Password Security using Mutation Technique in Cloud Computing

Agraj Magotra, Akansh Srivastava, Heena, Navjot Singh

ABSTRACT

In the past few decades, Cloud computing is playing a significant role in the industry, whether it's the IT sector, Health sector or supply chain management. As a result of its easy access, many people are going towards Cloud. Hence, this demand increases the existence of a new cloud service provider, allowing end-users to store and share their private and sensitive data. However, the biggest challenge in Cloud Security is the customer's adoption and prevention of its services. Therefore, several security measures have been adopted by the cloud service provider. This paper focuses on providing an advanced level of password security. We have presented a password security technique called the string mutation technique to prevent it from unauthorized access. Also, we have implemented the md5 hashing technique to encrypt the password before storing it in the database. We have used the .net platform to implement this technique. In the result section, we have shown the executed result.

Keywords: *Cloud Security, Password Encryption, MD5 hashing, Data security*

I. INTRODUCTION

In recent times, Cloud computing effects business. Cloud computing not only sticks to IT Industry but also in other industries, like healthcare or manufacturing. With developing name, an ever-increasing number of organizations are using Cloud system [1]. Even though cloud management has far and wide support, the dread relating to the Security and protection of these system keeps on being an open challenge. These systems could be effectively removed through PDAs with quick, innovative progressions, permitting clients to share video, reports, pictures and other significant information in different platforms and in-house [2]. In any case, a security breach in their cloud data could prompt taken information which would, in reality, bring about massive disasters. Security has consistently been a worry in the space of data innovation. Security is a conspicuous concern with Cloud management taking care of essential information from any place through the web [3]. The inevitable thought of the Cloud and its disbursement of information around different topographical areas adds up to higher chances of security. Whereas discussing Security of Cloud, numerous perspectives must be considered, such as disclosing sensitive data, valid approval, information security, and protection. These are a few fundamental security objectives that are very important for each cloud supplier to consolidate [4]. Since Security has been viewed as a trait for data innovation, information encryption has been a vital measure in guaranteeing information security insurance. The different algorithm in the past has been proposed for driving capable data encryption. This algorithm is RSA, DES to AES, 3DES, Diffie-Hellman, and RC4. Every one of these algorithms enjoys its benefits besides its faults. These algorithms are extensively appointed, being symmetric or unbalanced. Our concentration here is to make a Cloud Security Network that influences both asymmetric and symmetric encryption advantages. We use AES (Symmetric) algorithm for doing file encryption and string manipulation for securing password. We target establishing an extensive Cloud Environment with safety efforts from making and putting away username and password, multifaceted support, the transmission of client information and encryption of data. The remaining of the research is structured as below: Section II discusses issue of cloud safety. Segment III clarifies the proposed work wherein the proposed system and its

working are clarified. Section IV describes the algorithm that defines the work process of the whole framework, though its helpful reenactment and its outcomes are examined in Section V. At last, concluded the paper in Section VI.

II. ISSUE IN CLOUD SAFETY

As we know data on the cloud are managed and processed. The security responsibility is on the cloud service providers. This creates belief and faith between the CSP and the customer. It's CSP responsibility to provide security from attacks and privacy to end-users data. Some of the points are given which needs to provide by the CSP to end-users data

Safeguarding of Data: Cloud platforms attract lots of security breach at the client, CSP and brokers' end.

Various SLAs are involved between the cloud users, providers and agent, indicating specific information theft. It is normally seen that it gets hard for the cloud client to check the information dealing with practices of the cloud supplier [5]. Further, there can be difficulties because of the unpredictable organization geography between the cloud and the end client, extending many organization-related attacks.

Data Loss: In some cases, the application not uses proper encryption while data were transferring, which causes data exposure on the browser. Some of the common mistakes are SQL injection and XSS.
However, some application shares there same application pool which also causes security breach. Cloud service providers like Azure and Aws have a Tags feature, allowing users to access their instance through a remote desktop. This feature also has some limitations; if the user enables it to all IPs, users can access that instance from anywhere if they get a pem file and IP address.

Hijacking of Traffic: is moreover one of the weaknesses that end customers face while chipping away at distributed computing. It was positioned number 3 as the most weak assault by the cloud specialist co-op in 2013. In such an assault, programmers will, when all is said in done, gain a customer's security affirmations and open unapproved admittance to its information. After which, all the customer tasks, including its mysterious trades happening on the cloud, are by and by open to a developer [6]. The programmer can adequately utilize the customer's data and access its applications running on the cloud. A comparable assault was found by amazon on year 2010, when a programmer parodied the meeting id and gain admittance to the client's records.

