Differential cryptanalysis of full-round ANU-II ultra-lightweight block cipher

Ting Fan\textsuperscript{1,2}, Lingchen Li\textsuperscript{1,\ast}, Yongzhuang Wei\textsuperscript{1,2} and Enes Pasalic\textsuperscript{1,3}

Abstract
Lightweight ciphers are often used as the underlying encryption algorithm in resource-constrained devices. Their cryptographic security is a mandatory goal for ensuring the security of data transmission. Differential cryptanalysis is one of the most fundamental methods applicable primarily to block ciphers, and the resistance against this type of cryptanalysis is a necessary design criterion. ANU-II is an ultra-lightweight block cipher proposed in 2017, whose design offers many advantages such as the use of fewer hardware resources (logic gates), low power consumption and fast encryption for Internet of Things devices. The designers of ANU-II claimed its resistance against differential cryptanalysis and postulated that the design is safe enough for Internet of Things devices. However, as addressed in this article, the security claims made by designers appear not to be well grounded. Using mixed-integer linear programming–like techniques, we identify one-round differential characteristic that holds with probability 1, which is then efficiently employed in mounting the key recovery attack on full-round ANU-II with only 2\textsuperscript{2} chosen plaintexts and 2\textsuperscript{62.4} full-round encryptions. The result shows that the designers’ security evaluation of ANU-II against differential cryptanalysis is incorrect and the design rationale is flawed. To remedy this weakness, we provide an improved variant of ANU-II, which has much better resistance to differential cryptanalysis without affecting the hardware and/or software implementation cost.

Keywords
IoT devices, lightweight block ciphers, differential cryptanalysis, key recovery attack, MILP model

Introduction
With a rapid development of 5G technology and networks, the demand for Internet of Things (IoT) devices is constantly increasing in daily life. The information exchange between IoT devices is commonly performed through embedded encryption algorithms for the reasons of providing secure data transmission, device authentication and other sensitive services. A majority of the IoT devices (especially sensors and microcontrollers) implement as encryption algorithm a class of lightweight block ciphers that are designed to operate under certain power, memory and/or computation constraints.\textsuperscript{1} In general, this family of ciphers is characterized by a low hardware implementation cost and low power consumption, while at the same time providing sufficient security margins in sensitive applications. On the other hand, for applications without the mentioned limitations, traditionally designed encryption algorithms (e.g. advanced encryption standard (AES)) generally have lower computing efficiency due to their robust design rationales.

\textsuperscript{1}Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, China
\textsuperscript{2}State Key Laboratory of Cryptology, Beijing, China
\textsuperscript{3}FAMNIT, University of Primorska, Koper, Slovenia

Corresponding author:
Lingchen Li, Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, 1 Jinji Road, Qixing District, Guilin 541004, Guangxi, China.
Email: lilingchen601@126.com

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License
(https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
Being more suitable for use in restrained environments, lightweight block ciphers have attracted much attention of cryptographic society. To date, there are numerous lightweight block ciphers that have emerged during the last decade, including PRESENT\(^2\) that uses a substation permutation network (SPN), the LBlock\(^3\) cipher implementing the Feistel network, Simon and Speck\(^4\) having highly efficient (fast) software and hardware implementation, and RECTANGLE\(^5\) using the so-called bit-slice technology.

Differential cryptanalysis\(^6\) is one of the earliest generic cryptanalytic tools (along with linear cryptanalysis) applicable to block ciphers. This technique is currently well understood and for testing the resistance of ciphers against differential cryptanalysis the following approach is commonly applied. First, lower bounds on the number of active substitution boxes (S-boxes) is estimated and then a propagating differential characteristic with as high probability as possible needs to be specified. However, the computational complexity of specifying optimal differential paths (involving a few active S-boxes and holding with a high probability) grows exponentially with the number of rounds, which leads to efficient use of different algorithms that may identify optimal solutions.

The use of mixed-integer linear programming (MILP) techniques for this purpose has been quite a common method during the last decade. In 2011, lower bounds of the number of active S-boxes for Enocoro128v2 and AES were obtained by Mouha et al.\(^7\) using MILP-based models. This approach is, however, only applicable to byte-oriented block ciphers. To fill this gap, Sun et al.\(^8\) extended the byte-oriented MILP model to bit-oriented block ciphers for the purpose of searching for lower bounds on active S-boxes and good (having a high probability) differential characteristics. Also, in the work of Sun et al.,\(^9\) a heuristic search algorithm for finding the best differential characteristic was proposed. Later, automated MILP-based search algorithms have been widely used to evaluate the security of block ciphers with quite outstanding results. Fu et al.\(^9\) applied MILP-based techniques to ARX ciphers, where also a search for good differential characteristics and a linear approximation of the SPECK block cipher were addressed. Abdelkhalek et al.\(^10\) further developed the MILP modelling for (large) S-boxes to optimize the probability of differential characteristics. In addition, Sadeghi et al.\(^11\) presented zero-correlation linear approximations and the related-tweak impossible differential characteristics for the block cipher SKINNY. Zhou et al.\(^12\) greatly improved the efficiency of MILP-based search algorithms by embedding the divide-and-conquer idea into the MILP model. In Crypto2020,\(^13\) a novel MILP model with the possibility of handling simultaneously the propagation of differentials and their values was developed. In FSE2021,\(^14\) a suitable MILP model for finding more efficient distinguishers that cover more encryption rounds was introduced.

ANU is an ultra-lightweight block cipher proposed by Bansod et al.\(^15\) However, the full-round ANU cipher was broken by Sasaki\(^16\) using the related-key boomerang attack. ANU-II is an improved version of ANU and was proposed by Dahiphale et al.,\(^17\) which is superior to ANU in terms of memory, latency, throughput and power consumption. However, in 2020, Wang et al.\(^18\) conducted an eight-round integral attack on ANU-II (designed for 25 encryption rounds) and mounted a key recovery attack using \(2^{60}\) chosen plaintexts. The designers made a claim that ANU-II achieves a good resistance against differential cryptanalysis by investigating the differential properties of a round-reduced version of ANU-II. In this article, we show that the statement given by designers about the good resistance against differential cryptanalysis is incorrect and the current design of ANU-II contains serious security flaws.

