Enhanced Security for Personal Information using Hybrid Techniques

Sritam Mishra (20BCI0112) *Student, Vellore Institute of Technology,Vellore, TamilNadu,India* (sritam.mishra2020@vitstudent.ac.in, 7077407001)

Kunal Vijay (20BCI0122) *Student, Vellore Institute of Technology,Vellore, TamilNadu,India* (kunal.vijay2020@vitstudent.ac.in ,8178050765)

Karthik Srinivas B (20BCI0124) *Student, Vellore Institute of Technology,Vellore, TamilNadu,India* (karthiksrinivas.b2020@vitstudent.ac.in , 6302539984)

Konark Patel (20BCI0169) *Student, Vellore Institute of Technology,Vellore, TamilNadu,India* (konarkvinodbhai.patel2020@vitstudent.ac.in , 8849492455)

Under Faculty: Dr. Anbarasi M

Senior Assistant Professor,

School of Computer Science and Engineering,

Vellore Institute of technology,Vellore,Tamil Nadu,India

(manbarasi@vit.ac.in,9488054290)

Abstract:

Data masking is the method of protection of sensitive data from hackers (data snoopers). It creates an anonymized version of data without tampering with the format of the data present. The main goal is to transform such that the original data cannot be retrieved and protecting the individuals for their data against the "re-identification". Our aim is to give a uniform data masking architecture for the sensitive data like business, finance, etc. In this paper, we study the comparison of different data masking technique on business data. The conclusion provides the replacement of the data using different masking technique in business data. Despite outsourcing third-party capabilities, the company must nevertheless access to its data. Therefore, it is crucial for business to retain data authenticity even while disguising the data. Data masking is a technique used by corporations to decrease human mistake that could jeopardize data security. Data masking therefore reduces the possibility of such errors. We have successfully used the different masking technique like shuffling, encryption, nulling for the replacement data to safety for the third-party and the result strongly provides the order of security using data masking technique.

Keywords: Re-identification, Encryption, Nulling, Shuffling, Substitution, Cryptosystem.

Introduction:

In today's digital era, every single piece of data has a cost. With the advancement in technology and greater storage devices, the amount of data generated has vastly increased, including personal information such as contact and address details, credit card information, etc. If this data ends up in the wrong hands, it could be exploited improperly. Security is therefore required for such sensitive data. [1] Within the period of Jan 2005 and Nov 2017, there were around 8,000+ data breaches, and even more, increased exponentially in recent years. Data Masking comes to application here. It provides protection to digital sensitive data from breaches and other threats[16]. To provide the required security, a variety of data masking techniques are employed, which is the process of hiding particular data components in data repositories. It guarantees that personal data is swapped out for fake but realistic data. The aim of data masking is to produce the same data format but modify the data's values, making a version that cannot be cracked or reverse-engineered.

[1] Data masking is a crucial technique used to protect sensitive data from unauthorised access or theft. [2] The main objective of data masking is to create an anonymized version of the data without altering its format. [3] This process transforms the original data in such a way that it cannot be retrieved, thus protecting the privacy of individuals and their data. [4] The anonymized data can be further secured using encryption techniques to prevent re-identification and unauthorised access. [7][8][13] A data masking model for applications use heterogeneous large data, including texts, photos, sounds, and databases, as well as four essential modules. [9][17] For example, data masking can be used in attendance verification and ensure the confidentiality of student data from being viewed by unauthorised parties. [11][17][19] It is also helpful in healthcare as data masking module based on the key-based reversible approach can be used to protect patients' data privacy, while maintaining the data utility to meet the need for data analytics within BI platforms of the healthcare environment.

[5][18][24] Data substitution, shuffling and nulling are all techniques used to protect sensitive information from unauthorized access or disclosure. [1] Corporations use these techniques to reduce the risk of human errors that could compromise data security. [4] These techniques minimize the possibility of errors and helps to ensure the safety of third-party data. [2][12][24] Data substitution is the process of disguising or obfuscating sensitive data with fake or altered data while preserving its original format and structure. This technique is often used in test environments where sensitive data is required to perform tests, but the actual data cannot be exposed due to security or privacy concerns. For example, this technique can be used to replace real names with pseudonyms, or real credit card numbers with fake ones. Shuffling is a technique that randomizes the order of data values in a dataset. This technique is often used to reduce the risk of exposing sensitive information when multiple datasets are combined or aggregated. For example, shuffling can be used to scramble the order of medical records to ensure that a particular patient's data is not easily identifiable. Nulling is the process of replacing sensitive data with null or empty values. This technique is often used when it is not necessary to preserve the original data format or structure. For example, nulling can be used to replace a social security number with an empty value in a data set that is used for research purposes.

Encryption is the process of converting sensitive data into a coded or encrypted format that can only be read with the correct decryption key. This technique is often used to protect data in transit or at rest. For example, encryption can be used to secure credit card transactions during online purchases or to protect sensitive documents stored on a computer. [15][20][23][6] Hybrid Encryption provides the best security as it combines multiple encryption technique making the data encrypted robust. The symmetric and asymmetric encryption cryptosystems are the two main types of encryption algorithms used by the hybrid cryptosystem. There are many benefits of using hybrid cryptography for data security in various fields, including healthcare, finance, and e-commerce [10]. For instance, researches on privacy preserving techniques for deep learning on Machine Learning as a Service (MLaaS) platforms are taking an uplift [20]. These privacy preserving techniques can be categorised in 3- encryption-based, perturbation-based and hybrid. Since, it is still in study phase, these techniques are not yet much capable. However, the data being an essential part of Machine Learning testing and

training, it prioritizes the security of data. We see the same as authors describe on the importance of privacy in machine learning and the potential risks associated with sharing sensitive data for model training in [21] related to privacy like model inversion attacks, membership inference attacks and so. Homomorphic and Secure Multi-Party computation are two of techniques that can be used for privacy preserving in this scenario, the authors evaluate the effectiveness of the same. In [21], the authors emphasize the need for a holistic approach to privacy in machine learning, which should involve not only the application of privacy-preserving techniques but also the establishment of privacy policies, the implementation of data access controls, and the monitoring of data usage. In the paper [9], RSA and DES were the algorithms that were used. However, we have found a few flaws in the method. We noticed the following issues with this algorithm. The 56-bit key size is the biggest defect of DES because it takes a long time to search the entire key space and runs relatively slow. DES can also be easily broken and so are 2DES and 3DES. Therefore, for encryption, we decided to implement the symmetric Advanced Encryption Standard (AES) and Rivest-Shamir-RSA Adleman's algorithm for encryption for data masking after observing many combinations tried with and used for Hybrid Cryptosystem.

