IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2021, No. 3, pp. 36–53. DOI:10.46586/tosc.v2021.i3.36-53

Cryptanalysis of the SoDark Cipher for HF Radio Automatic Link Establishment

Marcus Dansarie[1,2]

[1] Swedish Defence University, Stockholm, Sweden
marcus.dansarie@fhs.se
[2] University of Skövde, Skövde, Sweden

Abstract. The SoDark cipher is used to protect transmitted frames in the second and third generation automatic link establishment (ALE) standards for high frequency (HF) radios. The cipher is primarily meant to prevent unauthorized linking and attacks on the availability of HF radio networks. This paper represents the first known security analysis of the cipher used by the second generation ALE protocol—the de facto world standard—and presents a related-tweak attack on the full eight round version of the algorithm. Under certain conditions, collisions of intermediate states several rounds into the cipher can be detected from the ciphertext with high probability. This enables testing against the intermediate states using only parts of the key. The best attack is a chosen-ciphertext attack which can recover the secret key in about an hour with 100% probability, using 2^9 chosen ciphertexts.

Keywords: automatic link establishment · ALE · SoDark · cryptanalysis · block cipher

1 Introduction

This paper presents related-tweak attacks on the full eight round SoDark cipher used to protect frames in the second generation (2G) automatic link establishment (ALE) standard for high frequency (HF) radios. SoDark is a 24-bit block cipher with a 56-bit key and a 64-bit tweak [DoD17]. It is described further in Section 2. The cryptanalysis presented here may also be relevant for the third generation (3G) ALE standard which uses the same algorithm, but with sixteen rounds instead of the eight used in 2G ALE.

1.1 HF radio and automatic link establishment

The HF radio band between 3 and 30 MHz makes it possible to transmit radio messages globally without any supporting infrastructure. This is a unique property of the HF band. Therefore, HF radio is widely used by military, emergency response, and humanitarian aid organizations around the world. The propagation conditions for HF radio constantly change and are dependent on numerous factors, including time of day, season, space weather, and properties of the transmitting and receiving equipment.

Establishing successful communications via HF radio has traditionally required skilled radio operators. In order to decrease the dependence on operator skill, protocols for ALE over HF radio were developed during the 1980s. In an ALE system, the frequency selection and call to a receiving station is performed automatically by the radio equipment in lieu of manual frequency selection and voice call by an operator. In ALE, frequency selection is normally performed based on a radio propagation model and can be aided by automated sounding calls. If a call is successful and answered by the receiving ALE station, the established channel is handed over to a higher level protocol which can be, for example, a

Received: 2021-06-01 Accepted: 2021-08-01 Published: 2021-09-17

standard radio voice call or a data transmission. Upon completion of the call, ALE frames are again automatically exchanged to disconnect the link and make the radios available for new calls [Int02].

ALE standards of the first generation were proprietary and radios from different manufacturers could not interoperate. This was rectified by standardizing the physical and data-link layers of second generation (2G) ALE in the US standards FED-STD-1045 and MIL-STD-188-141. Later versions also introduced third generation (3G) and fourth generation (4G) ALE standards, with added functionality and increased performance [Int02, JKF$^+$12]. The North Atlantic Treaty Organization (NATO) has also adopted 2G and 3G ALE as the NATO standard STANAG 4538.

In 2G ALE, radios exchange frames made up of several 24-bit words, each consisting of a 3-bit preamble and three 7-bit characters. As an example, the simplest frame used when calling another station contains three 24-bit words. The first has the preamble "TO" (binary 010) and contains the called station's callsign. The second word is a repetition of the first. The third has the preamble "TIS" (binary 101) and contains the calling station's callsign. Callsigns longer than three characters can be transmitted using extension words with the preamble "DATA" (binary 000) after a "TO" or "TIS" word [DoD17].

ALE frames are vulnerable to spoofing. By transmitting crafted frames, an attacker can affect availability in the radio network by, for example, disconnecting established calls or establishing false links, thereby preventing legitimate traffic. To prevent this and to prevent data such as callsigns from being transmitted in the clear, the standard includes an optional linking protection mode. With linking protection enabled, all ALE words are encrypted with a 24-bit block cipher. The cipher is always used in ECB mode, i.e. all words are encrypted independently.

The SoDark cipher was originally developed for use in 2G ALE [Joh92] and was later adopted for use with 3G ALE, where the cipher is extended to include a version with 48-bit block size. The 24-bit versions in 2G and 3G ALE differ only in the number of rounds. In 2G ALE, the cipher is known as the Lattice algorithm and in 3G ALE as SoDark [DoD17]. The latter name is used here to avoid confusion with the unrelated field of lattice cryptography.

ALE networks, both with and without linking protection enabled, are in active use by numerous government and non-government organizations all over the world [ALE21]. Although successor standards have been available for a long time, 2G ALE is still the de facto world standard [Int02, Sig21]. This is likely because of compatibility requirements with older generations of radio equipment that are still in use.

1.2 Differential cryptanalysis

Differential cryptanalysis [BS93] is one of the most powerful general attacks on block ciphers. It was developed as an attack on the Fast Data Encipherment Algorithm (FEAL) and was later adopted for the Data Encryption Standard (DES) [Bih06]. Since then, the technique has proven to be useful in the analysis of many other ciphers. A central idea of differential cryptanalysis is that of differences between states in two parallel applications of the same cipher. Most commonly, differences are defined as the exclusive or (XOR) of two states. In ciphers where key addition is performed with XOR, the difference between two states before and after the addition does not change. This becomes very useful in cryptanalysis, as it can be used to deduce differences in internal cipher states without any knowledge of the key. The structure of SoDark makes it possible to calculate very high probability differences many rounds into the cipher. In ciphers which use other operations, such as multiplication, suitably defined differences can be used in the same way.

In many ciphers, including SoDark, S-boxes are the only non-linear operation. This makes their resistance to differential cryptanalysis important. The properties of an S-box with respect to differential cryptanalysis are commonly described through its difference

Table 1: Summary of attacks on SoDark presented in this paper.

Rounds	Complexity Time	Complexity Data	Success probability	Setting
6	2^{47}	$2^{12.7}$	50%	Known plaintext
7	$2^{46.5}$	$2^{12.7}$	50%	Known plaintext
8	$2^{45.7}$	$2^{12.7}$	50%	Known plaintext
8	$2^{44.2}$	2^{8}	50%	Chosen ciphertext
8	$2^{45.2}$	2^{9}	100%	Chosen ciphertext

distribution table (DDT). An 8-bit S-box with 256 inputs and 256 outputs will have a DDT of dimension 256×256. The rows of the DDT represent input differences and the columns output differences. The value of each cell in the DDT represents the number of output pairs that will have a particular difference, given a particular input difference. For any input difference, there will be many output differences that are impossible, i.e. where the value of the corresponding cell in the DDT is zero. S-boxes that have not been designed to resist differential cryptanalysis may have certain input–output differences that are significantly more probable than others. The measure of this is the maximum cell value in the DDT and is known as the differential uniformity [Nyb93].

