

HYBRID ALGORITHM FOR SECURITY OF DATA OVER WIRELESS TRANSFER WITH DISEASE PREDICTION IN MACHINE LEARNING

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

AAKKASH SHREYAN M (17SCS5002)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600019**

MARCH - 2021



SATHYABAMA



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AAKKASH SHREYAN M (17SCS5002)** who carried out the project entitled “**Hybrid Algorithm for security of data over wireless transfer with disease prediction in machine learning**” under my supervision from November 2020 to March 2021.

Internal Guide

MS. D. DEEPA., M.E., (Ph. D)

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **AAKKASH SHREYAN M and ABILASH R** hereby declare that the Project Report entitled **“HYBRID ALGORITHM FOR SECURITY OF DATA OVER WIRELESS TRANSFER WITH DISEASE PREDICTION IN MACHINE LEARNING”** is done by me under the guidance of **Ms. D. Deepa., M.E., (Ph.D)** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of science and technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.L.Lakshmanan M.E., Ph.D., and Dr.S.Vigneshwari M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. D. Deepa., M.E., (Ph.D)., Assistant Professor** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science** and Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Data security is major issue in all the field. With the evolving technological advancements in our day to day life security of personal data is a primary concern for all. Many intruders and hackers are eagerly waiting for personal data of the people which can be used in various means for their works. They always have an eye on the personal information of the people. And the second issue is that diagnosing of diabetes by the doctor will be some time prone to mistakes because it is natural that human makes mistakes but not the machines. So to overcome this issue we develop a hybrid secure algorithm which encrypts the patient personal information which they wish not to share to anyone outside the hospital by crypto AES encryption and hashing using SHA-256 and then storing the data in database which will be accessible only to the patient and the doctor. Together with encryption of the patient personal data and medical data our proposed system will effectively evaluate the data by a trained machine learning model to check whether a person is actually suffering from diabetes or not. We use logistic regression for evaluating the patient medical data to predict the health status so mistakes can be avoided in prediction as the prediction is done by the trained machine. Then doctor can give the treatment depending on the health status. As a result this system will give a sustainable solution to people for safely transmitting their personal data as well as medical data through wireless medium along with diabetes disease prediction.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	iv
	LIST OF TABLE	v
	LIST OF ABBREVIATIONS	vi
1	INTRODUCTION	1
	1.1 OUTLINE	1
	1.2 OVERVIEW OF TECHNOLOGY	2
	1.2.1 MACHINE LEARNING	2
	1.2.2 APPROACHES	2
	1.2.3 DATA ENCRYPTION	4
	1.2.4 ENCRYPTION ALGORITHMS	7
2	LITERATURE SURVEY	9
3	AIM AND SCOPE	14
	3.1 AIM OF THE PROJECT	14
	3.2 OBJECTIVE	14
	3.3 SCOPE	14
	3.4 ADVANTAGES	14
4	SYSTEM DESIGN & METHODOLOGY	15
	4.1 EXISTING SYSTEM	15

	4.1.1 CONS OF EXISTING SYSTEM	15
	4.2 PROPOSED SYSTEM	15
	4.3 DETAILED DESIGN	16
	4.3.1 ARCHITECTURE DIAGRAM	16
	4.4 WORKING	17
	4.5 MODULE	17
	4.6 SOFTWARE DEVELOPMENT	32
	4.6.1 MICROSOFT VISUAL STUDIO	32
	4.6.2 Node JS	33
5	RESULTS AND DISCUSSIONS	38
	5.1 FINAL RESULTS OBTAINED	38
	5.2 DISCUSSIONS	41
6	CONCLUSIONS AND FUTURE WORK	42
	6.1 CONCLUSION	42
	6.2 FUTURE WORK	42
	REFERENCES	43
	APPENDIX	45
	A.SAMPLE CODE	45
	B. PLAGIARISM REPORT	50

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1.1	SUPERVISED LEARNING	3
1.2	UNSUPERVISED LEARNING	3
1.3	REINFORCEMENT LEARNING	4
1.4	DATA ENCRYPTION PROCESS	5
1.5	JWT PROCESS	7
1.6	CRYPTO AES ENCRYPTION	8
4.1	ARCHITECTURE OF THE SYSTEM	16
4.2	TRAINING THE MODEL	17
4.3	A TYPICAL JWT STRUCTURE	19
4.4	AES DESIGN	21
4.5	A TYPICAL DATABASE INTEGRATION	24
5.1	LOGIN PAGE	38
5.2	REGISTRATION FORM	39
5.3	DETAILS ENCRYPTED AND SAVED	39
5.4	DOCTOR WEBPAGE	40
5.5	DATA QUERY TAKES 100 TO 200MS	40

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
5.1	COMPARISION OF SYSTEM	41

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
AES	ADVANCED ENCRYPTION STANDARD
API	APPLICATION PROGRAMMING INTERFACE
JWT	JSON WEB TOKEN

CHAPTER 1

INTRODUCTION

1.1 OUTLINE

Data security refers to the process of protecting data from unauthorized access and data corruption throughout its lifecycle. Data security includes data encryption, hashing, tokenization, and key management practices that protect data across all applications and platforms.

Organizations around the globe are investing heavily in information technology (IT) cyber security capabilities to protect their critical assets. Whether an enterprise needs to protect a brand, intellectual capital, and customer information or provide controls for critical infrastructure, the means for incident detection and response to protecting organizational interests have three common elements: people, processes, and technology.

Diabetes is fast gaining the status of a potential epidemic in India with more than 62 million diabetic individuals currently diagnosed with the disease. In 2000, India (31.7 million) topped the world with the highest number of people with diabetes mellitus followed by China (20.8 million) with the United States (17.7 million) in second and third place respectively. According to Wild et al. the prevalence of diabetes is predicted to double globally from 171 million in 2000 to 366 million in 2030 with a maximum increase in India. It is predicted that by 2030 diabetes mellitus may afflict up to 79.4 million individuals in India, while China (42.3 million) and the United States (30.3 million) will also see significant increases in those affected by the disease. India currently faces an uncertain future in relation to the potential burden that diabetes may impose upon the country. Many influences affect the prevalence of disease throughout a country, and identification of those factors is necessary to facilitate change when facing health challenges. So what are the factors currently affecting diabetes in India that are making this problem so extreme?

The diabetes in India is multifactorial and includes genetic factors coupled with environmental influences such as obesity associated with rising living standards,

steady urban migration, and lifestyle changes. Yet despite the incidence of diabetes within India, there are no nationwide and few multi-centric studies conducted on the prevalence of diabetes and its complications. The studies that have been undertaken are also prone to potential error as the heterogeneity of the Indian population with respect to culture, ethnicity, socio- economic conditions, mean that the extrapolation of regional results may give inaccurate estimates for the whole country.

1.2 OVERVIEW OF TECHNOLOGY

1.2.1 MACHINE LEARNING

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: *The ability to learn*. Machine learning is actively being used today, perhaps in many more places than one would expect.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is need

1.2.2 APPROACHES

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically, supervised learning is a learning in which we teach or train the machine using data which is well labelled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labelled data. Supervised learning is where there are input variables (x) and an output variable (Y) and an algorithm is used to learn the mapping function from the input to the output. The goal is to approximate the mapping function so well that when there is a new input data (x) that the output variables (Y) for that data can be predicted easily.

