

A secure communication protocol for civil drones

Ayad Al-Adhami¹, Rajaa K. Hasoun², Ekhlas K. Gbashi¹, Soukaena Hassan¹

¹Computer Sciences Department, University of technology, Baghdad, Iraq

²Information System Management, College of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq

Article Info

Article history:

Received Sep 10, 2021

Revised May 11, 2022

Accepted Jun 9, 2022

Keywords:

Cipher block chaining

Drones

Encryption

Hash function SHA-1

RADG algorithm

Ribonucleic acid RNA

ABSTRACT

This paper introduces a secure communication protocol that provides secured communication pathways to manipulate drones through unsecured communication. The deployment of the proposed protocol works through providing two secured communication paths; drones to the drone's controller path and controller to data centre path. The first secured communication path has achieved a high level of security and privacy by using a modification of SHA-1 method and an advanced encryption method. The modification of the SHA-1 is called 83SHA-1. These modifications can increase rounds in the first stage up to 83 rounds, inject each round with expansion and S-Boxes procedures that are used in DES to extend length from 160 to 240 bits then reduce it from 240 to 160 bits. After hash data from the drone then use the advanced encryption method which is called Geffe-Genetic (GG) Encryption algorithm where three types of keys will be used for deception attackers. The second accomplishment is to ensure providing secure communication between the drone's controller and datacentre by using RNA-RADG-CBC (RRCBC) encryption algorithm where will generate an initialization vector (IV) for cipher block chaining (CBC) randomly, generate keys, and propose an encryption/decryption method. The security analysis shows a promising high security level of drones's data.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Ayad Al-Adhami

Computer Sciences Department, University of Technology

Baghdad, Iraq

Email: ayad.h.ibrahim@uotechnology.edu.iq

1. INTRODUCTION

Unmanned aerial vehicles (UAVs), drones, are remotely controlled flying vehicles using compound automation systems. They are used for several civil missions such as (shipping and delivery, filming and journalism, wildlife monitoring, rescue operations and healthcare, disaster management, and geographic mapping) and military missions such as (air strikes, surveillance, and bomb detection) which are growing at matchless average. Security and privacy challenges are added to these vehicles because of their high mobility and dynamic topology. Drones are tending to several various types of attacks that can cause major challenges in safeguarding [1]. Depending on the attacker's resources, attacks can be categorized into external attacks (in which false information is transmitted to other drones) and internal attacks (in which messages are inserted, resent, or deforming by the attacker). On the other hand, attacks can be categorized as passive attacks (which observe the traffic of data among drones) and active attacks (which cause big damage to the network by altering data and causing a denial of services by resending the former messages) [1].

Many researchers have introduced secured protocol to avoid these types of attacks; Samid [2] explains a cryptographic premise and avoids intensive computation. Non-complex processing of at-will size keys will achieve security. Their proposed method is to increase the role of randomness and build ciphers that

can handle variable size keys without choking on computation. Orthodox cryptography seeks to create a thorough mix between key bits and message bits, resulting in heavy-duty computation. A scenario that's used as described for the proposed cipher: a drone taking videos of variable sensitivity and hence variable required security-handled by the size of the 'park'. However, the author did not provide a full security analysis for drone attacks. Cheon *et al.* [3] proposed an efficient linearly homomorphic authenticated encryption (LinHAE). In their protocol, they claim that their encryption protocol guarantees the security against passive and active attacks, unlike homomorphic encryption which does not offer means to check whether the received signal at the drone side is compromised or authentic. Nassi *et al.* [4] proposed a new privacy invasion attack technique that can detect whether a specific point of interest (POI) is being video streamed by a drone. Their proposed method shows that applying a periodic physical stimulus on a target/victim being video streamed by a drone causes a watermark to be added to the encrypted video traffic that is sent from the drone to its operator and how this watermark can be detected using interception. Ozmen *et al.* [5] proposed a protocol that can meet the requirements of battery-limited. Their implementation procedure was over the internet of Drones and on two common drone processors, namely 8-bit AVR and 32-bit ARM. Their result shows that protocol is secured but there are no attack models in their assumptions. Ozmen *et al.* [6] propose an improved cryptographic framework for small aerial drones, which offers significant energy efficiency and speed advantages over standard cryptographic techniques. The proposed techniques and the standard counterparts were implemented on an actual small aerial drone (Crazyflie 2.0) and provided an in-depth energy analysis.

From the previous discussion, most of the related works are mainly focused on providing privacy and security for data that are stored or captured by drones. Even though their protocols are assumed to be provided a level of security for drones' data, there are no scenarios that can consider possible attacks over both paths from drones to controller and drones' controller to a data centre that deal with analysis data from drones. Therefore, it is necessary to provide a high level for all levels of communication that takes into consideration any particular attacks against drones or their managing analysis system. Therefore, this paper introduces a new security design that guarantees a high level of privacy and confidentiality for the data from drones and the communication over drones. The security assumptions combine multi-levels of integrity and encryptions. First of all, the design of the proposed system studies the possibility of capturing photos from drones and recording short and securely sending these data to the controller using a modification of SHA-1 hash function then encrypting the hashed data using an advanced encryption method that combines genetic algorithm and cryptographic algorithms. The second achievement is to discuss all the possibilities of providing an advanced security level for the assumed secured communication between the drones controller and the data centre system that manipulate all records for drones and their controller.

The rest of the paper is as follows: Section 2 describes some related theoretical cryptographic methods which are used in the design idea and the implementation. Section 3 introduces the workflow and the architecture for the proposed protocol with all its description. Section 4 presents the experiment result with security analysis which considers some related attacks that could affect the integrity & confidentiality level of the whole system. Section 5 concludes the work.

