STAP
Smart Technologies Academic Press®

Journal of Cyber Security and Risk Auditing

https://www.jcsra.thestap.com/

JOURNAL OF CYBER SECURITY AND RISK AUDITING
Smart Technologies Academic Press
STAP

A Hybrid Genetic Algorithm and Hidden Markov Model-Based Hashing Technique for Robust Data Security

Aseel AlShuaibi[1], Muhammad Waqas Arshad[2], Mohammed Maayah[3]

[1]Department of Computer Networks and Communications, King Faisal University, Al-Ahsa 31982, Saudi Arabia
[2] Department of Computer Science and Engineering, University of Bologna, Italy
[3]Fellowship Researcher, INTI International University, Nilai 71800, Malaysia

ARTICLE INFO

Article History

Received: 29-03-2025
Revised: 04-05-2025
Accepted: 07-05-2025
Published: 08-05-2025

Academic Editor:

Prof. Youakim Badr

Vol.2025, No.3

DOI:
https://doi.org/10.63180/jcsra.thestap.2025.3.6

ABSTRACT

The growing dependence on technology to store, process, and transmit data across interconnected systems has significantly elevated the need for robust data security. Modern computer systems emphasize the critical principles of authentication and data integrity. With the rise in cyber threats, the importance of securing data transactions against unauthorized or unintentional modifications has become more apparent than ever. As computers continue to play an increasingly vital role in daily operations, managing and safeguarding data alterations is essential. To address these challenges, businesses must adopt proactive measures to reinforce the security of sensitive data and passwords. Hashing functions, a well-established cryptographic approach, have proven effective in addressing a wide range of authentication and data integrity issues. A hash function generates a fixed-length output, known as a "digest," from an input. This one-way function is irreversible, providing a secure method of encoding data. However, hash functions are still vulnerable to various attacks, including dictionary attacks, brute-force attacks, and the use of lookup tables. The strength of a hashing function can be evaluated based on the number of attempts required to break it, the size of the hash key, and the specific algorithm employed. In response, this study proposes a novel hashing technique that integrates a Genetic Algorithm (GA) and Hidden Markov Models within a block hashing framework. Inspired by evolutionary biology, the GA applies operators such as mutation, crossover (recombination), and selection to simulate natural selection, offering a dynamic and efficient method for enhancing data security. The proposed algorithm utilizes the Hill cipher as an encryption mechanism and incorporates a singular Hill cipher key matrix to enhance security and reduce the likelihood of hash collisions or reversals. Experimental results demonstrate that the developed algorithm exhibits strong resistance to multiple attack types and outperforms several existing methods in terms of accuracy and robustness.

Keywords: Cryptography, Genetic Algorithm, Hash Function, Encryption, Cybersecurity, Data Protection, Hidden Markov, Hill Cipher, Plaintext, Ciphertext, Key, Fitness Function, Algorithms, Metrics.

How to cite the article
Alshuaibi, A., Arshad, M. W., & Maayah, M. (2025). A Hybrid Genetic Algorithm and Hidden Markov Model-Based Hashing Technique for Robust Data Security. Journal of Cyber Security and Risk Auditing, 2025(3), 42–56. https://doi.org/10.63180/jcsra.thestap.2025.3.6

**Corresponding author. Email:* m.almaiah@ju.edu.jo

1. Introduction

With the increasing reliance on technology to process and transmit data across interconnected systems, the demand for robust security mechanisms has grown significantly. Cryptography plays a crucial role in ensuring data security by employing various computational techniques to achieve confidentiality, integrity, authentication, and non-repudiation [1]. As cybersecurity continues to be a pressing concern in the computing field, innovative techniques are continually emerging to counter evolving threats. One such technique is the use of cryptographic hash functions, which will be introduced and discussed in this section. Furthermore, the paper explores how genetic algorithms can be utilized to enhance the security and resilience of these hash functions. According to IBM’s Cost of a Data Breach Report, the average global cost of a data breach rose from $4.24 million in 2021 to $4.35 million in 2022, highlighting the growing impact of cyber threats. This paper presents a comprehensive discussion that begins with the foundational background of the proposed algorithm, followed by a review of related work, the adopted methodology, and a detailed analysis of implementation results.

1.1 Background

Hash functions are a fundamental cryptographic technique widely used to address authentication and data integrity challenges. A hash function processes an input and generates a fixed-length output known as a "digest." As a one-way function, it ensures that the original input cannot be derived from the digest, thereby providing a layer of security. The input typically includes both the data message and cryptographic keys, with the objective of selecting the optimal key combination to maximize security and minimize vulnerabilities such as collisions and key breaches.

Despite their effectiveness, traditional hash functions are susceptible to several attack vectors, including dictionary attacks, brute-force attacks, and the use of lookup tables. The resilience of a hash function against such attacks is generally assessed based on factors such as the number of attempts required to break it and the robustness of the underlying algorithm. Commonly used hash functions—such as MD5, SHA-1, Whirlpool, and RIPEMD-160—have demonstrated various degrees of vulnerability to attacks like birthday attacks, rebound attacks, and theoretical cryptanalysis.

To address these limitations, this work proposes a novel hashing mechanism that integrates Genetic Algorithm (GA) operators with Hidden Markov Model (HMM) features to generate secure and adaptive hashing keys. Genetic Algorithms, inspired by the principles of natural selection and biological evolution, provide an effective solution for both constrained and unconstrained optimization problems. GA evolves a population of candidate solutions using operators such as mutation, crossover (recombination), and selection, which enhance its dynamic and powerful search capability.

In this context, GA is employed to identify the most optimal and robust key for the hash function. Furthermore, it is leveraged to construct a more secure hashing algorithm tailored for protecting data at rest. By harnessing the adaptive nature of GA and the probabilistic modeling of HMMs, the proposed hash function aims to overcome existing vulnerabilities and provide enhanced resistance against a wide range of cryptographic attacks [2].

1.2 Problem Statement

The hash function converts a variable-length plaintext into a fixed-length hash value, and it's frequently employed in data signing and authentication. Moreover, the use of hash functions is well known to those who work in the field of cryptography. Although hash functions have been providing necessary security services for a long time and are difficult to crack, there are still major issues that directly or indirectly assist cracker in breaking the security of hash functions or encoding functions to some extent. On different hash functions, numerous types of attacks are possible. A secure hash function must be one-way, safe against various types of attacks such as, birthday attacks, brute-force attack, pre-image attack, collision attack, crypto analysis attack, and meet-in-the-middle attacks. Finding a plaintext with the appropriate hash value is difficult due to the one-way feature. However, the hash function should be resistant to birthday attacks, making it difficult to discover two plaintexts that have the same hash value. It should also be resistant to a meet-in-the-middle attack, in which finding a plaintext with the same hash value as one of the provided plaintexts is challenging. In recent years, hash methods like MD5 and SHA-1, which are commonly used, are not secure enough and vulnerable to many attacks such as birthday attacks [3]. Thus, a new hash function should be investigated and developed to suit the practical applications.

1.3 Scope

The paper develops and validates a new one-way hash function based on genetic algorithm and hidden-Markova. The new hash function takes advantage of GA operators such as mutation, recombination (crossover), and choice/selection that provide GA a strong search algorithm. Moreover, manipulated the GA fitness function based on hidden-Markova to develop a secure hash function for data at rest.