1.3 Results

The attacks presented in this paper break the full eight-round SoDark used in 2G ALE in practice. Table 1 summarizes the attacks presented in the following sections. The attacks have been implemented in software and verified experimentally. On a computer with several Nvidia graphics processing units (GPUs), both the known-plaintext and chosen-ciphertext attacks require about an hour to recover the key.

1.4 Structure of the paper

This paper is structured as follows: Section 2 describes the SoDark algorithm in the detail required for the following sections. Section 3 presents the cryptanalysis of SoDark. Section 4 describes a software implementation of the attacks presented in the previous section along with results of the experimental verification. Section 5 concludes the paper and considers some wider implications of the findings.

2 The SoDark Cipher Algorithm

2.1 Notation

All operations used by the SoDark cipher are byte oriented. The bytes of the cipher's state are represented by the capital letters $A, B, C \in \{0,1\}^8$. Input byte A to round r is denoted by $A^{(r-1)}$. The corresponding output byte from round r is $A^{(r)}$.

Concatenation of bytes into longer words is denoted by $\|$. In particular, the full plaintext is $\mathcal{P} = A^{(0)} \,\|\, B^{(0)} \,\|\, C^{(0)}$ and the full ciphertext after r rounds is $\mathcal{C} = A^{(r)} \,\|\, B^{(r)} \,\|\, C^{(r)}$. $\mathcal{K} \in \{0,1\}^{56}$ and $\mathcal{T} \in \{0,1\}^{64}$ are used for the full key and tweak (see Subsection 2.2), respectively. Individual bytes of $\mathcal{K}$ and $\mathcal{T}$ are denoted by k and t with subset indices where k_1 and t_1 refer to the most significant bytes. Partial round keys of round r are denoted by $a^{(r)}$, $b^{(r)}$, $c^{(r)} \in \{0,1\}^8$.

The cipher uses two operations: exclusive or (XOR) and single-byte S-box lookups. $\oplus$ is used as the XOR operator. An S-box lookup of the value in X is denoted by $S(X)$.

Table 2: The SoDark S-box. Adapted from [DoD17].

	$\star$0	$\star$1	$\star$2	$\star$3	$\star$4	$\star$5	$\star$6	$\star$7	$\star$8	$\star$9	$\star$A	$\star$B	$\star$C	$\star$D	$\star$E	$\star$F
0$\star$	9C	F2	14	C1	8E	CB	B2	65	97	7A	60	17	92	F9	78	41
1$\star$	07	4C	67	6D	66	4A	30	7D	53	9D	B5	BC	C3	CA	F1	04
2$\star$	03	EC	D0	38	B0	ED	AD	C4	DD	56	42	BD	A0	DE	1B	81
3$\star$	55	44	5A	E4	50	DC	43	63	09	5C	74	CF	0E	AB	1D	3D
4$\star$	6B	02	5D	28	E7	C6	EE	B4	D9	7C	19	3E	5E	6C	D6	6E
5$\star$	2A	13	A5	08	B9	2D	BB	A2	D4	96	39	E0	BA	D7	82	33
6$\star$	0D	5F	26	16	FE	22	AF	00	11	C8	9E	88	8B	A1	7B	87
7$\star$	27	E6	C7	94	D1	5B	9B	F0	9F	DB	E1	8D	D2	1F	6A	90
8$\star$	F4	18	91	59	01	B1	FC	34	3C	37	47	29	E2	64	69	24
9$\star$	0A	2F	73	71	A9	84	8C	A8	A3	3B	E3	E9	58	80	A7	D3
A$\star$	B7	C2	1C	95	1E	4D	4F	4E	FB	76	FD	99	C5	C9	E8	2E
B$\star$	8A	DF	F5	49	F3	6F	8F	E5	EB	F6	25	D5	31	C0	57	72
C$\star$	AA	46	68	0B	93	89	83	70	EF	A4	85	F8	0F	B3	AC	10
D$\star$	62	CC	61	40	F7	FA	52	7F	FF	32	45	20	79	CE	EA	BE
E$\star$	CD	15	21	23	D8	B6	0C	3F	54	1A	BF	98	48	3A	75	77
F$\star$	2B	AE	36	DA	7E	86	35	51	05	12	B8	A6	9A	2C	06	4B

The SoDark S-box is bijective and the inverse operation, used in decryption, is denoted by $S^{-1}(X)$. The S-box is provided in Table 2.

The cryptanalysis relies on the use of XOR differences and their development through the cipher (see Subsection 1.2). Subset letters a and b are used to differentiate between the same variable in two parallel encryptions, for example $A_a^{(1)}$ and $A_b^{(1)}$. To improve readability, differences are expressed using the shorthand $\Delta X = X_a \oplus X_b$.

2.2 Tweak

In addition to the 56-bit key $\mathcal{K}$, SoDark uses a 64-bit tweak $\mathcal{T}$. Its primary purpose appears to be prevention of replay attacks. The tweak format is briefly presented here with emphasis on the parts required for the following cryptanalysis. The reader is referred to the standard [DoD17] for a complete description. The standard and other publicly available descriptions of the cipher (e.g. [Joh92]) use the term *seed* in place of tweak. This paper uses the term *tweak* in accordance with current convention.

There is no mechanism in the ALE protocols for transferring a tweak. For this reason, the tweak must be generated from information known to both the sender and receiver at the time of transmission and reception. The information used is date and time of day, transmission frequency, selected protection interval (PI), and word number in the current PI. Table 3 shows how the tweak is formed using that information. The PI is a system setting which defines the format of the time fields in the tweak. When the PI is equal to one second, the coarse and fine time fields respectively correspond to the number of minutes since midnight and seconds in the current minute. For other PIs, fields increment differently, including modes where the fine time field is always set to all ones. Regardless of PI, GPS time is normally used. The word number field starts at zero and is incremented for each word encrypted in the current PI, except in some cases where the standard requires transmitting stations to reset the word counter to facilitate synchronization.

The information used to generate the tweak is available to any outside observer, with the exception of the PI system setting and thus the format used for the coarse and fine time fields. However, there are only a handful of formats that would have to be tried by

Table 3: Construction of the SoDark tweak from information known to both sender and receiver. The same bit numbering as in [DoD17] is used, starting with 1 at the most significant bit (MSB) of the first byte and ending with 64 at the least significant bit (LSB) of the last byte. (BCD=Binary Coded Decimal, PI=Protection Interval)

MSB	LSB	Content	Range
1	4	Month	1 – 12
5	9	Day	1 – 31
10	20	Coarse time	0 – 1439
21	26	Fine time	0 – 59
27	34	Word number in PI	0 – 153
35	36	Always zero	0
37	40	BCD frequency (100 MHz)	0 – 9
41	44	BCD frequency (10 MHz)	0 – 9
45	48	BCD frequency (1 MHz)	0 – 9
49	52	BCD frequency (100 kHz)	0 – 9
53	56	BCD frequency (10 kHz)	0 – 9
57	60	BCD frequency (1 kHz)	0 – 9
61	64	BCD frequency (100 Hz)	0 – 9

an attacker.

