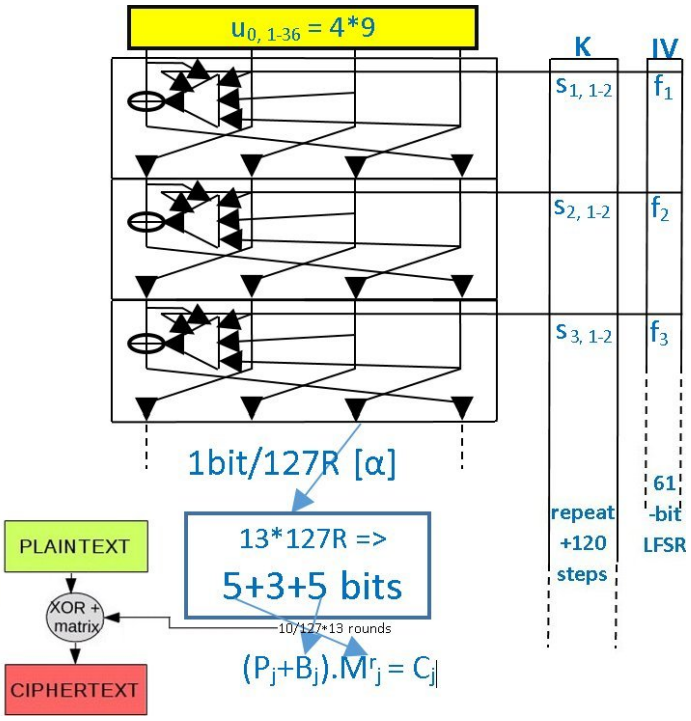


**Fig. 1.** T-310 Cipher.

The key used in different rounds repeats every 120 steps. This cyclic structure is the key vulnerability which we will exploit in this article. In contrast the IV bits are expanded with an LFSR which produces a sequence with a very large period. This makes T-310 potentially stronger than, for example, GOST or KeeLoq, where the exact same permutation consisting of many rounds is repeated many times, which is a source of numerous self-similarity attacks [4, 9, 11, 6, 1]. However this sequence remains entirely predictable for the attacker and it is possible to design a self-similarity attack on T-310.

### 1.2 On the Strength Of Individual Encryption Rounds

The structure of one round of T-310 is shown in Fig. 2. One round $m \geq 1$ uses 2 bits of the key $s_{m,1}, s_{m,2}$ and 1 bit derived from IV $f_m$. It is a peculiar variant of a so-called "Contracting Unbalanced Feistel cipher" with 4 branches, cf. [18]. The original Feistel cipher construction had 2 branches and was invented around 1971 [14]. Then East German cipher designers had already in 1970s [21] mandated a substantially more complex internal structure.

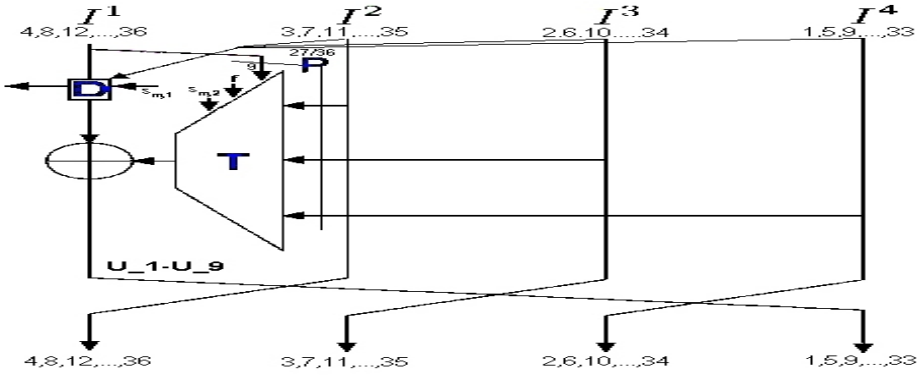


**Fig. 2.** The internal structure of one encryption round of T-310.

This peculiar internal structure is largely irrelevant in this article and we will not study it. The attack which we will study does not depend on this structure The only property which will matter in this article is that some bits CAN be correlated to one input bit at the same position few rounds earlier, for example bit $\alpha = 30$ after $d = 7$ rounds. This property is a property of the whole round which depends on countless technical details such as the choice of the Boolean function inside $T$, the connections of $T$, the connections in Fig. 2, and the long-term key $D, P$, cf. [13]. In this article we are just going to show one example of such correlation, which is relatively strong, and the resulting attack. It is easy to see that many other [possibly weaker] correlations of this type exist for various versions of T-310 and potentially **just one** would be sufficient to make our later attack work.