2. CRYPTOGRAPHIC BACKGROUND

Cryptography is a method of protecting communications and information through the use of codes so that only those for whom the information is intended can read and process it. There are several ways of classifying cryptographic algorithms. They will be categorized based on the number of keys that are employed for encryption and decryption. Secret key cryptography (SKC) is dependent on a single key for both decryption and encryption; also, it is called symmetric encryption. Public key cryptography (PKC) is one key encryption that is used for encryption and another one for decryption; also called asymmetric encryption. Hash Functions is a mathematical transformation that is used to irreversibly "encrypt" information, which is used for message integrity [7], [8]. Another methodology in the domain of cryptography is based on deoxyribonucleic acid (DNA) sequences. DNA molecule having the capacity to store, process and transmit information inspires the idea of DNA cryptography. It works on the concept of DNA computing which uses four bases i.e., Adenine (A), Guanine (G), Cytosine (C), and Thymine (T) to perform computation with 0 and 1. RNA is a single-stranded molecule that contains the bases adenine (A), cytosine (C), and guanine (G) but not the thymine; instead, it contains uracil (U) base [9]-[12].

Reaction automata direct graph (RADG) RADG is an algorithm that depends on reaction states and direct graph, which is keyless algorithm cryptography, this means, it is not used as a key during encryption and decryption and there is no agreement between two parties of communication. RADG is represented by a set of tuples $\{R, Q, \Sigma, \Psi, J, T\}$, where R is a reaction set that has m of length, which have λ of values for each element, Q is a standard design set that has n length, also have λ of values, Σ represents input data (or alphabet), Ψ represents output transition, J is jump set which is a subset of Q set, that have k length, which

has no value, just transmit from one state to another in Q set and T represents transition function. The encryption process in RADG algorithm is begun by selecting a random state from Q set if Q state is transmitted to J state and then selecting a new random state to continue the encryption process [13].

Secure hash algorithms (SHA), a family of cryptographic functions used to keep data secured. It depends on transforming the data by using a hash function: which is an algorithm that consists of bitwise operations, modular additions, and compression functions. A fixed-size string the hash function produces that looks nothing like the original. SHA are designed to be one-way functions, that's means once they're transformed into their respective hash values, it's impossible to transform them back into the original data. A few algorithms of interest are SHA-1, SHA-2, and SHA-3, each of which was successively designed with increasingly stronger encryption in response to hacker attacks [14]. The stream cipher is one of the important branches of modern cryptography. The stream cipher systems depend basically on linear feedback shift register (LFSR) units [15], [16]. The Geffe generator can be defined by 3 maximum-length LFSRs whose lengths w_1, w_2, w_3 are pairwise relatively prime, with nonlinear combining function.

$$F_3(y_1, y_2, y_3) = y_1 * y_2 \oplus (1 \oplus y_2) * y_3 = y_1 * y_2 \oplus y_2 * y_3 \oplus y_3 \quad [17]$$

Genetic algorithms (GA) are heuristic search algorithms based on the ideas of evolutionary natural selection and genetics. These algorithms are based on the principle of Darwinian idea of survival of the fittest and natural genetics. Generally, a GA consists of three basic operations. Selection, crossover and mutation [18]-[21]. Cipher block chaining (CBC) is a type of operation for a block cipher. CBC uses an initialization vector (IV) of a certain length. The key characteristics of CBC are that it uses a chaining mechanism that causes the decryption of a block of ciphertext to depend on all the preceding ciphertext blocks. As a result, the entire validity of all preceding blocks is contained in the immediately previous ciphertext block. A single-bit error in a ciphertext block affects the decryption of all subsequent blocks. Rearrangement of the order of the ciphertext blocks causes decryption to become corrupted. In CBC each plaintext block is XORed with the immediately previous ciphertext block, and then be encrypted [22]-[25].

3. SECURE CHANNEL PROPOSED PROTOCOL FOR CIVIL DRONES

This section initiates the general architecture of the proposed system. The proposed system uses different cryptographic techniques to ensure a high level of security and to avoid confidentiality and integrity threats. The proposed system has different phases with different responsibility controls for the drones.

3.1. Design overview

The overall structure for the proposed system is to consider a general example application that uses civil drones. The overall structure for the proposed system is to consider a general example application that uses civil drones to record secured short videos and capture photos around specific trade Malls. The scenario of using civil drones is to record all events around the trade mall in a specific area and to speed up the process of analyzing data from the general CCTV systems. As shown in Figure 1, the proposed system uses three stages; the first stage is a drone's state that uses to capture live photos and record short videos. The second stage is the ground-base stage which controls and detects drones, tracks the drones' bath, and monitors all processes from drones. The last stage is the data-centre stage which analyses all data from drones and ground-base.

During the transit, drones are responsible to capture photos and short videos and securely send this information to the ground base (drones' controller) through an unsecured channel (wireless transmit). Taking into consideration, some of the stored data are sensitive and need to be securely analysed by the data centre. Therefore, to ensure the confidentiality and integrity of the stored data in drones, the enhanced protocol from SHA-1 hash function is used which is called 83-SHA-1 algorithm. The 83-SHA-1 hash function algorithm is used to hash drones' data then the output data will be encrypted using ground glass diffuser (GGD) to confirm that the sensitive data are not manipulated by a spoofer. All data from drones are stored in the Ground-Base whose obligation is to send drones' data to the data-centre as well as to detect drones, track the path of drones, and monitor the drone's activity. Although the communication between the ground base and the data-centre is considered secured, the data from the ground base are encrypted using the suggeste RNA-RADG CBC (RRCBC) encryption algorithm. As a final phase, the data-centre is collected all data from drones and ground-Base and analyse the whole data to give a decision for a specific purpose.

