University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Department of Computer Science, University of Ibadan, Ibadan, Nigeria

Volume 14 No. 1, June, 2025

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/
uijslictr@gmail.com

University of Ibadan
Journal of Science and Logics in ICT Research

An Hyperelliptic with Cellular Automata Encryption-based Data Protection Technique for Cloud Systems

[1]✉ Raheem M. and [2] Adeyemo A. B.

[1,2]University of Ibadan, Ibadan, Nigeria
[1]mmoshood@gmail.com; [2]sesanadeyemo2014@gmail.com

Abstract
The difficulty of data protection and privacy substantially inhibits the adoption of cloud technology, despite its popularity for efficiently providing access to data and other computing resources over the internet. Existing cloud data protection techniques have been found of failing to achieve lower computational costs, effective management of lengthy keys, efficient parallel computation, and the maintenance of balance between reliability of data protection and overhead. This research therefore, developed an improved encryption technique (HeCA) for achieving a high level of security with minimal computational cost and complexity in cloud-based systems. The HeCA was developed by combining a CA-based encryption/decryption, and a hyper-elliptic based signcryption/unsigncryption algorithms, that were formulated for the encryption/decryption of text and image data, and signcription/unsincryption of the secret keys respectively. The performance of the developed technique in comparison to existing BCAL and CA-RSA encryption techniques, was evaluated in the MATLAB R2020a environment using key length, processing time, and throughput as metrics. The results show that the developed HeCA technique outperforms the SBCAL and CA-RSA schemes in terms of key length, processing time and throughput. It is therefore concluded that the development of the HeCA has offered a superior data protection technique for cloud-based systems.

Keywords: *Cloud technology, Encryption technique, Data protection, Secret keys, HeCA*

1. Introduction

In the current technologically advanced world, information security is now a necessary and essential concern [1]. In modern circumstances, the top priority requirement is the privacy of data and resources across conventional and cloud-based networks. Therefore, more advanced and effective security methods are required to secure them because a sizable volume of information is generated, transported, and kept [2].

Cloud computing is spreading throughout the society because it provides quick and effective on-demand services for hardware, software, networks, and storage over the internet. Cloud computing gives new resources to businesses, organizations, and the public at large, as well as affordable computing resources for IT-based services [3]. The use of networks for communication in organizations like colleges, research institutes, and the military in developed nations is nothing new, but the term "cloud computing" is relatively recent [4]. More users are relying on third-party storage as a result of the widespread adoption of cloud computing on the internet because it lowers costs and makes information easier to share and access. Cloud computing is perceived as the most utilized resource after gas, water, electricity, and telephones [5].

Cloud provides different categories of services via its service models like Infrastructure as a service (e. g. Amazon's EC2, IBM Blue cloud), Platform as a Service (e. g. Google App Engine), and Software as a Software (e. g. saleforce.com, Gmail). User can access these services on pay- as-you-go basis without engaging the services of IT professionals or procuring hardware / software systems [6]. Despite the advanced nature of cloud technology, the difficulty of data protection and privacy substantially inhibits its adoption [7].

Raheem, M. and Adeyemo, A. B. (2025). An Hyperelliptic with Cellular Automata Encryption-based Data Protection Technique for Cloud Systems. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 14 No. 1, pp. 188 - 198

As a result of security flaws in its architecture, users are hesitant to move their data to the cloud. In addition to the rise of cloud-based on-demand apps, cybercrime has also risen, causing a rise in both latent and aggressive attacks. To keep the cloud safe and confidential, a number of distinct methods or protection algorithms are employed. Encryption, restricted access, controlled service access, and data backup and recovery to facilitate data retrieval are a few of these. A crucial approach to ensuring the secrecy and privacy of data obtained from the cloud is the adoption of an encryption method that offers sufficiently strong security [3].

Encryption is germane to the cloud computing system due to frequent transmission of data between the resources of the cloud and from local computers to the cloud computing platform through the internet. Data being transferred in plain text poses a serious security and data privacy concern because the entire transmission takes place via the internet. This data can be intercepted through networks by hackers in a variety of ways which may result in individuals, organizations, industries, institutions losing asset, reputation and customers' transactions data. Thus, necessitating greater data security measures like encryption, which prevents data from being read in transit [8].

A number of encryption methods, including RSA, AES, FHE, elliptic and hyperelliptic curve-based encryption, and cellular automata encryption algorithms, that researchers proposed to safeguard transferred cloud data have been criticized due to their shortcomings. RSA, bilinear pairing and elliptic curves provided protection for data over the internet with high cost of computation and computational complexity. This was remedied using the Hyper-elliptic Curve Cryptosystem (HECC) [9]. The proposed approach utilized certificates in generating public keys for producing signed encrypted text, which is associated with the key escrow challenge, whose eradication could result in lower computational costs and computation overhead [10]. Moreover, quantum computing poses a possible and increasing threat to current encryption techniques. These elements suggest increased inefficiency, necessitating the use of state-of-the-art encryption methods that can safely handle massive volumes of data without sacrificing performance [11].

A comparative analysis of some fully homomorphic encryption approaches like SDC FHE, RSA, and Paillier revealed that, in spite of their suitability for effectively protecting data stored in the cloud, there was a need to find a way to shorten public and private keys because they were lengthy and cryptographic algorithms had trouble managing huge keys [12]. This challenge was alleviated by the intervention of a reversible cellular automata-based throughput-optimized block encryption strategy that provides parallel computation via the exchange of different keys between senders and receivers. But could not attain the high level of protection required by smart devices that are highly susceptible to attacks by hackers [2].