2. Related Works

Security is a large and enormous field. Many researchers are attempting to attain security and privacy in the digital and data field. Researchers developed cryptography with several evolutionary algorithms. The section presents a review of some existing hash functions, and earlier work related to the robustness of cryptography done by professionals and researchers.

Gupta, S., Goyal, N., & Aggarwal, K. [3] analyzed one of the widely known cryptographic hash functions which is MD5 (Message Digest 5). In MD5 function, the input message is decomposed into 512-bit blocks. If the message's length is not an integer multiple of 512-bit blocks, it is padded to make it divisible by 512. MD5 is made up of 64 operations, which are divided into four rounds of 16 operations each. F is a non-linear function; each round uses a different function. Ki indicates a 32-bit constant that is unique for each operation, and Mi indicates a 32-bit block of the message input. The letter s indicates the left bit rotation. The MD5 function works with a 128-bit state that is separated into four 32-bit words, A, B, C, and D. These are defined as a set of predetermined constants. The main function then modifies the state by using each 512-bit message block in turn. A message block is processed in four rounds, each of which consists of 16 operations based on a non-linear function F, modular addition, and left rotation. Figure 1 presents a process of one operation in the round.

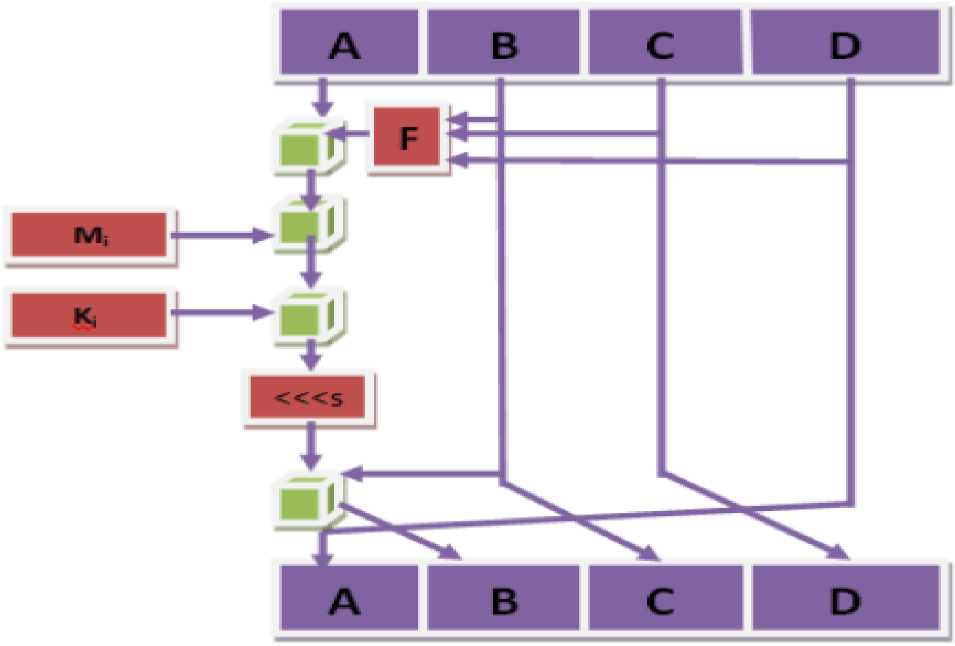

Figure 1. MD5 Function [3]

However, MD5 can quickly compromised by simply find changes that can be made to a payload's end or start to make it look legitimate. Moreover, it can be easily attacked by Birthday Attack, since it produces a lot of collisions (The exact output value for two different inputs). Similarly, Maetouq, A., Daud, S., Ahmad, N., Maarop, N., Sjarif, N. N. A., & Abas, H. [4] confirmed that the collision attack was discovered on RIPEMD-160 hash function as well as preimage attack. RIPEMD-160 is a widely used hash function. It generates a message digest that is 160 bits long and its block size 512 bits. Also, it consists of 80 rounds and 6 operations.

Moreover, Kitsos, P., & Koufopavlou, O. [5] discussed the Whirlpool, which is a one-way, 512-bit hash algorithm that works on messages that are less than 2256 bits long. It is made up of the iterative execution of a compression algorithm that is based on an underlying specialized 512-bit block cipher with a 512-bit key. However, Whirlpool can be easily attacked by Birthday Attack, since it produces a lot of collisions as well as Rebound attack. Additionally, SHA-1 (Secure Hash Algorithm 1) described by Pittalia, P. P. [6], which is a hash function that accepts an input and generates a message digest of 160 bits. It helps to determine whether modifications occurred or not during the transfer of data from the sender to the receiver. Figure 2 illustrates the process of one iteration within SHA-1 compression process.

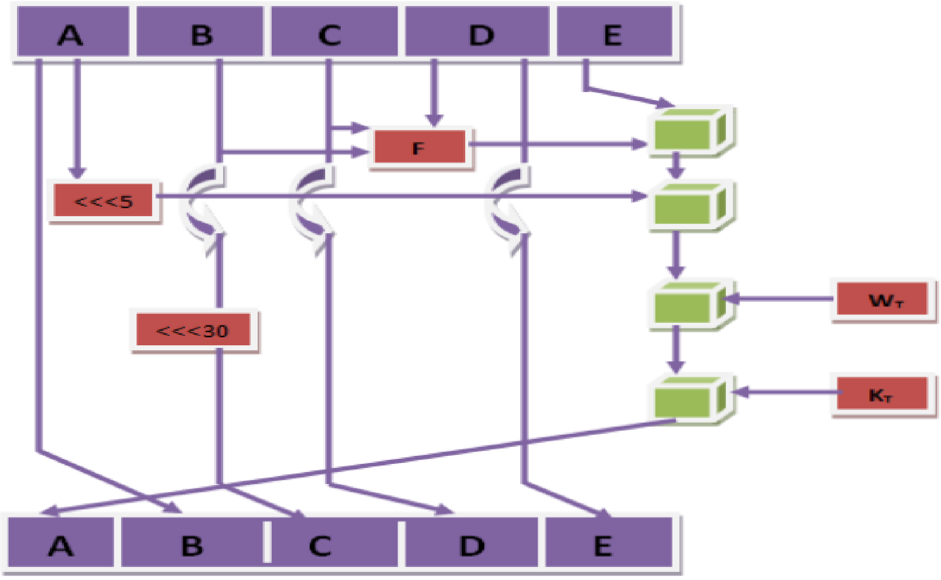

Figure 2. SHA-1 Process [3]

In Figure 2, A, B, C, D, E are 32-bit condition words. F is a changing nonlinear function. Left shiftn refers to a left bit rotation of n places; n will be different in each operation. The round t extended message word is Wt; and the round constant is Kt. The cube indicates the addition modulo 232. However, SHA-1 can be attacked by Theoretical attack and Birthday attack, since it produces a lot of collisions

