Dataset of Cryptographic Algorithms for UAV Image Encryption based on Artificial Neural Networks

Sergiy Gnatyuk[1], Andrew Okhrimenko[2], Denys Navrotskyi[1], Dmytro Proskurin[1], and Bohdan Horbakha[1]

[1] National Aviation University, 1 Liubomyra Huzara ave, Kyiv, 03058, Ukraine
[2] Mariupol State University, 6 Preobrazhenska str., Kyiv, 03037, Ukraine

Abstract

Equipped with the latest image processing systems, sensors, and high-resolution cameras, they can conduct real-time aerial photography, monitor enemy activity, and gather critical intelligence without putting the military at risk. UAVs make it possible to conduct long-term operations in conditions of secrecy, providing commanders with valuable information for making strategic decisions. However, the issue of ensuring the confidentiality of critical data (images) collected using UAVs remains unresolved. In this paper universal dataset of cryptographic algorithms is proposed and it uses a neural network model to select the optimal encryption algorithm. To form such a dataset, it was necessary to evaluate the speed and security of the cryptographic algorithms as well as other important parameters. The developed dataset in synthesis with a neural network model can be used to select the optimal crypto algorithm. In further research, the authors plan to determine the criteria for using the generated dataset by neural networks and develop a knowledge base for neural network training.

Keywords

UAV, image, security, confidentiality, cryptography, cryptographic algorithm, encryption, data transfer, surveillance, neural network, dataset.

1. Introduction

In the modern world, unmanned aerial vehicles (UAV) play a key role in many areas of human activity – from civil services and logistics to reconnaissance and warfare [1]. However, such use of UAVs requires data (images) security, that is transmitting from the onboard computer and input devices to the control point of the UAV. In particular, it is important to ensure the confidentiality of such images as a basic cybersecurity feature [2].

To provide a high level of data confidentiality cryptographic algorithms can be used. There are many advanced directions and technologies in cryptography such as post-quantum cryptography, quantum key distribution and quantum secure direct communication, lightweight cryptography, etc.

Many of these cryptographic methods and protocols can be used in UAV-based systems for image encryption. However, its usage depends on many factors and parameters that should be analyzed for the development of the universal cryptosystem.

2. Literature Review

In [3], an analysis of crypto algorithms was carried out according to several important criteria. The results of the analysis showed that each cryptographic algorithm has advantages and disadvantages—there is no universal crypto algorithm capable of solving all privacy problems in UAVs. Given the limited resources in the process of UAV operation, there is a need to create a universal set of data—the so-called dataset (library) of cryptographic algorithms, which would be able to solve various problems in constantly changing conditions. In addition, a relevant and innovative approach today is the use

CQPC-2023: Classic, Quantum, and Post-Quantum Cryptography, August 1, 2023, Kyiv, Ukraine
EMAIL: s.gnatyuk@nau.edu.ua (S. Gnatyuk); andrew.okhrimenko@gmail.com (A. Okhrimenko); d.navrotskyi@nau.edu.ua (D. Navrotskyi); dmytro.proskurin@gmail.com (D. Proskurin); jmorosr2@gmail.com (B. Horbakha)
ORCID: 0000-0003-4992-0564 (S. Gnatyuk); 0000-0001-8270-2863 (A. Okhrimenko); 0000-0003-3160-3480 (D. Navrotskyi); 0000-0002-2835-4279 (D. Proskurin); 0000-0003-0713-4426 (B. Horbakha)

of artificial intelligence methods (in particular, neural networks) to select the optimal cryptographic algorithm from a dataset according to certain parameters [4–6], as well as other variations of the synthesis of artificial intelligence models and cryptographic methods [7–9].

According to this, *the work aim*s to ensure the confidentiality of data during transmission from UAVs due to the use of neural networks and the creation of a universal dataset of modern cryptographic algorithms [10, 11].

For the effective formation of the dataset, it is also necessary to evaluate the speed of crypto-algorithms, their crypto-resistance, etc. In the future, the developed dataset can be used in combination with a neural network to select the optimal encryption algorithm depending on the operating conditions of the UAV.

