Protection IoT Data Using Hybrid Cryptography Algorithms

Maha Altalhi[1†] and Khalid Alsubhi[2††],

King Abdulaziz University, Faculty of Computing and Information Technology,
Jeddah, KSA

Summary
The Internet of Things (IoT) is a fast-growing technology that has modernized human lives and provided numerous benefits worldwide. The IoT connects many objects over the Internet to transmit information and perform tasks based on sensor information. This technology has become widely used in many fields, such as smart homes, smart cities, and medicine. With this revolution in IoT technology and the increase in demand for it, security concerns and data confidentiality have become an important concern for consumers of IoT applications. If IoT applications depend on the production of big data, keeping it secure is a significant challenge. The most important way to protect data from security threats is to store it in encrypted form.
In this research, we will study three cases. In Case (1), we apply a proposed hybrid cryptography algorithm consisting of two types of encryption algorithms, the symmetric DES algorithm and the asymmetric RSA algorithm. This approach is applied to Mhealth, a big IoT dataset, to protect it from unauthorized access during data storage. In Case (2), a data compression technology is applied before the hybrid cryptography algorithm. This reduces both the required storage space and the encryption and decryption time. Finally, in Case (3), we use a deep learning model, the auto-encoder model, to extract some critical and sensitive data features before applying the hybrid cryptography algorithm. The three approaches are compared by measuring their encoding time, decoding time, and throughput. We determine that Case (3) is the most efficient approach: it achieves 10% faster encryption and decryption than Case (2), which is in turn 49% more efficient that Case (1).

Keywords:
Hybrid cryptography algorithm, big data, compression, encryption, decryption.

1. Introduction

There is rapid growth in the use of digital data. Data has become more widespread and voluminous due to the spread of information technology in our daily lives, so there is an urgent need to increase the security of this data to increase our confidence in it and make it easier for us to deal with it. The Internet is expanding, and many new applications and technologies for the Internet have appeared, such as the Internet of Things (IoT). The IoT is a group of heterogeneous objects connected to each other, interacting and exchanging information to accomplish specific tasks [1]. IoT applications increase users' comfort, efficiency, and speed in completing work. Therefore, IoT applications have spread widely in many fields, such as education, medicine, smart cars, and smart cities.

In 2018, nearly seven billion people used the Internet for tasks such as sending and receiving e-mail, playing electronic games, browsing news, and buying products and services [2]. This spread of the Internet has led to expansion of the IoT. This term is based on the communication of many devices via the Internet Protocol to complete tasks, make decisions, and exchange information [1]. Each connected object generates significant amounts of data, which is often highly sensitive, to allow it to take the required actions.

Many IoT applications depend on their work on producing sensitive big data and then storing it in data storage centers or their database. Authorized parties need to be able to efficiently access all or a portion of this data. The rapid development and our daily dependence on the IoT increase our concerns about data security during storage. It is essential that this data in storage cannot be read or modified by unauthorized persons. Therefore, one of the most critical and prominent challenges we face is maintaining the security of this big data. To implement access control to stored data while enabling effective data retrieval, more sophisticated strategies are required [3].

Researchers have paid great attention to increasing the security of IoT-based systems in recent years. One of the most important ways to keep stored data se-cure and confidential and prevent intruder access is to use encryption. This converts a plaintext into unintelligible code using various encryption algorithms [4]. There are two types of cryptographic algorithms [5]:

1) Symmetric encryption algorithms.

Symmetric cryptography lies in using a shared key for encryption and decryption. The sender encrypts the data with a safe key, and the receiver utilizes the same key for data decryption. including Advanced Encryption Standard (AES), Data Encryption Standard (DES), triple des, Ron's Code, and Blowfish [5].

2) Asymmetric encryption algorithms.

Called public-key cryptography as well, two different keys are used for encryption and decryption. In which these

Manuscript received November 5, 2022
Manuscript revised November 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.11.98

two keys, known as a public key and private key. Each of the sender and the receiver has their own private and public keys; one for encryption and one for decryption. such as RSA, Elliptic curve cryptography, Diff-Hillman, and ElGamal [5].

Since data encryption can increase data security and confidentiality and pre-vent its disclosure and unauthorized access, hybrid cryptography algorithms are used to improve the degree of complexity for attackers. These use more than one encryption algorithm at the same time to take advantage of their strengths, as the combination of more than one algorithm provides better results and enhances security during data storage.

Although many researchers have studied the effectiveness of hybrid cryptography algorithms in data security, testing them on samples of data and demonstrating their effectiveness, most have not been tested on big data. This motivated us to move in this direction and apply a hybrid cryptography algorithm to big data.

In this research, we study three cases. In Case (1), we examine how hybrid cryptography technology can encrypt big data and keep the data secure and confidential. We apply a proposed hybrid cryptography algorithm consisting of two types of encryption algorithms, the symmetric DES algorithm and the asymmetric RSA algorithm. This approach is applied to Mhealth, a big IoT dataset, to protect it from unauthorized access.

In Case (2), we first apply a data compression technology. Then, the proposed hybrid encryption algorithms are used to encrypt the compressed data. This helps improve the efficiency of the use of storage space by reducing the size of the big data before encrypting it.

Because compressed data is safer and more manageable, data compression is an essential aspect of information security [6]. The goal of compression is to reduce the data size [7]. Data compression technology that works effectively produces data that is efficient, safe, and easy to store. Lossy and lossless compression algorithms are the two types of compression algorithms. In lossless compression, when data is retrieved during the decompression process, we can reconstruct the exact data from the compressed data—it is mainly used to compress text data. Lossy compression, often used for photos, results in a fraction of data being lost during the decompression phase. These methods may be used for any data file type, including text, audio, video, and picture files [8].

