Data and Metadata. 2025; 4:743
doi: 10.56294/dm2025743

ORIGINAL

Novel Key Generator-Based SqueezeNet Model and Hyperchaotic Map

Nuevo modelo de red de compresión basado en generador de claves y mapa hipercaótico

Hayder Najm[1], Mohammed Salih Mahdi[2], Sanaa Mohsin[2]

[1]Department of Computer Techniques Engineering, Imam Alkadhim University College. Baghdad, Iraq.
[2]Business Informatics College, University of Information Technology and Communications. Baghdad, Iraq.

Cite as: Najm H, Salih Mahdi M, Mohsin S. Novel Key Generator-Based SqueezeNet Model and Hyperchaotic Map. Data and Metadata. 2025; 4:743. https://doi.org/10.56294/dm2025743

Submitted: 29-04-2024 **Revised:** 26-08-2024 **Accepted:** 22-03-2025 **Published:** 23-03-2025

Editor: Dr. Adrián Alejandro Vitón Castillo

Corresponding Author: Hayder Najm

ABSTRACT

Cybersecurity threats are evolving at a very high rate, thus requiring the use of new methods to enhance the encryption of data and the communication process. In this paper, we propose a new key generation algorithm using the simultaneous use of the SqueezeNet deep learning model and hyperchaotic map to improve the hallmark of cryptographic security. The method employed in the proposed approach is built around the SqueezeNet model, which is lighter and faster in extracting features from the input image, and a hyperchaotic map, which is the main source of dynamic and non-trivial keys. The hyperchaotic map enhances complexity and randomness, securing the new cryptosystem against brute force and statistical attacks, and the key length depends on the number of features in the image. All our experiments prove that the proposed key generator works well in generating long, random, high entropy keys and is highly resistant to all typical cryptographic attacks. The promising profound synergy of deep learning and chaotic systems provides directions for the development of secure and effective methods of cryptography amid the exacerbated cyber threats. The technique was found to meet all the 15 criteria as tested through the NIST statistical test suite.

Keywords: Data Security; Cybersecurity; Deep Learning; Transfer Learning; SqueezeNet Model; Hyperchaotic Map.

RESUMEN

Las amenazas a la ciberseguridad están evolucionando a un ritmo muy rápido, por lo que requieren el uso de nuevos métodos para mejorar el cifrado de datos y el proceso de comunicación. En este artículo, proponemos un nuevo algoritmo de generación de claves que utiliza el uso simultáneo del modelo de aprendizaje profundo SqueezeNet y el mapa hipercaótico para mejorar el sello distintivo de la seguridad criptográfica. El método empleado en el enfoque propuesto se basa en el modelo SqueezeNet, que es más ligero y rápido en la extracción de características de la imagen de entrada, y un mapa hipercaótico, que es la principal fuente de claves dinámicas y no triviales. El mapa hipercaótico mejora la complejidad y la aleatoriedad, asegurando el nuevo criptosistema contra ataques de fuerza bruta y estadísticos, y la longitud de la clave depende del número de características en la imagen. Todos nuestros experimentos demuestran que el generador de claves propuesto funciona bien para generar claves largas, aleatorias y de alta entropía y es altamente resistente a todos los ataques criptográficos típicos. La prometedora sinergia profunda del aprendizaje profundo y los sistemas caóticos proporciona direcciones para el desarrollo de métodos seguros y efectivos de criptografía en medio de las crecientes amenazas cibernéticas. Se determinó que la técnica cumplía con los 15 criterios evaluados mediante el conjunto de pruebas estadísticas del NIST.

Palabras clave: Seguridad de Datos; Ciberseguridad; Aprendizaje Profundo; Aprendizaje por Transferencia; Modelo SqueezeNet; Mapa Hipercaótico.

INTRODUCTION

Data encryption is one of the most crucial sectors in information and communication technology, which prevents data alteration and safeguards against security breaches. Cryptography has been introduced due to the complexity of technology, the value attached to information security, and the concern about users' privacy.[1] Furthermore, the integrity of cryptographic keys is directly proportional to the overall security of the encryption system. Reliable key generation can be a concern because insecure key generation methods may emerge, making the encrypted data susceptible to attacks from adversaries, thus threatening its confidentiality and integrity.[2] Therefore, it is crucial to establish specific best practices for generating keys to ensure that, at its core, encryption mechanisms for protecting secret data are applicable in various applications.[3]

Deep learning has recently been widely applied to classification and recognition tasks. The SqueezeNet model, particularly when trained on large-scale image datasets such as ImageNet, has proven highly effective at extracting detailed and distinctive features from images.[4] Hyperchaotic maps are mathematical structures chosen because they are sensitive to initial conditions, which makes them perfect for cryptography.[5] Therefore, these systems can generate arbitrary sequences useful in key generation operations. Hyperchaos has the advantage of producing difficult-to-predict and analyze keys for encryption algorithms using the dynamic behavior of hyperchaotic systems to create forms of protection against brute-force attacks.[6]

This work offers a new way by which the SqueezeNet model architecture can be integrated with hyperchaotic maps for key generation in another way that sets a new angle in relating deep learning to chaos theory. The SqueezeNet model can be trained to take various inputs as feature maps to modify the parameters of the hyperchaotic map. This means that the key generation process is highly flexible in matching it to the kind of data that needs to be protected, the degree of encryption, and how possible attackers can be disoriented. This paper highlights the following key contributions:

1. Introduce a new approach based on the SequeezeNet model and Hyperchaotic map where long keys can be generated for large inputs, and hence, the number of keys required can be minimized.
2. It is possible to see that the key length that characterizes our algorithm depends on the number of features in the input image.
3. The suggested method enables a flexible mechanism for generating keys without being bound to a specified image type.
4. Appending feature extraction to the key generation process increases the randomness and unpredictability of the resultant encryption key.
5. When very lengthy keys are produced, the attack based on brute force is effectively countered.

The rest of this paper is structured as follows: Section 2 discusses related works. Section 3 explains symmetric encryption. Section 4 explains transfer learning. The existence of a hyperchaotic map schema is the topic of Section 5. In section 6, the proposed methodology is presented. Sections 7 and 8 compare the key length, Shannon entropy, and randomness tests based on specific criteria in the standard namespace defined by NIST. Finally, Section 9 presents a conclusion.

