Cryptographic algorithm for protection of communication in drones control

Ivan Ivanov*
Department of Information Technology,
Faculty of Telecommunications and Management,
University of Telecommunications and Post,
Sofia 1700, Bulgaria
Email: i.ivanov@utp.bg
*Corresponding author

Krume Andreev, Stella Vetova and Rumen Arnaudov
Department of Radio Communications and Video Technologies,
Faculty of Telecommunications,
Technical University of Sofia,
Sofia 1000, Bulgaria
Email: andreev.k@abv.bg
Email: vetova.bas@gmail.com
Email: ra@tu-sofia.bg

Abstract: The paper below presents an optimised cryptographic algorithm, designed to protect communication in the management of drones. In this communication the need for the data to be protected as much as possible and (in case) stored reliably rises. The use of microcontrollers requires an optimised algorithm in terms of number and type of operations, due to lower computing power and a small amount of memory. The algorithm is based on DES algorithm, according to the Feistel scheme. It is a 64-bits symmetric block cryptographic algorithm, using a 512-bits cryptographic key.

Keywords: cryptography; scheme Feistel; cryptographic algorithm; drones; data protection.


Biographical notes: Ivan Ivanov is an Associate Professor in PhD at the Department of Information Technology, Faculty of Telecommunications and Management at University of Telecommunications and Post. His research is related to the applied cryptography, information and network security, cybersecurity, computer networks and protocols.

Krume Andreev received his MSc from the Department of Air Transport, Faculty of Transport, Technical University of Sofia, Bulgaria, in 2015. Since 2016, he is a PhD student at the Department of Radio Communications and Video Technologies, Faculty of Telecommunications at Technical University of Sofia. His research interests include unmanned aerial systems, flight control, auto-landing, security, regulations, and technical condition.

Stella Vetova received her MSc from the Department of Computer Systems, Faculty of Computer Systems and Technologies, Technical University of Sofia, Bulgaria. She has been a PhD student at the Department of Radio Communications and Video Technologies, Faculty of Telecommunications since 2017. Her research interests include content-based image retrieval, databases, artificial intelligence and computer networks.

Rumen Arnaudov is a Professor in PhD at the Department of Radio Communications and Video Technologies, Faculty of Telecommunications at Technical University of Sofia. His research activities are in the field of automated information and control systems, GPS/INS navigation systems, telecommunications measurements, intelligent sensors and sensor networks, technological measurements etc. He is an author of more than 180 scientific papers, patents and textbooks. He is also participates in over 70 scientific research projects and has 12 graduate PhD students. He is also a founder of the International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST).
1 Introduction

Nowadays, each kind of communication can be intercepted, unless measures are taken to protect it. This communication issue in the management of unmanned aerial vehicle systems (drones) is particularly relevant. They are vulnerable to malicious attacks such as eavesdropping, manipulation, interception, or attempted manipulation of their management (Shah et al., 2003; Zhou et al., 2006). Therefore, security is one of the most important safety requirements for the flight safety using the drones (Mincheva and Stanchev, 2018).

There are different approaches and algorithms used to protect communication such as AES, triple DES (Stallings, 2013), and others. However, there is a need to use an optimised algorithm requiring low computing power and a small amount of memory, as well as high cryptographic resilience to external attacks.

The algorithm proposed in this article (IDA-A) (Ivanov, Dikov, Arnaudov – Andreev) is an optimised version of the IDA algorithm (Ivanov et al., 2019; Alexiev et al., 2017; Ivanov, 2017). The IDA is designed on the DES algorithm and in accordance with Faistel’s scheme. It is a 64-bit, symmetric block cryptographic algorithm using a 256-bit cryptographic key. It consists of 16 internal cycles containing transpositions, substitutions and nonlinear procedures. The algorithm itself is a subject of long-term research [4 ÷ 6].

In contrast, IDA-A uses a 512-bit master key that increases protection from ‘brute force’ attacks to great extent. At the same time, this version reduces the total number of operations used to achieve higher performance and smaller program code for implementation in systems with less memory. This is especially suited for application with microcontroller configurations as used in the electron systems of the drones and subsequent selection of algorithm parameters (Iontchev, 2012).

