Automated security proof of SQUARE, LED and CLEFIA using the MILP technique

Gabriel Cardoso de Carvalho[1], **Tertuliano Souza Neto**[2], **Thiago do Rêgo Sousa**[2]

[1]Instituto de Computação - Universidade Federal Fluminense (UFF) - Niterói - RJ

[2]CEPESC, Agência Brasileira de Inteligência, Brasília - DF

Abstract. *Provable security in cryptography is extremely relevant nowadays, since it is regarded as the basis for the proposal of new ciphers. In that sense, the designers of new ciphers have to find ways to prove that the proposed cipher is secure against the most pertinent forms of attack. Being safe against differential and linear cryptanalysis is still considered the bare minimum standard for any new cipher. In the last decade, a great deal of attention has been given to automated ways of proving the security of ciphers against both forms of attacks, the original one being generating mixed linear integer programs that model the given cipher in such a way that, by solving it, we are able to know the minimum number of rounds necessary for the cipher to be secure. In this paper, we apply this technique in the well known block ciphers LED, SQUARE and CLEFIA, and compare the results with the original security claims.*

1. Introduction

Symmetric cryptography is a very active research field since it is one of the basic building blocks for secure communication. In particular, many new block ciphers have been developed, the most famous probably being data encryption standard (DES) and advanced encryption standard (AES) algorithms. Their design usually takes into account recent developments of new attacks, and the designers try to build ciphers which are resistant to particular classes of attack. Resistance against differential and linear cryptanalysis was and still is one of the main goals of new cipher designs, but proving such things might be a bit cumbersome for the designer.

In the last decade a couple of automated ways of proving under which conditions a cipher is resistant to these attacks was developed. Along with the Boolean Satisfiability Problem [Mironov and Zhang 2006], mixed linear integer programming (MILP) is one very important tool as an alternative approach to analyze the security bounds of a cipher [Mouha et al. 2011]. In particular, the MILP method is used to prove security against both differential and linear cryptanalysis by means of solving a linear mixed integer program which is directly connected to the working mechanism of the cipher. Since its introduction, MILP has been used to prove the security of many ciphers, including SIMON, PRESENT, LBlock [Sun et al. 2014], LIZARD [Karthika and Singh 2023], and Midori64 [Zhao et al. 2020].

Besides its use as a means of providing provable security, MILP based methods have also been used in a variety of applications in modern cryptanalysis, such as searching for integral distinguishers [Xiang et al. 2016] and looking for differential and linear trails [Fu et al. 2016]. Moreover, much work has been done in improving the speed of the MILP model [Zhou et al. 2019] as well as using it in cryptography design [Pal et al. 2023].

Therefore, since its introduction in 2011, the use of MILP models in the area of cryptology has been relevant, which in turn justifies our choice of topic. However, applying the MILP modeling from [Mouha et al. 2011] requires the underlying cipher to have some properties. First of all, the cipher needs to be word-based (nibble, byte) and its internal mechanisms being based on S-boxes, XOR operations, linear permutation and/or three-forked branches. This is the case of AES and Enocoro, which were analyzed in the original paper of [Mouha et al. 2011].

SQUARE [Daemen et al. 1997] is one classic cipher that is still relevant nowadays for being the precursor of the AES, as well as having introduced the SQUARE attack - a type of cryptanalysis directed to AES-like ciphers, such as KIASU-BC [Dobraunig et al. 2016] and Midori64 [Wardhana and Indarjani 2019]. The LED cipher [Guo et al. 2011] is well known for being one of the first lightweight ciphers (Light Encryption Device), which are commonly used in Internet of Things and other embedded systems, and for being target of extensive cryptanalysis since its creation although remaining secure. CLEFIA [Katagi and Moriai 2011] is also a lightweight cipher. In fact, it is one of the standardized lightweight encryption algorithms of ISO/IEC 29192-2:2019.