3. Research Results

3.1. Description of Selected Encryption Algorithms

To ensure the protection of information in modern information and communication systems, a large number of crypto algorithms are presented, which differ in their basic characteristics (parameters) - cryptoresistance (to various known methods of cryptanalysis), speed of cryptographic data processing, convenience of software/hardware implementation, etc. Given the hardware limitations of modern UAVs, the speed factor and the amount of resources needed to encrypt information become key characteristics when choosing an algorithm. Among the algorithms that, according to the authors, should be represented in the dataset, the following symmetric stream and block ciphers were chosen: Salsa20; PANAMA; HC-256; AES; DES; Triple DES; Serpent; Blowfish; Twofish; MARS; RC2; RC6; GOST 28147-89 (DSTU GOST 28147: 2009); Kalyna.

3.1.1 Salsa20

A crypto algorithm designed by D. Bernstein and presented at the eSTREAM competition, the purpose of which was to create European standards for data encryption transmitted in postal systems. The algorithm became the winner of the competition in the first category (stream ciphers for high-bandwidth software applications). For implementation, it is necessary to create a key with a length of 128 or 256 bits, as well as an initialization vector with a length of 64 bits. The algorithm uses the following operations:

- Addition of 32-bit numbers.
- Exclusive OR (XOR).
- Bits shift.

The basis of this algorithm is a 64-byte hash function that works together with a counter and is 20 cycles performed on the internal state. The disadvantage of Salsa20 is that it can only be used to protect personal data stored on a PC or hard disk. Since the integrity of the encrypted text is not checked, therefore, for more important data, it is necessary to use authenticated encryption [12].

3.1.2 PANAMA

The main transformations of the PANAMA cipher operate on 32-bit words. This algorithm can be used for hashing large information arrays. When used as a stream cipher and generator of pseudorandom sequences, the algorithm has a rather long initialization procedure. The field of use of the algorithm is the encryption of video information, for example, in the field of pay TV - in this field, where the intensity of the data flow is very high and a high-performance processor is used for its processing, an algorithm that uses the already excessively loaded processor to the smallest extent is needed. The algorithm itself is based on a 544-bit state register and an 8192-bit buffer register. The state of the state register is updated using parallel nonlinear transformations. The buffer register is an LFSR, which is similar to the register used in the SHA hashing algorithm. The state register consists of seventeen 32-bit words. The state of the registers can be changed using two iterations:

- Push iteration accepts input data but does not generate output data.
- Pull iteration does not accept input data but generates output.

There is also a Blank Pull iteration, which is similar to the Pull iteration, but the output data is discarded [13].

3.1.3 HC-256

HC-256 is a stream encryption algorithm developed by Wu Hongjun, a cryptographer from the Singapore Institute of Information and Communications Research and first published in 2004. A 128-bit version of the cipher was presented at the aforementioned eSTREAM competition. The algorithm became one of the four finalists of the competition in one of the

categories. This algorithm generates a 2128-bit key sequence using a 256-bit key and a 256-bit initialization vector. The cipher contains two secret tables, each of which has 1024 32-bit elements. At each step, one element from the table is updated using a nonlinear feedback function, and after every 2048 steps, all elements of the two tables will be updated. Such operations as bitwise exclusive OR, concatenation, shift to the left/right, and cyclic shift to the right are also used [14].

3.1.4 AES (Rijndael)

A symmetric block encryption algorithm, a finalist (winner) of the AES competition and adopted as an American encryption standard by the US government [15]. The algorithm itself replaced the previous encryption standard - DES [16]. The AES block size has a fixed length of 128 bits and the key size can be 128/192/256 bits. Data is represented by 8 bytes. The algorithm itself includes the following operations: SubBytes (substitution operation, every 8 bytes are replaced according to the substitution table), ShiftRows (shifting the elements of the square), MixColumns (multiplication by a polynomial modulo), AddRoundKey (bitwise addition of data with a round key by module 2 (XOR)), key extension. AES is a fairly fast encryption algorithm, which makes it possible to consider it a worthy candidate for use in modern information and communication systems, particularly in UAV systems [15].