## 2 A Short Description of T-310

Following Fig. 2, one round of encryption is given by

$$(u_{i,1-36}) = \phi\left(s_{i,1}, s_{i,2}, f_i;\ \ u_{i-1,1-36}\right)$$

Given Fig. 1, in order to fully specify the cipher T-310 we need exactly:

1. To specify $u_0$ the initial 36-bit state $I^{1-4}$ of the block cipher which is a constant equal to 0xC5A13E396, cf. [21].
2. To specify the internals of one round $\phi : \{0,1\}^3 \times \{0,1\}^{36} \to \{0,1\}^{36}$, which is one full round of encryption, which uses 3 bits of key+IV per round, and (inside this) for the round function $T$. We refer to [21] for a detailed description.

3. To specify $D, P$, e.g. one of the actual keys from 1977-1990 listed in [13].
4. We specify how the 3 bits of the key and IV $(f_m, s_{m,1}, s_{m,2})$ used by in each round are generated in round $m \geq 1$.
5. The $f_m$ sequence is obtained with an LFSR and it starts at $f_{-60}, \ldots, f_0$ which is the 61-bit IV. These bits are not used in encryption and the first bit used is $f_1$. The LFSR is defined by:
$$f_i = f_{i-61} \oplus f_{i-60} \oplus f_{i-59} \oplus f_{i-56}.$$
6. The key has 240 bits $s_{1-120,1-2}$. Contrary to popular belief[1] the effective key size is NOT reduced to 230 bits, cf. [21, 20]. Key bits are repeated every 120 rounds as follows:
$$s_{m+120,1-2} = s_{m,1-2}.$$

This description is sufficient for the purpose of this article, we refer to [21] for more details. On Fig. 3 and earlier Fig. 1 we show how all these things come together.

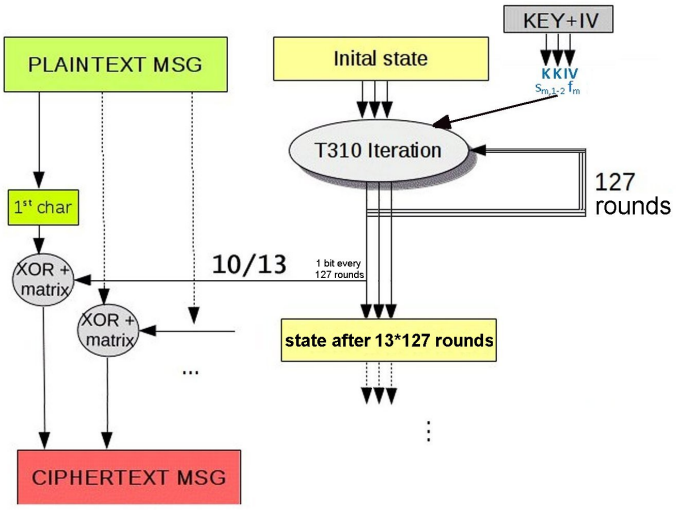


**Fig. 3.** T-310 Encryption Process.

### 2.1 How Encryption is Performed - Double One-Time Pad

From our iterated block cipher we extract just 1 bit per 127 rounds: $u_{127,\alpha}, u_{2\cdot 127,\alpha}, u_{3\cdot 127,\alpha}, \ldots, u_{13\cdot 127,\alpha}$ and for every 13 bits we discard 3 and use 5+5 bits. Here $\alpha \in \{1 \ldots 36\}$ is a constant which is a part of the long-term key. More precisely we put:

$$C_j = (P_j \oplus B_j) \cdot M^{r_j},$$

[1] In [21, 20] we read that 10 out of 240 bits should be parity bits. However according to specialists who studied the actual T-310 machines [12], the parity bits specified in 1980 in [20] were NOT subsequently used in real life transmissions.

where $P_j/C_j$ is the plaintext/ciphertext character on 5 bits, respectively, then $B_j = (a_{7+13(j-1)}, \ldots, a_{11+13(j-1)})$ are 5 consecutive bits out of the 13 previously discussed and $r_j$ is a "stepping" output which is derived from the FIRST consecutive 5 bits out of the 13 as follows:

$$r_j = \begin{cases} 0 & \text{if} \quad R_j = (0,0,0,0,0) \\ 0 & \text{if} \quad R_j = (1,1,1,1,1) \\ 31 - r & \text{if } R_j \cdot M^r = (1,1,1,1,1) \end{cases}$$

where $R_j \stackrel{def}{=} (a_{1+13(j-1)}, \ldots, a_{5+13(j-1)})$ and

$$M = \begin{pmatrix} 0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,1 \\ 0\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0 \end{pmatrix}, \quad \text{which is such that} \quad M^{31} = Id.$$

