A Hybrid Cryptographic Model Using AES and RSA for Sensitive Data Privacy Preserving

Satish B Basapur
Department of Information Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru, India.
E-mail: satish.basapur@gmail.com

B.S. Shylaja
Department of Information Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru, India.
E-mail: shyla.au@gmail.com

Venkatesh
Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India.
E-mail: venkateshm.uvce@bub.ernet.in

Received May 06, 2021; Accepted August 08, 2021
ISSN: 1735-188X
DOI: 10.14704/WEB/V18SI05/WEB18219

Abstract

In the present scenario, big data is facing many challenges regarding the data privacy and data security. Nowadays, new laws and regulations like GDPR is required for companies to define privacy policies complying with the preferences of their users. This type of regulations prevents the disclosure of sensitive data of users, even if occurs accidentally. In this research, a hybrid cryptographic model based on AES and RSA is proposed to identify and mask the sensitive data to identify many threats to information confidentiality. A hybrid cryptography algorithm in the context of masking is proposed to effectively transfer big data through the cloud. The hybrid algorithm is created by combining more than one algorithm. This algorithm enables the user to select the data to be masked and encrypted. For protecting data stored in the cloud, the proposed hybrid algorithm includes RSA and AES. Along with these algorithms, the multilayer perceptron neural network is used for key generation and key exchange. A credit card client's dataset from the UCI machine learning repository is used for the evaluation. From the dataset the sensitive attributes are selected using the depth first search (DFS) technique. The sender encrypts the data using the RSA algorithm and a key to create masked data using public key. The AES algorithm is used to encrypt the RSA key. The encrypted key and masked data are then sent to the receiver. To decrypt the RSA key, the receiver uses the AES decryption. Finally, the decrypted RSA key is used by the receiver to translate the masked data back to the original data with private key. The proposed model obtained the overall accuracy of 95.23% accuracy and an average computational time of 300 nano secs.

Keywords

Privacy Preserving, Data Security, AES, RSA, DFS, Key Generation and Exchange, Neural Network, Information Confidentiality.

Introduction

Privacy (data privacy) is characterized as an individual's or organization's right to control when, who, and how much data in a computer network is exposed to the third party. It is enshrined in the Indian constitutions as a constitutional right. Any person or entity is supposed to protect data privacy when disseminating data and technologies. Typically, an agency collects data ranging from a tiny amount to a significant amount with or without the consent of the customer. This information can be utilized for determined collection like other uses such as data mining, marketing, automatic decisions making, and profiling. A company can also exchange this information with third parties without the data owners' prior knowledge. Furthermore, data that is not adequately covered by a system could expose the user to risks such as present location, profile exposure, data traffic analysis, identity exposure, or behaviour tracking (Ram G S and Sushmita R. 2020).

Many countries have enacted data security frameworks or other types of regulations to safeguard users' data. The European Union (EU) was the first to propose the General Data Protection Regulation (GDPR), which would act as the legislative tenet for protecting consumer information within EU territories. It is the duty of data processor, according to GDPR, to secure the user's information. GDPR gives data owners some privileges about their data, imposes processing regulations, and limits how companies can access personal information. Likewise, the Indian Government recently proposed the Personal Data Protection Bill-2019 (PDPB-2019) for establishing a legislative basis to protect personal information from organizations working in India. It was a draught designed for securing the information and privacy of every person living in India, and it explains how data processing will take place outside of India (Nils G et al. 2018).

Data Protection = Legal Framework + Technical Standards

Big Data is described as a massive amount of data with a mix of different data formats and a high variety (Priyank J et al. 2016). The vast majority of databases are stored in the cloud storage system for retrieval and analysis. Instead of a PC or a local server, cloud computing allows us to compute over the internet by using a network of distributed server located on the Internet for storing, processing, and handling information. It is the rapidly evolving model for providing easy, on-demand networks and ubiquitous accesses to the variety of computing services (Pan Y et al. 2020). Through the use of Hadoop, cloud computing could perform the distributed query over various databases responses in a periodic way. Hadoop offers a platform for distributed data collection, allowing for the storage and analysis of massive amounts of information on commodities hardware (Ismail H et al. 2018). Cloud computing is a new technique that is increasingly being used for Big Data processing. Since the cloud is untrustworthy, certain cryptographic methods are used to encrypt the user's sensitive data (Yuanzhao et al. 2020).

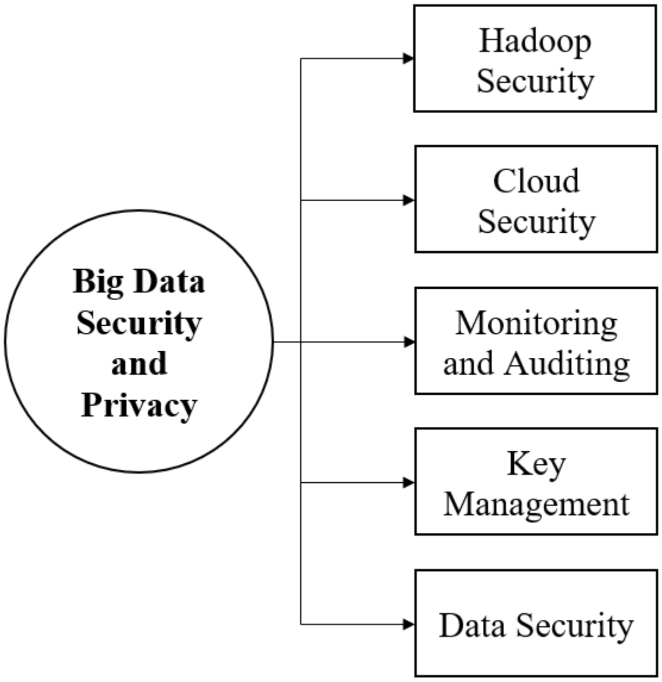


Figure 1 Big Data Security and Privacy Fields

In this research, a hybrid cryptography algorithm in the context of masking is proposed to effectively transfer big data through the cloud. The hybrid algorithm was created by combining more than one algorithm. This algorithm enables the user to select the data to be masked and encrypted. For protecting data stored in the cloud, the proposed hybrid algorithm includes RSA (Rivest Shamir Adleman) and AES (Advanced Encryption Standard). Along with these algorithms, the multilayer perceptron neural network is used for key generation and key exchange. From the dataset the sensitive attributes are selected using the depth first search (DFS) technique. The sender encrypts the data using the RSA algorithm and a key to create masked data using public key. The AES algorithm is used to encrypt the RSA key. The encrypted key and masked data are then sent to the receiver. To decrypt the RSA key, the receiver uses the AES decryption. Finally, the decrypted RSA key is used by the receiver to translate the masked data back to the original data with private key.

Related Works

When used individually, the RSA and AES algorithms were incapable of meeting the requirements of file encoding durability and protection. To address this problem, Abhishek G and Asha A proposed a hybrid algorithm for encryption using AES and RSA for improving the encryptions efficiency and security. According to the comparisons of the RSA and AES algorithm related to encryption and decryption duration, security, management of key, and key lengths, this model integrated the advantages of the two algorithms. In the encryption process, the speed advantage of the AES algorithm was utilized, as well as the reliability and management of key benefit of the RSA algorithm, and the encoding power of each was combined for encrypting the data. However, some flaws remain in this model, like the failure to protect from replay attacks in the file encoding method and tampering (Abhishek G and Asha A. 2021).