One of the biggest challenges that we face when dealing with big data is the consumption of significant amounts of energy during its transmission and the need for large amounts of storage space. Therefore, compression algorithms for big data can reduce consumption of energy and electronic components [6].

Finally, in Case (3), we use a deep learning model, the auto-encoder model, to extract some critical and sensitive data features and then encrypt the results using the proposed hybrid encryption algorithm in Case (1).

Three appropriate performance measures have been identified: encoding time, decoding time, and throughput. These performance metrics are calculated for the three cases and compared to determine the most efficient approach.

We can summarize the contributions of this research as follows:

- Improve the security of IoT data, while it is being stored, by applying a hybrid encryption algorithm (RSA+DES) to a big data of IoT data (the Mhealth dataset) to prevent it from being accessed by intruders.
- Attempt to improve the performance of the hybrid encryption algorithm by means of a lossless compression algorithm (Huffman coding) to reduce the size of the data before encrypting it.
- Attempt to improve the performance of the hybrid encryption algorithm by using a deep learning model, the auto-encoder model, to extract some important and sensitive data features before encryption.
- Compare the three approaches in terms of the following performance measures: encoding time, decoding time, and throughput.
- Contribute to the IoT literature. Since the number of cryptography studies based on the Mhealth dataset is small, working with this dataset could be considered a significant contribution to the literature.

The remainder of the paper is organized as follows: Section 2 we review related work; Section 3 show methodology for proposed approach, followed by the Results and Discussion in Section 4. and finally, we conclude this paper with Conclusion and Future Works in Section 5.

2. Literature Review

2.1 Hybrid Cryptography Algorithms

This section provides a detailed survey of published works on hybrid cryptography algorithms.

In [9], a new hybrid cryptography approach, involving the RSA and DES algorithms, was used to improve the data security of information stored in the cloud. The proposed approach uses a hybridization of the RSA and DES algorithms to encrypt data when it is uploaded to the cloud. The same algorithms are used in reverse for data decryption to download data from the cloud. The technique was implemented using the Java programming language and was tested on a sample plaintext. The study concluded that the proposed algorithm is practical and provides high data security for cloud storage. It is also concluded that the

proposed hybrid approach provides more security compared to applying these algorithms alone. [9] highlight the study limitations that could be future work directions, test the applicability of the proposed approach for other IoT applications, and test these algorithms on a dedicated simulator or a real environment. Moreover, test the performance of combining other cryptography techniques. Also, the effectiveness in terms of time for secure cloud storage has to be measured and compared with different existing techniques.

Other work has highlighted the need to provide a secure algorithm that protects internet communications from cryptanalysts.

In [10], a hybrid encryption approach was proposed with the goal of increasing private key security by integrating the study's module with the RSA algorithm. The proposed model provides a password that combines two parts; the sender generates one, and the algorithm generates the other part. The password is sent to the user before encrypting the message, and the user can proceed further only if the password is validated. So, such a model increases the private key's security, since if the private key is known to the hacker, another step is needed for them to compromise the system. The hybrid RSA algorithm has three modules: password generation, validation, and the RSA algorithm. By adding both password generation and validation modules to the RSA algorithm, the user can transmit the message securely to the receiver. Another security layer encrypts the message, increasing the difficulty for someone to intercept the message. Two-factor authentication is used to improve the RSA algorithm security. The study concluded that the hybrid RSA algorithm's security is significantly better than the standard RSA algorithm, but it requires more processing time.

Reliable data transmission is an important consideration for the security of IoT systems. A Hybrid encryption algorithm was proposed in [11] that aims to increase the security and encryption speed by reducing safety risks and computational complexity. The aims of this hybrid algorithm are confidentiality, data integrity, and nonrepudiation in data communication for IoT. The novel proposed HAN algorithm uses a hybrid of AES symmetric encryption algorithms and the NTRU asymmetric encryption algorithm. The HAN algorithm generates a key with high speed and performs encryption and decryption with satisfactory security in IoT applications. The HAN algorithm applies a multinomial approach for encryption, decryption, and digital signatures. The proposed encryption algorithm was simulated using MATLAB software, and it was evaluated in terms of the velocity and safety efficiency in comparison with existing algorithms. The main conclusion was that the safety of this algorithm was confirmed due to the multinomial approach used in encryption phase, decryption, and digital signatures. The algorithm uses a small amount of memory due to its low space complexity, and also offers a high level of security and protection from attacks.

In [12], a hybrid security algorithm was proposed to enhance data integrity, confidentiality, and security for IoT-based smart home applications. It was concluded that the hybrid security technique is more secure than traditional algorithms. The proposed technique consists of three phases: key generation, encryption, and decryption. The proposed hybrid security technique is a combination of the TwoFish and NTRU symmetric algorithms. NTRU is used for encryption and decryption as it provides robust security in a range of IoT devices. The Python programming language was used to implement the algorithms. The study presented a comparative analysis with existing security algorithms regarding the execution time and the level of security. The results verified that the hybrid algorithm gives better results in terms of execution time, with improved security and confidentiality.

2.2 Compression and Cryptography

The study in [13]compared different cryptography algorithms for data security. The data was first compressed, then encryption techniques were applied, and then a comparative analysis was performed for various combinations of compression and encryption techniques. Encryption and compression were performed simultaneously, resulting in faster processing. The researchers assessed the performance using various metrics. They revealed important information regarding cryptography and compression. For data protection, a variety of compression and cryptographic techniques were employed. The researchers compressed the data first and then encrypted the compressed data to secure it. This has numerous benefits: it allows large amounts of data to be transmitted and stored cheaply, quickly, and securely.

In [14], a novel algorithm, called compression with encryption and compression (CEC), was proposed to secure and compress information. This algorithm shortens the data to decrease its size. The consolidated data is then encrypted and compressed, employing a novel encryption algorithm without compromising the reduction competence and the data protection. This CEC algorithm achieves a higher compression percentage and improved data protection, with better privacy and authentication than two existing transmission techniques.