Resource Sharing: In a recent scenario, the two fundamental features of cloud computing are multiple users and resource sharing. This vulnerable class considers measures works and managing resources like management, memory, data transmission, and even standing between various occupants. Thus, the cloud gives a distinct stage to various sorts of uses from different clients. Moreover, this shared asset pool adds security issues, making the client information more helpless against information in fragmentation.

Vindictive Insider: Generally, it is seen that the damage caused by malicious insiders is significantly higher than was expected. These attackers use their device to inject the code into any site. In addition, there comes into the picture an Ip jumping technique. In this technique, hackers switch their Ip address from time to time so that other authentic users cannot track its Ip. After that, the malicious user can quickly get the file access.

III. PROPOSED WORK

In the last few decades, many security techniques have been proposed related to cloud computing. But all method has their limitations. Some methods focus only on Cloud security; some are related to data breaching prevention technique; some are related to unauthorized access. No approaches are proposed which can handle all these problems in one solution. Our proposed method focused on the password security technique and a data security technique using various algorithms in one application, enhancing Cloud security data and user privacy.

A.Password Mutation Technique

In this Method, We ask users to register themselves in our application. While recording, the user has to enter the Username, Email, and three passwords like password1,

password2, password3 when the user registers, but the password must be six characters. Then, we mutate the password and store it in the database; we apply the MD5 hashing algorithm to hash the plain text before storing it in the database. The mutation is the technique in which we interchange the password characters from each other, like character one of password1 with character 1 of password 2, character 2 of password 2 with character 2 of password three and so on till 6th character, which generates the new password.

B. File Encryption using AES

The more acclaimed and for the most part accepted symmetric encryption computation obligated to be capable these days is the Advanced Encryption Standard (AES). It is multiple times quicker than triple-DES in terms of time and memory consumption.

The need for a new algorithm over DES was due to the size of the key. DES key size is very small in length. With growing figuring control, it was considered defenseless against comprehensive key pursuit attack. Triple DES was intended to overcome its limitation; however, it is slow in n nature and consumes more memory.

The main features of AES are as follows −

- Cipher block is Symmetric
- Bit sizes are 128/192/256
- Faster and unbreakable than Triple-Des

AES is a Feistel cipher rather than iterative. Therefore, it depends on a 'replacement stage organization'. It's anything but a progression of related activities, some including supplanting contributions with detailed results, and others include rearranging pieces around (changes).

Ironically, AES plays out the whole of its estimations on bytes rather than bits. Thusly, AES treats 16 bytes which are 128 pieces of plaintext. The course of action measures as a network that comprises 16 bytes with four lines and four segments.

As conpared to DES, the AES number of the round is variable and depends upon the length of the key. AES utilizes 10 rounds for 128-digit keys, 12 rounds for 192-piece keys and 14 rounds for 256-cycle keys. All of these rounds use another 128-digit round key decided from the main AES key.

C. System design

In this section, we are discussing our proposed system. The proposed system consists of how we are designing our overall system to protect the user password and its files in our application using AWS cloud using windows instance. The below figure explains the overall process of our work.

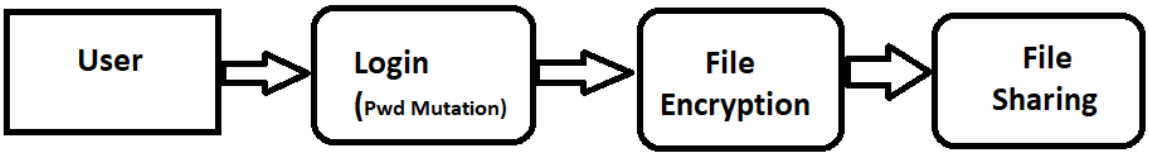

Fig 1: System design of Our Proposed Work

Authenticated Entry: An authentic person can enter our application if they are willing to use our service. They need to register by inserting username, Email, and three passwords.

Password Mutation: The system uses password mutation technique. Applying this technique can enhance the password security.

Hashing: MD5 hashing technique is applied to add extra layer of security to securing of generated password.

