Secure Text Transfer Using Diffie-Hellman Key Exchange Based On Cloud

L . Vijeeth Reddy, Mohd.Tajammul

PG student, Associate Professor, Department of MCA, Jain Deemed-To-Be-University, Bangalore, India
20mcar0059@jainuniversity.ac.in

Abstract—Cloud computing architectures are widely used in enterprises. It offers basic services, great network connectivity, flexibility, and more. However, the use of these features is difficult due to several security concerns. When someone views a text file, they don't get the impression that it contains hidden information. To compose the cipher text, the recipient needs a key. In this way, you can double-check that your private messages are being sent to hackers or crackers without outside influence. When the sender posts this text file, people will receive it without knowing what it is.

INTRODUCTION

The mechanism provides a key or password that allows the user to send a text through the secret message of the text file and upload the file to encrypt the content. Hacker, can't read the text. The receiver must decode the hidden text. Then send a text file and key to the recipient, first type the file, type the key or password to decrypt the text, and then press the decrypt key to give the sender the secret content. This approach allows you to confirm that confidential communications are shipped discreetly and are transported discreetly without outside influence from hackers or jailbreaks. When the sender transfers this text file publicly, no one knows what it is and the recipient will receive it.

PROBLEM STATEMENT

It's said that because the Cloud is designed to manage enormous volumes of data, bushwhackers might hope a significant bounty for their works.

Problem with the current situation

- ➔ There was no alike security approach used for hiding text data and transferring it securely to the receiver.
- ➔ By applying varied combinations, hackers would effortlessly break the security of the text communication.
- ➔ No similar secure path was used to transfer a text securely.

PROPOSED SYSTEM

➔ **Sender's Part:**

- Sender loads a text file which he/she wants to send
- Then he enters the text
- He sets the password for text and finally encrypts it.
- He saves the text file and sends it across to the receiver (Through email)

➔ **Receiver's Part:**

- The receiver opens the text file in the application.
- Enter the password which was used for encrypting (Password can be pre-decided or shared)
- After typing the password press Decrypt.
- Text will be represented in original format.

LITERATURE SURVEY

Character units use the standard functions of a character unit where the character units must be shaded in a secure manner (e.g., horse, encryption). Internally, two distinct methods are outlined. To begin, a (one-time-pad) is used to generate cypher text for each character's unit, providing a kind of blurring of both intangible and non-critical analysis of frequency. Secondary process develops the primary key by using coughing and squeezing technique which includes the fake kernels ("cereals") of valid cereals ("wheat") so that each coded wires are flippantly dispensed.[1]

For example, suppose we wish to encrypt the symbol. unit "AARON" within the data collection S. We implement a safe random key creator that utilizes alpha-numeric origins and is cryptographically secure:

"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 "and reset the keypad key" DO88Z0DLE7SZZI6ABAD4CHIJJ6PTYZUYZYKT".[1]

Outline of Chaff and Winnow

The second level of obfuscation is used in addition to the one-time pad. Borrowed language in the practice of "wheat

sowing" farming, the concept of coughing and crossing as a way to achieve privacy in messaging was first proposed by MIT computer scientist Ronald Rivest. Chaffing & Winnowing introduces a technique that do not use encryption keys instead uses "authentication" keys. A verification key allows the identification of valid bits from invalid bits of data. By using this information, the message can be sent with both valid and invalid parts and the recipient can remove invalid fragments with positive parts (cross races) to receive the message (wheat). Current system, (one-time-pad-chip) cipher can be "override" with farther characters, changing in the obtainment of a unit of characters with the same number of characters. [1]

AES (Enhanced Writing Level)

The most popular and widely accepted symmetric encryption algorithm that can be met today is the Advanced Encryption Standard (AES). It is available six times faster than DES three times.[3]

Exchange of DES is must because, as its origin size pales in comparison. With the expansion of PC capabilities, a risk factor for every search term is considered. The DES Triple was designed to overcome this evil but was found to be slow. [3]

