Data Integrity, Security and Privacy in Healthcare Management System

Ketaki Deshmukh[1] Prof. Pramila M. Chawan[2]
[1]M. Tech Student [2]Associate Professor
[1,2]Department of Computer Engineering & IT
[1,2]VJTI College, Mumbai, Maharashtra, India

Abstract— The proposed research paper aims to find the most secure means to maintain the integrity and ensure privacy and security in Healthcare Management System. Internal database constraints and restrictions ensures integrity of the data. Moreover, Cross Object Resource Sharing (CORS) filters the devices which can access the project. Data privacy will be ensured by safeguarding sensitive data using International Data Encryption Algorithm (IDEA) and Advanced Encryption Standard (AES). JWT Signature Algorithm HS256 ensures Session security. Password safety is provided through SHA256 along with salt. In this article, the focus will be on implementing IDEA cryptographic algorithm for data privacy and understanding its pros and cons over AES-256 encryption algorithm implementation.

Keywords: Cross Object Resource Sharing, International Data Encryption Algorithm, Advanced Encryption Standard, JWT Signature Algorithm HS256, SHA256 Hashing Algorithm, salt

I. INTRODUCTION

The business environments in which the current businesses operate are challenging the traditional Requirements Engineering approaches [1]. Electronic medical reports of the patient provided by the doctor form the electronic health record, and it comprises all the critical directorial medical data that is with regard to the care given to a patient by a specific provider such as demographic details, medical reports, issues, medications and drugs, crucial observations, medical history, allergy reports, laboratory results and radiology data. Medical data is the most sensitive data on the web and is still ignored when it comes to securing it. Privacy and security breaches are still common in this field even after having numerous policies and guidelines [11]. Electronic healthcare data is quite prone to all types of attacks from outside as well as within the organization [5]. Patient data privacy is effectively protected by administering a combination of techniques and efficient methods of de-identification of critical data, data access control and security methods for numerous technical issues. To maintain the relevancy, electronic health record system must suffice particular necessities such as data consistency, failure management, availability and be consistent to privacy strategies. As of now, there are many concerns regarding data privacy and integrity of secured healthcare data and these concerns are the biggest hurdles in implementing EHR; and hence there's a need for healthcare management system.

II. RELATED WORKS

Privacy and Integrity policies must be designed and implemented publicly. Despite much privacy and safety defiances, numerous techniques have already been executed by the healthcare domain. Many privacy and integrity frameworks are already available, and we must anchorage the available processes as we utilize these standard field of healthcare infrastructure. Fig. 1 shows an overview and types of these methods. However, the widespread use of electronic health records and medical data warehouses, as well as the recent attacks on patient sensitive data have reinforced the necessity of a new era of data privacy and integrity methodologies. The huge amount of healthcare data from numerous sources is stored scattered across various destinations and distributed systems. Security breaches in any of these systems can cause the leak of sensitive data to unauthorized intruders and manipulations [4]. EHRs additionally have issues in keeping up information security [4], to the extent that authoritative staff can for instance access information without patient consent [5]. As privacy and security of these areas has been the most important aspect in the design, execution and maintenance of the shared prototype, we integrated an information source that looks at the clinical applications and the underlying data warehouse of Houston Methodist's METEOR [6] that gives faster alerts to problems and is crucial to timely determination and minimization of the scope of damage.