3.1.5 DES

A symmetric encryption algorithm that was the US encryption standard from 1976 to the end of the 1990s and over time gained international use in various countries around the world. Even from the time of its development, the algorithm caused mixed reviews, as it contained classified elements of its structure - there were fears about the possibility of control by the US National Security Agency. The algorithm encrypts data in blocks of 64 bits, and the key length is 56 bits. The following operations are used in the encryption process: shuffling, substitution, XOR, key expansion, and cyclic shift. Currently, DES is considered unreliable mainly due to the small key length (56 bits) and block size (64 bits) [16], but its use in the created dataset is due to experimental research to compare it with other algorithms. In 1999, the DES key was publicly cracked at 10 p.m. 15 min. and also proved that DES is not resistant to linear cryptanalysis. The algorithm is believed to be robust enough to be used in a 3-DES modification, although theoretical attacks have been developed.

3.1.6 Triple DES (3DES)

A block symmetric cipher that applies the mentioned DES algorithm three times to each block of data. It was created in 1978 based on the DES algorithm to eliminate the main drawback of the latter - the short length of the key (56 bits), which can be broken (chosen) by the sorting method. The 3DES algorithm works 3 times slower than DES, but its crypto resistance is much greater – the time required for 3DES cryptanalysis is 109 times greater than for DES. The length of the key is 192 bits, but it is 168 bits long, because as in DES, in which a 64-bit key is divided into 8 bytes, only 7 bits are used in each byte, so the key length is 56 bits. Similar to DES, the encryption process uses operations such as shuffling, substitution, XOR, key expansion, and cyclic shift. The main advantage of the algorithm is its high crypto resistance, however, it has a low speed of data encryption [17].

3.1.7 RC2

The algorithm was developed in the late 1980s and is the property of RSA Data Security. The development of the algorithm was initiated and partially sponsored by Lotus, which needed a robust encryption algorithm for use in the Lotus Notes system. The stability of the algorithm was checked by the US National Security Agency, certain recommendations were also formulated and implemented by the developers. Encryption takes place in blocks of 64 bits using keys of variable size: from 8 to 1024 bits inclusive (the recommended key size is 64 bits). The RC2 algorithm is a Feistel network, in which 18 rounds of transformations are performed, which are divided into 2 types:

- Mixed rounds *(mix)*.
- Meshed rounds *(mesh)*.

Also, the following operations are used in the algorithm: bitwise logical operation AND, bitwise complement to x, cyclic shift to the left by the number of bits determined by the substitution table, and key expansion procedure. According to studies of the impact of differential and linear cryptanalysis on the algorithm, the following result was obtained: the algorithm is not vulnerable to an attack by the method of linear cryptanalysis, but it can theoretically be revealed by the method of differential cryptanalysis [18].

3.1.8 Serpent

Block encryption algorithm, which was one of the finalists of the second stage of the AES competition. The size of the block is 128 bits, the length of the key can be 128/192/256 bits, and the algorithm itself has 32 rounds (16 were initially planned, but to counter unknown methods of cryptanalysis, it was increased to 32). Serpent is an SP network, meaning it has permutation tables as well as permutation tables. The algorithm has key expansion operations, linear transformations, as well as inverse linear transformations (for decryption). During the development and analysis of the Serpent algorithm, no vulnerabilities were found in the full 32-round version (as well as in the other finalist algorithms). According to the authors of the algorithm, a new mathematical theory is needed to break the cipher [19].

3.1.9 Blowfish

A block symmetric algorithm developed by B. Schneier in 1993, is not patented and freely distributed. The length of the key can be from 32 to 448 bits, and the length of the block is 32 bits. The algorithm is a Feistel network and, accordingly, includes such operations as XOR, substitution tables, and addition [20]. Regarding the crypto-resistance of the algorithm, it is possible to note the developed attack that made it possible to break the 3-iteration Blowfish – it is based on the fact that the addition modulo 232 and XOR operations are not commutative. Successful attacks are only possible due to implementation errors. That is why Blowfish has proven itself as a reliable algorithm, it is used, in particular, in SSH (transport layer), PuTTY (transport layer), OpenVPN, etc. [21].

3.1.10 Twofish

A symmetric block encryption algorithm developed by a group of specialists led by B. Schneier, who was also (like Serpent) among the five finalists of the second stage of the AES competition. The algorithm is developed based on Blowfish, SAFER, and Square, the block size is 128 bits, and the key length is 256 bits with the number of rounds being 16. One of its features is permutation tables, which are formed depending on the key. The algorithm itself was implemented as a mixed Feistel network with 4 branches that modify each other using Hadamard cryptotransformations. The algorithm includes such functions as a cyclic shift by 1 bit, as well as a whitening function. The study of Twofish with a reduced number of rounds showed that the algorithm has a large margin of stability, and compared to the other finalists of the AES competition, it turned out to be the most stable. However, its unusual structure and relative complexity raised some doubts about the quality of this stability—this is exactly what played against it at the AES competition [22].