The features of AES are as follows

Symmetric key symmetric block cipher

128-bit data, 128/192/256-bit keys

It is stronger and faster than Triple-DES

Provide complete specifications and design details

The software works on C and Java[3]

AES performance

AES is a duplicate of the Feistel cipher. It depends on 'exchange permission network'. It Includes a set of connected functions. Some of them include replacement and some of them include pushing pieces.[3]

Interestingly, the AES does all its calculations by bits instead of bits. Therefore, AES treats 128 bits of blank text as 16 bits. These 16 bytes are arranged in four (columns and rows) most likely (4*4) matrix.[3]

Unlike DES, number of cycles in AES differs and depends on length of the key. Each one of those spherical uses a unique 128-bit round key, calculated from the primary AES key.[3]

The Secret Writing Process

Here, we limit the definition of a standard AES encryption cycle. Each cycle consists of four sub-processes.[3]

Byte Conversion (Minor memory units)

The 16 input units are replaced by a stand-up table (S-box) provided for construction. It gives matrix of 4*4.[3]

Shift lines

Any 'falling' input is additionally inserted to the proper of the road. Changes area unit created as follows

The first line was not deleted.

The second row is moved one by one (byte) to the left.

The third is moved two positions to the left.

The fourth row ran three places on the left. The result is a new matrix consisting of 16 identical modified buttocks. .[3]

Assemble the Columns

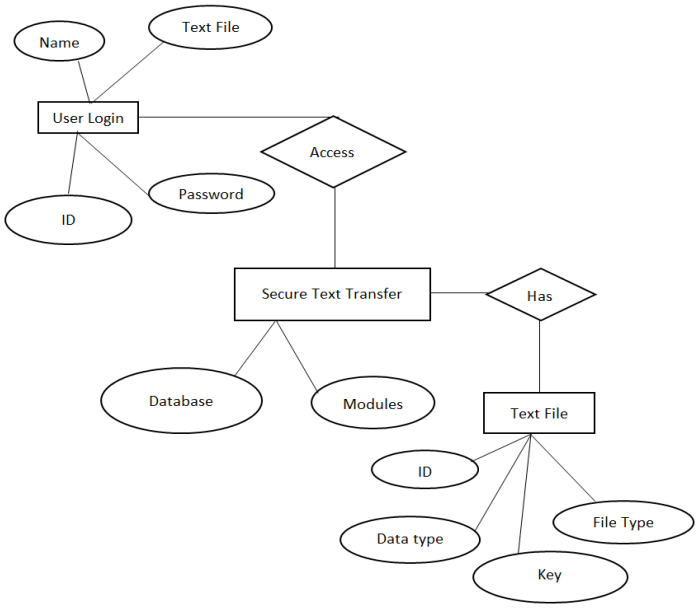
Each of the four to four columns is currently updated using a special function. This function takes the form of inserting four bytes into one column and then four new bytes to be cathartic, alternating the first column. A new matrix which consists of 16 new bytes will be created. Note: This current step will not be considered within the final spherical.[3]

Add a round key

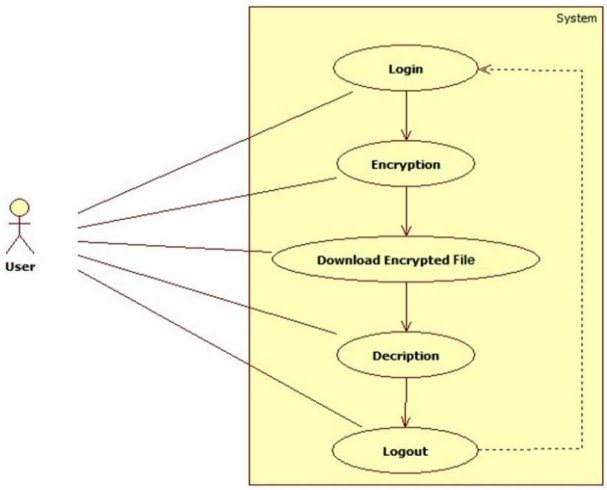
16 matrix byte is now taken 128 bits and XORed in 128 round keys. If this is the last cycle, that means ciphertext text. If that is not the case, 128 bits will be converted to 16 bits and we will prepare one more cycle same as this.[3]