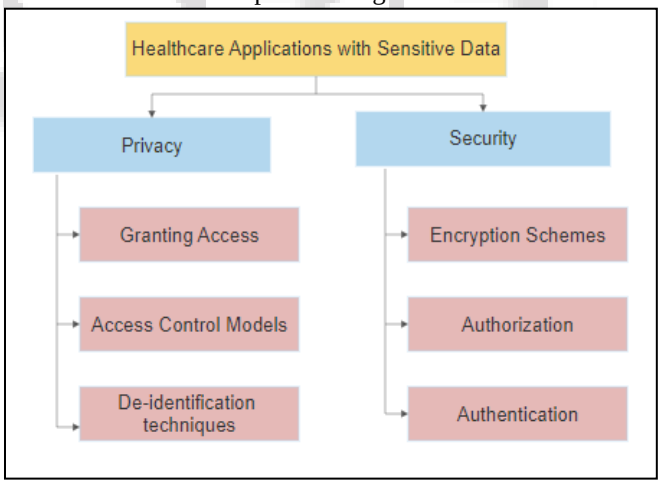


Fig. 1: Classification of privacy and security approaches

III. PROPOSED SYSTEM

A. Problem Statement

"A huge amount of sensitive data like mobile numbers, email ids, addresses, dependent records, credit card details, insurance records and critical medical data of the patients is at stake once it's out on the web. Such data induce several attacks like identity theft, financial frauds and leak of sensitive private medical and personal data. Human organ trafficking rackets are in continuous search of such sensitive patient data to carry out their crimes. Our project intends to safeguard the integrity and privacy of patient data using internal database security by implementing SHA256 along

with IDEA and compare it with the security and privacy that AES provides."

B. Problem Elaboration

In the proposed project, internal database constraints and restrictions ensure safety of critical patient data. It is achieved by providing a public and private key to the patient. Patient public key will be known to everyone but, in the database patient private key will be referred to everywhere. This will ensure that even if an intruder gets access to the critical patient data, he won't be able to know exactly whose critical data he has acquired. CORS (Cross Object Resource Sharing) configuration will be used for filtering the hosts accessing the project. Patient sensitive data will be safeguarded using International Data Encryption Standard Algorithm (IDEA) and Advanced Encryption Standard (AES). Session will be secured using JSON Web Tokens which internally uses Signature Algorithm HS256. Passwords will be safeguarded using SHA256 Hashing Algorithm with salt. Electronic Health Record is most vulnerable data yet it's the least secure data on the web. Recent attacks on the EHR have made securing this data even more important. The proposed project effectively fulfills this criterion.

C. Proposed System Architecture

Requirement's engineering is a step-by-step process by which a vague statement of needs is converted to system component requirements and even subsystem requirements and the corresponding models that are formed in the process form the basis for a detailed model further down the process stream in the requirements engineering process. Also, the requirements can help in modularization of the processes for the development of the software project, so that non overlapping processes can be developed in parallel without crossing over each other [1]. Fig -2 shows the multiple layers of security which proposed system provides. In case of failure of one layer, few others will ensure data privacy and integrity in the project.

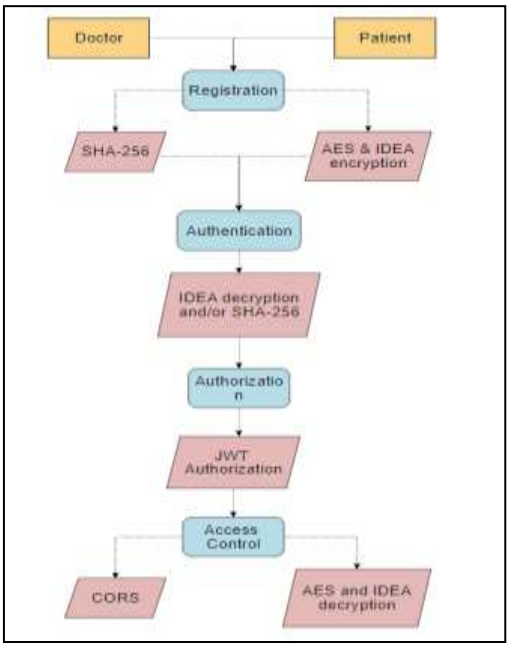


Fig. 2: Healthcare Management System Architecture

D. Proposed Methodology

1) SHA-256 Hashing Algorithm

A cryptographic hash, also called as a digest, is a type of signature for a text or a data file. SHA-256 outputs an almost-unique 256-bit (32-byte) signature for a text. Encryption and hash are not the same concepts. A hash cannot be decrypted back to the original text as in the case of encryption. SHA-256 is one of the strongest hashing functions available [10].