Pujari, S.K., Bhattacharjee, G., Bhoi, S. [7] presented a hybridized technique that combines DNA sequences with genetic algorithms for image encryption. The robustness of the DNA Sequence and GA-based technique was demonstrated in an experiment. Just as in [8], the authors suggested an image encryption technique based on DNA sequences that employs GA ideas to choose the optimal cipher image. Moreover, Jiang, C., Pang, Y., & Wu, A. [9] proposed an image-hashing approach based on a Back Propagation (BP) Neural Network and a genetic algorithm (GA) for content authentication. To build the image feature matrix, the lifting wavelet transform is employed to extract image low-frequency coefficients. To build image-hashing code, a GA-BP network model is built. Experiments show that the suggested hashing approach is resistant to random attacks, JPEG compression, additive Gaussian noise, and other types of noise. The suggested method greatly outperforms previous ways for resilient image hashing, according to a receiver operating characteristics (ROC) study of a large image database. Similarly, Biswas, K., Muthukkumarasamy, V., & Singh, K. [10] suggested a wireless sensor network image encryption based on an N-logistic tent map and genetic algorithms. They employed mutation and two-point crossover. Moreover, the authors of [11] suggested an image encryption technique based on DNA sequences that employs Genetic Algorithm ideas to choose the optimal cipher image. In [12], the authors proposed an image transmission method to ensure confidentiality and less complexity in a secure manner using the Hidden Markov Model and Genetic algorithm. However, in [13], the research demonstrated that a known-plaintext attack (which included a divide-and-conquer attack) could be used to break the GA-based image encryption. Also, the focus of previous studies is entirely on image encryption and hashing not data and text.

On the other hand, Rodríguez, J., Corredor, B., & Suarez, C. [14] proposed a symmetric-key text encryption method based on Genetic Algorithms concept, entropy, and modular arithmetic. Over a deterministic system, an experimental technique is utilized to redistribute and modify the parameters and phases of the evolutionary algorithm that directly impact its behavior, while performing a continual evaluation using the fitness function to improve the outcomes. The auxiliary key is encrypted separately from the main key. In that case, the main key is used to improve the security. As a result, the encryption method has better performance than DES, RSA, and AES. In the same way, Sheeja, S. [15] introduced a feistel cipher method based on DNA sequences. In this work, each round in the feistel cipher has a sub key, sub keys are generated using a genetic algorithm, which has been demonstrated to improve the selection of the best-fit keys. However, the proposed method is slower than the traditional symmetric key algorithm because of the round functions used. Similarly, the authors in [16] created a GA-based encryption systems that provide network message authentication, confidentiality, non-repudiation, and integrity. This study uses a novel approach that is based on the Crypto-GA Genetic Algorithm (GA) to address data security and privacy problems. However, since the previous studies are symmetric encryption techniques, the same key will be used for both encryption and decryption. Thus, the key is vulnerable to disclosure by man in the middle attack.

Furthermore, in [17], Nazeer, M. I., Mallah, G. A., Shaikh, N. A., Bhatra, R., Memon, R. A., & Mangrio, M. I. discussed the GA implication in in cryptography in improving the security. Several stages are used to encrypt data in the proposed approach. The method begins by creating the key using a random number generator and genetic processes. The data is then diffused using genetic operators, and the data is encrypted using logical operators between the dispersed data and the key. The method improves the security of the encryption key. However, compared to DES and AES, the method has less computational efficiency and longer encryption time. In the same way, to address the primary key distribution, the authors [18] presented a GA-based cryptographic key generation technique. As well as in [19], Jawaid, S., & Jamal, A. employed a GA to create the best-fit keys for a cryptographic system. However, in this work, the authors implement and verify the robustness of the key on DES only, which can result in unreliable results. Moreover, in [20], the authors proposed a novel iterative encryption technology that combines DNA sequences and a genetic algorithm. In the same way, the authors in [21] used DNA sequences to improve the security of the Vigenere encryption. Also, Safdari, M. [22] investigates the use of Genetic Algorithms to build Universal Hash Functions that can effectively hash a set of keys. However, in this work, more efficient chromosome encoding, and a better fitness function design are needed.

Moreover, the authors in [23] define a method that involves two stages. The location of pixels and their values are altered from adjacent pixels in the substitution phase to minimize the correlation between pixels. The input image is encrypted, and the pixel values are modified during the modification phase. Binary patterns are used in both phases. Local binary patterns (LBP) are employed in the substitution phase, while bit plane slicing is used in the modification phase (BPS). A preset key and a random number generator are employed to choose the population. By combining the fitness function, histogram analysis, and entropy, an image can be encrypted quickly and effectively. In this work, the fitness function is applied at each generation, and the entropy is high and quick. However, it increases computational inaccuracy and burden. Similarly, in [24], Al-Husainy, M. A. employs a GA to create a novel encryption method that includes strong areas of crossover and mutation. Furthermore, Soni, A., & Agrawal, S [25] introduced a new method in which a key is created using a pseudorandom number generator, and these random numbers are generated using the computer system's current time. GA is also applied, and image encryption is performed using the [AES] symmetric key technique. The approach will improve efficiency and reduce computation time, while the irregularity of the key will raise the complexity of key generation.

The authors in [26, 27] describe an encryption method based on the NLFFSR (Non-Linear Feed Forward Shift Register) that employs a crossover operator and a pseudorandom sequence generator. The crossover point will be determined by the pseudorandom sequence, resulting in completely encrypted data. Authors in [26] extended the method by using mutation after encryption. The encrypted data can be further concealed within the steno-image. Moreover, authors in [28] introduced a methodology that uses GA to generate pseudo-random numbers. The encryption method is based on the crossover and mutation operator procedure. The author employs a sophisticated technique of mimetic algorithms and pseudorandom binary sequences. Similarly, in [29], they proposed a method based on Pseudo random binary sequence, brain thoughts, GA, and Mu waves concepts. This method of protecting confidential data is extremely safe and reliable.

In [30], the Genetic Algorithm developed as an encryption algorithm, with a secret key for encryption and decryption. Furthermore, Singh, D., Rani, P., & Kumar, R. [31] demonstrate that when a high level of security is required, symmetric and asymmetric methods fail. Therefore, to achieve a perfect result in the shortest amount of time, the author proposed an algorithm in which GA is combined with cryptography, and the output of that combination is an optimal solution.
On the other hand, Bhandari, S. U., Subbaraman, S., Pujari, S., & Mahajan, R. [32] with the use of a physical model that can act on cryptography, the authors utilize a genetic algorithm, image encryption, and video encryption. Signal and image processing are used in the proposed method.

3. Methodology

This section presents the methodology that is used to accomplish the work goals and expectations. The paper introduces a new hash function based on a hidden-Markova model and a genetic algorithm. The new hash function with the power of GA and HMM will be robust against a variety of existing hash function attacks, including birthday attacks, rebound attacks, and theoretical attacks. The GA characteristics will be utilized to select the best and most robust key before constructing the robust hash function. Furthermore, utilizing the GA algorithm in hashing functions eliminates the need to destroy the hash key because it is held in a safe way that cannot be detected or broken because the key matrix used is not reversible. The probability of breaking the hash keys is used to determine the robustness, the measurement of probability done by manipulating the fitness function inside the GA with the HMM.