SYSTEM ARCHITECTURE

➢ **E-R Diagram**



➢ **Use Case Diagram**



➢ **Data Flow Diagram**

DFD is a tool of graphic which is for describing and analysing data flow in a system. This will be the primary tool and the foundation from which various sections are constructed. The conversion of data from input to outbound, by processing, can be interpreted logically and without system-related components

. These are called sensible DFDs. Visual DFD's show the real implementation and data distribution between individuals, departments, and workplaces. The full description of the program contains a set of data flow diagrams. Similar writings Yourdon, Gane, and Sarson notation are used to improve DFD's. Each section in the DFD is marked with a descriptive name. The procedure is also identified by a number that will be used for diagnostic purposes. DFDs development is done at enormous levels. Every process in ground level diagrams can be divided into a more detailed DFD at the next level. Lop level drawing is often referred to as content drawing. It contains one process girl, who plays a key role in the study of the current system. The mechanism at the level of straight-forward content moves into another process at the first level of DFD.

The process of explosion which is caused by the view, in multiple processes is that influencing at one data level explodes which will cause more detail at the next level. This is done until further explosion is required and a sufficient amount of data is defined for analysts to understand the process.

Larry Constantine began developing DFD as a way to express system requirements in the form of an image, this led to the creation of a modular.

DFD is also known as the "bubble chart" and aims to identify system requirements and identify major changes that will be systems in system design. It is therefore the beginning of a project to a very low level of detail. DFD contains a set of bubbles which are connected to a data stream in a system.

➔ **DFD TYPES:**

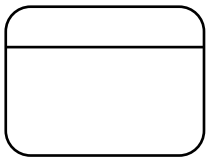
There are four forms of DFD.

1. The term square refers to a source (composer) or placement of data records.

2. The arrow represents the data stream. It is a conduit via which data passes.

3. The flow of incoming data into the flow of outgoing data is represented by circle or bubble which converts the Stream.

4. Open central data quadrate, default data repository, or momentary data repository



A process that transforms data flow.

Source or Destination of data

Data flow

Data Store

➔ **CONSTRUCTING A DFD:**

Several rules apply to the design of DFDs:

1. Course of action must be labelled and symbolled for effortless use. Every word must represent a procedure.

2. The management of the stream from top to bottom & left to right. Data traditionally flows from the source to the destination although it may flow back to the source. There is a unique way to present this, which is by designing a far flowing line back to the source. One way is to repeat the source sign as the destination. Used more than once in DFD it is marked with a short diagonal.

3. When the process is blown up to a minimum level, it counts.

4. The names of the data stores and destinations are capitalized. Process words and data flow have the first letter of each capital letter

DFD usually indicates minimal data store content. Each data store must contain all the features of incoming and outgoing data.

The questionnaire should contain all the incoming and outgoing data features. Decrease of lost workplaces and the like is frequently converse about it in talks.

IMPORTANT FEATURES OF DFD

1. DFD's mirrors data flow, it will not manipulate loops and decisions which are controlled by assumptions that will not arise from DFD.

2. DFD's do not mirror the time factor involved in any of the process whether data flow occurrence is daily (or) weekly (or) monthly (or) yearly.

3. The order of circumstances was not revealed in the DFD.

TYPES OF DATA TRANSPORT DIGGS

1. Current Physical

2. Reasonable Current

3. Sensible New

4. Fresh Meat

CURRENT BODY:

In current Physical DFD, process labels that include the names of people or their positions or the names of computer systems may provide other entire system processing labels that include identifying technologies used to process data. In the same way, data streams & data stores are frequently named with genuine media names which is where the data is stored in the form of file folders, computer files or computer tapes, etc.,

CURRENT FACTS:

The solid characteristics of the structure are separated based on the possibility so the existing structure is decreased to its central data and processors that converts it regardless of the authentic situation.