In this paper, we compare data masking techniques and then evaluate their effectiveness in preserving data utility while also ensuring data privacy. The most effective technique is selected for a given dataset, and the sensitive data is masked using that technique. To enhance the security of the masked data, encryption is applied using Hybrid Encryption to prevent unauthorized access. The resulting encrypted data is much more secure as it cannot be read by unauthorized users without the appropriate decryption key.

Proposed Workflow

The increasing amount of sensitive data being collected and stored by organizations has led to an increase in the need for data masking techniques to protect the privacy and security of individuals. However, with the variety of data masking techniques available, it can be challenging to determine the most effective approach.

In this paper, we compare several popular data masking techniques and evaluate their effectiveness based on their ability to preserve data utility while also ensuring data privacy. After comparing the various techniques, we select the most effective one for the given dataset and use it to mask sensitive data. To further enhance the security of the masked data, we then apply encryption using symmetric or asymmetric algorithms. The resulting encrypted data is much more secure, as it is rendered unreadable to unauthorized users without the appropriate decryption key.

Overall, our study provides valuable insights into the selection and application of data masking techniques to ensure both data privacy and utility, while also demonstrating the importance of additional security measures such as encryption to enhance the overall security of sensitive data. The following flowchart summarizes the workflow of our study.

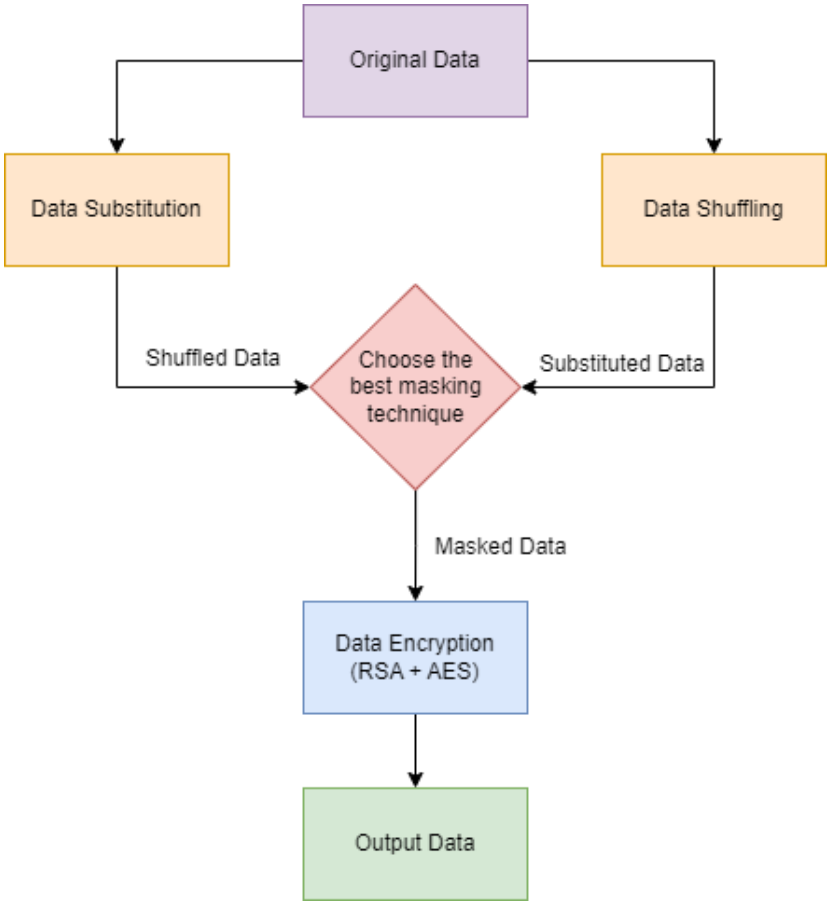

Figure 1.1: Workflow Flowchart

Methodology

I. Data Substitution:

Data substitution is the process of replacing one value in the dataset with another. The mask, for instance, may replace a student's name in a dataset with names drawn at random from a phone book entry. However, unless you have access to the original substitution table, the resulting dataset does not logically relate to the actual name that was originally used. In this method, fake but plausible alternative values are used in place of the original data values. It is one of the best data masking techniques that maintains the data's original look and feel. Although sometimes quite challenging to carry out, it is an extremely effective method of preventing data breaches.

For example,

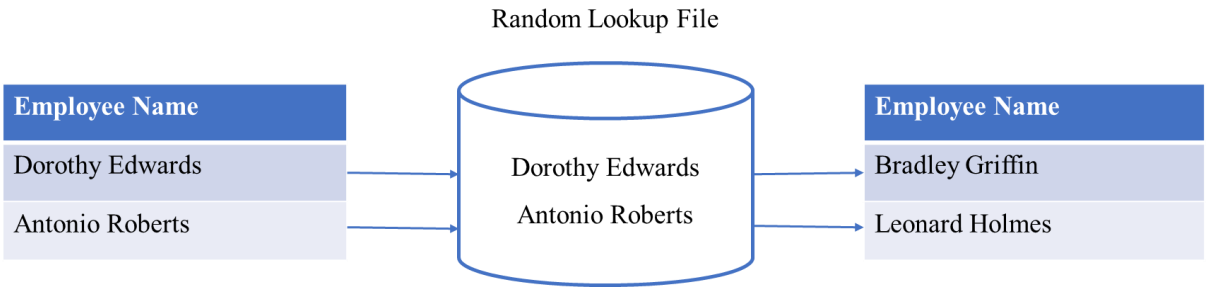

Figure 1.1: Data Substitution: Replacing the data in table 1 with lookup table values in table 2

II. Data Shuffling:

Similar to data substitution, data shuffling involves switching data values inside a single dataset. Each column's data is rearranged using a random sequence. Changing original employee names, for instance, between different employee records. Although the output set appears to be real data, it does not provide accurate details for each person or a data entry. Although the output shuffled data appears to be accurate, no actual personal information is disclosed. Data shuffling can be mainly used to protect sensitive financial or healthcare data, such as credit card numbers, social security numbers, and medical records. By replacing this information with statistically similar data, companies can still analyze and use the data for business purposes without putting individuals' privacy at risk. For example,