The identified obstacles to cloud users' adoption of current encryption algorithms are high computational costs, difficulty in managing lengthy keys, convoluted procedures, lack of efficient parallel computation, and suboptimal maintenance of the balance between reliability of data protection and overhead in terms of communication and computing resources. As such, by integrating the hyper-elliptic encryption approach with the cellular automata encryption technique, an improved encryption technique for achieving a high level of security with little computational cost and complexity in cloud-based systems was developed in this research.

The other sections of the paper are arranged as follows: Section 2 provides a review of some recent related works; Section 3 discusses the architecture, design, implementation, and performance evaluation of the developed HeCA encryption technique; Section 4 presents and discusses the results of the performance evaluation of the developed technique in comparison to two state of the art encryption techniques; and Section 5 presents the major findings of the research, strength, and limitations of the developed technique and areas for future research.

2. Related Works

Mantri *et al*. [13] developed an encryption scheme for encrypting text and image information by utilising the concept of hybrid cellular automata along with Caesar Cypher. The RSA public key algorithm was used to encode the symmetric key. The scheme is proposed on the basis of changing the pixel values of a given image by using iteration of cellular automata rules a specified number of times. The image is encrypted using a symmetric key then the symmetric key is again encrypted using a public key scheme, which makes the scheme more robust. The encrypted images produced by the scheme have histograms that are remarkably distinct from that of the original images. This establishes the achievement of the confusion property by the scheme. The scheme also achieved a coefficient correlation between original and encrypted images of close to 1, which confirms that both the images are the same without any alteration of the value of the pixels. Lastly, the encrypted image has the entropy value of 7.9227 in the course of evaluation of the scheme, which is close to 8. Hence, a random cipher-image which cannot be predicted is generated using the scheme. However, the authors recommended the use of elliptic curve cryptography in place of RSA for future work.

A methodology for improved security and owner's data privacy in cloud computing was presented by Awan *et al*. [14]. Using the double round key feature, they tweaked the 128-bit Advanced Encryption Standard (AES) method to accelerate encryption to 1000 blocks per second. The proposed architecture reduces energy use by 14.43%, network utilization by 11.53%, and delay by 15.67%, according to the data. As a result, the suggested approach reduces latency while installing services in computational clouds and improves security and resource efficiency.

Nanda *et al.* [2] developed a throughput-optimized block encryption strategy employing reversible cellular automata in light of the paucity of research on software implementation strategies that can enhance the performance of encryption techniques. The disadvantages of the prior algorithms, which included complicated procedures, a lack of efficient parallel computation, and the need for numerous keys, were addressed by the suggested technique using private key cryptography. Key-1 and Key-2 are two pairs of reversible keys that the algorithm uses. Keys 1 and 2 are both secret and exchanged between senders and receivers. The encryption of 128-bit blocks of data forms the basis of this approach, but it is easily adapted to increased amounts of bits that are multiples of two. The technique is easy and employs simple operations, which leads to modularity and enables us to supply separate blocks of plaintext for execution in various cores to increase throughput. However, this technique can be improved by the introduction of an alternative approach where the sender and recipient can use different keys in different batches.

Naskar *et al.* [1] recommended utilizing chaotic tent maps and cellular automata to encrypt unique images in an effort to create a cryptosystem that can offer excellent security, low computing overhead and computational power, great sensitivity and pseudo-randomness qualities. The suggested approach is based on block ciphering, followed by shuffling of the ciphered bytes, where each block utilizes a distinct key stream of varying size comparable to the associated block size, which distinguishes the technique from other modern strategies in the literature. The chaotic tent map and elementary cellular automata (ECA) are employed to cipher each particular block using the individual key stream. The ciphered block's bytes are then further scrambled to further obscure the algorithm. Images that were both plain and encrypted were used in various tests. The genuine and encrypted images have a 0.000479 and 99.620901 correlation and NPCR, respectively, which both attest to the resilience of the system.

The key stream for the first block is created using a 64-byte secret key and a 64-bit precision value taken from the plain picture. This creates a key space of size 2576, which is significantly resistant against potential brute-force assaults. Low UACI (33.365006) and high entropy (7.998461), almost flat histograms of the encrypted images—all of these factors contribute to the scheme's resistance against potential statistical assaults. To meet the current need, the suggested system is a better option than image encryption. But in place of the ECA, 2-dimensional cellular automata that has been determined to be difficult to hack could be used to strengthen the suggested approach even further.

In order to build a cryptosystem that can achieve previously unheard-of-levels of security, Kumar and Ragbava [15] introduced an image encryption method that makes use of a one-dimensional elementary cellular automaton (ECA) and Henon chaotic map. The proposed algorithm is based on the Lookup Table scheme, which allows many low-resource and restrained devices to communicate, compute, and make choices in communication networks while executing with limited resource capabilities compared to most other methodologies that are built around the concepts of block and stream cipher. The elementary cellular automata (ECA) rule space, which is established as a table lookup for the cryptographic protocol, is used to examine and extract state indicators and transitions. Through simulation results and comparisons with other approaches, it is clear that the suggested algorithm is effective and resistant to all types of statistical attacks, yielding security supremacy in a number of cryptographic applications.

The principle of encryption and decryption is built on index-based lookup tables employing ECA, and it is simple to implement with logic gates. Based on experimental findings, it is clear that the suggested algorithm is trustworthy, resilient, and flexible in IoT and sensor networks and can defend images against both passive and active attacks in open channels of communication. By leveraging steganography methods for generating the cryptosystem for the sensor's real-time data, the study can be advanced.

A cellular automata-based visual cryptographic encryption scheme was suggested by Venugopal and Rajan [16] to encrypt images defined by pixels. This strategy is distinct from all graphic plans put forward up to this point. In order to create the cipher picture and ultimately decrypt the original data, the suggested method first encrypted plain text using a cellular automata algorithm of dimension 2. The suggested method improves the security elements of smart devices, such as 1-button-based access control and communication, which researchers discovered could be easily hacked. The proposed solution solved this by introducing cellular automata-driven visual cryptography, the complexity of which has been confirmed and shown to make it hard for a hacker to quickly decipher the information in signatures. Its effectiveness could be increased by combining it with a hyper-elliptic cryptography method to reduce communication and computational overhead while maintaining a better level of security.