**Our contributions**

We investigate the security of ANU-II and the main contributions are summarized as follows:

1. First, we construct an accurate MILP model for specifying exact lower bounds on the number of active S-boxes for ANU-II. Most notably and quite surprisingly, this model identifies an interesting property that the lower bounds of the minimum number of active S-boxes of any round always equal zero.
2. Then, by observing the input and output of each round, we unexpectedly (disproving the claims made by the designers) find a one-round iterative differential characteristic with probability 1.
3. Consequently, using repeatedly found one-round differential characteristic, we could mount a key recovery attack on the full-round ANU-II (consisting of 25 encryption rounds), which shows that the design has severe security flaws.
4. Finally, we propose an improved variant of ANU-II with the same hardware or software implementation expenses as the original cipher. Compared to the actual design, this variant is much more robust against differential cryptanalysis.

The summary of our main results and the previous works is shown in Table 1.
The article is arranged as follows. In the ‘Preliminaries’ section, we introduce some basic notations and give a description of the ultra-lightweight block cipher ANU-II. A MILP-based differential search model for a minimal number of active S-boxes is given in the ‘Bit-oriented MILP model for differential analysis’ section. In the ‘Differential cryptanalysis of ANU-II’ section, a one-round iterative differential characteristic with probability 1 for ANU-II is specified, along with its application to the full-round key recovery attack. In the ‘An efficient modification of ANU-II’ section, we propose a new modified variant of ANU-II and estimate its resistance against differential cryptanalysis. Some concluding remarks are given in the ‘Conclusion’ section.

Preliminaries

Notations

The following notations are used throughout the article:

- $P$: 64 bits input plaintext;
- $P'_L$: 32 bits input for the left branch of $r$th round;
- $P'_R$: 32 bits input for the right branch of $r$th round;
- $P'^{+1}_L$: 32 bits output for the left branch of $r$th round;
- $P'^{+1}_R$: 32 bits output for the right branch of $r$th round;
- $C$: 64 bits output ciphertext;
- $\Delta C_r$: $r$th bit of the $r$th round ciphertext pairs’ difference ($0 \leq i \leq 127$);
- $RC_r$: $r$th round constant;
- $w$: block size of the cipher;
- $v$: input size of the S-box;
- $r$: iterative rounds ($0 \leq r \leq 24$);
- $K$: 128 bits master key;
- $K^{(r)}$: $r$th round subkey;
- $K^{(r)}$: 0–31 bits of $K_r$;
- $K^{(r)}$: 32–63 bits of $K_r$;
- $K^{(r)}$: $i$th bit of the $r$th round subkey ($0 \leq i \leq 63$);
- $\oplus$: XOR;
- $\|$: concatenation;

The description of ANU-II

ANU-II is an ultra-lightweight block cipher with a 64-bit block size and an 80/128-bit key size. ANU-II belongs to the family of Feistel networks and uses iteratively 25 encryption rounds. We mainly focus on the variant of ANU-II that uses a 128-bit master key. Let the left- and right-branch inputs of the $r$th round be denoted as $P'_L$ and $P'_R$, respectively, and the corresponding outputs as $P'^{+1}_L$ and $P'^{+1}_R$. The round function is shown in Figure 1.

There are four operations used in the round function: SubBox, AddRoundKey, Rotation and XOR. Among these, the operation SubBox is the only non-linear component and it employs eight S-boxes of size $4 \times 4$ in parallel.

Key schedule

When the master key is of length 128 bits, the key schedule extracts 64-bit key and XOR these with the intermediate encryption state. The first-round subkey is

![Figure 1. The round function of ANU-II.](image-url)
equal to the master key, and the update process for deriving round subkeys is as follows

\[ K = K_{127}, \ldots, K_0 \]
\[ K^{(0)} = K = K_{127}, \ldots, K_0 \]
\[ K^{(0)}_1 = K_{31}, \ldots, K_0 K^{(0)}_2 = K_{63}, \ldots, K_{32} \]

For \( r = 0 \) to 24 do

\[ K^{(r)}_1 = K^{(r)}(31 \sim 0) \]
\[ K^{(r)}_2 = K^{(r)}(63 \sim 32) \]

Rotation to the left by 13 bits

\[ K^{(r)} \lll 13 \]

Applying S-box operation

\[ K^{(r+1)}(3 \sim 0) = S[K^{(r)}(3 \sim 0)] \]
\[ K^{(r+1)}(7 \sim 4) = S[K^{(r)}(7 \sim 4)] \]

Applying S-box and XOR with a round constant

\[ K^{(r+1)}(63 \sim 59) = S[K^{(r)}(63 \sim 59)] \oplus R_{C_r} \]

Note that the S-box in the above key schedule still uses the description in Table 2, and the value of \( R_{C_r} \) equals to \( r \) for the \( r \)th encryption round.

### Bit-oriented MILP model for differential analysis

MILP is a type of integer linear programming, especially suitable for solving optimization problems. Its goal is to minimize or maximize the value of the objective function under certain linear constraints. At the moment, commercial software such as in Sun et al.\(^{19}\) is usually used to solve such optimization problems. In this section, we introduce a bit-oriented MILP model for searching the exact lower bounds of active S-boxes with respect to propagation of differentials and the best differential characteristics in the single-key model.

### Searching for the exact lower bounds of active S-boxes

To identify which S-boxes are active when specific differentials pass through the intermediate states of a given block cipher, we recall the following notation.