Due to their relevancy, these three block ciphers will be the subject of this paper. We will apply the MILP modeling to find the minimum number of active S-boxes that should be activated during a differential/linear attack and compare with their original security claims. MILP has also been applied before to light encryption device (LED) and CLEFIA, but in another context. [Hadipour et al. 2022] applied MILP to get faster distinguishes for LED (8 rounds) and CLEFIA (10 and 11 rounds) using the division property and [Derbez and Lambin 2022] used MILP to attack 11 rounds of CLEFIA in the key-recovery setting.

The rest of the paper is organized as follows. In Section 2, we describe the MILP model from [Mouha et al. 2011] and how it is applied in the context of linear and differential cryptanalysis. A brief description of SQUARE, LED and CLEFIA algorithms focusing on the important mechanism required by MILP will be given in Section 3. Then, in Section 4, we give the details on the application of the method in each algorithm, reporting the results and making appropriate comparisons. Section 5 concludes the paper and give further directions of research.

2. The MILP model

Since its introduction by [Mouha et al. 2011], MILP modeling of ciphers has been extensively applied to prove security against linear and differential cryptanalysis in many different encryption algorithms. One of the advantages of this method is its generality and its adaptability. In particular, it can be applied to any word-based cipher constructed using linear permutation layers, S-boxes, XOR operations and/or three-forked branches, as is the case of Enocoro [Watanabe and Kaneko 2007] and AES.

Another important advantage of this technique is that the workload of the cryptanalist is reduced to simply describing the cipher by means of linear equations expressing how the input and output chunks of words relate to each other. Once this step is done, the equations, and restrictions generated by the internal mechanisms of the ciphers are put into a linear solver, and an integer linear equation representing the number of active Sboxes is minimized. This gives us an almost automatic way of proving security

bounds against both linear and differential cryptanalysis.

It is also worth noting that there are other techniques suitable for finding the minimum number of active Sboxes in a cipher but MILP requires less programming efforts from the cryptanalist (see [Mouha et al. 2011, page 3]).

We will give a brief idea of how the method works in the context of differential cryptanalysis and refer to the reader to look at [Mouha et al. 2011] for more information. To make the explanation clear we will describe the case where the input and output of the operations in a cipher are represented in the level of bytes.

When working with differential cryptanalysis we are interested in the difference of byte strings and a key concept for applying MILP is the difference vector. For a string $\Delta = (\Delta_0, \Delta_1, \ldots, \Delta_{n-1})$ of n bytes, the difference vector $x = (x_0, x_1, \ldots, x_{n-1})$ is such that each $x_i = 0$ if the byte $\Delta_i = 0$ and $x_i = 1$ otherwise. This is because the only important information here is whether the byte difference is zero or not, regardless of its value.

All input and output variables of the ciphers are treated as unique variables, independent of the round we are in (in case the cipher is based on rounds). For each operation (linear transformation, XOR, etc) we have a set of equations describing it. Every operation involving input and output variables is analyzed, and possibly generates a restriction involving these variables, which can be written in terms of inequalities. In the end, an objective function involving only the input variables that enter the Sboxes is created. Gathering these variables together with these inequalities and the objective function, we can generate what is commonly known as a MILP problem, which we can type into one of many available solvers and find out the answer. That answer tells us what is the minimum number of active Sboxes that comprises with the working mechanisms of the cipher being analyzed.