File Encryption: The system uses AES Encryption to encrypt the user's files. Before storing the file on our cloud server, we prompt the user to encrypt the file. We have hardcoded the key in our program. DotNet framework has a class called Rijndael Managed which has properties like Key KeySize, BlockSize and IV. With the help of this class we have encrypted and decrypted the file.

IV. PROPOSED ALGORITHM

Our proposed work explains through the outline of the calculation that shapes the center for it. The analysis portrays the working of the framework by addressing the whole interaction from client validation to capacity and recovery of client information from the Cloud.

STEP 1: Create an Account with Email, Name, Password1, Password2 and Password3
STEP 2: Password Generate using String Mutation. Suppose we have three passwords like 123456, 234567, and 345678.
The strings Mutation will mutation each character.
STEP 3: Applying MD5 hashing Technique in the new generated password.
STEP 4: Enter login credentials by inserting Email and three passwords like Password1, password2 and password3.
STEP 5: Converting the password in MD5 hashing to match it with the password stored in our database.
STEP 6: List all the files stored in cloud-based on logged-in user.
STEP 7: Transferring the user's data over the network we use FTP and TLS protocol.
STEP 8: AES encryption algorithm is used to store secure file on the cloud.
STEP 9: The user can share the files.
STEP 10: other users view the shared files.
STEP 11: The other user download and decrypts the file.

Step 1 to Step 5 explains the verification process in which users register themselves using Email, contact, and three passwords. The string mutation occurs, which is explained in the below figure.

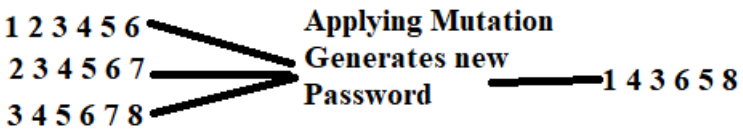

Fig 2: String Mutation

In Step 6, the User will view all the list of files that they had uploaded earlier. In Step 7 and 9 User can share the files with other users. The files which are being shared must be encrypted. For encrypting the file, we are using AES Encryption. In Step 10 and Step 11, the User can view and download the files.

V. IMPLEMENTATION

The proposed system is implemented in Asp.net with c# using the Dotnet framework, which has many predefined classes. With the help of these classes, we have implemented password mutation. For string mutation, we used the string array and split function provided by this framework.
For MD5[8] hashing, we used the MD5 class, which is having the function Create and computehash. The computehash function takes a string as a parameter which, in our case, mutated password. This function returns a byte array. After getting the return value as a byte array from this function, we append each byte using the Append function of StringBuilder class using for loop this creates a new hash password. This hash password, including email and name, are stored in the database.
We apply the same technique to match the password when the user is trying to log in. Once the user is

successfully logged in, they can view the files list; this is done by using the Entity Framework technique, which is ORM provided by the Dotnet framework.

For the file encryption, the Dotnet has an Aes class which is having created function.

Rfc2898DeriveBytes [12] is a constructor that takes parameter as encryption key and byte length. This class has a property called key that helps to set the key of the byte. FileStream is a class that takes path and mode. Path defines where files to encrypt and mode defines whether to create new or open. In our case, it is to create the file in a specific location. CryptoStream is a class used to create the encrypted file and create the file where the user is willing to save it. In the file-sharing section, we use file upload control available in asp.net. This control has a function called saveas. This function takes the parameter as the location where we want to store the file on the server. This helps the user to upload the files to the server folder. In the download section, we get the list of files that other users have shared. Again for downloading the files, we used FTP protocol to download the files.

This way, we have implemented overall technique.