**Definition 1.** Let \( X = (X_0, X_1, \ldots, X_{n-1}) \) and \( X^* = (X^*_0, X^*_1, \ldots, X^*_{n-1}) \) be two binary strings of length \( n \).\(^{19}\) The difference between \( X \) and \( X^* \) is defined as \( \Delta X = X \oplus X^* \) \((\Delta X_i = X_i \oplus X^*_i, 0 \leq i < n - 1)\). Then

\[
\Delta X_i = \begin{cases} 
0, & \text{\( \Delta X_i \) is inactive} \\
1, & \text{\( \Delta X_i \) is active}
\end{cases}
\]

**Definition 2.** Let \( N_t (t \in [0, n/w - 1]) \) be a binary indicator that specifies whether the \( t \)th S-box is active or not, given as\(^{19}\)

\[
N_t = \begin{cases} 
1, & \text{\( t \)th S-box is active} \\
0, & \text{otherwise}
\end{cases}
\]

Next, we explain how to model diverse operations, such as: SubBox, XOR, and linear layer operation.

**SubBox.** Assume that \( x = (x_0, x_1, \ldots, x_{w-1}) \) and \( y = (y_0, y_1, \ldots, y_{w-1}) \) are the input and output differences, respectively, of a \( w \times v \) S-box \( N_t \). Then, \( N_t = 1 \) holds if and only if \((x_0, x_1, \ldots, x_{w-1})\) is not the all zero vector, which leads to the following linear inequalities

\[
\begin{align*}
x_0 + \cdots + x_{w-1} & \geq N_t \\
N_t - x_i & \geq 0 \quad (0 \leq i \leq w - 1)
\end{align*}
\]

In addition, we need to consider the propagation of differential values through an (active) S-box. Using the \( H \)-representation of the convex hull (a polyhedron obtained by restricting the Euler space through linear (in)equalities), we transform the propagation of possible differences of an S-box into inequality constraints. Assume that \( x = (x_0, x_1, \ldots, x_{q-1}), q = w + v \) (where \( w \) and \( v \) are the input and output size of the S-box, respectively) is a \( q \)-dimensional possible differential pattern. Then, the induced inequalities, of cardinality \( m \) say, can be expressed as follows
Reduced by Sasaki and Todo. The latter algorithm is called the XOR. The inequality generator is shown to be more efficient in removing redundant inequalities than the algorithm proposed by Sun et al., whose removal of redundant inequalities is more efficient. Generally, two methods are used: the greedy algorithm by Fan et al. and the branch swapping algorithm. In this article, we focus on the greedy algorithm since it appears to be more efficient in removing redundant inequalities. To achieve this, we use the SageMath software that calls the inequality-generator function in the package sage.geometry.polyhedron. The Sage code is given in Figure 2.

\[ \alpha_0, 0x_0 + \cdots + \alpha_0, q x_q + 1 + \alpha_0, q \geq 0 \]
\[ \vdots \]
\[ \alpha_{m-1, 0} x_0 + \cdots + \alpha_{m-1, q-1} x_{q-1} + \alpha_{m-1, q} \geq 0 \]

where \( \alpha_{i,j} \) are binary coefficients generated by SageMath.

For instance, let \( x = (x_0, x_1, x_2, x_3) = 1001, y = (y_0, y_1, y_2, y_3) = 1100 \) denote the input and output differences of a 4x4 S-box. Assuming that \( (1001) \rightarrow (1101) \) is a possible propagation (it occurs with non-zero probability), we express this propagation as an eight-dimensional tuple \((10011101)\). Then, we use the SageMath software that calls the inequality-generator function in the package sage.geometry.polyhedron. The Sage code is given in Figure 2.

Similarly, we can convert all possible differential propagation patterns into inequalities in accordance to the differential distribution table (DDT) given in Appendix 1 for the S-box of ANU-II. SageMath is generally used to transform possible differential patterns into inequalities in a large number of linear inequalities.

When this phase is completed, some redundant inequalities need to be excluded. To achieve this, commonly two methods are used: the greedy algorithm by Sun et al., whose removal of redundant inequalities is not optimal and the MILP-reduced algorithm introduced by Sasaki and Todo. The latter algorithm is preferred in this article since it appears to be more efficient in removing redundant inequalities than the greedy algorithm.

**XOR.** Let \( a \oplus b = c \), where \( a, b \) and \( c \) represent a single bit. Then, the following constraints can be derived
\[ \begin{align*}
  a + b + c & \geq 2d \\
  d & \geq a, d \geq b, d \geq c \\
  a + b + c & \leq 2
\end{align*} \]
where \( d \) is a dummy variable with value in \( \{0, 1\} \).

**Linear operation.** For ciphers that employ the Feistel network, the rotation and branch swapping are both included when modelling the linear layer operation. These operations can be efficiently described by linear equalities. To clarify this, we consider the right (cyclic) rotation of ANU-II as an example. Suppose the input is \( y = (y_1, \ldots, y_0) \) and that \( z = (z_1, \ldots, z_0) \) denotes the output after rotation to the right by 3 bits. The following constraints are then easily deduced
\[ \begin{align*}
  z_0 &= y_3 \\
  z_1 &= y_4 \\
  \vdots \\
  z_9 &= y_0 \\
  z_{10} &= y_1 \\
  z_{11} &= y_2
\end{align*} \] (5)

An initial constraint. The inequality \( x_0 + x_1 + \cdots + x_{p-1} \geq 1 \) needs to be introduced to ensure that the initial input differential is non-zero. Here, \( n \) denotes the block size of the cipher.

**The objective function.** The \( r \)-round objective function, minimizing the number of active S-boxes, is set as (\( w \) indicating the input size of an S-box)
\[ \min_{\text{differentials}} \sum_{j=0}^{r-1} \left( \sum_{j=0}^{w-1} N^j \right) \] (6)
where \( N^j \) indicates whether the \( j \)-th S-box in the \( j \)-th encryption round is active or not.

**Search for the best differential characteristic.**

To find the best differential characteristic, we encode the probability information for possible differential patterns of an S-box. Except for the non-linear SubBox operation, the constraints induced by the XOR and Rotation operations are same as described in the ‘Searching for the exact lower bounds of active S-boxes’ section.