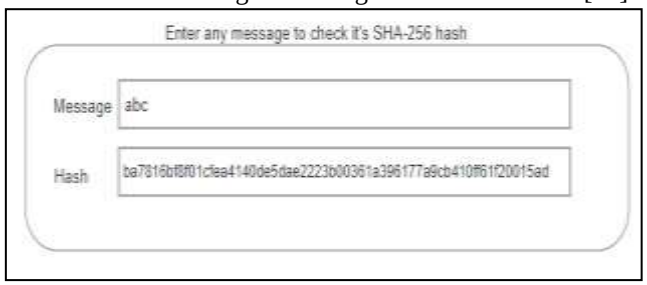


Fig. 3: Sample Hash Code

2) AES Cryptographic Algorithm

As shown in Fig -4, a 128-bit plaintext block is given as an input to the AES algorithm. It then performs numerous rounds of transformations to generate the output encrypted data, also known as the cipher text. Each 128-bit data block is processed in a four-by-four array of bytes, called the *state*. The *roundkey* size can be 128, 192 or 256 bits. The number of rounds repeated in the AES, *Nr*, is defined by the length of the key, which is 10, 12 or 14 for key lengths of 128, 192 or 256 bits, respectively. Four basic transformations in AES are shown below [2] [7].

1) *SubBytes:* Each byte *state* [*i, j*] in the *state* matrix is replaced by the value of *S-Box* (*state* [*i, j*]), where *S-box* is a 16-by-16 array of bytes substitution box. The *S-box* is computed before the AES encryption by deriving from the multiplicative inverse over *GF* (28), which is a finite field known to have good non-linearity properties.
2) *ShiftRows:* In this step, we cyclically left shift every row *i* of the state matrix by *i*, $0 \leq i \leq 3$.
3) *MixColumns:* In this step, the four bytes of each column of the *state* array are combined using an invertible linear transformation. Then we multiply each column of the *state*, taken as a polynomial of degree below 4 with coefficient in *GF*(28), by a fixed polynomial modulo $x4+1$.
4) *AddRoundKey:* In this step, the *r*-th *roundkey* is added by combining each byte of the *state* with the corresponding byte of the *r*-th *roundkey* using bitwise XOR. As a result, *roundkey*s can be calculated before the encryption process, and kept constant during encryption time.

During the decryption phase, the ciphertext can be transformed back into the original plaintext by applying a set of reverse rounds using the same encryption key [7].