For a given operation $\mathcal{O}$, let $(x^{\mathcal{O}}_{in_1}, x^{\mathcal{O}}_{in_2}, \ldots, x^{\mathcal{O}}_{in_M})$ be the input differences and $(x^{\mathcal{O}}_{out_1}, x^{\mathcal{O}}_{out_2}, \ldots, x^{\mathcal{O}}_{out_N})$ be the corresponding output differences. Another useful concept is of the *differential branch number* $\mathcal{B}_D$. It is defined as the minimum of the sum of the number of active bytes in the input and output of the operation, excluding the trivial case where all input and output bytes are zero. It describes how the input and output bytes are related in a certain operation. Using $\mathcal{B}_D$ and the input and output bytes of $\mathcal{O}$, the inequalities describing it in the MILP model are the following:

$$
\begin{aligned}
x^{\mathcal{O}}_{in_1} + x^{\mathcal{O}}_{in_2} + \ldots x^{\mathcal{O}}_{in_M} + x^{\mathcal{O}}_{out_1} + x^{\mathcal{O}}_{out_2} + \cdots + x^{\mathcal{O}}_{out_N} &\geq \mathcal{B}_D d^{\mathcal{O}}, \\
d^{\mathcal{O}} &\geq x^{\mathcal{O}}_{in_1}, \\
d^{\mathcal{O}} &\geq x^{\mathcal{O}}_{in_2}, \\
&\ldots\ldots \\
d^{\mathcal{O}} &\geq x^{\mathcal{O}}_{in_M}, \\
d^{\mathcal{O}} &\geq x^{\mathcal{O}}_{out_1}, \\
d^{\mathcal{O}} &\geq x^{\mathcal{O}}_{out_2}, \\
&\ldots\ldots \\
d^{\mathcal{O}} &\geq x^{\mathcal{O}}_{out_N},
\end{aligned}
\tag{1}
$$

where $d^{\mathcal{O}}$ is a *dummy* binary variable, which is zero when all input and output is zero and 1 otherwise. This is necessary to avoid the trivial case where all input and output are zero and to avoid the resolution of several integer linear programs as was needed in [Bogdanov 2011].

If $\mathcal{O}$ is the XOR operation, then $\mathcal{B}_D = 2$ and we can write the first inequality of (1) as

$$x^{\oplus}_{in_1} + x^{\oplus}_{in_1} + x^{\oplus}_{out_1} \geq 2d^{\oplus}, \tag{2}$$

where $d^{\oplus}$ is the corresponding dummy variable. The equations describing a linear transformation in a cipher are similar, except that the differential branching number changes depending on the operation.

For the case of linear cryptanalysis we only need to change the differential branch number (DBN) to the linear branch number (LBN) $\mathcal{B}_L$ and also treat the three-forked branch as an operation whose input and output variables need to be taken into account.

With this in mind, we describe in the next section a collection of algorithms that we found suitable for the application of the MILP technique due to their structure not only in terms of what operations they involve, but also in terms of how the algorithm is applied.

3. Chosen block ciphers suitable for MILP applications

All of the following ciphers were chosen due to their academic relevancy and for having two properties required for MILP to work.

The first required property is being word-based, which means that all operations work only in a word to word level: there are no bit-wise shifts/rotations or bit-wise permutations. For instance, if a cipher contains an XOR operation of two 8-bit words in a given part and a 3-bit shift operation in another, than it is not word-based. Using words of different size throughout the cipher is also unwanted.

The second property we need is having Sboxes as the only non-linearity source. This means that the cipher's design adds non-linearity through a function or lookup table that substitutes a given input by an output in such a way that is non-linear. This is needed because the MILP model intends to count the minimum number of such Sboxes that must be activated in order to attack the cipher, either in a differential or a linear context. Other sources of non-linearity cannot be dealt through this method nor can they be ignored by it.

Therefore, we looked for the most relevant ciphers with these two properties that were not already tested (which excluded the most prominent example, AES) and ended up with two similar ciphers in design. The first one is the SQUARE cipher, well known for being the predecessor of the Rijndael (AES), by the same authors [Daemen et al. 1997], and for introducing the Square Attack, a new form of cryptanalysis at the time. The second algorithm targeted is LED, one of the first in a class of block ciphers labeled as lightweight which means that this cipher had software implementation speed as the main focus, while still maintaining enough security to be used. While based on a different design, the CLEFIA cipher also has the desired property of being word-based and using S-boxes as the only non-linearity source. It is the security standard for the products of the

Sony company[1] and has been subjected to many different analysis by the cryptography community since its introduction.