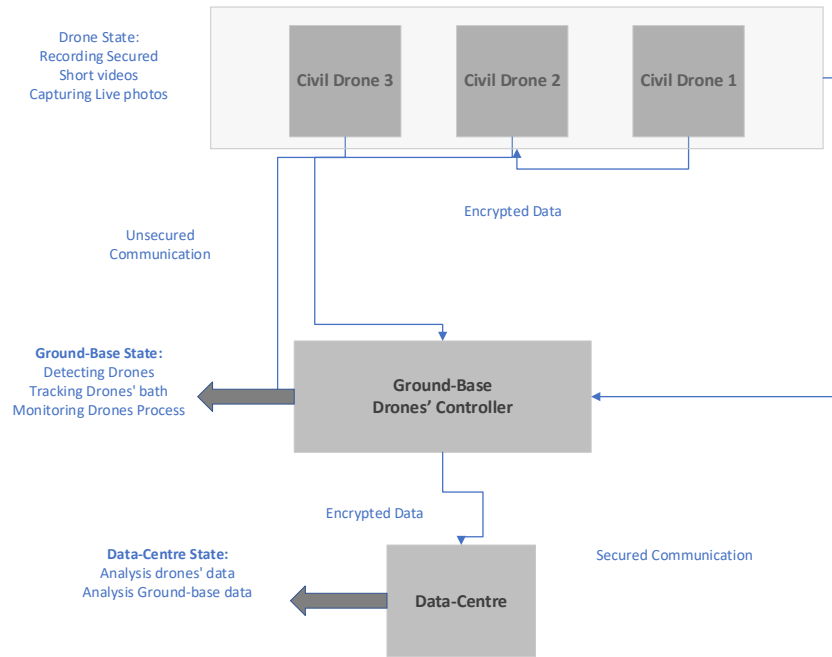


Figure 1. General block diagram of the proposed system

3.2. System communication phases

Drones have played the main role in the proposed system as they take recursively short videos captured and transmit them to the data-centre through the ground base controller. As mentioned before there are two phases for the proposed system: The drones to ground-base phase and ground-base to data-centre phase. The drones are only connected to the ground-base. Each trade mall represents the ground base for a set of drones d1, d2, ..., dn. The set of drones send their data to the ground base via a wireless communication channel and this communication is at high risk to be captured by illegitimate users that can seize all events from drones 'data. Therefore, drones send their secured data afterward encrypting the drone's data and hashing the output data. After capturing short videos, around 30 minutes continuously, the recorder videos are hashed using a suggested 83-SHA-1 algorithm, then encrypt the hashed videos and the original videos using GG algorithm. Subsequently, drones are saved and sent the encrypted data to the controller then, after receiving the acknowledgment received from the controller, all data from drones will be deleted.

After initialising the communication with drones, ground-base will receive all required information from drones. The first stage is to verify the drone's hashed identification (ID), then match the hashed videos by using the 83-SHA-1 algorithm. This stage has significantly insured the integrity of all data and improved the overall security of the proposed system. Afterward, the next stage is a communication channel between ground-base and data-centre. This phase is started after receiving the encrypted hashed videos and verifying their confidentiality and integrity then encrypting the files of these videos using a suggested encryption/decryption algorithm called DRRCBC algorithm. Sending encrypted files to the analytical server. The analytical server will receive the encrypted videos from controllers, decrypt them using the suggested encryption/decryption algorithm called DRRCBC algorithm, and rebuild continuous videos for each ground controller for each drone.

3.3. Cryptographic models for the proposed system

All outsourced information that is being written from the Drones' data such as videos recorder and live photos are encrypted by using different cryptographic models to guarantee that these data cannot be manipulated by intercepting the communication. Therefore, two-levels of confidence and integrity are proposed to increase the level of security of the system. Moreover, the two cryptographic models, which are introduced, are installed in both communication phases. Thus, to ensure that there is no information leak for the whole system.

3.3.1. Suggested 83-SHA-1

The first cryptographic model is intended to ensure the integrity and confidentiality of the first phase (Drone to the controller) by combining a suggested hash function and a suggested encryption scheme. The first step is used to hash the drones ID by using the suggested 83-SHA-1 algorithm which is used for hashing

the files in drones. This hashing algorithm increases the round to 83 instead of 80 and increases randomness by injecting the DES S-Boxes in each round. The procedure of the suggested 83- SHA-1 is as shown in:

- Get the video to be hashed and then to be x .
- Padding the message (x) to appropriate the size of 512-bit multiples.
- Divide the message which is padded into blocks.
- Set predefined constant to initial value H_0 .
- Padding of the message, if x is the message with l of bits as a length, then the size of the overall message will be 512-bit multiples, the padding is single "1" at end of the message which is followed by zeroes of k - bit, and the 64-bit added of l . consequently, the k which is the number of needed zeroes is calculated by:

$$W_j = \left\{ \begin{array}{l} x_i^j, 0 \leq j \leq 15 \\ (W_{j-16} \oplus W_{j-14} \oplus W_{j-8} \oplus W_{j-2}) \lll 1, 16 \leq j \leq 82 \end{array} \right\} k \equiv 512 - 64l = 448(l + 1) \text{ mod } 512$$

The padded message will be divided before performing the function of reducing, must the message is divided into blocks of 512-bit x_1, x_2, \dots, x_n . For each 512-bit block will divide it into words are 16; each word with a size of 32-bit. For example, the block i^{th} for the message, x is divided into:

$$x_i = (x_i^{(0)} x_i^{(1)} \dots x_i^{(15)}) \quad (1)$$

where, $x_i^{(k)}$ is a word with a 32-bit size.