In [15], the Huffman coding algorithm was used to shorten the plaintext, and the cover image was compressed by a discrete wavelet transform, which reduces the size of the cover image via lossy compression. The least significant bit was used to implant the encrypted data in the compacted cover image. The researchers evaluated the system based on percentage protection, squeeze period, reduction percentage,

bits per pixel, mean squared error, extreme signal to noise percentage, structural proximity index, and compression speed. The methodology and implementation in this study were promising, leading to further studies on similar techniques.

In [16], a mixture of AES and Huffman coding was proposed to construct loss-less compressed and encrypted files. Encryption was carried out first and, after that, the compression procedure was applied. To evaluate the level of file protection, measures were made employing the avalanche effect and entropy after encryption and compression. According to the test outcomes, AES encryption has an undesirable effect in growing the file size to about 25% more than the original. However, after Huffman coding, the encrypted file size was reduced by about 30%, resulting in a smaller file size than the original plaintext. The Huffman coding also had a positive impact on security, with a substantial increase in the avalanche effect and entropy.

3. Methodology

This research aims to improve the security and confidentiality of IoT big data by applying a hybrid encryption algorithm (RSA+DES) to provide two layers of encryption, preventing it from being accessed by intruders and increasing the complexity for eavesdroppers.
This section describes the dataset we used, the encryption algorithms used for the hybrid encryption approach, and the three cases we focused on in our study and presents our implementation steps.

3.1 Dataset

A detailed description of the dataset used in this work is as follows. The dataset is called Mhealth [17], and it is based on a mobile app that gathers data using tiny sensors affixed to the user's body in three places: their chest, right wrist, and left ankle. The sensors observe the patient's activity while they are performing 12 specific physical workouts. Additionally, an ECG is continuously recorded by an intelligent sensor to monitor the patient's heart electrical activity. Ten participants were used in this study to collect data on heart related behaviors.

3.2 Encryption Algorithms

3.2.1 DES Algorithm

Information security is the safeguarding of confidential or personal data from various threats to ensure privacy. DES, a well-known symmetric-key block cipher, was released by NIST in 1977. DES uses a very distinctive encryption technique [18] and is categorized as a union cipher class. The encryption key and decryption key are generated from the same key. DES works on a 64-bit block size. DES encrypts 64 plaintext bits into a 64-bit ciphertext utilizing 56 private key bits or subkeys. The inner key is developed from an exterior key that is 64 bits long. DES uses 16 rounds to increase the difficulty of attacks. It is a highly secure cryptographic algorithm and including it in an encryption scheme can help ensure data security [19].

3.2.2 RSA Algorithm

The RSA algorithm, which provides encryption and execution using asymmetric cryptography, also known as public-key cryptography, is one of the most reliable and secure encryption algorithms currently in use. It is commonly used in communication software such as web browsers, chat and email services, and VPNs. The reason it is so frequently used is that people believe the algorithm provides secure encryption that is adequate for their needs. Asymmetric cryptography uses two distinct keys, a public key and a private key, in the encryption and decryption processes, respectively. The public key can be provided to anyone without compromising security, while the private key is kept hidden. The public key is used to encrypt messages from plaintext to ciphertext. The messages that are encrypted with this specific public key can, however, only be decrypted with the corresponding private key. The RSA algorithm's key generation procedure, which has a high level of sophistication compared to other cryptographic methods, is what makes it so secure and reliable [20].

3.3 Research Cases

We studied the applicability of hybrid cryptography algorithms to the dataset in three cases, each employing a different methodology.

3.3.1 Case (1): Hybrid Cryptography Algorithms (RSA+DES)

Because of the increasing use of big data and the wide variety of IoT devices that export big data, it has become imperative for us to pay great attention to the security and confidentiality of this data throughout the data life cycle: when it is transmitted, stored, and used.
Our research focuses only on data during storage, where the big dataset is kept in a repository until it is needed. It must be stored in such a way that no unauthorized party can access, read, or modify it.

One of the most popular and efficient ways to maintain data security and confidentiality is encryption. However, now there is a need to use more advanced and complex techniques to increase the degree of security for big data.

Therefore, we suggest using hybrid encryption algorithms.
In Case (1), a combination of the RSA and DES algorithms was used to store the data securely. The initial stage was to use the RSA algorithm to encrypt the data. After that, the DES algorithm was used for the second layer of security to encrypt the data produced by the first stage. Decryption was carried out in reverse order.
After using the DES and RSA algorithms, data uploading and downloading becomes more secure. An overview of the proposed approach in Case (1) is presented in Figure 1.

We can summarize the steps of hybrid encryption as follows. When the big data is uploaded to a database, it is encrypted by the following steps:
1) RSA is applied to the plaintext for the first level of encryption.
2) DES is applied to the RSA-encrypted text for the second level of encryption.
3) The resulting ciphertext is stored in the database.
When the data is downloaded again, it is decrypted by the following steps:
1) The DES algorithm is applied for the first level of decryption.
2) RSA is applied for the second level of decryption.
3) Finally, the plaintext is available to the authorized user.

3.3.2 Case (2): Compression + Hybrid Cryptography Algorithms (RSA+DES)

In Case (2), the data was first compressed by Huffman coding, with the goal of reducing the necessary storage space and the time needed for encryption and decryption. Then, the proposed hybrid encryption algorithm in Case (1) was applied to secure the data. Data compression technology that works effectively produces data that is efficient, safe, and easy to store. An overview of the proposed approach in Case (2) is presented in Figure 2.