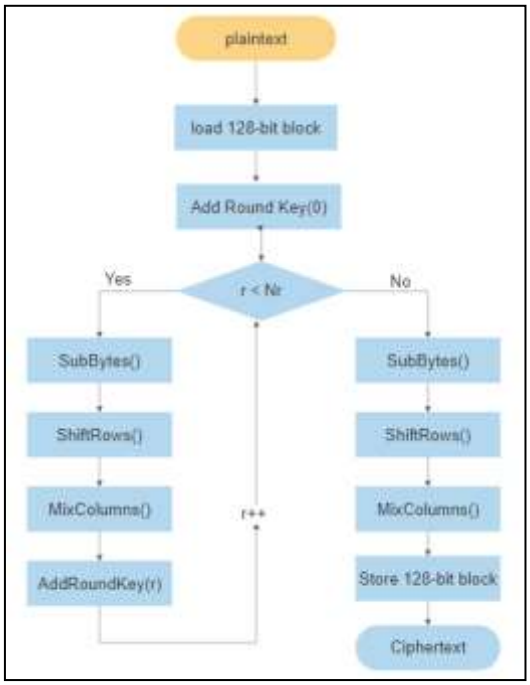


Fig. 4: Workflow of AES Algorithm

3) IDEA

IDEA (International Data Encryption Algorithm) is a cryptographic block cipher encryption algorithm. It is a symmetric block cipher that takes 64 bit as an input, 16-bit key and performs 8 identical rounds of encryption in which 6 different subkeys are used, and four keys are used for output transformation. The block size in IDEA is 16 bytes of 128 bits. A typical series of operations of round blocks are applied to the block cipher where part of the key is applied to the round and then remaining operations are executed. After 8 number of rounds along with a final half round we end up with our ciphertext for that block. The ciphertext/encrypted block is exactly the same size as the original plain text block, which is 16 bytes. A part of the encryption key also known as the round key is used to operate on the block for each round. We obtain the multiple round keys from the round key using a key schedule. Shifts, XORs, Multiplies and other types of operations are performed on the original round key by key schedule algorithm key in order to come up with these round keys [3].

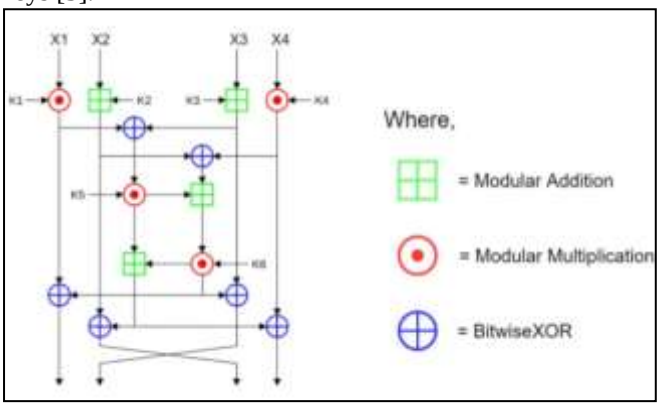


Fig. 5: Single IDEA round

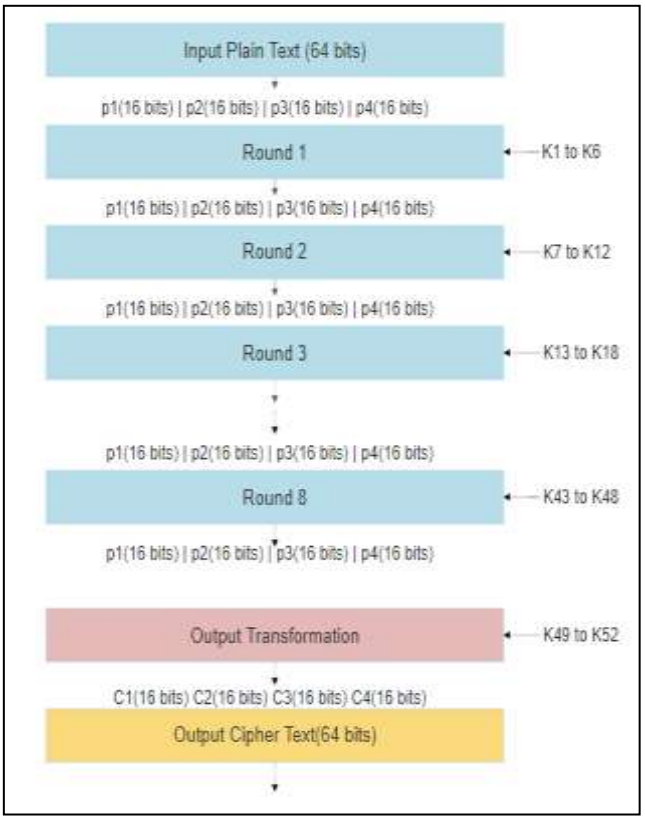
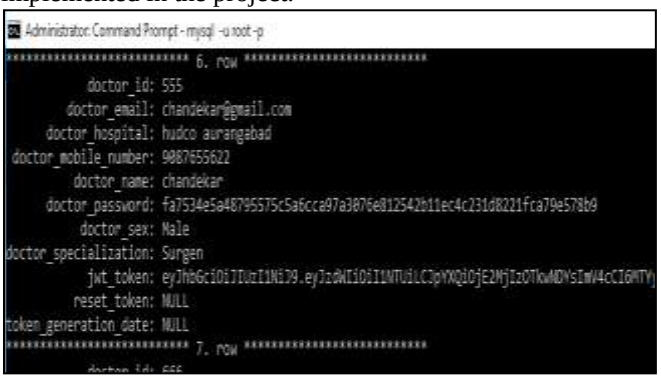


