Computers, Materials & Continua
DOI: 10.32604/cmc.2023.030899
Article

Two Layer Symmetric Cryptography Algorithm for Protecting Data from Attacks

Muhammad Nadeem[1], Ali Arshad[2], Saman Riaz[2], Syeda Wajiha Zahra[1], Shahab S. Band[3] and Amir Mosavi[4,5,6,*]

[1]Department of Computer Science, Abasyn University, Islamabad, 44000, Pakistan
[2]Department of Computer Science, National University of Technology, Islamabad, 44000, Pakistan
[3]Future Technology Research Center, National Yunlin University of Science and Technology, Douliu, Yunlin, 64002, Taiwan
[4]Institute of Information Society, University of Public Service, Budapest, 1083, Hungary
[5]John Von Neumann Faculty of Informatics, Obuda University, Budapest, Hungary
[6]Institute of Information Engineering, Automation and Mathiltics, Slovak University of Technology in Bratislava, Slovakia
*Corresponding Author: Amir Mosavi. Email: amir.mosavi@kvk.uni-obuda.hu
Received: 05 April 2022; Accepted: 17 May 2022

Abstract: Many organizations have insisted on protecting the cloud server from the outside, although the risks of attacking the cloud server are mostly from the inside. There are many algorithms designed to protect the cloud server from attacks that have been able to protect the cloud server attacks. Still, the attackers have designed even better mechanisms to break these security algorithms. Cloud cryptography is the best data protection algorithm that exchanges data between authentic users. In this article, one symmetric cryptography algorithm will be designed to secure cloud server data, used to send and receive cloud server data securely. A double encryption algorithm will be implemented to send data in a secure format. First, the XOR function will be applied to plain text, and then salt technique will be used. Finally, a reversing mechanism will be implemented on that data to provide more data security. To decrypt data, the cipher text will be reversed, salt will be removed, and XOR will be implemented. At the end of the paper, the proposed algorithm will be compared with other algorithms, and it will conclude how much better the existing algorithm is than other algorithms.

Keywords: Cryptography; symmetric algorithm; encryption; decryption; cipher text; cloud security; asymmetric algorithm

1 Introduction

Cloud computing is an online application-based software where various servers are connected [1] and data is exchanged between servers, and online services are provided to end-users. Many organizations store their data online for long-term storage and remote access [2], but in addition to storing data in cloud computing, data protection is also important [3]. Different attackers try to

break the security of cloud servers and gain access to data [4], for which they develop other attacking mechanisms. The most common cause of cloud server attacks is pirated software. An attacker can never directly access the cloud server, but he can take support of different attacking mechanisms to gain access to the cloud server and make every effort possible for him. Firewalls are commonly used to protect cloud computing from attacks [5]. A firewall has a set of rules and policies designed to protect the cloud server from intrusion, but a firewall is not sufficient to protect the cloud from attacks [6].

As the number of users on the network grows, so does the number of attacks on the network [7]. Many researchers have developed various security mechanisms to prevent attacks that can be successful, including authentication and authorization, access control, data backup, and intrusion detection systems, but all of these mechanisms cannot completely secure the cloud server. Despite the many unsecured security mechanisms, the chances of cloud attacks are increasing rather than decreasing. There are so many secured mechanisms that are also available that can protect cloud servers from attacks that, if fully implemented, can reduce attacks from cloud servers. The best of which is the cryptography mechanism.

1.1 Cryptography

Cryptography is a data encryption mechanism that prevents sender and receiver data from third-party interruption and access [8]. The encryption mechanism converts the original text to cipher text via a symmetric or asymmetric key, as shown in Fig. 1. This data must be decrypted to bring it back from cipher form to the original form. A cipher text is a non-readable form of any data [9]. Whenever data is encrypted with the help of cryptography algorithms, a decryption mechanism has to be used to make that data readable again, as in this article.

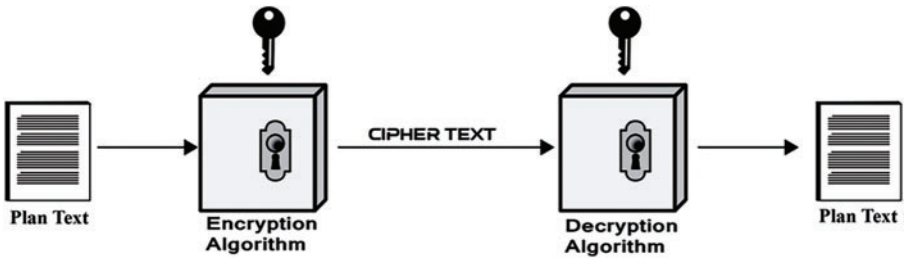

Figure 1: Encryption mechanism

It is better to use an efficient algorithm to protect cloud server data from unauthorized access [10], instead of having different firewalls and mechanisms around the cloud server data [11]. Suppose the researcher develops a mechanism to secure the cloud server. In that case, the attacker develops a double mechanism to destroy the data, increasing the attack ratio on the cloud server instead of decreasing it [12]. Suppose efficient encryption algorithms are used instead of any security techniques for data transmission in cloud computing. In that case, it will be difficult for the attacker to understand the non-readable format and convert it to a readable format [13].

1.2 Cryptography Goals

The primary purpose of encryption is to provide users with a secure platform for communication and data transmission in the presence of malicious third parties [14]. There are four main cryptography goals: confidentiality, integrity, authentication, and non-repudiation. The goals are described in Tab. 1.