Alexan *et al.* [17] proposed a novel colour image encryption scheme that combines ideas from chaos theory and CA to address the data security issues resulting from exponential growth in transmission of multimedia over the Internet and unsecured channels of communications. The proposed scheme is implemented over three stages. The first stage makes use of Rule 30 cellular automata to generate the first encryption key. The second stage utilises a well-tested S-box, whose design involves a transformation, modular inverses, and permutation.

Finally, the third stage employs a solution of the Lorenz system to generate the second encryption key and diffusion properties of a cryptographic system and enhances the security and robustness of the resulting encrypted images. Specifically, the use of the PRNG bitstreams from both of the cellular automata and the Lorenz system as keys, combined with the S-box, results in the needed non-linearity and complexity inherent in well-encrypted images, which is sufficient to frustrate attackers. Performance evaluation of the proposed scheme using statistical and sensitivity analyses results in an MSE value of 8923.03, a PSNR value of 8.625 dB, an information entropy of 7.999, an NPCR value of 99.627, and a UACI value of 33.46. The proposed scheme is shown to encrypt images at an average rate of 0.61 Mbps. The proposed scheme showcased a superior performance when compared with counterpart image encryption schemes from the literature. However, future work could be done to find a dynamical system with a trade-off among high ergodicity, improved distribution in phase space, and low computational complexity.

Ike *et al.* [18] proposed a novel approach to cloud data encryption using Homomorphic Encryption (HE), which enables computations on encrypted data without requiring decryption. The proposed system integrates optimized HE algorithms to reduce computational overhead, addressing one of the key challenges in HE adoption. They evaluated the security and performance of the approach by implementing a case study on encrypted data analytics in a cloud environment. The experimental results show that although HE adds computational complexity, its ability for practical applications is greatly increased by current developments in hardware acceleration and algorithm optimization. The suggested strategy makes use of all the capabilities of the cloud while guaranteeing data secrecy. Future research could concentrate on enhancing efficiency, scalability, and hybrid cryptographic models.

The review of related works on cryptographic techniques and cloud-based data encryption algorithms revealed that the latter need to be enhanced to achieve better results. A major finding of the review uncovered the need to combine the strengths of existing encryption techniques to realise a more secured cloud data protection system with minimal computational complexity and cost. This research was able to address this by developing a novel hybridized data protection technique (HeCA) via the combination of a CA based approach and a hyper-elliptic scheme that were utilized for the encryption/decryption of text and image data, and signcryption/unsigncryption of the secret keys respectively.

3. Methodology

3.1 The Developed HeCA Based Cloud Data Security Technique

The framework of the developed model is shown in Figure 1. The HeCA model is a hybridization of the Hyperelliptic curve (HEC) and Cellular Automata (CA) cryptography techniques. It entails an identity-based encryption for verification of data integrity by the user. The model consists of four entities namely Data source (signcrypter), Certificate Authority Centre (CAC), Cloud Storage/Server, and Receiver (unsigncrypter). The CA uses the Secret Keys (SK) to encrypt the data/message to be sent; and the HEC uses the ID$_r$ (public key of the unsigncrypter), d$_s$ (private key of the signcrypter) and the other cloud public parameters (cpp) to signcrypt the SK. The Data source (signcrypter) uploads both the signcrypted SK and CA-encrypted data to the cloud storage/server for access by the legitimate Receiver (unsigncrypter). Signcryption denotes signature plus encryption. The cloud public parameters (cpp) are the divisor of the HEC ($\mathcal{D}$), the irreversible hash functions (h_1, h_2, h_3), the encryption and decryption functions ($\mathcal{E}/\partial$), a hyper-elliptic curve over the field f_q (HEC (f_q)), and a large prime number (q).

The process involves the following steps:

i. At the beginning of the process, the signcrypter and the unsigncrypter generate their public keys ID$_s$ and ID$_r$, respectively, and forward them to the CAC. The signcrypter and the unsigncrypter generate their public keys as $ID_s = d_s.\mathcal{D}$ and $ID_r = d_r.\mathcal{D}$, respectively, where $\mathcal{D}$ is the divisor on a HEC.

ii. The CAC makes the cloud public parameters available to all users.

iii. The Data source (or signcrypter) generates two reversible CA rules, K_1 and K_2, as part of the Secret Keys (SK); and then carries out one-dimensional (1-D) CA encryption of the data using K_1.

iv. The signcrypter then utilizes its private key (d_s) and the public key of unsigncrypter (ID_r) to carry out HEC signcryption of the SK.

v. The signcrypter then forwards the signcrypted text (i.e. SK) and cipher text (i.e. data/message) to Cloud Storage. A cloud storage (or server) stores and processes data for the users.

vi. The Receiver (unsigncrypter) that is requesting the message receives the cipher texts from the Cloud Storage, unsigncrypts the SK via HEC by utilizing its private key (d_r) and the public key of the signcrypter (ID_s). It then uses the SK to decrypt the encrypted message via CA. The CA utilizes the SK parameters K_1, K_2 and N; where N is the number of transitions (or repetitions) of the cells' state. The parameters K_2 and N are used by the CA for decrypting the cipher text.

The reversed CA rule K_2 is obtained from K_1 as:

$$K_2 = 2^d - K_1 - 1 \qquad (1)$$

with

$$d = 2^{2r+1} \qquad (2)$$

where $r\ (= 1)$ is the radius for the neighbourhood of the 1-D CA. That is, for $K_1 = 102$ and $d = 8$; $K_2 = 2^8 - 102 - 1 = 153$. Thus, $[K_1, K_2] = [102, 153]$ is a pair of reversible rules. The list of reversible rules used in this study is contained in Table 1.