3.3.3 Case (3): Extract Important Features + Hybrid Cryptography Algorithms (RSA+DES)

In Case (3), we trained a deep learning model to compress data to a size of 10 (or even less, like 5) then tried to reconstruct the data by using the model to find a latent code capturing the most important features. The model used was an auto-encoder model consisting of two modules. First, the encoder accepted as input a vector of length 24 represent the features in each sample. The encoder down-sampled the vector to the desired size using a full connected layer of size [batch size – 24(features size)]. After that, we used an activation function (ReLU) to check which node was activated. The output of the encoder was a latent code of size 10—a smaller code size could lead to an unstable model which could fail to reconstruct the vector. The decoder's goal was to reconstruct the 24-length vector back; this module also contained a fully connected layer. The model is highly sensitive to the loss function. We found that L2 was a suitable choice in this case. The model was trained for 2 epochs with a batch size of 5. the model toke around 1 hour to converge. The full model is shown in Figure 3.

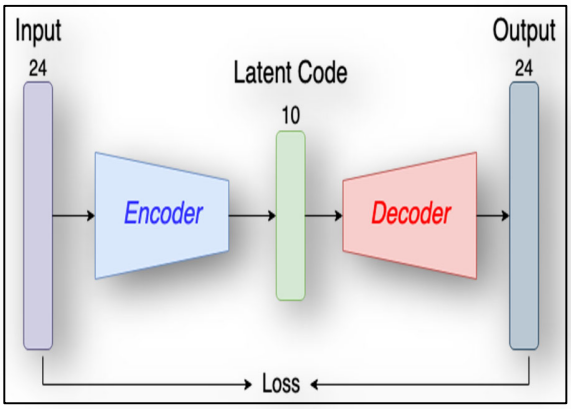

Fig. 3 The auto-encoder model

After carrying out feature extraction using the auto-encoder model and achieving the goal of reducing the size of the dataset by extracting the important features in the form of a latent code, the latent code was encrypted using the proposed hybrid encryption algorithms in Case (1). An overview of the proposed approach in Case (3) is presented in Figure 4.

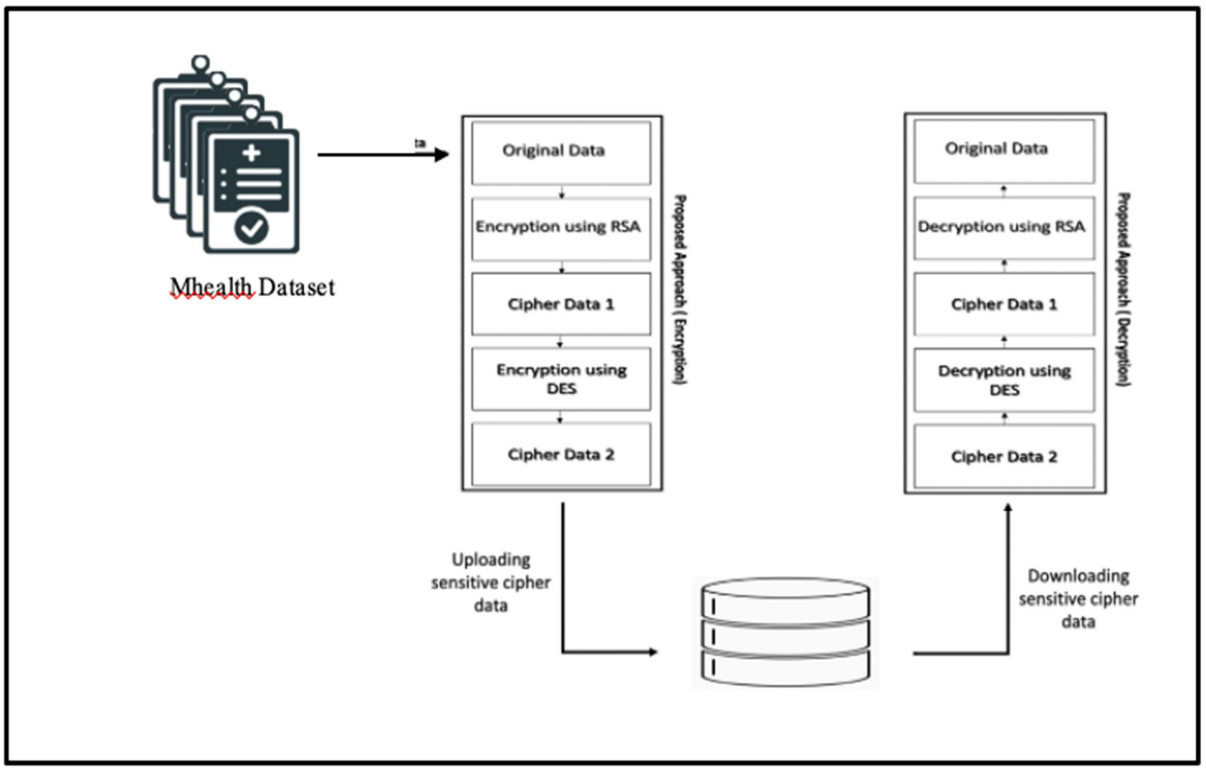

Fig. 1 Overview of the proposed approach in Case (1)