3.1.11 MARS

A block symmetric algorithm developed by IBM Corporation. According to the results of the AES competition, MARS also reached the finals but lost to Rijndael. MARS is currently distributed under a royalty-free license. The block size in the algorithm is 128 bits, and the key size can be from 128 to 448 bits (must be a multiple of 32 bits). In the encryption process, the algorithm uses the following operations: addition/subtraction, exclusive OR, substitution tables, fixed cyclic shift, and data-dependent cyclic shift, multiplication modulo 2^{32}, key expansion. According to IBM, the company's 25-year cryptanalytic experience is invested in the MARS algorithm and, along with high cryptographic stability, the cipher allows effective implementation even within such limited frameworks as is characteristic of smart cards, which allows it to be used in UAVs. From the point of view of cryptanalysis, there are currently no effective attacks on this algorithm, but it has several weaknesses, in particular:

- Subkeys with a large number of repeated zeros or ones can lead to effective attacks on MARS since weak subkeys will be generated based on them.
- The two least significant bits used in multiplication are always equal to one, that is, two input bits are unchanged during the process of multiplication by the key, as well as two output bits that are independent of the key [23].

3.1.12 RC6

Developed by RSA Data Security, RC6 is an evolutionary improvement over its predecessor, RC5. It was designed specifically for the AES competition and was one of the finalists. The algorithm operates on data blocks of 128 bits and supports key sizes of 128, 192, and 256 bits. RC6 is a parameterized algorithm where the block size, key size, and number of rounds are adjustable. The primary innovation of RC6 is the introduction of integer multiplication as an additional operation. The algorithm includes operations like

data-dependent rotations, modular addition, and bitwise XOR. The use of multiplication aims to provide additional diffusion over RC5 and to confound linear and differential cryptanalysis [24]. The structure of RC6 makes it suitable for hardware implementation and parallel processing, which can be advantageous in applications requiring high-speed encryption.

3.1.13 GOST 28147-89

GOST 28147-89 is a Soviet and Russian government standard symmetric key block cipher. Developed in the 1970s, the standard had been marked "Top Secret" and then downgraded to "Secret" in 1990. It was a Soviet alternative to the United States' DES. The algorithm operates on 64-bit data blocks with a key length of 256 bits. It's a Feistel network of 32 rounds. The key schedule is simple, and the S-boxes (substitution boxes) were classified, with a few different sets known to exist. In the context of the standard, the algorithm was usually called Magma. The cipher has been used in various Russian state standards (STBs) and is used in some Russian cryptographic systems [25]. The algorithm is considered secure, but it's less studied than other contemporary algorithms.

3.1.14 Kalyna (DSTU 7624:2014)

Kalyna is a symmetric block cipher that was selected as the new encryption standard of Ukraine. The algorithm was designed to have a high level of resistance against known cryptanalytic attacks and to achieve a high-speed performance on contemporary processors. Kalyna supports block sizes of 128, 256, or 512 bits and key sizes that can be either equal to or double the block size. The cipher's structure is based on a substitution-permutation network (SPN) and includes operations like SubBytes, ShiftRows, MixColumns, and AddRoundKey, similar to AES. However, Kalyna uses different S-boxes and has a more complex key schedule. The development of Kalyna was part of a larger effort to develop cryptographic standards in Ukraine that would be free from foreign patents and would meet the modern international cryptographic strength criteria [26, 27]. The size of the block and the number of rounds are dependent on the size of the key (see Table 1).

Table 1
Kalyna algorithm characteristics

Block length	Key length	Number of rounds
128	128	10
	256	14
256	256	14
	512	18
512	512	18

The algorithm itself uses transformation operations, substitution, and permutation tables, key expansion, addition modulo 2, linear transformations, and pre- and post-whitening [28]. Studies of the algorithm's cryptographic strength have shown only a few effective attacks on truncated versions of the algorithm, but they are not practical.

3.2. Neural network model for determination of the crypto algorithm

Artificial neural networks (ANNs) are considered tools that can help analyze causal relationships in complex systems [24]. That is why it was decided to analyze exactly this approach for choosing a cryptographic algorithm from the created dataset. A typical algorithm of ANN operation is shown in Fig. 1.

