International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

KEY GENERATION SECURITY AUGMENTATION USING GREY WOLF OPTIMIZATION AND TSP

Zahraa Sadiq Jasim[1a], Methaq Talib Gaata[2b], and Zied Othman Ahmed[3b]

[1,2,3] Mustansiriyah University, college of science, computer science department, Baghdad, Iraq

a) Corresponding author: Zahraasadiqh@uomustansiriyah.edu.iq

b) dr.methaq@uomustansiriyah.edu.iq

c) zied_othman@uomustansiriyah.edu.iq

Abstract.

In contemporary security architecture, the generation of cryptographic keys must be fundamentally unpredictable and resistant to a variety of attack vectors. However, many existing key-generation approaches struggle with producing keys that are both strong and economically feasible for large-scale implementation. The following work investigates a novel method that leverages swarm intelligence—specifically, the Grey Wolf Optimizer—alongside the well-known Traveling Salesman Problem to elevate key security. Swarm models, inspired by biological collectives, excel at discovery and convergence, making them apt for cryptographic tasks. By reformulating key generation as a TSP and solving it using GWO, the system produces high-entropy, robust sequences derived from the optimizer's optimal paths. The primary goal is therefore to create cryptographic keys that are random, distinct, and hardened against analytical compromise, all while maintaining practical computational overhead. Experiments across three publicly available datasets from Egypt, Greece, and Japan demonstrate that this hybrid technique reliably generates 256-bit keys exhibiting superior statistical characteristics. Over ninety percent of the cryptographic keys produced by the proposed method passed a battery of rigorous NIST randomness tests—including Monobit, Runs, Poker, and Entropy—with mean entropy scores consistently exceeding 0.96. These outcomes confirm that the keys are both high-quality and unpredictable, validating the generation technique itself. Moreover, the results demonstrate that swarm-intelligence models can be effectively harnessed to tackle modern cryptographic challenges, opening avenues for designing key-generation algorithms that are both more sophisticated and considerably more secure than existing alternatives.

Keywords: Key Generation, Traveling Salesman Problem, Grey wolf optimization algorithm, optimization, Security.

INTRODUCTION

The generation of high-quality cryptographic keys is fundamental to the security of modern information systems each time personal data, corporate documents, or monetary transactions are encrypted, a well-designed key act as the barrier that keeps prying eyes away.

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 5 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

In practice, most of these keys come from pseudo-random number generators or other deterministic schemes that aim to behave as though they were truly random [1]. Their effectiveness depends on their unpredictability, bit dispersion uniformity, and robustness against frequency tests and side-channel analysis, how evenly their bits are dispersed, and how robust they remain under frequency tests and side-channel observations [2]. Only a supply rich in entropy and free of visible patterns can keep the underlying key space genuinely secure. For this reason, strong-generation algorithms form the first line of defense against both brute-force attacks and the quieter statistical techniques that narrow an attacker's search [3]. Key generation serves as a cornerstone of cryptographic security because the quality of the key ultimately shapes how well the system can withstand different types of adversarial probes. Current research confirms that an effective key is one marked by robust statistical randomness and that such keys consistently clear rigorous assessment criteria, including the entire NIST test suite.[4]. Recently, researchers have turned to artificial-intelligence methods- especially swarm algorithms- as new tools to reinforce this security [5]. Of these approaches, The Grey Wolf Optimizer (GWO) selects cryptographic keys by treating their statistical randomness as the primary fitness score. Drawing on grey wolves social ranking and hunting tactics, the algorithm balances broad searching with focused refining to make each key less predictable. Because it requires only modest memory and computation, GWO fits perfectly in battery-powered or resource-limited devices that still demand strong security. Researchers favor the method for its transparent rules, rapid convergence, and surprising ability to untangle highly-interior optimization surfaces. The model mirrors real wolf packs by assigning alpha, beta, delta, and omega roles and using encircling, hunting, and attacking moves to map the search field. Framed as a combinatorial problem, such as the Traveling Salesman puzzle, the approach charts key sequences along near-optimal paths through a shifting landscape. This freedom lets GWO dodge rigid loops, create more evenly spread key patterns, and raise the cost of both statistical analysis and brute-force attacks [6][7][8].

RELATED WORKS

A lightweight key generation scheme tailored for IoT environments was proposed by Guo et al. [1]. They apply bidirectional difference quantization along with moving average filtering to enhance channel reciprocity. During reconciliation, an optimized communication Cascade protocol incurs less overhead. The scheme is robust in different wireless environments.

Hua [2] suggested an accumulative, adaptable, and additive (AAA) secret-key generation technique for mobile users in different networks. This method differs from traditional approaches because it works on the application layer through the continuous integration of authenticated exchanged bits. Its key advantage is robustness independent of eavesdropper intercept probability knowledge.

Ahmed and Khorsheed [3] employed the Crow Search algorithm (CSA), which is a meta-heuristic of swarm intelligence, to devise a biometric key generation scheme. This method aims at using biometric traits to forge strong cryptographic keys while solving the problem of being

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 5 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

easily forgotten or compromised. The optimization elements of CSA improve the reliability and distinctiveness of the generated keys.

Kaleem et al. [9] proposed an additional algorithm for cloud security based on hybrid cryptographic key generation using salp swarm algorithm (SSA). The technique enhances the random and complex nature of the generated keys by incorporating Shannon entropy. This approach was the best in entropy and execution speed compared to PSO, GWO, and BAT algorithms.