The new hash function is based on the robustness of genetic algorithm and hidden-Markova model. The GA processes are used to choose the optimal key size and the optimal fit key. The best key size for the encryption should be the same size of the plaintext but consider huge size plaintext (for example: MS Word file), the key with huge size will not be efficient for fast hashing algorithm. Thus, the main goal of GA is to choose the optimal key for any size of plaintext to be used in the hashing algorithm. The fitness function inside the GA is manipulated using hidden Markova that is chosen to calculate the probability of retrieving the plaintext using the key generated, therefore, the robustness of the key is measured based on the uniform probability that the key can be found. Then, the hill cipher encryption algorithm is used as a hashing function. The Hill cipher is a cipher algorithm based on Linear Algebra ideas.

It is more mathematical than others since it uses modulo arithmetic, matrix multiplication, and matrix inverses. Since the Hill cipher is also a block cipher, it may theoretically be used with blocks of any size. In the project, only the encryption process of the hill cypher is used since the algorithm is one-way hash function. The Hill cipher's encryption algorithm is based on the following operation (1):

$$E\ (Non-Invertible\ Key\ Matrix, P) = (Non-Invertible\ Key\ Matrix * P)\ mod\ 26 \qquad (1)$$

Where P is the plaintext and Non-Invertible Key Matrix is the proposed chromosome from GA. These two terms are multiplied in a matrix to produce the encrypted cipher text. In the work, the condition to make the plaintext non reversible is to accept only singular or quasi singular matrix. By using a singular matrix, the plaintext cannot be reversed by authorized or unauthorized recipients or eavesdroppers if the key is a singular matrix [39]. We cannot define P= Non-Invertible Key Matrix $^{-1}$ E (since there is no Non-Invertible Key Matrix $^{-1}$). As a result, we can say that the Hill Cipher is very secure when employing singular matrix (non-invertible matrix) keys. Figure 3 illustrates the process of Hill Cipher.

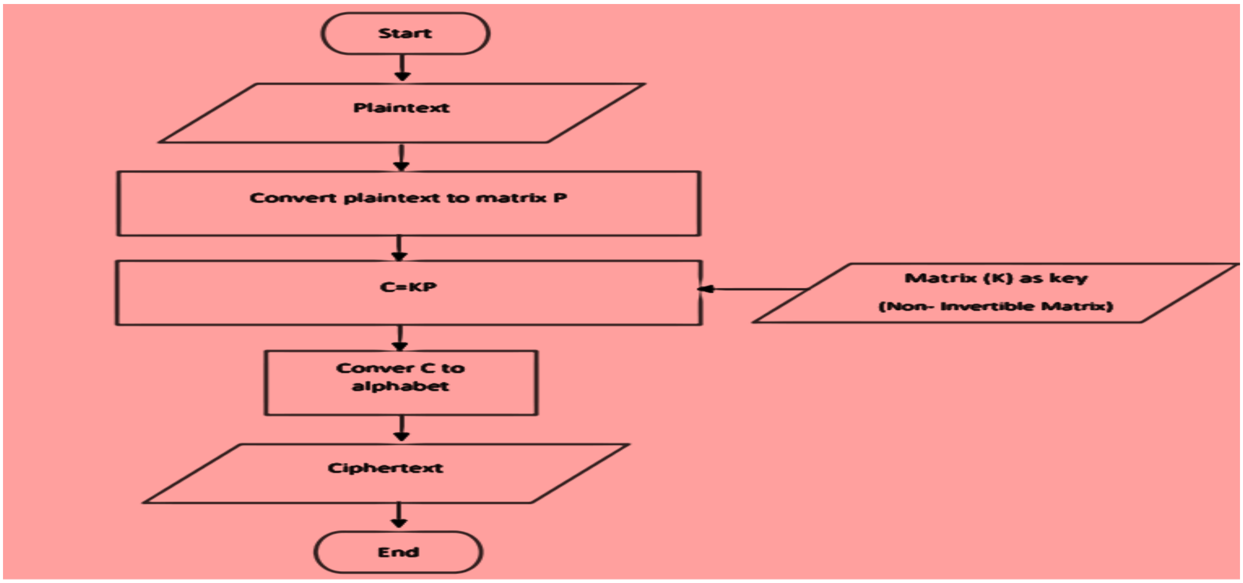

Figure 3. Hill Cipher Flowchart

However, the input to the hill cipher in the GA hash function is the singular key matrix from the GA and the plaintext, hill cipher inside the fitness function will encrypt the data and produce the cipher text, then the HMM will calculate the probability of occurrence of the keys. This process is continuous until the robust key matrix (key matrix with uniform probability of occurrence of key) is found by the GA. The methodology of the work is illustrated in Figure 4.

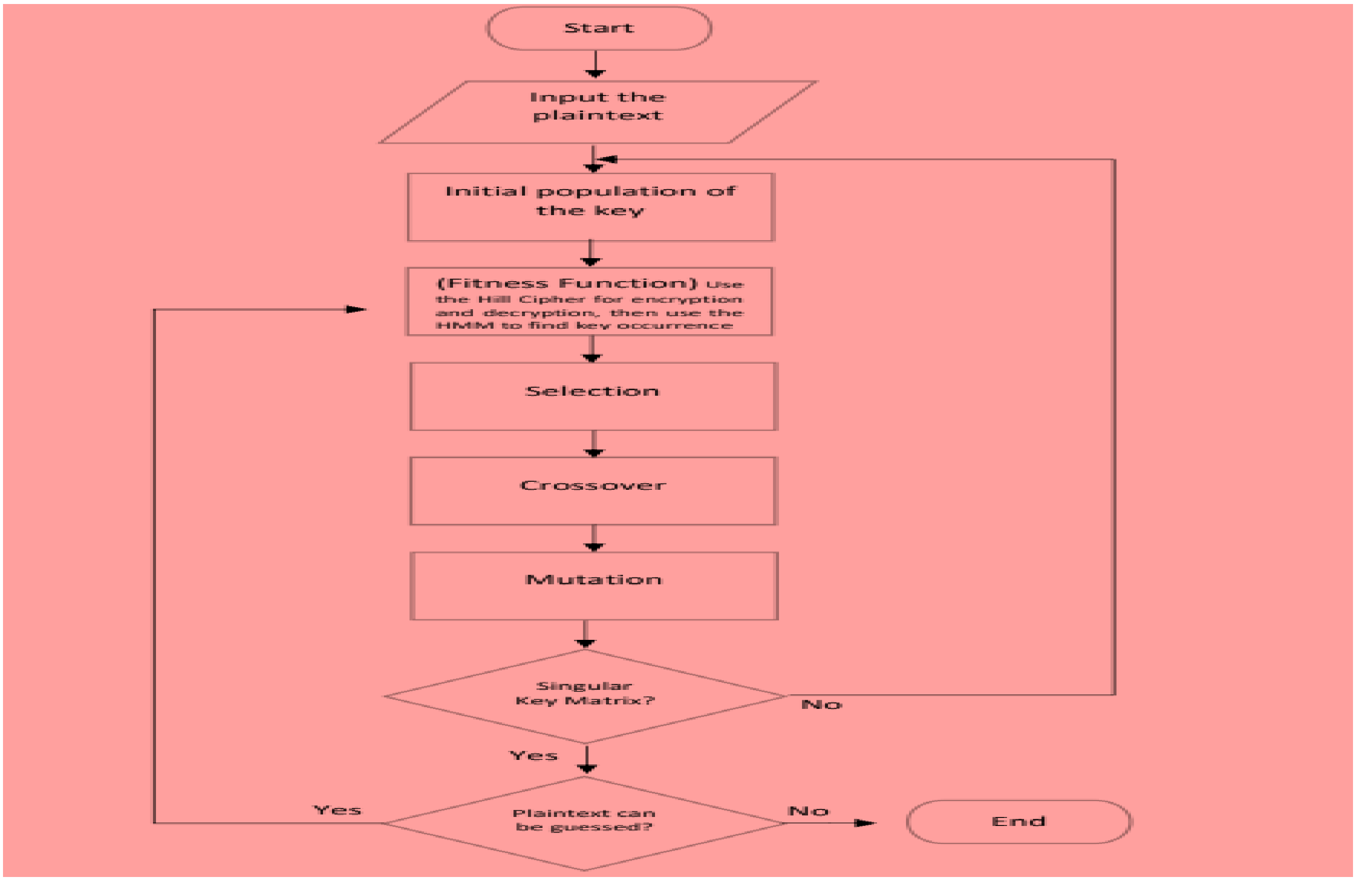
Start
Input the plaintext
Initial population of the key
(Fitness Function) Use the Hill Cipher for encryption and decryption, then use the HMM to find key occurrence
Selection
Crossover
Mutation
Singular Key Matrix?
No
Yes
Yes
Plaintext can be guessed?
No
End