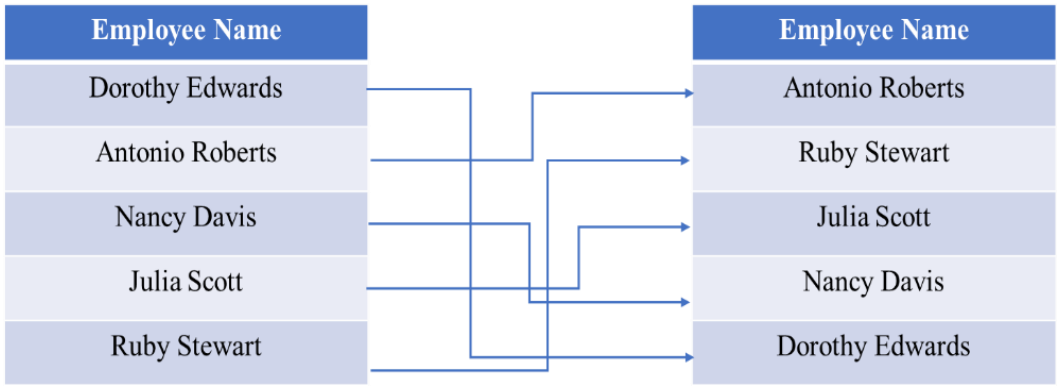

Figure 1.2: Data Shuffling: Shuffling the data in the table to protect sensitive information

III. Nulling

This method replaces the sensitive data in the dataset with null values. Simply put, a generic value, like X, is replaced with sensitive data in this technique. For instance, we may use XXX-XXX-XXXX as a substitute for a phone number. This method is the quickest and easiest type of data masking. However, this method renders the data less helpful for research and testing. However, nulling has limitations and cannot be used for certain types of sensitive information, such as names. In the case of names, nulling removes all the information, making it impossible to identify a person, which is not desirable in most business cases. For example, in healthcare, it's important to know which patient has received which treatment, and nulling patient names would make it difficult to track the patient's medical history.

For example,

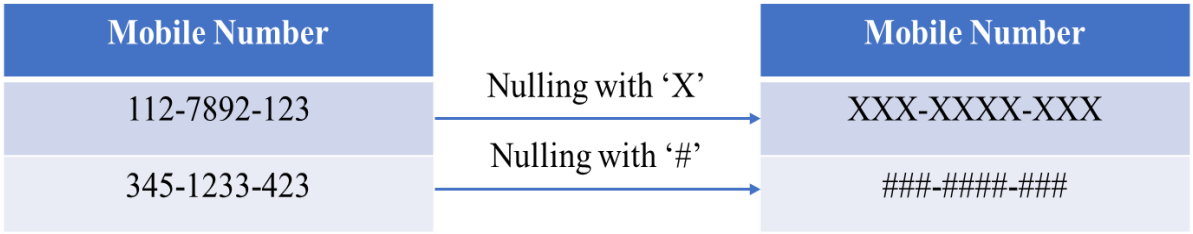

*Figure 1.3: Nulling: Masking the values in table 1 with * and # for privacy preservation*

IV. Data Encryption

The art or science encompassing the principles and methods of transforming an intelligible message into one that is unintelligible is called encryption. Encryption or enciphering is the process of transforming original data into a form where data is in an unreadable format to an external reader. But this format is reversible as and when required. Only the ones who have the solution or the transformation key can get access to the original data by decrypting the same- the reverse process. Encryption as a data masking technique provides comprehensive data protection, including confidentiality, security, compliance, flexibility, and integrity, ensuring sensitive data is secure and accessible only to authorized users. By encrypting sensitive data, companies can protect against data breaches and comply with data privacy regulations, while maintaining the integrity and utility of their data. For example,

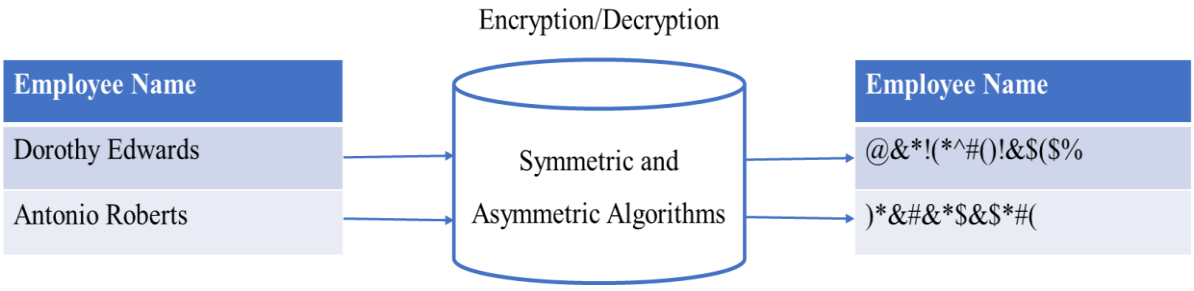

Figure 1.2: Data Encryption: Encrypting the data in the table using symmetric and asymmetric algorithms

For Encryption, we implemented Format preserving encryption, where, most encryption algorithms produce strings of arbitrary length, format-preserving encryption transforms the data into an unreadable state while retaining the format (overall appearance) of the original values.

The novelty of our implementation in encryption comes when we implemented a Hybrid cryptography algorithm, instead of a classic, one-process encryption. We propose to use both the symmetric (AES) and asymmetric algorithms (RSA) for data encryption. Symmetric encryption uses a single key for encrypting and decrypting data, while asymmetric encryption uses two keys for increased security. The Hybrid cryptography algorithm is as follows:

i. Advanced Encryption Standard (AES)

- The AES algorithm is a symmetric block cipher algorithm. The AES algorithm is considered secure, it is the worldwide standard.
- AES has 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key. Each Round performs certain functions before moving on to the next round.
- Each Round has 4 Functions to perform.
- These rounds include Byte Substitution, Shifting Rows, Mixing the Columns (Not for the Final Round), and Adding Round Key.
- Inverse of functions in round function happens while decryption.

Working of AES Rounds:

Step 1: BYTE SUBSTITUTION

In this step, each byte is substituted by another byte. It is performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a complement of the current byte. The result of this step is a 16-byte (4 x 4) matrix like before.