**A Strong Point in T-310.** This selection of extremely few bits is where T-310 appears to be a particularly strong cipher, potentially actually stronger than other comparable classical Feistel ciphers such as RC2, DES, and Skipjack. This is an incredibly **low quantity** and the cryptanalytic literature knows extremely few examples where the cipher would actually be broken under such difficult circumstances. One major example is the so-called "Courtois Dark Side Attack" on MiFare classic [7] one of the most widely used cryptosystems on our planet, with approximately 2 billion RFID smart cards sold. In this attack the attacker obtains only 4 bits from each encryption [7]. Here we can obtain only 1 bit per 127 rounds of encryption. The more rounds, the harder it becomes to develop any sort of cryptographic attack.

### 2.2 Estimating Strength Against Direct Software Algebraic Attacks

Here the security of T-310 can be compared to KeeLoq, also a block cipher which locally looks like a stream cipher, and which has hundreds of rounds. General-purpose software key recovery attacks on KeeLoq with a SAT solver can recover the key for about 160 rounds only, cf. [6, 1] for attacks running within hours/days on a PC. The complexity of KeeLoq is lower than T-310: in KeeLoq we have 1 Boolean function with 5 inputs per round, in T-310 we have 4 evaluations of a Boolean function with 6 inputs per round. it may be reasonable to expect that a SAT solver can break 120 rounds of the T-310 block cipher in a similar way as it can break 8 rounds of GOST; see [10] and Table 1, Section 9.1. in [9].
**Application:** Initially it seems that the attacker has little choice other than to work on the first character of the ciphertext $C_1$ and try to develop an attack on $11 \cdot 127 = 1397$ rounds. The main point of this article is that there exists a non-trivial method which allows the attacker to obtain Plaintext/Ciphertext (P/C) pairs for as few as 120 rounds.

## 3 Decryption Oracle Attacks - Recovering the Keystream

A plausible attack scenario is that the attacker would have access to a decryption oracle. The attacker can send any IV and the ciphertext and obtain the plaintext. For the $j$-th character we have:

$$C_j = P_j \cdot M^{r_j} \oplus B_j \cdot M^{r_j}$$

for every $j < k$ where ciphertexts submitted to the oracle have length $k$ characters. Then in all these encryptions $r_j$ and $B_j$ will be the same

$$C_j \oplus C'_j = (P_j \oplus P'_j) \cdot M^{r_j} \quad \text{for all } \ 0 \leq j < k.$$

This allows to recover $M^{r_j}$ uniquely in a proportion of 1-1/32 of cases where $C_j \neq C'_j$ and the attacker could chose ciphertexts such that $C_j \neq C'_j$ for most pairs. Moreover following Section 2.1, $M^{r_j}$ does almost always allow to determine $R_j$ except when $r_j = 0$. One of the two problematic events happens with overall probability less than 2/32. Overall in at least 30/32 of the cases over all possible pairs, $P/C, P'/C'$, the 5 bits of $R_j = (a_{1+13(j-1)}, \ldots, a_{5+13(j-1)})$ are uniquely determined, because $r_j \neq 0$, which avoids the ambiguity[2]. We can then also determine $B_j = (a_{7+13(j-1)}, \ldots, a_{11+13(j-1)})$. However we cannot hope to ever recover any $a_j$ with $j \equiv 0, 6$, or 12 modulo 13, because these $a_j$ are never used for encryption. Moreover we also make a deliberate choice NOT to recover all the $a_i$ which could be recovered, in order to minimize the data [decryption oracle query] complexity of our later attack. We have the following result:

**Theorem 3.0.1 (General Decryption Oracle Attack).** For every $IV$ chosen by the attacker, and for every $k \geq 1$, the attacker can obtain a proportion of $30/32 \cdot 10/13 + 1/32 \cdot 5/13 \approx 0.73$ of the internal keystream bits $a_{1-13k}$ with a computation effort of about $2k$ and with about $K = 2$ "Chosen IV and Chosen Ciphertext" (CIVCC) queries on average, with one fixed chosen IV and random ciphertexts, and with ciphertext lengths of about $k$ characters. For the remaining values $a_i$ we make the algorithm return "don't know".