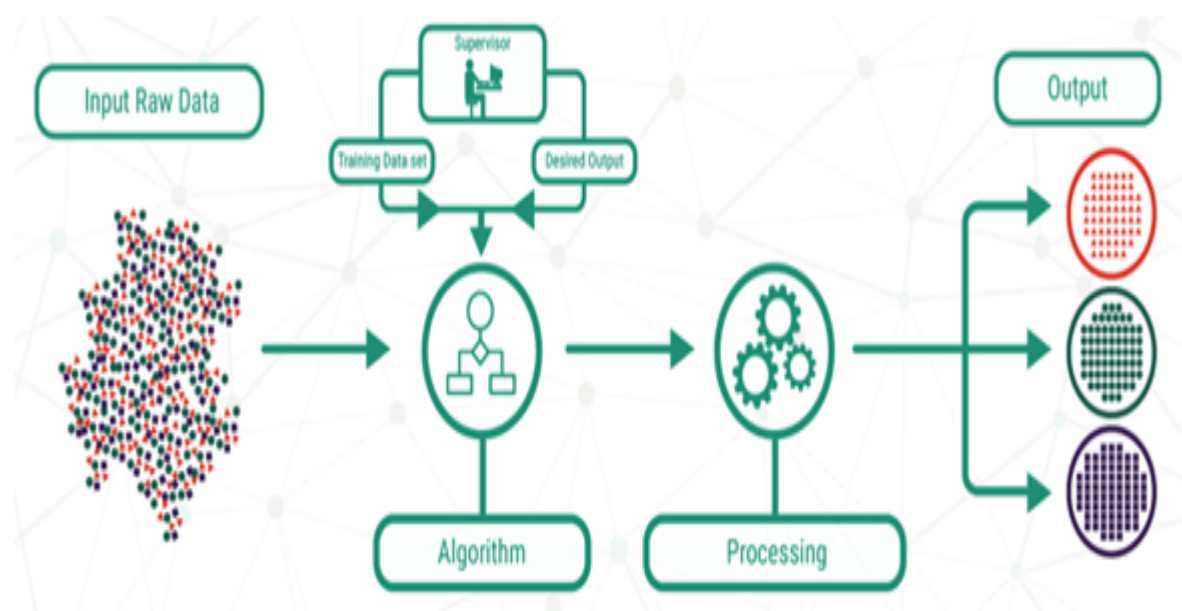


Fig 1.1: Supervised Learning

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data. Unlike supervised learning, no teacher is provided that means no training will be given to

the machine. Therefore, machine is restricted to find the hidden structure in unlabeled data by itself.

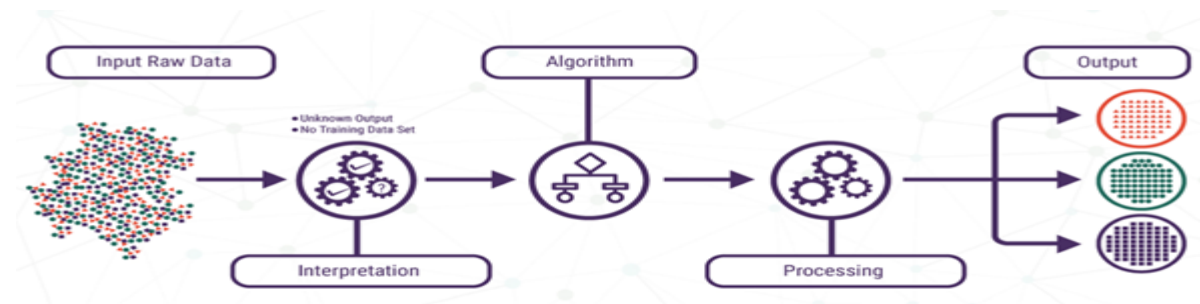


Fig 1.2: Unsupervised Learning

A computer program interacts with a dynamic environment in which it must perform a certain goal. As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.



Fig 1.3: Reinforcement Learning

1.2.3 DATA ENCRYPTION

Data encryption translates data into another form, or code, so that only people with access to a secret key (formally called a decryption key) or password can read it. Encrypted data is commonly referred to as cipher text, while unencrypted data is called plaintext. Currently, encryption is one of the most popular and effective data security methods used by organizations. Two main types of data encryption exist - asymmetric encryption, also known as public-key encryption, and symmetric

encryption. A complete encryption solution provides businesses and their owners with peace of mind because it protects data in all states – at rest and in transit. While perimeter security solutions can make it more difficult to access the data from the outside, once the network is breached your data is only as safe as the data encryption you employ. With the right encryption solution, you can go through each day knowing that your data is safe and that there is no reasonable way in which hackers could potentially get their hands on the raw data. It would take a brute-force style program more than a lifetime to successfully decode all of the information. Although there are other means in which data could potentially be accessed, taking the simple step toward encryption helps to make the job too difficult for hackers that might ordinarily be interested in targeting your business. One of the most vulnerable aspects of data emerges during the transport process. While SSL/TLS is the industry standard for data in motion it has many disadvantages for your data security. An effective encryption solution helps to ensure that data is protected at all times, at rest and in motion. Files that are shared or uploaded to cloud systems should be to ensure that the files remain safe throughout the transport process. One of the worries that many organizations of all sizes share is whether or not the encryption process will affect the integrity of their data. Although data theft is a very common problem, another way for hackers to commit data fraud is to knowingly alter the data that is available. Encryption keeps your data safe from alterations, and recipients of the data will be able to see if it has been tampered with. Alteration of data is something that many businesses often overlook when they are looking into ways to Complete Data Protection

A complete encryption solution provides businesses and their owners with peace of mind because it protects data in all states – at rest and in transit. While perimeter security solutions can make it more difficult to access the data from the outside, once the network is breached your data is only as safe as the data encryption you employ. With the right encryption solution, you can go through each day knowing that your data is safe and that there is no reasonable way in which hackers could potentially get their hands on the raw data. It would take a brute-force style program more than a lifetime to successfully decode all of the information.

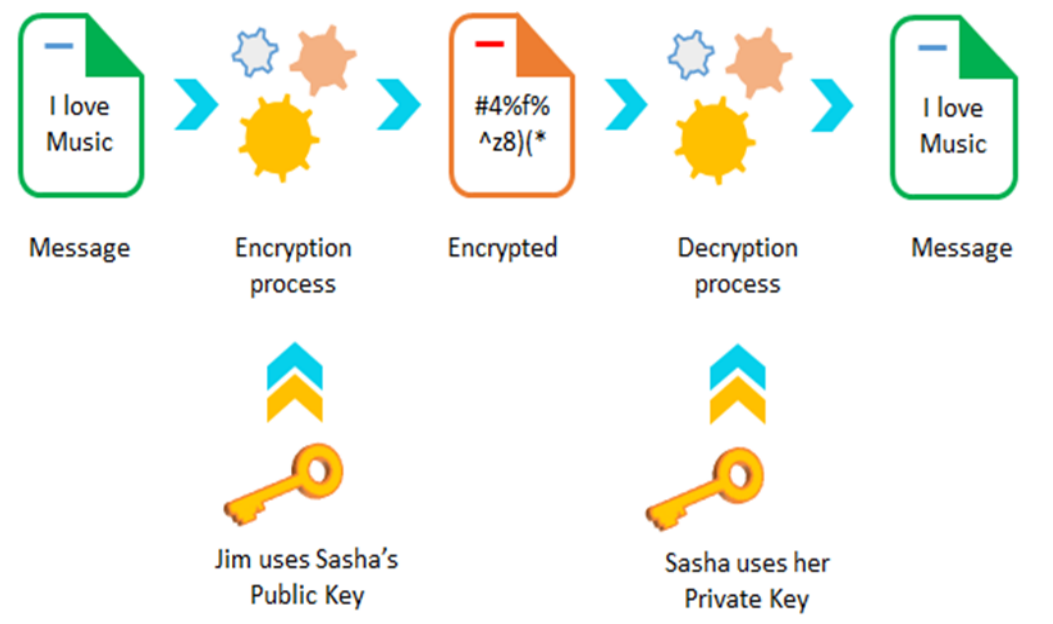


Fig 1.4: Data Encryption Process

Although there are other means in which data could potentially be accessed, taking the simple step toward encryption helps to make the job too difficult for hackers that might ordinarily be interested in targeting your business.

With smartphones and other mobile devices gaining popularity in recent years, many companies have struggled to find a solution for keeping the data stored and passed through these devices safe from potential theft. Luckily, data encryption software will allow you to ensure that all data, across any device, is completely encrypted with the same protections in place that you would find in data stored on a desktop computer. Data encryption will help to take an untenable, stressful situation and make it manageable, while providing peace of mind. On top of that, device authentication can eliminate risk of infiltration from unwanted users.

One of the most vulnerable aspects of data emerges during the transport process. While SSL/TLS is the industry standard for data in motion it has many disadvantages for your data security. An effective encryption solution helps to ensure that data is protected at all times, at rest and in motion. Files that are shared or uploaded to cloud systems should be to ensure that the files remain safe throughout the transport process.

One of the worries that many organizations of all sizes share is whether or not the encryption process will affect the integrity of their data. Although data theft is a very common problem, another way for hackers to commit data fraud is to knowingly alter the data that is available. Encryption keeps your data safe from alterations, and recipients of the data will be able to see if it has been tampered with. Alteration of data is something that many businesses often overlook when they are looking into ways to keep their data safe.