Table 1: Goals of cryptography

Goals	Description
Confidentiality	In confidentiality, the original data is hidden, encrypted, and converted to non-readable format, and this data can only be decrypted or accessed by authorized users. The recipient uses either the same keys that the sender used or another key to decrypt the data.
Integrity	Data integrity means that the recipient must receive the same data that was sent to the recipient and that data has not been modified.
Authentication	In authentication, it is checked whether an authorized person is sending the data sent to the recipient or not. Similarly, on the receiver side, it is checked whether the data it will receive is authentic or not.
Non-repudiation	Non-repudiation means that whenever a sender or receiver makes a transmission, they cannot deny that they did not send or receive data. The sender can prove that he sent the data, and the receiver can prove that he has received it.

1.3 Encryption and Decryption

Encryption is an efficient security mechanism that converts plain text to non-readable format [15]. The best way to encrypt data in a cloud server is to encrypt sensitive data [16]. Sensitive data means data that needs to be protected from third parties. If sensitive data is to be encrypted from the transmitter side, then this data has to be converted in a readable format on the receiver side. Decryption is used to convert a non-readable format to a readable format. Encryptions and decryptions require a specific algorithm [17].

1.4 Cryptography Algorithms

Cloud encryption mechanisms are divided into two categories as shown in Fig. 2. One is first Key-Based Encryption Mechanism and the second Key Less Encryption mechanism. Keys-Based encryption mechanisms involve the simultaneous use of public, private, or both keys, while the keyless mechanism uses addressable physically unclonable functions for generating the cipher text. In Key-Based mechanism, Data is encrypted and decrypted using the same keys [18]. In Asymmetric cryptography mechanism, data is encrypted and decrypted with different keys. The first key used to encrypt the data and the second key is used to decrypt the data [19]. In symmetric cryptography mechanism, data is encrypted and decrypted with same keys. Different Symmetric mechanisms are DES, AES, RC4, RC5, Blowfish, TEA, 3DES whereas Asymmetric mechanisms are RSA, ECC, Diffie-Hellman and EES.

1.4.1 Asymmetric Key Encryption Mechanism

The asymmetric encryption method uses two keys for data encryption. The first key is public, and the second is private [20]. This method uses two keys (public and private) on the transmitter and receiver sides to exchange data. If the data is encrypted with the private key of the sender or receiver, this data will also be decrypted with the same sender or receiver public key. Similarly, If the data is encrypted with the public key of the sender or receiver, this data will also be decrypted with the same sender or receiver private key.

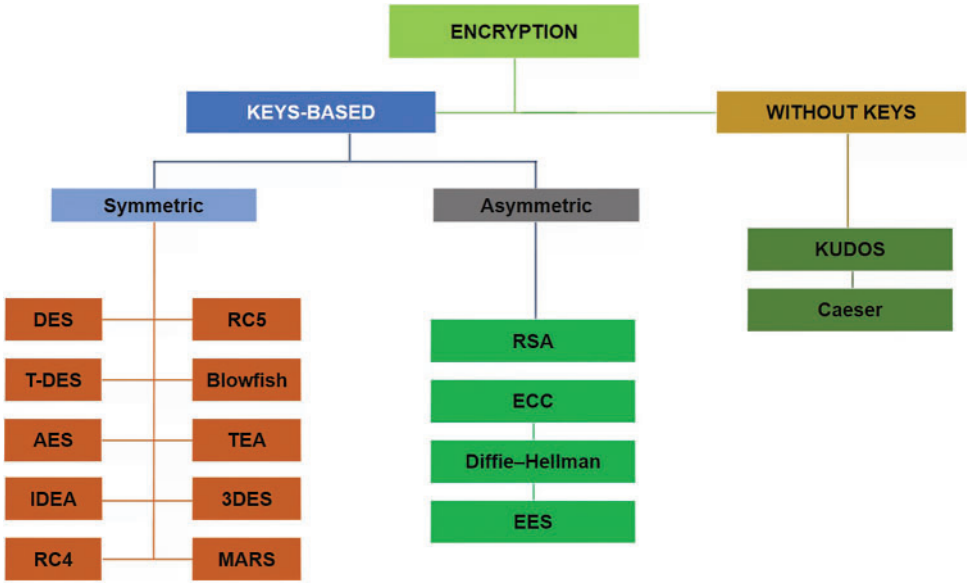

Figure 2: Cryptography algorithms

1.4.2 Symmetric Key Encryption Mechanism

In Symmetric cryptography, data is encrypted and decrypted by the same key [21]. Symmetric key cryptography is also known as secret-key cryptography. It is fast as compared to asymmetric key cryptography [22]. When the data is transmitted, the data is converted to a non-readable text by the symmetric cryptography mechanism. Non-readable text is also known as cipher text [23]. When the receiver receives the cipher text, the text must be converted to readable text. When the non-readable is converted to readable text, this process is called decryption.

1.5 Problem Formulation

In this paper, a double encryption symmetric mechanism has been developed with the help of which data has been sent and received in a secure form. Whenever data is transferred to a cloud server, the attacker either gains access to the account via pattern matching keys or attempts to use key data using any algorithm. This article develops an encryption technique that will prevent the attacker from accessing the data even if the attacker tries to access the cloud data through any mechanism. A double encryption mechanism is used to securely transfer the data so that if the attacker tries to decrypt the data, the encrypted data cannot be decrypted and the data is sent to the recipient in a trusted format. Only authorized users can access the data.

The rest of the paper is distributed like this. Section 2 deals with recent work. Section 3 discusses the previous work. Section 4 consists of testing. Section 5 provides comparative Analysis while Section 6 discusses the conclusion.