Consider the potential of training a deep neural network (DNN) with multiple hidden layers, akin to the human neural system. The efficacy of the prior generation of neural networks is primarily confined to SNNs with one or two hidden layers. This is because training DNNs poses challenges, often resulting in a final accuracy that is inferior to SNNs [25]. The main challenges in training DNNs include the vanishing gradient problem as the number of hidden layers increases and the pitfalls of local minima. Structurally, a DNN resembles a shallow neural network but boasts more hidden layers and a pronounced hierarchical structure. One can view DNNs as advancements over SNNs. Recent advancements in machine learning have rendered DNNs trainable, introducing tools like pre-training with RBM and SAE for smaller datasets. Furthermore, the nature of certain tasks underscores the viability of employing DNNs with limited datasets. For instance, DNNs used in image recognition typically require more than 10^4 input variables (a 100×100 pixel image demands 10^4 input variables), necessitating the definition of a vast

number of parameters (often exceeding 10^6). Thus, while extensive datasets are preferable, DNNs for specific tasks might only need around 100 input variables, and fewer parameters, making compact DNNs (with fewer hidden layers and neurons) adequate [29].

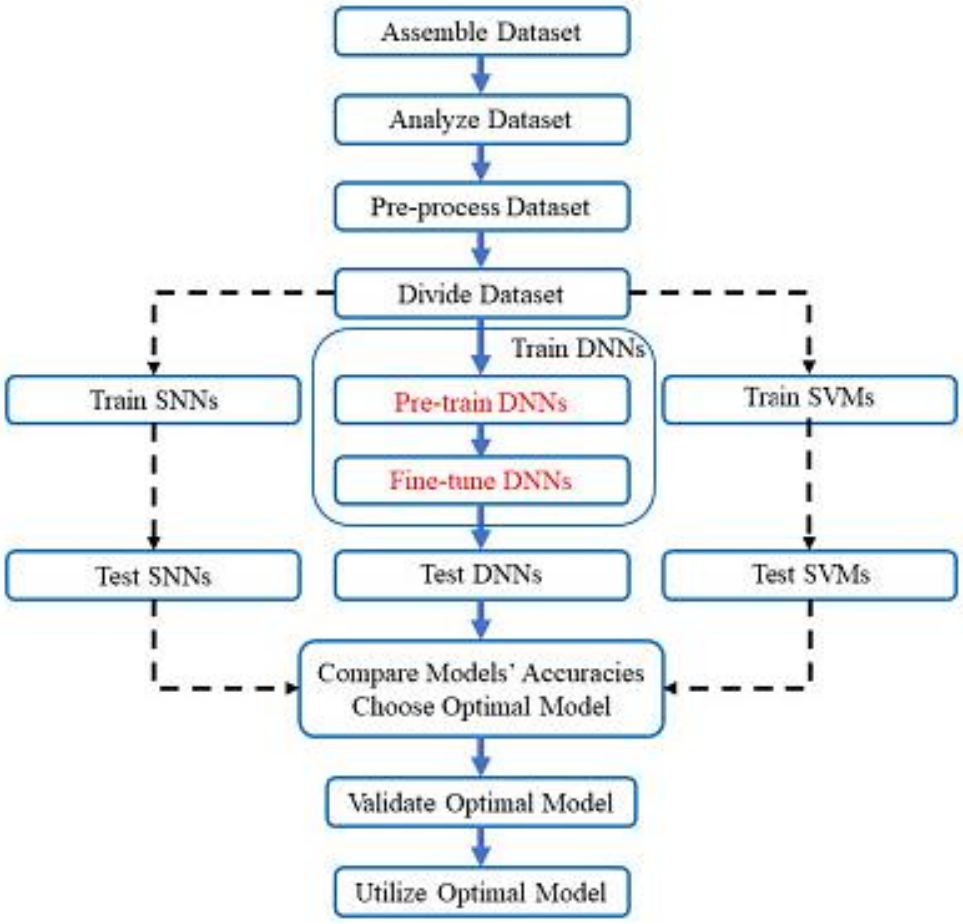

Figure 1: ANN algorithm

The advantages of employing DNNs with limited datasets for cryptographic algorithm determination are evident. Large regression or classification challenges, previously addressed using conventional machine learning techniques (like SNN, SVM, etc.) with small datasets, can now be tackled using DNNs, yielding superior accuracy and enhanced generalization [30, 31]. This research employs SCS prediction as a case study to demonstrate that SAE pre-training is a potent method for DNN regression on small datasets. Moreover, a fully connected DNN outperforms both SNN and SVM in terms of accuracy and generalization. The subsequent analysis will encompass:

- Data set pre-processing and partitioning into training/testing subsets.
- Training of SVM, SNN, and DNN.
- Evaluation of the trained machine learning models.
- Model comparison.
- Determining accuracy and selecting the best model.
- Employing the chosen model for predictions.

The DNN training process is bifurcated into pre-training and fine-tuning phases. Our analysis will also involve training and evaluating a support vector machine and a shallow neural network to validate DNN's superior accuracy. Post the training of SVM, SNN, and DNN, a summary of their training/testing accuracies will be drawn and juxtaposed. The PCA pre-processing adversely impacts both training and testing accuracies, potentially due to the loss of non-linear data. The subsequent segments of this research will rely on an unprocessed dataset. A fully connected DNN, encompassing three or more hidden layers, demonstrates its superiority over shallow ANNs and support vector machines by achieving enhanced prediction accuracy and generalization [29]. While DNNs paired with extensive datasets

remain the gold standard, DNNs with limited datasets, supplemented with pre-training, emerge as a viable alternative when vast datasets are inaccessible. In this study, we frequently encounter small datasets, and the challenges at hand demand fewer input variables.

4. Results of Experiments and Discussion

To facilitate experimental studies, a dedicated library was crafted, encompassing all the aforementioned algorithms in a software format. The development leveraged the Python programming language, and the Pycryptodome library was employed to instantiate encryption algorithms. This library boasts pre-compiled encryption algorithms available in the .pyc format. The software manifests as a command-line utility, enabling users to designate files for encryption, select the desired algorithm, and determine its operational mode.

The utility operates as follows:

1. Before initiating any task, users must activate the virtual environment to access algorithms from the Pycryptodome library.

2. If no arguments are provided during utility execution, or if the utility is invoked with the -h flag, users are presented with a concise guide on utility usage in the format below.

```
python main.py -[enc/dec] [Enc/Dec method] [Mode] [Key] [InFile] [OutFile]
To see supported methods use -> python main.py -s
To see supported modes use -> python main.py -m
```

where [enc/dec] denotes encryption or decryption, [Enc/Dec method] specifies the chosen algorithm, [Mode] indicates the operational mode, [Key] represents the encryption key, [InFile] is the source file, and [OutFile] is the destination file. Additionally, users can invoke the utility with the -s or -m flags to peruse available algorithms and operational modes, respectively.

3. Upon successful encryption or decryption, the resultant files are located in the program's root directory.

To streamline the software tools' implementation and facilitate future enhancements, abstract classes were devised and housed in the Libs/CryptoAbstract.py directory. These classes ensure uniformity across the cryptographic algorithms, guaranteeing a consistent data input and output interface. Furthermore, the Libs/env.py path contains a file that enumerates dictionaries detailing available algorithms and their respective operational modes.

Each algorithm and its various versions are stored in a directory with the appropriate name, and implemented as a class with the following methods:

- Initialization.
- Block encryption/decryption.
- Reading the file byte by byte for further data encryption.

The software implementation of the dataset is created in such a way that it can be modified and supplemented rather quickly in the future.

For the experiment, files of various sizes (from several kilobytes to several gigabytes) were encrypted. Each file was encrypted by each algorithm up to 10 times (to increase the accuracy of the study).

Experiments were conducted on the following hardware platform: processor—Intel core i7 (9^{th} gen); video card – Nvidia Geforce GTX 1060; RAM—16GB RAM DDR3; operating system – Windows 10.

The results of the experiment were processed and presented in the dataset in the form of a quantitative assessment—a comparison table (Table 2) was created with expert assessments of parameters:

- Data on cryptographic resistance (CRS).
- Cryptographic resistance reserve (FCRS).
- Encryption speed (ECR).
- Extension key (KEA).

The first two parameters were taken from open sources, others were verified by experimental research by the authors.