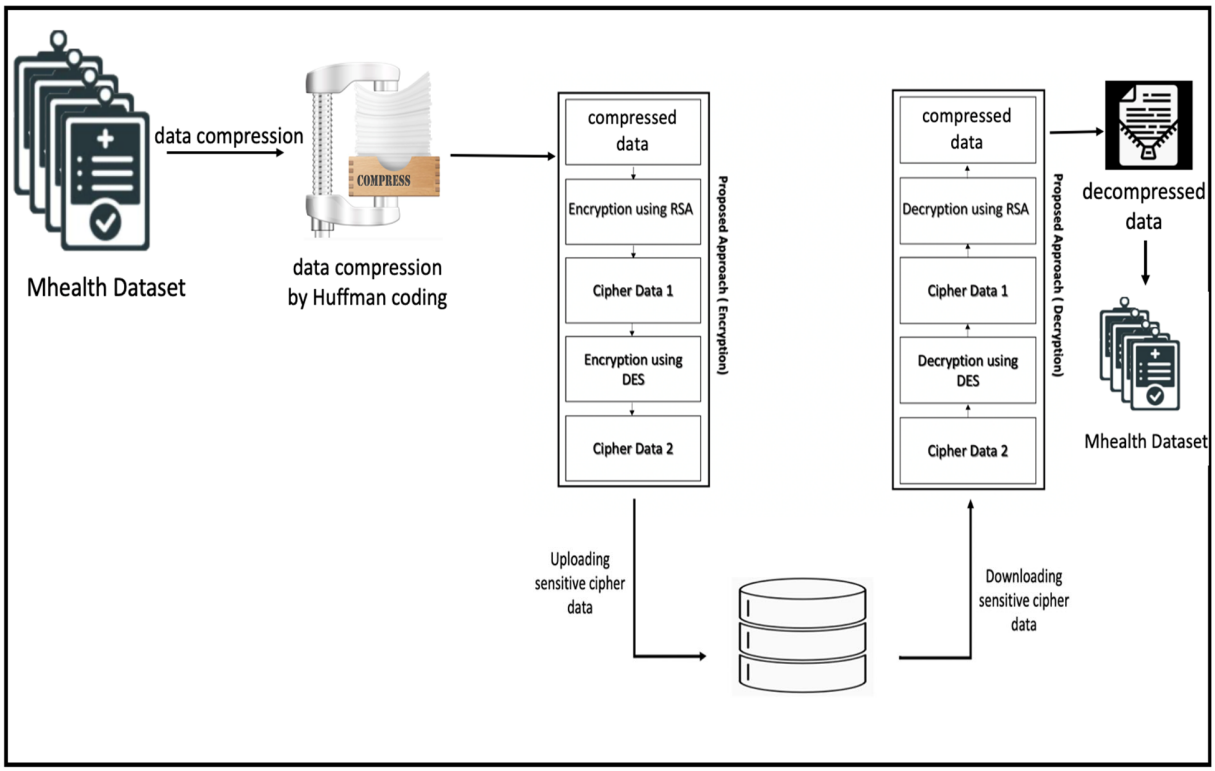

Fig. 2 Overview of the proposed approach in Case (2)

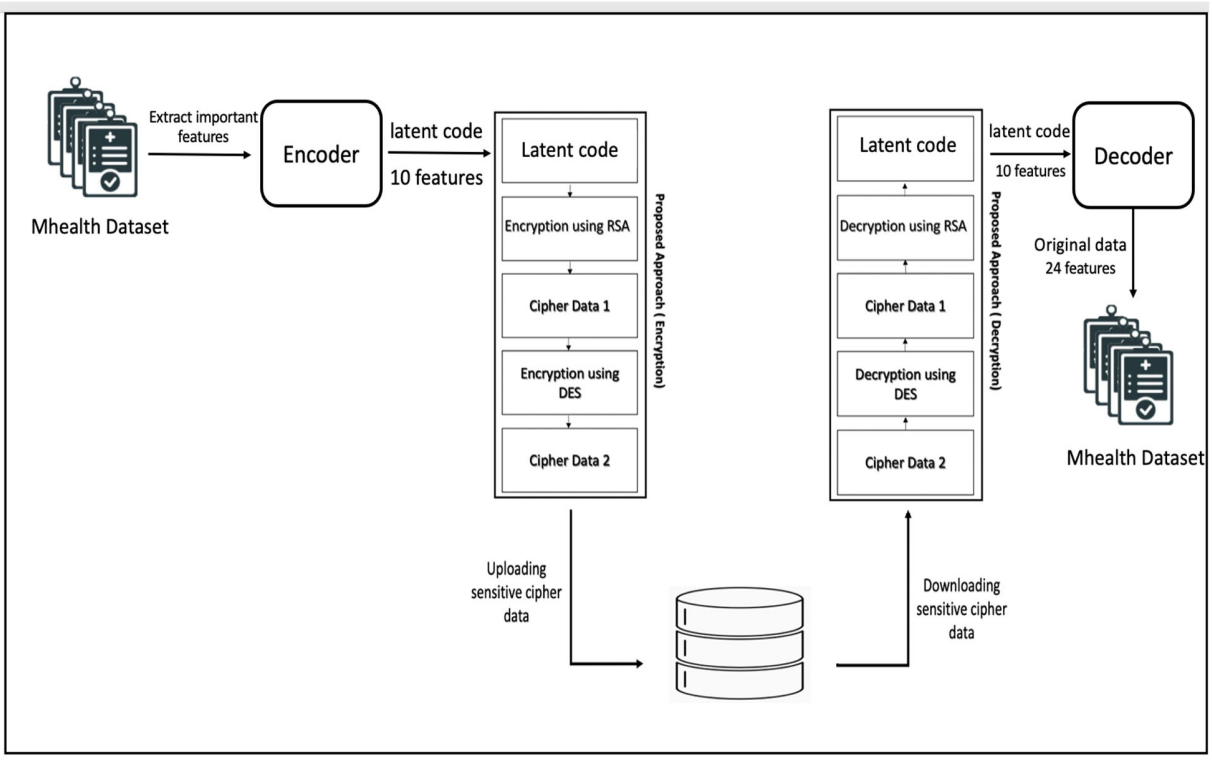

Fig. 3 Overview of the proposed approach in Case (3)

4. Results and Discussion

This section presents and discusses the experiments used to assess the performance of the proposed hybrid cryptography algorithms. An IoT-based big dataset was used for this purpose. The goal of the proposed approach is to protect against third-party access, intermediary users, and unlawful access.