- Value initialization, H_0 is a buffer of size 160-bit which is exploited to store the hash value initialized in iteration number one. There are five words with a length of 32-bit; those are constant and set by hexadecimal representation as:

$$A = H_{(0)}^0 = 67452301, B = H_{(0)}^1 = EFCFAB89, C = H_{(0)}^2 = 98BADCFE \\ D = H_{(0)}^3 = 10325476, E = H_{(0)}^4 = C3D2E1F0$$

- Computation of hashing, each block x_i of the message is processed in stages (four stages) each stage is 20 rounds (but the first stage will be increased to be 23 rounds). SHA-1 uses the following steps:
 - The scheduling of message, produces an (83) 32-bit word, not as traditional which compute (80) 32-bit word, those words are: W_0, W_1, \dots, W_{92} or each of the 83 rounds. Words W_i are abstracted from the 512-bit message block as in following:

$$W_j = \left\{ \begin{array}{l} x_i^j, 0 \leq j \leq 15 \\ (W_{j-16} \oplus W_{j-14} \oplus W_{j-8} \oplus W_{j-2}) \lll 1, 16 \leq j \leq 82 \end{array} \right\} \quad (2)$$

where $x \lll n$ for word X circular shift to left by n bit positions; indicated by this operation.

- Registers selected for the work are five with a size of 32-bits A, B, C, D, E .
- As the traditional, value of hashing $H_{(i)}$ has five words with size 32-bit $H_i^{(0)}, H_i^{(1)}, H_i^{(2)}, H_i^{(3)}, H_i^{(4)}$. First hash value captures the initial value $H_{(i)}$, which is substituted by a new hash value after treating every single block of message. The last H_n hash value is output $h(x)$ of SHA-1 algorithm.
- The four stages of SHA-1, have the same infrastructure but internal functions f_t and constants K_t are different, where $1 \leq t \leq 4$. Since the stage is consisting of 20 rounds unless the first stage has 23 instead of 20, each of the pieces in the message block is treated by the f_t function in the same time with constant K_t related to the stage. After completing 83 rounds the final output is inserted into the input value H_{i-1} modulo 2^{32} in the fashion of word-wise. The process through round m in stage n is given by:

$$A, B, C, D, E = (E + f_t(B, C, D) + (A) \lll 5 + W_j + K_t), A, (B) \lll 30, C, D \quad (3)$$

- After each round the five registers will randomize depending on DES procedures; that by expanding each register from 32-bit to 48-bit using DES expansion procedure. Then reduce each 48-bit to 32-bit using DES S-Boxes.

3.3.2. Suggested GG algorithm

The next security level is to ensure a high level of confidentiality by combining Geffe generator and Genetic algorithm, in which is called GG encryption method. The encryption procedure of the suggested GG algorithm is as shown in:

- a. Convert the file of hashed video to binary and then to text message
- b. Create 3 seeds to generate the secret key (16 bits) using Geffe Generator.
- c. Divide the secret key (16 bits) into two parts each one (8 bits).
- d. Consider the keys as chromosomes to apply the genetic algorithm step.
- e. Read the text characters of the converted video.
- f. Convert the secret text characters to two numbers in the form (n1, n2) where n1 represents the row and n2 represents the column using the below formula.

	0	1	2	3	4	5
0	A	B	C	D	E	F
1	G	H	I	J	K	L
2	M	N	O	P	Q	R
3	S	T	U	V	W	X
4	Y	Z	0	1	2	3
5	4	5	6	7	8	9

- g. Convert the numbers (n1, n2) to binary representation within 6 bits only depending on the formula in step6. Then add padding (2 bits) to every (6 bits).
- h. Convert step7 result (binary representation) to characters coding.
- i. Using the S-box lookup table which consists of character symbols and makes swapping for these two character symbols where the first character symbol is used as determination to the row of the S-box lookup table and the other character symbol is used as determination to the column of the S-box lookup table. The row determination character symbol stays and the corresponding cell of that row is changed with the column determination
- j. Convert character symbols to binary representation.
- k. Make XOR between Genetic algorithms first children.
- l. Make XOR between step9 result and Genetic algorithm second children.
- m. Send the step12 result.

The decryption of the suggested GG method is as shown in:

- a. Get the encrypted hashed video file
- b. Create 3 seeds to generate the secret key (16 bits) using Geffe Generator.
- c. Divide the secret key (16 bits) into two parts each one (8 bits).
- d. Consider the keys as chromosomes to apply the genetic algorithm
- e. Read the received binary code.
- f. XOR the received binary code with the second genetic algorithm children.
- g. Make XOR between the step6 result and the first genetic algorithm children.
- h. Convert the binary representation to character codes.
- i. Making inverse S-box converting where each two-character symbol will be swapped. The first character symbol will be as an indication to the row and the second character symbol as an indication to cell content to retrieve the corresponding column character symbol with the row character symbol using the below formula in GGD encryption step9.
- j. Convert character codes to binary representation.
- k. Delete the padding bits (2 bits).
- l. Convert to decimal representation in the form (n1, n2) where every 3 bits will be converted to a decimal value.
- m. Retrieve the characters from these numbers using the S-box lookup as in the formula below that is used at the sender side in GG encryptionstep6.
- n. Read the secret text characters convert it to binary then convert it to video.

3.3.3. Suggested RRCBC algorithm

The second cryptographic model is to encrypt the communication between the controller and the data-centre using RRCBC algorithms for encryption and decryption files where both suggested algorithms are installed in the controller and data centre to encrypt-decrypt the collected videos from drones using a combination of RNA, RADGE, and CBC algorithms. The overall encryption/ decryption procedure of the RRCBC is subsequently shown in Algorithm 1 and Algorithm 2 as shown in:

Algorithm 1. Encryption process

- 1.1 Load video file.
- 1.2 The loaded file must be converted to binary form.
- 1.3 Binary file divided into n blocks.
- 1.4 Input two seeds.
- 1.5 Use the above seeds to generate IV as follows:
 - By using the following formula in Table 1 replace the seed characters with corresponding characters.