In Table 2 numbers from 1 to 10 determine the effectiveness of the algorithm according to the specified criterion (1 is the worst score, and 10 is the best).

Table 2
Comparison of dataset crypto algorithms

Algorithm	CRS	FCRS	ECR	KEA
Salsa20 (S)	7	8	9	6
Panama (S)	6	7	7	—
HC-256 (S)	6	4	7	—
DES (B)	2	2	4	4
Triple DES (B)	7	5	2	2
RC2 (B)	5	4	6	5
AES (B)	9	8	10	7
RC6 (B)	8	7	10	7
Blowfish (B)	6	7	7	6
Twofish (B)	9	7	7	7
Serpent (B)	9	9	7	8
GOST (B)	7	6	6	10
Kalyna (B)	10	9	9	7

Therefore, a dataset of crypto-algorithms was created, which can be used to ensure the confidentiality of data during transmission from UAVs. This dataset can also be used by ANN to select one or another encryption algorithm depending on the given requirements. Next, the authors plan to select the criteria for the application of this ANN dataset, after which a knowledge base for ANN training will be created. In the future, a method of increasing the productivity of ANNs will also be developed.

5. Conclusions and Future Research

The research has successfully culminated in the creation of an open dataset of cryptographic algorithms. This dataset is designed to bolster the efficacy of information protection during UAV transmissions, leveraging the capabilities of ANNs. As of now, the dataset encompasses a variety of block and stream cryptographic algorithms, detailed within this article. The robustness and performance speed of these algorithms has been ascertained based on extensive scientific investigations conducted by diverse researchers.

Future endeavors aim to refine the dataset by incorporating a broader spectrum of algorithms, including proprietary ones developed by the authors. Efforts will also be directed towards optimizing the software rendition of these algorithms and undertaking supplementary experiments. These experiments will be geared towards curating a comprehensive repository of algorithms tailored for UAV applications, especially in synergy with ANNs. Furthermore, the research ambit will be expanded to explore other challenges associated with UAV operations that can be addressed using the advanced tools of ANN.

Acknowledgments

The work was carried out as part of the research project “Intellectualized System of Secure Transmission of Packet Data based on Reconnaissance and Search Unmanned Aerial Vehicle” (#0122U002361), financed by the Ministry of Education and Science of Ukraine from 2022–2024.

References

[1] X. Du, et al., Data Processing and Encryption in UAV Radar, in: 4th Adv. Information Management, Communicates, Electronic and Automation Control Conf. (2021) 1445–1450. doi: 10.1109/IMCEC51613.2021.9482373

[2] V. Sokolov, P. Skladannyi, A. Platonenko, Video Channel Suppression Method of Unmanned Aerial Vehicles, in: IEEE 41st International Conference on Electronics and Nanotech-nology (2022) 473–477. doi: 10.1109/ELNANO54667.2022.9927105

[3] S. Gnatyuk, et al., Analysis of Methods for Ensuring the Confidentiality of Data Transmitted from UAVs, Cybersecur.: Education, Science, Technology 1(17) (2022) 167–186.

[4] T. Dong, T. Huang, Neural Cryptography based on Complex-Valued Neural Network, IEEE Transactions on Neural Networks and Learning Systems 31(11) (2020) 4999–5004.

[5] S. Jhajharia, S. Mishra, S. Bali, Public Key Cryptography using Neural Networks and Genetic Algorithms, in 6th International Conference on Contemporary Computing (IC3) (2013) 137–142.

[6] Y. Xiao, Q. Hao, D. Yao, Neural Cryptanalysis: Metrics, Methodology, and Applications in CPS Ciphers, in IEEE Conference on Dependable and Secure Computing (DSC) (2019) 1–8.

[7] M. Niemiec, M. Mehic, M. Voznak, Security Verification of Artificial Neural Networks Used to Error Correction in Quantum Cryptography, in 26th Telecommunications Forum (TEL-FOR), Belgrade, Serbia (2018) 1–4. doi: 10.1109/TELFOR.2018.8612006

[8] G. Das, M. Kule, A New Error Correction Technique in Quantum Cryptography using Artificial Neural Networks, in IEEE 19th India Council International Conference (INDICON), Kochi, India (2022) 1–5. doi: 10.1109/INDICON56171.2022.10040091

[9] T. Schmidt, H. Rahnama, A. Sadeghian, A Review of Applications of Artificial Neural Networks in Cryptosystems, in World Automation Congress, Waikoloa (2008) 1–6.