Vanaja M et al. 2019, proposed an encryption method that used a variety of encryption algorithms to protect data. The conjunction of the RSA and AES algorithms was utilized in this work to efficiently protect the transmitted data. The RSA algorithm was utilized to encrypt plain text files, and the AES algorithm was later utilized for double encryption. In comparison to various encryption techniques, the integration of AES and RSA was effective in cracking the key and reading the results. In comparison to a single algorithm with the big key length, two separate algorithms could keep the data safe.

Roshan J et al. 2021, proposed an algorithm for security that served as a data encryption method. The encryption algorithm encrypted all data saved in the cloud for users. This encryption algorithm was used to encrypt documents transferred to cloud servers. In addition, the proposed algorithm took less time to encrypt and decrypt files than other algorithms, according to the results. Avalanche effect and entropy values showed that the algorithm's cipher data was equal to industry requirements. The encryption time pattern demonstrated that this algorithm would become increasingly useful as file size increased.

Youssef M.E et al. 2019, proposed the Intelligent Framework for Health care Data security as a security system. IFHDS enables the reliable and efficient processing of big data using a column-based technique with minimal effect on data processing efficiency. The model planned to hide personal information and encode only sensitive information. The IFHDS divided sensitive information into various parts based on sensitivity levels, with every part was saved independently on distributed servers of cloud. Dividing data depending on sensitivity levels prohibited the cloud providers from destroying the whole record of data if a portion of the data was successfully decrypted. The experimental findings represented that this model secured confidential patient data while taking a reasonable amount of time to compute as opposed to other latest security techniques.

Viswanath G and P. V Krishna 2020, proposed an encryption algorithm for protecting big data in a multi-cloud computing environment. This architecture protected big data from tampering, insider, and denial-of-service attacks. For protecting the big data saved in the multi-cloud, this architecture utilized the uploading of data, indexing, slicing, encoding, distributions, decoding, recovery, and combining processes. For achieving a high Avalanche effect, the encryption algorithm in this work combined the features of AES and the Feistel networks. This hybrid encryption algorithm was created to protect big data before storing it in multi-cloud.

Kalaiselvi R.C and S.M. Vennila 2021, implemented a framework with key features like enhanced user’s data protection and privacy. The modified AES-128-bit algorithm was utilized to increase the encryption method's speed. The parameters of the custom-AES-Algorithm like block-sizes, salt-size, secret-key length, and cipher-modes, were modified. This model reduced the duration required for encoding and decoding, utilized lesser memory, and maximized entropy. This model was appropriate for applications where time and memory were main factors.

Proposed Methodology

In this research, the hybrid algorithm is proposed to secure the big data in cloud system. The proposed model protects the big data from privacy and enhances the security. The developed hybrid algorithm integrates the functionalities of RSA and AES algorithms for achieving higher efficiency related to encryption and decryption with masking the data. The Key generation and key exchange are performed by the using the multilayer perceptron neural network model. The dataset from the UCI repository is used for the evaluation.

Big data refers to very large databases. It is an aggregation of massive datasets that cannot be managed using conventional computing techniques. Big data refers to a whole topic rather than just data that can be analyzed using different methods, software, and frameworks. Hadoop is a dynamically distributed computing platform that can efficiently store and share very large datasets at once on servers that can run in parallel. Hadoop stores data in clusters, offering a one-of-a-kind computing approach built on distributed file systems. Hadoop's special capability of projecting data across clusters speeds up data analysis. Hadoop allows businesses to easily access and process data in order to produce the values needed by the organizations, while providing companies with the resources to gain useful insights from different types of data sources that are running in parallel (Divya D and G Santhi. 2021).

The key objective of this proposed model is to improve a secure and efficient model for data privacy with masking by integrating the cryptographic process. For masking the data for preserving privacy, the AES and RSA algorithm with Multilayer Perceptron algorithm is used for key generation and key exchange. The reason for using RSA and AES is that AES is quicker in encoding and decoding whereas RSA utilizes challenging mathematical operations. This integration is significant to encrypt sensitive data. The Depth First Search (DFS) method is used in this research for selecting the sensitive attributes from the dataset. This hybrid algorithm transmits the masked data via the cloud effectively in a secured way.

RSA Algorithm

The most widely utilized public key algorithm for encryption is the RSA. It is both an asymmetric encryption algorithm and an encryption approach based on massive integer factorizations. The RSA algorithm's keys emerge in pairs, and data encryption is performed by the public and private keys. The public key was recognized to all and is utilized for encrypting the information and validate signatures; the private key decodes and signs the data. The RSA algorithm with 1024-bit size is used in this research.

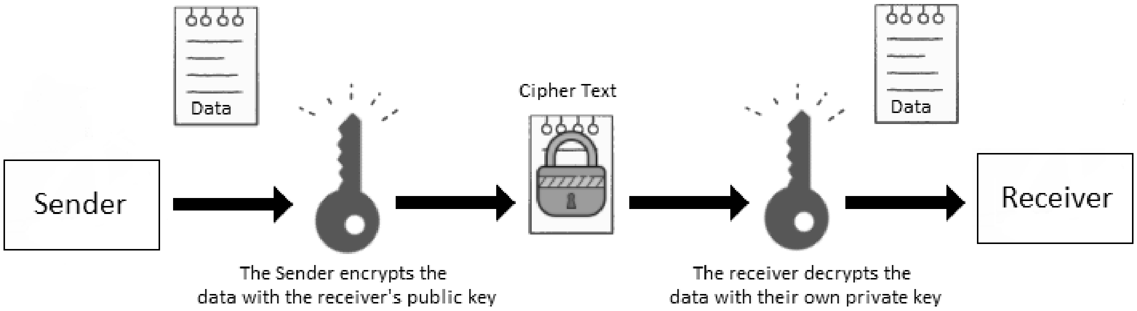

Data
Cipher Text
Data
Sender
Receiver
The Sender encrypts the data with the receiver's public key
The receiver decrypts the data with their own private key

Figure 2 RSA Algorithm Process [Matthias D et al. 2021]

Steps to be followed for RSA Algorithm

- Generate the pair of large, random prime’s *p* and *q*.
- Calculate the modulus *n* as *n* = *p***q*.

- Choose an odd public exponent *e* among 3 and *n*-1 that was similarly prime to *p*-1 and *q*-1.
- Calculate the private exponent *d* from *e*, *p* and *q*.
- Output (*n*, *d*) as private key and (*n*, *e*) as the public key.

$$c = \text{ENCRYPT}\ (m) = m^e \bmod n \qquad (1)$$
$$m = \text{DECRYPT}\ (c) = c^d \bmod n \qquad (2)$$

Overall, the most significant benefit of the RSA technique was that it has strong management of key functionalities and securities, and its security is computationally better than the AES approach. The RSA approach has two private and two public keys. The public key was utilized to encrypt, and the private key was utilized to decrypt, and the key for encryption differs from the decryption’s key. If the data was encrypted with the specific key, it should be decrypted with the same key that vastly improves its protection. The complexity of the mathematical inverse equation of the encryption approach determines the security of RSA cryptosystem. It's known as the complexity of factorizing big numbers. Keys in the RSA approach are powers of big integers; if the key size is longer, it is very difficult to crack (Matthias D et al. 2021).