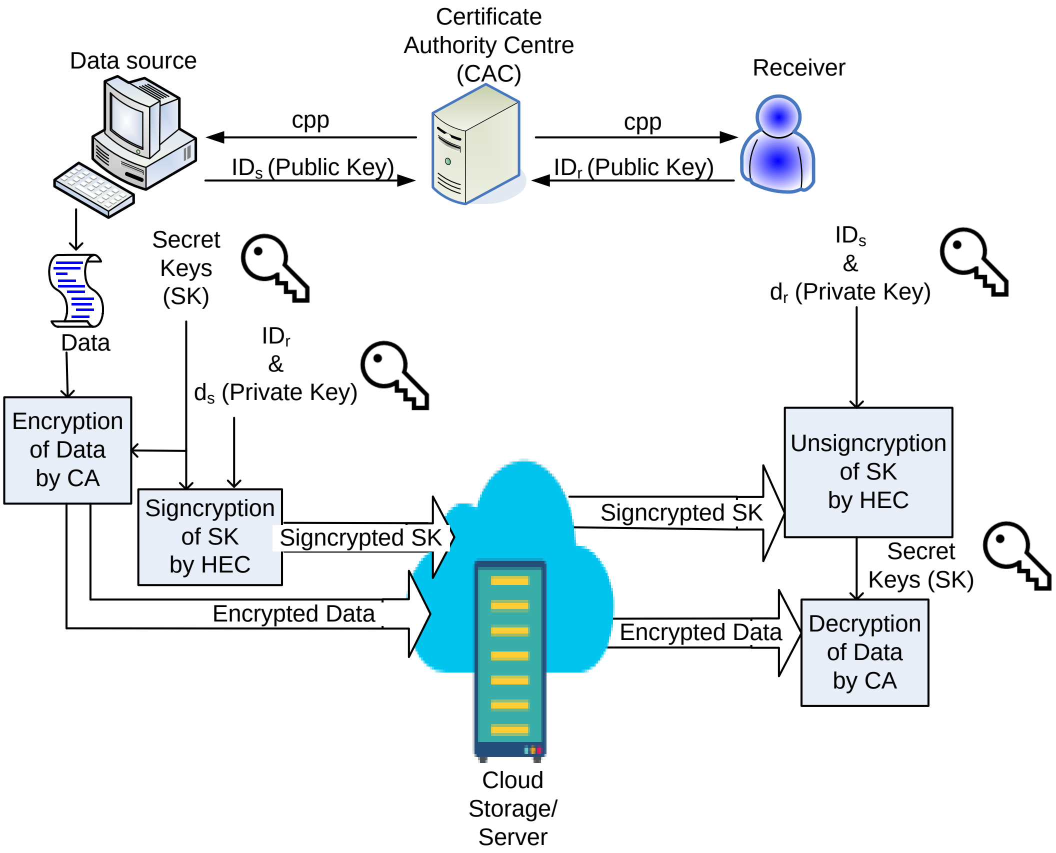
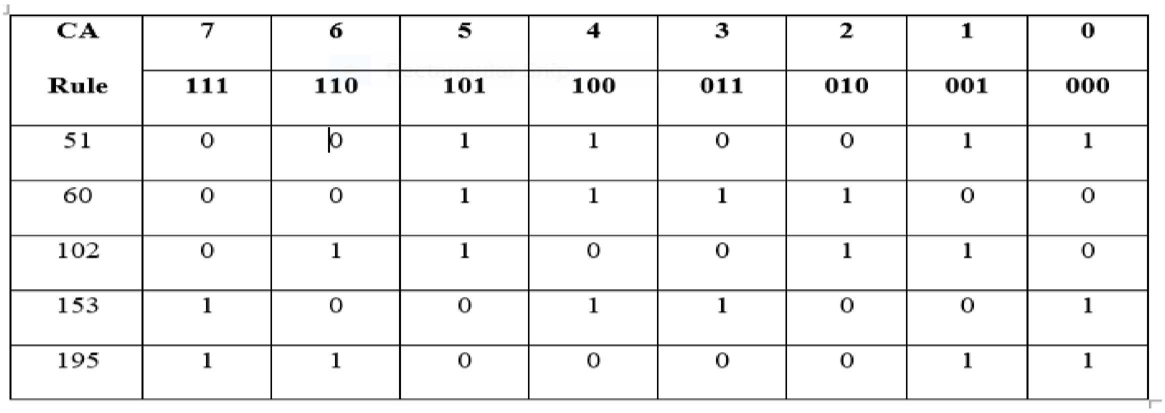


Figure1: Framework of the Developed HeCA Based Cloud Data Protection Model

Table 1: Reversible CA rules' list (Roy and Nandi [2])



CA	7	6	5	4	3	2	1	0
Rule	111	110	101	100	011	010	001	000
51	0	0	1	1	0	0	1	1
60	0	0	1	1	1	1	0	0
102	0	1	1	0	0	1	1	0
153	1	0	0	1	1	0	0	1
195	1	1	0	0	0	0	1	1

The pair of reversible CA rules were used as the Secret Keys (SK) which are randomly selected from the list of reversible CA rules. A pair of reversible CA rules as presented in Table 1 involves one rule helping to transit in the forward direction (encryption) and the other helping to transit in the reverse direction (decryption). During the transition, the state of every cell is updated or changed to achieve encryption. The number of times to transit or update the cells is chosen randomly at the beginning of the process and applied to all the cells equally. The CA cryptographic procedure for the developed technique is depicted in Figure 2.