2.3 Round Function

When used in 2G ALE, the cipher has eight rounds. Before the first round, the 24-bit plaintext is split into three bytes $\mathcal{P} = A^{(0)} \,\|\, B^{(0)} \,\|\, C^{(0)}$. The round function performs the following operations to calculate $A^{(r)}$, $B^{(r)}$, and $C^{(r)}$.

$$a^{(r)} = k_{(3r-3) \bmod 7+1} \oplus t_{(3r-3) \bmod 8+1} \tag{1}$$

$$c^{(r)} = k_{(3r-2) \bmod 7+1} \oplus t_{(3r-2) \bmod 8+1} \tag{2}$$

$$b^{(r)} = k_{(3r-1) \bmod 7+1} \oplus t_{(3r-1) \bmod 8+1} \tag{3}$$

$$A^{(r)} = S\Big(A^{(r-1)} \oplus B^{(r-1)} \oplus a^{(r)}\Big)$$

$$C^{(r)} = S\Big(C^{(r-1)} \oplus B^{(r-1)} \oplus c^{(r)}\Big)$$

$$B^{(r)} = S\Big(B^{(r-1)} \oplus A^{(r)} \oplus C^{(r)} \oplus b^{(r)}\Big)$$

The first two rounds are illustrated in Figure 1. After the final round, the ciphertext is formed by concatenation of the three state bytes: $\mathcal{C} = A^{(r)} \,\|\, B^{(r)} \,\|\, C^{(r)}$.

As seen from Equations (1), (2), and (3), the SoDark key schedule is entirely linear, with bytes from the key and tweak used one by one in rotation. Each round uses only 24 bits of the 56-bit key and 24 bits of the 64-bit tweak.

3 Cryptanalysis

3.1 Key Schedule

Despite its simplicity, the SoDark key schedule precludes many standard attacks on block ciphers. The use of a tweak dependent on time and transmission frequency means that each word will be encrypted with a unique tweak. This effectively prevents time-memory

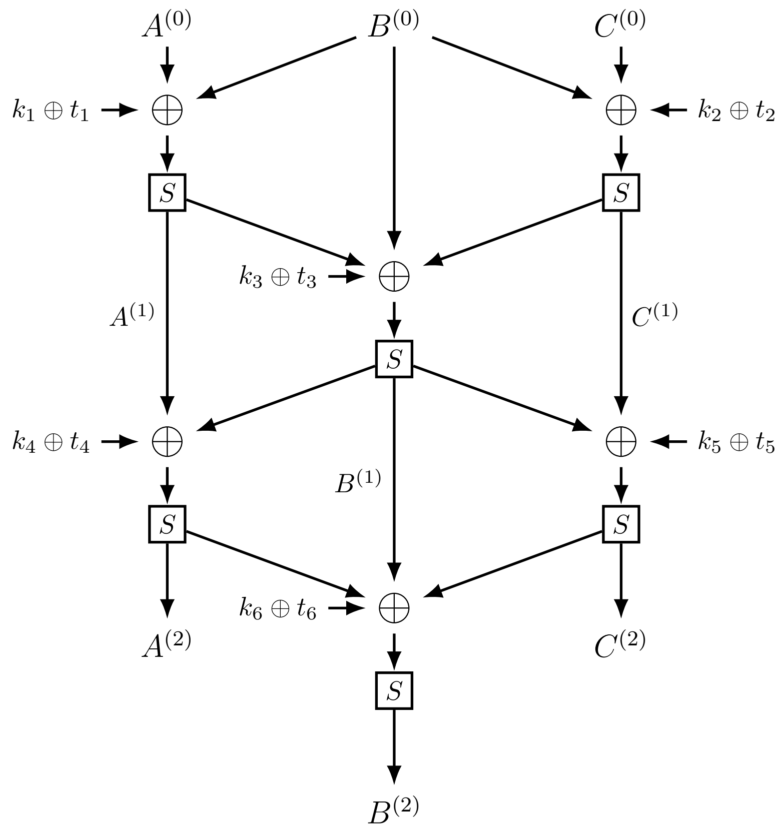

Figure 1: The first two rounds of the SoDark algorithm.

trade-off attacks [Hel80]. Considering its key and block size, the cipher would otherwise have been very vulnerable to such attacks.

Although it lacks round constants, the key schedule along with the use of unique tweaks still prevent slide attacks [BW99]. The numbers of key (7) and tweak (8) bytes are relatively prime, meaning that round keys don't repeat until after $7 \cdot 8 = 56$ rounds.

SoDark with five rounds or fewer is however vulnerable to meet-in-the-middle (MITM) attacks. In the five round case, the first two rounds do not use key byte 7 and the last two rounds do not use key byte 2. This can be leveraged to perform a three-subset MITM attack [BR11].

As pointed out by an anonymous reviewer, the symmetry of the cipher structure means that the 256 keys where all key bytes are identical are weak keys. The presence of a weak key can be tested for in just two operations by encrypting the plaintexts $\mathcal{P}_a = A^{(0)} \parallel B^{(0)} \parallel C^{(0)}$ and $\mathcal{P}_b = C^{(0)} \parallel B^{(0)} \parallel A^{(0)}$ using a tweak where all bytes are the same. For a weak key, this will result in the ciphertexts $\mathcal{C}_a = A^{(r)} \parallel B^{(r)} \parallel C^{(r)}$ and $\mathcal{C}_b = C^{(r)} \parallel B^{(r)} \parallel A^{(r)}$ for any number of rounds r.

3.2 The SoDark S-box

No information on the criteria used in the selection of the S-box has been found. The methods described by Biryukov and Perrin in [BP15] were applied in an attempt to reverse engineer any design criteria, but did not yield any positive results.

To investigate whether the S-box is a randomly generated permutation, its linear approximation table (LAT) and DDT were compared to those of randomly generated permutations of $\mathbb{GF}(2^8)$. In [DR07], Daemen and Rijmen provide a formula for the probability distribution of the LAT of a random Boolean permutation, originally from [O'C95]:

$$P\left[c_{i,j} = 2x\right] = \frac{\binom{2^{n-1}}{2^{n-2}+x}^2}{\binom{2^n}{2^{n-1}}}.$$

Here, $P\left[c_{i,j} = 2x\right]$ is the probability that a particular combination of input and output bits in an n-bit S-box will have the bias $2x$.

A two-sided two-sample Kolmogorov–Smirnov test [Kol08] was performed to compare the linear bias and differential uniformity of the S-box with the expected values in random permutations. With $p > 0.999999$ in both cases, there is no evidence to refute the hypothesis that the S-box is a randomly selected permutation. In particular, the S-box shows no signs of having been chosen with resistance against linear or differential cryptanalysis as a goal.