Figure 4. Methodology

4. Implementation and Results

In this study, the implementation process for each method and the results of the work are clarified. Genetic Algorithm updates a population of keys iteratively. The keys are evaluated every iteration using a fitness function. In the fitness function, the plaintext and the key should have the same size to increase the encryption robustness, if the size is different, zeros are added to make the plaintext and the key equal, then, each key goes through the hill cipher encryption of the plaintext and decryption of the ciphertext. Then the probability of key occurrence is measured by the Hidden Markov. By probabilistically choosing best individuals from the current generation, a new generation of the population is created. To produce new offspring, some are subjected to genetic operators including crossover and mutation.

4.1 Pseudocode of the proposed hash function

Where n is the number of keys in each population; x is the fraction of the population to be replaced by a crossover in each iteration; and m is the mutation rate. The hidden Markov probability can be utilized as a fitness function within the genetic algorithm to determine the most effective encryption key by following these steps:

Step1. Define the chromosome: First, define the chromosome representation in the genetic algorithm. In this case, an encryption key can be represented as a string of characters or binary values.
Step 2. Generate an initial population: Create an initial population of random encryption keys. Each encryption key serves as a potential solution.
Step 3. Evaluate the fitness: Evaluate the fitness of each individual in the population using the hidden Markov probability. This involves using the encryption key to encode or encrypt a known plaintext, and then calculating the probability of the resulting ciphertext using the hidden Markov model. The lower the probability, the more effective the encryption key.
Step 4. Select parents: Perform selection by using a selection method such as tournament selection or fitness proportionate selection, where individuals with lower fitness have a higher probability of being selected as parents for reproduction.
Step 5. Apply genetic operators: Apply genetic operators such as crossover and mutation to the selected parents to create offspring. Crossover involves combining parts of the parents' encryption keys, while mutation introduces small random changes to the offspring's encryption key.
Step 6. Evaluate offspring fitness: Evaluate the fitness of the offspring by applying the hidden Markov probability model, similar to step 3.
Step 7. Select the next generation: Select the individuals for the next generation based on a combination of the parents and offspring, while also considering elitism (keeping the best individuals from the current generation).
Step 8. Repeat steps 4-7 for multiple generations: Continue the process for multiple generations or until a termination condition is met (e.g., a maximum number of generations reached or fitness plateau).
Step 9. Output the best solution: After the termination condition is met, output the best solution, i.e., the encryption key with the lowest fitness.
Thus, by using the hidden Markov probability as the fitness function and iteratively applying the principles of genetic algorithms, this process seeks to converge toward an encryption key that has a high likelihood of producing secure and effective encryption.

```
Algorithm: GA(n, x, m)
 // Initialize generation 0:
 k := 0;
 $P_{k:}$ = a population of n randomly generated key matrix of size 16×16;
 // Evaluate $P_{k,}$ using hill cipher encryption and decryption, and evaluate the probability keys occurrence:
 Compute fitness(i) for each i ∈ $P_k$;
 Do
 {          // Create generation k + 1:
     // 1. Select:
     Select (1 − x) × n members of $P_k$ and insert into $P_k$+1;
     // 2. Crossover:
     Select x × n members of Pk; pair them up; produce offspring; insert the offspring into $P_k$+1;
     // 3. Mutate:
     Select m × n members of $P_k$+1; invert a randomly selected bit in each;
     // Evaluate $P_k$+1:
     Compute fitness(i) for each i ∈ $P_k$;
     // Increment:
     K: = k + 1;
 }
 While the probability of HMM is not uniform & the matrix is not quasi-singular or singular;
 Return the fittest individual from $P_k$;
```

4.2 Robustness Metrics

This section specifies the metrics used to ensure algorithm robustness.

• Key Generation
The key generation process involves creating a population of potential encryption keys. Each key is represented as a matrix of appropriate dimensions. In this algorithm, 3x3 matrices are used to represent keys, as the Hill cipher operates on 3-character blocks.
• Genetic Algorithm
The genetic algorithm employs the principles of evolution and natural selection to improve the encryption keys. During each iteration, the algorithm selects key matrices with lower fitness scores and combines their characteristics to produce offspring keys.
• Fitness Function
The fitness function plays a crucial role in evaluating the effectiveness of encryption keys. It measures the similarity between the generated ciphertext and statistically expected patterns in natural language. A lower fitness value indicates a better key, as the algorithm aims to minimize this value. The fitness function is customized to include a hidden Markov model (HMM) component to enhance its ability to assess key quality. The customized fitness function can be represented as follows (2):
Where:
Fitness: fitness score of the encryption key.
decryptedMessage: message decrypted using the current key.
numericalMessage: original numerical message.

However, a lower fitness score indicates a better key, as the algorithm aims to minimize the sum of absolute differences between the decrypted message and the original numerical message.

4.3 Performance Metrics

Several performance metrics are calculated, including accuracy, precision, recall, and F1-score. Performance metrics such as accuracy, precision, recall, and F1-score are calculated based on the confusion matrix, which contains four values: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).
Where:
TP = sum (encrypted_message == plaintext);
TN = sum(encrypted_message ~= plaintext & ~isletter(encrypted_message));
FP = sum (encrypted_message ~= plaintext & isletter(encrypted_message));
FN = sum (encrypted_message == plaintext & isletter(encrypted_message));

The equations for these metrics are as follows:
• Accuracy:
Accuracy = TP + TN / TP + TN + FP + FN (3)
• Precision:
Precision = TP / TP + TN (4)
• Recall:
Recall = TP / TP+FN (5)
• F1-Score:
F_score = 2 * precision * recall / precision + recall (6)

However, various plaintext sizes are used to compare the results and calculate the average measurement metrics to ensure the algorithm's robustness. Table.1 below compares the algorithm results.

