IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 6, December 2026, pp. 6005~6014
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i6.pp6005-6014

Secure cloud healthcare data using blockchain and optimized blowfish encryption

Abdelilah Chahid[1,2], Youssef El Marzak[1], Khalifa Mansouri[1]
[1]Laboratory of Modeling and Simulation of Intelligent Industrial Systems, ENSET of Mohammedia, Hassan II University of Casablanca, Casablanca, Morocco
[2]Artificial Intelligence and Systems Laboratory, Faculty of Sciences Ben M'Sik, Hassan II University of Casablanca, Casablanca, Morocco

Article Info

Article history:

Received Oct 8, 2025
Revised Aug 13, 2026
Accepted Sep 20, 2026

Keywords:

Blockchain
Blowfish
Cloud
Elephant herding optimization-opposition-based learning
Healthcare

ABSTRACT

In cloud-based healthcare systems, protecting electronic health records (EHRs) is challenging due to centralized storage, unauthorized access, and cyberattacks. This paper proposes a hybrid security framework combining a blockchain layer for secure access management with an enhanced blowfish encryption scheme, whose session keys are generated using elephant herding optimization (EHO) with opposition-based learning (OLA). This integration improves both confidentiality and computational efficiency. For a 10 kb file, EHO-OLA reduces key generation time by 48.6%, 90.8%, 90.7%, and 90.9% compared with blowfish, Rivest–Shamir–Adleman (RSA), advanced encryption standard (AES), and elliptic curve cryptography (ECC), respectively, and outperforms EHO, moth-flame optimization (MFO), and whale optimization algorithm (WOA). At 60 kb, encryption time is reduced by 55.7%, 88.7%, 88.6%, and 88.8%, and decryption time by 56.5%, 88.5%, 88.4%, and 88.6%, against the same four schemes. Key sensitivity (0.89–0.93), entropy, number of pixels change rate (NPCR), unified average changing intensity (UACI), and adjacent-byte correlation confirm strong diffusion and randomness. The proposed framework offers a practical and efficient solution for secure cloud-based healthcare data management.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Abdelilah Chahid
Laboratory of Modeling and Simulation of Intelligent Industrial Systems, ENSET of Mohammedia
Hassan II University of Casablanca
Casablanca, Morocco
Email: chahidabdelillah@gmail.com

1. INTRODUCTION

Recent advances in digital health have spotlighted blockchain for its decentralized, tamper-resistant design. This promises more reliable management and secure exchange of electronic health records (EHRs) while tackling persistent issues of trust, security, and interoperability [1]–[3]. Alongside this, the internet of medical things (IoMT) and mobile cloud computing (MCC) enable patients to capture health data at home via wearables or smartphones and transmit it instantly to clinicians. This yields faster, more efficient, and more personalized care [4]–[6]. These capabilities bring clear benefits, most notably remote monitoring that reduces hospital visits and supports individualized treatment paths through cloud-backed EHRs [7], [8]. However, they also raise concerns. Hosting EHRs in the cloud increases exposure to cyberattacks that can jeopardize privacy and system integrity, and many patients struggle with records fragmented across multiple platforms [9], [10]. Addressing these risks requires robust, fine-grained access control for cloud-based EHRs. Without such safeguards, widespread adoption will remain limited despite the technology's promise [11], [12]. In response, this study combines a blockchain layer with tailored encryption

to secure cloud-hosted medical data. It upgrades the blowfish cipher for stronger and more efficient protection and presents elephant herding optimization (EHO) with opposition-based learning (OLA), a hybrid method for fast, secure generation of cryptographic secrets.

The privacy-preserving approaches adopted by the blockchain technology in health care environments adopt different techniques, all with their own pros and cons. For example, as observed in [13], there is a need to improve the security and privacy in electronic medical records; however, further research needs to be conducted for its evaluation in clinical practice [14]. De-centralized access control, which attempts to save time and ensure privacy, is still faced with inefficiencies and the need for more light-weighted solutions. Qi *et al.* [15] discusses internet of things (IoT) healthcare systems where emphasis is placed on strong privacy protection, but one of the biggest challenges is the problem of deployment of such systems on a large scale. Xi *et al.* [16] proves that blockchain technology is capable of ensuring secure sharing of EHRs but faces the problem of scalability and compatibility with other systems. Taherdoost [17] introduces a privacy-preserving blockchain approach with high levels of confidentiality, but its implementation is limited due to lack of standards and lack of real-world cases. Durneva *et al.* [18] uses public key infrastructure (PKI) encryption for healthcare data exchange; however, it faces problems with integration into modern environments. Solutions to blockchain privacy have been examined in [19]; however, issues like linkable transactions and data leakage still need to be addressed.

Despite the huge strides that have been made in terms of blockchain privacy protection, current solutions still suffer from some important limitations that include scalability issues, key management problems, and vulnerability to sophisticated crypto-attacks. To address the above limitations, this research introduces a novel hybrid approach that combines the power of blockchain with an advanced encryption scheme that involves the blowfish encryption algorithm and key generation based on EHO, which has been named EHO-OLA. Unlike other schemes, this approach has several advantages including efficiency, security, and compatibility.