Literature Review

Very few papers consider the generation of keys. Sadim et al.[7] present a combined model of the neural synchronization Blowfish algorithm to generate secret keys exchangeable on public channels. This approach is far more robust than traditional encryption practices like the Advanced Encryption Standard and Blowfish but provides better data throughput rates.[8] Significant results illustrate how the proposed method integrates encryption and decryption runtimes with synchronization. By employing a novel mutual learning strategy, the key lies in encrypting the data with neural networks that search for shared secret keys. The study concludes that neural techniques should be integrated into cryptographic systems to enhance performance and security.

Sohel et al. address the pressing issue of security and performance in resource-limited devices.[8] To tackle this, they propose a lightweight cryptographic algorithm based on neural networks (NN-cipher), which offers lower key generation cycles and reduced power consumption. As demonstrated by bridge histogram evaluations, the encryption technique employed show's reliability in encrypting images. Key sensitivity tests confirm that decryption requires the correct key, ensuring a strong security measure.[9] Amina et al. emphasize the necessity for new cryptographic techniques due to the rising number of attacks on existing systems. The paper introduces chaotic systems and deep learning methods by specifically training an artificial neural network (ANN) using chaotic time series forecasting models to generate encryption keys. Comparative tests involving MLP, LSTM, and GRU models yielded nearly identical results in chaotic time series prediction. The optimal ANN design achieved a Mean Squared Error (MSE) of $3{,}2 \times 10^{-3}$. The results demonstrate effective and comparable performance among the tested models, validating the approach for cryptographic applications.

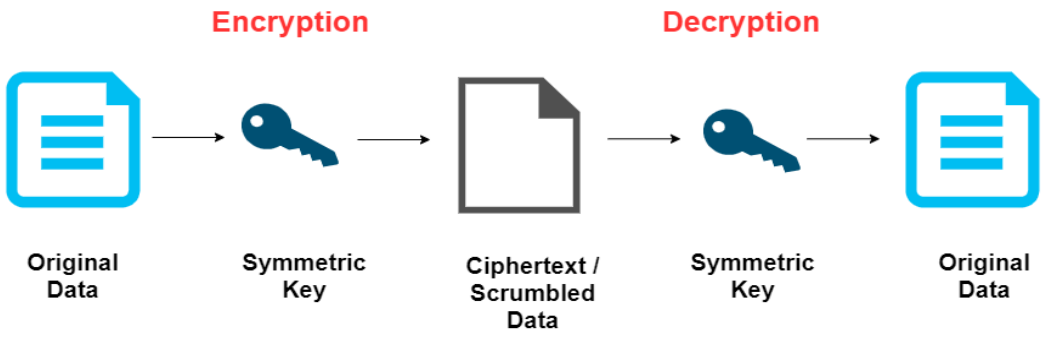
Youcef et al. address the necessity for efficient pseudo-random number generators (PRNGs) in cryptographic applications.[10] Their proposed solution is an innovative hybrid PRNG design that integrates artificial neural

networks (ANNs) with chaos theory, utilizing the classical Lorenz chaotic system. The techniques include training the ANN with numerically resolved datasets and assessing them with the mean square error (MSE) metric. The results show significant efficiency improvements, achieving an MSE of $2{,}3751\times10^{-7}$ and expanding the key space by as much as threefold. The PRNG sequences successfully passed the NIST SP 800-22 tests, confirming their suitability for future cryptographic application algorithms.

Symmetric Encryption

It is important to note that the key used for encryption is the same as that used for decryption. Symmetric algorithms often have fast and effective applications in communications and banking.[11] A current example of a symmetric technique is the Triple Data Encryption Standard (Triple-DES) method. However, Triple-DES is an outdated approach gradually replaced by the more advanced Advanced Encryption Standard (AES). AES employs larger key and encryption block sizes, making it more robust than Triple-DES.[12]

Namely, in symmetrical encryption, the data is split into blocks, and these blocks operate on the cipher system to produce ciphertext; data is encrypted and decrypted. It is fundamentally for the ability to process only small data chunks that symmetric encryption methods are said to perform well.[13] One distinctive attribute of symmetrical key algorithms is their efficiency or capability of encrypting massive data streams. However, the primary disadvantage of using symmetric key encryption is determining the most secure means of exchanging the encryption key among the other parties in the communication channel. This means that if anyone possessing the key intercepts the message in the form of ciphertext, then they can easily convert it into plaintext. Thus, to ensure the anonymity and privacy of respondents, the key must remain confidential. As shown in figure 1 below, encryption and decryption involve using the same key, whereas in text decryption, the key is used to reverse the data, making it readable.[14]



Source: Oladoyimbo et al.[15]

Figure 1. Symmetric key for encryption and decryption processes

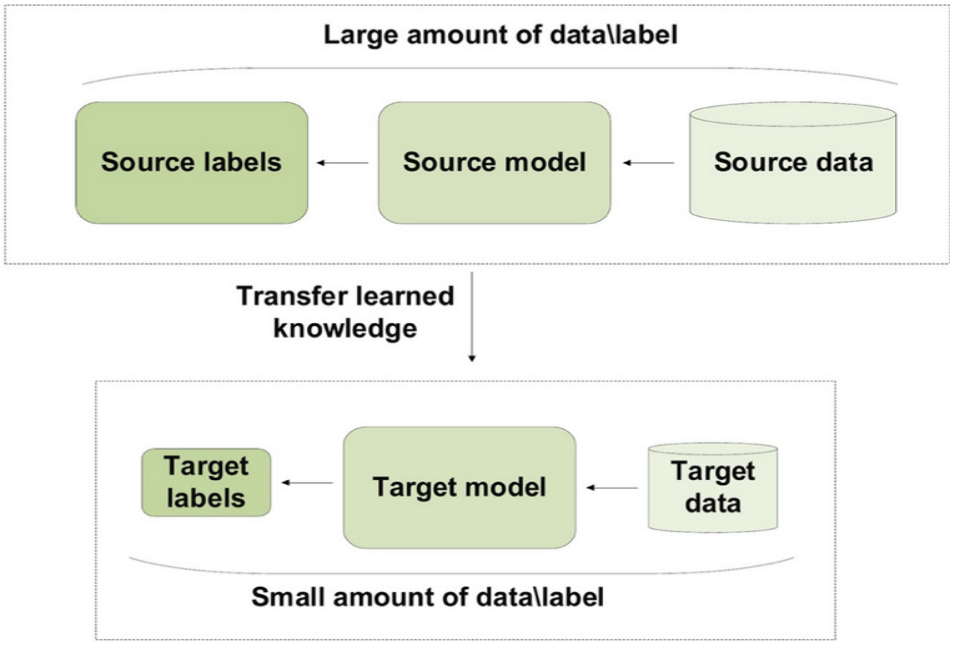
Transfer Learning

Transfer learning is a specific model designed for one role to be utilized in a related capacity within another organization. Solving one issue can assist another network in understanding the same challenge, promoting quicker advancement and improving effectiveness in addressing the second concern.[16,17]