2 Literature Review

The authors described many security techniques and algorithms [24]. This article provides as much information about cryptography as possible to researchers, students, and inexperienced people. The main purpose of this article is to identify the gaps in cryptography in which the authors have identified

two research gaps. The first gap is that researchers ignore the authentication process in security whereas user authentication is the biggest problem of any organization. The second gap is that many researchers focus only on theory or partial implementation and ignore complete implementations. The authors have not discussed all the cryptographic algorithms in this paper but discuss the pros and cons of these algorithms which can be used for new research.

Author discussed different symmetric and asymmetric encryption algorithms to identify the performance and accuracy in terms of security [25]. According to the authors, the data can be secured if the same keys are used in the symmetric keys algorithm and if the asymmetric keys algorithm is used, the data can be encrypted and decrypted by two keys. If these algorithms are made efficient, many internal and external threats can be minimized by maintaining data security, integrity, and confidentiality. At the end of the paper, the authors have discussed different results and compared different techniques. They conclude that the performance of the symmetric algorithm is faster and more reliable than the asymmetric algorithm and it is a more reliable algorithm in terms of encryption, decoding, speed, and structure.

In this article author analyzes many of the cryptographic methods used by cloud providers [26]. Authors have proposed a cryptography algorithm to encrypt data and protect against man-in-the-middle attacks. The data is first converted to ASCII values and converted from ASCII values to binary numbers to encrypt the data. If the binary number does not have 8 bits, then 0 is added for full bits, then 1's complement of the last four bits was taken, and the binary number was again converted to ASCII. To decrypt the data, the ASCII value was first converted to binary, reversed the last 4-bits of binary, and then converted to ASCII.

In this article, the authors introduce an encryption algorithm, which will encrypt the file before uploading it to the client-side cloud [27]. Once the receiver side is downloaded, the file will be decrypted with the same keys that were used during encryption. First, a platform was created for uploading a file, and then a file was encrypted. This encrypted file was then uploaded with the keys. After that, the encrypted file was downloaded from the local storage and then decrypted. This algorithm is suitable for text files encryption.

This article [28] proposed an encryption system based on biometric authentication, user authentication and a searchable encryption scheme. According to the authors, If these components are combined, the data can be accessed in the document without decrypting and searched by keywords and only that data can be displayed, which will match with keywords. The main purpose is to introduce a new authentication technique by combining two-factor authentication and searchable encryption mechanism. The authors conducted various experiments and, based on these experiments, discussed the result that when a user search using keywords, the correct result will be displayed and argued that authentic users can provide results without searching algorithm decryption.

In this paper [29], the authors proposed a distributed privacy protection scheme for random linear network coding in the smart grid and used the homomorphic encryption function to reduce the complications in the forward node. This method proposed data confidentiality and many privacy preserving features. At the end by different experiments and security analysis authors concluded that this proposed method can effectively maintain privacy.

In this paper [30], the authors proposed a location data record privacy protection method that is based on a differential privacy mechanism, which used the structure of a multi-level query tree and produced location data on the database. After that discussed that this method is more effective in protecting privacy and accessing data than other privacy mechanisms.

According to the authors [31], Identity Proxy Signatures (IDPs) are signatures that contain multiple applications, which main include Mobile communication and distributed network. Most of the existing IDPS schemes suffer from loose security reduction shortcomings In this paper [31], the authors proposed an IBPS framework and presented a detailed security model for IBPS in the standard model. and reduced the security assumption with the help of Diffie-Hellman. After that, to show the efficiency of the work, compare the latest paper scheme with identity-based proxy signature schemes and concluded that their scheme is more efficient and secure.

Group signature is a cryptographic technique that allows each group member to sign messages from the group to which they belong. In this paper [32], the authors proposed a fully traceable identity-based group signature scheme (TIBGS) which lacks the security of the computational Diffie-Hellman (CDH) assumption, and also proposed a formal security model for traceable identity-based group signature and concluded that the proposed method is efficient in traceability and anonymity.

In this paper [33], the authors proposed a traceable threshold signature scheme based on water's signature scheme and also presented a traceable threshold proxy signature (TTPS) frame which has security reduction to CDH assumption, and discussed the proposed method is more secure in the standard model.

In this paper [34], authors proposed a hashing method named as discrete multi-graph hashing (DMGH), to deal with the challenges of use of hashing to handle multi view data. This proposed method reduces distortion errors. After experiments on large scale datasets authors concluded that the proposed method performs better in multi view hashing methods.

According to the researchers [35], WBAN is a wireless sensor-based network and these sensors are connected to the human body that is used to measure the psychological information of patients. This information is transmitted to doctors remotely so that the doctor could diagnose the patient's problems remotely. When data is transferred to WBAN, patient information is sensitive, which can lead to patient privacy. To address this issue, the researchers proposed two techniques: mutual verification and key agreement for multi-server switching in WBAN. The purpose of using these techniques is that the patient can do communication in a safe environment at any time using multi-server.

Different researchers have discussed different algorithms to secure the data. Some researchers have tried to identify the gaps in cryptography [24]. While some researchers have checked the efficiency of the algorithm by comparing symmetric and asymmetric cryptography [25]. Some researchers have converted data into bits and inverted bits to prevent data from being attacked [26]. Some researchers have encrypted the data with a unique key and sent it to the receiver to implement the decryption mechanism. With the help of all these mechanisms, basic level security of data can be provided, but with these mechanisms, cloud data cannot be fully secured, nor can data be transmitted in an efficient and secure way. The way this paper works is different from all other papers. A two-layer cryptographic algorithm has been developed in this paper, which will help secure data in two phases. If the attacker tries to access the data using some mechanism, having two-layer encryption will make it impossible for the attacker to access the data. A complete encryption security mechanism will be developed using techniques such as Salting, Bit conversion, Bits reversing, Key generator, and Bits XORtion for encrypting data. With the help of these mechanisms the data will be encrypted and the data decryption can be done at the same time when all these mechanisms will be used.