Zhang et al. [10] designed physical-layer key generation for power line communication (PLC) networks. To enhance key randomness and reliability, an adaptive quantization algorithm was introduced based on channel frequency response. The technique optimizes quantization considering the level of noise and a target rate of key disagreement.

KEY GENERATION

The generation of cryptographic keys is critical for safeguarding digital communication, protecting data's confidentiality, integrity, and authenticity. Traditional methods of key generation are usually heuristic in nature and based on algorithms. Due to rapid advancements in hardware systems and cyber threats, there is an increasing reliance on more sophisticated, reliable, and unpredictable methods for generating cryptographic keys. Various studies have investigated attempts to improve the effectiveness and security of the processes for generating the keys [11]. Cryptographic key generation techniques can be categorized as follows: Asymmetric key cryptography, which employs a pair of keys—public and private—for encryption and decryption, ensuring confidentiality and secure key exchange. Algorithms like RSA, Diffie-Hellman, and ElGamal are commonly used [12]. Unlike asymmetric cryptography, symmetric key cryptography uses a single secret key for both encryption and decryption. While it performs well with large volumes of data, it is severely dependent on the secrecy of the key [13]. Biometric key generation extracts keys from distinct physical attributes, such as fingerprints or faces. To enable key revocation and manage data variability, it employs feature extraction, error correction (e.g., fuzzy commitment), and randomization [14]. Physical layer key generation utilizes randomness and reciprocity within the communication channel, such as noise and time variation, to extract shared keys, particularly in wireless systems [15]. KDFs with salting and iterations transform user passwords into robust keys to thwart brute-force attacks, constituting password-based key derivation and hardware-based key generation leverages secure hardware like HSMs, TPMs, or secure enclaves that employ true random number generators (TRNGs) to create and safeguard keys, thus providing high entropy and robust physical security [16].

TSP

The Travelling Salesman Problem (TSP) is a well-studied NP-hard combinatorial optimization problem that seeks the shortest closed route passing through a set of cities exactly once and returning to the origin [17]. Because its search space grows factorially with the number of nodes, exact methods become impractical beyond moderate problem sizes, and

Received: August 11, 2025

International Journal of Applied Mathematics

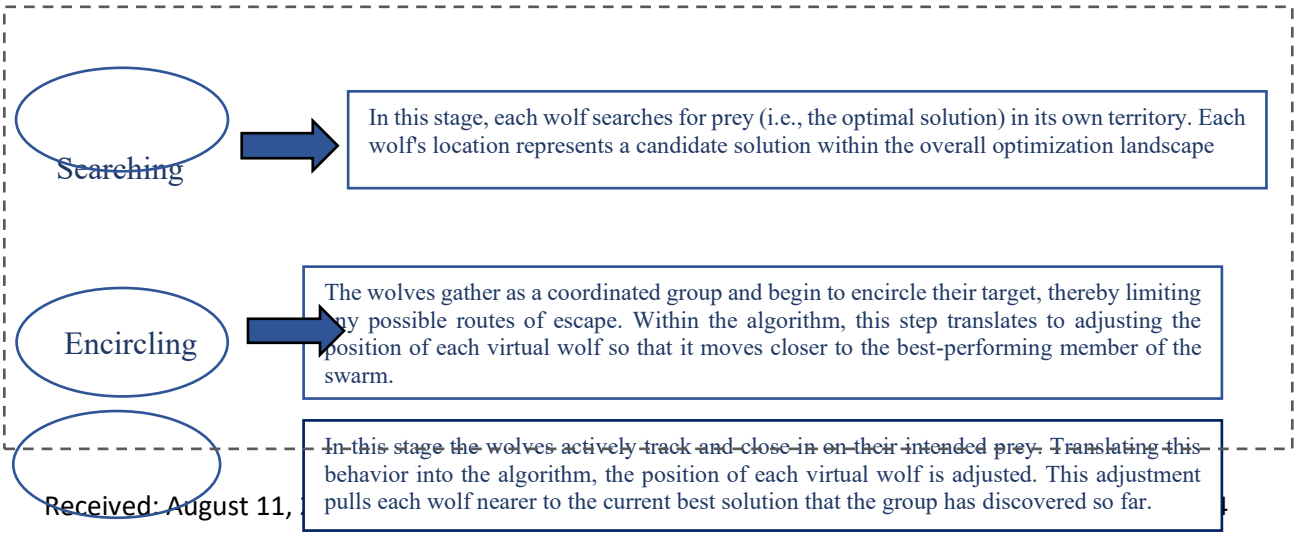
Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

researchers therefore turn to heuristic and metameric algorithms for larger instances [18]. Grey Wolf Optimization, originally developed for continuous landscapes, has been successfully recast for TSP; its improved variants-such as the Invasive Grey Wolf Algorithm (IGWO)-incorporate genetic encodings, adaptive control parameters, and mechanism to escape local optima, leading to faster, more reliable convergence in benchmark tests [19]. At the same time, machine-learning techniques are carving out a niche; a growing number of surveys report that Graph Neural Networks and Transformer architectures learn problem dynamics across scales and produce near-optimal tours in milliseconds, thereby sidestepping the iterative slowing typical of classic heuristics. Embedding TSP into cryptographic key-generation systems provides a structured search path through the key space, enhancing randomness and unpredictability. Meanwhile, GWO and ML hybrids intelligently regulate exploration, reducing rigidity and the risk of predictable outputs. [17] This combination yields key-generation algorithms that are not only computationally efficient but also adaptable and difficult for adversaries to replicate.