*Proof:* In this article we made a "minimalistic" choice of $K = 2$. Exactly, and only, two things can go wrong for our pair of decryptions obtained from the oracle. Either we have $C_j = C'_j$ or $r_j = 0$. Avoiding both cases happens with probability at least 30/32. In this case we can determine 10/13 the bits uniquely from the decrypted pair. We also have a case where $C_j \neq C'_j$ but unhappily $r_j = 0$ and $R_j$ cannot be determined for sure (ambiguity), in this case however $B_j$ can be obtained from $C_j = P_j \cdot M^{r_j} \oplus B_j \cdot M^{r_j} = P_j \oplus B_j$. This happens with probability about 1/32, in this case we only get 5/13 of the bits of $B_j = (a_{7+13(j-1)}, \ldots, a_{11+13(j-1)})$.

[2] When $r_j = 0$ we are not able to determine $R_j$, either $R_j = (0, 0, 0, 0, 0)$ or $R_j = (1, 1, 1, 1, 1)$.

## 4 On the Existence of Suitable $\alpha \to \alpha$ Correlations

We consider some relation of type

$$120s = 127t + d,$$

where $d$ is small in absolute value and also $s$ and $t$ are not too large. For example $(s, t, d) = 18, 17, 1$ or $(s, t, d) = 1, 1, -7$. In this article we will concentrate on the case of $d = \pm 7$. Other cases will be studied in future works. Then we want two bits used for encryption in two encryptions shifted by $120s$ rounds to be correlated, cf. Fig 5 page 8. In other words order to make our slide attack we need a correlation property of type: one special bit $\alpha \in \{1 - 36\}$ of the block cipher state is correlated with the same bit $\alpha$ after $d$ rounds for some small $d$.

$$s_{i,\alpha} \stackrel{?}{=} s_{i+d,\alpha} \quad \forall i$$

This can be seen as a special case of linear cryptanalysis (LC). However we only look at invariant linear characteristics with Hamming weight 1, which will be substantially less frequent. In general the answer depends on the values of $d$ and the choice of the term key LZS with specific values for $D, P, \alpha$. We conjecture that for every $D, P, \alpha$ there exists several $d$ such that our attack can be made to work (with $s > 1$ it will be harder).

In this article we present a simple attack with $s = 1$ and $d = \pm 7$ and we will later evaluate the complexity of our attack on one key 701 which we have generated ourselves and which exhibits a suitable correlation. For these parameters $s, d$ we have NOT found a more convincing real-life example which indicates that the East-German cryptologists have somewhat managed to prevent the basic slide attack, described in this article, from being effective in practice. In general however, probably there is no way to prevent such attacks from working with a sufficiently large $d$ or/and with a weaker correlation. For this purpose we are going to use the key 701 for which we have an invariant linear characteristic [30] $\to$ [30] for some 20 % of key and IV choices. Let 701 be a key defined by `P=31,10,33,6,32,8,5,3,9,15,13,26,19,28,21,7,16,25,34,12,22,17,35,29,30,23,4` and `D=4,2,17,32,12,35,0,24,20`. We also recall that key 27 is defined in [13]. We have:

**Table 1.** Examples of a one-bit invariant correlation in T-310

| LZS nb | rounds | input $\to$ output | bias | probability |
|---|---|---|---|---|
| 701 | 7 | [30] $\to$ [30] | $2^{-11}$ | 0.2 |
| 27 | 16 | [27] $\to$ [27] | $2^{-7.2}$ | 0.2 |

It is worth noticing that some very good results and for as many as 16 rounds can be obtained for key 27, however this key is an anomalous key which was ever approved for use, cf. [13].

## 5 A Decryption Oracle with a Slide Attack

Now we are going to design our slide attack [15, 3, 6] We want to exploit the self-similarity of the T-310 block cipher: the key bits repeat every 120 rounds, and we need to adjust the IV bits in order to obtain identical permutations. Then the question will be whether these identical permutations can have identical inputs. Traditionally researchers call such pairs of inputs a 'slid pair' [3]. Here is our first basic slide attack. Let $s$ again be a small integer with $120s = 127t + d$, where $d$ is small in absolute value. In this article we only study the case $(s, t, d) = 1, 1, -7$. Other cases will be studied in future work. Now the key point is that **if** by some sort of "happy" accident for some encryption with some IV, we have

$$u_{120s} = u_0 = 0xC5A13E396,$$

then the attacker can detect this fact efficiently, **if** there exist correlations on bit $\alpha$ for $d$ rounds, cf. Section 4 **and if** the attacker has access to a certain type of chosen IV and chosen ciphertext attack (with partial recovery of the internal keystream) such as in Thm. 3.0.1.