Fig. 6: Workflow of IDEA

IV. IMPLEMENTATION

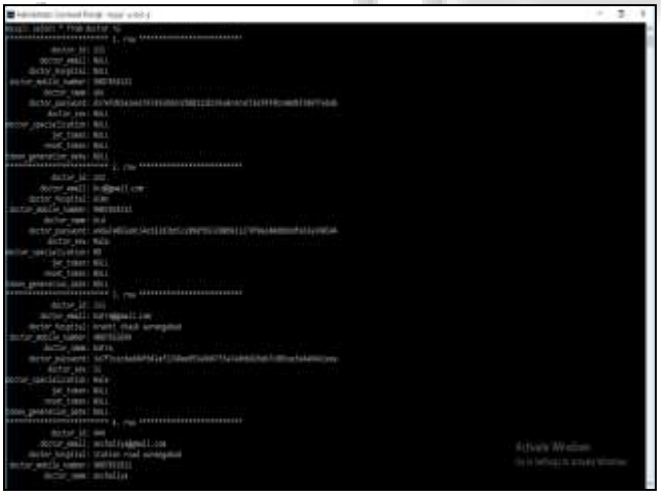
A. Authorization using JWT

A JSON web token (JWT) is JSON Object which is used to securely transfer data over the internet between two hosts. It can be used as an authentication system and can also be used for information exchange. A JWT token is generated when the user logs into the system. It could be stored as cookies on the user system, but doing so won't make the project front-end independent. So the proposed project directly stores it in the database. Once the user logs into the system, a JWT token will be generated and sent to the front end via which the user will be able to access all the REST-APIs. It will be valid for a time of only 5 minutes. This way Authorization will be implemented in the project.



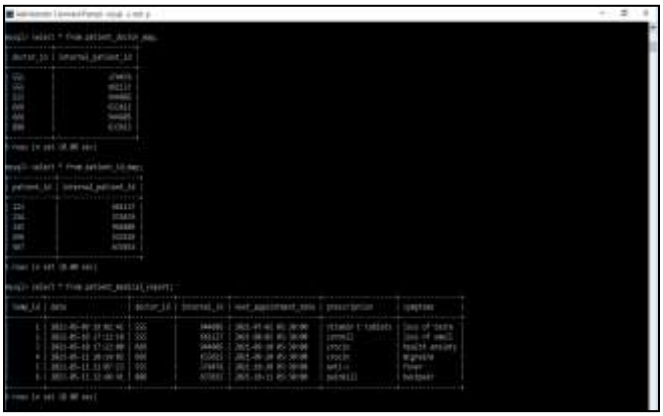
B. Password hashing through SHA-256

Doctor and patient passwords are not needed in their original form in the future. Hashing is known as one way cryptography. So instead of applying encryption/decryption functions on them we will be using hashing algorithm SHA-256. In that way when the password is stored in the table it will be stored in the hashed format. When the user (Doctor/Patient) tries to log in to the system he will be entering the credentials. The password entered by him will again be hashed and then compared with the one stored in the database at the time of user registration. If matched, only then will the user get the authentication. Moreover, intruder won't be able to access the original password even if he gets hold of this hashed password as the original data can't be derived once the data is hashed.