Table 1. **Algorithm Results**

Plaintext Size (KB)	**Encryption Time (ms)**	**TP**	**TN**	**FP**	**FN**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
0.5	0.036	6	8	1	1	0.88	0.85	0.85	0.86
1	0.073	14	18	2	3	0.86	0.87	0.82	0.84
1.5	0.109	21	27	3	5	0.85	0.86	0.80	0.84
2	0.146	28	36	4	6	0.86	0.87	0.82	0.85
5	0.365	70	91	14	16	0.84	0.83	0.81	0.82
10	0.732	141	182	29	31	0.84	0.82	0.82	0.82

100	0.838	250	223	32	45	0.86	0.89	0.84	0.87
500	0.989	1366	1142	187	305	0.83	0.87	0.82	0.84
1000	1.145	2754	2359	380	624	0.84	0.87	0.81	0.85
5000	1.302	14298	12136	1929	4037	0.81	0.88	0.77	0.82
10000	1.408	28438	24124	3767	7813	0.82	0.88	0.78	0.83
50000	1.640	144900	123163	19137	39600	0.82	0.88	0.79	0.83
100000	1.816	289998	247221	38013	79268	0.82	0.88	0.79	0.83
200000	2.108	580996	496440	72086	160546	0.83	0.89	0.78	0.83
Average	0.908					0.84	0.87	0.81	0.84

Table 1 above measures the results of different experiments, each with different plaintext sizes. It includes Encryption Time (ms), Accuracy, Precision, Recall, and F1-Score. The results average reached 84% of accuracy, 87 % of precision, 81% of recall, and 84% of F1-score. The highest accuracy is 88% and highest precision is 89% which indicates high performance algorithm.

However, the optimization figure below is the fitness value graph which is visual representation of the optimization process. It shows how the algorithm is progressing towards better solutions and can help identify any anomalies or patterns, it is clear from the figure that the value remains constant after around 179 iterations. In each trial, the genetic algorithm optimizes the encryption key to achieve the lowest fitness, which is a measure of how well the ciphertext aligns with statistically expected patterns in natural language. The iterations will terminate after reaching the maximum number of iterations specified in the algorithm. See Fig.5.

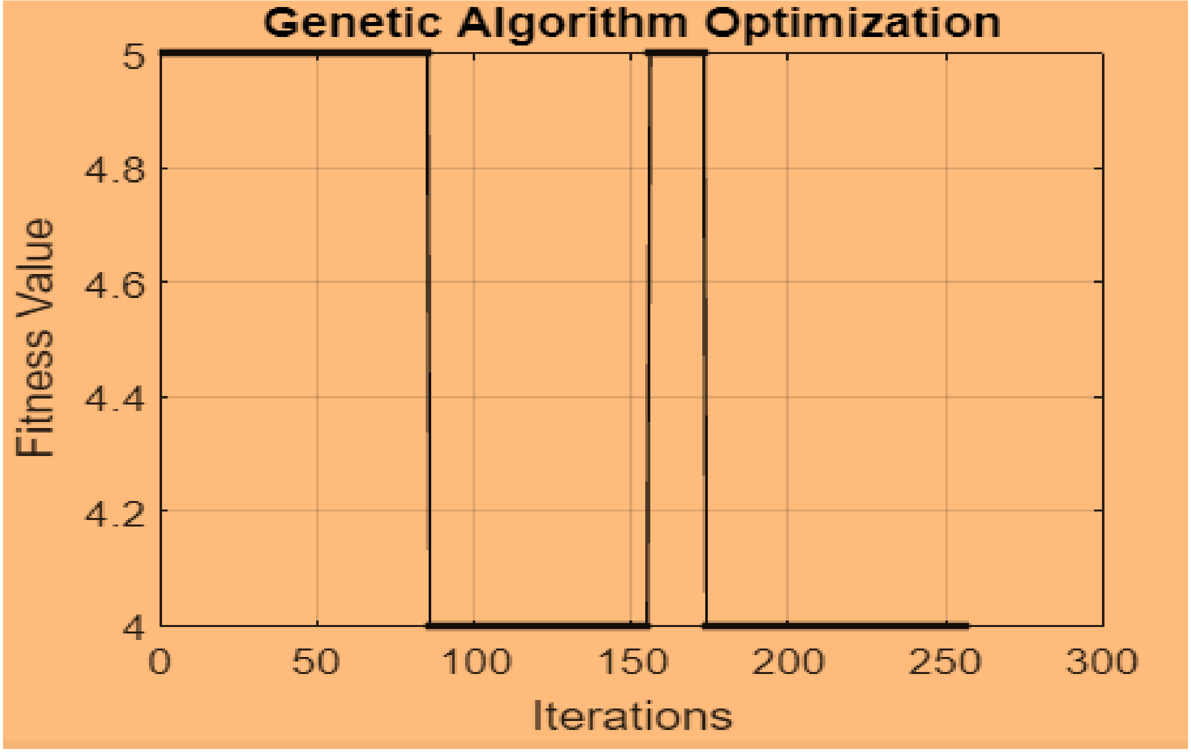

Figure 5. Genetic Algorithm Optimization

5. Comparison and Validation with Similar Algorithms

This study shows a comparison and validation of the proposed algorithm and similar existing algorithms.

5.1 Encryption Time

The duration of time it takes for each algorithm to create cipher text after creating the key is known as the encryption time. The encryption times (measured in milliseconds (ms)) of the proposed algorithm, and the generic algorithm with (NW) [40] algorithm is compared in Table 2.

Table 2. Comparison of Encryption Time

Size in KB	**Encryption Time (ms)**	
	GA with NW Algorithm	Proposed Algorithm
0.5	0.00016	0.036
1	0.00035	0.073
1.5	0.00067	0.109

The results indicate that the proposed algorithm requires more processing time than Genetic Algorithm with NW algorithm because in the proposed algorithm, hill cipher matrix encryption takes longer than XOR operation in Genetic Algorithm with NW algorithm.

5.2 Accuracy

The accuracy of the encryption algorithm is compared with some of the other algorithms that are used nowadays [41]. The percentages are shown in Table 3 below.

Table 3. Results of Accuracy

Algorithm	**Accuracy of Encryption**
AES	74 %
DES	76 %
Proposed Algorithm	88 %

In a ciphertext-only attack, the attacker has some knowledge of the plaintext and some access to some ciphertext. If the attacker can obtain the plaintext or the key, the attack will be successful. This is not conceivable in the proposed algorithm due to the randomness of the key produced by the genetic algorithm. In this scenario, the plaintext also goes through several phases before becoming ciphertext. The known-plaintext attack is a type of attack in which the attacker tries to obtain the key while having access to both the plaintext and the ciphertext. Because the key created by the genetic algorithm is random, the attacker cannot succeed in the proposed algorithm. Each time, a different key is provided. A set of selected plaintexts is used by the attacker to generate ciphertext in a differential cryptanalysis attack. This is unachievable in the proposed algorithm due to the randomness of the key produced by the genetic algorithm. Each time, a separate key is provided. The comparison of the proposed algorithm in response to the numerous attacks with other algorithms is described in Table 4.