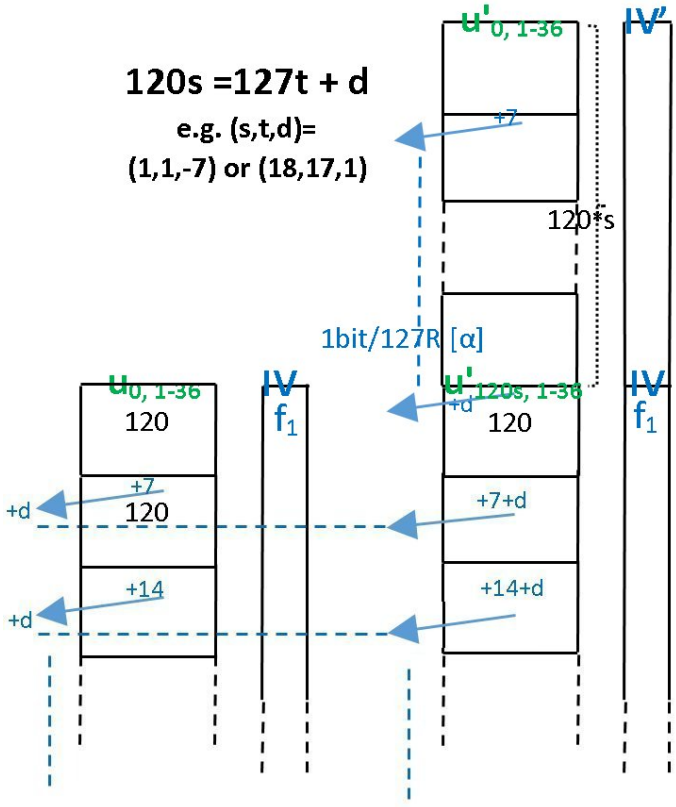


**Fig. 4.** Slide Attacks on T-310 (case $d > 0$).

**Theorem 5.0.1 (Sliding Property Detection with a Decryption Oracle).** For every $IV$ chosen by the attacker, and for every $s \geq 1$, the attacker can detect with near-certainty if $u_{120s} = u_0$ for the unknown key, by requesting a decryption of $K = 2$ ciphertexts for this IV and another related IV' which we specify below, with length $120s + k$ each, and with time complexity about $2k$ as in Thm. 3.0.1, where $k$ is the decryption oracle query data capacity requested for a fixed set of parameters $s, d$, which will be determined later to achieve the desired confidence level four our distinguisher.

*Proof:* We describe how the distinguisher works in four Steps.

**Step 1.** We select two IVs which are distant by $120s$ steps of our 61-bit LFSR, called $IV, IV'$. We recall that $120s$ mod 127 is small. We recall that the key is repeated after every multiple of 120 rounds, but the keystream is extracted every 127 rounds. Then IF in some two encryptions have the same state

$$u_0 = u'_{120s} \quad \text{[Sliding Assumption]}$$

which occurs with probability $2^{-36}$ THEN we have

$$u_i = u'_{120s+i}$$

for any number of steps $i \geq 0$.

**Step 2.** Then for both encryptions the attacker can recover most of the keystream with $K = 2$ decryption queries per IV, cf. Thm. 3.0.1.

**Step 3.** We have $120s = 127t + d$ with $d$ small. This means that IF again $u_0 = u'_{120s}$ the keystream extracted from the second encryption is shifted by $127t + d$, i.e. it is extracted at $t$ "big" $a_i$-scale steps later with 127 rounds each, and with a $\phi^d$ offset. We can hardly hope that these bits will be identical BUT we can hope they will be in some cases correlated. We have

$$a_j = u_{127j,\alpha}$$

and

$$a'_j = u'_{127j,\alpha} = u'_{127(j-t)-d+120s,\alpha} = u_{127(j-t)-d,\alpha}$$

### 5.1 Step 4 - Correlation Analysis

Now as a first approximation, we see that the attacker has access to the sequences $u_{127j,\alpha}$ and $u_{127j'-d,\alpha}$ for any $j, j'$ which are shifted by $d = 1$ encryption round $\phi$. The question now is if there is a **correlation** between these 2 bits which makes that the slide assumption $u_0 = u'_{120s}$ will be detected. In this article we put LZS=701, $d = -7$ and we have observed that $u_{127j+7,\alpha} = u_{127j,\alpha}$ with probability $0.5 - \varepsilon$ with $\varepsilon = 2^{-11}$. This means that the attacker can easily detect if our sliding condition on 36 bits is true for $\alpha = 30$.