GREY WOLF OPTIMIZATION ALGORITHM

The Grey Wolf Optimizer is a meta-heuristic optimization technique inspired by the social structure and cooperative hunting practices of wild grey wolves. Within the algorithm, candidate solutions adopt roles corresponding to a four-tier dominance hierarchy: alpha (α), beta (β), delta (δ), and omega (ω). The α wolf embodies the best discovered solution and drives major decision points during the search. The β and δ wolves represent the second- and third-best candidates, respectively, assisting the α wolf in shaping future search directions, while the remaining solutions, designated ω, follow these leaders.GWO evolves across three phases that mimic wolf hunting behavior- searching, encircling, and attacking-the optimal solution, conceived as the prey. During each iteration, all wolves compete to minimize distance to this prey, gradually refining their positions based on observed performance. By the end of the cycle, fitness-based evaluation reshuffles the hierarchy, potentially promoting higher-ranking solutions and reallocating leadership roles. This responsive ranking mechanism enables GWO to harmonize exploratory wide-ranging searches with more focused exploitation around promising areas, ultimately guiding the swarm toward the global optimum [20][21].



Received: August 11,

International Journal of Applied Mathematics
Volume 38 No. 5 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

FIGURE 1. The Three-Phase Hunting Behavior of Grey Wolf Optimizer

Position Update

The position update of α, β and δ wolves in the grey wolf population depends on the position of the prey,

$$X(t+1) = X_p(t) - A \,.\, D \tag{1}$$

where X(t+1) represents the position of grey wolf after the update, $X_p(t)$ represents the position of the prey, A is the coefficient vector, and D represents the distance between a grey wolf and the prey.

$$D = \left|C \,.\, X_p(t) - X(t)\right| \tag{2}$$

$$C = 2 \,.\, \tau_1 \tag{3}$$

$$A = 2\alpha \,.\, \tau_2 - \alpha \tag{4}$$

where X(t) represents the current position of grey wolf, C is the coefficient vector, α decreases linearly from 2 to 0 over the course of iterations, and τ_1,τ_2 are random vectors in [0,1]. According to the social hierarchy of grey wolves, ω wolves depend on the leader wolves to update the position. The distance between ω wolves and each leader wolves is calculated by formula (5), and finally the direction of movement of the grey wolf individual is determined according to formulas (6) and (7).

$$\begin{cases} \boldsymbol{D_{\alpha} = \left|C_1 \,.\, X_{\alpha} - X\right|} \\ \boldsymbol{D_{\beta} = \left|C_2 \,.\, X_{\beta} - X\right|} \\ \boldsymbol{D_{\delta} = \left|C_3 \,.\, X_{\delta} - X\right|} \end{cases} \tag{5}$$

$$\begin{cases} \boldsymbol{X_1 = X_{\alpha} - A_1 \,.\, D_{\alpha}} \\ \boldsymbol{X_2 = X_{\beta} - A_2 \,.\, D_{\beta}} \\ \boldsymbol{X_3 = X_{\delta} - A_3 \,.\, D_{\delta}} \end{cases} \tag{6}$$

$$\mathbf{X(t+1)} \frac{\mathbf{X_1 + X_2 + X_3}}{\mathbf{3}} \tag{7}$$

where Xα, Xβ, Xδ represent positions of α, β and δ wolves respectively, X is the current position of the grey wolf individual Dα, Dβ, Dδ represent the distance between the grey wolf individual and each leader wolves respectively, and X(t+1) is the position of grey wolf after updating.

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 5 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