AES Algorithm

AES is a block cipher with a 128-bit block length. It serves three key lengths: 128, 192, and 256 bits. In this research, AES with a key length of 256 bits was proposed. For 256-bit keys, the encryption process consists of 14 rounds of processing. Except for the final round in every case, the rest of the rounds are equal. Since AES executes all of its operations in terms of bytes rather than bits, the 256-bit was treated as 32-byte. This 32-byte is organised in a 4X4 matrix.

Every round of encryption and decryption consists of the four phases mentioned below:

- SubBytes: a non-linear substitution phase in which every byte was changed with another based on the S-box table.
- ShiftRows: a transposition phase in which every row of the state was moved circularly certain times.
- MixColumns: a mixing operation that works on the state's columns, adding the four bytes in every column.
- AddRoundKey: every byte of the state was integrated with the round key; every round key was extracted using a key schedule from the cipher key.

Just three moves are performed in the final round: SubBytes, ShiftRow, and AddRoundKey. While the same steps are utilized in both encryption and decryption, the order in which the steps are performed differs, as shown in Figure 3.

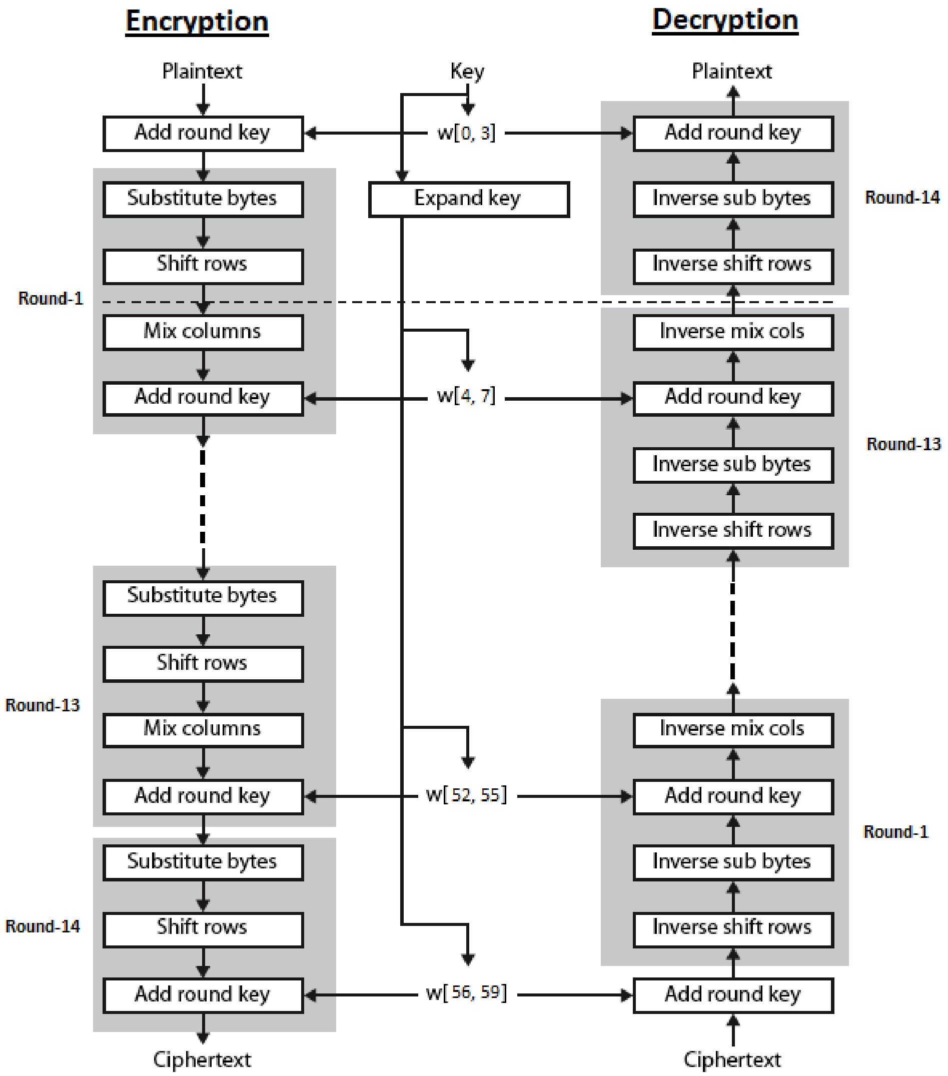


Figure 3 AES-256 Bit Architecture [Shashikant K et al. 2015]

AES reliably outperforms in both software and hardware platforms in a diverse variety of conditions. These incorporate 8 and 64-bit architectures, as well as DSPs. Its intrinsic parallelism allows for more effective use of processing power, resulting in excellent program efficiency. This method has a quick key initialization time and excellent key agility. It takes small memory to execute, making it appropriate for constrained-space conditions. The framework has a high likelihood of learning from instructions-level parallelism. AES has no significant weaker key. It accepts any blocks and key size that is a multiple of 32 (Higher than 128-bits). Even with the large number of test cases, statistical analysis of the cipher text was not feasible.