More precisely, it is not quite correct to say that the attacker has access to the sequences $u_{127j,\alpha}$ and $u'_{127j,\alpha}$ for every $j$. Following cf. Thm. 3.0.1 only 73 % of these bits can be recovered on each side. This makes that only some pairs $u_{127j,\alpha}, u'_{127(j-17),\alpha}$ will actually be available, in fact a proportion of $(0.73)^2 \approx 0.53$. This is of course sufficient to detect the correlation with about twice the value $k$ than otherwise needed, and we will estimate $k$ below. This ends the proof of Thm. 5.0.1.

### 5.2 Sliding Step - Summary

We see that the attacker can obtain P/C pairs on 36+36 bits for the T-310 block cipher for $120s$ rounds away and with arbitrarily chosen IVs, and where the second IV is obtained by clocking the LFSR $120s$ steps backwards.

More precisely, following Thm. 5.0.1 the attacker can **detect** if the internal states on the 36 bits are identical. He can know with near-certitude that

$$u_0 = u'_{120s} \quad \text{[Sliding Assumption]}$$

is true for some pairs $IV, s$ and for the current secret key. This condition is true with probability $2^{-36}$ in general and when it occurs the attacker will detect it.

### 5.3 Data Complexity Required in Our Attack

At this stage we see that the attacker can generate P/C pairs for 120 rounds given that $s = 1$, and following Section 2.2 key recovery for 120 rounds with a SAT solver is assumed to be feasible. It is then easy to see that the we need to generate 7 such P/C conditions on 36 bits: one is not sufficient to uniquely determine a key on 240 bits. We need to estimate the data complexity needed to see if $u_{120s} = 0xC5A13E396$ will be simultaneously true in 7 cases with probability of at least 1/2 and to reliably discard as many as $2^{39} - 7$ cases. Therefore we need to operate with a precision which is sufficient to have the standard Gauss error function erf(), to predict less than one false positive in $2^{39}$ experiments. We must be able to reject most cases with Thm. 5.0.1 operating at $z$ standard deviations, where $z$ is such that that $erf(z/\sqrt{2}) < 2^{-39}$, which gives $z = 7$, see the table in [22].

The standard deviation for $N$ events, where equality of some two bits of type $u_{127i,\alpha}$ holds in Thm. 5.0.1, which is assumed true with probability $1/2 \pm \varepsilon$, with $\varepsilon = 2^{-11}$ here, will be about $\sqrt{N}$ and the deviation in observed probability will be $\sqrt{N}/N$. In order to detect correlations with confidence at or exceeding 7 standard deviations we need, approximately, $7\sqrt{N}/N \leq \varepsilon$. This leads to $N \geq 7^2 \cdot \varepsilon^{-2}$. Now, not all bits $u_{127i,\alpha}$ are simultaneously known in 2 distinct encryptions. Inside $13k$ possible bits $a_i$ for each of $K = 2$ decryptions with $k$ characters, only 73 % are available, and out of these only 73 % are such that the correlated bit for the other decryption is also available to the attacker. This leads to $N \approx 13k \cdot (0.73)^2 \approx 6.9k$. We need $k = N/6.9 \approx 7\varepsilon^{-2}$ with $\varepsilon = 2^{-11}$.

### 5.4 A Basic Full Sliding Key Recovery Attack with $d = -7$

Below we describe a full combined attack.

1. We consider key 701, $d = -7$ and $s = 1$.
2. The attacker will try some $7 \cdot 2^{36} \approx 2^{39}$ random $IV_i$ on 61 bits. He can then expect that there exists some $2^{39-36} \approx 7$ "good" IVs where he has $u_{120s} = u_0 = 0xC5A13E396$. At this moment he does not know which 7 IVs are the "good" ones.
3. For each of $IV_i$, $i = 1 \ldots 2^{39}$ the attacker will step the $IV$ exactly $120s$ steps backwards to obtain $IV'_i$.
4. The pairs $IV, IV'$ are always shifted by a multiple of 120 rounds, so that they key bits $s_{i,1-2}$ are also aligned.