Table 4. Comparison with Other Algorithms in Response to Attacks

Attack	**Model**		
	Echo State Network [42]	DES-COMB [43]	Proposed Algorithm
Ciphertext-Only	Can Prevent	-	Can prevent because of the key randomness
The Known-Plaintext	Can Prevent	-	Can prevent because of the key randomness
Differential cryptanalysis	-	Cannot Prevent	Can prevent because of the key randomness

5.3 Confidentiality

Data must be kept private and secure from unauthorized parties. This is accomplished by the proposed algorithm hashing of all data at rest. The results in Table.1 indicate effective algorithm performance using the robustness metrics specified in section *A*.

5.4 Integrity

Integrity is the ability to guarantee that data is correct and constant. This is accomplished in the proposed algorithm because with the using of the singular matrix implemented in the algorithm design, it is hard to get the matrix inverse and decipher the plaintext.

5.5 Availability

It is crucial to make sure that the authorized viewer always has easy access to the relevant information. This is possible with the proposed algorithm since it supports various plaintext sizes as shown in results of Table.1.

6. Conclusion and Future Works

In conclusion, hash functions remain a foundational cryptographic mechanism, widely utilized to address authentication, confidentiality, and integrity challenges. However, to be effective, these functions must be sufficiently robust to withstand evolving security threats. Prior studies have demonstrated that widely used hash functions, such as MD5 and SHA-1, are vulnerable to various attacks, including birthday attacks, man-in-the-middle attacks, and rebound attacks. To address these limitations, this research introduced a novel cryptographic hash function that integrates a Hidden Markov Model (HMM) and a Genetic Algorithm (GA). The proposed approach leverages random key generation and employs HMM as a fitness function to guide the optimization process. Based on standard robustness metrics, the algorithm achieved superior performance, with an accuracy of 88%, precision of 87%, recall of 81%, and an F1-score of 84%. These results highlight the effectiveness and resilience of the developed algorithm compared to existing methods. The study also emphasizes the critical need for secure data hashing, particularly in an era where data may exist in various forms, including images and audio—both of which require protection from unauthorized access and cryptanalysis. As a direction for future work, the proposed hash function can be extended to support additional data formats. Furthermore, since both Neural Networks and Genetic Algorithms offer powerful solutions for complex, non-linear problems, combining these two techniques could lead to even more robust and adaptive security models.

Corresponding author
Dr. Mohammed Almaiah
m.almaiah@kfu.edu.sa

Acknowledgements
Not applicable.

Funding
No funding.

Contributions
A.A; M.A; M.W.A; S.Y.A; Conceptualization, M.A; Investigation, M.W.A; S.Y.A Writing (Original Draft), A.A; M.A; M.W.A; S.Y.A;; and A.A; M.A; M.W.A; S.Y.A; Writing (Review and Editing) Supervision, A.A; M.A; M.W.A; S.Y.A; Project Administration.

Ethics declarations
This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for publication
Not applicable.

Competing interests
The author declares no competing interests.

References

[1] Naik, R. Lakshman, Sourabh Jain, and Manjula Bairam. "Development of hybrid weighted networks of RNN and DBN for facilitating the secure information system in cyber security using meta-heuristic improvement." *Wireless Networks* (2025): 1-36.

[2] Ganesh, N. S., et al. "Deep learning-based user authentication with hybrid encryption for secured blockchain-aided data storage and optimal task offloading in mobile edge computing." *Wireless Networks* 31.3 (2025): 2389-2417.

[3] Ali, A., Almaiah, M. A., Hajjej, F., Pasha, M. F., Fang, O. H., Khan, R., ... & Zakarya, M. (2022). An industrial IoT-based blockchain-enabled secure searchable encryption approach for healthcare systems using neural network. *Sensors*, *22*(2), 572.

[4] Almaiah, M. A., Ali, A., Hajjej, F., Pasha, M. F., & Alohali, M. A. (2022). A lightweight hybrid deep learning privacy preserving model for FC-based industrial internet of medical things. *Sensors*, *22*(6), 2112.

[5] Almaiah, M. A., Yelisetti, S., Arya, L., Babu Christopher, N. K., Kaliappan, K., Vellaisamy, P., ... & Alkdour, T. (2023). A novel approach for improving the security of IoT–medical data systems using an enhanced dynamic Bayesian network. *Electronics*, *12*(20), 4316.

[6] Pittalia, P. P. (2019). A comparative study of hash algorithms in cryptography. *International Journal of Computer Science and Mobile Computing*, *8*(6), 147-152.

[7] Pujari, S.K., Bhattacharjee, G., Bhoi, S., A Hybridized Model for Image Encryption through Genetic Algorithm and DNA Sequence, Procedia Computer Science, 2018.

[8] Chikkareddi, V., Ghosh, A., Jagtap, P., Joshi, S., & Kanzaria, J. (2020). Hybrid image encryption technique using Genetic algorithm and Lorenz Chaotic system. In *ITM Web of Conferences* (Vol. 32, p. 03009). EDP Sciences.

[9] Jiang, C., Pang, Y., & Wu, A. (2015, November). A novel robust image-hashing method for content authentication. In *2015 International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec)* (pp. 22-27). IEEE.

[10] Almaiah, M. A., Hajjej, F., Ali, A., Pasha, M. F., & Almomani, O. (2022). A novel hybrid trustworthy decentralized authentication and data preservation model for digital healthcare IoT based CPS. *Sensors*, *22*(4), 1448.

[11] AlSalem, T. S., Almaiah, M. A., & Lutfi, A. (2023). Cybersecurity risk analysis in the IoT: A systematic review. *Electronics*, *12*(18), 3958.

[12] Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2024). Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms. *Sensors*, *24*(2), 713.

[13] Wong, K. W., Yap, W. S., Wong, D. C. K., Phan, R. C. W., & Goi, B. M. (2020). Cryptanalysis of genetic algorithm-based encryption scheme. *Multimedia Tools and Applications*, *79*(35), 25259-25276.

[14] Rodríguez, J., Corredor, B., & Suarez, C. (2019). Genetic Operators Applied to Symmetric Cryptography. *International Journal of Interactive Multimedia & Artificial Intelligence*, *5*(7).

[15] Sheeja, S. (2017). Secure symmetric encryption scheme using genetic algorithm. *International Journal of Applied Engineering Research*, *12*(21), 10828-10833.

[16] Sohal, M., & Sharma, S. (2022). BDNA-A DNA inspired symmetric key cryptographic technique to secure cloud computing. *Journal of King Saud University-Computer and Information Sciences*, *34*(1), 1417-1425.

[17] Nazeer, M. I., Mallah, G. A., Shaikh, N. A., Bhatra, R., Memon, R. A., & Mangrio, M. I. (2018). Implication of genetic algorithm in cryptography to enhance security. *International Journal of Advanced Computer Science and Applications*, *9*(6), 375-379.

[18] Chunka, C., Goswami, R. S., & Banerjee, S. (2019). A novel approach to generate symmetric key in cryptography using genetic algorithm (ga). In *Emerging Technologies in Data Mining and Information Security* (pp. 713-724). Springer, Singapore.