Table 1. Replacing seeds with corresponding characters

Seq.	character	Corresponding swap character	Seq.	character	Corresponding swap character	Seq.	character	Corresponding swap character
1	A	;	17	Q	5	33	6	P
2	B	'	18	R	4	34	7	O
3	C	“	19	S	3	35	8	N
4	D	[20	T	2	36	9	M
5	E]	21	U	1	37	:	L
6	F	}	22	V	0	38	.	K
7	G	{	23	W	Z	39	?	Kj
8	H)	24	X	Y	40	(I
8	I	(25	Y	X	41)	H
10	J	?	26	Z	W	42	{	G
11	K	.	27	0	V	43	}	F
12	L	:	28	1	U	44	[E
13	M	9	29	2	T	45]	D
14	N	8	30	3	S	46	“	C
15	O	7	31	4	R	47	‘	B
16	p	6	30	5	Q	48	;	A

- The resulted characters are converted to binary representation.
 - The first seed is XORed with the second seed.
- 1.6 R1= XOR IV with the first block.
 - 1.7 Input two secret keys (secretkey1, secretkey2).
 - 1.8 Use the proposed key generation method to generate the key
 - Load binary secret seed.
 - The secret seed is converted to a character code.
 - Convert the character coding to binary.
 - Using RADG to swap step3 bits to decimal values as shown in Figure 2.

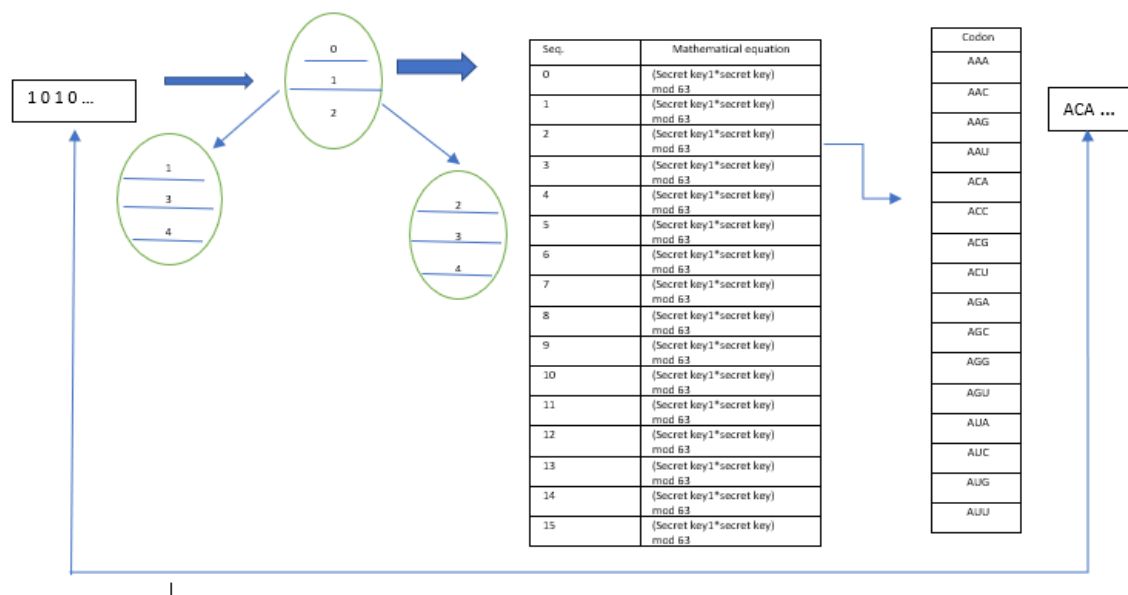


Figure 2. RADG Swap

- The mathematical equation will use secretkey1 to choose the RNA codes.
 - Load secretkey2.
 - Rearrangement of every RNA code.
 - RNA codes will be changed to binary procedure as A to 11, U to 10, G to 01, and C to 11.
 - Input the key that will use in the proposed sub-encryption method
- 1.9 For J=1 to n
- Begin (Sub method) rotation Integrity and complexity
- If J=1 then
 1. Use the generated key to encrypt R1 (sub-method/encrypt, see c.3.1 subsections).
 2. R2=Get Jciphertext.
 - Else
 1. Read J block.
 2. W= R2 XOR J block.
 3. Encrypt W using the generated key (sub-method/encrypt see c.3.1 subsections).
 4. W=Get Jciphertext.
- End
- 1.10 Send the result (n ciphertexts).

The decryption process is explained in Algorithm 2

Algorithm 2. Decryption process (input to the algorithm is the received n ciphertexts in binary form)

- 2.1 input the two seeds.
 - 2.2 Generate the IV by using these seeds.
 - 2.3 C1= IV XOR first block.
 - 2.4 Input the secret two keys.
 - 2.5 By using the proposed key generation method generate the key (see step1 (1.8)).
 - 2.6 For J=1 to n
 - If J =1 then
 1. Decipher C1 using the produced key (sub-method/decrypt).
 2. C2=Get J plaintext block.
 - Else
 1. Read J block.
 2. M= C2 XOR J block.
 3. Decrypt M using the generated key (sub-method/decrypt see c.3.2 subsection).
 4. Get J clear text block.
 5. M= the decryption result.
- End
- 2.7 the binary form of the plaintext blocks will be converted to characters.
 - 2.8 Finally, read the resulted secret message characters.
- Step3: End.