As mentioned in Section 3, we applied hybrid coding algorithms to the Mhealth dataset to evaluate the proposed approach and its efficiency, and our study was divided into three cases, as follows:

Case (1): hybrid cryptography algorithms (RSA+DES).

Case (2): compression + hybrid cryptography algorithms (RSA+DES).

Case (3): extract important features + hybrid cryptography algorithms (RSA+DES).

The implementations were developed using the PyCharm application. The code was written in Python. It was applied to 10 files of different sizes to see the effect of data size on performance. The results were obtained using the Aziz Supercomputer at the KAU High-Performance Computing Center.

Then, the effectiveness of the proposed approach was evaluated using several performance measures, such as encryption time, decryption time, and throughput for the three cases. The results were compared to determine which of the cases achieved the highest efficiency.

4.1 Encryption Time

Encryption time is a popular indicator for assessing the effectiveness of encryption algorithms. Encryption time is the total amount of time it takes to convert plaintext to ciphertext, including all layers of encryption [21]. The throughput of the encryption technology can be estimated based on the encryption time.

In this work, the encryption time was calculated for a set of files of different sizes from the Mhealth dataset, for each of the three different cases. The obtained encryption times are listed in Table 1.

Table 1 shows the encryption times for the three cases, and these results are also plotted in Figure 5. It is clear that Case (3) is the most efficient, as it achieved the lowest encryption time for each of the file sizes.

As can be seen in Figure 5, Case (2) also achieved substantially faster encryption than Case (1). On average, Case (2) achieved a 49% faster encryption time than Case (1), while Case (3) was 10% faster than Case (2). This demonstrates that smaller file sizes resulted in shorter encryption times. Thus, we conclude that extracting

features using the auto-encoder model and the standard compression algorithm can both successfully reduce encryption times.

Table 1: Encryption times (s)

S.NO	*File size (MB)*	*File size after compression (MB)*	*Latent code size (MB)*	*Case 1*	*Case 2*	*Case 3*
1	30	15.3	13.9	5275	2690	2444
2	25.1	12.9	11.7	4428	2275	2064
3	24.1	12.6	11.2	4234	2213	1967
4	24	12.4	11.3	4296	2219	2022
5	22.8	11.7	10.6	3995	2050	1857
6	22.3	11.3	10.4	3924	1988	1830
7	21.8	10.9	10.1	3841	1920	1779
8	19.5	10.1	9	3425	1773	1580
9	18.5	9.5	8.6	3253	1670	1512
10	18.2	9.4	8.5	3218	1662	1502

4.2 Decryption Time

Decryption time estimates the time to convert the ciphertext back to plaintext. The throughput of the described hybrid cryptographic algorithms may be predicted using the decryption time.

The decryption time was calculated for a set of files of different sizes from the Mhealth dataset, for all three cases. The obtained decryption times are depicted in Table 2.

Table 2 shows the decryption times for the three cases. It is clear that the efficiency of Case (3) is the highest because it achieved a shorter encryption time for each file size, so the third case achieves better results. The results are plotted in Figure 6

It can be seen in Figure 6 that Case (3) achieved lower decryption times than Case (2)—about 10% on average—while Case (2) was substantially more efficient than Case (1). On average, the decryption times achieved by Case (2) were about 49% shorter than Case (1). Thus, we conclude the efficiency of extracting features using the auto-encoder model then hybrid encryption is substantially more efficient than the hybrid encryption alone.

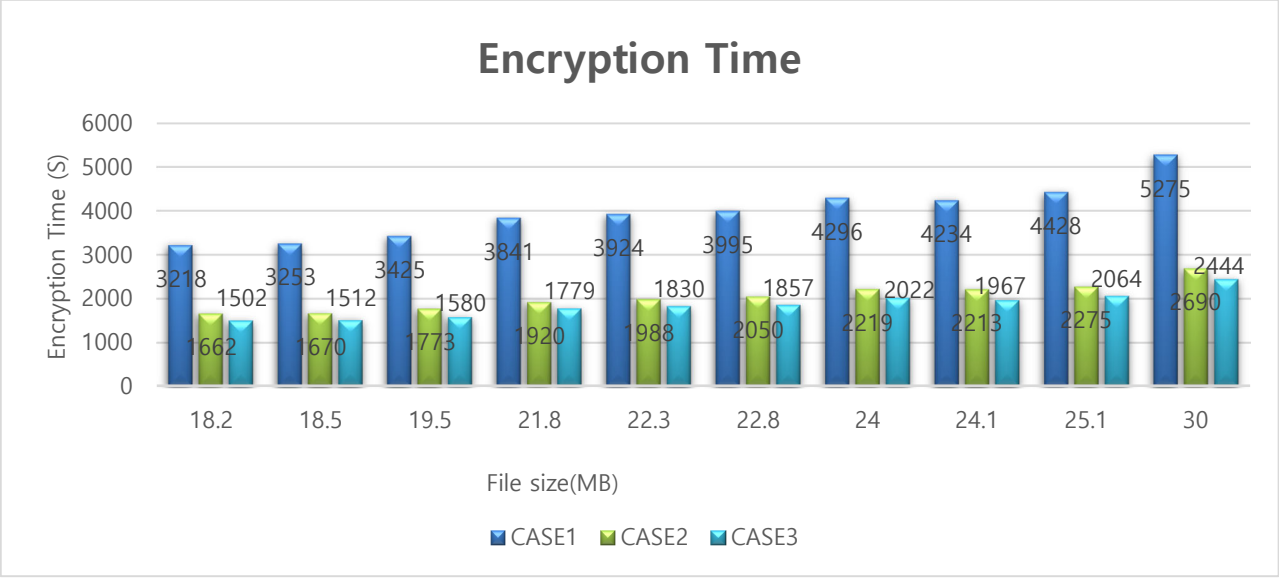

Fig. 5 Encryption times

Table 2: Decryption times (s)