It should be noted however, that the S-box actually has linear and differential properties that are slightly worse than expected from the average random permutation. Its maximum linear bias is $\frac{38}{256}$ whereas the expected in a random permutation is $\frac{34}{256}$. Using the method from [BP15], the probability of a random permutation having linear bias of more than or equal to $\frac{38}{256}$ was calculated to be 0.057, meaning that the higher than expected linear bias is not statistically significant. Likewise, the S-box's differential uniformity is 14, higher than the expected value of 12.

3.3 Complexity Model

The attacks presented here are primarily evaluated based on their time complexities. A brute force attack is expected to require, on average, 2^{55} encryption operations to find the correct 56-bit key. The speed of SoDark is bounded by the number of S-box lookups and the complexity of other operations involved is negligible in comparison. Practical experience has shown this to be true even in bitsliced software implementations (see Section 4) and the same is expected for hardware implementations. For those reasons, the number of

S-box operations required to test a single key in a brute force known-plaintext attack is used as the benchmark when estimating the time complexity of the attacks presented here.

A single encryption in an r-round version of SoDark involves $3r$ S-box lookups. Two obvious optimizations are possible. First, the S-box lookups in the last round provide no security and can be eliminated by performing inverse lookups on the ciphertext beforehand. This reduces the number of S-box lookups per key to $3 \cdot (r-1)$. Second, the first round uses only key bytes k_1, k_2, k_3 and the second round only key bytes k_4, k_5, k_6. This means that the intermediate states after the first and second round can be cached and reused for most key attempts. A single byte in the third round can also be cached. Together, these optimizations reduce the average number of S-box lookups per key in a brute force attack to

$$3 \cdot (r-3) + 3 \cdot \frac{2^{24}}{2^{56}} + 4 \cdot \frac{2^{48}}{2^{56}} - 1 \approx 3 \cdot (r-3) - 1.$$

This number of S-box lookups is used to assess the time complexity of the attacks.

Since the cipher has a block size of 24 bits and a 56-bit key, a brute force attack is expected to generate about 2^{32} false positives which must be checked against other plaintext–ciphertext–tweak tuples. The increase in complexity due to this check has been considered insignificant.

For data complexity, the metric used is simply the number of plaintext–ciphertext–tweak tuples required for a 50% probability of success.

3.4 Known-plaintext Attacks on Six and Seven Rounds

As a prelude to the attack on the full eight round cipher as used in 2G ALE [DoD17], this section presents a known-plaintext attack on six- and seven-round versions of the cipher. The basis for the attack is that, in cases where tweaks differ only by a single byte, it is possible to calculate differences of internal states between two encryptions using only the ciphertext and tweak. As evident from the description in Subsection 2.2, tweaks with the single byte difference used here are present in practically every transmitted ALE frame.

For any round r of SoDark and any pairs of plaintext, ciphertext, and tweak, the differences after the previous round, $\Delta A^{(r-1)}$, $\Delta B^{(r-1)}$, and $\Delta C^{(r-1)}$, can be calculated without any key guessing:

$$\Delta B^{(r-1)} = \Delta S^{-1}\left(B^{(r)}\right) \oplus \Delta A^{(r)} \oplus \Delta C^{(r)} \oplus \Delta t_{(3r+7) \bmod 8+1} \tag{4}$$

$$\Delta C^{(r-1)} = \Delta S^{-1}\left(C^{(r)}\right) \oplus \Delta B^{(r-1)} \oplus \Delta t_{(3r+6) \bmod 8+1} \tag{5}$$

$$\Delta A^{(r-1)} = \Delta S^{-1}\left(A^{(r)}\right) \oplus \Delta B^{(r-1)} \oplus \Delta t_{(3r+5) \bmod 8+1}. \tag{6}$$

For the six-round attack, consider a case with two plaintext–ciphertext–tweak tuples where $\mathcal{C}_a = \mathcal{C}_b$ and all bytes in the tweak are identical except for t_5. Thus, $\Delta A^{(6)} = \Delta B^{(6)} = \Delta C^{(6)} = 0$, $\Delta t_5 \neq 0$ and $\Delta t_x = 0$ for all $x \in \{1, 2, 3, 4, 6, 7, 8\}$. Using the fact that $\Delta S^{-1}(X) = 0$ when $\Delta X = 0$, it becomes possible to calculate a part of the difference after the third round using only the ciphertexts and tweaks:

$$\begin{aligned}
\Delta B^{(5)} &= \Delta S^{-1}\left(B^{(6)}\right) \oplus \Delta A^{(6)} \oplus \Delta C^{(6)} \oplus \Delta t_2 = 0\\
\Delta C^{(5)} &= \Delta S^{-1}\left(C^{(6)}\right) \oplus \Delta B^{(5)} \oplus \Delta t_1 = 0\\
\Delta A^{(5)} &= \Delta S^{-1}\left(A^{(6)}\right) \oplus \Delta B^{(5)} \oplus \Delta t_8 = 0\\
\Delta B^{(4)} &= \Delta S^{-1}\left(B^{(5)}\right) \oplus \Delta A^{(5)} \oplus \Delta C^{(5)} \oplus \Delta t_7 = 0\\
\Delta C^{(4)} &= \Delta S^{-1}\left(C^{(5)}\right) \oplus \Delta B^{(4)} \oplus \Delta t_6 = 0\\
\Delta A^{(4)} &= \Delta S^{-1}\left(A^{(5)}\right) \oplus \Delta B^{(4)} \oplus \Delta t_5 = \Delta t_5\\
\Delta B^{(3)} &= \Delta S^{-1}\left(B^{(4)}\right) \oplus \Delta A^{(4)} \oplus \Delta C^{(4)} \oplus \Delta t_4 = \Delta t_5\\
\Delta C^{(3)} &= \Delta S^{-1}\left(C^{(4)}\right) \oplus \Delta B^{(3)} \oplus \Delta t_3 = \Delta t_5.
\end{aligned}$$

Knowing that $\Delta C^{(3)} = \Delta t_5$, matching against this value is possible using the corresponding known plaintexts by iterating over six of the seven key bytes:

$$\begin{aligned}
A^{(1)} &= S\left(A^{(0)} \oplus B^{(0)} \oplus k_1 \oplus t_1\right)\\
C^{(1)} &= S\left(C^{(0)} \oplus B^{(0)} \oplus k_2 \oplus t_2\right)\\
B^{(1)} &= S\left(B^{(0)} \oplus A^{(1)} \oplus C^{(1)} \oplus k_3 \oplus t_3\right)\\
A^{(2)} &= S\left(A^{(1)} \oplus B^{(1)} \oplus k_4 \oplus t_4\right)\\
C^{(2)} &= S\left(C^{(1)} \oplus B^{(1)} \oplus k_5 \oplus t_5\right)\\
B^{(2)} &= S\left(B^{(1)} \oplus A^{(2)} \oplus C^{(2)} \oplus k_6 \oplus t_6\right)\\
C^{(3)} &= S\left(C^{(2)} \oplus B^{(2)} \oplus k_1 \oplus t_8\right)\\
\Delta C^{(3)} &= C_a^{(3)} \oplus C_b^{(3)}.
\end{aligned}$$

This is illustrated in Figure 2.