[19] Jawaid, S., & Jamal, A. (2014). Generating the best fit key in cryptography using genetic algorithm. *International Journal of Computer Applications*, *98*(20), 33-39.

[20] Siam, A. I., Almaiah, M. A., Al-Zahrani, A., Elazm, A. A., El Banby, G. M., El-Shafai, W., ... & El-Bahnasawy, N. A. (2021). Secure health monitoring communication systems based on IoT and cloud computing for medical emergency applications. *Computational Intelligence and Neuroscience*, *2021*(1), 8016525.

[21] AlSalem, T. S., Almaiah, M. A., & Lutfi, A. (2023). Cybersecurity risk analysis in the IoT: A systematic review. *Electronics*, *12*(18), 3958.

[22] Almaiah, M. A., Al-Zahrani, A., Almomani, O., & Alhwaitat, A. K. (2021). Classification of cyber security threats on mobile devices and applications. In *Artificial Intelligence and Blockchain for Future Cybersecurity Applications* (pp. 107-123). Cham: Springer International Publishing.

[23] Afarin, R., & Mozaffari, S. (2013, August). Image encryption using genetic algorithm and binary patterns. In *2013 10th International ISC Conference on Information Security and Cryptology (ISCISC)* (pp. 1-5). IEEE.

[24] Almomani, O., Almaiah, M. A., Alsaaidah, A., Smadi, S., Mohammad, A. H., & Althunibat, A. (2021, July). Machine learning classifiers for network intrusion detection system: comparative study. In *2021 International Conference on Information Technology (ICIT)* (pp. 440-445). IEEE.

[25] Soni, A., & Agrawal, S. (2012). Using genetic algorithm for symmetric key generation in image encryption. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *1*(10), 137-140.

[26] Almaiah, M. A., Dawahdeh, Z., Almomani, O., Alsaaidah, A., Al-Khasawneh, A., & Khawatreh, S. (2020). A new hybrid text encryption approach over mobile ad hoc network. *Int. J. Electr. Comput. Eng.(IJECE)*, *10*(6), 6461-6471.

[27] Alamer, M., & Almaiah, M. A. (2021, July). Cybersecurity in Smart City: A systematic mapping study. In *2021 international conference on information technology (ICIT)* (pp. 719-724). IEEE.

[28] F. Ahamad, Mohd. Shahid ,S. Khalid (2012), ”Encrypting The Information Using The Features of Mimetic Algorithm With Encryption and Decryption Technique” , *(IJERA) International Journal of the Engineering Research and the Applications*, Vol. 2nd, Issue 3rd, May-Jun 2012, page.3049-3051.

[29] N. Kumar, R.Bedi, Rajneesh Kaur (2011), “A Special Technique for the Information Cryptography by Using Genetic Algorithms and Brain Mu Waves”, *International Journal of Scientific and Engineering Research* Volume 2nd, Issue 5th, June 2011 ISSN 2229-5518.

[30] A. Aggarwal (2012), “The Secret Key Data Encryption Algorithm Using Genetic Algorithm” at *IJARCSSE*, 2012.

[31] Singh, D., Rani, P., & Kumar, R. (2013). To design a genetic algorithm for cryptography to enhance the security. *Int J Innov Eng Technol (IJIET)*, *2*(2), 380-385.

[32] Bhandari, S. U., Subbaraman, S., Pujari, S., & Mahajan, R. (2009, May). Real time video processing on FPGA using on the fly partial reconfiguration. In *2009 International Conference on Signal Processing Systems* (pp. 244-247). IEEE.

[33] Bartkewitz, T. (2009). Building hash functions from block ciphers, their security and implementation properties. *Ruhr-University Bochum*.

[34] Jhingran, R., Thada, V., & Dhaka, S. (2015). A study on cryptography using genetic algorithm. *International Journal of Computer Applications*, *118*(20).

[35] Kumar, A., & Chatterjee, K. (2016, March). An efficient stream cipher using genetic algorithm. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 2322-2326). IEEE.

[36] Mishra, S., & Bali, S. (2013). Public key cryptography using genetic algorithm. *International Journal of Recent Technology and Engineering*, *2*(2), 150-154.

[37] Adil, M., Almaiah, M. A., Omar Alsayed, A., & Almomani, O. (2020). An anonymous channel categorization scheme of edge nodes to detect jamming attacks in wireless sensor networks. *Sensors*, *20*(8), 2311.

[38] Bubukayr, M. A. S., & Almaiah, M. A. (2021, July). Cybersecurity concerns in smart-phones and applications: A survey. In *2021 international conference on information technology (ICIT)* (pp. 725-731). IEEE.

[39] CHOKPRASOMBAT, K. (2011). Synthesis of patterned media by self-assembly of FePt nanoparticles. *Walailak Journal of Science and Technology (WJST)*, *8*(2), 87-96.

[40] Kalsi, S., Kaur, H., & Chang, V. (2018). DNA cryptography and deep learning using genetic algorithm with NW algorithm for key generation. *Journal of medical systems*, *42*(1), 1-12.

[41] Venkatesan, R., Anni Princy, B., Ambeth Kumar, V. D., Raghuraman, M., Gupta, M. K., Kumar, A., ... & Khan, A. K. (2021). Secure online payment through facial recognition and proxy detection with the help of TripleDES encryption. *Journal of Discrete Mathematical Sciences and Cryptography*, *24*(8), 2195-2205.

[42] Ramamurthy, R., Bauckhage, C., Buza, K., & Wrobel, S. (2017, September). Using echo state networks for cryptography. In *International Conference on Artificial Neural Networks* (pp. 663-671). Springer, Cham.

[43] Hussain, U. N., Chithralekha, T., Raj, A. N., Sathish, G., & Dharani, A. (2012). A hybrid DNA algorithm for DES using central dogma of molecular biology (CDMB). *International Journal of Computer Applications*, *42*(20), 1-4..

Biographies

Aseel AlShuaibi received a master degree in Cybersecurity from King Faisal University. He has an excellent experience in the cybersecurity field in both theoretical and practical. He has several certificates in cybersecurity like CEH and others. He several publications in cyber risk assessment. His research interests including cyber security, risk assessment and cyber-attacks. 221445338.student@kfu.edu.sa

Dr. Muhammad Waqas Arshad received Ph.D. from University of Bologna, Bologna, Italy. He is a visiting Scholar | Purdue University, Fort Wayne, USA. He is a researcher at Ferrari, Italy. He is working as a general secretary of ISOC-BSIG. He is a S-Member at IEEE. muhammadwaqas.arsha2@unibo.it

Dr. Mohammed Maayah Maayah is among the top 2% scientists in the world from 2020 up to now. He is working as Editor in Chief for the International Journal of Cybersecurity and Risk Assessment. He has published over 115 research papers in highly reputed journals such as the Engineering and Science Technology, an International Journal, Education and Information Technologies, IEEE Access and others. Most of his publications were indexed under the ISI Web of Science and Scopus. His current research interests include Cybersecurity, Cybersecurity-Risk Assessment and Blockchain. m.almaiah@ju.edu.jo