Compliance is extremely important, and many IT departments must comply with legal, insurance and industry restrictions on how data can be handled and transmitted. Encryption provides one of the safest ways for a business to transmit and store data and comply with the restrictions that your business currently has in place such as FIPS, FISMA, HIPAA, PCI/DSS or Gramm-Leach Bliley.

Basic data protection is absolutely essential for your business. For those that want to take the next step or are recovering from their own data theft incident, encryption is one of the best methods to ensure that your data is still safe. It allows you to achieve complete security across all devices, send and receive data safely, and guarantee absolute integrity, whenever a file is accessed or shared. If you are interested in protecting your business from the impact of data breaches, encryption software is an excellent investment. keep their data safe.

1.2.4 ENCRYPTION ALGORITHMS

JWT is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed.

JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key.

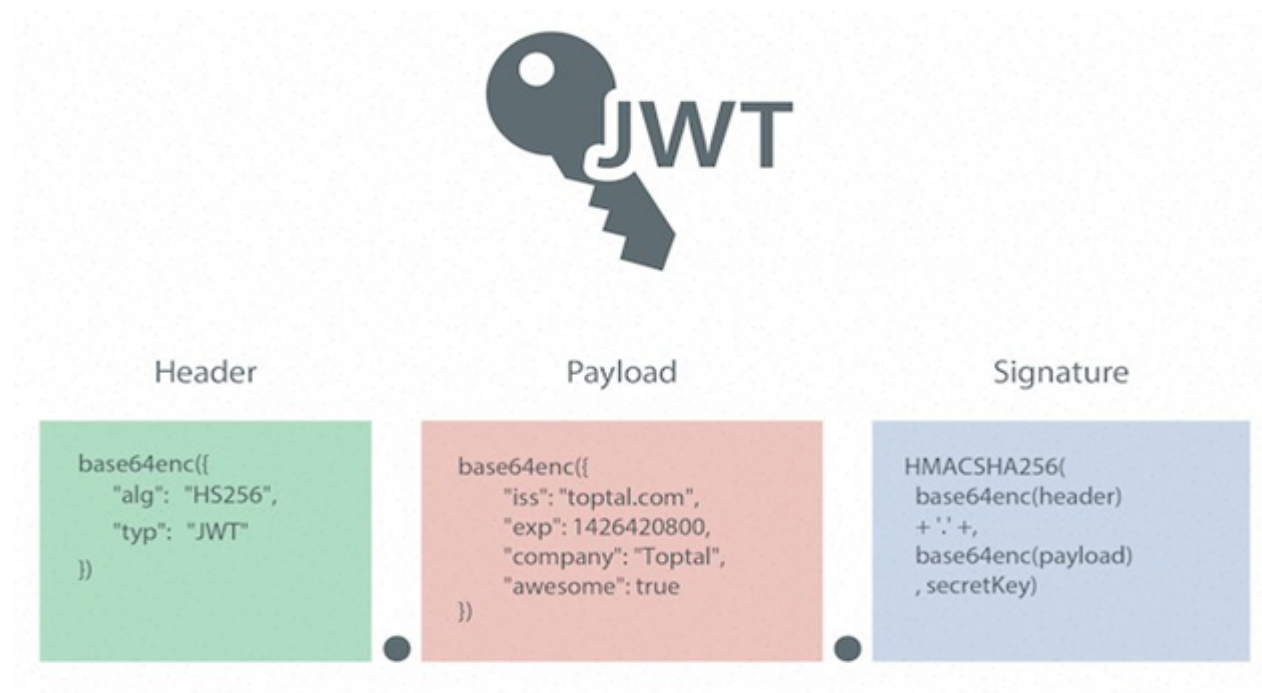


Fig 1.5: JWT Process

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six times faster than triple DES. A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

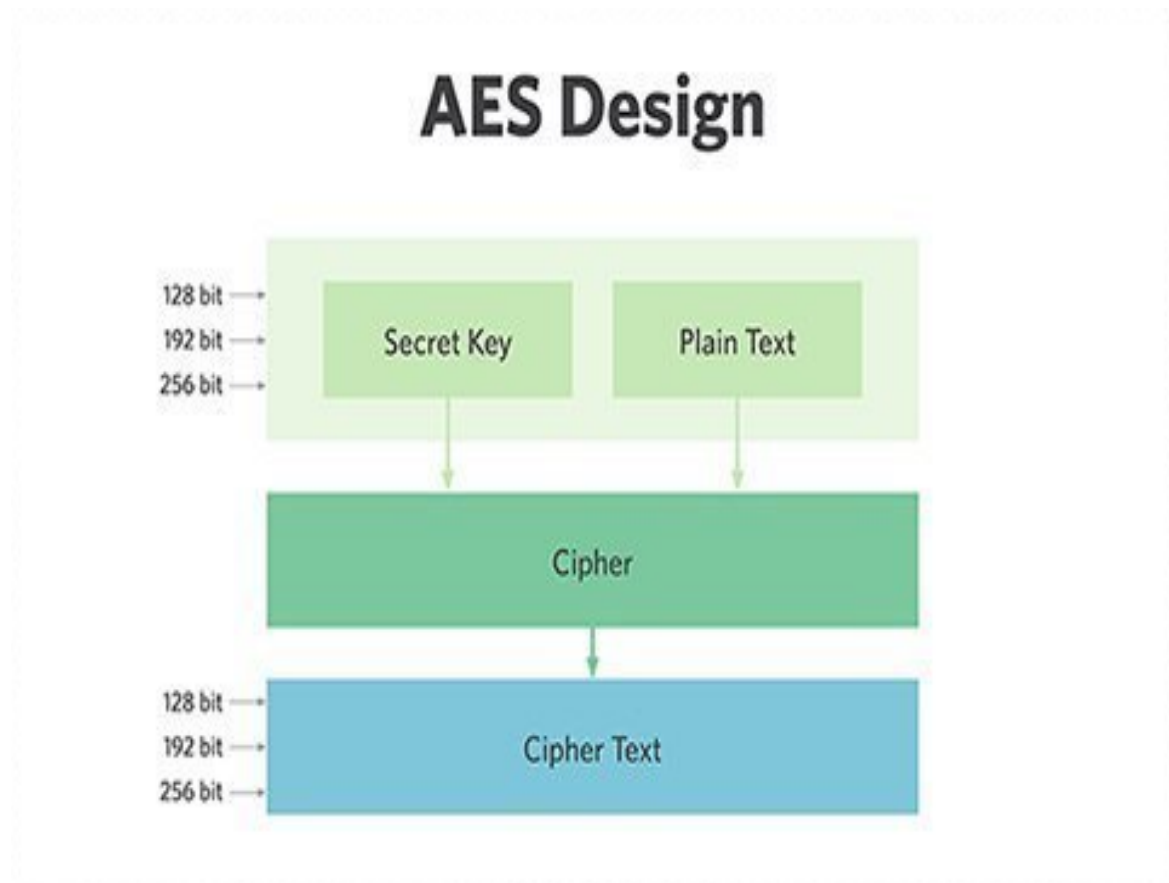


Fig 1.6: Crypto AES Encryption

A cryptographic hash (sometimes called 'digest') is a kind of 'signature' for a text or a data file. SHA-256 generates an almost-unique 256-bit (32-byte) signature for a text. A hash is not 'encryption' – it cannot be decrypted back to the original text (it is a 'one-way' cryptographic function, and is a fixed size for any size of source text). This makes it suitable when it is appropriate to compare 'hashed' versions of texts, as opposed to decrypting the text to obtain the original version. SHA-256 is one of the successor hash functions to SHA-1 (collectively referred to as SHA-2), and is one of the strongest hash functions available. SHA-256 is not much more complex to code than SHA-1, and has not yet been compromised in any way. The 256-bit key makes it a good partner-function for AES. It is defined in the NIST (National Institute of Standards and Technology) standard ". NIST also provide a number of test vectors to verify correctness of implementation.

CHAPTER 2

LITERATURE SURVEY

Jakov Andric, et al [1] This gives a encryption against intruders attacks in which a linear encryption is given to reduce the attack. Here the attacks against data is prevented which lesser the mistakes. He also explained the comparing of various attacks against data which can be reduced.