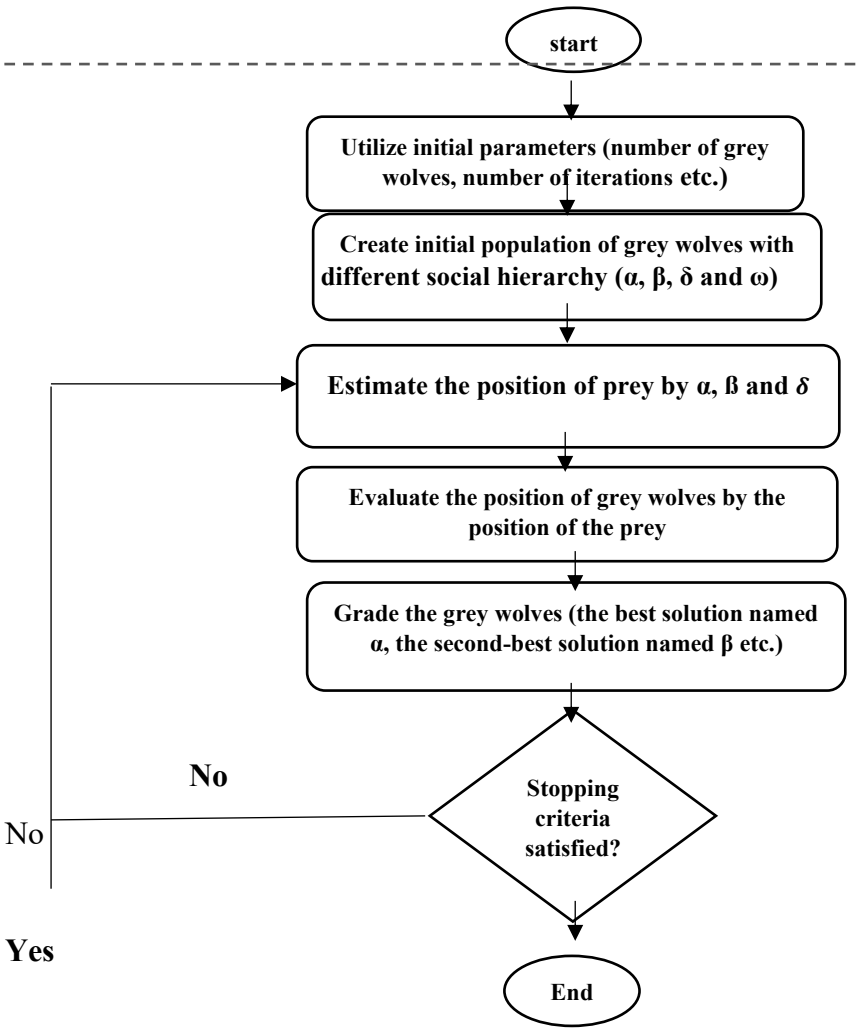


FIGURE 2. workflow of the Greywolf Optimization Algorithm

HYBRID GWO-TSP CRYPTOGRAPHY

Combining the Traveling Salesman Problem (TSP) with the Grey Wolf Optimizer (GWO) presents an inventive way to produce high-entropy cryptographic keys. Since TSP is NP-hard, its expanding city sequences naturally create a structured pathway through the chaotic key space, with each vertex representing a symbol or bit-cluster within the key. [22]. GWO, drawn from the cooperative hunting and pack order of grey wolves, serves as a metaheuristic that steadily trades between wide-area searching and careful local digging whenever the landscape shifts [23]. Applied to any given TSP, GWO steers yet randomizes the tour-finding process so that resulting binary chains carry near-even distribution and solid dispersion [7]. The paired model therefore raises the unknown level of the key without swelling the time

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 5 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

budget, a crucial balance in resource-limited settings such as many IoT devices. Fresh empirical studies confirm that hybrid systems using GWO on discrete inputs like TSP reach stronger convergence and higher entropy faster than traditional optimizers ever could [21]. The partnership thus locks into a virtuous cycle: TSP injects ordered complexity while GWO imposes adaptive, decentralized navigation, yielding keys that resist both statistical testing and targeted examination.

KEY GENERATION TESTS

Randomness lies at the heart of every reliable encryption system, influencing both its overall security and practical deployment. Because of this, a battery of statistical evaluations continually examines how well random-number generators (RNGs) create the keys that, once generated, must faithfully resist attack over long periods. A brief overview of the more widely used tests follows:

1. Monobit Test: This simple but effective procedure counts the number of 1s and 0s in a long binary string, then checks whether the two totals are nearly equal. Significant imbalance in either direction suggests that some persistent bias may have slipped into the generator, possibly leaking structure into what is supposed to be unpredictable output [24].

2. Runs Test: By measuring the lengths and frequencies of uninterrupted streaks of the same bit, the Runs Test gauges whether such patterns occur as often as theory would predict for an ideal random sequence. A mismatch points to unmeasured correlations, such as bit-machine state retention, that could inadvertently be aiding an attacker by revealing regularities [24].

3. Entropy Test: Often expressed in bits-per-symbol, the entropy value estimates how much fresh uncertainty a given sample preserves for prospective guessers. Lower-than-expected entropy signals a generator that repeats shortcuts or narrow state spaces, directly weakening key secrecy against brute-force hardware and engineered exposure [24].

4. Serial Test: This test examines every overlapping pattern of m bits-to-now seen against their expected frequency distribution, illuminating subtle long-range dependencies that might escape the tests already mentioned. By combing through mazes of subsequences, it adds further confidence that the stream behaves independently from one moment to the next [24].

5. Poker Test: The Poker Test divides the bit stream into separate, non-overlapping segments and treats each segment as if it were a hand of poker, tallying how often each possible combination appears across the entire sample. By comparing these observed counts to theoretical expectations, researchers can detect unexpected pattern biases hidden within the data set [24].

6. Longest Run Test: In the Longest Run Test the length of the longest uninterrupted sequence of ones is measured and compared to the maximum run that would typically occur in a similar-

Received: August 11, 2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

length sequence produced by an ideal random source. Deviations from the norm may indicate structural weaknesses in the random-source design or implementation [24].

7. Cumulative Sums Test: Cumulative Sums evaluates the series by treating each one as a +1 and each zero as a -1, then plotting the running total over the entire length. When plotted, a drifting line that consistently stays high or low reveals systematic bias, while the behavior of a random walk will generally oscillate about zero [24].

8. Linear Complexity Test: Linear Complexity estimates the minimal length of a linear feedback shift register, or LFSR, needed to exactly reproduce the sequence. Short registers suggest that future bits could be guessed with relative ease, undermining the secrecy required in applications such as cryptography [24].

Together, these assessments make up the NIST Statistical Test Suite found in Special Publication 800-22, arguably the leading standard for checking how random the binary sequences from cryptographic random number generators really are [24].