3 Proposed Algorithm

This article has developed an efficient symmetric cryptography algorithm to securely send and receive data that will keep cloud data safe from intruders.

3.1 Functional Overflow

Cloud cryptography mechanism has been developed into two categories, as shown in Fig. 3. First, the data has been encrypted by the sender, and then the data has been decrypted by the recipient. The plain text has been converted to the corresponding ASCII for encryption. After that, ASCII values have been converted to binary. A random key has been generated from a key generator corresponding to the length of the binary number obtained from the ASCII value to provide security for the data. The keys have been then XORed with binary bits to secure the data. The salt bits have been appended with the bits obtained from the XOR function so that the data can be provided with double encryptions and the bits obtained after that have been reversed. The reason for reversing is that if the attacker tries to decrypt the data using any mechanism, then in that case he will not be able to retrieve the original data. After reversing bits, the cipher text has been obtained.

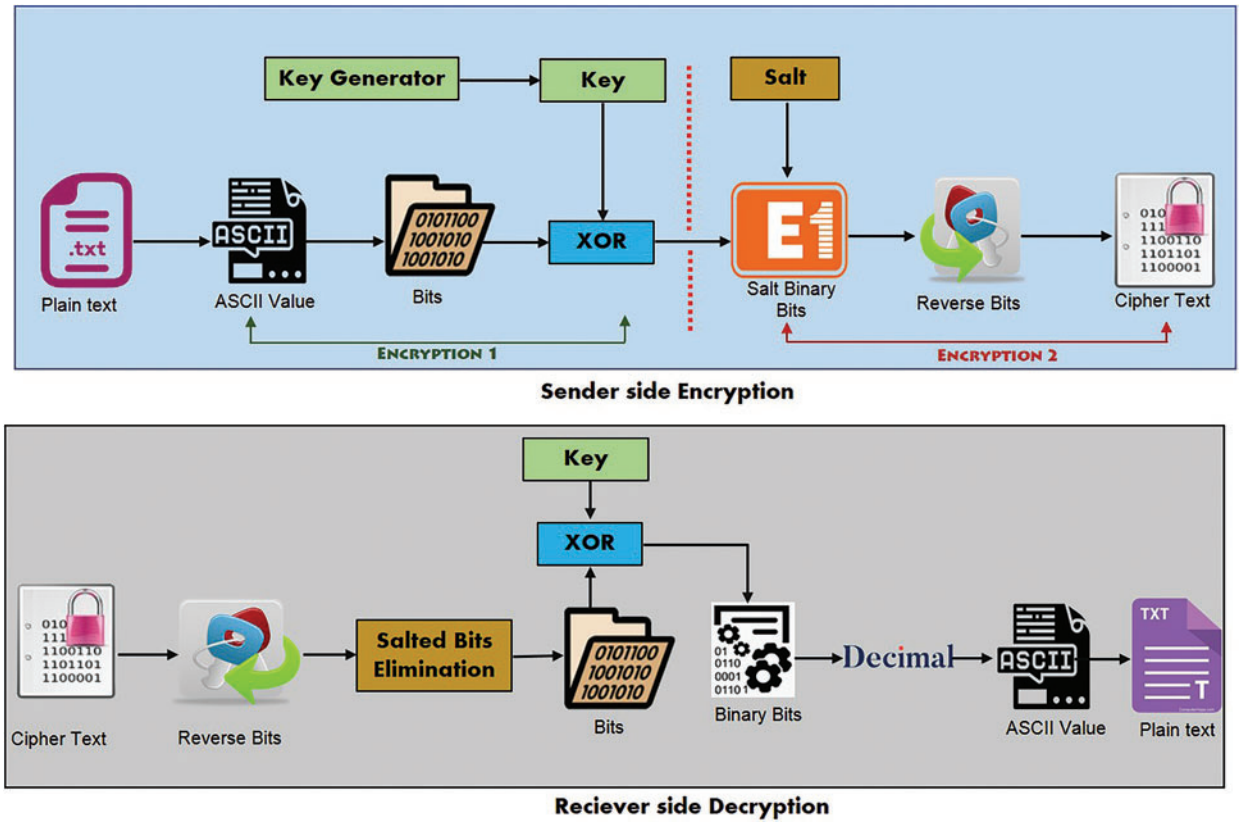

Figure 3: Classification of encryption

First of all, encrypted data has been reversed for decryption. after that salt bit has been removed. Bits obtained after removal of salt bits, and key have been XORed. The bits received after the XOR function have been converted to decimal. Decimal values were then converted to their equivalent ASCII values, after which the data were decrypted.

Algorithm 1: Encryption

Input: Plain text
Output: Cipher text

1. Convert each character into its equivalent ASCII.
2. Convert each ASCII into 8-bit binary, as shown in Fig. 4.
3. XOR binary values and the generated key.
4. Add salt to the binary values obtained in step 3.
5. Reverse bits.
6. Get cipher text.

Algorithm 2: Decryption

Input: Cipher text
Output: Plain text

1. Reverse the encrypted bits.
2. Remove salt bits by eliminating last 4 bits, as shown in Fig. 5.
3. XOR the key and the binary value obtained in step 2.
4. Convert the binary values into decimal value.
5. Convert decimal to ASCII.
6. Get plain text.