NEW LOGICAL:

This is exactly the same as the current logical model if the user is completely happy with the performance of the current system but has problems with the way it is used usually the new logical model will differ from the current logical model while having more functions, and complete removal functions. and visual dysfunction.

BODY NEWS:

The new portable represents the physical implementation of the new system.

LAWS GOVERNING DFD'S

PROCEDURE

1) No process can only output.

2) No process can have input only. If in case the object has only a single insert, that must be a sink.

3) Procedure has a work sentence name.

DATASTORE

1) Data cannot move directly from one data store to another data store, the process must transfer data.

2) Data can't mobile straight from an external source to the data store, process, recipient, first is must remove data from the source and place the data in the data store.

3) The data store has a name tag.

SOURCE OR GIVE

Source and / or location of data.

1) Data which is not able to flow straight from upstart to downfall must be processed.

2) The source and / or sink has a country name expression

DATA FLOW

1) Data Flow has only one way to flow between symbols. It may stream both directions between the process and the data store to reflect readings prior to review. End is normally specified by two distinct arrows as this happens in distinct class.

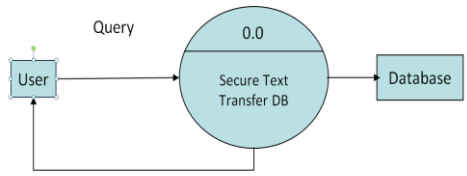
2) Combined DFD, it means the data output is same which comes from two or more different processes, data is getting stored or stored in the similar location.

3) The course of data cannot come back directly to the exact procedure it is leading. There should be at least one other process that handles data flow and generates another data flow that restores the original data to the original process.

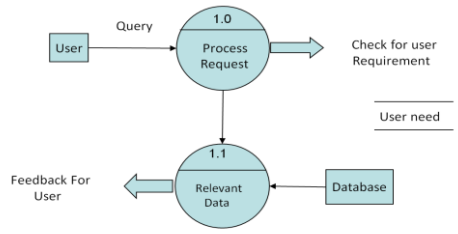
4) Course way of data to the data store which means update(replacement or deletion).

5) Flow of data from a data store means retrieval or use.

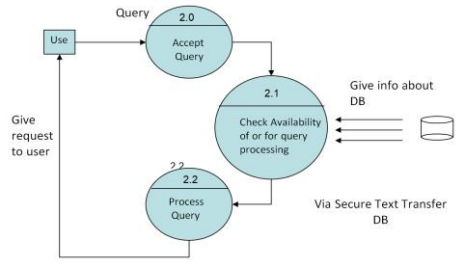
➢ **Data Flow Diagrams**



DATABASE DETAIL



LEVEL 1 DFD



LEVEL 2 DFD: PREDICTION

Implementation

➢ **OVERVIEW OF TECHNOLOGIES USED**

➔ **Front End Technology**

Microsoft .NET Framework

1. The .NET Framework is a new computer platform that facilitates application development in the most widely distributed internet environment. .NET Framework is a design that helps to attain the below mentioned goals:

2. Provide an object setting that is consistent with the object whether the object code is stored and used locally, locally used but still distributed online, or remotely used.

3. Provide a coding environment that minimizes software usage and version conflicts.

4. Provide a coding environment that ensures secure use of the code, including code created by anonymous or less trusted third parties.

5. Provide code deployment that eliminates written or translated translation problems.

6. Creating developer authentication for all types of applications, such as Windows-based applications and web-based applications.

7. Build all connections to industry standards to ensure that a code based on .NET Framework can integrate with any other code.

The .NET Framework has two main components: the working time of the common language and the .NET Framework classroom library. Normal working language time is the basis of .NET Framework. You can think of it while working as a code control agent during practice, providing essential services such as memory management, cable management, and remote control, while also enforcing sturdy type security and other forms of coding understanding that ensure security and durability.