The paper proceeds as follows. Section 2 details a cloud-centric model to reinforce data protection. Section 3 presents the EHO-OLA method and its role in optimizing key generation. Sections 4 presents the result of this study. Lastly, section 5 concludes the paper.

2. A PRIVACY PRESERVATION MODEL DESIGNED FOR SECURING HEALTHCARE SYSTEMS IN CLOUD ENVIRONMENTS

Conceptual model of trustless signature infrastructure blockchain (TSI-BC) is shown in Figure 1. It is aimed at securing patient records according to the following steps. Practical realization of the model will be considered later on.

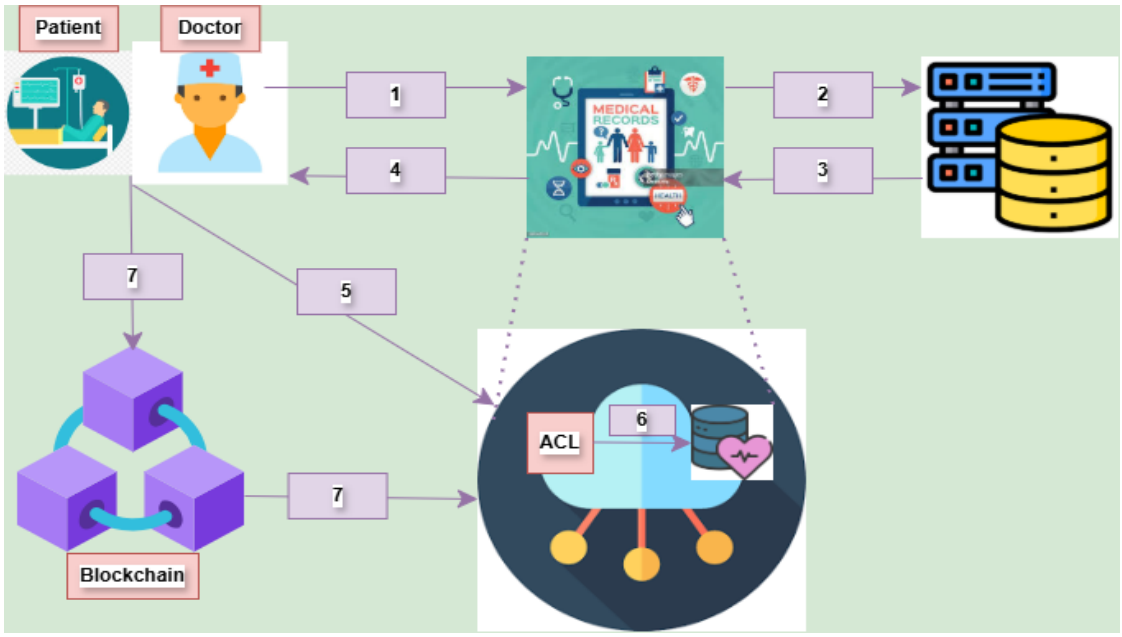


Figure 1. Secure access and data sharing workflow in a blockchain-based healthcare system

The workflow of the proposed model is described in the following steps:

i) Initial access request (steps 1 and 2): requests are made by both physicians and patients, with the use of a private key, that helps identify the user in the system. This key is then used for generating a digital signature that is authenticated by the system.
ii) Session key generation via EHO-OLA (steps 3 and 4): after successful authentication, EHO-OLA is applied for generating session keys. This session key is then used to protect all transactions that occur between the user and the cloud database.
iii) Secure access to data in the cloud (steps 5 and 6): the doctor or patient uses a session key for accessing the health database from the cloud storage that includes the confidential data about patients. Before accessing, access control list (ACL) is verified to check if the person has the required access rights [20].
iv) Data security with improved blowfish encryption and blockchain storage (step 7): the information is encrypted using the advanced blowfish technique before the exchange takes place between the user and the cloud database. The ciphertext is then stored (anchored) inside the blockchain in such a way that interception of the data does not allow for reading the intercepted data. The blockchain provides tamper resistance to the recorded medical transactions [21].

2.1. Improved blowfish algorithm with efficient key generation

Blowfish is a symmetric block cipher developed by Bruce Schneier as an efficient and fast substitute for modern ciphers. Blowfish is now widely acknowledged for its efficiency and strong security structure. It uses blocks of 64 bits in size and the variable-length key that consists of 32 - 448 bits, which allows flexibility depending on the security requirements. In the classic implementation, the 64-bit input is split into two 32-bit halves. In enhanced versions, it can handle 128-bit blocks, divided into four 32-bit segments, improving both performance and adaptability to modern computational needs. At its core, blowfish relies on simple yet powerful operations such as exclusive OR (XOR), substitutions, and modular arithmetic. This structure consists of four substitution boxes, or S-boxes, each holding 256 entries that contain 32 bits each, in addition to an array of 18 subkeys referred to as the P-array, wherein each subkey comprises 32 bits. The primary function of this encryption algorithm is the function $F(x)$ that breaks a 32-bit block into four 8-bit blocks denoted as a, b, c, d, as defined in (1).

$$F(x) = \left((S_1[a] + S_2[b]) \oplus S_3[c]\right) + S_4[d](mod\ 2^{32}) \quad (1)$$

Where x is a 32-bit input; a, b, c, d are the 8-bit segments of x; S1…, S4 are the S-boxes; $\oplus$ denotes XOR; and $mod\ 2^{32}$ indicates 2^{32}.