Transfer learning proves highly advantageous when limited samples are available for training a model. Rather than starting from scratch, the model can utilize pre-trained weights. It aims to transfer the acquired knowledge from the source to the target, relaxing the assumption of independence and identical training data distribution concerning the test samples. This strategy can significantly improve challenging domains due to insufficient training data.[18]

Deep neural networks can be trained with less data than traditional deep learning methods using transfer learning. You require large data, computational capability, and model training time. A pre-trained model acts as a starting point to cut down on all three: This isn't raw data for which developers are training a large model that is training another large model on even bigger data. The beauty of this is that if the second task is or should be related to the first (or, equivalently, if there is less data for the second task), there is no need to model the relevance of the second task to the first. The learning of features can be made faster, and prospective performance improves with the initial task, where features learned for the initial task are used for the model to perform the next task. It also decreases the chances of overfitting since the simple fact that the model possesses features of the first task will generalize the second task.[19]

Figure 2 depicts the conceptual diagram of the transfer learning approach. TL is a strategy that employs representations of knowledge generated by a diverse range of equally usable tasks. We have also noted that improvement can be achieved if the two tasks are similar.[20]

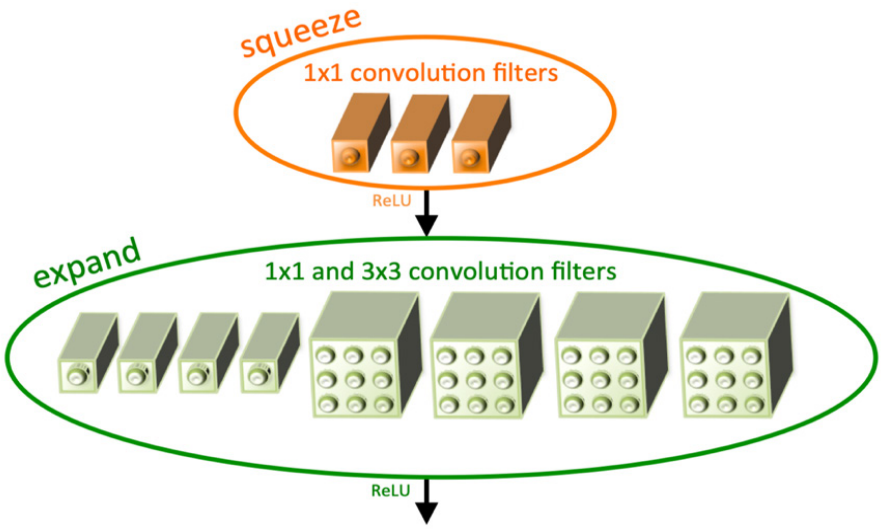


Source: Alzubaidi et al.[21]
Figure 2. Transfer learning concept

Feature Extraction using SqueezeNet Architecture

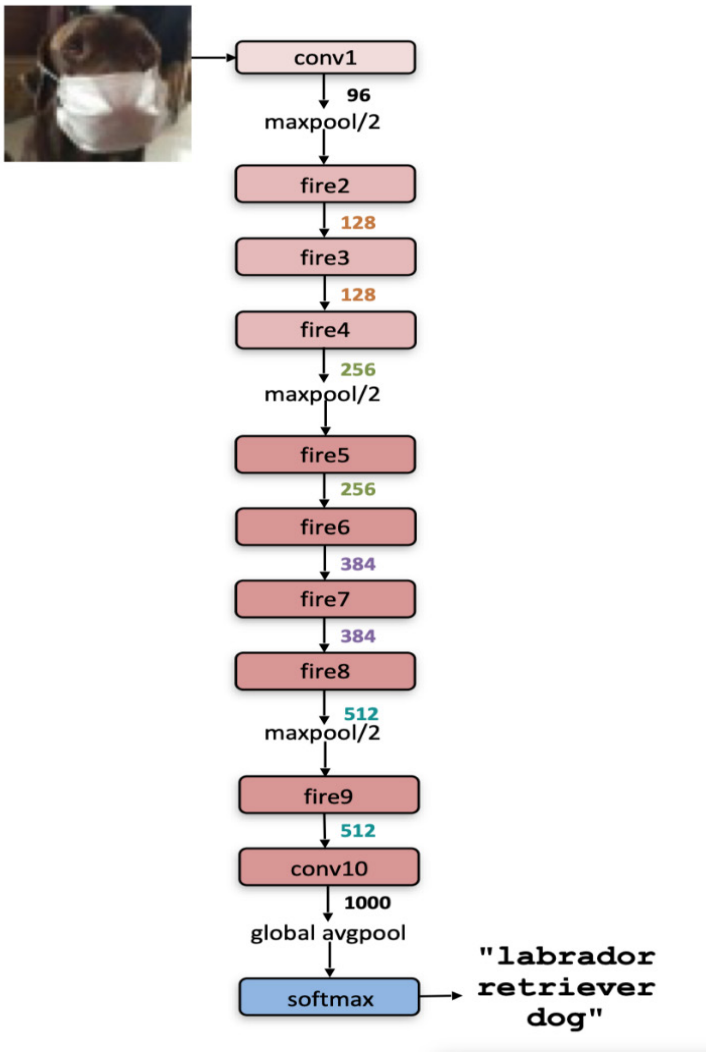
SqueezeNet aims to reduce the parameter count by 50 times or even more while achieving an AlexNet-level top-5 error rate to support use in FPGAs and other resource-scarce systems. To accomplish this, three key design strategies are employed: transforming 3x3 convolutional filters into 1x1 filters, which require nine times fewer parameters; reducing the number of inputs to the 3x3 filters through the introduction of squeeze layers; and performing downsampling at a later stage of the network to maintain the high spatial resolution of features in the early layers, thereby enhancing the accuracy of the classifiers. All these strategies ensure a compact, efficient model for companies and organizations without compromising output.[22,23]