The following subsections describe the main mechanisms of each cipher and, since the key schedule part does not play a role in the application of MILP, we leave it out of the algorithm descriptions below.

3.1. SQUARE

The block cipher square was introduced in [Daemen et al. 1997] and it has block and key length of 128 bits. Its design takes into account resistance against differential and linear cryptanalysis. In terms of performance, careful choice of the building blocks was made to allow for efficient implementations on many processors.

Each round of SQUARE comprises four different transformations. A linear transformation applied separately to each state row, an Sbox (nonlinear transformation), a byte transformation (basically interchanging of columns and rows of the state) and a bit-wise round key addition. All of them operating on a 4 by 4 array of bytes.

The whole cipher SQUARE is defined as eight rounds which are applied after a key addition.

3.2. LED

LED is a 64-bit block cipher dedicated to compact hardware implementation. It was presented in [Guo et al. 2011] and is based on the design principles of AES which allows one to obtain simple bounds in terms of the number of active S-boxes during encryption. Its design features the well known AES operations such as S-boxes, ShiftRows and MixColumns. To understand the working mechanisms of LED, one can think of the cipher state as arranged in a 4 by 4 grid where each nibble represents an element from $GF(2^4)$. Field multiplication is done with the polynomial X^4+X+1. The initial state of the cipher is a 4 by 4 grid whose entries are formed by the 16 four-bit nibbles of the message. The cipher has no key-schedule and nibbles of subkeys are added to the state using bit-wise exclusive-or. The key size can be either 64 or 128 bits and the number of steps during encryption varies (8 applications of the step function for 64 bit and 12 applications for 128 bit key).

The step function is the core for encryption and it is defined as the application of the AddConstants (bitwise shift applied just to some constants, which does not affect the state), SubCells (applies S-Boxes to each state nible), ShiftRows (rotation of rows to the left) and MixColumnsSerial (multiplication of array state vector by a matrix) operations in order. The SubCells operation is the most important one since it is the one that introduces non-linearity to the cipher. For more details see [Guo et al. 2011, Section 2.1].

3.3. CLEFIA

CLEFIA was presented in [Katagi and Moriai 2011]. It is a highly efficient block cipher and achieves a good performance both in hardware and software. It has 128 bit block size with key lengths of 128, 192 and 256. It is constructed based on a generalized Feistel structure using two 32 bit F functions at each round. These functions are

[1] `https://www.sony.net/Products/cryptography/clefia/`. Accessed 06/06/2023.

different compared to traditional Feistel structure and they require more rounds. On the other hand, the F-functions are smaller and plural F-functions can be processed simultaneously, surpassing the disadvantage of having more rounds. Two F functions are used in each round, but they have the same design, except for the choice of the internal S-boxes. The S-boxes are used to introduce non-linearity through the Diffusion Switching Mechanism, which led to strong resistance against certain attacks studied in the original paper [Katagi and Moriai 2011].

In addition, CLEFIA allows for flexible implementation in both software and hardware.

4. Results

In this section, we study the security of SQUARE, LED and CLEFIA against differential and linear cryptanalysis through the semi-automated MILP technique. The reasoning for choosing these algorithms is that they share two properties that are needed for applying the original MILP method as was explained in the beginning of Section 3.

For each of these ciphers, the main objective was to obtain the MILP program in such a way that a solver is able to find out what is the minimum number of Sboxes needed to be activated for a differential or linear attack to be applied. Both SQUARE and LED have the same MILP program for both linear and differential cryptanalysis because the differential and linear branch numbers are equal for each operation of these ciphers, as they are all SPNs. On the other hand, CLEFIA is a Feistel cipher, which implies the presence of three-forked branch operations. This operation is only relevant for linear cryptanalysis. We have a similar situation for the XOR operation, which is needed for differential cryptanalysis but not for linear cryptanalysis.