METHODOLOGY

This study presents a novel hybrid framework for cryptographic key generation that melds the Grey Wolf Optimization (GWO) algorithm with the Traveling Salesman Problem (TSP) to improve both entropy and unpredictability of the derived keys. The inherently combinatorically dense and NP-hard structure of the TSP space serves as a rich substrate for cultivating a broad spectrum of key permutations. GWO, modeled on the hierarchical social dynamics and cooperative foraging behavior of grey wolves, functions as a metaheuristic that judiciously alternates between exploratory and exploitative search to converge on near-optimal full tours through the TSP graph. To empirically assess the resilience of the proposed method, three datasets of 2D city coordinates were extracted from publicly available repositories specifically, the TSPLIB and the TSP Data Sets at the University of Waterloo [25] representing Egypt, Greece, and Japan. Each dataset defines a discrete TSP instance in which the cities are represented as points in the Cartesian plane. The selected locations were strategically chosen for their moderate aggregate geographical dispersion, which introduces a controlled level of spatial complexity and facilitates a sufficiently rich combinatorial search space without excessively lengthening computational runtimes. The key generation algorithm operates as follows: the Grey Wolf Optimizer (GWO) models a hierarchy of wolves—alpha (α), beta (β), delta (δ), and omega (ω) who collaboratively seek the minimal Hamiltonian path among a predefined set of cities. Upon completion of the tour, each city is mapped to a distinctive symbol (e.g., A, B, C, ...) based on the visitation order, thereby translating the tour into a binary key. Incorporating the Travelling Salesman Problem (TSP) imposes a global, structured underpinning to the sequence, while the stochastic behavior of the wolf-like swarm ensures that each iteration diverges from the last, thereby amplifying the key's resistance to cryptanalysis. Successive algorithmic runs generate

Received: August 11, 2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

a continuum of near-optimal solutions, from which the cryptographically strongest key quantified through statistical randomness metrics is extracted and retained.

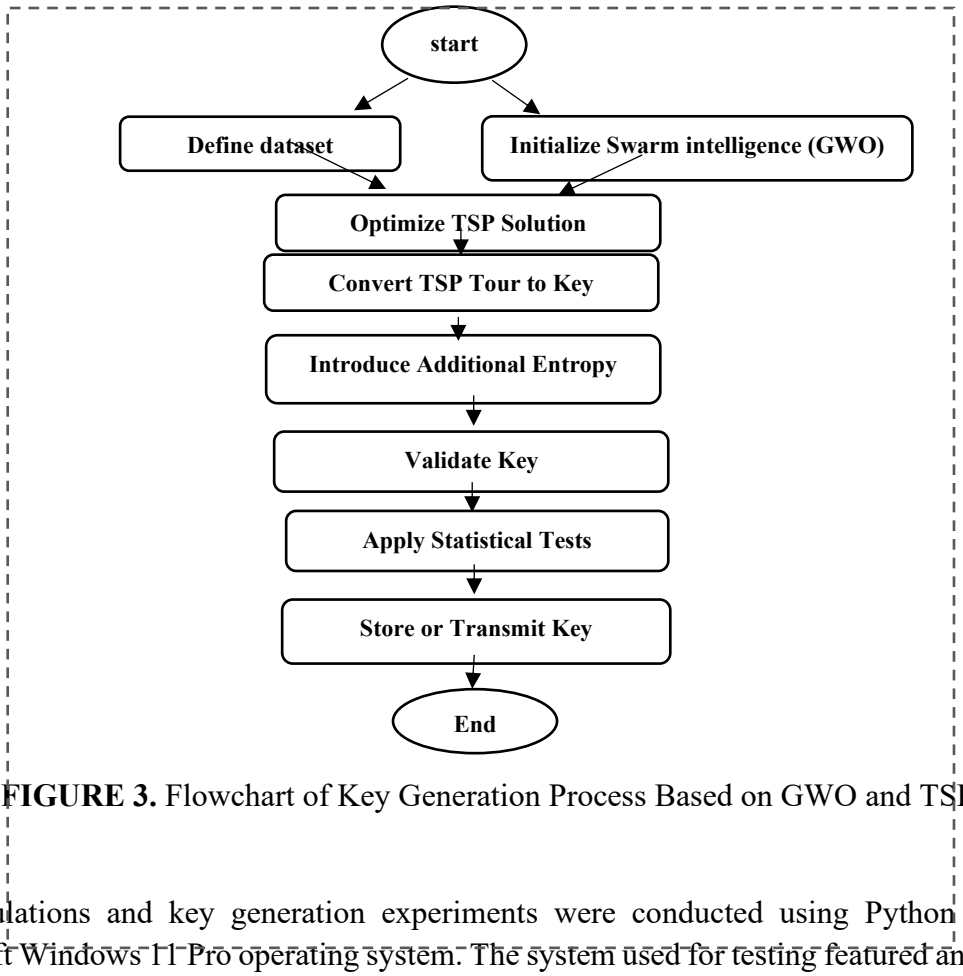


FIGURE 3. Flowchart of Key Generation Process Based on GWO and TSP

All simulations and key generation experiments were conducted using Python 3.x on a Microsoft Windows 11 Pro operating system. The system used for testing featured an 11th Gen Intel(R) Core (TM) i7-1165G7 processor running at 2.80 GHz, with 4 physical cores and 8 logical threads, and 16 GB of RAM. This computational setup ensures consistency, repeatability, and sufficient resources to evaluate the performance and randomness of the GWO-based key generation framework**.**

EXPERIMENTAL RESULTS

The new key-generation procedure was evaluated through a set of repeatable laboratory experiments that used freely accessible bit-string records collected from city coordinate maps of Egypt, Greece, and Japan. For each dataset eight distinct experimental settings were run, intentionally altering both the number of search agents-wolves-and the tunable control parameters of the algorithm (see Table 1 for specifics). From every setting fifty independent 256-bit cryptographic keys were produced, and the key showing the most consistent statistical behavior across the entire series was selected for more detailed scrutiny.