2 Algorithm description

The algorithm works on 64-bit data blocks using a 512-bit encryption and decryption key. Bitwise and logical operations, as well as tabular permutations, are involved in data manipulation. The simplicity of operations allows the application of the algorithm on a very wide range of computing systems – embedded microcontrollers, general purpose processors and programmable logic devices.

The information stream is divided into blocks of 64-bit clear information (Figure 1). 64-bit blocks are fed into both the initial and the final displacement phases, and blocks of the same size are generated. Shuffling is a process of changing bit positions based on a table function, which is performed according to Table 1 and Table 2. The application of the table functions is as follows. The numbers stored in the tables show the sequence number of the bits of the input sequence and its corresponding position in the output sequence. In other words, the number 58 in Table 1 shows that in the first position in the output sequence will come 58-bits from the input, the second position will come 50th, the third 42nd, etc., the result is also 64-bit sequence. The table function is similarly applied to Table 2 but on 64-bit after the 16th cycle.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>‘Initial transpositions’</th>
</tr>
</thead>
<tbody>
<tr>
<td>58</td>
<td>50</td>
</tr>
<tr>
<td>60</td>
<td>52</td>
</tr>
<tr>
<td>62</td>
<td>54</td>
</tr>
<tr>
<td>64</td>
<td>56</td>
</tr>
<tr>
<td>57</td>
<td>59</td>
</tr>
<tr>
<td>61</td>
<td>63</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>‘Final transpositions’</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>8</td>
</tr>
<tr>
<td>39</td>
<td>7</td>
</tr>
<tr>
<td>38</td>
<td>6</td>
</tr>
<tr>
<td>37</td>
<td>5</td>
</tr>
<tr>
<td>36</td>
<td>4</td>
</tr>
<tr>
<td>35</td>
<td>3</td>
</tr>
<tr>
<td>34</td>
<td>2</td>
</tr>
<tr>
<td>33</td>
<td>1</td>
</tr>
</tbody>
</table>

The obtained bit sequence is divided into two sequences – left L(0) and right R(0), each containing 32-bits. Then, the encryption process is performed using the F(.) function. In this case, this function only occurs on the right side. The output 32-bit sequence of functions F(.) is fed to the first input of an XOR adder. At the second input of this adder is 32-bit L(0). The output 32-bits from this add-on are fed to the one input of the next XOR adder. At its second input, the 32-bit K2 key is fed. Then, the sequences of the output of the XOR sumator and R(0) change their positions, i.e., the left ones become right and vice versa, in turn input sequences for the next cycle. These procedures are performed 16 times, based on internal cycles.

The implementation of the F(.) (Schneier, 2013; Ferguson et al., 2010; Hoffstein et al., 2014) function is performed as follows. First, the right part Ri-1 of the information block extends from 32- to 48-bits by repeating the bits with numbers 1, 4, 5, 8, 9, 12, ..., 24, 28, 32 (the boundary bits of the groups of four consecutive bits). The new bits of the extension join cyclically to the neighbouring 8-bit structures of 4-bits under the scheme:

r32r12r2r3r4r5, r4r5r6r7r8r9, ...., r28r29r30r31r32r1
Then, the 48-bit length $R_l-1$ shaped is multiplied by modality 2 with the 48-bit key $K_l$ and divided into eight consecutive 6-bit structures. These structures are fed to S-boxes performing the nonlinear procedure – selection of four outputs (bits) of six inputs (bits). The transposition $P$ calculated according to a given scheme and the final result of cryptographic processing $F(.)$ with a length of 32-bits are presented.

The 32 key $\{K_j\}$ of 48-bits and 32-bits is executed according to the general algorithm shown in the right part of Figure 1. The key $\{K_j\}$ is entered into the cryptographic scheme with a length of 512-bits.