In the following subsections, we describe and compare our results to the existing literature. All of our tests were conducted by generating the MILP program of each relevant number of rounds for each cipher. The resulting programs were then fed to the open-source MILP solver SCIP [Bestuzheva et al. 2021] which returned the minimum amount of Sboxes necessary for each of the aforementioned cases. For replication purposes, our MILP programs as well as the results for the SCIP execution can be found at `https://github.com/MfMhj3uNy5gfp4Z/MILP-results/tree/master`.

We ran all tests on a AMD FX(tm)-8320 Eight-Core Processor 3.50 GHz and, in general, the results were obtained in less than 20 seconds for up to 10 rounds for all ciphers, ramping up to anywhere between 5 to 10 minutes for cipher LED and CLEFIA for 18 rounds. Since LED has 48 rounds, it lakes long periods of time to finish all of the above 19 rounds and thus, for any attempt to execute past a established time constraint given by the available computational capacity we forcefully stopped the solver if the expected results based on its design were already attained (see the observation in Table 2).

4.1. SQUARE

The SQUARE cipher was constructed with the famous wide trail design strategy [Daemen 1995], in which the choices for the construction of the cipher are based in two criteria:

1. The maximum difference propagation probability δ (security against differential

attacks), as well as linear propagation probability λ (security against linear attacks) of the chosen Sboxes must be as low as possible.
2. The linear parts must not leave any trail with few active Sboxes.

This design strategy has been used across the board in the creation of block ciphers and specially Substitution Permutation Networks. The most prominent ones being the SQUARE and Rijndael [Daemen and Rijmen 2002] ciphers. The latter became known as AES and is the American national standard for symmetric encryption, as well as one of the most used block ciphers in the world.

In this context, SQUARE presents an Sbox with $\delta = 2^{-6}$ and $\lambda = 2^{-3}$, which means that any attack that activates at least 22 Sboxes is unfeasible since it would imply $\frac{1}{\delta^{22}} = 2^{132}$ plaintext-ciphertext pairs for a differential attack whereas there are only 2^{128} possible pairs for the SQUARE cipher.

As for a linear attack, at least 43 Sboxes need to be activated to make the attack unfeasible, since $\frac{1}{\lambda^{43}} = 2^{129}$ is bigger than the available 2^{128} pairs.

Table 1 shows our results on the cipher SQUARE. It is possible to notice that 4 rounds are enough to guarantee security against differential cryptanalysis and 6 rounds suffices for resistance against linear cryptanalysis.

Table 1. Minimum amount of Sboxes to be active in any given differential or linear attack for the SQUARE cipher, obtained through the MILP program.

# rounds	1	2	3	4	5	6	7	8	9	10	11	12
# Sboxes	1	5	9	25	26	30	34	50	51	55	59	75

4.2. LED

The LED cipher is also termed as an AES-like cipher since it uses the same kind of operations as the AES. AES-like ciphers all use as basis the wide trail strategy. Therefore, we should expect to find the same minimum Sboxes per round as the SQUARE. Furthermore, the states and the branch numbers of the operations are remarkably similar.

The main difference though is that LED uses a 64 bit state divided into words of size 4, also called nibbles. Consequently, the Sbox has to be different and, accordingly, has different values for the differential probability δ and linear probability λ. The Sbox is reused from the PRESENT cipher [Bogdanov et al. 2007] and it has $\delta = 2^{-2}$ and $\lambda = 2^{-7}$.

Since the block state has 64 bits, the maximum amount of plaintext-ciphertex pairs to be used for an attack has to be at most 2^{64}. Table 2 shows that the threshold or number of rounds for guaranteed security through the MILP method against differential attacks is such that the corresponding minimum number of Sboxes s is given by $\frac{1}{\delta^s} = 2^{2s} \geq 2^{64}$. The minimum number of rounds is 7 ($2^{2\times 34} = 2^{68} \geq 2^{64}$). For the linear case, one has $\frac{1}{\lambda^s} = 2^{7s} \geq 2^{64}$ and hence 4 rounds are enough ($2^{7\times 25} = 2^{175} \geq 2^{64}$).