VI. RESULT

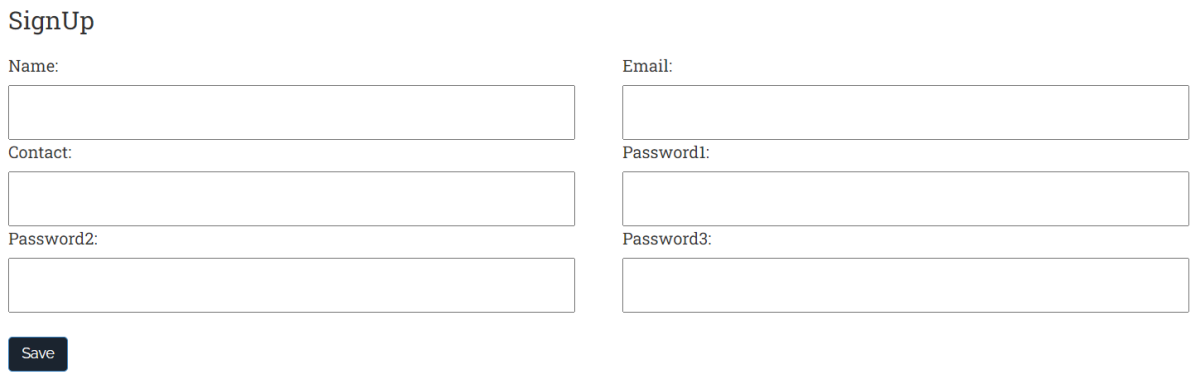

Fig 1: Registration page

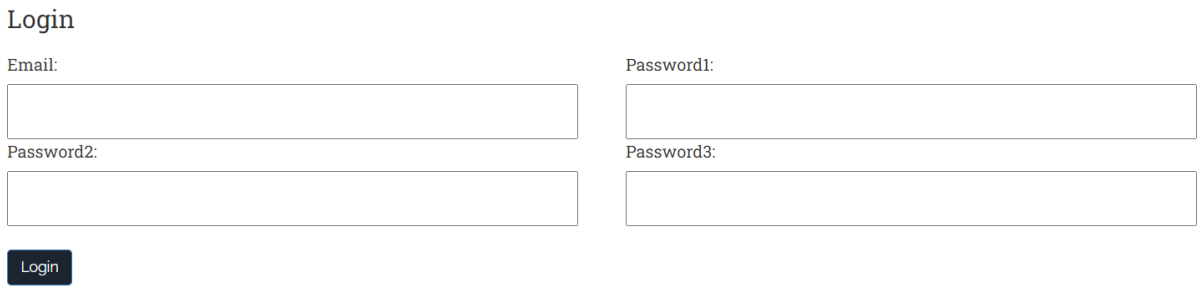

Fig 2: Login Page

VII. CONCLUSION & FUTURE WORK

In this paper, we have proposed a password mutation technique that enhances the user's password. We also implemented the MD5 hashing technique and encryption technique, which increases the security layer of password and file storage.

In the future, we can implement some other enhanced encryption algorithm and remove the limitation of the password length from six characters to higher. We also try to implement the different hashing technique and allow users to apply encryption of their choice.

REFERENCES

[1] Subashini, Subashini, and Veeraruna Kavitha. "A survey on security issues in service delivery models of cloud computing." Journal of network and computer applications 34.1 (2011): 1-11.
[2] Pawar, Pramod S., et al. "Security-as-a-service in multi-cloud and federated cloud environments." IFIP International Conference on Trust Management. Springer International Publishing, 2015.
[3] Nair, Nikhitha K., K. S. Navin, and Soya Chandra. "Digital Signature and Advanced Encryption Standard for Enhancing Data Security and Authentication in Cloud Computing." (2015).
[4] Wang, Cong, et al. "Privacy-preserving public auditing for data storage security in cloud computing." INFOCOM, 2010 Proceedings IEEE. Ieee, 2010.
[5] Hendre, Amit, and Karuna Pande Joshi. "A semantic approach to cloud security and compliance." 2015 IEEE 8th International Conference on Cloud Computing. IEEE, 2015.
[6] Khanna, Abhirup, Sarishma. (2015). Mobile Cloud Computing: Principles and Paradigms. IK International. 2017
[7] Khanna, Abhirup. "RAS: A novel approach for dynamic resource allocation." Next Generation Computing Technologies (NGCT), 2015 1st International Conference on. IEEE, 2015.
[8] https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.md5?view=net-5.0
[9] Huang, Wei, et al. "The State of Public Infrastructure-as-a-Service Cloud Security." ACM Computing Surveys (CSUR) 47.4 (2015): 68. [10] Aich, Asish, Alo Sen, and Satya Ranjan Dash. "A Survey on Cloud Environment Security Risk and Remedy." Computational Intelligence
and Networks (CINE), 2015 International Conference on. IEEE, 2015.
[11] Singh, Aarti, and Manisha Malhotra. "Security Concerns at Various Levels of Cloud Computing Paradigm: A Review." International Journal of Computer Networks and Applications 2.2 (2015): 41-45. 292
[12]https://docs.microsoft.com/enus/dotnet/api/system.security.cryptography.rfc2898derivebytes?view=net-5.0