Hyper Elliptic Curve (HEC) is adopted to signcrypt the Secret Keys (SK) owing to its key size of 80 bits as against the Elliptic Curve (EC) that requires 160 bits for the same level of security. Thus, the HEC is suitable for low-power devices. Let C be a HEC of genus g defined over a finite field K, and let J be the Jacobian of C. Let $P = (x, y) \in C$, and let σ be an automorphism of $\overline{K}$ over K; then $P^{\sigma} = (x^{\sigma}, y^{\sigma})$ is also a point on C. Assume that g is over a finite field f_q, where the order of this field is q. Figure 3 represents HeCA encryption and signcryption implementation for the developed technique.

Furthermore, if $g = 1$ (i.e. EC), then the group order of f_q is $g.log_2(q^{2^{160}})$. If $g = 2$ (i.e. HEC) then the curve will require a field f_q with $|f_q| = 2^{80}$, which means that it needs an 80-bit key. Suppose f^* is the algebraic closer of a finite field f of the HEC, then, the HEC of $g > 1$ over f representing the solution set $(\alpha, \beta) \in f^* f$ is given as:

$$\beta^2 + h(\alpha)\beta = f(\alpha) \ mod \ q \qquad (3)$$

where

$h(\alpha) \in f[\alpha]$ is a polynomial and the degree is $h(\alpha) \leq g$

$f(\alpha) \in f[\alpha]$ is the monic polynomial, and the degree is $f(\alpha) \leq 2g + 1$

q is a large prime number

The HEC works on divisor $\mathcal{D}$ which is the formal and finite sum of points on an HEC. With $\mathcal{D}$ of order q from the group of Jacobian (f_q) and an equation $\mathcal{D}_1 = L \cdot \mathcal{D}$ where $L \in f_q$, finding the integer L is called HEC discrete logarithm problem. The Secret Keys (SK) are denoted by $m = [K_1, K_2, N]$ if Text Data is to be sent; or by $m = [K_1, K_2, N, K_c]$ if Image Data is to be sent as shown in Figure 3. The signcryption and unsigncryption of m are illustrated in Figures 10 and 11 respectively.

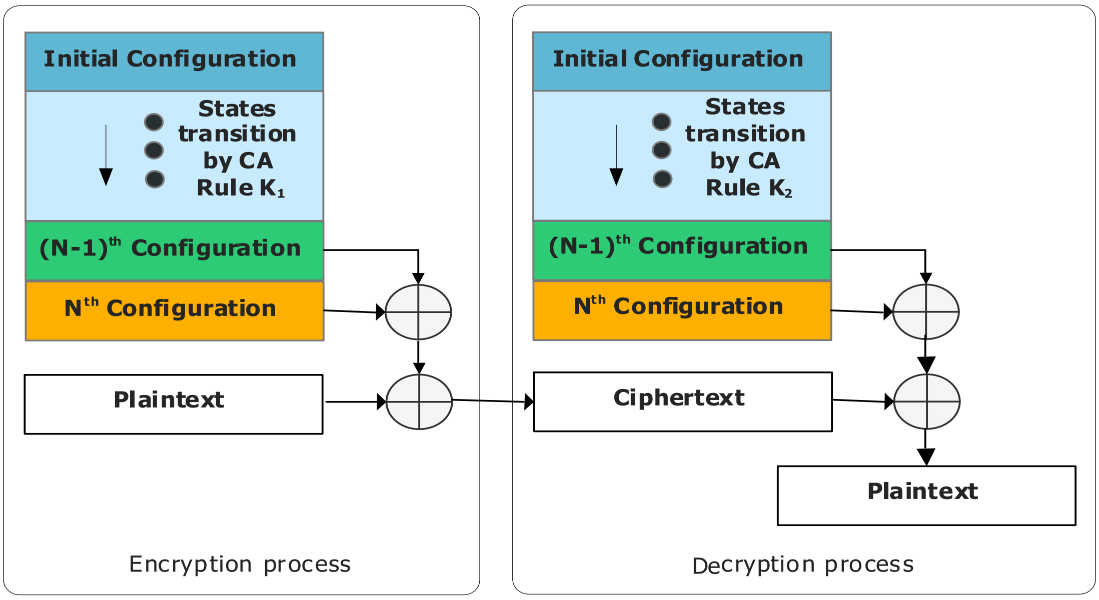


Figure 2: Block diagram of the CA cryptography for the developed model

The signcryption algorithm utilizes the public key of unsigncrypter (ID_s), divisor ($\mathcal{D}$), private key of signcrypter (d_s) to signcrypt the message (m). It then uploads the cipher text and signatures as ε and X, S, respectively, to the cloud server. The receiver takes the ciphertext (ε), signature (X, K, S), its private key (d_r), divisor ($\mathcal{D}$) and public key of signcrypter (ID_s); and then applies the unsigncryption algorithm verify the signature and decrypt. the ciphertext (ε) to plaintext (m). If the signature cannot be verified, the unsigncrypter rejects the ciphertext (ε).

3.2 Simulation of the Developed HeCA Technique

The flow chart showing the simulation process of the proposed HeCA technique is presented in Figure 4. The simulation was carried out in three stages. The first stage is the application of the Cellular Automata (CA) algorithm to encrypt the plain data (Texts or Image). The second stage is the simulation of the HEC Digital Signature Algorithm (HECDSA), that was used to signcrypt and unsigncrypt the Secret Keys (SK) of the CA. The third stage involves using the unsigncrypted SK to decrypt the cipher data to recover the plain data (Texts or Image).