Although we are handling the cipher as composed simply by rounds, it is actually composed by steps, which in turn are composed of 4 consecutive rounds. The 64-bit key version of LED has 6 steps while the 128-bit key version has 12 steps. Thus, using the aforementioned calculations, LED needs two steps to be secure against differential attacks and only one step to be secure against linear attacks.

Table 2. Minimum amount of Sboxes to be active in any given differential or linear attack for the LED cipher, obtained through the MILP program. These results were obtained by the solver before finishing its entire search. Although that means it would be theoretically possible finding a better result, the wide trails design strategy used to construct LED supports that the best estimate could not be smaller than the ones obtained bellow.

# rounds	# Sboxes	# rounds	# Sboxes	# rounds	# Sboxes	# rounds	# Sboxes
1	1	13	76	25	151*	37	226*
2	5	14	80	26	155*	38	230*
3	9	15	84	27	159*	39	234*
4	25	16	100	28	175*	40	250*
5	26	17	101	29	176*	41	251*
6	30	18	105	30	180*	42	255*
7	34	19	109	31	184*	43	259*
8	50	20	125*	32	200*	44	275*
9	51	21	126*	33	201*	45	276*
10	55	22	130*	34	205*	46	280*
11	59	23	134*	35	209*	47	284*
12	75	24	150*	36	225*	48	300*

4.3. CLEFIA

The CLEFIA cipher uses a different base structure in comparison to the other two ciphers we have seen so far. While SQUARE and LED are Substitution Permutation Networks, CLEFIA is a generalized Feistel cipher. Generalized Feistel ciphers characteristically have their state divided into a power of two number d and only half of those are used as input to a non-linear function F whose output is XORed to the other half of the state. Then, a simple rotation is applied to all parts in the state.

This internal structure implies that there are three-forked branches before the input is sent to F. This operation is relevant to the linear attack but not to the differential one, which in turn contrasts with the XOR operation, that is relevant only to the differential attack.

Notably though, Table 3 shows that both cases are remarkably similar, the only difference being the minimum amount of Sboxes necessary for 6 rounds, in which differential attacks need 12 Sboxes, while linear attacks require 11. This result was also obtained by the authors [Katagi and Moriai 2011] through ad-hoc computational search.

CLEFIA also uses two Sboxes (S_0 and S_1) while both SQUARE and LED use just one. For the Sbox S_0 we have $\delta_0 = 2^{-4.67}$ and $\lambda_0 = 2^{-4.38}$ and for S_1 we have $\delta_1 = \lambda_1 = 2^{-6}$. Since both are used in parallel, the analysis can become complex and even gets outside the scope of the MILP method.

Therefore, we chose $\delta = \delta_0$ and $\lambda = \lambda_0$ for all calculations, since they are the best ones. This implies that we assume all Sboxes that are active are S_0. Although unrealistic in a practical sense, this still conforms to the main purpose of the method, which is to find a lower bound to the number of rounds necessary to guarantee security.

We now use the same calculations as for SQUARE and LED to compute the

Table 3. Minimum amount of Sboxes to be active in any given differential or linear attack for the CLEFIA cipher, obtained through the MILP program.