5. Memory requirements are very small.
6. Then we apply Thm. 5.0.1 cf. also Fig. 4. The attacker - with the help of a decryption oracle - can see if $u_{120s} = u_0 = 0xC5A13E396$ by aligning 2 sequences $a_j$ and $a'_{j+t}$, where only $0.73^2 \approx 0.53$ of the pairs are known to the attacker, discarding all the pairs where either of $a_j, a'_{j+t}$ is not known, and counting how many times we have $a_j = a'_{j+t}$.
7. Following Section 5.3, the attacker needs to select 7 cases where $u_{120s} = 0xC5A13E396$ will be simultaneously true and reliably discard $2^{39} - 7$ cases. This leads to $k = N/6.9 = 7^2 \cdot \varepsilon^{-2}/6.9 \approx 7\varepsilon^{-2}$ with $\varepsilon = 2^{-11}$. We also need $120s$ more characters which is negligible.
8. Overall our attack requires $k = 7\varepsilon^{-2}$ characters of encrypted data where $\varepsilon = 2^{-11}$. We need about $k \approx 2^{25}$ characters of decrypted data per decryption query.
9. The data complexity is about $K \cdot 7 \cdot 2^{39} \approx 2^{43}$ chosen IV chosen ciphertext decryption queries, which are $2^{25}$ characters each in length.
10. The time complexity is about $2^{39} \cdot K \cdot 2^{25} \approx 2^{65}$ CPU clocks spent in examining correlations plus the time to recover the key from 7 P/C pairs for 120 rounds by a SAT solver attack. As long as this step takes less[3] than $2^{65}$ CPU clocks, this will NOT change the complexity of our attack. For the time being we assume it does.

Overall we see that we can recover the 240-bit key of T-310 with about $2^{43}$ chosen IV chosen ciphertext decryption queries with messages of less than $2^{25}$ characters each, cf. Section 5.4. The time required is about $2^{65}$ CPU clocks and the memory required is small.

## 6 Conclusion

T-310 is an important Cold War cipher which uses a block cipher from which it extracts extremely few bits for the actual encryption. This property makes that T-310 seems substantially stronger than other ciphers from the same historical period, such as RC2, DES, and Skipjack. The cryptanalytic literature knows extremely few examples where the cipher would actually be broken under such difficult circumstances. In one such example the attacker obtains only 4 bits from each larger encryption [7]. In T-310, bits from rounds as high as 1397 are used to encrypt just the first character. The key question for cryptanalysis of T-310 will then be, is there a "reduction" method or a self-similarity attack, where the attacker can obtain data for a substantially smaller number of rounds. For example, in [4, 9, 11] we discover many different methods to transform an attack on 8 rounds of GOST, into an attack on 32 rounds of GOST. The same occurs for KeeLoq, where a key recovery attack on 64 rounds allows us to break the full 528 rounds [6, 1]. In this article we show a non-trivial attack which gives the attacker the ability to generate pairs for 'only' 120 rounds of T-310.

[3] For example, in Table 1, Section 9, page 25, in [9], the time complexity decreases as the number of P/C pairs grows. We expect a similar result here and arguably 120 rounds of T-310 are the equivalent of 8 rounds of GOST in terms of complexity and key usage.

Our main result is to show how to recover the 240-bit key of T-310 with about $2^{43}$ chosen IV / chosen ciphertext decryption queries, which need to be $2^{25}$ characters long. Then if a suitable SAT solver software attack step can recover the key from 7 P/C pairs for 120 rounds in time less than $2^{65}$, then we get an overall attack with complexity $2^{65}$ to recover a 240-bit key with small memory requirements. Our current attack was designed for just one vulnerable long-term key, 701, and has $d = -7$. It seems that historical keys which were approved for use have a good level of resistance against this attack. Future research will show what will be the optimal parameters $s, t, d$ to obtain the best possible slide attack for various actual historical long-term keys listed in [13].