The concept of code management is the basic goal of operating time. The code that directs the time to work is known as the managed code, while the code that directs the working time is known as the managed code. Classroom library, another key component of the .NET Framework, is a comprehensive, user-friendly component that you can use to upgrade applications from the traditional command line or visual interface (GUI) to the latest ASP-based applications. .NET, such as web forms and XML web services. The .NET Framework can be hosted by unmanaged components that load the normal operating time of their systems and implement the managed code, thus creating a software environment that can use both managed and uncontrolled features.

The .NET Framework not only provides a few working time managers but also supports the development of external working time managers. For example, ASP.NET handles operating time to provide a scalable location, located next to a managed code server. ASP.NET works directly with working time to enable Web Forms applications and XML Web services, both of which are discussed later in this article. Internet Explorer is an example of an unmanaged runtime application (in type of MIME type extension). Using Internet Explorer to handle workflows enables you to embed controlled sections or controls of Windows Forms into HTML documents. Managing operating time in this way makes the coded mobile code (similar to Microsoft® ActiveX® controls) feasible, but with significant improvements that can only be managed with the coded code, such as less reliable use and more secure file storage. The following diagram illustrates the relation between the start time of a common language and class library on your apps and the whole

system. The design also presents how the managed code works inside a large structure.

➔ Practical features of working language for a common language In terms of security, managed components are rewarded with varying levels of reliability, depending on a number of factors including their origin (such as the Internet, business network, or local computer). This means that the managed component may or may not be able to perform file access tasks, register access tasks, or other sensitive tasks, even if used in the same operating system. The operating time emphasizes the security of access to the code. For example, users may expect that a usable object embedded in a Web page can play animation on the screen or sing a song, but it cannot access their data, file system, or network.

Operating time features allow official software installed on the internet to become incredibly rich. Operating time also exploits the strength of the code through a solid type- and code verification infrastructure called a common type system (CTS). CTS ensures that all managed code is self-explanatory. Various Microsoft language integrators and third parties produce managed COD compliant code. This means that coded code can use other manageable types and conditions, while strictly enforcing the type of reliability and security type. Operating time also speeds up engineer production. For example, program planners can write applications in their preferred development language, but make full use of working time, classroom library, and sections written in other languages by other developers. Any affiliate marketer who chooses to specify a working time may do so. The language coordinators who direct the .NET Framework make the .NET Framework features available in existing code written in that language, making it much easier to streamline existing applications. While operating time is designed for future software, we also support current and past software. The interaction between managed and unencrypted code enables developers to continue to use the required COM and DLL component components. Working time is designed to improve performance. Although the working language of the standard language provides many of the standard operating time features, the managed code is not interpreted. A feature called just-in-time (JIT) integration allows all managed code to run in the native language of the operating system. At the same time, the memory manager removes the possibility of different memory and enhances location-reference memory in order to further increase performance.[1-2]

➔ .NET Framework Class Library

The classroom library focuses on the object, providing the types in which your managed code can find functionality. This not only makes the .NET Framework types easier to use but also reduces the time associated with learning new .NET Framework features. In addition, third-party components can easily integrate with classes in the .NET Framework. [1]

For example, NET Framework collection classes use a set of visual connectors that you can use to enhance your collection classes. Your collection classes will easily integrate with the classes on the .NET Framework. [1]

As you will be expecting from an object-oriented library, the .NET Framework types allow you to complete a range of standard editing tasks, including tasks such as character unit management, collection of data, connectivity of website, & accessing of files. [1]

In addition to these general activities, the classroom library includes genres that support a variety of special development scenarios. For example, we [1]

can use .NET Framework as follows:

- Console applications.
- Written or hosted requests.
- Windows GUI applications (Windows Forms).
- ASP.NET applications.
- XML web services.
- Windows Services.