Differential				Linear			
# rounds	# Sboxes	# rounds	# Sboxes	# rounds	# Sboxes	# rounds	# Sboxes
1	0	10	18	1	0	10	18
2	1	11	20	2	1	11	20
3	2	12	24	3	2	12	24
4	6	13	24	4	6	13	24
5	8	14	25	5	8	14	25
6	12	15	26	6	11	15	26
7	12	16	30	7	12	16	30
8	13	17	32	8	13	17	32
9	14	18	36	9	14	18	36

minimum amount of rounds necessary. A CLEFIA block has 128 bits, thus any attack that requires over 2^{128} plaintext-ciphertext pairs is unfeasible. Then, the amount of rounds is such that the associated minimum number of active Sboxes s conforms to $\frac{1}{\delta^s} = 2^{4.67 \times s} \geq 2^{128}$ for differential and $\frac{1}{\lambda^s} = 2^{4.38 \times s} \geq 2^{128}$ for linear attacks.

For the differential case, 16 rounds are enough, since it requires 30 Sboxes to carry an attack which satisfies $2^{4.67 \times 30} = 2^{140.1} \geq 2^{128}$. Although the linear probability is lower than the differential, the same amount of rounds is necessary for linear attacks as the same 30 Sboxes satisfy $2^{4.38 \times 30} = 2^{131.4} \geq 2^{128}$.

Besides the use of two Sboxes, CLEFIA also has two different linear operations, which makes it harder to explore cancellations and thus increases the minimum amount of Sboxes necessary to apply an attack. Unfortunately, the word-based MILP method lacks flexibility to deal with these intricacies.

5. Conclusion

In this paper, we applied the MILP method to show proof of security against differential and linear cryptanalysis for the ciphers SQUARE, LED and CLEFIA. All three algorithms are secure as expected, SQUARE needing only 6 rounds to be safe against differential and 4 to be safe against linear, while LED needs 2 steps (7 rounds) for differential and 1 step (4 rounds) for linear. CLEFIA requires 6 rounds to be secure from differential attacks and 11 for linear.

The results obtained here are consistent with the ones presented by the original authors, i.e., the same minimum amounts of Sboxes expected are equal to the ones obtained by our MILP model for all three ciphers. Indeed, the authors of both SQUARE and LED got their results in a theoretical manner. Through the wide trail design strategy they showed that the minimum amount of Sboxes grows in a fixed rate following a $(1, 4, 4, 16)$ pattern (ex. LED with 4,5,6,7,8 rounds have 25, 26, 30, 34, 50 minimum Sboxes, which agrees with the results of both Tables 2 and 1). On the other hand, the authors of CLEFIA use a computer program to conduct an ad-hoc search to find the minimum amounts of Sboxes, since it's design does not have a theoretical result backing it up. Therefore, the MILP model is a formidable tool for proving the security against linear and differential

cryptanalysis for certain types of ciphers.

Although the word-based MILP method is useful for some ciphers, its use is limited since it is unable to give a more accurate lower bound to the number of Sboxes for more complex designs, such as the use of more than one linear operation in parallel in the CLEFIA cipher.

For that reason, further research involves exploring more complete ways to develop MILP programs, such as the use of the bitwise MILP method which is capable of dealing with the intricacy of CLEFIA, as well as other ciphers that contain bitwise linear operations, such as DES, ARIA, Twofish, FEAL and Serpent.

References

Bestuzheva, K., Besançon, M., Chen, W.-K., Chmiela, A., Donkiewicz, T., van Doornmalen, J., Eifler, L., Gaul, O., Gamrath, G., Gleixner, A., Gottwald, L., Graczyk, C., Halbig, K., Hoen, A., Hojny, C., van der Hulst, R., Koch, T., Lübbecke, M., Maher, S. J., Matter, F., Mühmer, E., Müller, B., Pfetsch, M. E., Rehfeldt, D., Schlein, S., Schlösser, F., Serrano, F., Shinano, Y., Sofranac, B., Turner, M., Vigerske, S., Wegscheider, F., Wellner, P., Weninger, D., and Witzig, J. (2021). The SCIP Optimization Suite 8.0. Technical report, Optimization Online.

Bogdanov, A. (2011). On unbalanced feistel networks with contracting mds diffusion. *Des. Codes Cryptography*, (59(1-3)):35–58.