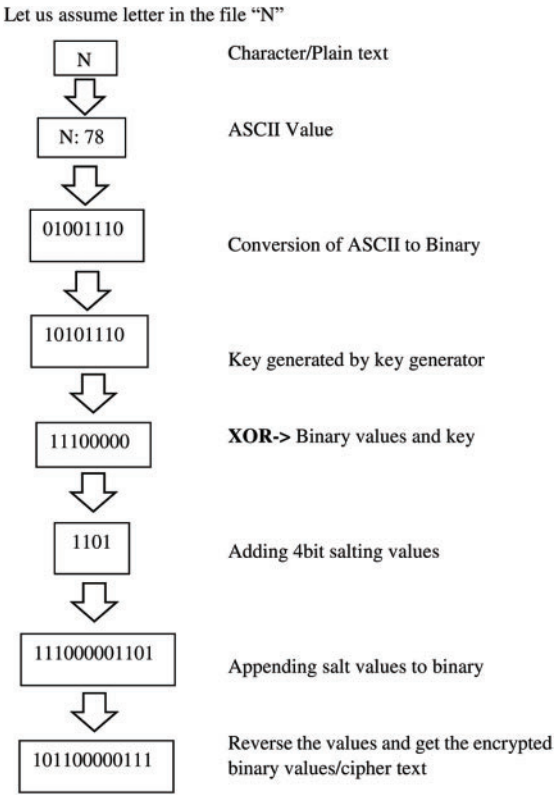

Figure 4: Encrypted text

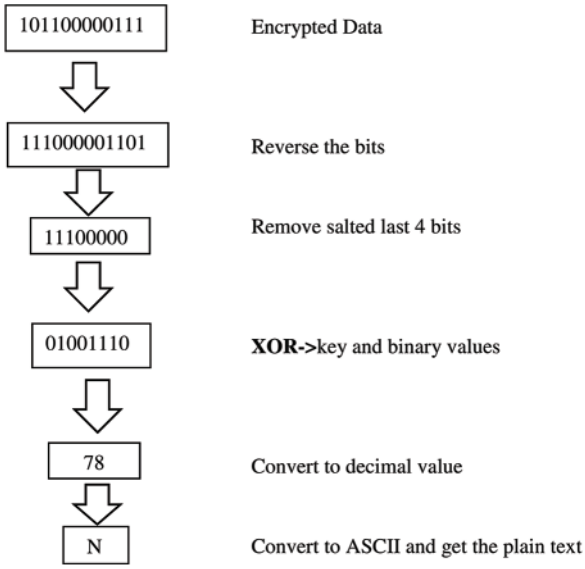

Figure 5: Decrypted text

3.2 Encryption Process Example

Let's suppose character 'N' from the file to be encrypted. The character has been converted to its equivalent ASCII that is 78. After converting the character to ASCII value, ASCII values have been converted into 8-bit binary values. After the conversion of ASCII to binary, a random key has been generated by using a key generator according to the length of binary values. After generating the key, binary values and key have been XORed. To provide more security to data, salt has been applied to the XOR result. 4-bit salt values have been appended at the end of the XOR result. After appending the salt, all the bits have been reversed, so if an attacker tries to access the data, due to the reverse mechanism, the attacker cannot decrypt the data and keep trying to decrypt the reversed data instead of accessing the original data. After reversing the bits, a ciphertext has been obtained.

3.3 Decryption Process Example

To decrypt the ciphertext, first of all, encrypted bits have been reversed. After reversing, 4-bit salt values have been removed from the end of the bits. After removal of salt, binary values have been XORed with the key generated by a key generator during the encryption process. When the data is sent to the receiver, the key generated by the key generator is also sent but only accessible to authentic users. The binary values obtained from XOR have been divided into 8-bits and then converted into decimal values. After getting decimal values, the ASCII value of each decimal value has been obtained. All ASCII values have been combined and plain text has been obtained.

4 Testing

A plain text has been taken to test the algorithm, and this text has first been encrypted under different steps to get the cipher text. After this, cipher text has since been decrypted in various stages.

4.1 Encryption Algorithm

Step 1: First of all, a plain text has been taken for encryption, as shown in Fig. 6.

```
Cryptography Algorithm
```

Figure 6: Plain text

Step 2: To convert text to binary, the text has been first divided into characters and then the ASCII of each character has been generated, as shown in Fig. 7.

```
C  r  y  p  t  o  g  r  a  p  h  y     A  l  g  o  r  i  t  h  m
67 114 121 112 116 111 103 114 97 112 104 121 32 65 108 103 111 114 105 116 104 109
```

Figure 7: ASCII of each character

Step 3: After converting plain text to ASCII values, each ASCII value has been converted to binary to provide data security, as shown in Fig. 8.

```
010000110111001001111001011100000111010001101111011001110111001001100001
011010000111100100100000010000010110110001100111011011110111001001101001
011101000110100001101101
```

Figure 8: Convert each ASCII to binary

Step 4: After converting ASCII to binary, a random key has been generated using the key generator according to the length of the key, as shown in Fig. 9.

```
111010110101101101010100010100100101110100111001101100100101011011101011
001000100101000101111011010110010110010001100101011000010111001001101001
010101000010100001101000
```

Figure 9: Random key generates

Step 5: After generating the key, Figs. 8 and 9 have been XOR using the following formula.

$$B_1B_2B_3B_4 \ldots\ldots\ldots\ldots..B_n$$
$$\oplus K_1K_2K_3K_4 \ldots\ldots\ldots\ldots..K_n$$
$$X_1X_2X_3X_4 \ldots\ldots\ldots\ldots..X_n$$