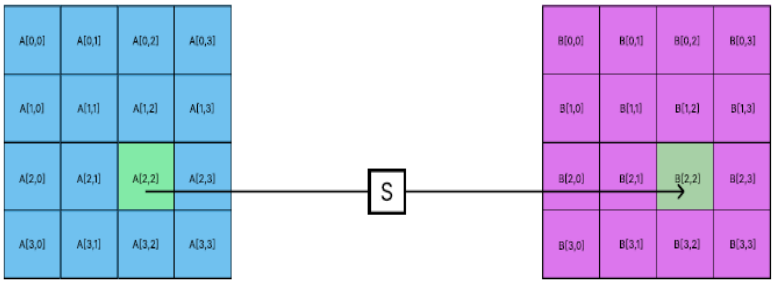

Figure 2.1: Byte Substitution

Step 2: SHIFTING ROWS:

This step is just as it sounds. Each row is shifted a particular number of times. The first row is not shifted. The second row is shifted once to the left. The third row is shifted twice to the left. The fourth row is shifted thrice to the left.

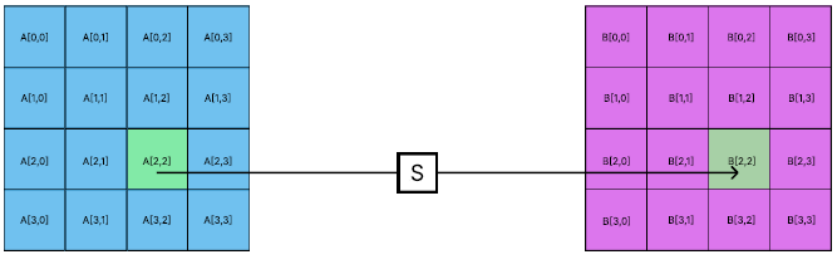

Figure 2.2: Shifting Rows

Step 3: MIXING COLUMNS:

This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result. This step is skipped in the last round.

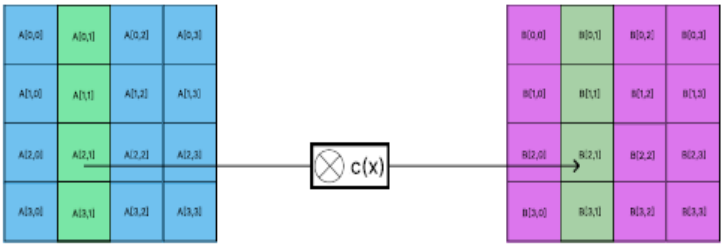

Figure 2.3: Mixing Columns

Step 4: ADD ROUND KEY:

Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes are not considered as a grid but just as 128 bits of data.

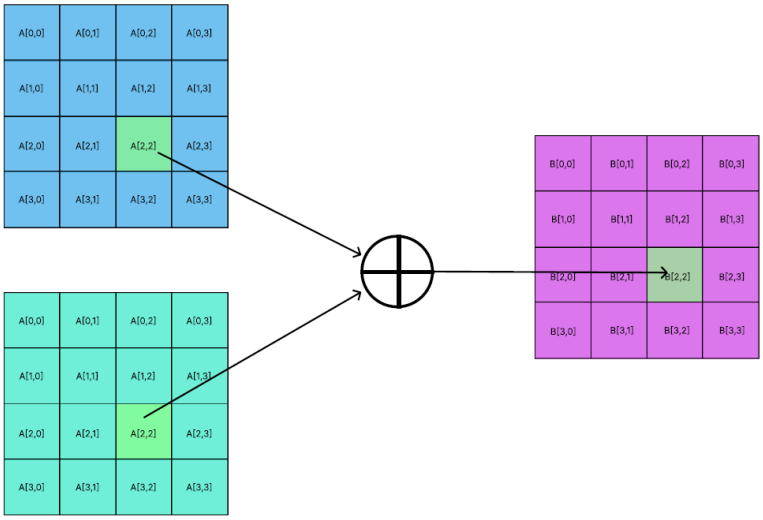

Figure 2.3: Add Round Key

ii. <u>RSA Algorithm:</u>

- The RSA algorithm is an asymmetric cryptography algorithm; this means that it uses a *public* key and a *private* key
- RSA is considered as the best asymmetric algorithm due to cost of factoring large numbers.
- It has mainly 3 stages.

i. Choosing & Generating Keys.

choose two large prime numbers x & y, e is *Public Key*. where e is ***co-prime*** to $\phi(n)$ and $1<e<\phi(n)$.

$$n = x \times y$$

$$\phi(n)=(x-1)(y-1)$$

$$e.d=1 \bmod \phi(n)$$

(d,n) is *Private Key*.

ii. Encryption

$$C=P^e \bmod n, \text{ C is Cipher}$$

TextP is Plain Text

iii. Decryption

$$P=C^d \bmod n$$

C is Cipher TextP is Plain Text

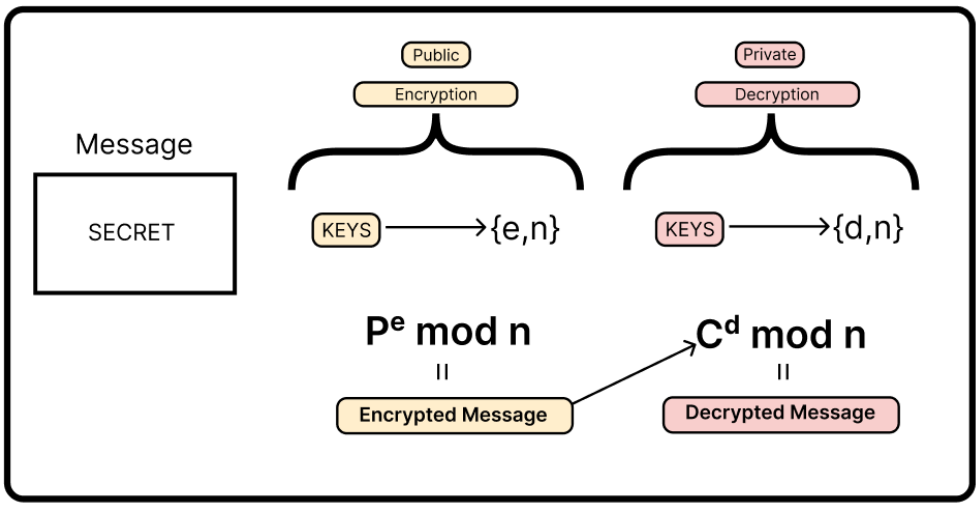

Figure 3.1: RSA Algorithm Example

Modular Design of Hybrid Cryptography

The flowcharts presented in the figures below illustrates the steps involved in the encryption and decryption process of our hybrid encryption algorithm that incorporates both AES and RSA. This algorithm has been designed to ensure the security and confidentiality of sensitive data.