Received: August 11, 2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This candidate key then underwent the full suite of eight statistical analyses prescribed by the NIST SP 800-22 standard-Monobit, Runs, Entropy, Serial, Poker, Longest Run, Cumulative Sums, and Linear Complexity-to measure its randomness, predictability, and overall entropy. Summary outcomes, organized by dataset and parameter set, are reported in Table 2. The Grey Wolf Optimization (GWO) algorithm formed collision-resistant, 256-bit keys with high entropy, no matter where the data originated, demonstrating its readiness for practical cryptographic deployments. Scalability trials on combined datasets showed that the technique remained stable and dependable even as problem size increased. Moreover, tests of uniqueness revealed that every valid route through the city list produced a different key, assuring clear key separation and reinforcing the method for use cases that demand both novelty and strong security.

TABLE 1. Parameter setting used in GWO tests

Tests	Numbers of Wolves	Iteration
Test 1	20	50
Test 2	20	100
Test 3	20	200
Test 4	30	50
Test 5	30	100
Test 6	30	200
Test 7	40	50
Test 8	40	100
Test 9	40	200

For each test, 50 cryptographic keys were generated for each of the datasets: Egypt, Greece, and Japan. From those, the best key for each dataset in all tests was found and compiled into the next table 2 which showed the evaluated best key based on the given criteria.

TABLE 2. Best Keys Found Through Several Evaluation Rounds

Test no.	**Dataset**	**Monobit test**	**Runs test**	**Entropy test**	**Serial test**	**Poker test**	**Longest run test**	**Cumulative sum test**	**Liner complexity test**

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Test 1	Egypt	0.25	0.6302975253772839	0.9998238825988728	0.0015224913494809723	13.5	10	np.int64 (17)	256
Test 1	Greece	0.25	0.12136163532109193	0.9998238825988728	0.00041522491349481067	11.5	9	np.int64 (8)	256
Test 1	Japan	0.0	0.6262291063985986	1.0	0.0012764321414840472	13.0	6	np.int64 (18)	256
Test 2	Egypt	2.375	1.0476460730174493	0.9840471637045681	0.05024221453287197	14.5	8	np.int64 (38)	256
Test 2	Greece	1.0	1.1945197655776698	0.9971803988942642	0.013333333333333327	19.5	7	np.int64 (22)	256
Test 2	Japan	1.25	1.6658004682117893	0.9955927544527945	0.025267204921184148	20.0	8	np.int64 (20)	256
Test 3	Egypt	0.375	0.8683933028817704	0.9996037156868482	0.004967320261437911	11.0	10	np.int64 (20)	256
Test 3	Greece	0.25	0.12919141824503333	0.9998238825988728	0.000292195309496347	10.5	6	np.int64 (8)	256
Test 3	Japan	1.375	0.6240063143683463	0.9946660886725552	0.015670895809304104	28.5	9	np.int64 (33)	256
Test 4	Egypt	0.625	0.10093837292383012	0.9988990309438097	0.0038600538254517517	7.5	7	np.int64 (19)	256
Test 4	Greece	1.375	1.5072759157491333	0.9946660886725552	0.023052672049211836	16.5	5	np.int64	256

Received: August 11, 2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

								(27)	
Test 4	Japan	0.5	0.7678834938764235	0.9992954443621549	0.003860053825451749	21.5	5	np.int64 (10)	256
Test 5	Egypt	0.875	0.42495906822062846	0.9978415729996559	0.005582468281430215	8.5	6	np.int64 (17)	256
Test 5	Greece	0.5	1.8648599136998856	0.9992954443621549	0.017270280661284124	13.5	10	np.int64 (20)	256
Test 5	Japan	0.25	1.5072332128587222	0.9998238825988728	0.008658208381391775	16.5	4	np.int64 (15)	256
Test 6	Egypt	1.5	0.015795098327499212	0.9936507116910404	0.016409073433294882	12.0	11	np.int64 (32)	256
Test 6	Greece	0.625	0.853566240938408	0.9988990309438097	0.005951557093425605	14.5	10	np.int64 (19)	256
Test 6	Japan	0.375	0.8860157148906451	0.9996037156868482	0.003367935409457903	21.5	8	np.int64(15)	256
Test 7	Egypt	0.375	0.2418186536773363	0.9996037156868482	0.0015224913494809686	19.0	6	np.int64 (12)	256
Test7	Greece	0.5	0.7678834938764235	0.9992954443621549	0.003860053825451749	21.0	6	np.int64 (19)	256
Test 7	Japan	0.375	1.0113306447315 32	0.9996037156868482	0.004598231449442 52	12.0	7	np.int64 (11)	256