## References


1. Gregory V. Bard, Shaun V. Ault and Nicolas T. Courtois: *Statistics of Random Permutations and the Cryptanalysis Of Periodic Block Ciphers,* In Cryptologia, Vol. 36, Iss. 3, pp. 240-262, July 2012.
2. E.F. Brickell, D.E. Denning, S.T. Kent, D.P. Maher, W. Tuchman, *SKIPJACK Review Interim Report The SKIPJACK Algorithm*, 8 June 2011, archived at `https://web.archive.org/web/20110608020227/ http://www.cs.georgetown.edu/ denning/crypto/clipper/SKIPJACK.txt`
3. A. Biryukov, D.Wagner: *Slide Attacks,* In proceedings of FSE'99, LNCS 1636, pp. 245-259, Springer, 1999.
4. Nicolas Courtois: *Security Evaluation of GOST 28147-89 In View Of International Standardisation,* in Cryptologia, volume 36, issue 1, pp. 2-13, 2012.
5. Nicolas Courtois, Gregory V. Bard: *Algebraic Cryptanalysis of the Data Encryption Standard,* In Cryptography and Coding, $11^{th}$ IMA Conference, pp. 152-169, LNCS 4887, Springer, 2007.
6. Nicolas Courtois, Gregory V. Bard, David Wagner: *Algebraic and Slide Attacks on KeeLoq,* In FSE 2008, pp. 97-115, LNCS 5086, Springer, 2008.
7. Nicolas T. Courtois: *The Dark Side of Security by Obscurity and Cloning MiFare Classic Rail and Building Passes Anywhere, Anytime,* In SECRYPT 2009 – International Conference on Security and Cryptography: pp. 331-338. INSTICC Press 2009, ISBN 978-989-674-005-4.
8. Nicolas T. Courtois: *New Frontier in Symmetric Cryptanalysis,* Invited talk at Indocrypt 2008, 14-17 December 2008. Extended version of slides presented: `http://www.nicolascourtois.com/papers/ front_indocrypt08.pdf`.
9. Nicolas Courtois: *Algebraic Complexity Reduction and Cryptanalysis of GOST,* Monograph study on GOST cipher, 2010-2014, 224 pages, available at `http://eprint.iacr.org/2011/626`.
10. Nicolas T. Courtois: *Low-Complexity Key Recovery Attacks on GOST Block Cipher,* In Cryptologia, vol. 37, issue 1, pp. 1-10, 2013.
11. Nicolas Courtois: *On Multiple Symmetric Fixed Points in GOST,* in Cryptologia, Iss. 4, vol 39, 2015, pp. 322-334.
12. Jörg Drobick: *T-310/50 ARGON,* a web page about T-310 cipher machines consulted 19 March 2017, `http://scz.bplaced.net/t310.html`
13. Jörg Drobick: *T-310 Schlüsselunterlagen,* a web page which enumerates several different known long-term keys for T-310 from 1973-1990, consulted 21 January 2017, `http://scz.bplaced.net/t310-schluessel.html`

14. H. Feistel, W.A. Notz, J.L. Smith, *Cryptographic Techniques for Machine to Machine Data Communications,* Dec. 27, 1971, Report RC-3663, IBM T.J.Watson Research.
15. E. K. Grossman, B. Tuckerman: *Analysis of a Weakened Feistel-like Cipher,* 1978 Intern. Conf. Communications, pp.46.3.1-46.3.5, Alger Press Limited, 1978.
16. Jongsung Kim, Raphael C.W. Phan: *Advanced Differential-Style Cryptanalysis of the NSA's Skipjack Block Cipher,* In Cryptologia, vol. 33, iss. 3, pp. 246-270, 2009.
17. Lars R. Knudsen, Vincent Rijmen, Ronald L. Rivest, Matthew J. B. Robshaw: *On the Design and Security of RC2,* In FSE'98, LNCS 1372, pp. 206–221, Springer, 1998.
18. Jacques Patarin, Valérie Nachef, Côme Berbain: *Generic Attacks on Unbalanced Feistel Schemes with Contracting Functions,* in Asiacrypt 2006, pp. 396-411, LNCS 4284, Springer 2006.
19. Vincent Rijmen, Bart Preneel: *Cryptanalysis of McGuffin,* In FSE '94, pp. 353–358, Springer, 1994.
20. Referat 11: *Kryptologische Analyse des Chiffriergerätes T-310/50. Central Cipher Organ, Ministry of State Security of the GDR, document referenced as 'ZCO 402/80', a.k.a. MfS-Abt-XI-594, 123 pages, Berlin, 1980.*
21. *Klaus Schmeh: The East German Encryption Machine T-310 and the Algorithm It Used,* In Cryptologia, 30: 3, pp. 251 – 257, 2006.
22. Wikipedia article: *Standard Deviation,* consulted 13 Mayb 2017, `http://en.wikipedia.org/wiki/Standard_deviation`.