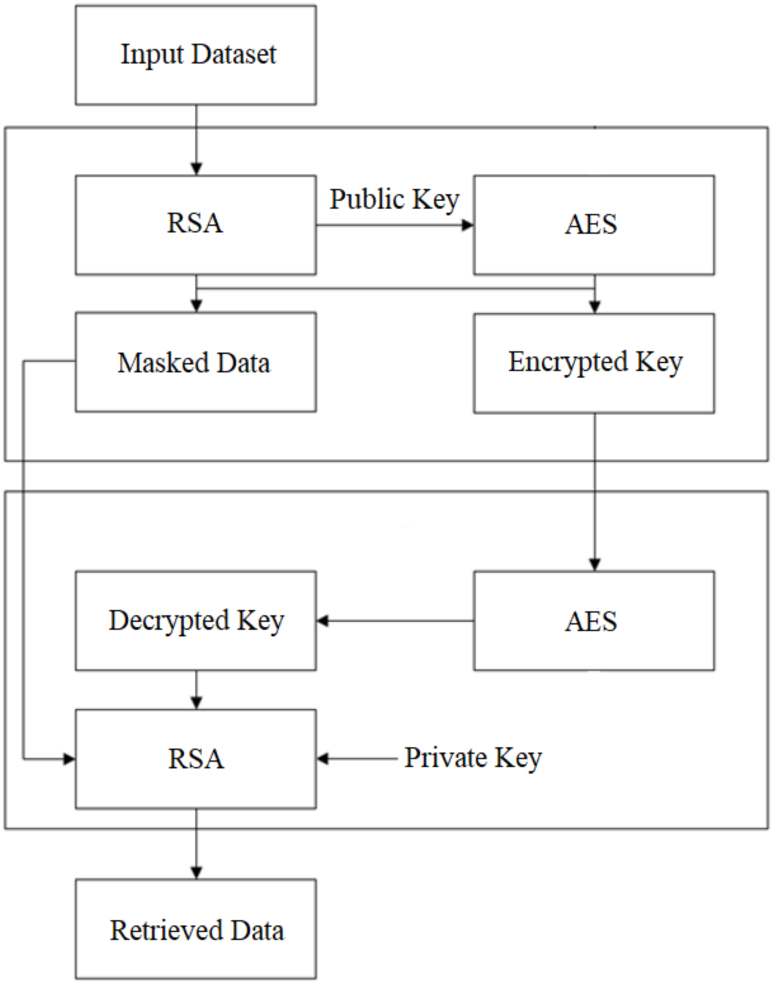


Figure 4 Process of RSA and AES Hybrid Model

The two algorithm’s advantages related to encryption and decryption duration, management of key, security, and key size are based on a comparison of the AES algorithm and the RSA techniques. This work fully utilizes the AES algorithm's speed advantage in the encryption process, as well as the RSA algorithm's reliability and advantage of managing key, and uses the power of both techniques for encrypting the data. Data privacy encompasses many areas, with this research focusing on IC (Information Confidentiality). The above was a basic privacy protection principle in which users require that secrecy of their personal information, also known as sensitive information, be preserved in order to secure it from unauthorized access (Shashikant K et al. 2015).

DFS Approach for Attributes Selection

Searching or tracking is one method to solve general problems. Searching is a process of looking for a solution of a problem through a set of possible state space (state space). Depth-First Search (DFS) algorithm is a method that included of blind search. That is, the search is done by ensuring all vertices have been visited, but without the exact solution or a suboptimal solution. DFS is a searching algorithm that utilizes the data structure collection as it reaches a vertex or node. DFS technique is integrated with the proposed model to select the attributes from the dataset. Initially, the data from the dataset is masked and encrypted. The encrypted data is transferred to the cloud and stored.

The original data is masked (obscured), and the outcomes are permanent; there is no need to reverse the masking. Data masking is a fine-grained security method used to protect field-level data features. It could leave the data extremely portable for data records containing private personally identifiable information (PII), thus leaving various non-personally identifying data types available for usage (Loredana C et al. 2020).

To maintain the AES algorithm’s security, the key should not be reached by the insecure third parties, and the key must be addressed earlier; otherwise, the key would be constantly modified. In this research, the RSA algorithm’s public key was utilized in the hybrid algorithm's encoding process to encrypt the input data to produce masked data, and then the AES algorithm was utilized for encrypting the RSA’s public key and the AES key for creating masked data.

In the RSA approach, the public key was made public, while the private key was utilized for decoding and is kept privately. As the private RSA key was kept secret, public RSA encrypted data cannot be decrypted because the AES key is not included in the key. The information is not encrypted. In the hybrid algorithm, mathematical operations generate the RSA algorithm's public and private keys at random. As the private RSA key is kept confidential, it is not possible to decrypt public RSA encrypted files because the AES key is not included in the key. The files are not encrypted. In the hybrid algorithm, mathematical operations produce the RSA algorithm’s private and public keys at random.

Multilayer Perceptron for Key Generation and Exchange

Between the receiver and sender, a multilayer perceptron (MLP) synaptic simulation weight dependent undisclosed key was generated. For a given session, the multilayer perceptron of the sender and receiver selects the same single hidden layer from a collection of multiple hidden layers. For this session, every other hidden layer was deactivated, which ensures that their hidden (processing) units do nothing to the inputs. Either the synchronized equal weight vectors of the inputs layer, active hidden layers, and output layers of sender and recipient becomes the session's key, or the session's key could be formed utilizing the equivalent hidden unit's output of the activated hidden layers.

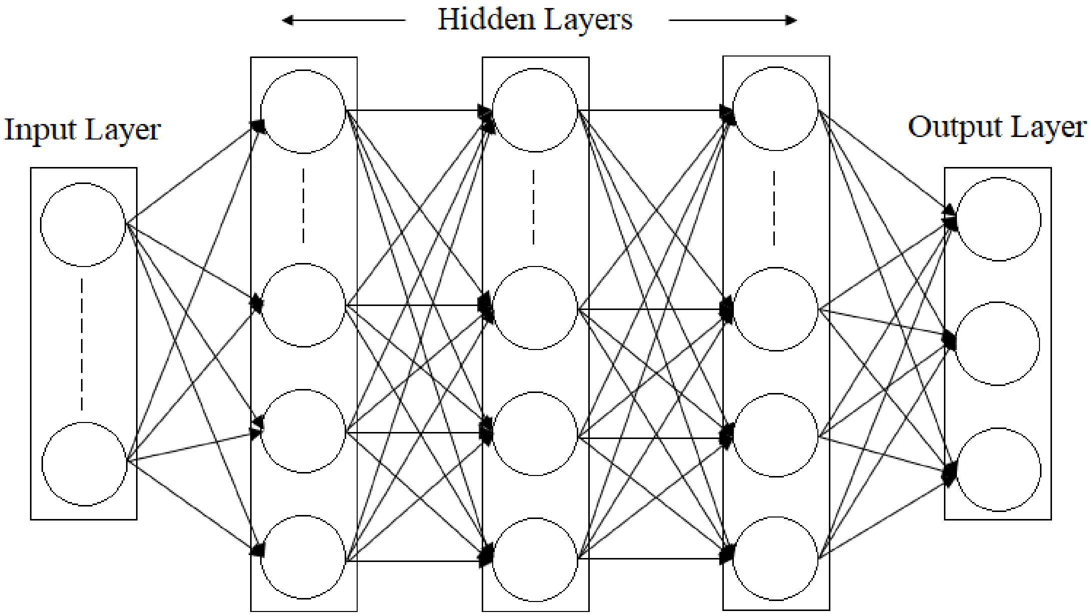


Figure 5 Multilayer Perceptron Neural Network Architecture

In every session, the sender and receiver MLP function as the single layer network with a selected active hidden layers dynamically and K numbers of hidden neuron, N number of inputs neuron with binary inputs vectors, $x_{ij} \in \{-1, +1\}$, discrete weight produced from inputs to outputs that lie among -L and +L, $w_{ij} \in \{-L, -L+1, \dots + L\}$. Where i = 1, ..., K indicates the perceptron's ith secret part and j = 1, ..., N the element of the vectors and single output neuron. The weighted average of existing input values was utilized to determine the output of hidden units. As a result, equation 1 is used to express the state of every hidden neuron.

$$h_i = \frac{1}{\sqrt{N}} w_i x_i = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} w_{i,j} x_{i,j} \qquad (3)$$

The ith hidden unit's output is expressed as,

$$\sigma_i = sgn(h_i) \qquad (4)$$

However, if $h_i = 0$ then $\sigma_i = -1$ to generate a binary result. As a result, $\sigma_i = +1$, if the weighted number of its input was positive; otherwise, it was inactive, $\sigma_i = -1$. A perceptron's total outputs was the hidden unit's product, as formulated in equ 4.

$$\tau = \prod_{i=1}^{k} \sigma_i \qquad (5)$$

Random weights and input vectors for a multilayer perceptron are used as inputs.