Sub Method/Encryption procedure

1. Convert the read binary bits to virtual characters.
2. Read (row rotation no's and column rotation no's).
3. Input the secret matrix of characters.
4. Use the sub-secret key, ensure row rotation to the characters matrix.
5. Use the sub-secret key, ensure column rotation to the characters matrix.
6. Use the above matrix to change the virtual characters to numbers.
7. The above numbers are converted to binary exemplification.
8. Read the which controls.
9. Apply rotation using the sub- secret key as in step 7.
10. By using the proposed generation method, read the main generated key binary code.
11. To apply row and column rotation, convert these binary codes to a 2D array.
12. The produced key binary codes matrix is applied row rotation using rotation's no.
13. Rotation of column is applied using rotation's no to the generated key binary codes matrix.
14. ID vector are covered from the generated matrix vector.
15. XOR Step 9 and Step 14.
16. Read the XOR result.

Sub Method/Decryption procedure

1. Look through the established binary bits.
2. Look through the generated-key binary code.
3. Convert Step2 binary code to 2D array
4. Read the Row and column rotations no.
5. Do inverse row rotation.
6. Do inverse column rotation.
7. Convert the result to 1D vector.
8. The received binary code XOR with Step7.
9. To retrieve the secret binary codes read the number of Rotations.
10. Apply inverse rotation to this code.
11. From this binary code retrieve the decimal numbers.
12. Read the no. Row and no. Column rotation.
13. Read the unrevealed matrix.
14. Do inverse Rotation to the rows and column of matrix.
15. Use the rotated matrix to restore the characters which are matching to the decimal numbers (step11).
16. Retrieve secret text characters to binary bits.

4. DISCUSSION AND EXPERIMENTAL RESULTS

The proposed system is designed to overcome the possible security threats that can affect the confidentiality and integrity of the drones' data. The proposed protocol secures communication for booth drone's controller, drones, and data centre through using a combination of complex cryptographic techniques that makes the possibility of breaching data negligible and need a high hardware and software resources to break down the security and privacy baths. The security achievement that has got in the proposed system ensures a high level of security and privacy which prevents an adversary from leaking any information even if the drone's ID is leaked. Moreover. The adversary goal, which is considered in the proposed system, is to exploit different types of vulnerabilities and to cause a further varying amount of harm to the system and user. In other words, the adversary will try to control over flight path, crashing the drones and cloning the drones' data or eavesdrop on the communication to order to get access to the drones' data or tamper with the drones' data. Another adversary's goal is to perform a DoS attack by incessantly sending requests to the drones to take control of numerous requests from an adversary that can be simultaneously sent. It will be similar to the Buffer-overflow attack, the directional application smashes which causes the drones to crash and affect the availability of the whole system. In all adversary scenarios, the drones' ID are hashed using 83-SHA-1 protocol and encrypted all drones' data (short videos and live captures) using GGD algorithm. These cryptography mixing protocols can result in satisfying that the possibility to breach any sensitive information from drones' communication is negligible.

The results will be introduced in three dimensions these are; the numeric value of hashing measures for Stansted SHA-1 and the proposed 83SHA-1. As shown in Table 1. The numeric value of the statistical test for secret-key of standard and proposed key generation in GG algorithm. Table 2 shows the numeric value of the statistical test for secret-key of standard and proposed key generation/encryption in DRRCBC algorithm. As shown in Table 3 and Table 4 the rate of difference (improvement) for secret-keys randomness in the proposed GGD algorithm is estimated at approximately (18%) at an average rate. The rate of difference (improvement) for secret-keys randomness in the proposed RRCBC algorithm is estimated at approximately (14%) at an average rate. Although the percentage rate of difference (improvement) for the secret keys of the proposed algorithm is (18%), it goes down to approximately (14%) for the ciphertext, due to using fixed plaintext which negatively affects the ratio.

Table 2. Measures comparisons between SHA-1 and 83SHA-1 [7]

Measures	Traditional SHA-1	Modified SHA-1
Block length	512 after padding	512 after padding
Numbers of rounds	80	83
No. of word	80	83
Digest length (bit)	160 bit	160 bit
Internal state size (bit)	160 (5*32)	160(5*32)
Max message size (bit)	$(2^{64})-1$	$(2^{64})-1$
Operations	And, Xor, Rot, Add (mod 2^{32}), Or	And, Xor, Rot, Add (mod 2^{32}), Or, expansion, pruning
Security (bit)	<80 (theoretical attack in (2^{61}) operations)	<83 (theoretical attack in (2^{70}) operations)
Example Performance (MiB/s)	192	145

Table 3. Statistical tests comparisons between CBC and RRCBC

#	Frequency		Run of Zero		Run of One		Pocker		serial		Auto_Correlation for N=5	
	CB	RR	CBC	RR	CBC	RR	CBC	RR	CBC	RR	CBC	RR
1	1	1.005	7.254	7.06	11.32	14.128	7.942	6.452	1.707	2.522	2.439	0.274
2	2.4 92	2.63	2.863	2.429	12.155	14.215	6.259	2.8	2.263	2.87	0.093	0.05
3	2.8 59	4.565	2.916	3.255	22.524	14.128	6.676	6.452	3.722	2.702	3.492	0.2
4	1.8 2	1.114	11.045	2.568	18.389	15.903	6.049	4.382	5.545	3.809	1.671	1.316
5	1.2 3	1.022	0.923	1.854	9.435	16.475	10.239	6.667	1.522	1.822	5.753	0.051
6	1.4 22	0.052	11.86	9.667	2.572	12.026	5.778	3.502	1.467	1.899	1.651	2.381
7	0.1 43	0.809	8.257	4.48	10.131	3.179	3.491	3.802	3.969	4.258	0.588	0.006
8	0.3 6	0.138	4.303	3.317	10.705	1.549	9.825	3.272	1.472	1.461	0.006	0.023
9	0.8 28	0.006	8.049	11.602	19.717	19.891	3.582	4.812	1.966	3.89	4.976	0.093
10	0.9 44	0.8	9.899	12.527	10.076	7.638	4.057	4.356	1.913	2	1.862	0.691
11	0.0 89	1.82	3.994	4.817	15.372	2.75	1.6	2.791	0.133	2.191	0.463	0.468
12	0.8 37	0.089	4.731	8.172	13.238	1.416	4.633	5.222	4.605	1.289	5.754	0.28