Improved models of blowfish introduce more advanced transformation functions, such as (2).

$$F_2(B) = \left((S_1[a] + S_2[b])\ mod\ 2^{32}\right) \times \left((S_3[c] + S_4[d])\ mod\ 2^{32}\right) mod\ 2^{32} \quad (2)$$

Where B is a 32-bit input block split into bytes a, b, c, and d; mod denotes the modulo operation, as defined in (3).

$$F_3(B) = (S_1[a] \times S_2[b] + S_3[c] \times S_4[d])\ mod\ 2^{32} \quad (3)$$

These variants F_2 and F_3 strengthen the mixing of data through more complex combinations of addition and multiplication, increasing resistance against cryptanalysis. Additionally, the use of larger key sizes and optimized block divisions enhances security and performance, making blowfish a highly versatile choice for encryption in various modern applications.

3. PROPOSED EHO-OLA ALGORITHM FOR KEY ENCODING AND THE OPTIMALIZATION OF THE OBJECTIVE FUNCTION

In this work, the EHO-OLA algorithm has been proposed to develop optimal keys to improve privacy assurance. The keys K are selected optimally to improve security in encryption. The input solution for the proposed algorithm framework has been shown in Figure 2, where T is the total number of keys under analysis. The objective function, represented by Obj, has been defined in (4). The main goal of the objective function is to maximize key-breaking resistance time (KBT).

$$Obj = Max(K_BT)\ K = [K_1, K_2, \dots, K_T] \quad (4)$$

Where K is the candidate key (or key vector), K_i is the i-th key component, T is the number of key components considered, and $K_BT(K)$ is the KBT induced by K. $K_BT(K)$ denotes the estimated computational

effort (or expected time) required by an adversary to recover the session key K using exhaustive search or cryptanalytic attacks. Since all candidate keys preserve the original key length and belong to the same cryptographic key space, the optimization process does not reduce key entropy. Instead, EHO-OLA selects candidate session keys that maximize resistance to cryptanalysis according to statistical security criteria while maintaining the randomness constraints imposed by the underlying key generation process.



Figure 2. Structure of the encoded solution for EHO-OLA-based key generation

3.1. Proposed elephant herding optimization-opposition-based learning algorithm

The EHO-OLA algorithm [22] is a widely recognized optimization method with a strong convergence rate for solving complex problems. This initialization improves convergence rates and accelerates the solution process [23], [24]. The algorithm presented here utilizes social dynamics seen in the behavior of elephants that live in clans headed by matriarch. The assumptions of this model are as follows:

i) The individuals are divided into various clans, comprising both males and females.
ii) Adult males often leave the herd to live alone.
iii) A matriarch leads the herd and determines its social structure.

The overall workflow of the proposed EHO-OLA algorithm is presented in Algorithm 1.

Algorithm 1. The proposed structured EHO-OLA model
Initialization
i) Specify the initial parameters:
    n: number of all individuals (elephants).
    nclan: number of clans (groups of elephants).
    nc: Number of elephants per clan, nc = (n/nclan).
    Factors used for adjustment::
    α : factor for updating the normal elephants.
    δ : factor for replacing the weak elephants.
    β : factor for updating the best elephant.
    Stopping criterion: Maximum number of iterations (tmaxt)
ii) Creation of initial population
    H={K1,K2,...,Kn}
iii) Calculation of the initial fitness values
    F(Ki) of each member of the population by means of the function of fitness
    F(K) = KBT.
Main iterations (repeat)
Repeat the following operations until the stop criteria is met:
i) Elephant organizing:
    Order each clan on the basis of their fitness level, from the best ($Z_{best,c}$) to the worst ($Z_{worst,c}$).
ii) Position updating process :
    For every clan c :
        For each elephant j in clan c:
            If the elephant is the best ($Z_{c,j}$=$Z_{best,c}$):
                Update its position by applying this equation:

$$Z_{n,c,j} = \beta \times Z_{center,c}$$

                where Z_{center} is the average position of the elephants in the clan c.
            Else :
                Update its position as:

$$Z_{n,c,j} = Z_{c,j} + \alpha \cdot r \cdot \left(Z_{best,c} - Z_{c,j}\right)$$

iii) Replace weak elephants:
For all clans, denoted by c:
Identify $Z_{worst,c}$ (the elephant with the lowest fitness).
Replace $Z_{worst,c}$ using $Z_{worst,c} = Z_{best,c} + \delta \cdot (Z_{best,c} - Z_{worst,c})$

iv) Evaluate new positions :
Compute the value of F(Ki) for all updated elephants.