Assuming the cipher's randomization properties are good (this is investigated in [Joh92]), all differences after the fourth round have probability 2^{-24}. The number of plaintext–ciphertext–tweak tuple pairs required for one of them to have the output difference required for the attacks with 50% probability is

$$\frac{-\log 2}{\log\left(2^{24}-1\right) - \log\left(2^{24}\right)} \approx 2^{23.5}.$$

When creating the plaintext–ciphertext–tweak tuple pairs, the tweaks in each pair must be identical in seven of the eight bytes. The most efficient way to satisfy this in an oracle model is to generate tuples for two different tweaks with $\Delta t_5 \neq 0$ and $\Delta t_x = 0$ for all $x \in \{1, 2, 3, 4, 6, 7, 8\}$. This way, with n generated tuples per tweak, n^2 tuple-pairs can be formed. Thus, only $\sqrt{2^{23.5}} \approx 2^{11.7}$ tuples are required for each of the two tweaks in order to find the required output difference with 50% probability.

In a more realistic setting where an adversary is limited to passively intercepting ALE frames sent between legitimate users, all words will be sent with a unique tweak except in rare circumstances. All 24-bit words that constitute a particular frame will, however, have tweaks that differ only in the fifth byte (see Table 3). In ALE networks where relatively

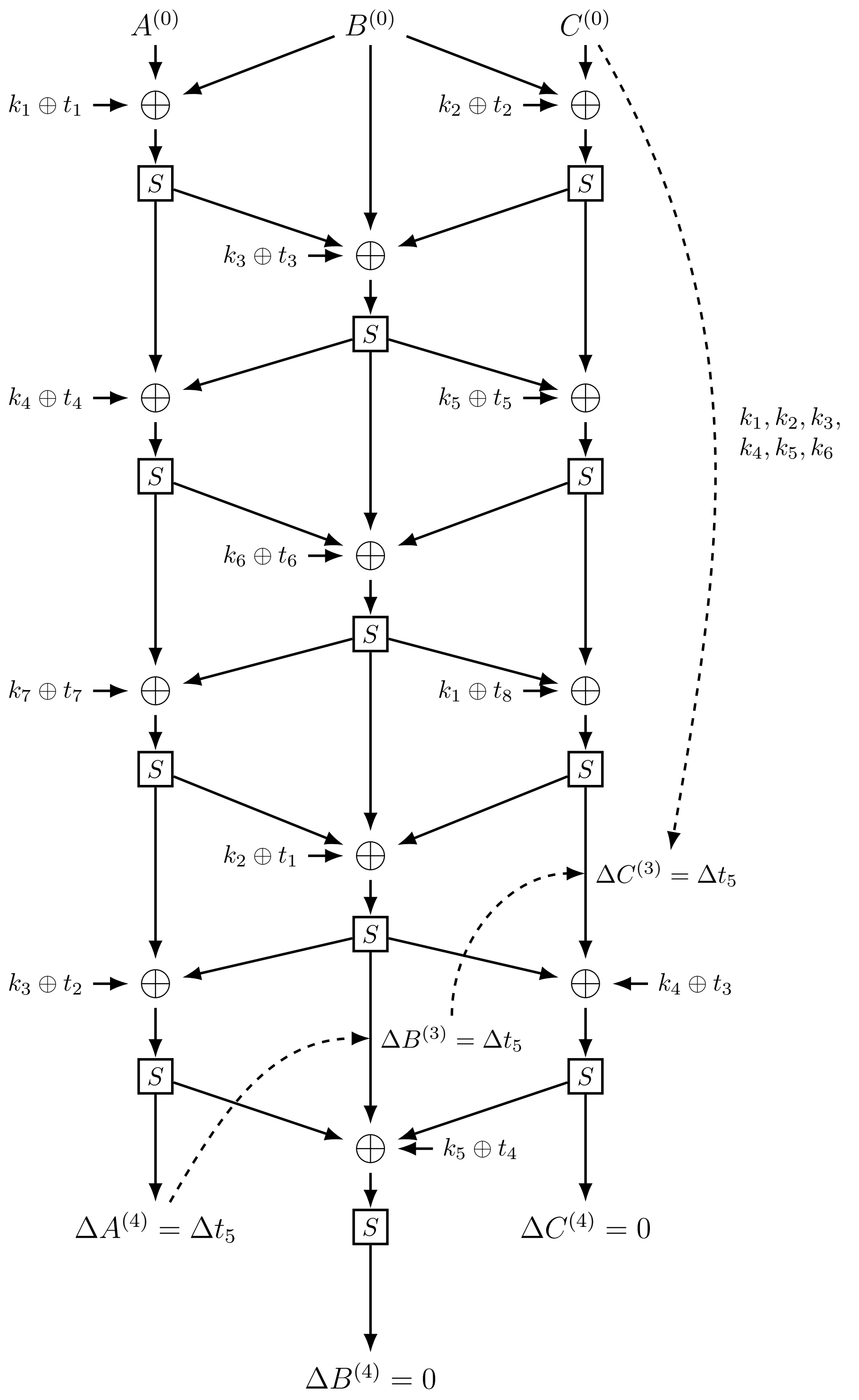

Figure 2: The first four rounds in the attacks on SoDark.

long protection intervals are used, this may also be true for words in different frames. As the simplest frame in a call consists of three 24-bit words (see Subsection 1.1), three-word frames are assumed in the calculation of the data complexity for a realistic setting. This means that each intercepted frame will yield three word pairs with the required tweak difference, each with a 2^{-24} probability of having the right difference after round six. Each frame thus has probability

$$1 - \left(1 - 2^{-24}\right)^3 \approx 2^{-22.4}$$

of having the sought difference. This means that

$$\frac{-\log 2}{3 \log\left(1 - 2^{-24}\right)} \approx 2^{21.9}$$

frames are required for 50% probability of success. Assuming that any completed call has involved at least four transmitted frames, this is equivalent to slightly less than a million completed 2G ALE calls.

The dominant source of time complexity in the attack is the iteration over six key bytes to calculate $\Delta C^{(3)}$. Like in the brute force case, the number of S-box lookups can be decreased through caching. With this optimization, the time complexity of the attack becomes

$$\frac{2 \cdot 2^{8} + 2 \cdot 2^{16} + 2 \cdot 2^{24} + 2 \cdot 2^{32} + 2 \cdot 2^{40} + 4 \cdot 2^{48}}{8} \approx 2^{47}.$$

The attack can be extended to seven rounds by observing that, with the same conditions as the attack on six rounds, $\Delta A^{(7)} = \Delta C^{(7)} = 0$ and $\Delta S^{-1}\left(B^{(7)}\right) = \Delta t_5$. After a ciphertext pair with this difference is found, the attack continues as in the six-round case.