The secret key is obtained by synchronizing inputs and outputs neuron as vectors are output.
Step 1: Initialize the weight value of synaptic connections randomly among the input layers and a chosen triggered hidden layer.

Where $w_{ij} \in \{-L, -L+1, \dots + L\}$

Step 2: Repeat step 3 to 6 till the full synchronization was accomplished.

$$w_{i,j}^{+} = g(w_{i,j} + x_{i,j}\tau\Theta(\sigma_i\tau)\Theta(\tau^A\tau^B)) \qquad (6)$$

Step 3: Produce a random input vector X. A third party generates the inputs.

Step 4: Using equation 7, calculate the value of the active hidden neuron of the activated hidden layers.

$$h_i = \frac{1}{\sqrt{N}} w_i x_i = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} w_{i,j} x_{i,j} \qquad (7)$$

Step 5: Using equation 8, calculate the values of the output neurons.

$$\tau = \prod_{i=1}^{k} \sigma_i \qquad (8)$$

Exchanging the outputs of the system, compare the output values of MLP. if Output (A) ≠ Output (B), Go to step 3, else if Output (A) = Output (B), hence one of the appropriate learning-rules was implemented, and only the hidden unit with the output bit equal to typical outputs are trained. Just upgrade the weight if the perceptron's final outputs values were equal. When synchronization was eventually accomplished, the synaptic weight for both systems were similar.

Weight vectors among the input layers and the active hidden layers of MLP system become equal at the completion of the full weight synchronization phase. The performance of the source MLP's enabled hidden layer was utilized to generate the secret sessions key. This session key was not sent across the public channels since the performance of the receiver

MLP is similar to that of the active secret layer. Equation 7 can be used to calculate the values of every hidden unit.

$$\sigma_i = sgn\left(\sum_{j=1}^{N} w_{ij}x_{ij}\right)sgn(x) = \begin{cases} -1 \; if \; x < 0, \\ 0 \quad if \; x = 0, \\ 1 \quad if \; x > 0. \end{cases} \qquad (9)$$

The considered eight hidden units of the active hidden layers with absolute values (1, 0, 0, 1, 0, 1, 0, 1) turns into an eight-bit block. This 10010101 was cascaded XORed with recursive substitution encrypted data to turn into a hidden session key for a specific session. The last session key-based masked data was now sent to the receiver. The receiver has the same session key, which is the contribution of the hidden units of the receiver's active hidden layer. The recursive substitute encrypted text was obtained from the final masked data using this session key. Both machines began tuning again in the following session to generate another session key. After maximum weight synchronization is reached, an identical weight vector obtained from the synaptic relation among the inputs and active hidden layers of both MLP might also become a secret session key for a specific session (Arindam S and J. K. Mandal. 2012).

Pseudocode for Proposed Model

```
INPUT: A data set D, the set of AES-RSA ∑, the policies set Ψ
OUTPUT: A data set masked and encrypted D*
∑′← REMOVE_KEY_AES-RSA (∑)
Λ ← POLICY_AGGREGATOR (∑′, Ψ)
for each $p_i$ ∈ Λ do
    $X_\zeta$ ← FILTER_BY_IC_ATTRIBUTES (∑′, $p_i$)
    $Z$ ← GET_IC_ATTRIBUTES ($X_\zeta$)
    $X_\zeta$ ← UPDATE_X_SET (Z)
        while BORDERLINE_CASE ($X_\zeta$, D) do
        $Z$ ← ADD_IC_ATTRIBUTE ($X_\zeta$)
        $X_\zeta$ ← UPDATE_X_SET (Z)
end while
Ψ′ ← DEAGGREGATOR_FOR_POLICIES (Λ, Z, $p_i$)
end for
D* ← DATASET_MASKING (D, Ψ′)
```

Algorithm represented above contains the proposed model's pseudo-code. It takes dataset D as input to be masked, the set ∑ of RSA-AES holding on D, and the set Λ of user specific policies, all of which are described related to IC attributes sets, and returns the masked data set D* as output. Step 1 to 2 of the algorithm call the functions POLICY_AGGREGATOR and REMOVE_KEY_RSA-AESs for removing key and group policies defined by other users. The algorithm then executes the following steps for each listed policy: (i) it calculates the associated confidentiality-violating attributes sets (Step 4), (ii) it uses one of the three mentioned heuristics to determine the added features to be encoded (Step 5), and (iii) it checks the existence of borderlines case by upgrading the confidentiality-violating attributes set (Step 6) and iteratively repeating the applications of heuristic before no further borderlines case are found (Step 7 to 10), and (iv) it performs the function DEAGGREGATOR_FOR_POLICIES for mapping the features to be encode to user’s data (Step 11). At last, the masking phase was carried out (Step 13).

The sender encrypts the masked data using the RSA algorithm and the RSA public key to create cipher data. The AES algorithm was utilized to encrypt public key of RSA using the key of the receiver. The encrypted key and cipher data are then sent to the receiver. To decrypt the RSA key, the receiver uses their own private key. Finally, the decrypted AES symmetric key is used by the receiver to translate the cipher data back to the original data.

Performance Evaluation

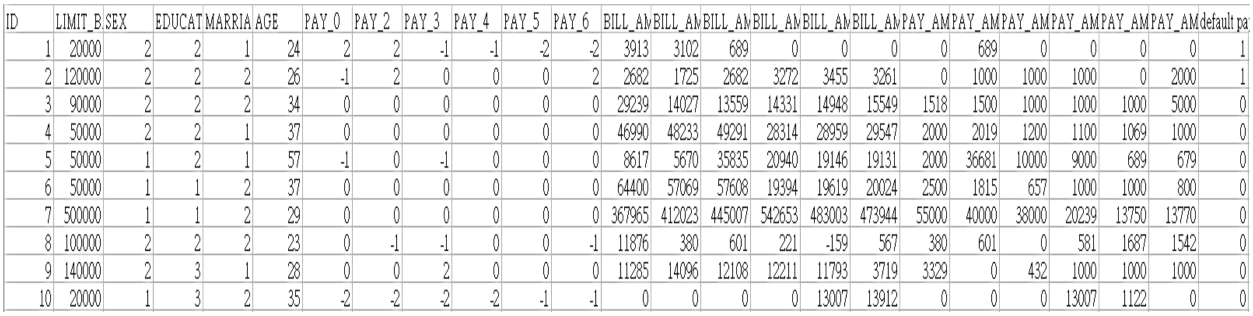
The proposed model is developed and evaluated using PySpark tool on Intel i5core-CPU with 2.3 GHz with Windows 10, 8GB of RAM. PySpark is not a programming language, but rather a Python API created by Apache Spark. It is utilized in Python programming to merge and function with RDD. This enables one to run tasks and computations on massive data and interpret the results.