SqueezeNet is built using a module known as the fire module. The purpose of the fire module is to make the model more parameter-efficient than other models.[24,25] It consists of two primary components: a squeeze layer with 1x1 filters to reduce the number of input channels and an expand layer combining 1x1 and 3x3 filters to regain the complexity necessary for feature extraction tasks, as shown in figure 3. This structure allows the Fire Module to strike the right balance between parameter reduction and computational efficiency, as the squeeze layer limits the dimensions of the input to the expanding layer and, consequently, reduces the total number of parameters.[26] Therefore, the fire module supports a more flexible yet slim design for reliable and efficient convolutional neural networks, featuring selectively tuned squeeze and expand layers architectures.[27]



Source: Iandola Forrest N et al.[22]
Figure 3. Organization of fire module

SqueezeNet begins with an initial convolution layer (conv1), followed by eight fire layers (from fire2 to fire9), and then a final convolution layer (conv10), as shown in figure 4. The number of filters in each fire module steadily increases from the initial to the final stage of the network. Max-pooling is performed every two layers after conv1, fire4, fire8, and conv10. The known implementation of pooling occurs later in this structure than in others.[28]



Source: Iandola Forrest N et al.[22]
Figure 4. SqueezeNet architecture

The SqueezeNet architecture incorporates several design choices and optimizations to enhance its performance and compactness. To ensure that:[29,30]

- In the expand layers, a 1-pixel zero-padding border is added to the input of 3x3 filters to ensure that the output activations from 1x1 and 3x3 filters have the same dimensions.
- ReLU (Rectified Linear Unit) activations are applied to the outputs from both the squeeze and expand layers, which introduces non-linearity.
- A dropout layer with a 50 % ratio is applied after the final Fire Module (fire9) to reduce overfitting and enhance performance generalization.
- The architecture avoids fully connected layers, drawing inspiration from the Network-in-Network (NiN) design to reduce the parameter count and enhance efficiency compactness.
- Training starts with a learning rate of 0,04, which decreases linearly over time training.
- Due to the framework's limitations, the expand layer is implemented as two distinct convolutional layers—one for 1x1 filters and another for 3x3 filters—and their outputs are concatenated along the channel dimension.
- Originally developed with the Caffe framework, SqueezeNet has been adapted for platforms like MXNet, Chainer, Keras, and Torch to ensure wider compatibility.

layer name/type	output size	filter size / stride (if not a fire layer)	depth	s_{1x1} (#1x1 squeeze)	e_{1x1} (#1x1 expand)	e_{3x3} (#3x3 expand)	s_{1x1} sparsity	e_{1x1} sparsity	e_{3x3} sparsity	# bits	#parameter before pruning	#parameter after pruning
input image	224x224x3										-	-
conv1	111x111x96	7x7/2 (x96)	1				100% (7x7)			6bit	14,208	14,208
maxpool1	55x55x96	3x3/2	0									
fire2	55x55x128		2	16	64	64	100%	100%	**33%**	6bit	11,920	5,746
fire3	55x55x128		2	16	64	64	100%	100%	**33%**	6bit	12,432	6,258
fire4	55x55x256		2	32	128	128	100%	100%	**33%**	6bit	45,344	20,646
maxpool4	27x27x256	3x3/2	0									
fire5	27x27x256		2	32	128	128	100%	100%	**33%**	6bit	49,440	24,742
fire6	27x27x384		2	48	192	192	100%	**50%**	**33%**	6bit	104,880	44,700
fire7	27x27x384		2	48	192	192	**50%**	100%	**33%**	6bit	111,024	46,236
fire8	27x27x512		2	64	256	256	100%	**50%**	**33%**	6bit	188,992	77,581
maxpool8	13x12x512	3x3/2	0									
fire9	13x13x512		2	64	256	256	**50%**	100%	**30%**	6bit	197,184	77,581
conv10	13x13x1000	1x1/1 (x1000)	1				**20%** (3x3)			6bit	513,000	103,400
avgpool10	1x1x1000	13x13/1	0									
	activations		parameters				compression info				1,248,424 (total)	**421,098** (total)

Source: Iandola Forrest N et al.[22]

Figure 5. Layers of SqueezeNet architecture

Hyperchaotic Map Schema

The hyperchaotic map is an advanced type of chaotic system characterized as a high-dimensional, complex system highly sensitive to initial conditions across multiple dimensions.[31] Most conventional chaotic systems have only one positive Lyapunov exponent, which suggests exponential separation in a single direction.[32] In contrast, hyperchaotic systems exhibit two or more positive Lyapunov exponents, resulting in greater uncertainty and complexity at the expense of simplicity. Hyperchaotic maps generally rely on nonlinear mathematical equations and typically occur in systems with at least four dimensions, whether discrete (difference equations) or continuous (differential equations).[33]

The system's behavior is nonlinear and unpredictable yet appears chaotic due to a minute change or close dependence on initial conditions. A specific starting parameter set creates a distinct shift in the system's phase space. Hyperchaotic maps typically have strange attractors associated with highly non-repeating, bounded movement patterns that reflect the geometry of the actual system.[34,35,36,37,38,39,40,41]

METHOD

The proposed methodology outlines a method of developing a cryptographic key by combining the feature extraction of a pre-trained convolutional neural network and a chaotically amplified map from a hyperchaotic system. Where the SqueezeNet model is imported from the Torchvision. For the models library, the next step is to convert the SqueezeNet into a PyTorch model. Being a light and efficient convolutional neural network pre-trained on the ImageNet dataset, SqueezeNet directly extracts useful features from images without needing further training while providing high-quality feature representation. The selection of SqueezeNet has been appropriate because of its low computational requirements while offering good feature extraction.