**SubBox.** We consider the 4-bit S-boxes of ANU-II block cipher. There are three non-zero differential probabilities according to the DDT of S-box: \( 16/16 = 2^0, 4/16 = 2^{-2} \) and \( 2/16 = 2^{-3} \). We need to introduce two probability variables \( (p_0, p_1) \) to encode and distinguish these three possible differential patterns of the S-boxes of ANU-II.
\[ (p_0, p_1) \in \{ (0, 0), \text{Pb(differential)} = 2^0 \\
(0, 1), \text{Pb(differential)} = 2^{-2} \\
(1, 1), \text{Pb(differential)} = 2^{-3} \] (7)
For instance, the possible differential pattern (0000 → 0000) can be expanded from the eight-dimensional tuple (0,0,0,0,0,0,0,0) to the 10-dimensional tuple (0,0,0,0,0,0,0,0,0,0,0). Similarly, using the DDT, the differential pattern (0001 → 0101) with probability 2⁻² is extended from (0,0,1,0,1,0,1,1,0,1,1) to (0,0,1,0,1,0,1,1,0,1,1). Also, (0010 → 0011) is extended to (0,0,1,0,0,0,1,1,1,0,1).

In this way, all possible differential patterns can be encoded. Then, using SageMath, these differential patterns (containing the probability information) are transformed into linear inequalities using the code in Figure 2.

The MILP-reduced algorithm is then applied to remove redundant inequalities.

**The objective function.** Since equation (7) only describes the probabilities of differentials of a single S-box, we need to introduce additional notation to refer to different S-boxes used. Using \( t \in [0, n/w - 1] \) to indicate the \( t \)th S-box in each round, we derive the following conditions

\[
(p_{t,0}, p_{t,1}) \in \begin{cases} 
(0,0), & p_{t,0} + 2p_{t,1} = 0 \\
(0,1), & p_{t,0} + 2p_{t,1} = 2 \\
(1,1), & p_{t,0} + 2p_{t,1} = 3
\end{cases} \tag{8}
\]

where the integer representation 0, 2 and 3 of the binary tuple \((p_{t,0}, p_{t,1})\) on the right of equation (8), is exactly of the opposite sign compared to the exponents in equation (7). Since a product of probabilities expressed as \( p_j = 2^a \) (treating different S-boxes as independent events) will correspond to the addition of exponents \( a_i \) (with \( a_i \leq 0 \)), it is natural to model the probability of differential characteristics of the \( j \)th round as \( \sum_{i=0}^{(n/w)-1} p'_{i,0} + 2p'_{i,1} \), which takes into account all the S-boxes in the \( j \)th round and the superscript \( j \) in \( p'_{i,j} \) refers to the \( j \)th round as well. Then, \( p = \sum_{j=0}^{n-1} \sum_{i=0}^{(n/w)-1} p'_{i,0} + 2p'_{i,1} \) will contain the information about the probability of differential characteristics for \( r \) encryption rounds. It is well-known that a random permutation will induce a uniform probability distribution of differentials being \( 2^{-n} \) (for a block cipher of size \( n \)), and therefore, we need a differential characteristic with probability \( 2^{-p} \gg 2^{-n} \). Thus, encoding the probability information of differential characteristics as above, our goal is to minimize the exponent \( p \). Therefore, the objective function for \( r \) encryption rounds is set as

\[
\min \sum_{j=0}^{r-1} \sum_{i=0}^{(n/w)-1} p'_{i,0} + 2p'_{i,1} \tag{9}
\]

**An initial constraint.** We need to add the inequality \( x_0 + x_1 + \cdots + x_{n-1} = 1 \) to ensure that the initial input differential is non-zero, as otherwise the MILP solver would return 0 as an optimal solution.

**Differential cryptanalysis of ANU-II**

In this section, we first establish an MILP model that specifically searches for lower bounds on the number of active S-boxes for ANU-II. Then, again using our MILP model, we specify a single round differential characteristic that holds with probability one. Finally, we mount a key recovery attack on the full-round ANU-II that only requires \( 2^5 \) chosen plaintexts and \( 2^{62.4} \) full-round encryptions.

**The exact lower bounds of the number of differential active S-boxes of ANU-II**

In the single-key setting, the linear constraints discussed previously include SubBox, XOR, rotation and the initial constraint. Assume that 64-bit input difference is \((x_63, \cdots, x_0)\), where \( x_0 \) is the least significant bit (LSB). More precisely, the input differences of left branch is represented as \((x_63, \cdots, x_32)\), and the right branch is \((x_31, \cdots, x_0)\). The corresponding MILP model, concerning the ANU-II cipher, is then built using the methods described in the ‘Searching for the exact lower bounds of active S-boxes’ section.

**SubBox.** ANU-II employs the Feistel network and uses eight S-boxes of size \( 4 \times 4 \). For each of the eight S-boxes of ANU-II, there are 97 possible differential patterns according to the DDT shown in Appendix 1. SageMath is used to transform 97 possible differential patterns into 287 inequalities. After applying the MILP-reduced algorithm, only 23 inequalities remain. Below, we use the convention that \( x_0 \) and \( y_0 \) refer to the LSBs of the input and output, respectively, of any S-box in ANU-II. Also, the variables \( N_i \) identify the eight S-boxes, for \( 0 \leq i \leq 7 \). Our MILP model for the first S-box is specified by equations (10) and (11).