The keys involved in each cycle are generated from the main one in the following manner. The required number of bits for the sub-keys are taken as follows: the first 48-bits of the master key are taken as a $K_1$ key and fed to the input of the $F(L_0, K_1)$ function. The next 32-bits are taken as $K_2$ and fed to the input of the second XOR adder on the first cycle, then the same pair of keys for the second cycle is generated, and so on until the 16th cycle. After reaching the last key of the master key, a linear 25-bit offset is performed in the 512-bit sequence as many times as necessary to obtain the required number of sub-keys.

On the basis of the proposed scheme of Figure 1, the cryptographic stability of encrypted data is increased.
Cryptographic algorithm for protection of communication in drones control

many times. This is due to the use of a 512-bit master key and the use of the 32 sub-keys for the 16 internal cycles.

2.1 Experimental part

To determine the cryptographic stability and the proper functioning of the algorithm scheme, it is necessary to do the following: determining the number of internal cycles, examining the avalanche property and determining the theoretically the numerical stability of the algorithm.

3 Number of internal cycles

The greater the number of internal cycles, the more difficult the cryptanalysis of the cipher, even with a relatively weak F function. Generally, the number of cycles should be chosen so that for all known crypto analysis methods it takes more time than the analysis using all possible keys. The recommended number of cycles is between 10 and 16. For this algorithm we have chosen the maximum number of cycles, namely 16.

4 Presentation of the ‘avalanche effect’
(Schneier, 2013)

Algorithms for encryption must have a high sensitivity to the result of changing the start data – any slight change to the open text or key must result in a significant change to the encrypted text.

In particular, changing each bit from the open text or the key should result in a change in the meaning of a large amount of bits from the encrypted text. If the changes in the encoded text are small, it can result in a significant reduction of the set of keys or the area of the text.

5 Investigation of the ‘avalanche effect’ property in the IDA-A algorithm

In order to investigate the IDA-A algorithm for the ‘avalanche effect’ property, two different open texts have to be encoded, differing in 1-bit only:

\[ P = 00000000 00000000 00000000 00000000 \]
\[ 00000000 00000000 00000000 00000000 \]
\[ P = 10000000 00000000 00000000 00000000 \]
\[ 00000000 00000000 00000000 00000000 \]

and also, with the same key:

\[ K = 11110101 11101110 11110000 11100101 11110100 \]
\[ 11110101 11101110 11110000 11100100 11110101 \]
\[ 11100000 00100000 11100100 11100100 11110101 \]
\[ 11100101 11101000 11101000 11100000 11101001 \]
\[ 11101000 11101110 11101011 11101000 11110000 \]
\[ 11110000 11110010 11110011 11101000 11110010 \]
\[ 11110000 11110010 11110011 11101000 11110010 \]
\[ 11110000 11110010 11110011 11101000 11110010 \]
\[ 11110000 11110010 11110011 11101000 11110010 \]

A similar study is also made for the case when an open text is encoded:

\[ P = 11110010 11100101 11101011 11110010 \]
\[ 11110100 11101110 11110101 11110100 \]

and two keys differing from one another:

\[ K = 11110101 11101110 11110000 11100101 11110100 \]
\[ 11110101 11101110 11110000 11100100 11110101 \]
\[ 11100000 00100000 11100100 11100100 11110101 \]
\[ 11100101 11101000 11101000 11100000 11101001 \]
\[ 11101000 11101110 11101011 11101000 11110010 \]
\[ 11110000 11110010 11110011 11101000 11110010 \]
\[ 11110000 11110010 11110011 11101000 11110010 \]

5.1 Evaluation of results

The results of the studies are shown in Table 3.

Table 3: Study results of the IDA-A algorithm

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Difference in bits</th>
<th>Cycle</th>
<th>Difference in bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>37</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>39</td>
<td>4</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>5</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>32</td>
<td>6</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>31</td>
<td>7</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>29</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>41</td>
<td>9</td>
<td>38</td>
</tr>
<tr>
<td>10</td>
<td>39</td>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>11</td>
<td>32</td>
<td>11</td>
<td>33</td>
</tr>
<tr>
<td>12</td>
<td>30</td>
<td>12</td>
<td>29</td>
</tr>
<tr>
<td>13</td>
<td>30</td>
<td>13</td>
<td>26</td>
</tr>
<tr>
<td>14</td>
<td>29</td>
<td>14</td>
<td>30</td>
</tr>
<tr>
<td>15</td>
<td>36</td>
<td>15</td>
<td>35</td>
</tr>
</tbody>
</table>

The graphical representation of the results is shown in Figure 2.
As Figure 2 and Table 3 show, the IDA-A algorithm performs a strong ‘avalanche effect’. It shows that after the third encryption cycle there is a difference of 37-bits. At the end of encryption the difference is observed in 36 positions (bits).