Massoud Masoumi, [2] This gives the security in power analysis data. It shows the memristor implementation of secure algorithm which can prevent the danger of differential power analysis attacks which comes at the affordable cost in which 128 bit Advanced Encryption Standard cryptoprocessor was done and done in CMOS and hybrid CMOS. It is confirmed that hybrid CMOS implementation gives very good improvement compared with implementation in regular CMOS architectures which has less energy absorbing and attack tolerance and in removing DPA attacks with low cost.

Nadia Mustafa Mohammed Alhag, et al [3] This gives a implementation of increasing the DES algorithm key length to 1024 bits compared to the earlier very short key length which is just 64 bits which can be cracked easily. In this proposed system 1024 bits is separated into many keys and is given separately for various algorithms which each key is given independently for various algorithm routines. The overall output is very good for detecting key here addition of key length makes it hard to find in a big space of number.

T V Sathyanarayana, et al [4] This gives securing of data in cloud is a very big problem. Here secure software life cycle is used to the protection of cloud services. So in this paper security of data in cloud is achieved in confidentiality ,Integrity and applied to dynamic architectures. The analysis of the given keys can be used to find lacunae in the security of data issues in the cloud.

Lei Xu, et al [5] This gives a method for authorization of medical records. In general medical records are to be stored with high confidentiality and authenticity in medical databases. Just encrypting those databases will reduce data utility because encrypted databases are not possible to search now and many people will be involved in sharing the medical data of the clients so authorizing different people to be done in a secure manner. So this paper proposes an authorizable searchable encryption process in which RSA enable a authority to reduce the search option to different people based on different people history. We use multi authority attribute encryption to select the authorization to be done only one time even from multiple authorities.

Ailar Rahimli and Maslin Masrom [6] This paper addresses the data security issue which are already in hospital information systems. In general hospital information systems are integrated systems which is to cover all the administrative and financial tasks where data security is a major issue. This paper provides better data security in hospital information systems which ensures that the patients data is secured HIS keeps using different level of security for keeping patients personal information safe. In this paper protection covers data acquisition, data storage and analytical reporting activities. It covers all the technical issue.

Yue Li, et al [7] This is a system to investigate how much users personal data is in the password. In general users add there personal information as password for better remembering. So here we study about a new metric called coverage to identify how user information is related with the passwords. Then we extend the probabilistic context free grammar to semantic to crack password by PCFG. In comparison with the personal PCFG which identifies password much easily than PCFG which makes attacks more vulnerable. To stand against such attacks we use that are given by users which efficiently reduce to risk of personal data attacks.

Mauro Mangia, et al [8] This is a system in which chaining that is each and every message is secured by considering a digital summary of its before ones to block cipher stages. So this technique will harden the upcoming sytem in contrast with very commonly occurring threats like cipher text only. Along with it robustness comes

with the price of less than 2 percent energy in contrast with the pure compression stage that gives a good amount of reduction considering the straight implementation of a cryptography. Here an energy increases of just 0.58 percent of the total for the compression or encryption block.

Elisa Bertino [9] This is a data security system for securing the data in the databases in which users access the data for their private uses which is the main target of the hackers to access data from outside already many cryptographic system has the done as a solution to this problem, Encryption is the main thing which provides security to data. But the major drawback of existing methods is that it can cause damage to the already existing schema as both the output length as well as the input length is different which changes data format.

Hou Tung Yew, et al [10] This paper gives the mobile-health (M-Health) system, also known as the wireless body area network (WBAN) for remote patient monitoring by internet of things , is a system which can remotely monitor a real-time human body's health status parameters. Generalized sign encryption can realize encryption, signature and sign encryption with only one keypair and one algorithm. To address the communication security requirement for M-Health system, Zhang et al. proposed a light-weight secure data transmission protocol for M-Health system by using a certificate less generalized sign encryption (CLGSC) scheme recently. In this paper, we point out that an insider attack exists in Zhang et al.'s CLGSC scheme, which destroys the confidentiality of the scheme. Thus, their light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health system is secure.

Asrul Nizam Asat, et al [11] Dengue is a baleful disease. This disease is found in several developed as well as developing countries like India. In India at very first dengue was detected in 1963 at Kolkata. This disease is caused by biting of mosquito. However, this can be prevented by taking proper measures. The datasets contain symptoms which are BP, heart rate, fever temperature, and platelets. It possesses symptoms which affect patient and verify. In this paper we consultation about Naive Bayes algorithm for classification that describes utilization for dengue

disease prediction. In this method, the real-time data is accepted and is stored on the server. Analysis of previous records of the patient. And that accepted real-time data compared with the dataset and it gives the probability of the dengue.

Xu Ruzhi, et al [12] This is a data security system for securing the data in the databases in which users access the data for their private uses which is the main target of the But the major drawback of existing methods is that it can cause damage to the already existing schema as both the output length as well as the input length is different which changes data Format preserving system is useful when it is necessary to integrate encryption into existing applications.

Neha Agarwal, et al [13] Defending against malicious attacks has become increasingly important in various cyber-physical systems. This paper presents an encryption-based countermeasure against stealthy attacks on remote state estimation. Smart sensors transmit data to a remote estimator through a wireless communication network, in which data packets can be intercepted and compromised by attackers. The remote end is equipped with a false data detector that monitors the system. To avoid being detected, the attack should follow the stealthiness constraint. A linear encryption scheme is proposed to reduce the influence of potential stealthy attacks. For arbitrary linear encryption, the worst-case linear attack that yields the largest estimation error is derived. Accordingly, the optimal linear encryption, which minimizes the worst-case estimation error, is designed based on the Stackelberg game analysis. The above optimal strategies are considered in both the complete and partial measurement information scenarios for the attacker. Moreover, the generalization to nonlinear encryption strategies is also discussed. Comparisons of attack and encryption strategies through numerical examples are provided to illustrate the theoretical results.

Hanan Rady, et al [14] Memory resistor or memristor is the forth fundamental circuit element that has attained considerable attention due to its unique characteristics and possible extensive applications in future generation nanoscale circuits and systems. In this brief, the contribution that memristor based circuits may offer to the evolution of cryptographic hardware and embedded systems is discussed. Specifically, it will

be shown how memristor-based implementation of security algorithms can mitigate the danger of differential power analysis attacks (DPA) at the technology level with lower cost and energy compared to conventional existing algorithmic countermeasure techniques. A 128-bit Advanced Encryption Standard (AES) crypto-processor was designed and implemented in both CMOS and hybrid CMOS/memristor technology. The robustness of the CMOS-based implementation against power analysis attacks was evaluated on Side-Channel Attack User Reference Architecture (SAKURA-GII) while the nanoscale counterpart system was evaluated by using a customized simulation and attack environment which was developed for extracting power traces using Synopsys and Cadence tools along with a DPA attack software implemented in MATLAB. It was observed that hybrid CMOS/memristor-based implementation provides considerable improvement over implementation with regular CMOS architectures in terms of energy consumption and attack tolerance and demonstrates good potential in mitigating DPA attacks without having to apply costly countermeasures such as masking or hiding.

Sabita Khadka [15] The health care services industry is always showing signs of change and supporting new advancements and advances. One of the predominant requirements in today's health care systems is to protect the patient's medical report against potential attackers. Hence, it is basic to have secure information that can just approve people can get to the patient's medical report. So, We have proposed Block chain technology as a disbursed approach to grant security in accessing the medical report of a patient. It's composed of three phases . Authentication, Encryption and Data Retrieval using Block Chain technology. For authentication – Quantum Cryptography, for Encryption – AES and for Data Retrieval SHA algorithms are used to resist the frequent attacks. This proposed framework may likewise ensure the protection of the patients and moreover keeps up the security and trustworthiness of the health care system.

Long Chen, et al [16] Recently, a practical public integrity auditing scheme supporting multi-user data modification was proposed. Although the protocol was claimed secure, in this paper, we show that the proposal fails to achieve soundness, the most essential property that an auditing scheme should provide. Specifically, we show that

a cloud server can collude with a revoked user to deceive a third-party auditor (TPA) that a stored file keeps virgin even when the entire file has been deleted.

Abdul Razaque, et al [17] Data sharing has never been easier with the advances of cloud computing, and an accurate analysis on the shared data provides an array of benefits to both the society and individuals. It allows a data owner to anonymously authenticate his data which can be put into the cloud for storage or analysis purpose. Yet the costly certificate verification in the traditional public key infrastructure (PKI) setting becomes a bottleneck for this solution to be scalable. In this paper, we further enhance the security of ID-based ring signature by providing forward security: If a secret key of any user has been compromised, all previous generated signatures that include this user still remain valid. This property is especially important to any large-scale data sharing system, as it is impossible to ask all data owners to reauthenticate their data even if a secret key of one single user has been compromised.