In calculating the time complexity of the attack on seven rounds, the S-box lookups in the filtering step are included:

$$\frac{2 \cdot 2^{23.5} + 2 \cdot 2^{8} + 2 \cdot 2^{16} + 2 \cdot 2^{24} + 2 \cdot 2^{32} + 2 \cdot 2^{40} + 4 \cdot 2^{48}}{11} \approx 2^{46.5}.$$

3.5 Known-plaintext Attacks on Eight Rounds

In the full eight-round version of SoDark used in 2G ALE, it is not possible to immediately identify the required difference after round six using just the ciphertext and tweak. However, through a fast filtering operation on the ciphertext, it is possible to reject all pairs with a zero probability of having the correct difference after the sixth round. For the pairs remaining after the filtering step, all candidate values for k_3 are also identified. For this reason, the eight-round attack has lower time complexity than the previously described attack on six and seven rounds while retaining the same data complexity.

In the first filtering step, Equations (4) – (6) are used to remove candidate pairs not satisfying the requirement that

$$\Delta A^{(7)} = \Delta C^{(7)} = 0. \tag{7}$$

The second filtering step continues by guessing k_3 to check if $\Delta B^{(6)} = 0$:

$$\begin{aligned} B_a^{(7)} &= S^{-1}\left(B_a^{(8)}\right) \oplus A_a^{(8)} \oplus C_a^{(8)} \oplus t_8 \oplus k_3 \\ B_b^{(7)} &= S^{-1}\left(B_b^{(8)}\right) \oplus A_b^{(8)} \oplus C_b^{(8)} \oplus t_8 \oplus k_3 \\ \Delta B^{(6)} &= \Delta S^{-1}\left(B^{(7)}\right) \oplus \Delta A^{(7)} \oplus \Delta C^{(7)} \oplus \Delta t_5. \end{aligned}$$

Figure 3 illustrates the filtering steps in the last two rounds of the algorithm.

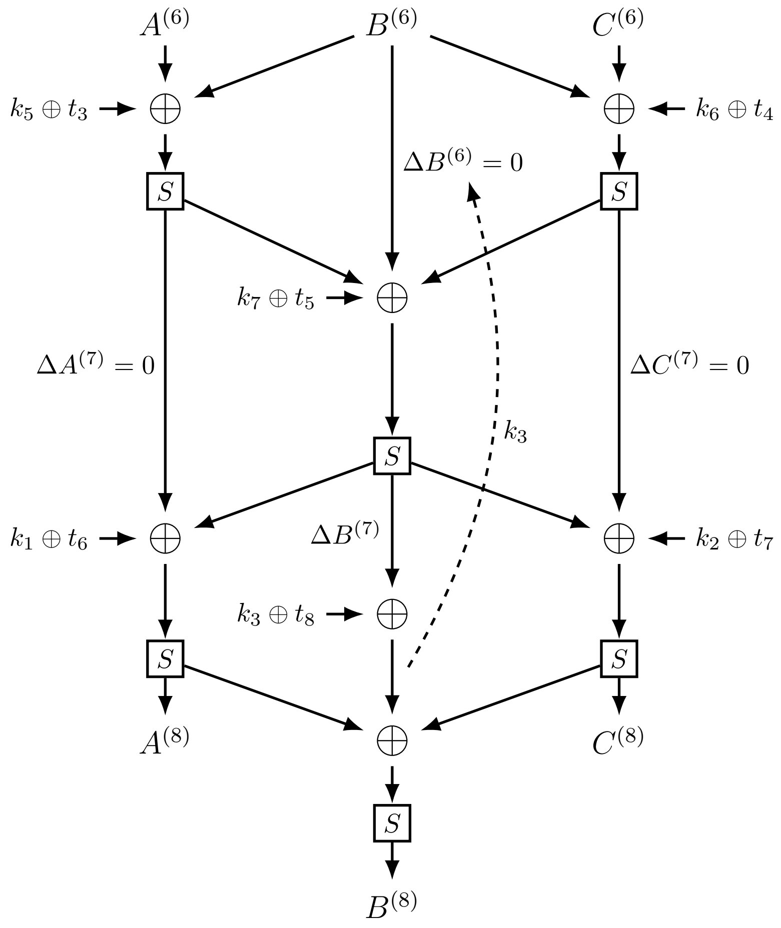

Figure 3: The last two rounds in the attack on eight-round SoDark.

For any values of k_3 where $\Delta B^{(6)} = 0$, the attack proceeds to match $\Delta C^{(3)} = \Delta t_5$ using known plaintexts and iterating over key bytes k_1, k_2, k_4, k_5, and k_6 as in the attacks on six and seven rounds (see Figure 2).

For 50% probability of success, the first filtering step is expected to require $6 \cdot 2^{23.5}$ S-box lookups. The probability that a random plaintext–ciphertext–tweak tuple will satisfy Equation (7) is 2^{-16}, meaning that about $2^{7.5}$ tuples remain for the second filtering step which uses $4 \cdot 2^8$ S-box lookups per tuple. With the SoDark S-box, the probability for a tuple to survive the second filtering step is $\frac{100}{256}$, resulting in $\frac{100}{256} \cdot 2^{7.5} \approx 2^{6.1}$ remaining tuples. The total time complexity for the attack on eight-round SoDark is therefore

$$\frac{1}{14} \cdot \Bigg(6 \cdot 2^{23.5} + 4 \cdot 2^{7.5} \cdot 2^8 + \\ 2^{6.1} \cdot \left(2 \cdot 2^8 + 2 \cdot 2^{16} + \frac{2.6}{2^8} \cdot \left(2 \cdot 2^{24} + 2 \cdot 2^{32} + 2 \cdot 2^{40} + 4 \cdot 2^{48}\right) \right) \Bigg) \approx 2^{45.7}.$$

As already mentioned, this is lower than the preceding attack on six and seven rounds. Naturally, by using a difference in another tweak byte, this attack can be adopted to fewer rounds. In the case of seven rounds, a difference in t_2 would be used instead of t_5.

3.6 Chosen-ciphertext Attacks

The previous known-plaintext attacks can be improved significantly in a chosen-ciphertext setting. From Equations (4) – (6) and Figure 3, it can be observed that it is easy to generate ciphertext pairs that satisfy $\Delta A^{(7)} = \Delta C^{(7)} = 0$ with probability 1, since they all satisfy

$$\Delta S^{-1}\left(A^{(8)}\right) = \Delta S^{-1}\left(C^{(8)}\right) = \Delta B^{(7)}$$

with $\Delta B^{(7)} \neq 0$ and $\Delta t_6 = \Delta t_7 = 0$. Furthermore, since the value of $\Delta B^{(6)}$ is only dependent on a single unknown key byte, a single chosen-ciphertext pair can be generated for each candidate value of k_3. This means that 2^7 chosen plaintext–ciphertext–tweak tuples are enough for 50% probability of success and 2^8 for 100%.