Dataset Description

This payment data is collected in the form of a credit card client's dataset from the UCI machine learning repository. The dataset contains 30000 observations, 6636 of which are default payment observations, indicating an imbalance between the two groups. The dataset utilized in this research is a list of credit card payment data from a significant bank (a cash and credit card issuer) from Taiwan in October 2005, and the focus were the bank's credit card holders. The following 24 variables are explanatory variables from the dataset:

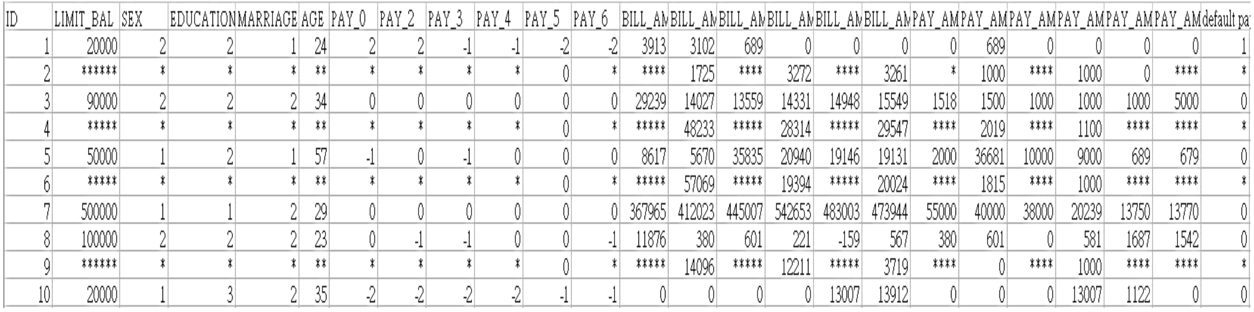
Table 1 Credit Card Dataset Attributes Description [Talha M A et al. 2020]

Attribute ID	Attribute Label	Description
1	Limit_Bal	Amount of the given credits
2	Sex	Gender (1 = male; 2 = female).
3	Education	Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
4	Marriage	Marital status (1 = married; 2 = single; 3 = others).
5	Age	Age (year).
6-11	Pay_1 to Pay_6	History of past payment.
12-17	Bill Amount 1 to 6	Amount of bill statement.
18-23	Pay Amount 1 to 6	Amount of previous payment.
24	Default payment next month	Default=1 and Healthy= 0



ID	LIMIT_B	SEX	EDUCAT	MARRIA	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AM	BILL_AM	BILL_AM	BILL_AM	BILL_AM	BILL_AM	PAY_AM	PAY_AM	PAY_AM	PAY_AM	PAY_AM	PAY_AM	default pa
1	20000	2	2	1	24	2	2	-1	-1	-2	-2	3913	3102	689	0	0	0	0	689	0	0	0	0	1
2	120000	2	2	2	26	-1	2	0	0	0	2	2682	1725	2682	3272	3455	3261	0	1000	1000	1000	0	2000	1
3	90000	2	2	2	34	0	0	0	0	0	0	29239	14027	13559	14331	14948	15549	1518	1500	1000	1000	1000	5000	0
4	50000	2	2	1	37	0	0	0	0	0	0	46990	48233	49291	28314	28959	29547	2000	2019	1200	1100	1069	1000	0
5	50000	1	2	1	57	-1	0	-1	0	0	0	8617	5670	35835	20940	19146	19131	2000	36681	10000	9000	689	679	0
6	50000	1	1	2	37	0	0	0	0	0	0	64400	57069	57608	19394	19619	20024	2500	1815	657	1000	1000	800	0
7	500000	1	1	2	29	0	0	0	0	0	0	367965	412023	445007	542653	483003	473944	55000	40000	38000	20239	13750	13770	0
8	100000	2	2	2	23	0	-1	-1	0	0	-1	11876	380	601	221	-159	567	380	601	0	581	1687	1542	0
9	140000	2	3	1	28	0	0	2	0	0	0	11285	14096	12108	12211	11793	3719	3329	0	432	1000	1000	1000	0
10	20000	1	3	2	35	-2	-2	-2	-2	-1	-1	0	0	0	0	13007	13912	0	0	0	13007	1122	0	0

Figure 6 Input Sample Data



ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AM	BILL_AM	BILL_AM	BILL_AM	BILL_AM	BILL_AM	PAY_AM	PAY_AM	PAY_AM	PAY_AM	PAY_AM	PAY_AM	default pa
1	20000	2	2	1	24	2	2	-1	-1	-2	-2	3913	3102	689	0	0	0	0	689	0	0	0	0	1
2	******	*	*	*	**	*	*	*	*	0	*	****	1725	****	3272	****	3261	*	1000	****	1000	0	****	*
3	90000	2	2	2	34	0	0	0	0	0	0	29239	14027	13559	14331	14948	15549	1518	1500	1000	1000	1000	5000	0
4	*****	*	*	*	**	*	*	*	*	0	*	*****	48233	*****	28314	*****	29547	****	2019	****	1100	****	****	*
5	50000	1	2	1	57	-1	0	-1	0	0	0	8617	5670	35835	20940	19146	19131	2000	36681	10000	9000	689	679	0
6	*****	*	*	*	**	*	*	*	*	0	*	*****	57069	*****	19394	*****	20024	****	1815	****	1000	****	****	*
7	500000	1	1	2	29	0	0	0	0	0	0	367965	412023	445007	542653	483003	473944	55000	40000	38000	20239	13750	13770	0
8	100000	2	2	2	23	0	-1	-1	0	0	-1	11876	380	601	221	-159	567	380	601	0	581	1687	1542	0
9	******	*	*	*	**	*	*	*	*	0	*	*****	14096	*****	12211	*****	3719	****	0	****	1000	****	****	*
10	20000	1	3	2	35	-2	-2	-2	-2	-1	-1	0	0	0	0	13007	13912	0	0	0	13007	1122	0	0

Figure 7 Masked Input Data

Figure 6 represents the sample input data from the dataset used in this research. The total attributes contained in this dataset are nine with 24 attribute ID's as shown in table 1. For this experiment, 18 attribute IDs were masked and encrypted as shown in Figure 7. Figure 7 represents the masked dataset as an outcome of the algorithm's implementation to the dataset. A module was built that randomly allocated secret properties to each user tuple, with the goal of simulating the development of users' policies. It is feasible to see that towards the end of the proposed methodology's implementation, only a few variables are masked and encrypted, while many more remain open. Furthermore, it is worth noting that several variations between the encrypted values are acquired. Thus, by enhancing the options for performing data analytic operations, this technique allows for a reduction in the total values to be masked and encrypted in order to maintain information confidentiality.