CHAPTER-3

AIM AND SCOPE

3.1 AIM OF THE PROJECT

The aim of this proposed system is to efficiently develop a hybrid algorithm with machine learning which is trained to classify whether a person is actually suffering from diabetes or not based upon the glucose level which is given as input. Along with it, it also transfers patient data and reports securely through wireless medium without any access by the intruders by providing a multilevel encryption. So both the privacy of patient data and accuracy of reports can be assured.

3.1.1 OBJECTIVE

The main objective of our project is to develop an application to perform diagnosis for diabetes. The objective of the project is to ensure secure transmission of data over a wireless communication and to design an efficient system where data

is secured by merging two security algorithms for data stored and accessed in cloud and to design a web application for entering and viewing data for encryption.

3.2 SCOPE

This model will help in identifying that whether a person is actually suffering from diabetes or not and it also securely transfers the patient personal data along with the health report securely through wireless medium by using hybrid encryption.

3.2.1 ADVANTAGES

- A medical application is developed to perform diagnosis and to predict if the patient is diabetic or normal using machine learning algorithm.
- An end to end data security is provided using hybrid secure algorithm.
- Ensures security over the wireless data transfer server to the database
- Bank and hospital details can be secured with utmost security

CHAPTER 4

SYSTEM DESIGN & METHODOLOGY

4.1 EXISTING SYSTEM

Defending against malicious attacks has become increasingly important in various cyber-physical systems. This paper presents an encryption-based countermeasure against stealthy attacks on remote state estimation. Smart sensors transmit data to a remote estimator through a wireless communication network, in which data packets can be intercepted and compromised by attackers. The remote end is equipped with a false data detector that monitors the system. To avoid being detected, the attack should follow the stealth constraint. A linear encryption scheme is proposed to reduce the influence of potential stealthy attacks. For arbitrary linear encryption, the worst-case linear attack that yields the largest estimation error is

derived. Accordingly, the optimal linear encryption, which minimizes the worst-case estimation error, is designed based on the Stackelberg game analysis. Moreover, the generalization to nonlinear encryption strategies is also discussed. Comparisons of attack and encryption strategies through numerical examples are provided to illustrate the theoretical results.

4.1.1 CONS OF EXISTING SYSTEM

- A single encryption algorithm is insufficient to transmit data securely.
- The system can be easily hacked.
- Cannot secure highly confidential data.
- Does not perform real time disease prediction

4.2 PROPOSED SYSTEM

Data security includes data encryption, hashing, tokenization, and key management practices that protect data across all applications and platforms. The security system used nowadays uses Data encryption software to effectively enhance data security by using an algorithm (called a cipher) and an encryption key to turn normal text into encrypted cipher text. To an unauthorized person, the cipher data will be unreadable. That data can then be decrypted only by a user with an authorized key. Whereas with the improving data insecurity nowadays leads to loss of confidential data as the key is easily hackable because of a single algorithm usage. Diabetes-related complications include damage to large and small blood vessels, which can lead to heart attack and stroke, and problems with the kidneys, eyes, feet and nerves. The risk of most diabetes-related complications can be reduced if diagnosed early. To overcome this problem this project presents a medical application that accepts and analyses a patient's medical data to give a diagnosis to check if he/she is diabetic with an efficient data security system where two security algorithms will be merged to secure the patient's medical data stored and accessed in cloud. In this project logistic regression algorithm, a regression machine learning algorithm is used to process the entered patient's medical data and to predict if the patient is diabetic or normal. A web application using reactJS is developed from which the patient's medical data which will be encrypted using the hybrid secure

algorithms before being uploaded to the database. In addition, the emerging block chain technology with the wireless data transfer system, makes easy the interaction between the data and cloud. The medical data is analyze and the result is stored securely in a cloud database such as mongo DB as an highly secure encrypted key. The result is received from the database and decrypted in the frontend to view the test results. Thus, this project presents an effective end to end security for medical applications.

4.3 DETAILED DESIGN

4.3.1 ARCHITECTURE DIAGRAM

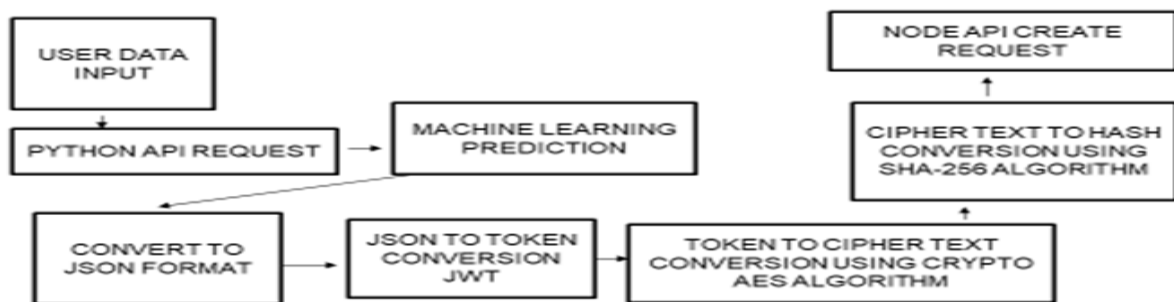


Fig 4.1: Architecture of the System

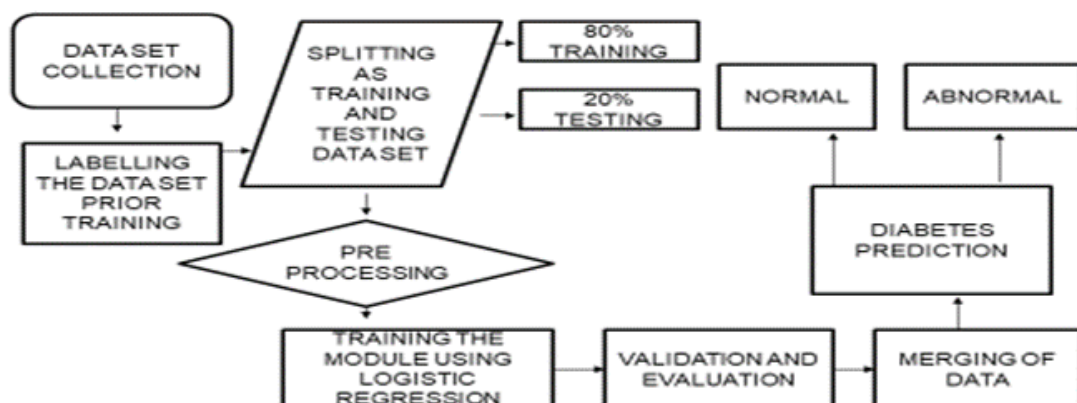


Fig 4.2: Training the Model

4.4 WORKING

The working of the total model from the above diagrams can be explained as follows. The medical data to be secured is given through a web application which is converted to the JSON format, that is to be secured and encrypted is first converted to a token using a HS-256 algorithm, with an inbuilt secret key. This token is then encrypted to a cipher text using a crypto AES algorithm, with an inbuilt secret key. These two secret keys from the JWT and crypto AES algorithm are converted to a hash key using a block chain algorithm SHA-256 algorithm. This hash key is then embedded with the cipher text which is nothing but the end output of the encryption process. This medical data is sent from a reactJS web application to a mongoDB database through API which is developed using nodeJS. A machine learning algorithm logistic regression is used to predict if the patient is diabetic or normal based on the input medical data. The data secured through block chain can never be decrypted it can only be encrypted using the hash key of another member of a network. Thus, the project has successfully ensured secure data transfer over wireless communications.

4.5 MODULE

DISEASE PREDICTION USING MACHINE LEARNING

In this project logistic regression a machine learning algorithm is used to predict if the patient is normal or diabetic based on the input medical data. Diabetes mellitus, commonly known as diabetes, is a metabolic disease that causes high blood sugar. The hormone insulin moves sugar from the blood into your cells to be stored or used for energy. Logistic Regression statistical method is used for analyzing the dataset and produces a binary outcome. One or more autonomous variables may have consisted of the dataset. The result is determined by these variables that are dichotomous in nature. Which means only two results are possible. It is a specific category of regression and it is used in the best way to predict the binary and categorical output. Logistical Regression method is used to regulate the impact of numerous autonomous variables which are conferred at the same time. This method also predicts any one of the two independent categories of variables. Logistic

regression designs the best-fitting function with the help of the maximum likelihood method in order to maximize the probability of classifying the recognized data into the proper division. Various appliances of logistic regression are forecast market trends, to find the success and failure rates in results, the true or false category in recruiting employees based on their performance in need of employment in a company, image categorization, health care and analyze a group of people affected by Myocardial Infarction.