$B_1 \ldots\ldots\ldots\ldots.B_n$ represents the binary keys, while K1 $\ldots\ldots\ldots\ldots$ Kn represents the random keys. When the binary keys have been XOR with random keys, the X1 $\ldots\ldots\ldots\ldots$ Xn result has been obtained, as shown in Fig.10.

```
101010000010100100101101001000100010100101010111011010100100100010101
010010100010100001011011000110000000100000000010000011100000000000000000
001000000100000000000101
```

Figure 10: XOR result

Step 6: To further secure the data, salt has been applied on the XOR results, and 4-bits values have been appended at the end of XOR result, as shown in Fig. 11.

```
Salting Value: 1101
10101000001010010010111010010001000101001010101101101010100100100010101
01001010001010000101101100011000000010000000001000001110000000000000000
00100000010000000000001011101
```

Figure 11: Appending of salt

Step 7: After appending the salt, all the bits have been reversed as shown in Fig. 12, So if an attacker tries to access the data, due to the reverse mechanism, the attacker cannot decrypt the data and keep trying to decrypt the reverse data instead of accessing the original data.

```
10111010000000000010000001000000000000000000111000001000000000100000001
10001101101000010100010100100101000100100100101010110110101010010100100
01001011010010010100000010101
```

Figure 12: Reverse bit mechanism (cipher text)

After reversing the bits, a cipher text has been retrieved.

4.2 Decryption Algorithm

All of these different steps must be followed to decrypt the cipher text. Fig. 13 shows an encrypted cipher text that will be decrypted in various steps.

```
10111010000000000010000001000000000000000000111000001000000000100000001
10001101101000010100010100100101000100100100101010110110101010010100100
01001011010010010100000010101
```

Figure 13: Cipher text

Step 1: First, the bits are reversed to decrypt the cipher text, as shown in Fig.14.

```
10101000001010010010111010010001000101001010101101101010100100100010101
01001010001010000101101100011000000010000000001000001110000000000000000
00100000010000000000001011101
```

Figure 14: Cipher bits reverse

Step 2: After reversing the bits, the salt mechanism has been removed from cipher text as shown in Fig. 15.

```
10101000001010010010111010010001000101001010101101101010100100100010101
01001010001010000101101100011000000010000000001000001110000000000000000
00100000010000000000101
```

Figure 15: Detach salt from bits

Step 3: After the removal of the salt, the binary values of Fig. 15 have been XORed with the keys obtained from the key generator using the formula given below. whenever data is sent to the receiver, the key obtained from the key generator will also be sent, but only the authentic users can use the key correctly otherwise key will be useless.

$X_1X_2X_3X_4 \ldots\ldots\ldots\ldots..X_n$

$\oplus K_1K_2K_3K_4 \ldots\ldots\ldots\ldots..N_n$

$M_1M_2M_3M_4 \ldots\ldots\ldots\ldots..M_n$

When salt and key values were XORed, a result was obtained, shown in Fig. 16.

```
01000011011100100111100101110000011101000110111101100111011100100110000 1
01101000011110010010000001000001011011000110011101101111011100100110100 1
01110100011010000110110 1
```

Figure 16: XOR result of salt with key

Step 4: The binary values obtained from the XOR function has been divided into 8-bits and converted into decimal values. Different ASCII values have been obtained, as shown in Fig. 17.

```
67 114 121 112 116 111 103 114 97 112 104 121 32 65 108 103
111 114 105 116 104 109
```

Figure 17: Results of 8-bits conversion

Step 5: After converting binary values into decimals, ASCII value each decimal number has been calculated to obtain original data, as shown in Fig. 18.

```
67 114 121 112 116 111 103 114 97 112 104 121 32 65 108 103 111 114 105 116 104 109
C  r   y   p   t   o   g   r   a   p   h   y     A   l   g   o   r   i   t   h   m
```

Figure 18: Results of decimal to ASCII

Step 6: The original data was converted to ASCII values, as shown in Fig. 19.

```
Cryptography Algorithm
```

Figure 19: Original text

Salting and bit reversing are such mechanisms with the help of which data can be given such an environment where it is difficult for attackers to identify and follow the same steps as used in decryption, as discussed in this paper. Data transmission can be done in secure and reliable environments when these mechanisms are used better.

5 Comparative Analysis

Different authors developed different techniques to protect cloud data. In 2019 [25], the authors made a comparison between two techniques one is symmetric and the other is Asymmetric, discussed various encryption and decryption algorithms, and identified which algorithm's performance is better. The author discussed that the Symmetric algorithm is better and more reliable than Asymmetric algorithms in encryption, decryption, speed, and structure. But this paper did not discuss how symmetric algorithm can be implemented to transmit data in a reliable form. In 2021 [26], the author has analyzed various cryptographic algorithms and proposed one cryptography algorithm to encrypt cloud data in which the author shows different steps to encrypt and decrypt data, but it is very easy for attackers to guess this mechanism and can easily decrypt the data. In 2021 [27], the author introduced one symmetric cryptography algorithm in which the client-side file was encrypted with one key and

uploaded this file with a key on the cloud and on the receiver-side decrypted the file with the same key used in the encryption process. The problem with this algorithm is that If the attacker gets this decryption key, it will be easier for the attacker to develop a decyption mechanism. In 2022 [28], the authors introduced an authentication technique by combining two-factor authentication and a searchable encryption mechanism. After performing different experiments and discussing that correct results would be displayed in searchable encryption when users search by using keywords. The problem with this article is that if the attacker understands the keyword searching mechanism, access to the data will be easier for the attacker. A comparative analysis with proposed work has been shown in Tab. 2.