As in the first case, the ‘avalanche effect’ is also strong when encrypting the obvious information using one-to-one keys (Figure 2 and Table 3). It can be seen that after the third encryption cycle there is a difference of 30-bits. At the end of the encryption the difference is observed in 35 positions (bits).

5.2 Results of comparison with existing algorithms

Objective evaluation of the results is obtained after examining the ‘avalanche effect’ property for several of the known cryptographic algorithms, namely: Vigenere, DES and AES, under the same conditions as described above, and compared to the obtained ones. The results are shown in Table 4. The percent calculation of the avalanche effect is as follows:

$$Avalanche\ effect = \frac{Number\ of\ changed\ bits}{The\ number\ of\ input\ bits} \times 100\%$$

<table>
<thead>
<tr>
<th>Encryption algorithm</th>
<th>Number of changed bits</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vigenere</td>
<td>2</td>
<td>3.13</td>
</tr>
<tr>
<td>DES</td>
<td>35</td>
<td>54.68</td>
</tr>
<tr>
<td>AES</td>
<td>19</td>
<td>28.71</td>
</tr>
<tr>
<td>IDA-A</td>
<td>37</td>
<td>57.81</td>
</tr>
</tbody>
</table>

As can be seen from Table 4, the IDA-A algorithm has the strongest avalanche effect, with the change of one bit at the input leading to a change of 57.81% of the output bits.

6 Determination of cryptographic stability by means of theoretically digital resistance

There is another assessment of the robustness of cryptographic protection, which has a quantitative expression – theoretical digital robustness. This assessment is of primary importance for the theoretical evaluation of the cryptographic algorithm, but can also be used by the user for guidance on the quality (level) of the cryptographic algorithm (cryptographic equipment) (Ferguson et al., 2010; Hoffstein et al., 2014).

The theoretical digital robustness is an estimate provided only when the cryptographic algorithm is qualitative and cannot be overcome by accelerated crypto-analysis attacks, but only through ‘brute force’ methods.

The evaluation of the theoretical digital robustness is made in the presence of the following important prerequisites (circumstances) (Martin, 2012; Hoffstein et al., 2014), which are very favourable for the crypto-analytical activities performed by the deceiver (the opponent):

- It is assumed that cryptographic analysis has complete information about the cryptographic algorithm and its particular implementation, including cryptographic equipment operating in accordance with this algorithm and the required documentation of the cryptographic algorithm. The reality of this premise is substantiated and justified given the possibilities of the intelligence agency.
• The crypto-analysis of the crypto-algorithm has been prepared and tested. This means a built-up technical base, a specific software program, and a plan to attack the crypto-algorithm.

• There is a record of an information array of encrypted information of sufficient volume and additional general information about the communications destination where encrypted information (potentially possible subscribers, supposed information exchange objectives, etc.) is captured (recorded).

• There is a brief sample of clear information from the saved encrypted dataset. Providing such a sample of clear information is possible. This can be done by individuals or by using the standard parts of the information in document exchange.

In fact, the malicious person (the opponent) has no information about the particular interaction key that was used in encryption. Therefore, crypto-analysis is limited to searching for the key used by the brute force method (Schneier, 2013), that is, using all of the interaction keys.

As far as the crypto-analysis of the brute force is performed by successively testing the interaction keys from the N-key size \{K_i\}, then the following formula can be used to assess the robustness of the cryptographic algorithms: for the mean time revealing the interaction key:

\[
\overline{T_{jcn}} = \frac{1}{2} N T_{ap}
\]

where \( T_p \) is the time to run a sample to reveal an interaction key.

\[
T_{ap} = \frac{n_{ap}}{B} S_{min}
\]

where \( n_{ap} \) is the amount of 1-bit machining operations, \( S_{min} \) [bit] – the minimum sample length, and \( B \) [mas. oper./s] is the performance of the processing in the crypto-analysis technical base.