S.NO	*File size (MB)*	*File size before decompression (MB)*	*Latent code size (MB)*	*Case1*	*Case2*	*Case3*
1	30	15.3	13.9	6340	3233	2937
2	25.1	12.9	11.7	5528	2841	2576
3	24.1	12.6	11.2	5340	2791	2481
4	24	12.4	11.3	5300	2738	2495
5	22.8	11.7	10.6	4005	2055	1861
6	22.3	11.3	10.4	3991	2022	1861
7	21.8	10.9	10.1	3884	1942	1799
8	19.5	10.1	9	3525	1825	1626
9	18.5	9.5	8.6	3380	1735	1571
10	18.2	9.4	8.5	3320	1714	1550

4.3 Throughput

Throughput is the rate at which a cryptographic system can process data, and the throughput of encryption is calculated by dividing the average total plaintext size by the encryption time. When comparing approaches, a high throughput corresponds to a high level of efficiency.

The encryption throughput is calculated using the equation

$$\text{encryption throughput} = \frac{\text{total size of plaintext}}{\text{total encryption time}} \quad (1)$$

Table 3 shows the throughput results. The total plaintext size in Case (1) is 226.3 MB. Then, by dividing the entire plaintext quantity by 10, the average encryption time can be calculated, yielding 22.63 s. The total encryption time can then be calculated, using Table 1, as 39889 s.
Similarly, the throughput value of the decryption process is calculated using

$$\text{decryption throughput} = \frac{\text{total size of plaintext}}{\text{total decryption time}} \quad (2)$$

From Table 3, Compared to Case (1) and Case (2), it is evident that Case (3) achieves a high throughput value. Figure 7 shows plots of the throughput values.

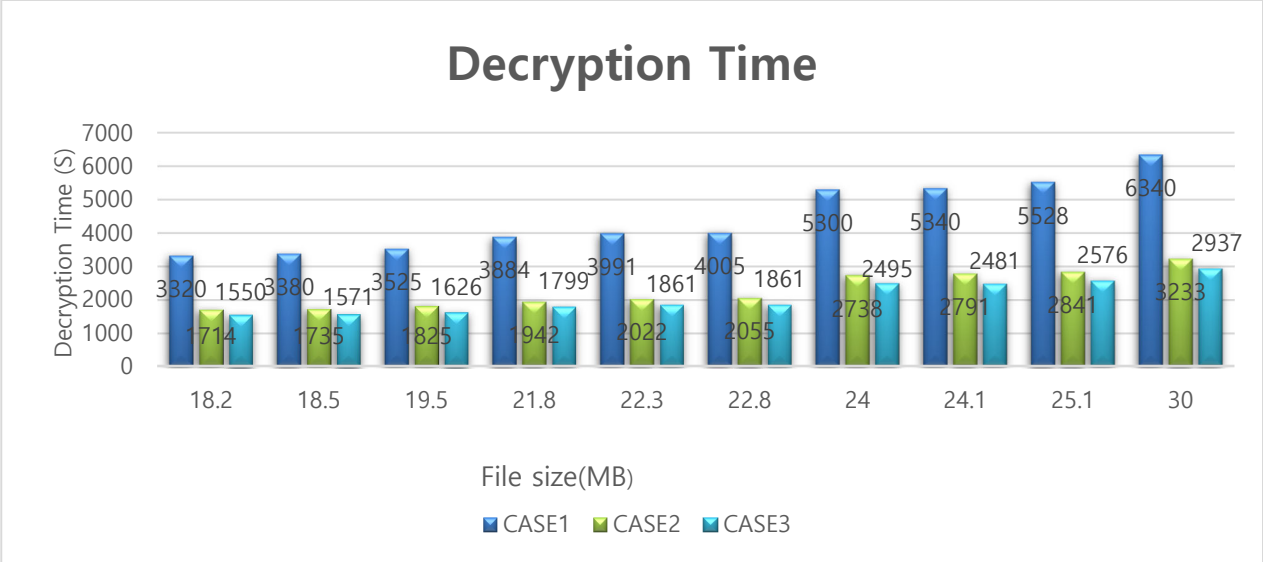

Fig. 6 Decryption times

Table 3: Throughput

	Case1	*Case2*	*Case3*
Throughput encryption	0.00056	0.011	0.012
Throughput decryption	0.00050	0.009	0.010

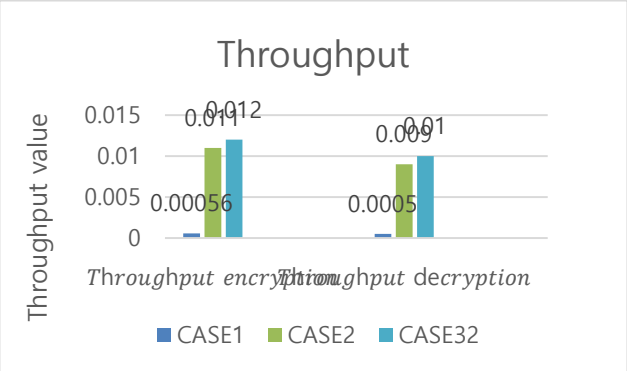

Fig. 7 Throughput

5. Conclusion

The aim of this study was to verify the efficiency of using hybrid cryptography techniques in solving big data security problems in the IoT. In this context, the hybrid encryption of big data was studied in three cases, which are as follows. In Case (1), we applied a proposed hybrid cryptography algorithm consisting of two types of encryption algorithms: DES, a symmetric algorithm, and RSA, an asymmetric algorithm. In Case (2), we compressed the data first to reduce its size using Huffman coding and then applied the same hybrid cryptography algorithm. Finally, in Case (3), we used a deep learning model, the auto-encoder model, to extract some important and sensitive data features and then encrypt them using the same hybrid encryption algorithm as the other two cases.