In the preprocessing step, the image goes through image enhancement to determine if it is fit for feature extraction. The image is then loaded from the image file path and resized to 224×224 pixels, which SqueezeNet accepts. It resizes them to be constant in size for input and to have a proper size to be fed into the model. The resized image is then transformed by subtracting the ImageNet mean and dividing the result by its standard deviation to minimize pixel intensity values, which form an appropriate input for networks. Further, the image is converted into tensor form so that the model can accept its structure, and a batch dimension is appended.

The preprocessed image was then fed into SqueezeNet to yield a high-dimensional feature map with detailed features of the image, including textural, edge, and pattern aspects. These feature maps are passed through a flattening layer, which converts their structure into a one-dimensional format. This flattened array composed

of the compressed image becomes the source from which the cryptographic is produced, thus generating the key tied or keyed to the actual image input.

The next step converts the feature map to a binary cryptographic key using a hyperchaotic map, which was identified to have a sensitive dependence on initial conditions and high dimensionality of chaos. At the start, they are excluded from the flattened feature map to avoid numerical issues in the presence of zero values. The rest of the values are split into four portions; every portion passes through the same hyperchaotic equation. The hyperchaotic map is constructed from four initial seed values (x_0, y_0, z_0, w_0) that control the dynamic behavior of the map. Each chunk is updated through a series of mathematical transformations governed by hyperchaotic equations, such as:

$$X_n = a \cdot (Y_n - X_n) + W_n \quad (1)$$
$$Y_n = b \cdot X_n - X_n \cdot Z_n \quad (2)$$
$$Z_n = c + X_n \cdot Y_n - d \cdot Z_n \quad (3)$$
$$W_n = d \cdot (X_n + Y_n + Z_n) \quad (4)$$

The system's regulating parameters are the quantities a, b, c, and d. However, to increase randomness, the values are updated using the modulo operation so that the maximum limit of the values is not exceeded.

Each updated value is then converted into binary form, thus enhancing the high entropy and randomness of the key. The obtained binary values from all chunks are then joined in a serial binary form that combines the image's specific structure with the hyperchaotic map's dynamic properties. The last generated string, suggested_binary_key, is a secure and unique binary string that may be used for cryptographic purposes. This key generation methodology ensures that the generated key is strongly based on the input image and further enhances the unpredictability with high entropy from the hyperchaotic system, making the system highly cryptographic and depending on the input image. Figure 6 elucidate the processing of the extracted features to produce the proposed key.

Algorithm 1. Novel key Generator based SqueezeNet model and Hyperchaotic map.

Input: An image as the seed

Output: suggested_binary_key

Step 1: Load SqueezeNet model.

Step 1.1: Load the pre-trained SqueezeNet model from torchvision. models.

Step 2: Preprocess input image.

Step 2.1: Read the image from the specified file path.

Step 2.2: Image resize with a fixed size (224 x 224).

Step 2.3: Standardize the image according to the standard ImageNet values.

Step 2.4: Add a batch dimension to the image, then put that image in a tensor.

Step 3: SqueezeNet extract features.

Step 3.1: Preprocess the image tensor and run it through SqueezeNet_model to get SqueezeNet_features_maps.

Step 4: Hyperchaotic map for convert SqueezeNet_features_maps to binary key.

Step 4.1: Remove zero values from flattening the SqueezeNet_features_maps.

Step 4.2: Split the SqueezeNet feature map into a chunk of 4 values each.

- Initialize the hyperchaotic map with 4 values.
- Update the chunk using the hyperchaotic map equations in section 6.
- Update the values and apply the modulo operation on these values.
- Update those values in binary format.

Step 4.3: Concatenating all of the binary chunks together to form suggested_binary_key.