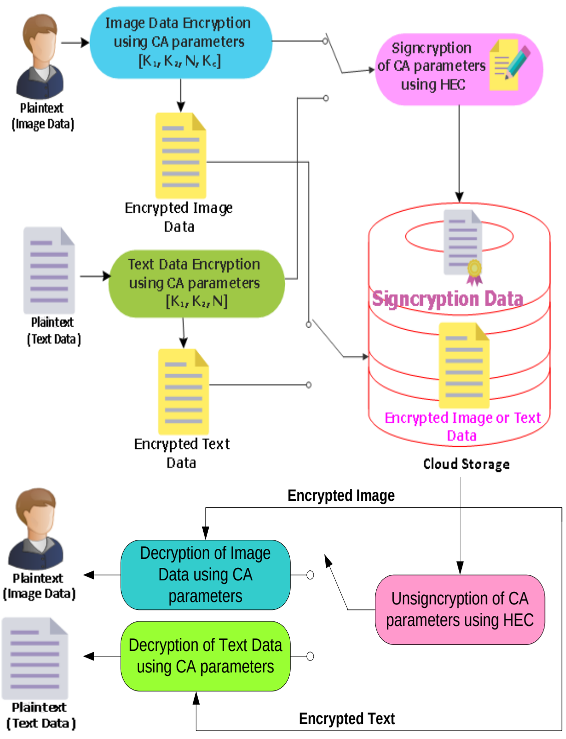

Plaintext (Image Data)
Image Data Encryption using CA parameters [K₁, K₂, N, Kc]
Signcryption of CA parameters using HEC
Encrypted Image Data
Plaintext (Text Data)
Text Data Encryption using CA parameters [K₁, K₂, N]
Signcryption Data
Encrypted Image or Text Data
Encrypted Text Data
Cloud Storage
Encrypted Image
Plaintext (Image Data)
Decryption of Image Data using CA parameters
Unsigncryption of CA parameters using HEC
Plaintext (Text Data)
Decryption of Text Data using CA parameters
Encrypted Text

Figure 3: Block diagram of the HeCA encryption and signcryption implementation

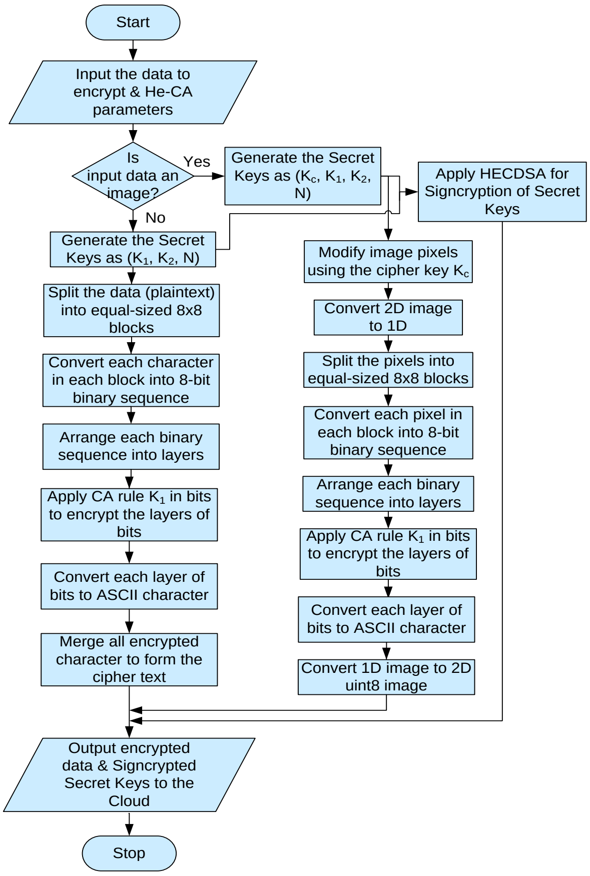

Start
Input the data to encrypt & He-CA parameters
Is input data an image?
Yes
No
Generate the Secret Keys as (Kc, K₁, K₂, N)
Apply HECDSA for Signcryption of Secret Keys
Generate the Secret Keys as (K₁, K₂, N)
Split the data (plaintext) into equal-sized 8x8 blocks
Convert each character in each block into 8-bit binary sequence
Arrange each binary sequence into layers
Apply CA rule K₁ in bits to encrypt the layers of bits
Convert each layer of bits to ASCII character
Merge all encrypted character to form the cipher text
Modify image pixels using the cipher key Kc
Convert 2D image to 1D
Split the pixels into equal-sized 8x8 blocks
Convert each pixel in each block into 8-bit binary sequence
Arrange each binary sequence into layers
Apply CA rule K₁ in bits to encrypt the layers of bits
Convert each layer of bits to ASCII character
Convert 1D image to 2D uint8 image
Output encrypted data & Signcrypted Secret Keys to the Cloud
Stop

Figure 4: Flow chart of the simulation procedure of the devel0ped HeCA technique

4. Results and Discussion

The developed HeCA technique was evaluated by comparing its performance to two existing schemes: SBCAL [17] and CA-RSA [13], using the metrics: Key Length, Processing Time and Throughput.

4.1 Evaluation of the Developed HeCA Technique with Key Length

The comparisons of the developed HeCA technique with the SBCAL and CA-RSA schemes in terms of the total Key Length are presented in Figure 5. The HeCA, SBCAL and CA-RSA gave the total Key Length of 110, 150 and 1084, respectively for the same security level. The comparison revealed that the HeCA technique generated the key with the lowest length of 110 bits, By implication, the HeCA had superiority percentages of 3% and 72.47% over SBCAL and CA-RSA schemes respectively.

The breakdown of the required key length for CA in each of the schemes is shown in Table 2. The HeCA, SBCAL and CA-RSA schemes required 30 bits, 100 bits and 60 bits, respectively. The developed HeCA technique required relatively lower key length compared to the other two schemes because the HeCA scheme utilizes only one CA rule and one-dimensional array to encipher an image whereas, both the SBCAL and CA-RSA schemes utilize either two

CA rules or multi-dimensional array to encipher an image. Reduced complexity is achieved by the HeCA scheme. This is in agreement with the result of [18].