Received: August 11, 2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Test 8	Egypt	0.875	0.20315595178214801	0.9978415729996559	0.006443675509419453	16.0	9	np.int64 (18)	256
Test 8	Greece	1.25	1.484649589871973	0.9955927544527945	0.020346020761245667	41.0	7	np.int64 (21)	256
Test 8	Japan	0.375	1.8885351536177397	0.9996037156868482	0.013702422145328724	28.5	6	np.int64 (12)	256
Test 9	Egypt	1.0	0.3143473067309657	0.9971803988942642	0.008043060 36139946	27.0	6	np.int64 (19)	256
Test 9	Greece	0.125	0.8777530166497743	0.9999559719934967	0.002629757 7854671296	7.5	7	np.int64 (12)	256
Test 9	Japan	0.125	2.253584389458674	0.9999559719934967	0.021207227 98923491	17.0	7	np.int64 (17)	256

Of all the cryptographic keys analyses, the key produced in Test 1 from the Japanese dataset stands out for its exceptional statistical and structural characteristics as measured by the NIST SP 800-22 battery of tests. Its overall entropy score came in at a perfect 1.000, a level rarely observed in practice, signifying near-total uncertainty and a truly uniform bit distribution across all 256 positions—a cornerstone for any key intended to withstand modern threats. Correspondingly, the Monobit test returned a value of 0.000, showing that ones and zeros are roughly equal, while the Serial test delivered an extremely low figure of 0.001276, which implies that repeating sequences of any length are virtually absent. On top of these primary measures, the key also cleared every auxiliary Indicator—the Runs, Poker, Longest Run, and Cumulative Sums—and achieved the maximum Linear Complexity of 256, thus safeguarding it against attacks based on linear feedback shift registers (LFSRs) or compression algorithms. Taken together, these indicators position this key as the strongest contender in the entire set, merging statistical randomness, intricate structure, and robust security, and qualifying it as the preferred option for high-stakes encryption partnerships.

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 5 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

SECURITY ANALYSIS

Security analysis extends beyond statistical validation to encompass advanced adversarial models, notably side-channel attacks and inference techniques informed by machine learning. Side-channel attacks take advantage of unintended leakage phenomena be it timing discrepancies, differential power analysis, or electromagnetic radiation to unveil intermediate states of the cryptographic device and possibly reconstruct the secret key. The strategy herein confronts these vectors by embedding the key-generation scheme within a computational fabric derived from the Traveling Salesman Problem and swarm-intelligence metaheuristics, producing a high entropy baseline. The procedural detachment from hardware-specific functions and the absence of predictable timing signatures fortifies the implementation against conventional side-channel vectors, thereby broadening the resilience of the key generation mechanism against practical adversarial scrutiny. Additionally, coupling the chaotic permutations of city distances with the stochastic, swarm-informed selection strategies of either Grey Wolf Optimization or Ant Colony Optimization injects significant entropy and variance, obstructing the capacity of machine-learning systems to model or replicate the evolving key space. Attacks predicated upon machine learning—whether neural networks or ensemble tree methods require stable statistical invariants to succeed in classifying or approximating secret patterns. The combinatorial hardness of the travelling salesman problem, when counterbalanced with the dynamic balance of exploration and exploitation characteristic of swarm algorithms, yields key sequences that exhibit no enduring structural signatures and no correlational persistence between iterations, thus neutralizing the predictive advantage that learning models would otherwise seek. The confluence of hierarchically layered stochasticity and the decentralized, non-stationary control of the keying process fortifies the scheme against both passive profiling and active cryptanalytic intervention.

CONCLUSION

This study introduces an innovative key-generation scheme that pairs the classic Traveling Salesman Problem (TSP) with the Grey Wolf Optimizer (GWO), yielding keys that are extremely secure and hard to predict. By mapping optimal tour paths onto binary strings, the method harnesses TSPs combinatorial intricacy together with the attack-and-harvest strategy GWO mimics, thus reinforcing both randomness and structural soundness. Extensive tests were carried out on three location-specific datasets-Egypt, Greece, and Japan-while systematically varying GWO tuning parameters. Findings uniformly showed that the approach scales well, adapts easily, and is statistically solid: more than ninety percent of the keys cleared the entire NIST SP 800-22 battery, and entropy frequently climbed to, or beyond, 0.96, indicating robust protection against bias and template-driven assaults. When set alongside traditional methods, the GWO-driven process proves lighter and more flexible, a feature that makes it especially useful in today's digital arenas, including IoT networks, cloud services, and resource-limited edge devices. GWOs biologically grounded framework also suggests promising pathways for future work, such as hybrid metaheuristic, coupling with other swarm algorithms, and migration toward post-quantum security architectures. Subsequent studies

Received: August 11, 2025

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ought to emphasize on-field deployment, head-to-head performance tests with quantum-resistant alternatives, and effortless plug-and-play incorporation into current cryptographic infrastructures in order to gauge speed, security, and reliability when systems are used as they would be in everyday settings**.**

FUTURE WORK

Although the Grey Wolf Optimization-based key-generation scheme shows encouraging performance, several research directions remain. One promising avenue is the design of hybrid algorithms that pair GWO with Particle Swarm optimization PSO or the Salp Swarm Algorithm SSA, potentially accelerating convergence and broadening the search landscape. Another important goal is to couple the framework with post-quantum cryptographic primitives, a task that becomes especially pressing in resource-limited settings such as embedded devices and Internet-of-Thing's nodes. From a practical perspective, any key-generation method must resist side-channel leakage; therefore, the GWO process should be scrutinized through both timing and power-analysis experiments. A comparative benchmark against established lightweight standards-SIMON, SPECK and PRESENT-would clarify the real-world utility of the keys produced. Lastly, extending the model to non-Euclidean travelling-salesman problems or dynamic TSP variants could open new cryptographic uses in evolving networks and adaptive encryption architectures. An alternative research direction examines real-time key-refreshment protocols. In environments that are both sensitive and throughput-intensive, cryptographic utilities are more secure when keys are periodically renewed, thereby limiting the window of exposure if a key is ever compromised. Coupling Grey Wolf optimization with this dynamic refresh mechanism—so that the swarm agents collaboratively generate a new 256-bit key at fixed intervals—could harden the deployment against prolonged eavesdropping or replay assaults. This safeguard proves vital in secure-messaging platforms, VPN tunnels, and over-the-air protocols, where adversarial persistence constitutes a recurring threat vector.