C. Patient public and private Keys

Patient has two keys: public and private. The public key is known to everyone. Private Key is the one which is referred to each time when there's need of the patient id. This is made possible by mapping these keys to one another in a separate table namely patient_id_map. As patient's private ID is used everywhere in the database, even if the intruder gets hold of the patient medical report, he won't be able to know exactly whose medical report he has accessed.



D. AES cryptographic algorithm on the patient_id_map table

An attack on this table is the only way for the intruder to know the patient medical report. So securing this table is of utmost priority. AES being the best cryptographic algorithm, is used to secure patient private and public key mapping from any kind of external attack.

```
mysql> select * from patient_id_map;
+-------------------------+-------------------------+
| patient_id              | internal_patient_id     |
+-------------------------+-------------------------+
| s9yb2hzWEppcpvl3k/iCug== | uvqb2hzWuTQcpvl6Y/iGht*& |
| Yiyb77zEWhucpv67k8iCug== | Opqb2hTtuJucpvl6Y/iGht%^ |
| TRyb7^7EW89cpvu7k$iCug*% | Mnqb6TTtuJUipv&iY/iGht%# |
| Ewsb7^&6gW89cpv8Hk$iCu*& | Tres8uTtuOUipvUintiGht*u |
| Ftvb9%&6gW89cpv8Hk^yCu*& | Riesuu9%uOUipvUintiIot*u |
+-------------------------+-------------------------+
5 rows in set (0.00 sec)
```

E. IDEA cryptographic algorithm on the patient_demographic_details table

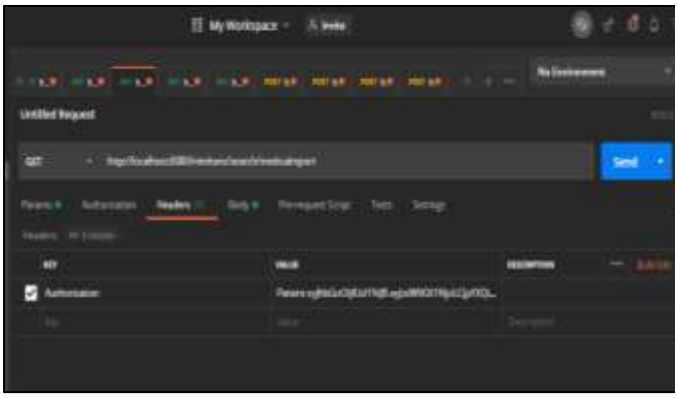
Attack on this table will result to the access of all personal information of the patient including mobile numbers, names, email ids, etc. Access to such details can lead to many types of attacks as we have seen in many scenarios in the past. The latest one being the Domino's data leak which has exposed sensitive demographic details including mobile numbers, addresses, email ids, credit/debit card details of over 18 crore customers. The leaked data is all over the web and is accessible to absolutely anyone on the internet. Thus, it is very important to protect this kind of data. As, similar kind of data will be stored in the patient_demographic_details table, we will be securing it with the IDEA cryptographic algorithm which is one of the most secure cryptographic algorithms available.

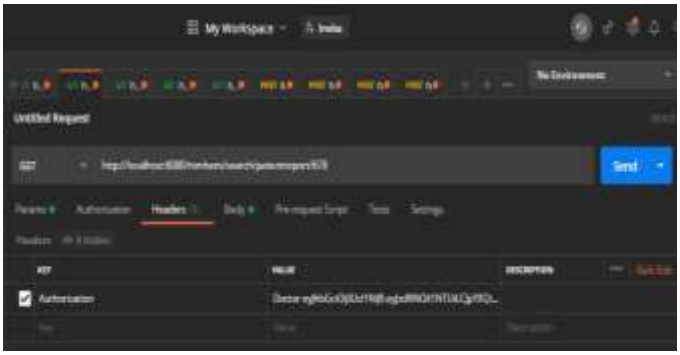
F. CORS for access control

Cross-Origin Resource Sharing (CORS) is a structure based on HTTP-header that allows a server to imply any other points of origins than its own from which a web-browser should allow loading of resources. CORS also relies on a structure by which browsers make a "pre-flight" call to the server hosting the cross-origin resource, in order to check that the server will allow the original request. In that pre-flight, the browser sends headers that indicate the HTTP function and headers that will be used in the original request. The below Doctor Controller shows the implementation of CORS.