Table 4. Statistical tests comparisons between Geffe and GG

#	Frequency		Run of Zero		Run of One		Pocker		Serial		AutoCorrelation for N=5	
	Geffe	GG	Geffe	GG	Geffe	GG	Geffe	GG	Geffe	GG	Geffe	GG
1	4.073	0.448	1.931	5.328	14.592	8.864	9.152	3.36	4.684	2.169	0.368	1.114
2	0.588	1.455	7.326	3.523	16.565	9.034	8.165	6.018	4.082	3.909	1.024	1.016
3	0.209	0.006	11.801	13.107	17.052	9.924	5.377	14.94	1.163	1.096	2.641	1.841
4	0.429	1.286	15.884	7.467	14.086	13.81	5.197	5.777	2.077	1.683	3.674	1.906
5	3.13	3.13	4.13	3.299	15.247	12.389	8.794	6.713	3.127	3.217	0.22	0.944
6	0.479	1.93	5.598	3.903	8.669	17.346	6.522	3.101	3.885	0.95	2.439	1.407
7	3.023	0.847	0.787	4.456	24.097	15.154	13.091	7.224	3.329	3.706	2.847	0.733
8	0.143	1.994	1.137	7.471	4.931	9.074	3.126	3.896	1.5	2.914	0.212	2.75
9	0.138	0.448	13.074	11.018	14.035	8.146	4.863	5.57	2.345	0.931	0.006	0.364
10	2.782	0.271	0.761	3.571	13.106	6.753	10.202	11.526	3.437	0.843	0.29	0.568
11	2.63	0.023	10.823	6.398	1.78	8.943	6.017	3.2	4.783	0.273	0.453	2.579
12	0.089	2.087	12.483	3.031	8.83	11.198	3.733	3.629	2.622	3.194	0.051	0.024

5. CONCLUSION

This paper proposes a secure communication for civil drones system that uses the different cryptographic protocols and runs over unsecured drone communication channels. The proposed protocol introduces an enhancement for privacy and security in drone systems. These enhancements come up with some novelties by introducing cryptographic modification techniques. The suggested 83 SHA-1 protocol increases the randomness and the complexity of the proposed system by increasing the number of rounds up to 83 and adding DES expansion and S-boxes operations in each round. The GG encryption method is fast and simple in implementation since it doesn't need any complex calculations. The private keys are kept secret meanwhile the secret keys are not used directly in encryption but will be processed using a genetic algorithm. An adversary cannot recognise the secret keys as not all the original secret keys will be used to make the encryption. The RRCBC encryption method increases the randomness since in every time the encryption occurs, the initialization vector (IV) is generated so the secret messages will be protected against the chosen-plaintext attacks. Multiple secret keys are used to provide the randomness to the generated key. As a result, the secret messages will be kept safe against the chosen-ciphertext attacks.





REFERENCES

[1] R. N. Akram *et al.*, "Security, privacy and safety evaluation of dynamic and static fleets of drones," in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, Sep. 2017, vol. 2017-Sept, pp. 1–12, doi: 10.1109/DASC.2017.8101984.
 [2] G. Samid, "Drone Targeted Cryptography.," in *IACR Cryptology ePrint Archive*, 2016, vol. 2016, p. 499, [Online]. Available: <http://dblp.uni-trier.de/db/journals/iacr/iacr2016.html#Samid16a>.