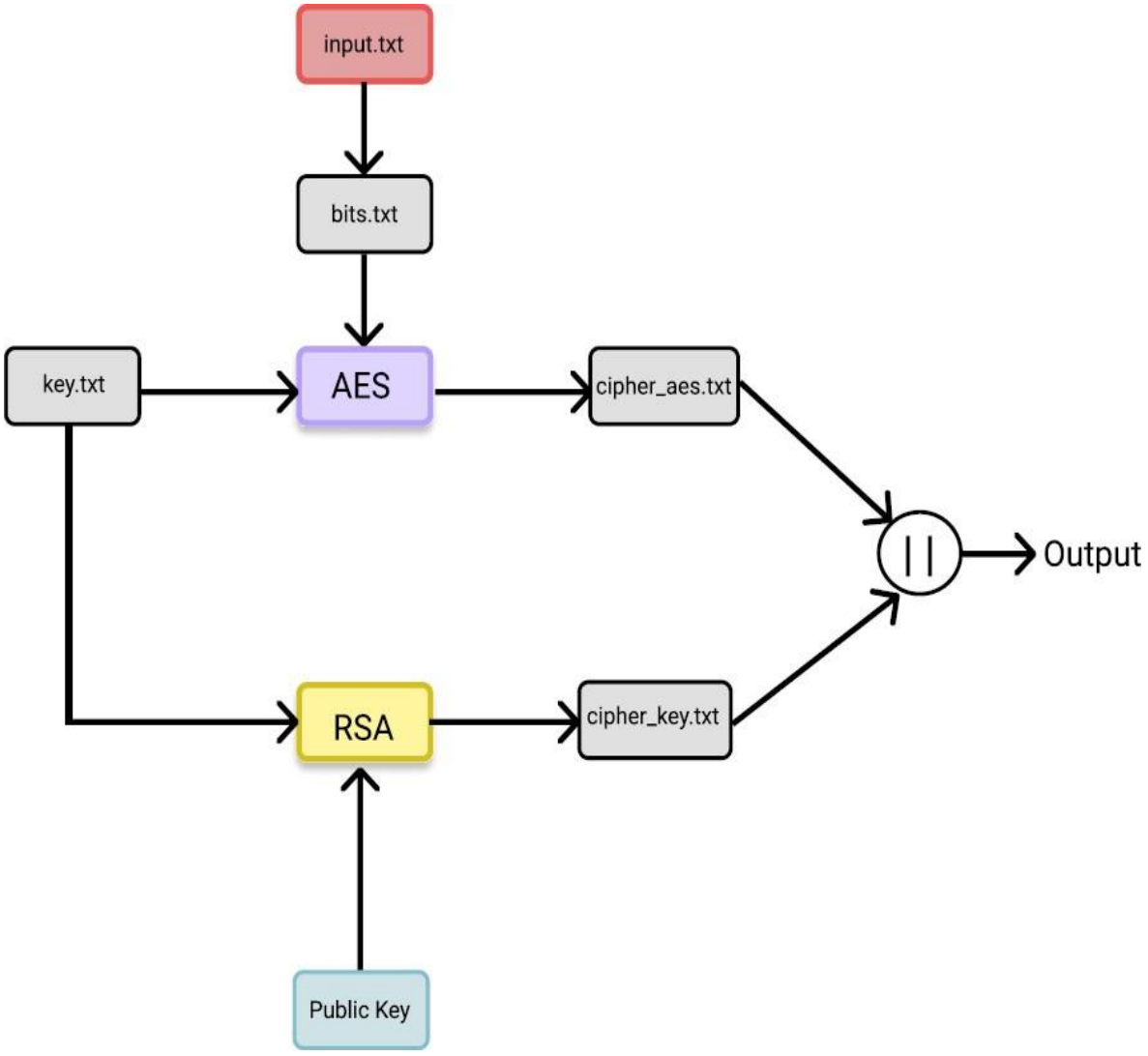

Figure 3.2: Sample Encryption of AES+RSA on file data

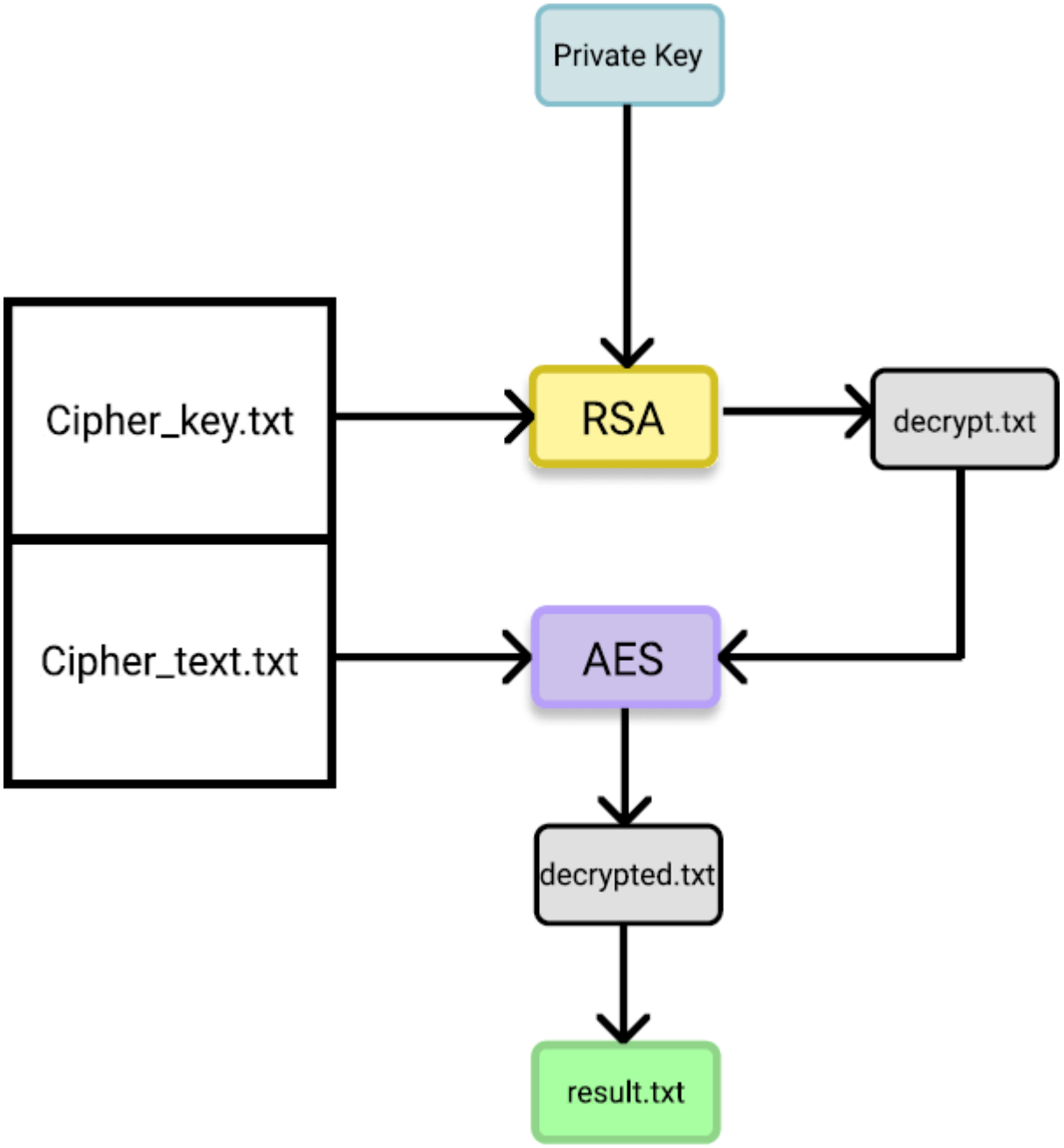

Figure 3.3: Sample Decryption of AES+RSA on the encrypted file

AES Module Pseudocode:

```
AESEncrypt(plstr,expandedKey,encryptedMessage)
BEGIN
        CHAR state=INPUT;
        INT Number_of_Rounds;
        FOR round=1 STEP 1 to (Number_of_Rounds-1)
                BytesSubstitution(state); // S-Box
        Transformation
                ShiftRows(state); // Shift Rows Left
                MixColumns(state); // Column Transformation
                AddRoundKey(state,key); // XOR with
        Expanded Key
        END FOR

        // Final Round
        BytesSubstitution(state);
        ShiftRows(state);
        AddRoundKey(state,key);
        OUTPUT=state;
END
```

```
AESDecrypt(encryptedMessage,expandedKey,decryptedMessage)
BEGIN
        CHAR state=INPUT;
        INT Number_of_Rounds;
        FOR round=Number_of_Rounds STEP -1 downto 1
                SubRoundKey(state,key); // XOR with Expanded
                Key InverseMixColumns(state); // Reverses Mix
                Columns
                ShiftRows(state); // Shift Rows Right
                BytesSubstitution(state); // Inverse S-Box
                Transformation
        END FOR
        // Initial Round
        SubRoundKey(state,key);
        ShiftRows(state);
        BytesSubstitution(state);
        OUTPUT=state;
END
```

RSA Module Algorithm:

```
factors(a):
   for i in range: [1,a+1]:
      if(a mod i is 0):
         add i to l1

   for i in 1 to a+1:
      for j in range: [1,i]:
         if( i mod j ==0):
            append i to l2

      if(l1 intersection l2 ={1} && i!=2 && a mod i!=0)
        add i to coprimes list

   return coprime[1]

private_key(e,n):
   for i in range:[1,n]:
      if ((e*i) mod n is 1):
         return i

RSA_encrypt(e,n,plain_text):
   cipher_text=(plain_text^e)mod n
   return cipher_text

p,q= 2 //PRIME NUMBERS: INPUT FROM USER
n=p*q
function_n=(p-1)*(q-1)
e=factors(function_n)
f= open(keyfile.txt,read_mode)
plain_text= f.read()
//convert plain_text to respective decimal values and store in
plain_text_list

for i in range:[0,plain_text_list.size]:
   cipher_text.add(RSA_encrypt(e,n,plain_text_list[i]))
//write into Key_Cipher.txt
```

```
//factors(a):

private_key(e,n):
   for i in range:[1,n]:
      if ((e*i) mod n ==1):
         return i


RSA_decrypt(e,n,plain_text):
   plain_text=(cipher_text^d)mod n
   return plain_text

e,n,function_n=public key: INPUT FROM USER
d=private_key(e,function_n)
f= open(Key_Cipher.txt,read_mode)
cipher_text= f.read()
cipher_text=cipher_text.split()
cipher_text_list=[]
for i in range:[0,plain_text_list.size]:
   cipher_text_list.add(RSA_decrypt(e,n,plain_text_list[i]))
//combine decrypted plain_text from cipher_text_list and write into
 decrypt.txt
```

Result and Analysis

Selecting a proper data masking technique is very important for an organization as it ensures that sensitive data is protected during non-production use cases, such as software development and testing. If a weak data masking technique is used, there is a risk that the sensitive data can be compromised, which can lead to significant financial and reputational damage to an organization.

The data masking technique that must be chosen depends on several factors, including the specific use case, the level of privacy protection required, and the impact on data utility. The level of privacy protection required will determine the degree of masking that is necessary while the level of data utility required will determine the degree of masking that is acceptable. Therefore, it is important to evaluate the pros and cons of each technique and choose the one that best meets the specific needs and constraints of the organization and also provides high utility and high privacy.

As we have seen that nulling is a useful data masking technique for some types of sensitive information, but it cannot be used for all types of data. When dealing with sensitive data such as names, data substitution or data shuffling are more appropriate techniques that balance data privacy and data utility.

I. DATA SUBSTITUTION

Data substitution can provide high privacy protection by replacing sensitive data with alternative values but the actual utility of the substituted data may be limited. It provides high privacy protection thus reducing the risk of re-identification and potential privacy breaches. The substituted data is not real, so it may not accurately reflect the underlying relationships and patterns in the original data. This can affect the accuracy and usefulness of the data for research and analysis purposes, potentially reducing its utility.

To perform data substitution, we imported the "anonymize" function from the "anonymizedf" library and the "Faker" class from the "faker" library, which are used for data anonymization and generating realistic fake data respectively. Then, we read a CSV file named "empdata.csv" and created a Pandas DataFrame called "df" using the "read_csv" function. We generated a new DataFrame called "anon" by applying the "anonymize" function to "df" using the "Faker" library to generate fictitious data for sensitive information. Finally, we replaced the original names in the "Employee_name" column of "anon" with fake names generated by the "Faker" library, resulting in a statistically similar DataFrame that protects sensitive information.