[10] B. Bebeshko, et al., Application of Game Theory, Fuzzy Logic and Neural Networks for Assessing Risks and Forecasting Rates of Digital Currency, Journal of Theoretical and Applied Information Technology 100(24) (2022) 7390–7404.

[11] K. Khorolska, et al., Application of a Convolutional Neural Network with a Module of Elementary Graphic Primitive Classifiers in the Problems of Recognition of Drawing Documentation and Transformation of 2D to 3D Models, Journal of Theoretical and Applied Information Technology 100(24) (2022) 7426–7437.

[12] D. Bernstein, The Salsa20 Family of Stream Ciphers, in New Stream Cipher Designs, Lecture Notes in Computer Science 4986 (2008). doi: 10.1007/978-3-540-68351-3_8

[13] Panama Stream Cipher (2012). http://nightcrowlwing.blogspot.com/2012/10/panama-stream-cipher.html

[14] H. Wu, A New Stream Cipher HC-256, Lecture Notes in Computer Science 3017 (2004). doi: 10.1007/978-3-540-25937-4_15

[15] NIST. Advanced Encryption Standard (2021). https://csrc.nist.gov/publications/detail/fips/197/final

[16] R. Davis, The Data Encryption Standard in Perspective, in: IEEE Communications Society Magazine 16(6) (1978) 5–9. doi: 10.1109/MCOM.1978.1089771

[17] J. Thakur, N. Kumar, DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis. International Journal of Emerging Technology and Advanced Engineering 1(2) (2011) 6–12.

[18] RC2. Block Cipher with Symmetric Secret Key (2020). http://www.crypto-it.net/eng/symmetric/rc2.html

[19] R. Anderson, E. Biham, L. Knudsen, Serpent: A Proposal for the Advanced Encryption Standard (2000). https://www.cl.cam.ac.uk/~rja14/Papers/serpent.pdf

[20] T. Gonzalez, A reflection attack on Blowfish (2007). http://karbalus.free.fr/sat/docsat/PaperGonzalezTom.pdf.

[21] O. Kara, C. Manap, A New Class of Weak Keys for Blowfish, in Fast Software Encryption. FSE 2007, Lecture Notes in Computer Science 4593 (2007). doi: 10.1007/978-3-540-74619-5_11

[22] B. Schneier, et al., The Twofish Encryption Algorithm: A 128-Bit Block Cipher. New York City: John Wiley & Sons (1999). isbn: 0-471-35381-7

[23] J. Kelsey, B. Schneier, MARS Attacks! Preliminary Cryptanalysis of Reduced-Round MARS Variants. AES Candidate Conference (2000) 169–185.

[24] S. Haykin, Neural Networks: A Comprehensive Foundation. The Knowledge Engineering Review 13(4) (1999) 409–412.

[25] L. Juracy, R. Garibotti, F. Moraes, From CNN to DNN Hardware Accelerators: A Survey on Design, Exploration, Simulation, and Frameworks (2023).

[26] Z. Hu, et al., High-Speed and Secure PRNG for Crypto-Graphic Applications, International Journal of Computer Network and Information Security 1(3) (2020) 1–10.

[27] Y. Sovin, et al., Efficient Implementation and Performance Comparison of “Kalyna” and GOST 28147-89 Ciphers using Vector Extensions SSE, AVX, and AVX-512. Information Protection 21(4) (2019) 207–223. doi: 10.18372/24107840.21.14266

[28] DSTU 7624:2014, Information Technology. Cryptographic Protection of Information. Algorithm for Symmetric Block Transformation. http://online.budstandart.com/ua/catalog/doc-page?id_doc=65314

[29] H. Bhadeshia, Neural Networks in Materials Science. ISIJ International 39(10) (1999) 966–979.

[30] V. Zhebka, et al., Optimization of Machine Learning Method to Improve the Management Efficiency of Heterogeneous Telecommunication Network, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3288 (2022) 149–155.

[31] Z. B. Hu, et al., Authentication System by Human Brainwaves Using Machine Learning and Artificial Intelligence, in: Advances in Computer Science for Engineering and Education IV (2021) 374–388. doi: 10.1007/978-3-030-80472-5_31