G. Header Constraints for accessing REST-APIs

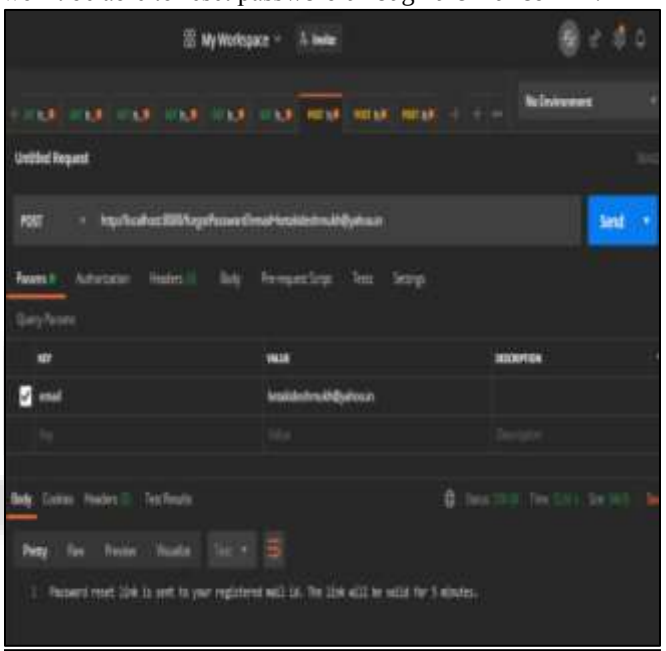
There will be a common GUI for both Patient and Doctor but few APIs will only be meant for Patient and few others will only be meant for Doctor. To maintain this exclusivity, we have specified two types of constraints in the URL header. Headers starting with "Patient" will only be able to access patient APIs and headers starting with "Doctor" will be able to access the doctor APIs.