3.2. Clan-updating operator

Taking into consideration the behavior of elephants, the matriarch takes the leadership position and greatly contributes to the formation of the spatial layout of all other clan members. From the viewpoint of the suggested theoretical model, provided that the previous fitness (PF) of elephant i of clan c is higher than its current fitness (CF), the contribution of the matriarch to the next configuration of the clan starts working. As a result, the new position of elephant j of clan c is calculated by (5), where $Z_{c,j}$ and $Z_{n,c,j}$ stand for the previous and new positions of elephant j of clan c, correspondingly.

$$Z_{n,c,j} = Z_{c,j} + a \cdot r.\left(Z_{best,c} - Z_{c,j}\right) \quad (5)$$

Where $Z_{c,j}$ is the current position of elephant j in clan c; $Z_{n,c,j}$ is its updated position, $Z_{best,c}$ is the best position in clan c; α∈ (0, 1) is the update factor; and r∼U (0, 1) is a uniform random number.

However, the position of the dominant elephant in the clan does not change with (5). Instead, the position of the dominant elephant gets modified using (6), where β is the parameter lying between [0, 1] and $Z_{center,c}$ is the center of clan c.

$$Z_{n,c,j} = \beta \times Z_{center,c} \quad (6)$$

Where $Z_{n,c,j}$ is the updated position of the best elephant in clan c, β∈ (0, 1) is a scaling factor, and $Z_{center,c}$ is the clan center (mean position of elephants in clan ccc).

4. RESULTS AND DISCUSSION

4.1. Experimental setup and benchmark protocol

Evaluation of the newly designed model for providing security and privacy in the cloud environment based on healthcare, which is dependent on the EHO-OLA algorithm, was performed using a computer system having the Intel® Core™ i7-12700H processor at 2.70 GHz, and 16 GB of memory under the operating system of Windows 11 (64-bit). The algorithms were coded in the programming language of Python 3.10. The performance of the suggested algorithm has been evaluated in comparison to some conventional techniques, which include blowfish [25], Rivest–Shamir–Adleman (RSA) [26], advanced encryption standard (AES) [27], elliptic curve cryptography (ECC) [28], EHO [22], moth-flame optimization (MFO) [29], and whale optimization algorithm (WOA) [30].

4.2. Dataset description

The dataset was sourced from https://archive.ics.uci.edu/ml/datasets/heart+disease. While the database contains 76 attributes, only a subset of 14 features has been consistently used in the literature. The Cleveland dataset remains the most reliable reference for machine learning-based studies on cardiovascular prediction. The output variable reflects disease presence, with values ranging from 0 (absence) to 4 (severe presence). In most works, the task is simplified into a binary classification problem: disease (1 – 4) vs. no disease (0).

4.3. Experimental performance analysis

The UCI heart disease dataset was selected because it represents realistic and widely used EHR content, making it suitable for simulating medical data in cloud-based healthcare environments. In this work, the dataset is not used for classification or prediction tasks, but rather as representative healthcare data to be encrypted and processed, in order to evaluate the behavior of the proposed security framework in a realistic e-health context. To validate scalability, experiments were conducted on multiple file sizes ranging from 10 kb to 60 kb. The results show that the proposed EHO-OLA based encryption scheme maintains stable and superior performance as file size increases, in terms of key generation, encryption, and decryption. This confirms the adaptability of the proposed framework to scalable cloud healthcare systems where medical data volumes may vary and grow over time.

4.3.1. Key generation performance

This section presents a performance comparison between the proposed EHO-OLA model and several well-known encryption algorithms namely blowfish, ECC, EHO, RSA, AES, MFO, and WOA across a range of file sizes (10 kb to 60 kb). The results demonstrate that the EHO-OLA approach delivers notably lower execution times, particularly in the key generation phase. For example, when processing a 10 kb file, the model achieves time reductions of 48.6%, 90.8%, 90.7%, 90.9%, 87.0%, 86.5%, 90.5%, and 91.1%, compared to blowfish, ECC, EHO, RSA, AES, MFO, and WOA, respectively (see Table 1). This performance advantage remains consistent as file size increases, highlighting the effectiveness and scalability of the EHO-OLA algorithm in optimizing cryptographic operations. This part will discuss the detailed performance analysis between the new EHO-OLA encryption model and other well-known encryption models such as blowfish, ECC, EHO, RSA, AES, MFO, and WOA for files whose size ranges between 10 kb to 60 kb. It has been found out through experiments that the new EHO-OLA encryption model always shows lower processing time for both encryption and decryption processes.

Table 1. Key generation performance comparison by file size

File size (kb)	Blowfish	EHO	MFO	WOA	RSA	AES	ECC	EHO-OLA
10	0.1247	0.4913	0.4762	0.6748	0.6976	0.6882	0.7067	0.0641
20	0.1372	0.5076	0.4938	0.7023	0.7194	0.7115	0.7281	0.0692
30	0.1498	0.5264	0.5127	0.7316	0.7418	0.7359	0.7495	0.0756
40	0.1631	0.5449	0.5314	0.7587	0.7653	0.7591	0.7726	0.0847
50	0.1776	0.5667	0.5521	0.7915	0.7917	0.7856	0.7989	0.0979
60	0.1928	0.5882	0.5743	0.8236	0.8198	0.8127	0.8274	0.1126

4.3.2. Encryption performance

As shown in Table 2, the suggested EHO-OLA algorithm is superior to other algorithms in terms of execution speed regardless of the file size under study. In the case of a 60 kb file size, the encryption execution speed is improved by 55.7%, 88.7%, 88.6%, and 88.8% compared with Blowfish, ECC, EHO, and MFO, respectively. Moreover, the proposed algorithm achieves improvements of 84.1%, 83.8%, and 88.6% over RSA, AES, and WOA, respectively.