Figure 6. Algorithm of the extracted features to produce the proposed key

https://doi.org/10.56294/dm2025743

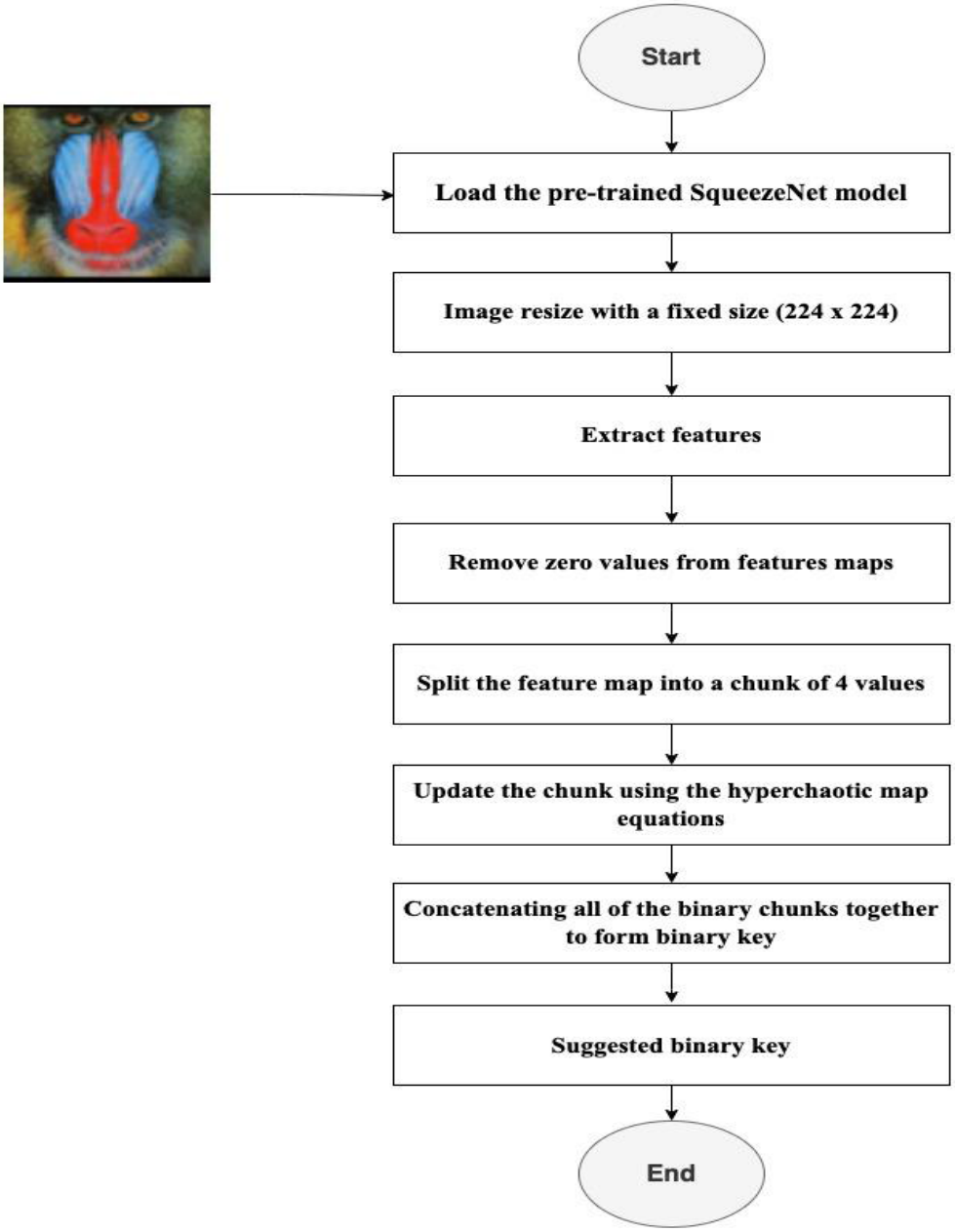


Figure 7. Flowchart of proposed novel key generation using SqueezeNet model and hyperchaotic map

Key Length and Shannon Entropy

The key length is dynamic since it depends on the number of features to be extracted concerning the image through the SqueezeNet. Based on the complexity and content of the input image and the layers in SqueezeNet, these features differ depending on the count of extracted features needed for analysis.

The SqueezeNet model extracts fewer features from images with fewer distinct patterns or small details, so the key length is shorter for shorter keys. However, rich-textured and complex images containing many colors and structures yield many features and larger key lengths. This variation results from SqueezeNet's ability to selectively encode different spatial and semantic structure levels according to the input image.

Another factor modifying the key length is Shannon entropy, a quantitative characteristic of a dataset's unpredictable (information) content. Shannon entropy of images under consideration was higher, meaning the images contain more diversified and complicated information, and therefore, more features are extracted from the image data. For this reason, these images need longer keys to cover the variability range correctly and cover all the existing photos. On the other hand, low entropy images with un-variant or near periodicity figures resulted in fewer features and relatively shorter keys.

This relationship is described in detail using the key length differences and Shannon entropy in table 1, which summarizes the SqueezeNet model and key lengths for a range of images. It demonstrates how image contents and feature density alterations substantially impact key lengths. Understanding this variability is important if

reliability is needed in applications based on the efficient representation of images, such as encryption, image compression, or image-based identification and authentication, where the key length or code size versus the computation time is a critical factor.

Table 1. Key length and entropy

Images	Key Length	Shannon Entropy
Baboon	67,680 bits	0,99999872
Peppers	64,416 bits	0,99999127
Cat	62,272 bits	0,99998799
Dog	56,800 bits	0,99998733

NIST Randomness Test

The NIST Statistical Test Suite is an important tool for testing whether a binary sequence is random regarding cryptographic key generation. Therefore, enhancing feature extraction from images delivered by SqueezeNet, a deep learning model, and hyperchaotic maps provides an innovative method of key formation. SqueezeNet can offer significant features from imagery while elucidating digital imagery into numerical input. When fed through a hyperchaotic map, these features perform different sequences whose nature is highly complex and unpredictable. As shown in table 2, the generated keys are very random and difficult for the adversaries to guess as they fail to predict them.

Table 2. NIST test

Test	P - Value
Monobit	0,7264
Frequency	0,6707
Runs	0,8209
Longest Run	0,5748
Binary Matrix Rank	0,7268
DFT	0,9890
Non Overlapping Template Matching	0,5768
Overlapping Template Matching	0,9467
Maurer's Universal	0,7921
Linear Complexity	0,6324
Serial	0,5245
Approximate Entropy	0,6777
Cumulative Sums	0,8900
Random Excursion	0,7920
Random Excursion Variant	0.6327

DISCUSSION

The proposed novel key generator, a combination of the SqueezeNet model and hyperchaotic map, provides an effective and efficient design for the actual key generation process. Squeezing the architecture of SqueezeNet makes the system compact enough to respond to resource scarcity; the choice of a hyperchaotic map protects the system because of its high sensitivity to initial conditions and genuine unpredictability. This leads to a key generation mechanism with high entropy and, simultaneously, highly resistant to cryptographic attacks. Moreover, experimental assessments validate the method's effectiveness in generating safe keys with little computational expenses compared to existing techniques where the process can be effectively applied in many areas, such as IoT devices, secure communication systems, etc. Cryptography can benefit from advancements in mathematical computational algorithms featuring deep learning and chaotic systems, as presented in this work toward fulfilling emerging digital challenges.

CONCLUSIONS

The research introduces an innovative key generation technique integrating SqueezeNet feature extraction with a hyperchaotic map to enhance cryptographic security. This method generates high-entropy, dynamic

keys that are robust against brute-force and statistical assaults, successfully meeting all 15 NIST randomisation standards. The customisable key length renders it appropriate for secure communications, especially in resource-limited settings such as IoT. The paper highlights the possibility of integrating deep learning with chaos theory in cryptography, proposing that future research may explore its use in diverse lightweight scenarios.

BIBLIOGRAPHIC REFERENCES

1. J. Ye, R. Zhang, M. Zhong, and Z. Zhang, "Design of Data Encryption and Compression Methods," Procedia Comput. Sci., vol. 243, pp. 1257–1264, 2024, doi: 10.1016/j.procs.2024.09.148.

2. S. E. Vadakkethil Somanathan Pillai and K. Polimetla, "Analyzing the Impact of Quantum Cryptography on Network Security," in 2024 International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India: IEEE, Feb. 2024, pp. 1–6. doi: 10.1109/ICICACS60521.2024.10498417.