JWT GENERATION

JSON Web Token is an Internet standard for creating data with optional signature and/or optional encryption whose payload holds JSON that asserts some number of claims. The tokens are signed either using a private secret or a public/private key. The tokens can be signed by one party's private key (usually the server's) so that party can subsequently verify the token is legitimate. If the other party, by some suitable and trustworthy means, is in possession of the corresponding public key, they too are able to verify the token's legitimacy. The tokens are designed to be compact, URL-safe, and usable especially in a web-browser single-sign-on (SSO) context. JWT claims can typically be used to pass identity of authenticated users between an identity provider and a service provider, or any other type of claims as required by business processes. JWT relies on other JSON-based standards: JSON Web Signature and JSON Web Encryption.

JWT structure consists of three phases:

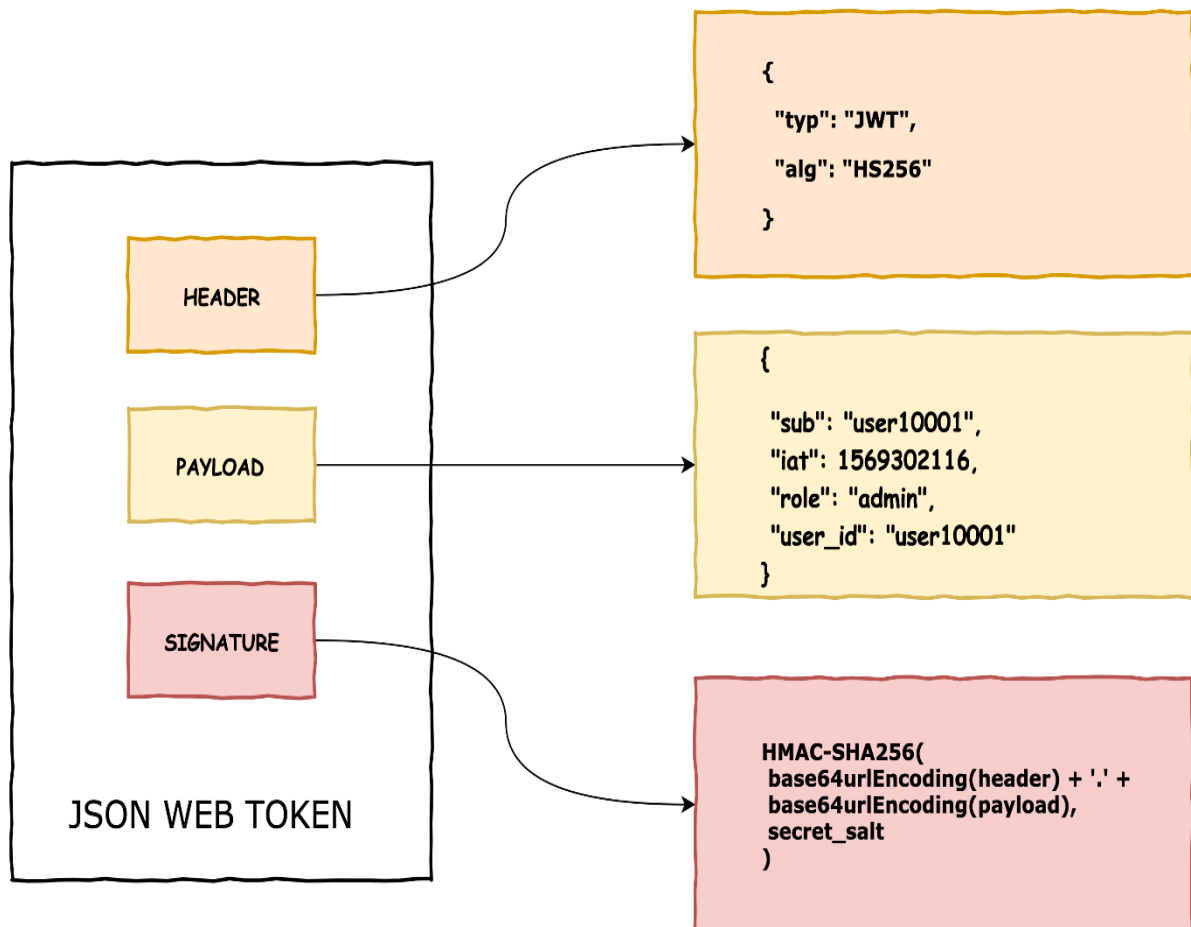


Fig 4.3: A Typical JWT Structure

HEADER:

The header typically consists of two parts: the type of token, which is JWT, and the hashing algorithm that is used, such as HMAC SHA256 or RSA.

Then, this JSON is Base64Url-encoded to form the first part of the JWT.

PAYLOAD:

The second part of the token is the payload, which contains the claims. Claims are statements about an entity

Registered claims: These are a set of predefined claims which are not mandatory but recommended, to provide a set of useful, interoperable claims.

Public claims: These can be defined at will by those using JWTs. But to avoid collisions they should be defined

Private claims: These are the custom claims created to share information between parties that agree on using them and are neither registered or public claims.

SIGNATURE:

To create the signature part, you have to take the encoded header, the encoded payload, a secret, the algorithm specified in the header, and sign that.

Then, you have to put it all together. The following shows a JWT that has the previous header and payload encoded, and it is signed with a secret.

JWT TO CIPHER GENERATION

The Advanced Encryption Standard (AES) is a symmetric block cipher chosen by the U.S. government to protect classified information. AES is implemented in software and hardware throughout the world to encrypt sensitive data. It is essential for government computer security, cybersecurity and electronic data protection. AES includes three block ciphers: AES-128, AES-192 and AES-256. AES-128 uses a 128-bit key length to encrypt and decrypt a block of messages, while AES-192 uses a 192-bit key length and AES-256 a 256-bit key length to encrypt and decrypt messages. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128, 192 and 256 bits, respectively. AES is used widely for protecting data at rest. Applications for AES include self-encrypting disk drives, database encryption and storage encryption. On the other hand, the RSA algorithm is often used in web browsers to connect to websites, in virtual private network (VPN) connections and in many other applications. Symmetric, also known as secret key, ciphers use the same key for encrypting and decrypting, so the sender and the receiver must both know -- and use -- the same secret key. The government classifies information in three categories: Confidential, Secret or Top Secret. All key lengths can be used to protect the Confidential and Secret level. Top Secret information requires either 192- or 256-bit key lengths. There are 10 rounds for

128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. A round consists of several processing steps that include substitution, transposition and mixing of the input plaintext to transform it into the final output of cipher text.