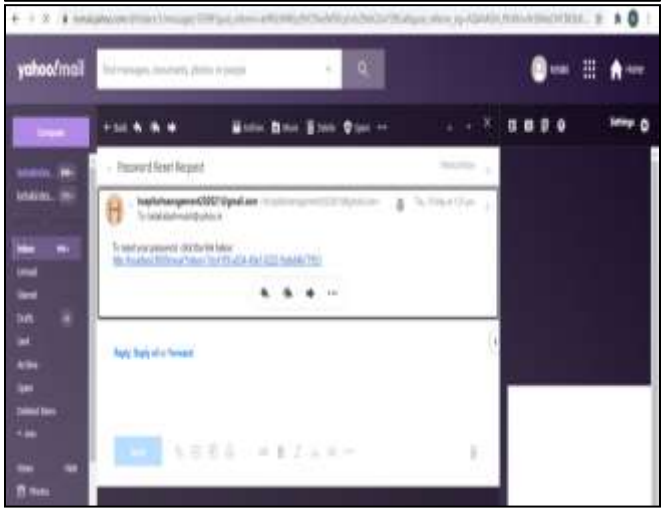


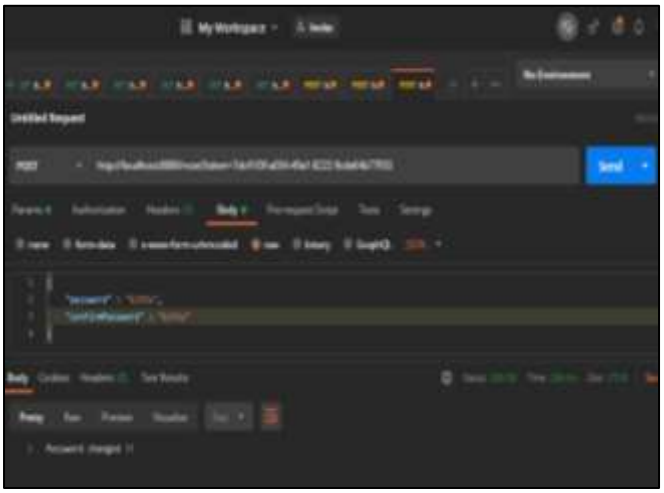


H. Password Reset using Reset token

Reset token and token generation date will help to reset password in case the user forgets password. Time limit for this too is 5mins after which the token will expire and user won't be able to reset password through the mailed link.







V. RESULTS

We successfully implemented the two most secure cryptographic algorithms: IDEA and AES. When it comes to IDEA, multiple computational modular algebra needs to be performed in each round. That is not the case in AES Algorithm. Thus, IDEA requires more resources as compared to AES. Also the number of steps required to generate cipher text in IDEA is a lot when compared to AES, thus exactly those many number of steps are required to de-cipher it back to the original plaintext. This makes IDEA more complex than AES. When it comes to security, both are equally secure. So it's better to go for AES than with IDEA for ensuring data security. Table -1 shows the comparison between the two algorithms AES and IDEA:

	AES	IDEA
Key length	128, 192 or 256 bits	128 bits
Cipher Type	Symmetric Block Cipher	Symmetric Block Cipher
Block Size	128 bits	64 bits
Security	Highly Secure	Highly Secure

Table I: Comparison between AES and Idea Cryptographic Algorithms

VI. CONCLUSION

In this paper, we successfully ensured the privacy and integrity of Healthcare Data. We achieved this by implementing multiple layers of security measures including JWT, CORS, SHA256, AES Algorithm and IDEA. We compared the two cryptographic algorithms: AES and IDEA for their efficiency. The inter-weaving of these measures resulted in a completely secure system which is extremely robust and difficult to break through. Thus, we successfully achieved all the goals this paper proposed.

REFERENCES

[1] B. S. S. P. M. C. Nikhil T. More, "An Insight into the Importance of Requirements Engineering," International Journal of Computer and Communication Technology, vol. 8, no. 1, p. 4, 2017.
[2] J. F. J. J. L. G. W. Z. H. F. a. G. Y. Liandeng Li, "SW-AES: Accelerating AES Algorithm on the Sunway TaihuLight," IEEE, p. 8, 2017.
[3] https://www.educba.com/idea-algorithm/.
[4] Q. L. M. T. A. I. M. AHMED RAZA RAJPUT, "EACMS: Emergency Access Control Management," IEEEAccess, vol. 7, p. 14, 2019.
[5] D. K. S. K. K. Inderpreet Singh, "Improving The Efficiency of E-Healthcare," IEEE, p. 4, 2019.
[6] A. O. Ismail Keshta a, "Security and privacy of electronic health records: Concerns and," Elsevier, p. 7, 2020.
[7] R. D.Bajaj, "AES ALGORITHM FOR ENCRYPTION," International Journal of Latest Research in Engineering and Technology (IJLRET), vol. 02, no. 05, p. 6, 2016.
[8] i.https://www.sciencedirect.com/science/article/abs/pii/S0026269208005661#:~:text=The%20AES%20implementation.
[9] C. S. K. &. B. S. &. H. V. &. A. Nealand, "Security Techniques for the Electronic Health Records," Springer, p. 9, 2017.
[10] https://www.movabletype.co.uk/scripts/sha256.html.
[11] P. P. M. C. Ketaki Deshmukh1, "Data Integrity and Privacy in Healthcare Management System: A Survey," International Research Journal of Engineering and Technology (IRJET), vol. 07, no. 11, p. 4, November 2020.