Windows Forms objects, for instance, are a full collection of customizable types that makes Windows GUI creation considerably easier. You may include Web-Forms classes while applying for an ASP.NET Web Form. [1]

➔ Client Application Development

Client apps are very close to the traditional Windows-based application style. These are apps that display windows or forms on the desktop & allow users to do operations. Client apps contain demands like word processors and spreadsheets, as well as specific business operations like input tools, reporting tools, etc. Client apps often make use of windows, menus, buttons, and other graphical user interface (GUI) elements, and they may have access to local services such as a system files and accessories such as printers. [1]

Another type of client application is the standard ActiveX controller (now replaced by the existing Windows Forms controller) installed on the Internet as a Web page. This app is fairly similar to previous client apps in that it is individualized, has access to local resources, and incorporates features of an image. [1]

In the past, engineers have created such programs using C / C ++ in partnership with Microsoft Foundation Classes (MFC) or in the fast-moving environment (RAD) environment such as Microsoft® Visual Basic®.[1]

The.NET Framework combines the capabilities of these technologies into a single, uniform development platform, making it easier to create client applications. [1]

Windows Forms classes contained in the .NET Framework are designed for GUI development. we can easily create windows command, buttons, [1]

According to shifting business requirements toolbars, menus and other screen elements will be customized. [1]

For example, the .NET Framework provides easy features to customize the visual attributes associated with the forms. In some circumstances, the core operating system will not allow for direct changes to these characteristics, in which situation the.NET Framework re-creates forms automatically. This is one of the many ways in which the .NET Framework integrates with the developer's visual interface, making code writing easier and more compatible. [1]

Unlike ActiveX controls, Windows Forms controls have less reliable access to the user's computer. This means that binary or native signature code can access other resources on the user's system (such as GUI objects and limited file access) without being able to access or compromise other resources. [1]

Because of the security of code access, many applications that once needed to be installed on a user's system can now be safely used by the Web. Your apps can use local application features while they are being distributed as a Web page. [1]

➔ Server System Development

Applications that are next to the server in the host world are used by operating moderators. Unmanaged applications limit the start time for standard language, allowing your customized code to control server behavior. This model provides you with all the features of working class language and classroom library while you get the functionality and robustness of the host server. [1]

The diagram below depicts the fundamental network design with code that works in many server contexts. When your app is executing encrypted code, servers such as IIS and SQL Server can continue to operate normally. [1]

➔ Server code on the server side

ASP.NET is a webhosting technology that lets developers to administer web-based applications using the.NET Framework. However, ASP.NET is more than just a host of working hours; is a complete framework for the development of sites and products distributed online using managed code. Both Web Forms and XML Web Services use IIS and ASP.NET as a publishing tool for applications, and both have a set of support classes in the .NET Framework.[2]

CONCLUSION

This was my research into designing a solution for "Secure Text Transfer" application depends on Asp.Net using C#, i.e. Web Application. Development of this System took a lot of effort. I think this system gives me a little bit of satisfaction. However, no work in the order to create and develop is ever said to be faultless, therefore more improvement in this app is possible. I learned so many things and gained a lot of knowledge about the development field.

Features

Load Rate:

Since the system will only be available when the administrator enters the load value on the server will be limited to the administrator access time.

Easy Access:

Records can be easily accessed and stored as other information in sequence.

Applicable:

For all users, the system will give an easy-to-use way.

Effective and reliable:

Maintaining all secure data on a server that will be accessible to the user's requirement without any cost of repairing it will be very effective compared to storing all customer data in a spreadsheet or physically in the record books.

Easy fix:

Secure Text Transfer is designed as an easy way. So maintenance is also easy.

REFERENCES

1. Microsoft Developer Network (MSDN): http://msdn2.microsoft.com/en-us/default.aspx: This is a valuable online resource, and is a must for any developer using Microsoft tools.

2. http://www.asp.net/: This is the official Microsoft ASP.NET web site. It has a lot of: tutorials, training videos, and sample projects.

3. Clemens Zeidler, Muhammad Rizwan Asghar "AuthStore Password-based Authentication and Encrypted Data Storage in Untrusted Environments",2018.