Table 2: Comparative analysis

Sr#	1	2	3	4	Proposed Work
Paper name	A Surveyon symmetric and asymmetric cryptography algorithms in information security [25]	Cloud cryptography-a security aspect [26]	Client-side cryptography based security for cloud computing system [27]	A searchable encryption scheme with biometric authentication and authorization for cloud environments [28]	
Proposed algorithm	Identification of symmetric and asymmetric encryption algorithm performance	To secure data transmission between clients	Encrypt data with unique key	Combination of two factor authentication and searchable encryption	Secure data transmission on cloud server
Novelty	Symmetric algorithm is reliable than asymmetric	Protection against Man-in-the-Middle attack	Suitable on text files	Searchable encryption	Double symmetric encryption algorithm

6 Novelty of Proposed Work

No such technique has been used in any of the articles [25–28] that would make it difficult for attackers to decrypt data if implemented, nor has been implemented a two-way encryption algorithm to provide security of data in any paper. If a two-layer encryption mechanism is implemented to protect the data, such a mechanism will make it difficult for attackers to attack and decrypt the data as developed in this paper and if the attacker develops a decryption mechanism, access to the data will not be possible due to the use of different mechanisms.

This paper has developed a two-layer encryption mechanism to secure data on the cloud server, enabling data transmission between clients in a secure environment. The data has been converted to the ciphertext in binary form so that if an attacker wants to decrypt the data, they get entangled in bits. Instead, they start decrypting the data. The salt mechanism has been used to provide further encryption of the data and the bits obtained from the salt mechanism have been reversed. Suppose an

attacker uses any data decryption technique. In that case, the different encryption phases will make data decryption difficult for the attacker. If the attacker tries to decrypt a single phase, the data will not be accessible due to double encryptions. If an attacker designs a data decryption mechanism, the reverse process will not allow that data to be decrypted.

7 Conclusion

An efficient two-layer algorithm has been developed to secure cloud data, in which plain text has been converted to bits. A value has been obtained by XORing the bits with the bits obtained from the key generator and salt has been applied to the obtained value. To make the text more secure, the value obtained after applying the salt was reversed and a cipher value was obtained. A cloud server can be secure only when its connected devices and data transmissions will be secure. Data on the cloud server can be accessed both internally and externally. Attackers can gain access to data by deceiving users through pattern matching, brute force or phishing techniques that are detrimental to cloud server data. Cryptography mechanism is the best way to solve this problem. When cryptography is applied to cloud servers, the amount of data stored in the cloud server will be stored in encrypted form, and whenever there is data transmission between clients, it will be in a secure form. The cloud servers cannot be secure unless the devices connected to them and their transmission are secure.

In the future, an asymmetric mechanism will be used for encryption. The receiver's public key will be encrypted using a triple encryption mechanism and a private key will be used to decrypt that data. Cloud shell cryptography mechanisms will be designed to access cloud server data which will be used to secure the cloud server.

Acknowledgement: We would like to thank Abasyn University for their resources and help throughout the development of this research, and our time in gaining the knowledge and tools we would need to succeed in the professional world.

Author Contributions: M. N. Conceptualization; A. A. Software, Writing—review & editing; S. R Writing—review & editing; S. W. Z Methodology; M. R Writing—review & editing; S. S. funding acquisition; and A. M. Editing.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Nadeem, A. Arshad, S. Riaz, S. S. Band and A. Mosavi, "Intercept the cloud network rom brute force and DDoSattacks via intrusion detection and prevention system," *IEEE Access*, vol. 9, pp. 152300–152309, 2021.

[2] S. M. Naser, "Cryptography: From the ancient history to now," *It's Applications and a new Complete Numerical Model*, vol. 9, no. 3, pp. 11–30, 2021.

[3] V. Vishal and K. Vasudha, "DOS/DDoSattack detection using machine learning: A review," in *Proc. of the Int. Conf. on Innovative Computing & Communication (ICICC)*, Delhi, India, 2021.

[4] N. G. McDonald, "Past, present and future methods of cryptography and data encryption," *A Research Review*, University of Utah, May 2020.