Table 2. Encryption time (s) by file size

File size (kb)	Blowfish	EHO	MFO	WOA	RSA	AES	ECC	EHO-OLA
10	0.1198	0.4837	0.4689	0.6624	0.6915	0.6796	0.6987	0.0518
20	0.1365	0.5012	0.4876	0.6897	0.7138	0.7024	0.7213	0.0576
30	0.1517	0.5196	0.5068	0.7163	0.7362	0.7271	0.7448	0.0637
40	0.1694	0.5381	0.5257	0.7449	0.7597	0.7518	0.7682	0.0715
50	0.1872	0.5574	0.5453	0.7742	0.7841	0.7776	0.7935	0.0806
60	0.2068	0.5783	0.5661	0.8057	0.8106	0.8039	0.8194	0.0917

4.3.3. Decryption performance

In the same way, the results shown in Table 3 indicate that the proposed method performs better than other algorithms in the decryption phase as well. For a file size of 60 kb, the EHO-OLA algorithm outperforms Blowfish, RSA, AES, and ECC by 56.5%, 88.5%, 88.4%, and 88.6%, respectively. Moreover, it achieves improvements of 83.9%, 83.5%, and 88.4% over EHO, MFO, and WOA, respectively.

Table 3. Decryption time (s) by file size

File size (kb)	Blowfish	EHO	MFO	WOA	RSA	AES	ECC	EHO-OLA
10	0.1286	0.4972	0.4815	0.6853	0.7148	0.7017	0.7236	0.0547
20	0.1453	0.5158	0.5007	0.7136	0.7381	0.7265	0.7469	0.0609
30	0.1628	0.5351	0.5204	0.7418	0.7617	0.7524	0.7705	0.0675
40	0.1814	0.5547	0.5402	0.7713	0.7862	0.7781	0.7954	0.0758
50	0.2007	0.5753	0.5611	0.8016	0.8117	0.8049	0.8213	0.0852
60	0.2215	0.5971	0.5834	0.8342	0.8385	0.8316	0.8479	0.0964

4.3.4. Comparative analysis of cryptographic overhead in hybrid frameworks

Table 4 shows that the proposed blockchain + improved blowfish + EHO-OLA framework achieves the lowest runtimes among the listed approaches, with 0.1126 s for key generation and only 0.0917 s and

0.0964 s for encryption and decryption at 60 kb. Compared with hybrid attribute-based encryption-based schemes (attribute-based elliptical curve cryptography (ABECC) / key policy attribute-based encryption (KP-ABE) / ciphertext policy attribute-based encryption (CP-ABE)) [31] and recent asymmetric–symmetric hybrids (RSA/AES, RSA/ChaCha20, ECC/AES) [32], the method emphasizes lightweight session-key optimization and efficient symmetric processing. As a result, it offers a strong practical trade-off for cloud healthcare workloads where latency and resource constraints matter, while blockchain preserves integrity and auditability of medical transactions.

Table 4. Key generation, encryption, and decryption times of EHO-OLA compared to hybrid schemes

Reference/ Year	Framework	Conditions in the paper	Key generation (s)	Encryption (s)	Decryption (s)
Dwivedi *et al.* [31], 2021	ABECC vs KP-ABE / CP-ABE	300 kb, 6 attributes	0.3	0.6	0.6
	KP-ABE	300 kb, 6 attributes	0.5	1.0	1.0
	CP-ABE	300 kb, 6 attributes	0.4	0.8	0.8
Saydahd *et al.* [32], 2025	RSA + AES (hybrid)	Multi-format average	2.5328	0.2756	0.3178
	RSA + ChaCha20 (hybrid)	Multi-format average	2.1298	0.5736	0.1960
	ECC + AES (hybrid)	Multi-format average	0.7396	0.7326	0.1292
This work	Blockchain + improved blowfish + EHO-OLA (proposed)	60 kb (tested 10–60 kb)	0.1126	0.0917	0.0964

4.4. Convergence analysis

The convergence behavior of the proposed EHO-OLA scheme was evaluated over 100 iterations. As illustrated in Figure 3, the convergence curve of EHO-OLA decreased more rapidly than those of the compared methods, reaching a lower cost function at an earlier stage. By the 60th iteration, the proposed model demonstrated superior performance, outperforming EHO, MFO, and WOA by approximately 24.9%, 17.0%, and 34.6%, respectively.