Essentially, equation (10) indicates whether the first S-box is active, whereas equation (11) specifies the propagation of differentials through the first S-box. Similarly, one can specify the constraints for the remaining seven S-boxes of ANU-II.

\[
\begin{aligned}
\{ x_0 + \cdots + x_3 \geq N_0 \\
N_0 - x_i \geq 0 \quad (0 \leq i \leq 3)
\end{aligned} \tag{10}
\]
Table 3. Lower bounds on the number of active S-boxes for ANU-II.

| Rounds | Dahiphale et al.\textsuperscript{17} | This article |
|--------|--------------------------------------|-------------|
| 1      | 0                                    | 0           |
| 2      | 2                                    |             |
| 3      | 6                                    |             |
| 4      | 13                                   |             |
| ...    |                                      |             |
| 25     |                                      |             |

\[
\begin{align*}
-&x_3 + x_1 - x_0 - y_2 - y_1 \geq 3 \\
-x_3 - 3x_2 + x_1 - x_0 - 3y_3 - 2y_2 - 3y_1 - y_0 \geq 10 \\
-x_3 - 2x_2 + 2x_1 - x_0 - y_3 + 2y_2 + 2y_1 - y_0 \geq 4 \\
2x_3 + x_2 - 2x_1 + 2x_0 + y_3 - 3y_2 - 3y_1 - y_0 \geq 6 \\
3x_3 - x_2 + 3x_1 - x_0 + 3y_3 - y_2 + 3y_1 + 2y_0 \geq 0 \\
6x_3 + 3x_2 + 3x_1 + 6x_0 - 2y_3 - 2y_2 - 2y_1 + y_0 \geq 0 \\
x_3 + 3x_2 - 2x_1 + x_0 - y_3 + y_2 + y_1 + 2y_0 \geq 0 \\
x_3 + 2x_2 - 2x_0 + y_2 + 2y_1 + y_0 \geq 0 \\
x_3 + 3x_2 + 2x_1 + 2x_0 + 2y_3 + y_2 - 2y_1 - y_0 \geq 0 \\
x_3 - 3x_2 - x_1 - 3x_0 + 2y_3 + 2y_2 + 2y_1 - y_0 \geq 0 \\
x_3 + x_2 - x_1 - 2x_0 + y_3 + y_2 - y_1 - 2y_0 \geq 4 \\
-x_3 + x_2 + 3x_1 + 3x_0 + 3y_3 + 3y_2 - y_1 + 2y_0 \geq 0 \\
2x_3 + 3x_2 + 2x_1 - x_0 + 2y_3 + 2y_2 - y_1 - y_0 \geq 0 \\
2x_3 + x_2 - 2x_1 - x_0 - 2y_3 + 2y_2 - y_1 + y_0 \geq 4 \\
x_3 - 3x_2 + 2x_1 + x_0 + 2y_3 + y_2 + y_1 + 2y_0 \geq 0 \\
-x_3 + 3x_2 - x_1 + 4x_0 - y_3 + 4y_2 - y_1 - 4y_0 \geq 0 \\
-2x_3 - 2x_2 - x_1 - 2x_0 + 3y_3 + 3y_2 + y_1 + y_0 \geq 5 \\
-x_3 + x_2 + x_1 + 2x_0 - y_3 + 2y_1 - 2y_0 \geq 2 \\
-2x_3 + x_2 - x_1 + x_0 + y_3 - y_2 + y_1 - 2y_0 \geq 4 \\
-2x_3 + x_2 - 2x_0 - y_2 - y_1 - y_0 \geq 5 \\
-2x_3 - 2x_2 - x_1 + 2x_0 - 2y_3 - y_2 - y_1 + y_0 \geq 7 \\
-2x_3 - 3x_2 - 3x_1 + x_0 - 3y_3 + y_2 - 2y_1 - y_0 \geq 11 \\
x_3 - 3x_2 - 3x_1 - 2x_0 - 3y_3 - 2y_2 + y_1 - y_0 \geq 11 \\
\end{align*}
\] (11)

Linear operation. We consider one of the cyclic rotations of ANU-II as an example. Suppose that \(z = (z_{31}, \ldots, z_0)\) denotes the right branch output difference after rotation to the right by 3 bits. The following constraints are easily deduced

\[
\begin{align*}
&z_0 = x_3 \\
&\vdots \\
&z_{29} = x_0 \\
&z_{30} = x_1 \\
&z_{31} = x_2 \\
\end{align*}
\] (12)

XOR. We only consider the constraints for the first XOR operation. (Similarly, the constraints for the second operation can also be attained.) Let \(y = (y_{31}, \ldots, y_0)\) be the left branch output difference through S-box layer, \(s = (s_{31}, \ldots, s_0)\) represents output difference after the first bitwise XOR operation, and \(d = (d_{31}, \ldots, d_0)\) is a dummy variable

\[
\begin{align*}
y_{31} + z_{31} + s_{31} &\equiv 2d_{31} \\
y_{31} + z_{31} + s_{31} &\equiv 2, \\
d_{31} &\equiv y_{31} \\
d_{31} &\equiv z_{31} \\
d_{31} &\equiv s_{31} \\
\end{align*}
\] (13)

An initial constraint

\[
x_0 + \cdots + x_{63} \geq 1
\]

The objective function. Set the \(r\)-round objective function as

\[
\min \sum_{j=0}^{r-1} \left( \sum_{i=0}^{7} N_i \right)_j
\] (14)

At this point, all inequality constraints are specified and a search for the best lower bound on the number of active S-boxes can be performed. We utilize the commercial software Gurobi\textsuperscript{22} for this purpose. The experimental results are shown in Table 3, and most notably, the lower bound on the number of active S-boxes equals 0. This observation is very interesting and it essentially indicates that ANU-II might have some serious design flaws.

Iterated differential characteristic

The above result related to the lower bound on the number of active S-boxed (being 0), initiated a careful investigation of the design rationales of ANU-II, which then resulted in the identification of an iterative differential characteristic that holds with probability 1, as shown in Figure 3. Apparently, this differential characteristic holds with probability 1, because active bits only appear on the right-hand side (in total 32 bits) and the corresponding S-boxes are not active.

Key recovery attack on full-round ANU-II

Employing the iterative differential characteristic of a single round found in the ‘Iterated differential characteristic’ section, one can construct a full-round differential characteristic by iterating the single round differential 25 times, as described in Figure 4. We use ‘1’ and ‘0’ to denote active and inactive bits,
The differential attack on 25-round ANU-II is now described, which heavily relies on some interesting relationships between the subkeys used in the last two encryption rounds.