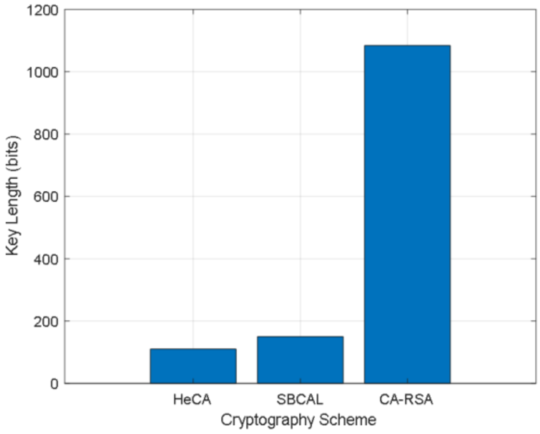


Figure 5: Comparison of HeCA with SBCAL and CA-RSA on Key Length

4.2 Evaluation of the Developed HeCA Technique with Processing Time

The performance of the proposed HeCA technique with regard to processing time was compared with those of the SBCAL and CA-RSA schemes as presented in Figure 6. The total processing time incurred by HeCA, SBCAL and CA-RSA schemes are 4.6622 secs, 7.7550 secs and 6.0620 secs, respectively. The results revealed that the HeCA technique is the most advantageous, as it had the least processing time. The breakdown of the processing times for encryption and decryption of a sample image on the three schemes is presented in Table 3. The results revealed that the encryption time is generally higher than the decryption time irrespective of the scheme. This is due to the key generation process involved in the encryption stage.

Table 2: Key Length Breakdown Comparison of the Three Schemes

Cryptography Scheme	Cellular Automata (CA) key length	Hyper-Elliptic Curve (HEC) key length	S-Box & Lorenz system key length	RSA	Total Key Length
HeCA (proposed)	30 bits	80 bits	Not Applicable	Not Applicable	110 bits
SBCAL	100 bits	Not Applicable	50 bits	Not Applicable	150 bits
CA-RSA	60 bits	Not Applicable	Not Applicable	1024 bits	1084 bits

Table 3: Processing time breakdown comparison of the three schemes for 128 × 128 image dimension (1.2 GHz Intel® Core^{TM} i3, 8 GB)

Cryptography Scheme	Encryption Time (s)	Decryption Time (s)	Total Time (s)
HeCA (proposed)	2.3365	2.3257	4.6622
SBCAL	3.9589	3.7961	7.7550
CA-RSA	3.6815	2.3805	6.0620

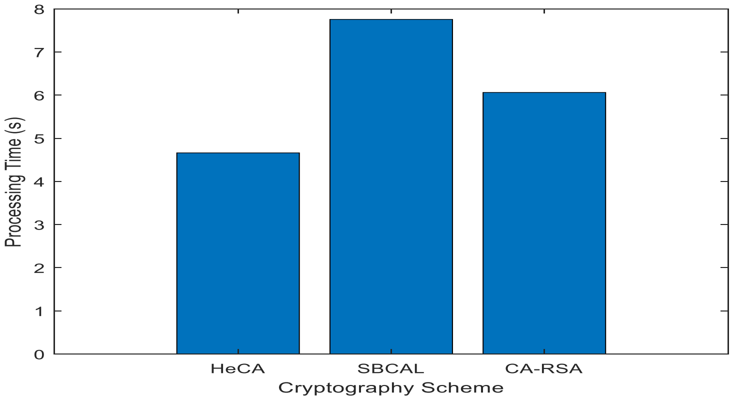


Figure 6: Comparison of HeCA with SBCAL and CA-RSA on Processing Time

4.3 Evaluation of the Developed HeCA Technique with Throughput

The performance of the developed HeCA technique based on throughput was compared with those of the SBCAL and CA-RSA schemes. The HeCA scheme achieved a throughput of 0.561 Mbps, the SBCAL scheme achieved 0.443 Mbps, and the CA-RSA scheme achieved 0.430. The HeCA outperformed the SBCAL and CA-RSA schemes by about 8 % and 9 %, respectively. The results revealed that the HeCA had the highest throughput, which is advantageous for time critical applications as depicted in Table 4 and Figure 7. This result could be traced to the achievement of simplified operations and computations, which is similar to the findings of [2].

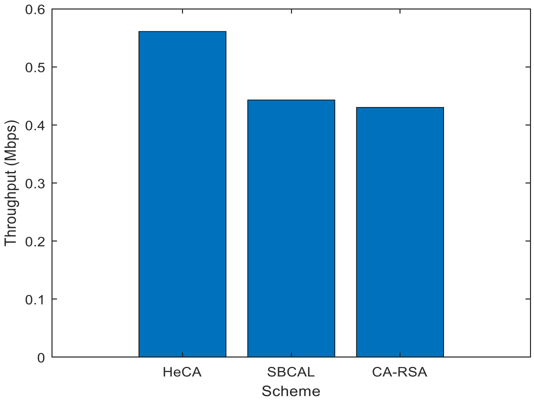


Figure 7: Comparison of HeCA with SBCAL and CA-RSA on Throughput

Table 4: Throughput comparison of the three schemes for 128 × 128 image dimension (1.2 GHz Intel® Core^{TM} i3, 8 GB)

Cryptography Scheme	Throughput (Mbps)
HeCA (proposed)	0.5610
SBCAL	0.443
CA-RSA	0.430