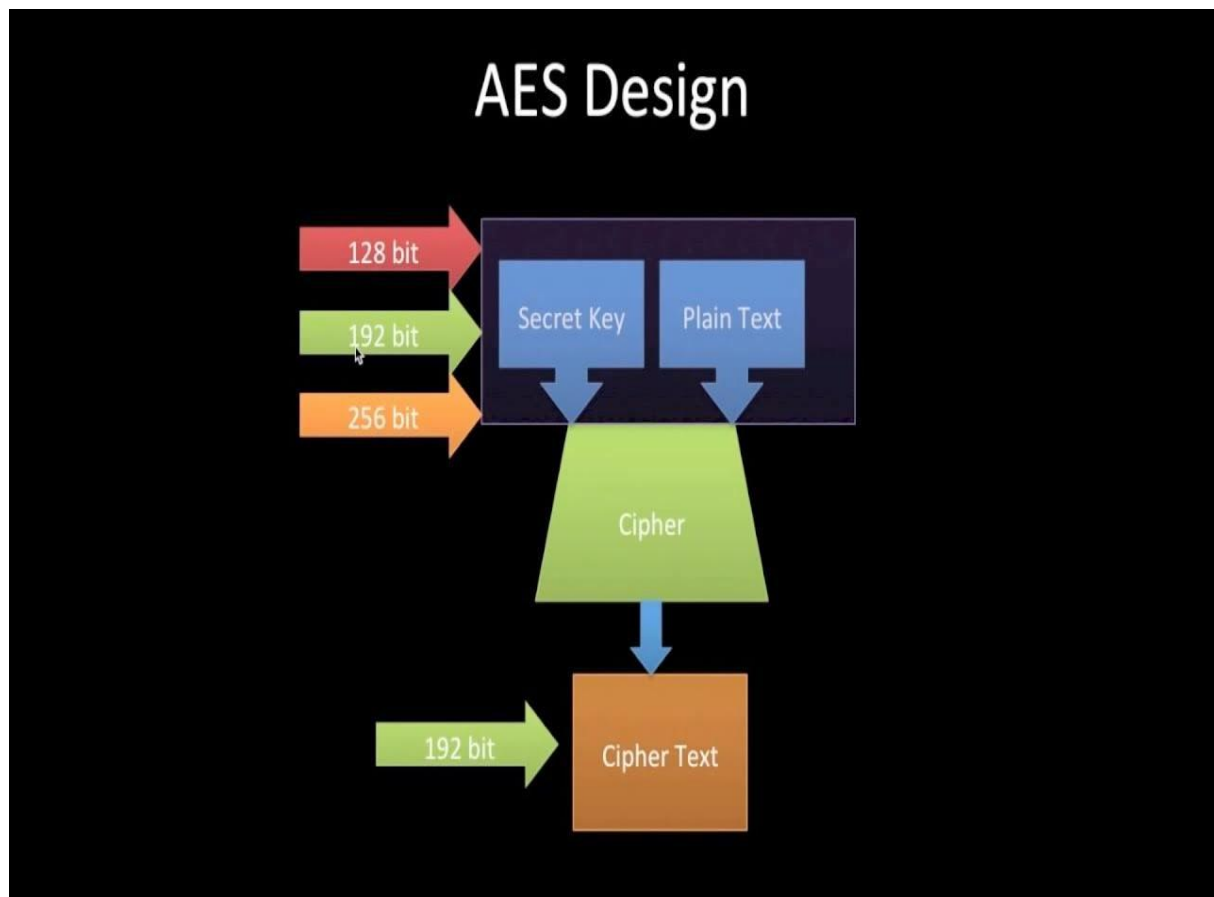


Figure 4.4: AES Design

AES features

NIST specified the new AES algorithm must be a block cipher capable of handling 128-bit blocks, using keys sized at 128, 192 and 256 bits. Other criteria for being chosen as the next AES algorithm included the following:

Security. Competing algorithms were to be judged on their ability to resist attack -- as compared to other submitted ciphers. Security strength was to be considered the most important factor in the competition.

Cost. Intended to be released on a global, nonexclusive and royalty-free basis, the candidate algorithms were to be evaluated on computational and memory efficiency.

Implementation. Factors to be considered included the algorithm's flexibility, suitability for hardware or software implementation, and overall simplicity.

MERGING OF CIPHER AND SHA 256

The SHA-256 algorithm is one flavor of SHA-2 (Secure Hash Algorithm 2), which was created by the National Security Agency in 2001 as a successor to SHA-1. SHA-256 is a patented cryptographic hash function that outputs a value that is 256 bits long. In encryption, data is transformed into a secure format that is unreadable unless the recipient has a key. In its encrypted form, the data may be of unlimited size, often just as long as when unencrypted. In hashing, by contrast, data of arbitrary size is mapped to data of fixed size. For example, a 512-bit string of data would be transformed into a 256-bit string through SHA-256 hashing. In cryptographic hashing, the hashed data is modified in a way that makes it completely unreadable. It would be virtually impossible to convert the 256-bit hash mentioned above back to its original 512-bit form. So why would you want to create a scrambled message that can't be recovered? The most common reason is to verify the content of data that must be kept secret. For example, hashing is used to verify the integrity of secure messages and files. The hash code of a secure file can be posted publicly so users who download the file can confirm they have an authentic version without the contents of the file being revealed. Hashes are similarly used to verify digital signatures. Password verification is a particularly important application for cryptographic hashing. Storing users' passwords in a plain-text document is a recipe for disaster; any hacker that manages to access the document would discover a treasure trove of unprotected passwords. That's why it's more secure to store the hash values of passwords instead. When a user enters a password, the hash value

is calculated and then compared with the table. If it matches one of the saved hashes, it's a valid password and the user can be permitted access. SHA-256 is used in some of the most popular authentication and encryption protocols, including SSL, TLS, IPsec, SSH, and PGP. In Unix and Linux, SHA-256 is used for secure password hashing. Cryptocurrencies such as Bitcoin use SHA-256 for verifying transactions.

SIGNIFICANCE OF SHA 256 ALGORITHM

SHA-256 is one of the most secure hashing functions on the market. The US government requires its agencies to protect certain sensitive information using SHA-256. While the exact details of how SHA-256 works are classified, we know that it is built a structure derived from a one-way compression function itself created with the Davies-Meyer structure from a specialized block cipher. Three properties make SHA-256 this secure. First, it is almost impossible to reconstruct the initial data from the hash value. A brute-force attack would need to make 2^{256} attempts to generate the initial data. Second, having two messages with the same hash value (called a collision) is extremely unlikely. With 2^{256} possible hash values (more than the number of atoms in the known universe), the likelihood of two being the same is infinitesimally, unimaginably small. Finally, a minor change to the original data alters the hash value so much that it's not apparent the new hash value is derived from similar data; this is known as the avalanche effect.

NODE API GENERATION

In this project node JS is used for API generation. An application programming interface (API) is a computing interface which defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc. An application programming interface (API) is a computing interface which defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc. Node JS is an open-source, cross-platform,

back-end, JavaScript runtime environment that executes JavaScript code outside a web browser. NodeJS lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node JS represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts. Node JS allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionalities. Modules are provided for file system I/O, networking (DNS, HTTP, TCP, TLS/SSL or UDP), binary data (buffers), cryptography functions, data streams, and other core functions. Node JS modules use an API designed to reduce the complexity of writing server applications.

DATABASE INTEGRATION

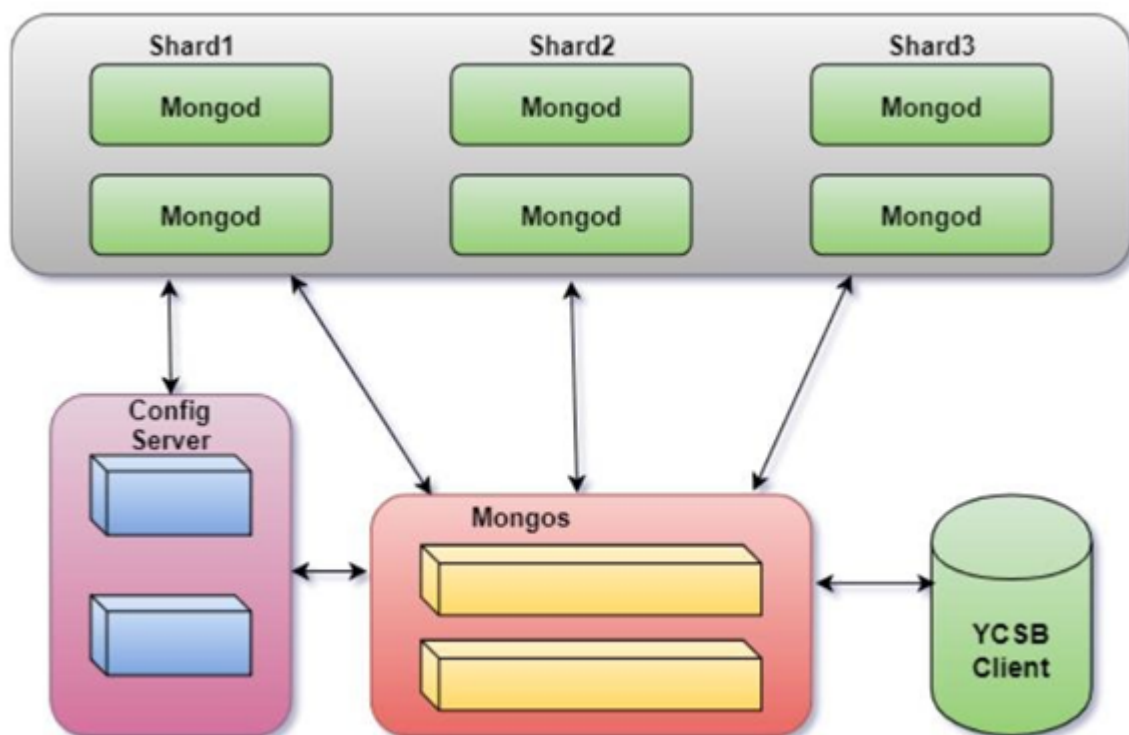


Fig 4.5: A Typical Database Integration

In this project mongo DB is used for database integration. MongoDB is a cross-platform, document-oriented database that provides, high performance, high

availability, and easy scalability. MongoDB works on concept of collection and document. MongoDB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of documents and function which is the equivalent of relational database tables.