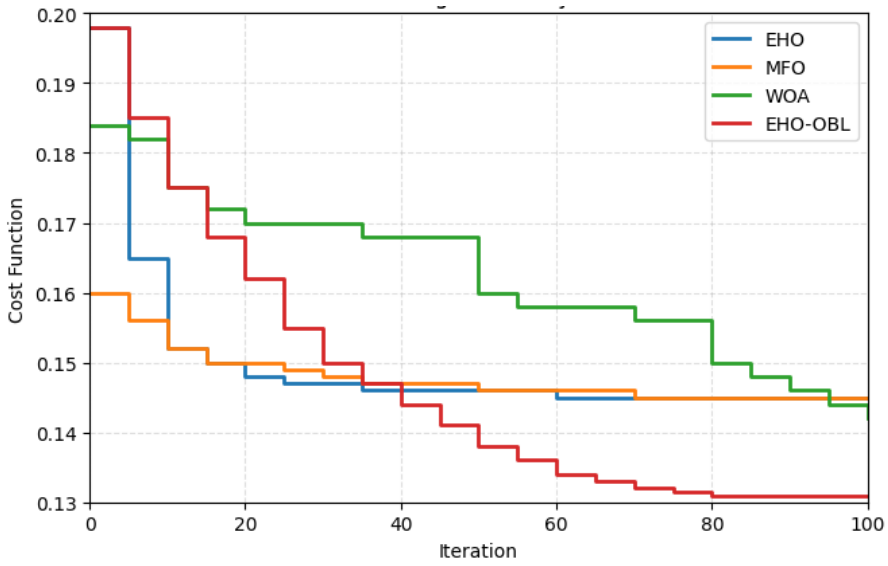


Figure 3. Convergence analysis

4.5. Key sensitivity analysis

Key sensitivity is measured by encrypting the same plaintext with two keys differing in a single bit, and computing the fraction of differing bytes between the two ciphertexts, as defined in (7).

$$KS = (1/L) \cdot \Sigma i \, \mathbb{1}[Ci \neq C'i] \quad (7)$$

Where L is the ciphertext length in bytes, C_i and C'_i denote the i-th bytes of the two ciphertexts, and $\mathbb{1}[\cdot]$ is the indicator function. Since KS operates at the byte level, its ideal value for an ideal cipher is

$1 - 2^{-8} \approx 0.996$, not the 0.5 expected under the bit-level strict avalanche criterion (SAC). As in Table 5, the EHO-OLA model consistently achieves a higher key sensitivity index (0.89 – 0.93) than the baseline methods (0.40 – 0.49), substantially closer to this theoretical optimum. This high byte-level dissimilarity indicates that even a single-bit change in the encryption key produces markedly different ciphertexts, supporting the robustness and unpredictability of the proposed scheme.

Table 5. Analysis of the model's sensitivity to variations in the encryption key

Size	Key sensitivity1	Key sensitivity2	Key sensitivity3	Key sensitivity4	EHO-OLA
16	0.4021	0.4268	0.4337	0.4234	0.8927
32	0.4237	0.4451	0.4586	0.4447	0.9185
64	0.4769	0.4712	0.4591	0.4305	0.9278
128	0.4860	0.4755	0.4669	0.4533	0.9288

4.6. Statistical security analysis

The results in Table 6 demonstrate that the proposed EHO-OLA based encryption scheme achieves the highest entropy among all compared algorithms. It also attains the strongest number of pixels change rate (NPCR) and unified average changing intensity (UACI) values. Furthermore, the scheme exhibits the lowest adjacent-byte correlation, confirming its superior randomness, diffusion capability, and resistance against statistical and differential cryptanalysis.

Table 6. Statistical security metrics of the proposed encryption scheme

Encryption scheme	Entropy (bits)	NPCR (%)	UACI (%)	Adjacent-byte correlation
Blowfish	7.742318	96.214521	28.436219	0.083214
EHO	7.865224	97.981553	30.821447	0.051332
MFO	7.893112	98.214338	31.104226	0.043981
WOA	7.901447	98.436772	31.356992	0.039214
RSA	7.698452	95.871104	27.902114	0.091337
AES	7.812947	97.103562	29.214883	0.072841
ECC	7.756138	96.487219	28.734510	0.079226
EHO-OLA	7.936972	99.548438	32.721488	0.003890

5. CONCLUSION

This study proposes a privacy-centric architecture to harden cloud-based healthcare systems, combining a blockchain ledger, an enhanced blowfish cipher, and a hybrid EHO-OLA driven secret-generation module. Benchmarks show consistent gains over conventional schemes—faster encryption/decryption, more efficient secret generation, and stronger resistance to cyber threats—while convergence studies and input-sensitivity tests confirm stability and a pronounced avalanche property, both critical for unpredictability and high security. Overall, the approach strikes a well-calibrated balance between robust protection and operational efficiency, aligning with the stringent requirements of cloud-native medical applications. Looking ahead, practical rollouts on telemedicine platforms and IoMT networks appear promising for secure, scalable, and adaptable health services. Future work should evaluate robustness under brute-force and ciphertext-only attack models not assessed here to illuminate edge-case behavior and guide targeted hardening. In particular, concrete mitigation measures can be integrated to further reduce brute-force exposure and improve ciphertext indistinguishability, including the use of standardized padding to mask structural patterns and message lengths, session-level randomization through a per-session nonce/initialization parameter, salt-based key derivation for session keys, and operational safeguards such as periodic key rotation and rate limiting on authentication and access requests. These additions can be incorporated without changing the overall architecture and will be systematically assessed in subsequent experiments.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Abdelilah Chahid	✓	✓	✓	✓		✓	✓	✓	✓				✓	
Youssef El Marzak		✓	✓	✓		✓		✓		✓	✓	✓	✓	✓
Khalifa Mansouri	✓			✓	✓		✓	✓	✓		✓		✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] R. Zou, X. Lv, and J. Zhao, "SPChain: Blockchain-based medical data sharing and privacy-preserving eHealth system," *Information Processing & Management*, vol. 58, no. 4, Jul. 2021, doi: 10.1016/j.ipm.2021.102604.