[5] M. M. Islam, M. Z. Hasan and R. A. Shaon, "A novel approach for clientside encryption in cloud computing," in *Int. Conf. on Electrical, Computer and Communication Engineering (ECCE)*, Cox'sBazar, Bangladesh, pp. 1–6, 2019.
[6] D. Boland, "Securing amazon web services (AWS) and simple storage service (amazon s3) security," *An Article Regarding Amazon Simple Storage Service Security*, available at http://www.infosecwriters.com, June 2020.
[7] P. Lula, O. Dospinescu, D. Homocianu and N. A. Sireteanu, "An advanced analysis of cloud computing concepts based on the computer science ontology," *Computers, Materials & Continua*, vol. 66, no. 3, pp. 2425–2443, 2021.
[8] Y. Ren, C. Wang, Y. Chen, M. C. Chuah and J. Yang, "Signature verification using critical segments for securing mobile transactions," *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 724–739, 2020.
[9] M. Hasan, Noor Ariffin and N. Sani, "A review of cryptographic impact in cybersecurity on smart grid: Threat, challenges and countermeasures," *Journal of Theoretical and Applied Information Technology*, vol. 99, pp. 2458–2472, 2021.
[10] S. A. Ahmad and A. B. Garko, "Hybrid cryptography algorithms in cloud computing: A review," in *2019 15th Int. Conf. on Electronics, Computer and Computation (ICECCO)*, Abuja, Nigeria, pp. 1–6, 2019.
[11] P. N. Brown, H. P. Borowski and J. R. Marden, "Security against impersonation attacks in distributed systems," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 1, pp. 440–450, 2019.
[12] I. Meraouche, S. Dutta, H. Tan and K. Sakurai, "Neural networks-based cryptography: A survey," *IEEE Access*, vol. 9, pp. 124727–124740, 2021.
[13] Q. Hu, B. Du, K. Markantonakis and G. P. Hancke, "A session hijacking attack against a deviceassisted physical-layer key agreement," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 691–702, 2020.
[14] Y. Sharma, H. Gupta and S. K. Khatri, "A security model for the enhancement of data privacy in cloud computing," in *Amity Int. Conf. on Artificial Intelligence*, Dubai, United Arab Emirates, pp. 898–902, 2019.
[15] Y. Han, L. Duan and R. Zhang, "Jamming-assisted eavesdropping over parallel fading channels," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2486–2499, 2019.
[16] A. P. Nirmala, B. Meenakshi Sundaram and R. Prema, "An analysis on security threats in cloud computing," *International Journal Of Information And Computing Science*, vol. 6, no. 3, pp. 263–266, 2019.
[17] A. Kumari, V. Kumar, M. Y. Abbasi, S. Kumari, P. Chaudhary *et al.,* "CSEF: Cloud-based secure and efficient framework for smart medical system using ecc," *IEEE Access*, vol. 8, pp. 107838–107852, 2020.
[18] R. Sivan and Z. A. Zukarnain, "Security and privacy in cloud-based e-health system," *Symmetry (Basel)*, vol. 13, no. 5, pp. 742, 2021.
[19] B. Seth, S. Dalal, V. Jaglan, D. N. Le, S. Mohan *et al.,* "Integrating encryption techniques for secure data storage in the cloud," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, pp. 1–24, 2020.
[20] D. Dahanukar and D. Shelke, "A review paper on cryptography and network security," *International Journal of Advanced Research in Science, Communication and Technology*, vol. 12, no. 1, pp. 108–114, 2021.
[21] N. Andola, G. Nitish, Y. Raghav, V. Vijay, V. Sharannya *et al.,* "Searchable encryption on the cloud: A survey," *The Journal of Supercomputing*, vol. 78, no. 7, pp. 9952–9984, 2022.
[22] Q. Zhang, M. Fu, Y. Huang and Z. Zhao, "Encrypted speech retrieval scheme based on multiuser searchable encryption in cloud storage," *Security and Communication Networks*, vol. 2022, no. 10, pp. 1–14, 2022.
[23] P. Saraswat and S. Raj, "Encryption and decryption techniques in cloud computing," *International Journal of Innovative Research in Computer Science & Technology*, vol. 9, no. 6, pp. 225–228, 2021.
[24] S. A. Ahmad and A. B. Garko, "Hybrid cryptography algorithms in cloud computing: A review," in *2019 15th Int. Conf. on Electronics, Computer and Computation (ICECCO)*, Abuja, Nigeria, pp. 1–6, 2019.
[25] M. A. Al-Shabi, "A survey on symmetric and asymmetric cryptography algorithms in information security," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 3, pp. 576–589, 2019.
[26] F. Bentil and I. Lartey, "Cloud cryptography A security aspect," *International Journal of Engineering Research & Technology (IJERT)*, vol. 10, no. 5, pp. 448–450, 2021.

[27] A. Musa and A. Mahmood, "Client-side cryptography based security for cloud computing system," in *2021 Int. Conf. on Artificial Intelligence and Smart Systems (ICAIS)*, India, pp. 594–600, 2021.
[28] M. I. Mihailescu and S. L. Nita, "A searchable encryption scheme with biometric authentication and authorization for cloud environments," *MDPI-Cryptography*, vol. 6, no. 1, pp. 8, 2022.
[29] S. M. He, W. N. Zeng, K. Xie, H. M. Yang, M. Y. Lai *et al.,* "PPNC: Privacy preserving scheme for random linear network coding in smart grid," *KSII Transactions on Internet & Information Systems*, vol. 11, no. 3, pp. 1510–1532, 2017.
[30] K. Gu, L. H. Yang and B. Yin, "Location data record privacy protection based on differential privacy mechanism," *Information Technology and Control*, vol. 47, no. 4, pp. 639–654, 2018.
[31] K. Gu, W. J. Jia and C. L. Jiang, "Efficient identity-based proxy signature in the standard model," *The Computer Journal*, vol. 58, no. 4, pp. 792–807, 2015.
[32] K. Gu, L. H. Yang, Y. Wang and S. Wen, "Traceable identity-based group signature," *RAIRO-Theoretical Informatics and Applications*, vol. 50, no. 3, pp. 193–226, 2016.
[33] K. Gu, Y. Wang and S. Wen, "Traceable threshold proxy signature," *Journal of Information Science & Engineering*, vol. 33, no. 1, pp. 63–79, 2017.
[34] L. Y. Xiang, X. B. Shen, J. H. Qin and W. Hao, "Discrete multi-graph hashing for large-scale visual search," *Neural Processing Letters*, vol. 49, no. 3, pp. 1055–1069, 2019.
[35] Z. Xu, C. Xu, J. Xu and X. Meng, "A computationally efficient authentication and key agreement scheme for multi-server switching in WBAN," *International Journal of Sensor Networks*, vol. 35, no. 3, pp. 143–160, 2021.