Property 1. The subkeys of two consecutive encryption rounds are heavily related, due to the key schedule of ANU-II. For our purpose, the relation between the subkey bits of the 24th and 25th rounds (96 bits in total) is of interest

$$\begin{align*}
K_1^{(24)}(31 \sim 13) &= K_1^{(23)}(18 \sim 0) \\
K_2^{(24)}(58 \sim 45) &= K_2^{(23)}(45 \sim 32) \\
K_3^{(24)}(63 \sim 59) &= K_3^{(23)}(50 \sim 46) \oplus RC_r
\end{align*}$$

where $RC_r$ is round constant and the notation $K_i^{(r)}(k \sim l)$ generally means that the key bits involved range from $k$th to $l$th position.

The attack consists of the following four steps:

1. Select $2^m$ plaintext pairs where the difference of each pair is the same $\Delta P = P \oplus P' = (\Delta L_0 \parallel \Delta R_0)$. The values of $\Delta L_0$ and $\Delta R_0$ are shown in Figure 4.
2. Since the 25-round differential characteristic holds with probability with 1, having the input difference $\Delta P$ in Step 1 implies that the output difference $\Delta L_{25}$ must be $\Delta L_{25} = C \oplus C' = (\Delta L_0 \parallel \Delta R_0)$. The value of $\Delta L_{25}$ and $\Delta R_{25}$ is shown in Figure 4. Therefore, no filtering process needs to be involved.
3. We now consider the totality of 96 key bits relevant to our attack, more precisely 64 bits of the subkey ($K_2^{(24)}, K_1^{(24)}$) used in the 25th round and 32 bits of the subkey $K_2^{(23)}$ used in the 24th round. Note that Step 3(b) is executed after Step 3(a) has been completed. For each plaintext pair selected in Step 1, we execute the following process:
   (a) For every guessed value of the 64-bit subkey $K_2^{(24)} = K_2^{(24)} \parallel K_3^{(24)}$ in round 25, decrypt the 25th round: $\Delta R_{24} = \Delta L_{25} \oplus (\Delta R_{25} \ll 10) \oplus K_2^{(24)}$ and $\Delta L_{24} = (\Delta L_{25} \oplus (\Delta R_{25} \ll 10) \oplus K_1^{(24)}) \gg 3 \oplus (\Delta R_{25} \oplus K_1^{(24)})$. If the output difference at round 24 ($\Delta L_{24}, \Delta R_{24}$) equals (0,...,0,1,...,1), as shown in Figure 4, increment the corresponding counter. Due to a large signal-to-noise ratio only a few plaintext pairs are sufficient to extract these 64 bits, see the discussion on data complexity below.
   (b) For every guessed value of the 32-bit subkey $K_3^{(23)}$, decrypt the 24th round: $\Delta R_{23} = \Delta L_{24} \oplus (\Delta R_{24} \ll 10) \oplus K_3^{(24)}$. If the 32-bit output difference of round 23 ($\Delta R_{23}$) equals (1,...,1), see also Figure 4, increment the corresponding counter.
4. Output the 96-bit subkey guessed $K_1^{(23)}, K_2^{(24)}$, and $K_3^{(23)}$ as the correct subkeys.

Data complexity. We utilize the signal-to-noise ratio $S/N = (2^b \times p)/(\alpha \times \beta)$, introduced by Biham and Shamir, to determine the cardinality of plaintext pairs, where $k$ is the number of key bits guessed in the
analysis, \( p \) is the probability of differential characteristic, \( \alpha \) is the average number of keys suggested by a pair and \( \beta \) is the ratio between the pairs that are not discarded and a total number of pairs. Since we are guessing 96 bits of the two subkeys and the probability of the differential characteristic is 1 (also implying that there is no additional filtering), then \( S/N = 2^{96} \times 1/1 \times 1 = 2^{96} \) (as \( \alpha = \beta = p = 1 \)). It also implies that only a few plaintext pairs are enough for a successful differential attack according to Biham and Shamir.\(^6\) Therefore, we select the number of plaintext pairs to be equal to \( 2^m = 4 \times p^{-1} = 4 \), which is also the data complexity of our attack.

**Time complexity.** The time complexity of the attack is clearly dominated by Step 3(a) that is executed first. In particular, Step 3(a) requires about \( 2 \times 4 \times 2^{64} = 2^{67} \) one-round decryptions. Note that in Step 3(b), 19 bits of the guessed 32 bits key are related to \( K^{(24)} \) (according to property 1), and therefore, these 96 bits of subkeys essentially give 77 subkey bits. Finally, we go through the exhaustive search over the remaining 51 bits. Therefore, the total time complexity is \( 2^{67} \times 1/25 + 2^{51} \approx 2^{62.4} \) full-round encryptions, where the factor 1/25 simply denotes that we perform one-round partial decryption.

**An efficient modification of ANU-II**

The experimental results confirm that the designers’ security evaluation of ANU-II against differential cryptanalysis is incorrect and the design is essentially flawed. The main reason for this is that suitable differences cancel out each other in the right branch (consequently, there is a differential characteristic that holds with probability 1). Therefore, we provide an improved variant of ANU-II given in Figure 5, which avoids this problem (modifying slightly the internal structure) and additionally has the best resistance to differential cryptanalysis without changing hardware and software implementation cost. The minimum number of active S-boxes of the proposed variant of ANU-II is listed in Table 4.