Table 2 Information Confidentiality Frequency and Count of Data

Client ID	IC Frequency	IC Count
1	9	88
2	9	33
3	9	52
4	6	8
5	3	66
6	2	43
7	3	20
8	3	71
9	3	19
10	7	95

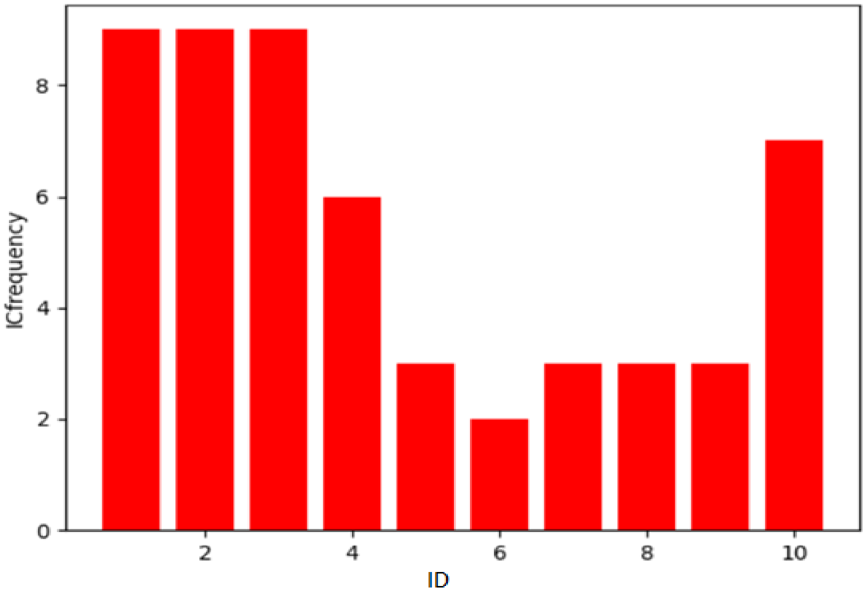


Figure 8 Graphical Plot of ID's IC-Frequency

The IC-count stands for Information Confidentiality Count and IC-frequency stands for Information Confidentiality frequency. The IC count and frequency are both measured using the client's ID as shown in table.2. Figures 8 and 9 represents the information confidentiality evaluation based on counting and weighted counting of the attributes in the dataset.

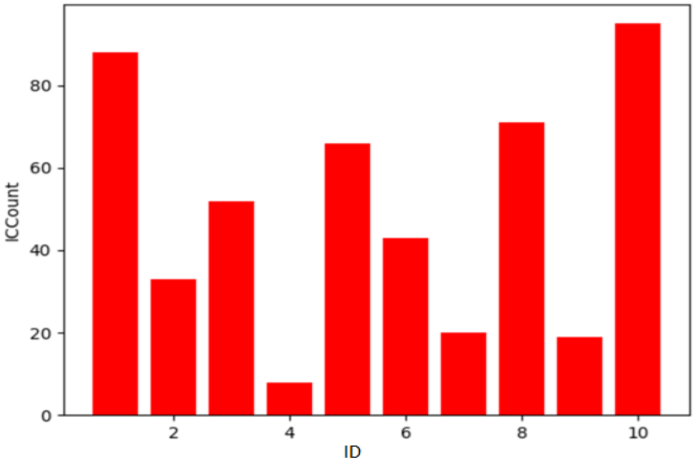


Figure 9 Graphical Plot of ID's IC-Count

The computational time of masking and encryption and decryption process of the data from the dataset are estimated as shown in table 3. The accuracy of those data was also estimated based on the decrypted data. Figure 10 shows the comparison of encryption and decryption computational time of the data. Figure 11 represents the accuracy of the decrypted data.

Table 3 Performance Analysis of Data

Client ID	Masking and Encryption Time	Decryption Time	Accuracy
1	0.109148	0.062497	0.9566
2	0.046879	0.063854	0.9968
3	0.109386	0.114792	0.922
4	0.109378	0.109401	0.904
5	0.031253	0.031252	0.9640
6	0.031252	0.046878	0.9314
7	0.07813	0.07813	0.9325
8	0.109813	0.07813	0.9938
9	0.031266	0.031252	0.9342
10	0.109382	0.125009	0.9873

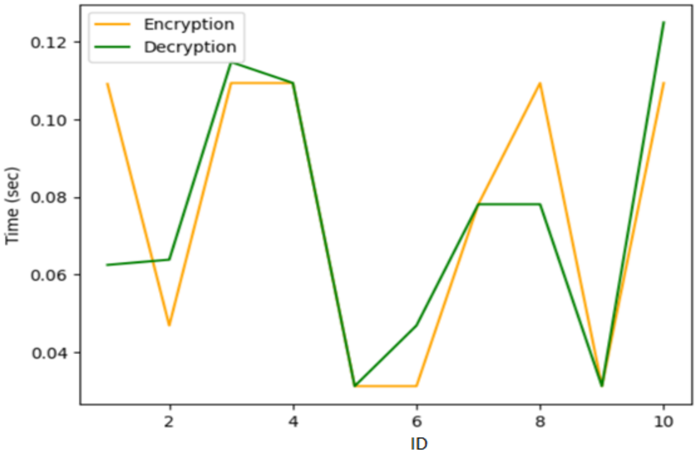


Figure 10 Graphical Plot of Encryption and Decryption Time

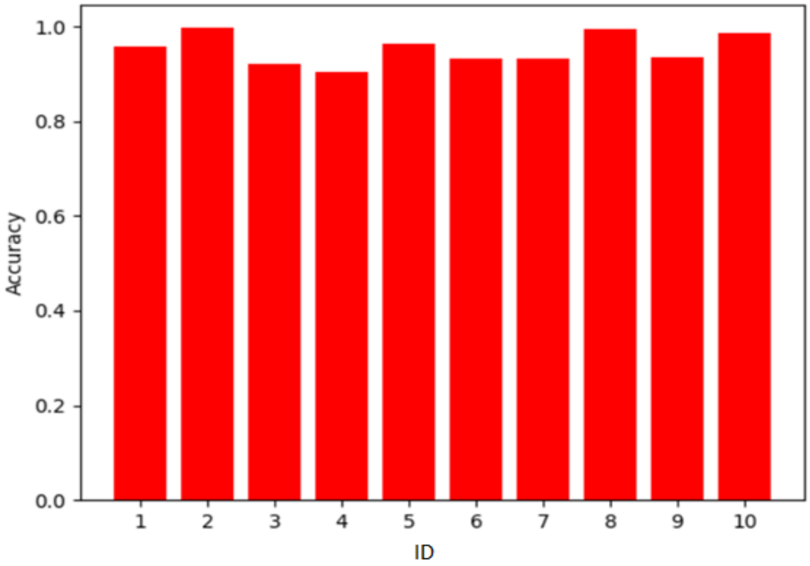


Figure 11 Graphical Plot of Accuracy

Table 4 shows the comparison of performance analysis of the proposed method with other existing cryptographic methods. The overall accuracy of these methods and average computational time of these methods are estimated. Figure 12 represents comparison of both the accuracy and computational time performance obtained by the methods.