Employee Data (Name) before Data Substitution

	Emp ID	Employee_name
0	677509	Lois Walker
1	940761	Brenda Robinson
2	428945	Joe Robinson
3	408351	Diane Evans
4	193819	Benjamin Russell
...	...	...
95	639892	Jose Hill
96	704709	Harold Nelson
97	461593	Nicole Ward
98	392491	Theresa Murphy
99	495141	Tammy Young

Employee Data (Name) after Data Substitution

	Emp ID	Fake_Employee_name
0	677509	Ms Jemma Curtis
1	940761	Mr Jay Clark
2	428945	Adrian Taylor
3	408351	Kelly Powell
4	193819	Dr Cameron Bailey
...	...	...
95	639892	Dr Daniel Clark
96	704709	James May-Richardson
97	461593	Trevor Watkins
98	392491	Wayne Thompson
99	495141	Jenna Nixon

Figure 4.1: The figure displays two columns, one containing the original employee data and the other containing the fake employee data after performing data substitution.

II. DATA SHUFFLING

In order to balance privacy protection and data utility, the data shuffling technique can be used. The method involves randomly rearranging the data within a data set, preserving the relationships and associations between variables, but not linking the data to individual records. This can reduce the risk of re-identification, providing high privacy protection. At the same time, the data remains usable for analysis, providing high utility. The level of privacy protection and data utility provided by data shuffling depends on the specific use case and the degree of shuffling applied. A higher degree of shuffling can increase privacy protection but decrease data utility, while a lower degree of shuffling can increase data utility but decrease privacy protection. Therefore, data shuffling appears to be a more effective technique for providing high utility and high privacy compared to data substitution as it can preserve the original data distribution and maintain the statistical properties of the data, while also protecting sensitive information.

To perform data shuffling, we first import the Pandas library and use the "read_csv" function to read a CSV file named "empdata.csv" and create a Pandas DataFrame called "df". We then use the "apply" function with a lambda function to shuffle the rows of the "df" DataFrame randomly. The "lambda" function uses the "sample" function to shuffle the rows of the DataFrame by generating a random fraction between 0 and 1 for each row. The resulting shuffled values are assigned to the "df" DataFrame, replacing the original values.

Employee Data before Data Shuffling

	Emp ID	Employee_name
0	677509	Lois Walker
1	940761	Brenda Robinson
2	428945	Joe Robinson
3	408351	Diane Evans
4	193819	Benjamin Russell
...	...	...
95	639892	Jose Hill
96	704709	Harold Nelson
97	461593	Nicole Ward
98	392491	Theresa Murphy
99	495141	Tammy Young

Employee Data after Data Shuffling

	Emp ID	Employee_name
0	134841	Steven Phillips
1	499687	William Hernandez
2	917395	Sharon Lopez
3	388642	Andrea Garcia
4	226714	Elizabeth Jackson
...	...	...
95	162402	Jimmy Howard
96	621833	Tammy Young
97	428945	Todd Hall
98	923947	Jack Campbell
99	623929	Ralph Flores

Figure 4.2: The figure displays two columns, one containing the original data and the other containing the data after performing data shuffling.

III. DATA ENCRYPTION:

The results of the above analysis suggest that data shuffling is a more effective masking technique than data substitution for providing high utility and high privacy. Therefore, applying encryption after data shuffling can help achieve high privacy by adding an additional layer of protection to sensitive data. By shuffling the data first, sensitive information is obscured and then encrypted, the combination of these two techniques provides a high level of privacy protection for sensitive data. Even if the shuffled data is intercepted, the sensitive information remains protected because it is encrypted and unreadable without the decryption key.

Encryption also helps to prevent unauthorized access to sensitive data, by making it unreadable to anyone who does not have the decryption key. This is especially important when sensitive data needs to be shared with third-party organizations or individuals for research or analysis purposes, as it helps to ensure that the sensitive information remains protected even when it is outside of the control of the original organisation.

We use a hybrid encryption algorithm that uses both AES-128 and RSA encryption. To perform data encryption, we use the following steps:

- We define a list of letters, including lowercase, uppercase, space, and period.
- We define several helper functions for finding factors, generating private keys, and checking for prime numbers.
- We define two encryption functions: AES128 and RSA_encrypt.
- We define a KeySetup function that generates a random AES key and prompts the user to input prime numbers or generates them randomly for RSA encryption. It also generates the public and private keys for RSA encryption and returns them.
- We use the KeySetup function to generate the necessary keys for encryption.
- We read the 'Employee_name' column from the shuffled pandas dataframe (df) and encrypt each name using AES128.
- We use RSA encryption to encrypt the generated AES key and print the encrypted key.
- We replace the original names in the dataframe with the encrypted names.
- We output the encrypted dataframe.

Shuffled Employee Data before Data Encryption

	Emp ID	Employee_name
0	134841	Steven Phillips
1	499687	William Hernandez
2	917395	Sharon Lopez
3	388642	Andrea Garcia
4	226714	Elizabeth Jackson
...	...	...
95	162402	Jimmy Howard
96	621833	Tammy Young
97	428945	Todd Hall
98	923947	Jack Campbell
99	623929	Ralph Flores

Employee Data after Data Encryption

	Emp ID	Employee_name
0	134841	b'\x0b`\xfdFK\xa9V\xc82z\xfa~pS\x15\xc0'
1	917937	b'\xd1\n\xdc~\x992)\xc1\xd6\x16\x0eu\xe5dv\x84'
2	969580	b'\x8c\xc6\xd8\x96Q\xe7QW\x97\xcb\xa8\xe7@2\xa...
3	407061	b'\xb8\x12\xeb\xff;\xce\xa6\x9b\xa38\xc0\xac\x...
4	818384	b'S\xcf\xa7\xd7\xf5\x13oU/\xa8\x93\xee\x0f\n8k'
...	...	...
95	499687	b'\x87p\x9b`a\x8bT\xb9\x02h\xf5p\xa1;\xd6\x06'
96	528509	b'\x8f#l\xf7\\\r\x11\xa7\xd2\x9c\xbe|)\x1a\xe7;'
97	683826	b'\x9c\xed\x95\xc5/K\xc9\xbf\x9c\x954\x07\xe4*...
98	940761	b'+\x1c]n\x87f}\xa2\xafb\xd5\x17?\xf1b\xe4'
99	687017	b'EF\xb4=.\xbd4\x82el\x1b\xf2\x10\x08\n\xdf'

Figure 4.3: The figure displays two columns, one containing the shuffled employee data and the other containing the data after performing data encryption.