**The best differential characteristic of our modified variant**

We have also performed experiments towards finding the best differential characteristic of our modified version of ANU-II. For this purpose, we have built an MILP model, as described previously in the ‘Searching for the best differential characteristic’ section.

\[
\begin{align*}
x_3 - 3x_2 + x_1 - x_0 - 4y_3 - 3y_2 - 4y_1 - y_0 - p_{0.0} + 13p_{0.1} & \geq 0 \\
-x_3 - x_0 - y_2 - y_1 - p_{0.0} + 4p_{0.1} & \geq 0 \\
-7x_3 - 4x_2 + x_1 + 6x_0 + 3y_3 + 5y_2 - 2y_1 + 11p_{0.0} + p_{0.1} & \geq 0 \\
x_3 - 2x_2 + x_0 + 5y_3 + 2y_2 + 2y_1 + y_0 + 3p_{0.0} - 3p_{0.1} & \geq 0 \\
x_3 + 2x_1 + x_0 + 2y_2 + y_1 + y_0 + p_{0.0} - 4p_{0.1} & \geq 0 \\
x_3 + 3x_2 + x_0 + 2y_2 + 2y_1 + y_0 - p_{0.0} - 3p_{0.1} & \geq 0 \\
2x_3 + 2x_2 + x_1 + 2x_0 + y_2 + y_1 + y_0 - p_{0.0} - 3p_{0.1} & \geq 0 \\
5x_3 + x_2 - x_1 - 3x_0 - 3y_3 + 3y_2 + 2y_1 - 2y_0 + 2p_{0.0} + 3p_{0.1} & \geq 0 \\
x_2 - x_1 - 3x_0 + y_3 + y_2 - 2y_1 - 3y_0 - p_{0.0} + 7p_{0.1} & \geq 0 \\
-3x_2 - 2x_1 + 2x_0 - 2y_3 + y_2 - y_1 - 2y_0 + p_{0.0} + 7p_{0.1} & \geq 0 \\
x_1 + x_2 - x_1 + 4x_0 - 2y_3 - 2y_2 - 2y_1 - y_0 - 2p_{0.0} + 4p_{0.1} & \geq 0 \\
-2x_1 + 3x_2 + x_1 + 3x_0 - 2y_3 - y_2 + 4y_2 - 4y_0 - p_{0.0} + 5p_{0.1} & \geq 0 \\
-x_3 - x_0 + y_2 + y_1 - p_{0.0} + 2p_{0.1} & \geq 0 \\
4x_3 - 3x_2 + 2x_1 - 5x_0 + 3y_3 - y_2 + 3y_1 + y_0 + 7p_{0.0} & \geq 0 \\
-4x_3 + 3x_2 - 2x_1 + x_0 + 2y_3 - 3y_2 + 2y_1 - 4y_0 - p_{0.0} + 9p_{0.1} & \geq 0 \\
-3x_3 + 4x_2 - 3x_0 + y_3 - 2y_2 - 2y_1 + y_0 + 3p_{0.0} + 5p_{0.1} & \geq 0 \\
x_3 - 3x_2 - 3x_1 - 2x_0 - 2y_3 - 7y_2 + 7y_1 + y_0 + 4p_{0.0} + 11p_{0.1} & \geq 0 \\
x_3 - 2x_2 - 2x_1 - x_0 - y_3 - y_1 + y_0 - 2p_{0.0} + 8p_{0.1} & \geq 0 \\
\end{align*}
\]
SubBox. Our modified variant of ANU-II also uses eight 4-bit S-boxes. For each 4-bit S-box, there are 97 possible differential patterns that essentially give 97 possible 10-dimensional points when the probability information is embedded. SageMath is used to transform 97 points into 1033 inequalities. After applying the MILP-reduced algorithm, only 18 inequalities then remain. We again assume that $x_0$ and $y_0$ denote the LSBs of the input and output of the S-box, respectively. Similarly as before, we need to introduce two probability variables $(p_0, p_1)$ to encode the three possible differential patterns of the S-boxes of ANU-II. The first S-box can be described by a set of inequalities (16) and similar constraints can be obtained for the remaining seven S-boxes of ANU-II.

XOR and linear operation. Similar to the method described in the ‘Searching for the exact lower bounds of active S-boxes’ section.

An initial constraint

$$x_0 + \cdots + x_{63} \geq 1$$

Objective function.

$$\min \sum_{j=0}^{1} \left( \sum_{i=0}^{7} p_{i,0} + 2p_{i,1} \right)$$

At this point, all inequality constraints have been specified and our MILP model can search for the best differential characteristics. The experimental results are shown in Table 5, where clearly the situation is now quite different compared to the original design.

Conclusion

In this article, we have successfully applied the standard differential cryptanalysis to the full-round ANU-II block cipher. Due to the design flaw, a differential characteristic that holds with probability 1 could be identified for the entire cipher, which then leads to an efficient attack that uses a few plaintext pairs and $2^{62.4}$ full-round encryptions. It appears that the internal structure within the Feistel network is responsible for this security issue and we propose a simple modification of the internal structure (without changing hardware and software implementation cost), which efficiently protects the cipher from this kind of attacks. This is also illustrated by applying our MILP model to the modified variant of ANU-II where the best differential characteristic (for five-round reduced ANU-II) holds with much smaller probability than one.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: The work was supported in part by the National Natural Science Foundation of China (61872103), in part by the National Natural Science Foundation of China (62162016), in part by the Guangxi Natural Science Foundation (2019GXNSFGA245004), in part by the Scientific Research Project of Young Innovative Talent of Guangxi (Guike AD20238082) and the Guangxi Natural Science Foundation (2020GXNSFBA297076).

ORCID iD

Lingchen Li https://orcid.org/0000-0002-7490-9669

References

1. Wu CK. An overview on the security techniques and challenges of the internet of things. J Cryptol Res 2015; 2(1): 40–53 (in Chinese).
2. Bogdanov A, Knudsen LR, Leander G, et al. PRESENT: an ultra-lightweight block cipher. In: Proceedings of the cryptographic hardware and embedded systems (CHES), Vienna, 10–13 September 2007, pp.450–466. Berlin: Springer.
3. Wu W and Zhang L. LBlock: a lightweight block cipher. In: Proceedings of the 9th international conference on applied cryptography and network security (ACNS), Nerja, 7–10 June 2011, pp.327–344. Berlin: Springer.
4. Beaulieu R, Shors D, Smith J, et al. The SIMON and SPECK lightweight block ciphers. In: Proceedings of the

Table 4. Lower bounds on the number of active S-boxes of our modified variant of ANU-II.