- [3] J. H. Cheon *et al.*, "Toward a secure drone system: flying with real-time homomorphic authenticated encryption," *IEEE Access*, vol. 6, pp. 24325–24339, 2018, doi: 10.1109/ACCESS.2018.2819189.
- [4] B. Nassi, R. Ben-Netanel, A. Shamir, and Y. Elovici, "Drones' cryptanalysis - Smashing cryptography with a flicker," in *Proceedings-IEEE Symposium on Security and Privacy*, May 2019, vol. 2019-May, pp. 1397–1414, doi: 10.1109/SP.2019.00051.
- [5] M. O. Ozmen, R. Behnia, and A. A. Yavuz, "IoD-Crypt: A lightweight cryptographic framework for internet of drones," *IEEE Transactions on Services Computing*, Apr. 2019, [Online]. Available: <http://arxiv.org/abs/1904.06829>.
- [6] M. O. Ozmen and A. A. Yavuz, "Dronecrypt - an efficient cryptographic framework for small aerial drones," in *Proceedings - IEEE Military Communications Conference MILCOM*, Oct. 2019, vol. 2019-October, pp. 971–976, doi: 10.1109/MILCOM.2018.8599784.
- [7] E. Tabassi and P. Grother, "Biometric Sample Quality," in *Encyclopedia of Biometrics*, Boston, MA: Springer US, 2015, pp. 194–206.
- [8] B. Abood, A. N. Faisal, and Q. A. Hamed, "Data transmitted encryption for clustering protocol in heterogeneous wireless sensor networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 1, p. 347, Jan. 2022, doi: 10.11591/ijeecs.v25.i1.pp347-357.
- [9] S. H. Hashem, "Proposed encryption method based Geffe generator, genetic algorithm and DNA coding," *Journal Of Madent Alelem College*, vol. 10, no. 2, pp. 42–57, 2018.
- [10] S. H. Hashem, "Proposal hybrid CBC encryption system to protect E-mail messages," *Iraqi Journal of Science*, vol. 60, no. 1, pp. 157–170, 2019, doi: 10.24996/ij.s.2019.60.1.17.
- [11] Z. N. Al-Khateeb and M. Jader, "Encryption and hiding text using DNA coding and hyperchaotic system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 2, p. 766, Aug. 2020, doi: 10.11591/ijeecs.v19.i2.pp766-774.
- [12] M. O. Arowolo, M. O. Adebisi, A. A. Adebisi, and O. J. Okesola, "Predicting RNA-Seq data using genetic algorithm and ensemble classification algorithms," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 2, pp. 1073–1081, Feb. 2020, doi: 10.11591/ijeecs.v21.i2.pp1073-1081.
- [13] S. A. K. Albermany, D. Amer, Sawzan, and Kamal, "S-RADG: A stream cipher RADG cryptography," *Journal of Engineering and Applied Sciences*, vol. 13, no. Specialissue1, pp. 2317–2321, 2018, doi: 10.3923/jeasci.2018.2317.2321.
- [14] S. H. Hashem, "Enhance SHA-1 for Building Secure System to Transfer Voice Files," *Al-Mustansiriyah Journal of Science*, vol. 19, no. 1, pp. 53–63, 2008.
- [15] C. Tan, X. Deng, and L. Zhang, "Identification of block ciphers under CBC mode," *Procedia Computer Science*, vol. 131, pp. 65–71, 2018, doi: 10.1016/j.procs.2018.04.186.
- [16] N. A. Kako, H. T. Sadeeq, and A. R. Abraham, "New symmetric key cipher capable of digraph to single letter conversion utilizing binary system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 1028–1034, May 2020, doi: 10.11591/IJEECS.V18.I2.PP1028-1034.
- [17] S. Deb, B. Bhuyan, and N. C. Gupta, "Design and Analysis of LFSR-based stream cipher," in *Lecture Notes in Networks and Systems*, vol. 24, 2018, pp. 631–639.
- [18] S. N. Sivanandam and S. N. Deepa, "Introduction to genetic algorithms," in *Principles of Adaptive Filters and Self-learning Systems*, no. 9781852339845, London: Springer-Verlag, 2005, pp. 325–336.
- [19] M. S. A. Forhad, M. S. Hossain, M. O. Rahman, M. M. Rahaman, M. M. Haque, and M. K. H. Patwary, "An improved fitness function for automated cryptanalysis using genetic algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 2, pp. 643–648, Feb. 2019, doi: 10.11591/ijeecs.v13.i2.pp643-648.
- [20] S. Aswad Mohammed, O. Ali Awad, and A. Merhej Radhi, "Optimization of energy consumption and thermal comfort for intelligent building management system using genetic algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 3, p. 1613, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1613-1625.
- [21] Y. Belgaid, M. Helaimi, R. Taleb, and M. B. Youcef, "Optimal tuning of PI controller using genetic algorithm for wind turbine application," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 1, p. 167, Apr. 2020, doi: 10.11591/ijeecs.v18.i1.pp167-178.
- [22] A. H. Al-Hamami, M. A. Al-Hamami, and S. H. Hashem, "A proposed modifications to improve the performance of blowfish cryptography algorithm," in *First National Information Technology Symposium (NITS 2006)*, 2006, pp. 5–7.
- [23] S. Abed, L. Waleed, G. Aldamkhi, and K. Hadi, "Enhancement in data security and integrity using minhash technique," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1739–1750, Mar. 2021, doi: 10.11591/ijeecs.v21.i3.pp1739-1750.
- [24] A. S. Abd and E. A. R. Hussein, "Design secure multi-level communication system based on duffing chaotic map and steganography," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 1, p. 238, Jan. 2022, doi: 10.11591/ijeecs.v25.i1.pp238-246.
- [25] Pronika and S. S. Tyagi, "Performance analysis of encryption and decryption algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1030–1038, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp1030-1038.

BIOGRAPHIES OF AUTHORS







Dr. Ayad Al-Adhami     received the MSc degree in 2008 specialised in data security. The PhD degree earned in 2018 and specified in computing (Computer security) Plymouth University, United Kingdom. Area of study are RFID security, Authentication & privacy, Cryptography and Information security. He can be contacted at email: ayad.h.ibrahim@uotechnology.edu.iq.







Dr. Rajaa K. Hasoun     received the MSc degree in 1999 specialised in data security. The PhD degree earned in 2018 and specified in data security from University of Technology, Baghdad, Iraq. Area of study are, Biometric, Authentication, Cryptography, image processing and Information security. She can be contacted at email: dr.rajaa@uoitc.edu.iq.



Dr. Ekhlas Khalaf Gbashi     earned her Ph.D. in networks security from the Department of computer sciences at the Technology University. Ekhlas earned her bachelor's and master's degree in computer sciences from the University of Technology (UOT), Baghdad, Iraq in 1998, 2005. Ekhlas is a faculty member in the computer sciences Department at the University of Technology (UOT) since 2000; where she became a Head of computer security Branch at the UOT from 2016 until 2020. Her research interested focus on in networks security (intrusion detection system), data security, computer networks, Comparative Education and Computer Architecture, image processing and AI. She can be contacted at email: Ekhlas.K.Gbashi@uotechnology.edu.iq.



Dr. Soukaena Hassan Hashem     earned her Ph.D. in networks security from the Department of computer sciences at the Technology University. earned her bachelor's and master's degree in computer sciences from the University of Technology (UOT), Baghdad, Iraq in 2000, 2002. She can be contacted at email: soukaena.h.hashem@uotechnology.edu.iq.