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., and Vikkelsoe, C. (2007). Present: An ultra-lightweight block cipher. In Paillier, P. and Verbauwhede, I., editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, pages 450–466, Berlin, Heidelberg. Springer Berlin Heidelberg.

Daemen, J. (1995). *Cipher and hash function design strategies based on linear and differential cryptanalysis*. PhD thesis, Doctoral Dissertation, March 1995, KU Leuven.

Daemen, J., Knudsen, L., and Rijmen, V. (1997). The block cipher square. In *Fast Software Encryption: 4th International Workshop, FSE'97 Haifa, Israel, January 20–22 1997 Proceedings 4*, pages 149–165. Springer.

Daemen, J. and Rijmen, V. (2002). *The design of Rijndael*, volume 2. Springer.

Derbez, P. and Lambin, B. (2022). Fast milp models for division property. *IACR Transactions on Symmetric Cryptology*, pages 289–321.

Dobraunig, C., Eichlseder, M., and Mendel, F. (2016). Square attack on 7-round kiasu-bc. In *Applied Cryptography and Network Security: 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings 14*, pages 500–517. Springer.

Fu, K., Wang, M., Guo, Y., Sun, S., and Hu, L. (2016). Milp-based automatic search algorithms for differential and linear trails for speck. In *Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers 23*, pages 268–288. Springer.

Guo, J., Peyrin, T., Poschmann, A., and Robshaw, M. (2011). The led block cipher. In *Cryptographic Hardware and Embedded Systems–CHES 2011: 13th International Workshop, Nara, Japan, September 28–October 1, 2011. Proceedings 13*, pages 326–341. Springer.

Hadipour, H., Nageler, M., and Eichlseder, M. (2022). Throwing boomerangs into feistel structures: Application to clefia, warp, lblock, lblock-s and twine. *Cryptology ePrint Archive*.

Karthika, S. and Singh, K. (2023). Cryptanalysis of stream cipher lizard using division property and milp based cube attack. *Discrete Applied Mathematics*, 325:63–78.

Katagi, M. and Moriai, S. (2011). The 128-bit blockcipher clefia. Technical report.

Mironov, I. and Zhang, L. (2006). Applications of sat solvers to cryptanalysis of hash functions. In *Theory and Applications of Satisfiability Testing-SAT 2006: 9th International Conference, Seattle, WA, USA, August 12-15, 2006. Proceedings 9*, pages 102–115. Springer.

Mouha, N., Wang, Q., Gu, D., and Preneel, B. (2011). Differential and linear cryptanalysis using mixed-integer linear programming. In *International Conference on Information Security and Cryptology*, pages 57–76. Springer.

Pal, D., Chandratreya, V. P., and Chowdhury, D. R. (2023). Efficient algorithms for modeling sboxes using milp. *arXiv preprint arXiv:2306.02642*.

Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., and Song, L. (2014). Automatic security evaluation and (related-key) differential characteristic search: application to simon, present, lblock, des (l) and other bit-oriented block ciphers. In *Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, ROC, December 7-11, 2014. Proceedings, Part I 20*, pages 158–178. Springer.

Wardhana, D. and Indarjani, S. (2019). Square attack on 4 round midori64. In *AIP Conference Proceedings*, volume 2168. AIP Publishing.

Watanabe, D. and Kaneko, T. (2007). A construction of light weight panama-like keystream generator. *IEICE Technical Report*.

Xiang, Z., Zhang, W., Bao, Z., and Lin, D. (2016). Applying milp method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*, pages 648–678. Springer.

Zhao, H., Han, G., Wang, L., and Wang, W. (2020). Milp-based differential cryptanalysis on round-reduced midori64. *IEEE Access*, 8:95888–95896.

Zhou, C., Zhang, W., Ding, T., and Xiang, Z. (2019). Improving the milp-based security evaluation algorithm against differential/linear cryptanalysis using a divide-and-conquer approach. *Cryptology ePrint Archive*.