Generation of the chosen plaintext–ciphertext–tweak tuples is rather straightforward. First, the attacker picks any two tweaks with the differences $\Delta t_5 \neq 0$ and $\Delta t_x = 0$ for all $x \in \{1, 2, 3, 4, 6, 7, 8\}$. The attacker continues by picking the values of $S^{-1}\left(B_a^{(7)}\right)$ and $S^{-1}\left(B_b^{(7)}\right)$ such that

$$\Delta S^{-1}\left(B^{(7)}\right) = t_5.$$

The corresponding values for $B_a^{(7)}$, $B_b^{(7)}$, and $\Delta B^{(7)}$ can now be calculated. Then, the values for $S^{-1}\left(A_a^{(8)}\right)$, $S^{-1}\left(A_b^{(8)}\right)$, $S^{-1}\left(C_a^{(8)}\right)$, and $S^{-1}\left(C_b^{(8)}\right)$ are selected such that

$$\Delta S^{-1}\left(A^{(8)}\right) = \Delta S^{-1}\left(C^{(8)}\right) = \Delta B^{(7)}.$$

Finally, $A_a^{(8)}$, $A_b^{(8)}$, $C_a^{(8)}$, and $C_b^{(8)}$ can be calculated directly from those values and are used to calculate up to 256 ciphertext pairs of the form

$$\mathcal{C}_a = A_a^{(8)} \parallel B_a^{(8)} \parallel C_a^{(8)}$$
$$\mathcal{C}_b = A_b^{(8)} \parallel B_b^{(8)} \parallel C_b^{(8)}$$

Table 4: Summary of implemented attacks and the times required to perform attacks with data for 50% success probability.

Rounds	Implementation	Data	Time
8	GPU brute force	2	5 days
8	CPU known plaintext	$2^{12.7}$	9 days
8	GPU known plaintext	$2^{12.7}$	42 minutes
8	CPU chosen ciphertext	2^8	9 days
8	GPU chosen ciphertext	2^8	42 minutes

where

$$B_a^{(8)} = S\left(B_a^{(7)} \oplus A_a^{(8)} \oplus C_a^{(8)} \oplus k_3 \oplus t_8\right)$$
$$B_b^{(8)} = S\left(B_b^{(7)} \oplus A_b^{(8)} \oplus C_b^{(8)} \oplus k_3 \oplus t_8\right)$$

by iterating over k_3. In an oracle model, the generated ciphertexts and tweaks can then be used to obtain the corresponding plaintexts.

With the elimination of the filtering steps, the time complexity of the chosen-ciphertext attack becomes

$$\frac{2^7 \cdot \left(2 \cdot 2^8 + 2 \cdot 2^{16} + 2 \cdot 2^{24} + 2 \cdot 2^{32} + 2 \cdot 2^{40}\right)}{14} \approx 2^{44.2}$$

in the 50% probability of success case and twice that for 100% probability of success.

4 Software Implementation and Experimental Verification

For experimental verification, the attacks described in the previous section were implemented in C and the Nvidia CUDA parallel computing framework for general-purpose GPUs [Nvi21]. Efficient attacks on two, three, four, and five rounds have also been implemented. For up to five rounds, the attacks can be performed on any pair of plaintext–ciphertext–tweak tuples with 100% probability of success. Additionally, a utility program for SoDark encryption and decryption has been developed. The utility can also be used for generation of random testing data. The implementation is available on Github under an open source licence.[1]

Kwan's algorithm [Kwa00] was used to generate a boolean gate representation of the SoDark S-box which is, by far, the most time-consuming operation in the cipher. This representation was used to develop a bitsliced [Bih97] CUDA implementation. On a computer with six Nvidia GPUs, the implementation is able to recover the key in about an hour when a sufficient number of plaintext–ciphertext–tweak tuples are available. The reference brute force implementation searches through the entire key space in about five days on the same computer. Table 4 summarizes the performance of the implemented attacks. The CUDA implementation of the attacks on eight rounds presented in this paper are about 170 times ($2^{7.4}$) faster than the bitsliced brute force implementation.

5 Conclusions and Future Developments

Both data and time complexities of the attacks on eight-round SoDark presented in this paper are such that they are feasible in practice, as demonstrated in the previous

[1] https://github.com/dansarie/SoCracked

section. This means that the linking protection facility in 2G ALE fails in its stated goals. Furthermore, trust in 3G ALE linking protection, which uses the same algorithm but with 16 rounds, is weakened.

While confidentiality is a stated goal, the main purpose of the SoDark cipher in the context of ALE linking protection is authentication, i.e. prevention of unauthorized access to HF radio networks [JKF+12]. Flaws in the cipher can directly affect the system at large by allowing unauthorized users to spoof legitimate traffic or inject data into connected systems. Security flaws in digital communication systems, like the one described here, also open up for new methods of attacks on availability. HF radio systems are normally resource constrained. In addition to low bandwidth, a radio station can usually only handle one or a few calls at a time. This can make the consequences of unauthorized access particularly severe. The traditional way of attacking availability in radio systems is jamming, where the attacker attempts to overpower a legitimate transmitter's signal. The jammer's probability of success increases with output power, antenna gain, and—most importantly—proximity to the receiver. By transmitting spoofed linking requests and thereby depleting a finite resource—the number of concurrent calls a radio station can maintain at a time—an attacker with knowledge of the linking protection key can achieve the same goals as with jamming, but without needing a power or proximity advantage over the legitimate users. Similarly, by transmitting spoofed disconnection frames, the attacker can disrupt calls in progress. Since group calls are possible in ALE, a single station can call many or all stations in a network. Together with the fact that HF radio has global reach, this means that an attacker can potentially deplete the resources of an entire HF radio network and prevent all legitimate traffic using only a few spoofed frames.

In addition to the 24-bit version of SoDark presented in this paper, a 48-bit version of the algorithm was introduced with 3G ALE. The 48-bit version retains the same key size and key schedule. The round function is slightly modified compared to the 24-bit version, presumably in order to improve mixing. The work presented here could prove a good start for attacks on the sixteen-round 24- and 48-bit versions of SoDark that are used in 3G ALE.

A new linking protection cipher, HALFLOOP, was introduced in MIL-STD-188-141D, which was released in December 2017. HALFLOOP was developed together with the fourth generation ALE standard and in response to worries about the security of SoDark, particularly when it comes to the short key length. HALFLOOP is based on the same basic operations as the Advanced Encryption Standard (AES) [AES01], but has significant differences to AES in both the round function and key schedule. It supports 24-, 48-, and 96-bit block sizes and 128- and 256-bit key lengths. Although HALFLOOP is intended to replace SoDark for linking protection in 2G and 3G ALE networks, it is not known if it is in practical use anywhere. This apparent low adoption is likely due to the fact that using HALFLOOP would cause incompatibility with older radio equipment [Joh16, DoD17].