Substitution of equation (2) in equation (1) is obtained:

\[
\overline{T_{jcn}} = \frac{1}{2} N n_{ap} S_{min} \frac{1}{B}, [s]
\]

Since this time is a very large dimension for modern cryptographic algorithms, it is more convenient to define it in years. Then the result is:

\[
\overline{T_{jcn}} = N n_{ap} S_{min} 10^{-7} \text{[години]}\]

\[
\overline{T_{jcn}} = N n_{ap} S_{min} 10^{-7} \text{[години]}\] (4)

Figure 3  Diagram of dependence for different algorithms and \( n_{ap} S_{min} = 10^8 \) (see online version for colours)
Figure 3 shows the dependence of the theoretical digital resistance, the amount of interaction keys $N$ at $n_{op} S_{\text{min}} = 10^8$ and different processing speeds $B = 10^8 \text{ op/s}; 10^{10} \text{ op/s}; 10^{15} \text{ op/s}; 10^{20} \text{ op/s}; 10^{30} \text{ op/s}$.

The amount of interaction keys $N$ are determined by the formula:

$$N = 2^k, \lceil \delta p \rceil$$ \hspace{1cm} (5)

where $k$ is the length of the key used by the cryptographic algorithm.

Based on formula (4), the dependence of the theoretical digital resistance, the amount of interaction keys at $n_{op} S_{\text{min}} = 10^2, 10^3$ and $10^4$, and the different processing speeds $B = 10^8 \text{ op/s}; 10^{12} \text{ op/s}; 10^{15} \text{ op/s}; 10^{20} \text{ op/s}; 10^{30} \text{ op/s}$ is obtained.

The graph shows that the theoretical digital robustness is reduced by increasing the processing speed in the cryptographic analysis base.

For the IDA-A algorithm, for $K = 512$-bits and the computing power of modern computers ($4 \div 5 \text{ GHz processors}$), the theoretical digital resistance is extremely high, for values of $B = 10^{20} \text{ op/s}$ and $B = 10^{30} \text{ op/s}$, it is: $T_{\text{ycm}} = 2.23 \times 10^{34}$ and $T_{\text{ycm}} = 2.23 \times 10^{24}$ years, which is a huge amount of time compared to human life.

For cryptographic algorithms where $K \geq 128$-bit the cryptographic resistance is guaranteed.

7 Conclusions

The proposed algorithm for communication protection in systems with microcontroller configuration is applicable in the management of drones. The algorithm has an enhanced level of protection because it is a 64-bit, symmetric block cryptographic algorithm using a 512-bit cryptographic key. It consists of 16 internal cycles containing transpositions, substitutions and nonlinear procedures. The data can be exchanged in real-time as well as reliably stored in a suitable buffer (memory) and delivered only to the intended recipient as intended. This algorithm is a good basis for implementation in a system of automated flights with over-protection.

On the base of the studies made, the following results were obtained:

- For the IDA-A algorithm, for the case of two open texts which differ in one bit and one key, after the third encryption cycle an average difference of 35-bits out of a total of 64-bits is observed for cycles from 3 to 15.

- For the IDA-A algorithm, for the case of one open text and two keys which differ in one bit, after the third encryption cycle an average difference of 33-bits out of total 64-bits is observed for cycles from 3 to 15.

- The IDA-A algorithm has the strongest avalanche effect compared to the cryptographic algorithms Vigenere, DES and AES.

- For the IDA-A algorithm, for $K = 512$-bits and the computing power of modern computers ($4 \div 5 \text{ GHz processors}$), the theoretical digital resistance is extremely high, for values of $B = 10^{20} \text{ op/s}$ and $B = 10^{30} \text{ op/s}$, it is: $T_{\text{ycm}} = 2.23 \times 10^{34}$ and $T_{\text{ycm}} = 2.23 \times 10^{24}$ years, which is a huge amount of time compared to human life.

References