ACKNOWLEDGMENTS

The authors would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq) in Baghdad, Iraq, for its support in the present work.

REFERENCES

1. D. Guo, K. Cao, J. Xiong, D. Ma, and H. Zhao, IEEE Internet Things J. 8, 12137–12149 (2021). https://doi.org/10.1109/JIOT.2021.3061700

2. Y. Hua, Signal Process. 227, 109744 (2025). https://doi.org/10.1016/j.sigpro.2024.109744.

3. Z. O. Ahmed and A. A. Khorsheed, Indones. J. Electr. Eng. Comput. Sci. 21, 208–214 (2021). https://doi.org/10.11591/ijeecs.v21.i1.pp208-214.

Received: August 11, 2025

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4. A. Sabah, S. Hameed, and K. Maisa'a-Abid-Ali, Al-Mustansiriyah J. Sci. 31, 41–46 (2020). https://doi.org/10.23851/mjs.v31i1.734

5. A. A. Shaban and I. M. Ibrahim, Int. J. Sci. World 11, 59 (2025). [DOI unclear – please verify]

6. B. A. Hameedi, M. A. Hatem, and J. N. Hasoon, JOIV Int. J. Inform. Vis. 8, 819–825 (2024). https://doi.org/10.30630/joiv.8.2.1741

7. S. N. Makhadmeh, M. A. Al-Betar, I. A. Doush, M. A. Awadallah, S. Kassaymeh, S. Mirjalili, and R. A. Zitar, IEEE Access 12, 22991–23028 (2023).

8. S. M. Almufti, H. B. Ahmad, R. B. Marqas, and R. R. Asaad, Int. Res. J. Sci. Technol. Educ. Manag. 1, 1 (2021).

9. W. Kaleem, M. Sajid, and R. Rajak, Int. J. Exp. Res. Rev. 31, 85–97 (2023).

10. J. Zhang, X. Liu, Y. Cui, and D. Xu, IEEE Access 10, 48539–48550 (2022). https://doi.org/10.1109/ACCESS.2022.3174909

11. E. D. Demir, B. Bilgin, and M. C. Onbasli, arXiv:2503.12952 (2025).

12. R. Banoth and R. Regar, Asymmetric Key Cryptography, in *Classical and Modern Cryptography for Beginners* (Springer, Cham, 2023), pp. 109–165. https://doi.org/10.1007/978-3-031-33067-1_6

13. S. Kumar, M. S. Gaur, P. S. Sharma, and D. Munjal, in *Proc. 2nd Int. Conf. Intell. Eng. Manag. (ICIEM)* (London, U.K., 2021), pp. 593–598.

14. P. Wang, L. You, G. Hu, L. Hu, Z. Jian, and C. Xing, Pattern Recognit. 111, 107733 (2021). https://doi.org/10.1016/j.patcog.2020.107733

15. G. Kodwani, S. Arora, and P. K. Atrey, in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)* (Rhodes, Greece, 2021), pp. 109–114.

16. M. H. Murtaza, H. Tahir, S. Tahir, Z. A. Alizai, Q. Riaz, and M. Hussain, J. Inf. Secur. Appl. 70, 103332 (2022). https://doi.org/10.1016/j.jisa.2022.103332

17. E. Alanzi and M. E. B. Menai, Artif. Intell. Rev. 58, 267 (2025).

18. U. J. Mele, L. M. Gambardella, and R. Montemanni, Algorithms 14, 267 (2021). https://doi.org/10.3390/a14090267

19. Z. Xu and X. Zhang, IAENG Int. J. Comput. Sci. 51, 6 (2024).

20. K. Li, S. Li, Z. Huang, M. Zhang, and Z. Xu, Sci. Rep. 12, 18961 (2022). https://doi.org/10.1038/s41598-022-22745-z

21. M. Ghalambaz, R. J. Yengejeh, and A. H. Davami, Case Stud. Therm. Eng. 27, 101250 (2021). https://doi.org/10.1016/j.csite.2021.101250

Received: August 11, 2025

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

22. M. Yousefikhoshbakht, Complexity 2021, 6668345 (2021). https://doi.org/10.1155/2021/6668345

23. M. H. Nadimi-Shahraki, E. Moeini, S. Taghian, and S. Mirjalili, J. Bionic Eng. 20, 2331–2358 (2023). https://doi.org/10.1007/s42235-023-00328-9

24. E. A. Luengo, B. A. Olivares, L. J. G. Villalba, and J. Hernandez-Castro, Appl. Math. Comput. 459, 128222 (2023). https://doi.org/10.1016/j.amc.2023.128222

25. D. Applegate, R. Bixby, V. Chvátal, and W. Cook, TSPLIB and TSP data sets, Univ. Waterloo, http://www.math.uwaterloo.ca/tsp/index.html

Received: August 11, 2025