[2] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Blockchain leveraged decentralized IoT eHealth framework," *Internet of Things*, vol. 9, Mar. 2020, doi: 10.1016/j.iot.2020.100159.

[3] S. Tanwar, K. Parekh, and R. Evans, "Blockchain-based electronic healthcare record system for healthcare 4.0 applications," *Journal of Information Security and Applications*, vol. 50, Feb. 2020, doi: 10.1016/j.jisa.2019.102407.

[4] C. Gan, A. Saini, Q. Zhu, Y. Xiang, and Z. Zhang, "Blockchain-based access control scheme with incentive mechanism for eHealth systems: patient as supervisor," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30605–30621, Aug. 2021, doi: 10.1007/s11042-020-09322-6.

[5] Y. Han, Y. Zhang, and S. H. Vermund, "Blockchain technology for electronic health records," *International Journal of Environmental Research and Public Health*, vol. 19, no. 23, Nov. 2022, doi: 10.3390/ijerph192315577.

[6] N. Z. Bawany, T. Qamar, H. Tariq, and S. Adnan, "Integrating healthcare services using blockchain-based telehealth framework," *IEEE Access*, vol. 10, pp. 36505–36517, 2022, doi: 10.1109/ACCESS.2022.3161944.

[7] E. Chukwu and L. Garg, "A systematic review of blockchain in healthcare: frameworks, prototypes, and implementations," *IEEE Access*, vol. 8, pp. 21196–21214, 2020, doi: 10.1109/ACCESS.2020.2969881.

[8] A. Shahzad, W. Chen, M. Shaheen, Y. Zhang, and F. Ahmad, "A robust algorithm for authenticated health data access via blockchain and cloud computing," *PLOS ONE*, vol. 19, no. 9, Sep. 2024, doi: 10.1371/journal.pone.0307039.

[9] U. Chelladurai and S. Pandian, "A novel blockchain based electronic health record automation system for healthcare," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 693–703, Jan. 2022, doi: 10.1007/s12652-021-03163-3.

[10] P. P. Ray, D. Dash, K. Salah, and N. Kumar, "Blockchain for IoT-based healthcare: background, consensus, platforms, and use cases," *IEEE Systems Journal*, vol. 15, no. 1, pp. 85–94, Mar. 2021, doi: 10.1109/JSYST.2020.2963840.

[11] S. Dhingra, R. Raut, K. Naik, and K. Muduli, "Blockchain technology applications in healthcare supply chains—a review," *IEEE Access*, vol. 12, pp. 11230–11257, 2024, doi: 10.1109/ACCESS.2023.3348813.

[12] A. Amanat, M. Rizwan, C. Maple, Y. B. Zikria, A. S. Almadhor, and S. W. Kim, "Blockchain and cloud computing-based secure electronic healthcare records storage and sharing," *Frontiers in Public Health*, vol. 10, Jul. 2022, doi: 10.3389/fpubh.2022.938707.

[13] N. Raghav and A. Bhola, "Blockchain based privacy preservation in healthcare: a recent trends and challenges," *Psychology and Education Journal*, vol. 58, no. 1, pp. 5315–5324, Jan. 2021, doi: 10.17762/pae.v58i1.2122.

[14] Q. Liu, Y. Liu, M. Luo, D. He, H. Wang, and K.-K. R. Choo, "The security of blockchain-based medical systems: research challenges and opportunities," *IEEE Systems Journal*, vol. 16, no. 4, pp. 5741–5752, Dec. 2022, doi: 10.1109/JSYST.2022.3155156.

[15] M. Qi, Z. Wang, Q.-L. Han, J. Zhang, S. Chen, and Y. Xiang, "Privacy protection for blockchain-based healthcare IoT systems: a survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 8, pp. 1757–1776, Aug. 2024, doi: 10.1109/JAS.2022.106058.

[16] P. Xi, X. Zhang, L. Wang, W. Liu, and S. Peng, "A review of blockchain-based secure sharing of healthcare data," *Applied Sciences*, vol. 12, no. 15, Aug. 2022, doi: 10.3390/app12157912.

[17] H. Taherdoost, "Privacy and security of blockchain in healthcare: applications, challenges, and future perspectives," *Sci*, vol. 5, no. 4, Oct. 2023, doi: 10.3390/sci5040041.

[18] P. Durneva, K. Cousins, and M. Chen, "The current state of research, challenges, and future research directions of blockchain technology in patient care: systematic review," *Journal of Medical Internet Research*, vol. 22, no. 7, Jul. 2020, doi: 10.2196/18619.

[19] J. B. Bernabe, J. L. Canovas, J. L. H. -Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: review and challenges," *IEEE Access*, vol. 7, pp. 164908–164940, 2019, doi: 10.1109/ACCESS.2019.2950872.