3. H. Najm, M. S. Mahdi, and W. R. Abdulhussien, "Lightweight Image Encryption Using Chacha20 and Serpent Algorithm," J. Internet Serv. Inf. Secur., vol. 14, no. 4, pp. 436–449, Nov. 2024, doi: 10.58346/JISIS.2024.I4.027.

4. M. F. Siddique, Z. Ahmad, N. Ullah, S. Ullah, and J.-M. Kim, "Pipeline Leak Detection: A Comprehensive Deep Learning Model Using CWT Image Analysis and an Optimized DBN-GA-LSSVM Framework," Sensors, vol. 24, no. 12, p. 4009, Jun. 2024, doi: 10.3390/s24124009.

5. M. S. Al-Batah, M. S. Alzboon, M. Alzyoud, and N. Al-Shanableh, "Enhancing Image Cryptography Performance with Block Left Rotation Operations," Appl. Comput. Intell. Soft Comput., vol. 2024, no. 1, p. 3641927, Jan. 2024, doi: 10.1155/2024/3641927.

6. O. Kuznetsov, N. Poluyanenko, E. Frontoni, and S. Kandiy, "Enhancing Smart Communication Security: A Novel Cost Function for Efficient S-Box Generation in Symmetric Key Cryptography," Cryptography, vol. 8, no. 2, p. 17, Apr. 2024, doi: 10.3390/cryptography8020017.

7. Mohd. Sadim, N. Pratap, S. Kumar, and A. Latoria, "WITHDRAWN: Hybrid neural synchronization blowfish algorithm for secret key exchange over public channels," Mater. Today Proc., p. S2214785320389811, Jan. 2021, doi: 10.1016/j.matpr.2020.11.363.

8. S. Rana, M. R. H. Mondal, and A. H. M. S. Parvez, "A New Key Generation Technique based on Neural Networks for Lightweight Block Ciphers," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, 2021, doi: 10.14569/IJACSA.2021.0120623.

9. A. Kadir, M. S. Azzaz, and R. Kaibou, "Chaos-based Key Generator using Artificial Neural Networks Models," in 2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS), BLIDA, Algeria: IEEE, Mar. 2023, pp. 1–5. doi: 10.1109/ICAECCS56710.2023.10105105.

10. Y. Alloun, M. S. Azzaz, A. Kifouche, and R. Kaibou, "Pseudo Random Number Generator Based on Chaos Theory and Artificial Neural Networks," in 2022 2nd International Conference on Advanced Electrical Engineering (ICAEE), Constantine, Algeria: IEEE, Oct. 2022, pp. 1–6. doi: 10.1109/ICAEE53772.2022.9962090.

11. H. Najm, H. K. Hoomod, and R. Hassan, "A New WoT Cryptography Algorithm Based on GOST and Novel 5d Chaotic System," Int. J. Interact. Mob. Technol. IJIM, vol. 15, no. 02, p. 184, Jan. 2021, doi: 10.3991/ijim.v15i02.19961.

12. H. Najm, H. K. Hoomod, and R. Hassan, "A proposed hybrid cryptography algorithm based on GOST and salsa (20)," Period. Eng. Nat. Sci. PEN, vol. 8, no. 3, pp. 1829–1835, 2020. doi:10.21533/PEN.V8I3.1619

13. Raghad Abdulaali Azeez, Abeer Salim Jamil, and Mohammed Salih Mahdi, "A Partial Face Encryption in Real World Experiences Based on Features Extraction from Edge Detection," Int. J. Interact. Mob. Technol. IJIM, vol. 17, no. 07, pp. 69–81, Apr. 2023, doi: 10.3991/ijim.v17i07.38753.

14. M. S. Mahdi, N. F. Hassan, and G. H. Abdul-Majeed, "An improved chacha algorithm for securing data on IoT devices," SN Appl. Sci., vol. 3, no. 4, p. 429, Apr. 2021, doi: 10.1007/s42452-021-04425-7.

15. T. O. Oladoyinbo, O. B. Oladoyinbo, and A. I. Akinkunmi, "The Importance Of Data Encryption Algorithm In Data Security". http://doi.org/10.36893/JNAO.2022.V13I02.001-011

16. A. F. Majeed, P. Salehpour, L. Farzinvash, and S. Pashazadeh, “Multi-Class Brain Lesion Classification Using Deep Transfer Learning With MobileNetV3,” IEEE Access, vol. 12, pp. 155295–155308, 2024, doi: 10.1109/ACCESS.2024.3413008.

17. N. F. Hassan, A . Al-Adhami and M. S. Mahdi, “Digital Speech Files Encryption based on Hénon and Gingerbread Chaotic Maps” Iraqi Journal of Science (2022): 830-842, doi: https://doi.org/10.24996/ijs.2022.63.2.36 .

18. M. A. Taha, S. A. A. A. Alsaidi, and R. A. Hussein, “Machine Learning Techniques for Predicting Heart Diseases,” in 2022 International Symposium on iNnovative Informatics of Biskra (ISNIB), Biskra, Algeria: IEEE, Dec. 2022, pp. 1–6. doi: 10.1109/ISNIB57382.2022.10076238.

19. H. M. Al-Dabbas and Mohammed Salih, “Classification of Brain Tumor Diseases Using Data Augmentation and Transfer Learning,” Iraqi J. Sci., pp. 2275–2286, Apr. 2024, doi: 10.24996/ijs.2024.65.4.41.

20. M. Laurer, W. Van Atteveldt, A. Casas, and K. Welbers, “Less Annotating, More Classifying: Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT-NLI,” Polit. Anal., vol. 32, no. 1, pp. 84–100, Jan. 2024, doi: 10.1017/pan.2023.20.

21. L. Alzubaidi et al., “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” J. Big Data, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

22. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” Nov. 04, 2016, arXiv: arXiv:1602.07360. doi: 10.48550/arXiv.1602.07360.

23. M. Fatima et al., “Two-Stage Intelligent DarkNet-SqueezeNet Architecture-Based Framework for Multiclass Rice Grain Variety Identification,” Comput. Intell. Neurosci., vol. 2022, pp. 1–13, Nov. 2022, doi: 10.1155/2022/1339469.