| Rounds | # Active S-boxes |
|--------|-----------------|
|        | This article    |
| 1      | 0               |
| 2      | 1               |
| 3      | 2               |
| 4      | 4               |
| 5      | 6               |
| 6      | 9               |

Table 5. The six-round best differential characteristic of our variant of ANU-II.

| Rounds | Input difference | $\min \sum_{j=0}^{1} \left( \sum_{i=0}^{7} p_{i,0} + 2p_{i,1} \right)$ |
|--------|-----------------|---------------------------------------------------|
| 1      | 1020 0002 140a 0801 | 6                                                 |
| 2      | 6000 0800 1000 0182 | 10                                                |
| 3      | 0000 0002 6000 0000 | 12                                                |
| 4      | 0000 0000 0000 0002 | 12                                                |
| 5      | 0000 0002 0000 0800 | 14                                                |
| 6      | 6000 0800 0020 0182 | 18                                                |

Table 4. Lower bounds on the number of active S-boxes of our modified variant of ANU-II.

| Rounds | # Active S-boxes |
|--------|-----------------|
|        | This article    |
| 1      | 0               |
| 2      | 1               |
| 3      | 2               |
| 4      | 4               |
| 5      | 6               |
| 6      | 9               |
52nd ACM/EDAC/IEEE design automation conference (DAC), New York, 8–12 June 2015, pp.1–6. New York: IEEE.

5. Zhang W, Bao Z, Lin D, et al. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Inform Sci 2015; 58(12): 1–15.

6. Biham E and Shamir A. Differential cryptanalysis of DES-like cryptosystems. J Cryptol 1991; 4(1): 3–72.

7. Mouha N, Wang Q, Gu D, et al. Differential and linear cryptanalysis using mixed-integer linear programming. In: Proceedings of the 7th international conference on information security and cryptology (Inscrypt), Beijing, China, 3 December 2011, pp.57–76. Berlin: Springer.

8. Sun S, Hu L, Wang M, et al. Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of related-key differential and linear characteristics with predefined properties, https://eprint.iacr.org/2014/747 (2015, accessed 8 February 2015).

9. Fu K, Wang M, Guo Y, et al. MILP-based automatic search algorithms for differential and linear trails for SPECK. In: Proceedings of the fast software encryption (FSE), Bochum, 20–23 March 2016, pp.268–288. Berlin: Springer.

10. Abdelkhalek A, Sasaki Y, Todo Y, et al. MILP Modeling for (large) S-boxes to optimize probability of differential characteristics. IACR Trans Symmet Cryptol 2017; 2017(4): 99–129.

11. Sadeghi S, Mohammadi T and Bagheri N. Cryptanalysis of reduced round SKINNY block cipher. IACR Trans Symmet Cryptol 2018; 2018(3): 124–162.

12. Zhou C, Zhang W, Ding T, et al. Improving the MILP-based security evaluation algorithm against differential/linear cryptanalysis using a divide-and-conquer approach. IACR Trans Symmet Cryptol 2019; 2019(4): 438–469.

13. Liu F, Isobe T and Meier W. Automatic verification of differential characteristics: application to reducedmila. In: Proceedings of the annual international cryptology conference (CRYPTO), Santa Barbara, CA, 17–21 August 2020, pp.219–248. Cham: Springer.

14. Zong R, Dong X, Chen H, et al. Towards key-recovery-attack friendly differentiators: application to GIF-128. IACR Trans Symmet Cryptol 2021; 2021(1): 156–184.

15. Bansod G, Patil A, Sutar S, et al. ANU: an ultra lightweight cipher design for security in IoT. Secur Commun Netw 2016; 9(18): 5238–5251.

16. Sasaki Y. Related-key boomerang attacks on full ANU lightweight block cipher. In: Proceedings of the 16th international conference on applied cryptography and network security (ACNS), Leuven, 2–4 July 2018, pp.421–439. Cham: Springer.

17. Dahiphale V, Bansod G and Patil J. ANU-II: a fast and efficient lightweight encryption design for security in IoT. In: Proceedings of the international conference on big data, IoT and data science (BID), Pune, India, 20–22 December 2017. New York: IEEE.

18. Wang H, Wei Y and Liu W. Integral distinguisher search of ANU, ANU-II and LIg block ciphers. J Chin Comput Syst 2020; 41(7): 1470–1475 (in Chinese).

19. Sun S, Hu L, Wang P, et al. Automatic security evaluation and related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In: Proceedings of the 20th international conference on the theory and application of cryptology and information security (ASIACRYPT), Kaohsiung, Taiwan, 7–11 December 2014, pp.158–178. Berlin: Springer.

20. SageMath, http://www.sagemath.org/index.html

21. Sasaki Y and Todo Y. New algorithm for modeling S-box in MILP based differential and division trail search. In: Proceedings of the 10th international conference for information technology and communications (SecITC), Bucharest, 8–9 June 2017, pp.150–165. Cham: Springer.

22. Gurobi optimizer 9.1.2, http://www.gurobi.com

Appendix I

DDT of ANU-II S-box, as shown in Table 6.

| a | b | c | d | e | f |
|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 2 |
| 2 | 0 | 0 | 0 | 2 | 2 |
| 3 | 0 | 0 | 0 | 4 | 0 |
| 4 | 0 | 0 | 4 | 0 | 0 |
| 5 | 0 | 0 | 4 | 2 | 2 |
| 6 | 0 | 2 | 2 | 0 | 0 |
| 7 | 0 | 2 | 2 | 2 | 0 |
| 8 | 0 | 2 | 2 | 2 | 2 |
| 9 | 0 | 2 | 2 | 2 | 2 |
| a | 0 | 4 | 2 | 0 | 0 |
| b | 0 | 4 | 2 | 0 | 0 |
| c | 0 | 4 | 2 | 0 | 0 |
| d | 0 | 4 | 2 | 0 | 0 |
| e | 0 | 4 | 2 | 0 | 0 |
| f | 0 | 4 | 2 | 0 | 0 |