Table 4 Comparison of Performance Analysis

Methods	Overall Accuracy	Average Computational Time (nano secs)
3DES-RSA	89.93	900
Blowfish-RSA	91.50	750
Proposed	95.23	300

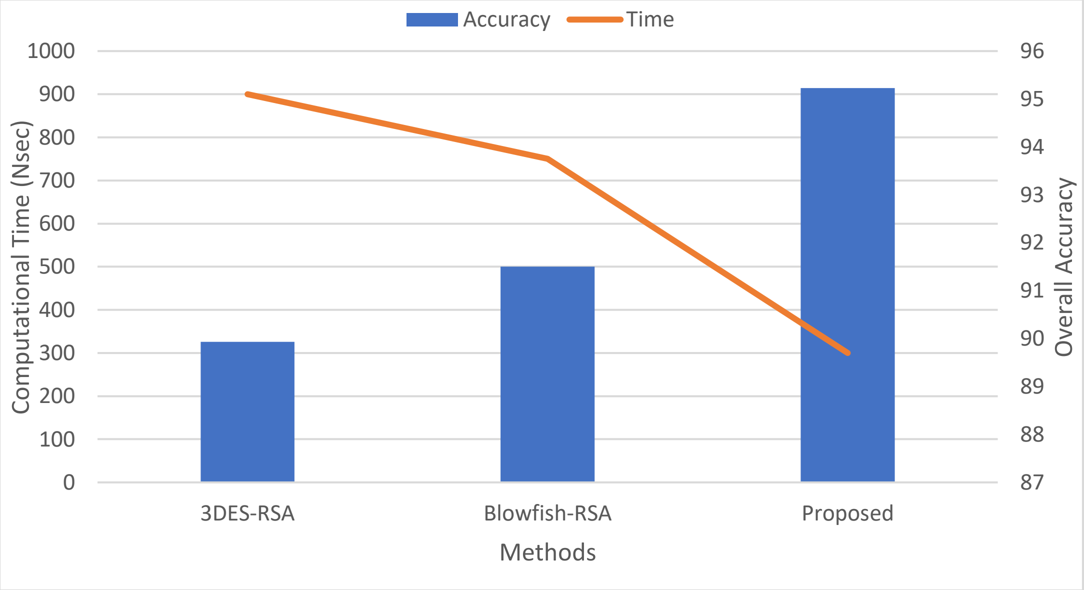


Figure 12 Graphical Plot of Performance Analysis Comparison

It can be concluded that proposed method influenced the number of attributes to be masked and encrypted. Because more attribute correlations are formed, DFS often increases the count of attributes to be masked and encrypted, as predicted, but this also results in improved information confidentiality protection. The results, in particular, illustrate the trade-off among the quantity of encryption and the degree of information confidentiality preservation that the proposed method could achieve.

Conclusion

In this research, a hybrid cryptographic model based on AES and RSA was proposed to identify and mask the sensitive information for detecting major threats to information confidentiality. A hybrid cryptography algorithm in the context of masking was proposed to effectively transfer big data through the cloud. The hybrid algorithm was created by combining more than one algorithm. This algorithm enabled the user to select the data to be masked and encrypted. For protecting data stored in the cloud, the proposed hybrid algorithm includes RSA and AES. Along with these algorithms, the multilayer perceptron neural network was used for key generation and key exchange. A credit card client's dataset from the UCI machine learning repository was used for the evaluation. From the data set the sensitive attributes were selected using the depth first search (DFS) technique. The sender encrypts the data using the RSA algorithm and a key to create masked data using public key. The AES algorithm was used to encrypt the RSA key. The encrypted key and masked data were then sent to the receiver. To decrypt the RSA key, the receiver uses the AES decryption. Finally, the decrypted RSA key was used by the receiver to translate the masked data back to the original data with private key. The proposed model obtained the overall accuracy of 95.23% accuracy and an average computational time of 300 nano secs. The proposed model was compared with other methods like triple DES with RSA and Blowfish with RSA techniques, where the proposed model outperformed these techniques. In future, the propose model can be improved by integrating the functional dependencies approach and privacy preserving approach.

References

Guru, M.A., & Ambhaikar, A. (2021). AES and RSA-based Hybrid Algorithms for Message Encryption & Decryption. *Information Technology in Industry*, *9*(1), 273-279.

Arindam, S., & Mandal, J.K. (2012). Multilayer Perceptron Guided Key Generation Through Mutation with Recursive Replacement in Wireless Communication (MLPKG). *International Journal on AdHoc Networking Systems*, *2*(3), 11-28.

Divya, D., & Santhi, G. (2021). Secured Multi-Party Data Release on Cloud for Big Data Privacy-Preserving Using Fusion Learning. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12*(3), 4716-4725.

Hababeh, I., Gharaibeh, A., Nofal, S., & Khalil, I. (2018). An integrated methodology for big data classification and security for improving cloud systems data mobility. *IEEE Access*, *7*, 9153-9163.

Kalaiselvi, R.C., & Vennila, S.M. (2021). Custom-Aes: A Novel Framework to Enhance Data Security in Cloud Environment. *International Journal of Future Generation Communication and Networking*, *14*(1), 314-323.

Caruccio, L., Desiato, D., Polese, G., & Tortora, G. (2020). GDPR compliant information confidentiality preservation in big data processing. *IEEE Access*, *8*, 205034-205050.

Matthias, D., Osakwe, B.P., & Anireh, V.I.E. (2021). A Secure Model on Cloud using a Modified Rivest, Shamir and Adleman Algorithm along with Gray Codes. *International Journal of Computers & Technology*, *8*(1), 207-214.

Gruschka, N., Mavroeidis, V., Vishi, K., & Jensen, M. (2018). Privacy issues and data protection in big data: a case study analysis under GDPR. *In IEEE International Conference on Big Data (Big Data)*, 5027-5033.

Yang, P., Xiong, N., & Ren, J. (2020). Data security and privacy protection for cloud storage: A survey. *IEEE Access*, *8*, 131723-131740.

Jain, P., Gyanchandani, M., & Khare, N. (2016). Big data privacy: a technological perspective and review. *Journal of Big Data*, *3*(1), 1-25.

Singh, R.G., & Ruj, S. (2020). A Technical Look at the Indian Personal Data Protection Bill. *arXiv preprint arXiv:2005.13812*.

Jahan, R., Suman, P., & Singh, D.K. (2021). An Algorithm to Secure Data for Cloud Storage. *Information Technology in Industry*, *9*(1), 1382-1387.

Kuswaha, S., Waghmare, S., & Choudhary, P. (2015). Data Transmission using AES-RSA Based Hybrid Security Algorithms. *International Journal on Recent and Innovation Trends in Computing and Communication*, *3*(4), 1964-1969.

Alam, T.M., Shaukat, K., Hameed, I.A., Luo, S., Sarwar, M.U., Shabbir, S., & Khushi, M. (2020). An investigation of credit card default prediction in the imbalanced datasets. *IEEE Access*, *8*, 201173-201198.

Malgari, V., Dugyala, R., & Kumar, A. (2019). A Novel Data Security Framework in Distributed Cloud Computing. *In Fifth International Conference on Image Information Processing (ICIIP)*, 373-378.

Viswanath, G., & Krishna, P.V. (2021). Hybrid encryption framework for securing big data storage in multi-cloud environment. *Evolutionary Intelligence*, *14*(2), 691-698.

Essa, Y.M., Hemdan, E.E.D., El-Mahalawy, A., Attiya, G., & El-Sayed, A. (2019). IFHDS: intelligent framework for securing healthcare bigdata. *Journal of medical systems*, *43*(5), 1-13. https://doi.org/10.1007/s10916-019-1250-4.

Gao, Y., Chen, X., & Du, X. (2020). A big data provenance model for data security supervision based on PROV-DM model. *IEEE Access*, *8*, 38742-38752.

Zeraatkar, A., Mirvaziri, H., & Ahsaee, M.G. (2019). Improvement of page ranking algorithm by negative score of spam pages. *Webology*, *16*(2), 43-56.