The cases were applied to a large IoT dataset, Mhealth-dataset to protect it from unauthorized access while storing data. The appropriate performance measures have been identified encoding time, decoding time, and throughput. The performance criteria were calculated for the three cases and compared to evaluate our results and determine the most efficient case. We noticed from our results that Case (3) had low encryption and decryption times compared to the first and second cases. Case (2) was also substantially faster than Case (1). The average drop-in encoding time and decoding time between the first and second cases was about 49%, while the drop was about 10% between the third and second cases. The encryption and decryption times were closely related to the file sizes: the smaller the file size, the lower the encryption and decryption times. As for the throughput, the third case achieved an increase compared to the first and second cases. Thus, we conclude that the extraction of some features using the auto-encoder model, is a highly efficient approach.

6. Future Work

In this research, we studied the efficiency of encrypting big data for the IoT using hybrid encryption algorithms (RSA+DES) during storage. In future, it would be interesting to study the effectiveness of hybrid encryption algorithms for IoT data during the transmission of data with security protecting them from eavesdroppers on the network.

Acknowledgments

The computations for the work presented in this paper were supported by the KAU High Performance Computing Center (Aziz Supercomputer) (http://hpc.kau.edu.sa)

References

[1] A. Assiri and H. Almagwashi, “IoT security and privacy issues,” in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, 2018, pp. 1–5.

[2] G. Cerullo, G. Mazzeo, G. Papale, B. Ragucci, and L. Sgaglione, “IoT and sensor networks security,” in *Security and Resilience in Intelligent Data-Centric Systems and Communication Networks*, Elsevier, 2018, pp. 77–101.

[3] A. Hamlin, N. Schear, E. Shen, M. Varia, S. Yakoubov, and A. Yerukhimovich, *Cryptography for big data security*. 2016. doi: 10.1201/b19694-13.

[4] A. Shrivastava and L. Singh, “A new hybrid encryption and steganography technique: a survey,” *Int. J. Adv. Technol. Eng. Explor.*, vol. 3, no. 14, pp. 9–14, 2016, doi: 10.19101/ijatee.2016.314005.

[5] F. Meneses *et al.*, “RSA Encryption Algorithm Optimization to Improve Performance and Security Level of Network Messages,” *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 16, no. 8, p. 55, 2016.

[6] O. F. A. Wahab, A. A. M. Khalaf, A. I. Hussein, and H. F. A. Hamed, "Hiding data using efficient combination of RSA cryptography, and compression steganography techniques," *IEEE Access*, vol. 9, pp. 31805–31815, 2021, doi: 10.1109/ACCESS.2021.3060317.

[7] K. Sayood, *Introduction to Data Compression (The Morgan Kaufmann Series in Multimedia Information and Systems)*, 4th ed.

[8] R. Makala, V. Bezawada, and R. Ponnaboyina, "A fast encryption and compression technique on SMS data," *Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017*, vol. 2018-Janua, pp. 1213–1217, 2018, doi: 10.1109/WiSPNET.2017.8299956.

[9] A. Kumar, V. Jain, and A. Yadav, "A new approach for security in cloud data storage for IOT applications using hybrid cryptography technique," in *2020 international conference on power electronics & IoT applications in renewable energy and its control (PARC)*, 2020, pp. 514–517.

[10] M. K. Sai, N. Alapati&Sivaramakrishna, R. Teja, and B. Kolla, "A Hybrid Approach for Enhancing Security in IOT using RSA Algorithm," *HELIX*, vol. 9, p. 4758, 2019.

[11] S. K. Mousavi, A. Tabe, K. Shaker, and P. Hatamzade, "Analyzing Security Strategies in the Internet Of Things," *Adv. Interconnect Technol. An Int. J.*, vol. 1, no. 1.

[12] S. Srivastava and S. Prakash, "Security enhancement of IoT based smart home using hybrid technique," in *International Conference on Machine Learning, Image Processing, Network Security and Data Sciences*, 2020, pp. 543–558.

[13] R. Sharma and S. Bollavarapu, "Data Security using Compression and Cryptography Techniques," *Int. J. Comput. Appl.*, vol. 117, no. 14, 2015.

[14] M. B. Begum and Y. Venkataramani, "A new compression scheme for secure transmission," *Int. J. Autom. Comput.*, vol. 10, no. 6, pp. 578–586, 2013.

[15] S. Singh and R. Devgon, "Analysis of encryption and lossless compression techniques for secure data transmission," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019, pp. 1–5.

[16] M. R. Ashila, N. Atikah, E. H. Rachmawanto, and C. A. Sari, "Hybrid AES-Huffman Coding for Secure Lossless Transmission," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 2019, pp. 1–5.

[17] O. Banos *et al.*, "mHealthDroid: a novel framework for agile development of mobile health applications," in *International workshop on ambient assisted living*, 2014, pp. 91–98.

[18] B.-Y. Sung, K.-B. Kim, and K.-W. Shin, "An AES-GCM authenticated encryption crypto-core for IoT security," in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, 2018, pp. 1–3.

[19] I. Hussain, M. C. Negi, and N. Pandey, "A secure IoT-based power plant control using RSA and DES encryption techniques in data link layer," in *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS)*, 2017, pp. 464–470.

[20] Y. Chandu, K. S. R. Kumar, N. V. Prabhukhanolkar, A. N. Anish, and S. Rawal, "Design and implementation of hybrid encryption for security of IOT data," in *2017 International conference on smart technologies for smart nation (SmartTechCon)*, 2017, pp. 1228–1231.

[21] M. F. Ali, N. Harum, N. A. Abu, M. N. Al-Mhiqani, M. S. Talib, and A. A. Mohammed, "Protecting iot based transmitted data security using tokenized multiple layered encryption techniques," *Int. J. Adv. Sci. Technol.*, vol. 28, no. 8, pp. 485–505, 2019.