24. Y. M. Abid et al., “Development of an intelligent controller for sports training system based on FPGA,” J. Intell. Syst., vol. 32, no. 1, p. 20220260, Aug. 2023, doi: 10.1515/jisys-2022-0260.

25. S. Jameer and H. Syed, “Deep SE-BiLSTM with IFPOA Fine-Tuning for Human Activity Recognition Using Mobile and Wearable Sensors,” Sensors, vol. 23, no. 9, p. 4319, Apr. 2023, doi: 10.3390/s23094319.

26. M. Khalaf, H. Najm, A. A. Daleh, A. Hasan Munef, and G. Mojib, “Schema Matching Using Word-level Clustering for Integrating Universities’ Courses,” in 2020 2nd Al-Noor International Conference for Science and Technology (NICST), Baku, Azerbaijan: IEEE, Aug. 2020, pp. 1–6. doi: 10.1109/NICST50904.2020.9280318.

27. A. Ahmed, “Pre-trained CNNs Models for Content based Image Retrieval,” Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 7, 2021, doi: 10.14569/IJACSA.2021.0120723.

28. Abd Aliwie, A. N. (2025). Conversational Silence in Harold Pinter’s The Birthday Party: A Pragmatic Perspective. International Journal of Arabic-English Studies. https://doi.org/10.33806/ijaes.v25i2.860

29. Faculty of Computer Sciences, Department of Computer Science, Lahore Garrison University, DHA phase 6, Lahore 54000, Pakistan et al., “Brain Tumor Detection Enhanced with Transfer Learning using SqueezeNet,” Decis. Mak. Adv., vol. 2, no. 1, pp. 129–141, Apr. 2024, doi: 10.31181/dma21202432.

30. R. Kait and K. University, “Enhancing Fog Computing Performance with SqueezeNet Approach for IoT Applications”. doi:10.1109/ICACCTech65084.2024.00114.

31. Y. N.-E. Aine and C. Leghris, “Secure IoT Seed-based Matrix Key Generator,” Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 3, 2024, doi: 10.14569/IJACSA.2024.01503108.

32. A. Yousif and A. H. Kashmar, “Key Generator to Encryption Images Based on Chaotic Maps,” vol. 60, 2019. doi: 10.24996/ijs.2019.60.2.16

33. M. S. Mahdi, R. A. Azeez, and N. F. Hassan, "A proposed lightweight image encryption using ChaCha with hyperchaotic maps," vol. 8, no. 4, 2020.doi: 10.21533/PEN.V8I4.1708.G696.

34. H. Ansaf, H. Najm, J. M. Atiyah, and O. A. Hassen, "Improved Approach for Identification of Real and Fake Smile using Chaos Theory and Principal Component Analysis," J. Southwest Jiaotong Univ., vol. 54, no. 5, 2019. https://doi.org/10.35741/issn.0258-2724.54.5.20.

35. H. R. Mahmood, D. K. Gharkan, G. I. Jamil, A. A. Jaish, and S. T. Yahya, "Eye Movement Classification using Feature Engineering and Ensemble Machine Learning," Eng. Technol. Appl. Sci. Res., vol. 14, no. 6, pp. 18509–18517, Dec. 2024, doi: 10.48084/etasr.9115.

36. D. K. Ghurkan and A. A. Abdulrahman, "Construct an Efficient DDoS Attack Detection System Based on RF-C4.5-GridSearchCV," in 2022 Iraqi International Conference on Communication and Information Technologies (IICCIT), Basrah, Iraq: IEEE, Sep. 2022, pp. 120–124. doi: 10.1109/IICCIT55816.2022.10010645.

37. E. H. Hassan, A. H. Ali, R. M. Shehab, M. S. Mahdi, and W. A. Abd Alrida, "Mask Laws to study Texture Features of the Kidney Infection," Iraqi J. Sci., pp. 2261–2270, May 2023, doi: 10.24996/ijs.2023.64.5.14.

38. E. H. Hassan, A. H. Ali, R. M. Shehab, W. A. A. Alrida, and M. S. Mahdi, "Using K-mean Clustering to Classify the Kidney Images," Iraqi J. Sci., pp. 2070–2084, Apr. 2023, doi: 10.24996/ijs.2023.64.4.41.

39. C. Li, J. C. Sprott, W. Thio, and H. Zhu, "A New Piecewise Linear Hyperchaotic Circuit," IEEE Trans. Circuits Syst. II Express Briefs, vol. 61, no. 12, pp. 977–981, Dec. 2014, doi: 10.1109/TCSII.2014.2356912.

40. Bakri, Bilal Ibrahim, Yaser M. Abid, Ghaidaa Ahmed Ali, Mohammed Salih Mahdi, Alaa Hamza Omran, Mustafa Musa Jaber, Mustafa A. Jalil, and Roula AJ Kadhim. "USING DEEP LEARNING TO DESIGN AN INTELLIGENT CONTROLLER FOR STREET LIGHTING AND POWER CONSUMPTION." Eastern-European Journal of Enterprise Technologies 117, no. 8 (2022), doi: 10.15587/1729-4061.2022.260077.

41. Alsudani, Munther Abdul Ameer. "Self-organizing control for telecommunication networks 5G." In AIP Conference Proceedings, vol. 2591, no. 1. AIP Publishing, 2023., doi: 10.1063/5.0119566.

FINANCING

No financing.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Hayder Najm, Mohammed Salih Mahdi, Sanaa Mohsin.
Data curation: Mohammed Salih Mahdi, Sanaa Mohsin.
Formal analysis: Hayder Najm, Mohammed Salih Mahdi, Sanaa Mohsin.
Research: Hayder Najm, Mohammed Salih Mahdi, Sanaa Mohsin.
Methodology: Hayder Najm, Mohammed Salih Mahdi, Sanaa Mohsin.
Project management: Hayder Najm, Mohammed Salih Mahdi, Sanaa Mohsin.
Resources: Hayder Najm.
Software: Hayder Najm, Mohammed Salih Mahdi, Sanaa Mohsin.
Supervision: Mohammed Salih Mahdi, Sanaa Mohsin.
Drafting - original draft: Hayder Najm, Mohammed Salih Mahdi, Sanaa Mohsin.
Writing - proofreading and editing: Hayder Najm, Mohammed Salih Mahdi, Sanaa Mohsin.