It is unfortunate that HALFLOOP, like SoDark, was developed and standardized without any public review. The story is very similar to that of other ciphers developed for wireless communication standards. For example, both the DECT Standard Cipher for cordless telephones [NTW10] and the A5/1 cipher for GSM mobile telephony [BSW01] have been shown to be insecure. They are just two of a number of specialized or proprietary ciphers for radio-based systems that share similar fates [Ver15]. When SoDark was developed for 2G ALE, there were likely not many suitable block ciphers or modes available. In light of this, the development of an application-specific cipher is understandable. The same can not be said for HALFLOOP, which is a recent development. A mode such as the Thorp shuffle [MRS09], together with for example AES, would likely be a much better candidate for replacing SoDark.

The reoccurring phenomenon of inventing new ciphers for wireless communication standards evokes both questions about the underlying reasons and conjectures about "Not

Invented Here"-type phenomena [KA82]. Non-technical research into the organizational causes could prove important for increasing security in future standards.

Acknowledgments

A large part of the work presented in this article was performed during the author's time as a student at the Naval Postgraduate School. The author would especially like to thank Dr. David Canright for his help and comments. The author also wishes to thank the anonymous reviewers, whose comments and insights helped improve this paper.

References

[AES01] Advanced encryption standard (AES). Federal Information Processing Standards Publication 197, National Institute of Standards and Technology, Gaithersburg, MD, 2001. doi:10.6028/NIST.FIPS.197.

[ALE21] List of automatic link establishment agencies and frequencies, 2021. http://www.ominous-valve.com/ale-list.txt.

[Bih97] Eli Biham. A fast new DES implementation in software. In Eli Biham, editor, *Fast Software Encryption – FSE 1997*, volume 1267 of *Lecture Notes in Computer Science*, pages 260–272. Springer, Berlin, Heidelberg, 1997. doi:10.1007/BFb0052352.

[Bih06] Eli Biham. How to make a difference – early history of differential cryptanalysis, 2006. Invited talk at FSE 2006.

[BP15] Alex Biryukov and Léo Perrin. On reverse-engineering S-boxes with hidden design criteria or structure. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, volume 9215 of *Lecture Notes in Computer Science*, pages 116–140. Springer, Berlin, Heidelberg, 2015. doi:10.1007/978-3-662-47989-6_6.

[BR11] Andrey Bogdanov and Christian Rechberger. A 3-subset meet-in-the-middle attack: Cryptanalysis of the lightweight block cipher KTANTAN. In *Selected Areas in Cryptography – SAC 2010*, volume 6544 of *Lecture Notes in Computer Science*, pages 229–240. Springer, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-19574-7_16.

[BS93] Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer, New York, 1993. doi:10.1007/978-1-4613-9314-6.

[BSW01] Alex Biryukov, Adi Shamir, and David Wagner. Real time cryptanalysis of A5/1 on a PC. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Bruce Schneier, editors, *Fast Software Encryption – FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Berlin, Heidelberg, 2001. doi:10.1007/3-540-44706-7_1.

[BW99] Alex Biryukov and David Wagner. Slide attacks. In Lars Knudsen, editor, *Fast Software Encryption – FSE 1999*, volume 1636 of *Lecture Notes in Computer Science*, pages 245–259. Springer, Berlin, Heidelberg, 1999. doi:10.1007/3-540-48519-8_18.

[DoD17] Interoperability and performance standards for medium and high frequency radio systems. United States Department of Defense Interface Standard MIL-STD-188-141D, 2017.

[DR07] Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. *Journal of Mathematical Cryptology*, 1(3), 2007. doi:10.1515/JMC.2007.011.

[Hel80] Martin Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26(4):401–406, 1980. doi:10.1109/TIT.1980.1056220.

[Int02] International Telecommunication Union. *Frequency-adaptive communication systems and networks in the MF/HF bands.* 2002. https://hdl.handle.net/11.1002/pub/800c949a-en.

[JKF$^+$12] Eric E Johnson, Erik Koski, William N Furman, Mark Jorgenson, and John Nieto. *Third-generation and Wideband HF Radio Communications.* Artech House, Norwood, MA, 2012.

[Joh92] Eric E. Johnson. A 24-bit encryption algorithm for linking protection. Technical Report ASQB-OSO-S-TR-92-04, USAISEC, March 1992.

[Joh16] Eric E Johnson. Wideband ALE – the next generation of HF. In *2016 Nordic HF Radio Conference HF âĂŸ16*, Fårö, Sweden, August 2016.

[KA82] Ralph Katz and Thomas J. Allen. Investigating the not invented here (NIH) syndrome: A look at the performance, tenure, and communication patterns of 50 R&D project groups. *R&D Management*, 12(1):7–20, January 1982. doi:10.1111/j.1467-9310.1982.tb00478.x.

[Kol08] Kolmogorov–Smirnov test. In *The Concise Encyclopedia of Statistics*, pages 283–287. Springer, New York, 2008. doi:10.1007/978-0-387-32833-1_214.

[Kwa00] Matthew Kwan. Reducing the gate count of bitslice DES. Cryptology ePrint Archive, Report 2000/051, 2000. https://eprint.iacr.org/2000/051.

[MRS09] Ben Morris, Phillip Rogaway, and Till Stegers. How to encipher messages on a small domain. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 286–302. Springer, Berlin, Heidelberg, 2009. doi:10.1007/978-3-642-03356-8_17.

[NTW10] Karsten Nohl, Erik Tews, and Ralf-Philipp Weinmann. Cryptanalysis of the DECT Standard Cipher. In Seokhie Hong and Tetsu Iwata, editors, *Fast Software Encryption – FSE 2010*, volume 6147 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Berlin, Heidelberg, 2010. doi:10.1007/978-3-642-13858-4_1.

[Nvi21] Nvidia Corporation. *CUDA C++ programming guide*, 2021. https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf.

[Nyb93] Kaisa Nyberg. Differentially uniform mappings for cryptography. In Tor Helleseth, editor, *Advances in Cryptology – EUROCRYPT 1993*, volume 765 of *Lecture Notes in Computer Science*, pages 55–64. Springer, Berlin, Heidelberg, 1993. doi:10.1007/10.1007/3-540-48285-7_6.

[O'C95] Luke O'Connor. Properties of linear approximation tables. In Bart Preneel, editor, *Fast Software Encryption – FSE 1994*, volume 1008 of *Lecture Notes in Computer Science*, pages 131–136. Springer, Berlin, Heidelberg, 1995. doi:10.1007/3-540-60590-8_10.

[Sig21] Signal Identification Wiki – automatic link establishment (2G ALE), 2021. https://www.sigidwiki.com/wiki/Automatic_Link_Establishment_(2G_ALE).

[Ver15] Roel Verdult. *The (in)security of proprietary cryptography.* PhD thesis, Radboud Universiteit Nijmegen, 2015. https://hdl.handle.net/2066/140089.