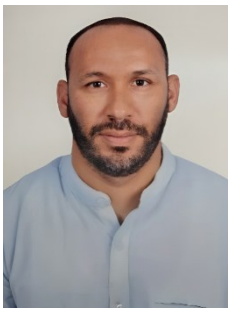
[20] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure EHRs sharing of mobile cloud based e-health systems," *IEEE Access*, vol. 7, pp. 66792–66806, May 2019, doi: 10.1109/ACCESS.2019.2917555.

[21] M. Agrawal and P. Mishra, "A modified approach for symmetric key cryptography based on blowfish algorithm," *International Journal of Engineering and Advanced Technology*, vol. 1, no. 6, pp. 79–83, 2012.

[22] G.-G. Wang, S. Deb, and L. D. S. Coelho, "Elephant herding optimization," in *2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, Dec. 2015, pp. 1–5, doi: 10.1109/ISCBI.2015.8.

[23] S. B. Halbhavi, S. F. Kodad, S. K. Ambekar, and D. Manjunath, "Enhanced invasive weed optimization algorithm with chaos theory for weightage based combined economic emission dispatch," *Journal of Computational Mechanics, Power System and Control*, vol. 2, no. 3, pp. 19–27, Jul. 2019, doi: 10.46253/jcmps.v2i3.a3.
[24] A. N. Jadhav and N. Gomathi, "DIGWO: Hybridization of dragonfly algorithm with improved grey wolf optimization algorithm for data clustering," *Multimedia Research*, vol. 2, no. 3, Jul. 2019, doi: 10.46253/j.mr.v2i3.a1.
[25] S. Kumar, D. Kumar, and N. Singh, "Performance and security analysis using B-128 modified blowfish algorithm," *Multimedia Tools and Applications*, vol. 82, no. 17, pp. 26661–26678, Jul. 2023, doi: 10.1007/s11042-023-15038-0.
[26] K. Sharma, A. Agrawal, D. Pandey, R. A. Khan, and S. K. Dinkar, "RSA based encryption approach for preserving confidentiality of big data," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 2088–2097, May 2022, doi: 10.1016/j.jksuci.2019.10.006.
[27] V. Nandan and R. G. S. Rao, "Minimization of digital logic gates and ultra-low power AES encryption core in 180CMOS technology," *Microprocessors and Microsystems*, vol. 74, Apr. 2020, doi: 10.1016/j.micpro.2020.103000.
[28] K. Sowjanya and M. Dasgupta, "A ciphertext-policy attribute based encryption scheme for wireless body area networks based on ECC," *Journal of Information Security and Applications*, vol. 54, Oct. 2020, doi: 10.1016/j.jisa.2020.102559.
[29] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, Nov. 2015, doi: 10.1016/j.knosys.2015.07.006.
[30] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.
[31] R. K. Dwivedi, R. Kumar, and R. Buyya, "Secure healthcare monitoring sensor cloud with attribute-based elliptical curve cryptography," *International Journal of Cloud Applications and Computing*, vol. 11, no. 3, pp. 1–18, Jul. 2021, doi: 10.4018/IJCAC.2021070101.
[32] S. J. Saydahd, R. K. Muhammed, S. A. Hassan, and A. M. Aladdin, "A comparative performance evaluation of hybrid encryption techniques using ECC, RSA, AES, and ChaCha20 for secure data transmission," *Iraqi Journal of Industrial Research*, vol. 12, no. 2, pp. 157–172, Dec. 2025, doi: 10.53523/ijoirVol12I2ID598.

BIOGRAPHIES OF AUTHORS



Abdelilah Chahid is a member of the Laboratory of Modeling and Simulation of Intelligent Industrial Systems at ENSET of Mohammedia and of the Artificial Intelligence and Systems Laboratory at the Faculty of Sciences Ben M'Sik, both at Hassan II University of Casablanca, Morocco. He holds a Ph.D. in Information Systems Governance from ENSET Mohammedia and a master's degree in Computer Networks from Hassan II University of Casablanca (2011). His research interests include IS governance in higher education, cybersecurity, network security, and AI-driven threat detection. He can be contacted at email: chahidabdelillah@gmail.com.



Youssef El Marzak holds a Ph.D. in Computer Science and Cybersecurity from the Higher Normal School of Technical Education of Mohammedia, Hassan II University of Casablanca, Morocco. His research focuses on information systems, cybersecurity, and digital governance. He can be contacted at email: youssef.elmarzak@gmail.com or youssef.elmarzak-etu@etu.univh2c.ma.



Khalifa Mansouri is currently a researcher-professor in Computer Science, training director, and director of the Laboratory of Modeling and Simulation of Intelligent Industrial Systems at ENSET of Mohammedia, Hassan II University of Casablanca. His research interests include information systems, e-learning systems, real-time systems, artificial intelligence, and industrial systems (modeling, optimization, and numerical computation). Graduated from ENSET of Mohammedia in 1991, CEA in 1992, and Ph.D. (Computation and Optimization of Structures) in 1994, HDR in 2010, and National Ph.D. (in computer science - distributed systems) in 2016. He is the author of 10 books in computer science, a scientific book with the publisher Springer, 425 research papers including 236 in the Scopus library, and supervised 35 defended doctoral theses. He can be contacted at email: khalifa.mansouri@enset-media.ac.ma.