4. http://ijics.com/gallery/58-may-1148.pdf

5. https://www.researchgate.net/publication/317339928_A_study_on_diffie-hellman_key_exchange_protocols

6. https://www.researchgate.net/publication/276232372_A_New_Three-party_Key_Exchange_Protocol_Based_on_Diffie-Hellman

7. https://www.researchgate.net/publication/321947600_The_Generalized_Diffie-Hellman_Key_Exchange_Protocol_on_Groups

8. https://www.researchgate.net/publication/289101681_Modification_of_Diffie-Hellman_Algorithm_to_Provide_More_Secure_Key_Exchange

9. https://link.springer.com/content/pdf/10.1007%2F3-540-48892-8_26.pdf

10. Alam T., Tajammul M., Gupta R. (2022) Towards the Sustainable Development of Smart Cities Through Cloud Computing. In: Piuri V., Shaw R.N., Ghosh A., Islam R. (eds) AI and IoT for Smart City Applications. Studies in Computational Intelligence, vol 1002.

11. Tajammul, M., Shaw R.N., Ghosh A., Parveen R. (2021) Error Detection Algorithm for Cloud Outsourced Big Data. In: Bansal J.C., Fung L.C.C., Simic M., Ghosh A. (eds) Advances in Applications of Data-Driven Computing. Advances in Intelligent Systems and Computing, vol 1319.

12. Tajammul, M, Parveen, R., "Cloud Storage in Context of Amazon Web Services", International Journal of All Research Education and Scientific Methods, vol. 10, issue 01, pp. 442-446, 2021.

13. Tajammul, M., Parveen, R., "Auto Encryption Algorithm for Uploading Data on Cloud Storage", BIJIT - BVICAM's International Journal of Information Technology, vol. 12, Issue 3, pp. 831-837, 2020.

14. Tajammul, M., Parveen, R., "Key Generation Algorithm Coupled with DES for Securing Cloud Storage," International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-5, June 2019 no. 5, pp. 1452–1458, 2019.

15. Tajammul M., Parveen R., "Two Pass Multidimensional Key Generation and Encryption Algorithm for Data Storage Security in Cloud Computing", International Journal of Recent Technology in Engineering, Vol. 8, Issue-2, pp. 4152–4158, 2019.

16. Tajammul M., Parveen R., "Algorithm for Document Integrity Testing Pre-Upload and Post- Download from Cloud Storage", International Journal of Recent Technology in Engineering, Vol. 8, Issue-2S6, pp. 973–979, 2019.

17. Tajammul, M., Parveen, R., "Auto Encryption Algorithm for Uploading Data on Cloud Storage", BIJIT - BVICAM's International Journal of Information Technology, vol. 12, Issue 3, pp. 831-837, 2020.

18. Tajammul, M., Parveen, R., and M. Shahnawaz, "Cloud Computing Security Issues and Methods to Resolve: Review," Journal of Basic Applied Engineering and Research, vol. 5, no. 7, pp. 545–550, 2018.

19. Tajammul, M., Parveen, R., Delhi, N. (2018). Comparative Study of Big Ten Information Security Management System Standards, International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 5, Issue 2, pp. 5-14, 2018.

20. M. Tajammul, R. Parveen, N. K. Gaur and S. D, "Data Sensitive Algorithm Integrated with Compression Technique for Secured and Efficient Utilization of Cloud Storage," 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), 2021, pp. 1-9, doi: 10.1109/GUCON50781.2021.9573648.

21. Tajammul, M., Parveen, R., (2017). Comparative Analysis of Big Ten ISMS Standards and Their Effect on Cloud Computing, 978-1-5386-0627 8/17/31:00c2017IEEE; 9001; 362367.

22. Tajammul, M., and R. Parveen, "To Carve out Private Cloud with Total Functionality," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020, pp. 831-835, doi: 10.1109/ICACCCN51052.2020.9362826.

23. M. Tajammul, R. Parveen and I. A. Tayubi, "Comparative Analysis of Security Algorithms used in Cloud Computing," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021, pp. 875-880, doi: 10.1109/INDIACom51348.2021.00157.