5. Conclusion

This study has developed a novel and more robust cryptosystem suitable for efficient protection of cloud data. The developed technique named HeCA is a hybridization of the Cellular Automata (CA) and the Hyper-Elliptic Curve (HEC) cryptography schemes. The CA aspect is used for the encryption and decryption of the main data, text or image, while the HEC aspect is used to carry out signcryption and unsigncryption of the CA's secret keys to provide additional security to the cloud data. The developed HeCA technique was simulated, and the performance was evaluated on both text and image data. The performance comparison of HeCA with two states of the art schemes (SBCAL and CA-RSA) using the Key Length (in bits), Processing Time (in seconds) and Throughput (in Mbps) as performance metrics showed that the HeCA scheme outperforms the two other schemes for the three metrics considered. The HeCA scheme provides security enhancements via digital signature that is not available in the SBCAL and CA-RSA schemes. Furthermore, the HEC encryption component of the HeCA technique requires only ten HEC divisors for encrypting the CA secret keys as against all the ASCII characters required in the RSA encryption; which gives the HeCA scheme an edge in terms of throughput. The findings from this study has provided a new and useful insight to industry and researchers in the field of cryptography. Further works can consider the use of 2-dimensional array for the CA encryption instead of the 1-dimensional array used for this study.

References

[1] Naskar, P.K.; Bhattacharyya, S.; Nandy, D.; Chaudhuri, A. (2020). A robust image encryption scheme using chaotic tent map and cellular automata. *Nonlinear Dyn.* 2020, 100, 2877–2898. [CrossRef]

[2] Nanda, S.K. Mohanty, S., Pattnaik, P.K. & Sain, M. (2022). Throughput Optimized Reversible Cellular Automata Based Security Algorithm. *Electronics* 2022, 11, 3190.http:// doi.org/10.3390/electronics11193190.

[3] Shabir, M. Y., Iqbal, A., Mahmood, Z. and Ghafoor, A. (2016). Analysis of Classical Encryption Techniques in Cloud Computing. *Tsinghua Science and Technology*, 21(1): 102-113.

[4] Krogstie, J. (2012). Model-Based Development and Evolution of Information Systems: A Quality Approach. Springer London Heidelberg New York Dordrecht, ISBN 978-1-4471-2935-6 ISBN 978-1-4471- 2936-3 (eBook) DOI 10.1007/978-1-4471- 2936-3.

[5] Wan, Z., Liu, J. E. and Deng, R. H. (2012). A hierarchical attribute-based solution for flexible and scalable access control, *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743–754, 2012.

[6] Birje, M. N., Challagidad, P. S. Goudar, R. H. Tapale, M. T. (2017). Cloud computing review: concepts, technology, challenges and security. *Int. J. Cloud Computing,* Vol. 6, No. 1, pp. 32 – 53.

[7] Balamurugan, B. and Krishna, P. V. (2014). Extensive survey on usage of attribute-based encryption in cloud, *Journal of Emerging Technologies in Web Intelligence,* vol. 6, no. 3, pp. 263–272.

[8] Qureshi, M. B., Qureshi, M. S., Tahir, S., Anwar, A., Hussain, S., Uddin, M., and Chen, C. L. (2022). Encryption Techniques for Smart Systems Data Security Offloaded to the Cloud.

Symmetry 2022, 14, 695. https:// doi.org/10.3390/sym14040695.

[9] Abid ur-Rahman, A., Noor-ul-Amin, H., Khattak, I., Ullah, M., Naeem, R. & Anwar, S. U. (2018). A Lightweight Multi-Message and Multi-Receiver Heterogeneous Hybrid Signcryption Scheme based on Hyper EllipticCurve (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 5, pp. 160-167.

[10] Ullah, I., Noor, U. A., Zareei, M., Zeb, A., Khattak, H., Khan, A. and Goudarzi, S. (2019). A Lightweight and Provable Secured Certificateless Signcryption Approach for Crowdsourced IIoT Applications. *Symmetry,* 11, 1386; doi:10.3390/sym11111386.

[11] Chetlapalli, H. (2023). Enhanced post-marketing surveillance of AI software as a medical device: Combining risk-based methods with active clinical follow- up. IMPACT: *International Journal of Research in Engineering & Technology,* 11(6), 1-14.

[12] Beyene, M. and Shekar, K. R. (2019). Performance Analysis of Homomorphic Cryptosystem on Data Security in Cloud Computing. International Conference on Computing, Communication and Networking Technologies (ICCCNT) pp. 1–7. doi: 10.1109/ICCCNT45670.2019.8944837.

[13] Mantri, J. K., Rajalaxmi, M. & Gahan, P. (2019). A Novel encryption Scheme using Hybrid Cellular Automata. In proceeding of 2019 International Conference on Information Technology (ICIT), pp. 382-387.

[14] Awan, I. A. et al. (2020) Secure Framework Enhancing AES Algorithm in Cloud Computing. Security & Communication Networks, pp. 1–16. doi: 10.1155/2020/8863345.

[15] Kumar, A. & Raghava, N.S. (2021). An efficient image encryption scheme using elementary cellular automata with novel permutation box. Multimed. Tools Appl. 2021, 80, 21727–21750. [CrossRef].

[16] Venugopal, M., Rajan, E, G., Raman, V. (2022). Visual Cryptography Using Neighbourhood Based Encryption Techniques in The Framework of Cellular Automata. *J Math Techniqie*, 1(2), 133-137.

[17] Alexan, W., ElBeltagy, M. and Aboshousha, A. (2022). RGB Image Encryption through Cellular Automata, S-Box and the Lorenz System. *Symmetry* 2022, 14(3), 443. https://doi.org/10.3390/sym14030443.

[18] Ike, E. J., Kessie, J. D., Popoola, R., Azeez, M.A. & Onibokun, T. (2024). A Novel Approach to Cloud Data Encryption using Homomorphic Encryption. *Journal of Frontiers in Multidisciplinary Research*, 5(6), pp. 9-