Therefore, the use of hybrid encryption algorithm in this paper provides a higher level of security than normal methods. The idea behind this approach is to take advantage of the strengths of different encryption techniques to create a more secure overall encryption scheme.

Conclusion

There is an increasing need to protect sensitive data for employees, customers, and the entire enterprise, regardless of where the data resides. Until recently, most data theft was done by malicious individuals hacking into production databases publicly available on the internet. Large companies around the world outsource their IT and business process work to service providers in countries such as India, China and Brazil. With a spate of high profile and costly thefts creating both enormous legal liability and bad publicity for the organizations involved, businesses have rapidly become sophisticated to protect against such schemes. Although the industry is grappling with the most pernicious aspects of data theft, many computer systems remain vulnerable to some degree of attack.

The general public has expressed serious concerns about the disclosure of personal information. Sweeney reports that a zip code, date of birth, and gender are sufficient to uniquely identify her in 83% of the US population. These environments use real data to test applications that contain the most sensitive and confidential information within your organization, such as: B. Social Security Numbers, Bank Records, Financial Documents.

This study describes various data masking techniques that use modified data in place of the original data. For sensitive data, data masking techniques can be applied in various ways to ensure privacy. Alternatively, it can be complex, such as shuffling the data or replacing the existing data with algorithmically generated random data to protect the original data. These practices include masking sensitive data in all environments using proven commercial solutions and integrating these data protection processes and technologies across the enterprise.

References:

[1] Kanchana, V., & Venkatesan, S. (2019). Secure data masking and data sharing for cloud environment. Journal of Ambient Intelligence and Humanized Computing, 10(3), 1113-1123.

[2] Li, Y., & Li, L. (2019). A survey on data masking techniques. IEEE Access, 7, 89955-89968.

[3] Zhang, X., Zhou, W., & Yu, Z. (2018). A secure and efficient data masking method based on binary tree. Journal of Ambient Intelligence and Humanized Computing, 9(3), 827-838.

[4] Huang, Y., Liu, X., Zhou, L., & Hu, W. (2020). A novel data masking algorithm for privacy-preserving data publishing based on differential privacy. Soft Computing, 24(22), 16877-16893.

[5] Alshahrani, Al-Balas, & Al-Ghamdi, 2023. Comparative analysis of data masking techniques for preserving privacy and utility

[6] Ahmad, M., Umer, T., Khan, S. U., & Ahmed, S. (2020). Data privacy and security: A review of data masking and hybrid encryption. International Journal of Advanced Computer Science and Applications, 11(4), 308-312.

[7] Tong, L. L., Li, P. X., Duan, D. S., Ren, B. Y., & Li, Y. X. (2022). Data masking model for heterogeneous big data environment. Journal of Beijing University of Aeronautics and Astronautics, 48(2), 249-257.

[8] Torra, V. (2022). Privacy for Data: Masking Methods. In Guide to Data Privacy: Models, Technologies, Solutions (pp. 159-209). Cham: Springer International Publishing.

[9] Miserom, S. F., Sabri, S., Aida, F. F., & Ismail, N. (2021, February). Secure Database on Attendance Verification System by using Data Masking Technique. In IOP Conference Series: Materials Science and Engineering (Vol. 1062, No. 1, p. 012030). IOP Publishing.

[10] Chinnasamy, P., Padmavathi, S., Swathy, R., & Rakesh, S. (2021). Efficient data security using hybrid cryptography on cloud computing. In Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020 (pp. 537-547). Springer Singapore.

[11] Ali-Ozkan, O., & Ouda, A. (2019, June). Key-based reversible data masking for business intelligence healthcare analytics platforms. In 2019 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-6). IEEE.
[12] Siddartha, B. K., & Ravikumar, G. K. (2019, December). Analysis of masking techniques to find out security and other efficiency issues in healthcare domain. In 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 660-666). IEEE.

[13] Archana, R. A., Hegadi, R. S., & Manjunath, T. N. (2018). A study on big data privacy protection models using data masking methods. International Journal of Electrical and Computer Engineering, 8(5), 3976.

[14] Gokulraj, S., Ananthi, P., Baby, R., & Janani, E. (2021). Secure File Storage Using Hybrid Cryptography. Available at SSRN 3802668.

[15] Jain, R. B., Puri, M., & Jain, U. (2018). A robust dynamic data masking transformation approach to safeguard sensitive data. Int. J. Future Revolution Comput. Sci. Commun. Eng, 4(2).

[16] Ali, O., & Ouda, A. (2017, April). A content-based data masking technique for a built-in framework in Business Intelligence platform. In 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-4). IEEE.

[17] Ajayi, O. O., & Adebiyi, T. O. (2014). Application of Data Masking in Achieving Information Privacy. IOSR Journal of Engineering, 4(2), 13-21.

[18] Chinnasamy, P., & Deepalakshmi, P. (2018, April). Design of secure storage for health-care cloud using hybrid cryptography. In 2018 second international conference on inventive communication and computational technologies (ICICCT) (pp. 1717-1720). IEEE.

[19] Chinnasamy, P., Padmavathi, S., Swathy, R., & Rakesh, S. (2021). Efficient data security using hybrid cryptography on cloud computing. In Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020 (pp. 537-547). Springer Singapore.

[20] Tanuwidjaja, H. C., Choi, R., Baek, S., & Kim, K. (2020). Privacy-preserving deep learning on machine learning as a service—a comprehensive survey. IEEE Access, 8, 167425-167447.

[21] Al-Rubaie, M., & Chang, J. M. (2019). Privacy-preserving machine learning: Threats and solutions. IEEE Security & Privacy, 17(2), 49-58.

[22] Biswas, C., Gupta, U. D., & Haque, M. M. (2019, February). An efficient algorithm for confidentiality, integrity and authentication using hybrid cryptography and steganography. In 2019 international conference on electrical, computer and communication engineering (ECCE) (pp. 1-5). IEEE.

[23] Murthy, S., Bakar, A. A., Rahim, F. A., & Ramli, R. (2019, May). A comparative study of data anonymization techniques. In 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS) (pp. 306-309). IEEE.