Why Use MongoDB

Below are the few of the reasons as to why one should start using MongoDB

Document-oriented – Since MongoDB is a NoSQL type database, instead of having data in a relational type format, it stores the data in documents. This makes MongoDB very flexible and adaptable to real business world situation and requirements.

Ad hoc queries - MongoDB supports searching by field, range queries, and regular expression searches. Queries can be made to return specific fields within documents.

Indexing - Indexes can be created to improve the performance of searches within MongoDB. Any field in a MongoDB document can be indexed.

Replication - MongoDB can provide high availability with replica sets. A replica set consists of two or more mongo DB instances. Each replica set member may act in the role of the primary or secondary replica at any time. The primary replica is the main server which interacts with the client and performs all the read/write operations. The Secondary replicas maintain a copy of the data of the primary using built-in replication. When a primary replica fails, the replica set automatically switches over to the secondary and then it becomes the primary server.

Load balancing - MongoDB uses the concept of sharding to scale horizontally by splitting data across multiple MongoDB instances. MongoDB can run over multiple

servers, balancing the load and/or duplicating data to keep the system up and running in case of hardware failure.

How MongoDB Stores Data

As you know, RDMS stores data in tables format and uses structured query language (SQL) to query database. RDBMS also has pre-defined database schema based on the requirements and a set of rules to define the relation between fields in tables. But MongoDB stores data in documents in spite of tables. You can change the structure records (which is called as documents in MongoDB) simply by adding new fields or deleting existing ones. This ability of MongoDB helps you to represent hierarchical relationships, to store arrays, and other more complex structures easily. MongoDB provides high performance, high availability, easy scalability and out-of-the-box replication and auto-sharding.

Where to use MongoDB

Since, MongoDB is a NoSQL database, so we need to understand when and why we need to use this type of database in the real-life applications. Since in normal circumstances, MongoDB is always preferred by the developers or project managers when our main concern is to deal with large volume of data with a high performance. If we want to insert thousands of records in a second, then MongoDB is the best choice for that. Also, horizontal scaling (adding new columns) is not so easy process in any RDBMS systems. But in case of MongoDB, it is very much easy since it is a schema less database. Also, this type of work can be directly handled by the application automatically. There is no need to any type of administrative work to perform any type of horizontal scaling in the MongoDB. MongoDB is good for the below types of situation.

- E-Commerce type of product-based applications
- Blog and Content Management systems
- High Speed logging, caching in the Real time
- Need to maintain location wise Geospatial data

- For maintains data related to the Social and Networking types
- If application is a loosely coupled mechanism – means design may change at any point of time.

Database

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

Collection

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

Document

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

MongoDB Advantages

Since, MongoDB is not only database which can perform only CRUD (Create, Read, Update & Delete) operations with the data. Except these, MongoDB also contains so many important features due which MongoDB become the most popular database in the NoSQL category. Some of the important features are :

- MongoDB is a Schema less document type database.
- MongoDB support field, range based query, regular expression or regex etc for searching the data from the stored data.
- MongoDB is very easy to scale up or down.

- MongoDB basically uses internal memory for storing the working temporary datasets for which it is much faster.
- MongoDB support primary and secondary index on any fields.
- MongoDB supports replication of database.
- We can perform load balancing in the MongoDB by using Sharding. It scales the database horizontally by using Sharding.
- MongoDB can be used as a file storage system which is known as a GridFS.
- MongoDB provides the different ways to perform aggregation operations on the data like aggregation pipeline, map reduce or single objective aggregation commands.
- MongoDB can store any type of file which can be any size without effecting our stack
- MongoDB basically use JavaScript objects in place of procedure.
- MongoDB support special collection type like TTL (Time-To-Live) for data storage which expire at a certain time
- The dynamic database schema used in MongoDB is called the BSON

WEB DEVELOPMENT

React JS is JavaScript library used for building reusable UI components. According to React official documentation, following is the definition –React is a library for building user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native. React implements one-way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

Why use reactJS

React is very a simple and lightweight library that only deals with the view layer. It is not a beast like other MV* frameworks such as Angular or Ember. Any Javascript

developer can understand the basics and start developing an awesome web application after only a couple of days reading tutorial. As the React guide says Thinking in React may be a little different than you used to since it brings a new approach to the table, but it will become much easier and natural as you gain experience with it.

When you are about to develop a web application that involves high user interaction and view updates, like the new form builder on JotForm 4.0, you have to consider the possible performance issues. Although today's java script engines are fast enough to handle such complex applications, DOM manipulations are still not that fast. Updating DOM is usually the bottleneck when it comes to the web performance. React is trying to solve this problem by using something called virtual DOM; a DOM kept in memory. Any view changes are first reflected to virtual DOM, then an efficient diff algorithm compares the previous and current states of the virtual DOM and calculates the best way (minimum amount of updates needed) to apply these changes. Finally those updates are applied to the DOM to ensure minimum read/write time. This is the main reason behind React high performance.

One of the powerful sides of React is that it provides a good abstraction which means that it does not expose any complex internals to the user. Compare this to Angular. Those kind of details better be kept as internals to provide a clean abstraction. You only need to understand a component's life cycles, states and props to master React to accomplish everything you need. React does not dictate any patterns or architecture like MVC/MVVM, after all its only about the view layer and you are free to design your app's architecture in any way you see fit. Yet there is one good architecture that fits React really well called Flux.

Developer toolset is another important factor when you are choosing a development platform. There are two great tools you should be aware of: React Developer Tools and Redux Developer Tools. Both can be installed as Chrome extensions. React Developer Tools is great for inspecting react components within their hierarchy and also great for observing their current props and states. You can see a snapshot below. Here we are inspecting JotForm 4.0 components. Selected

component is highlighted on the left side, its position in the hierarchy and its props are displayed in the rightmost pane. If you are using the Redux library you should definitely take a look at Redux Developer Tools for Chrome. You can observe dispatched actions, current store states and watch changes on stores. You can also dispatch actions or modify stores and see changes reflected to your view instantly. Since all the updates are kept within the extension, time travel is even possible. You can record and go back to any state of your application for debug purposes.

Learning React comes with a bonus: React Native. React is not a 'write once run anywhere library', as the creators says, it's a 'learn once write anywhere' library. Yes, you can write native apps for Android and iOS using React Native. Although you will not be able to use the exact same code you wrote for web, you will be able to use the same methodology and the same architecture. The new version of JotForm took our team 1.5 years. Since we have been using React, we were able to continuously integrate and deploy the new version of our functionality to the existing form builder. On February 1st, we are releasing the new version fully to our 2.5 million users. The beta testers so far has been incredibly impressed with it, so we are extremely excited about this release.

Earlier creating a dynamic web page required a lot of complex coding. It needed specific HTML strings for dynamic web pages. Using React JS is no more an issue. It makes use of JSX which has a particular syntax and lets HTML quotes and tags to render particular subcomponents. In addition to this, it also supports building machine-readable code and mixes components into an individual variable file. In React regular JavaScript is not used for creating its templates.

Reusing components is the major advantage of React JS. Even Facebook has implemented React as it facilitates the reuse of system components. A developer can start with usual components like checkbox, button, etc. Once this is done you can move to wrapper components comprised of internal logic which makes it easier to manipulate and define objects used. This ensures consistency of application and facilitates code maintenance. Many companies like Walmart tend to use React JS. As all companies have common components like payment form elements, image

carousel, bread crumbs, for example. Here comes into picture the reuse of components. Also to enable version management and further install components there are coding standards defined to maintain uniformity. This also enhances the speed of the application, which enjoys more customer retention.

React JS uses downward data flow. This is because it ensures that smallest of changes that happen in child structure do not affect the parents. When an object is changed by a developer it needs to be made sure that only the particular objects are updated and that just its state is modified. The data is bound together with the object and this structure sees to it that code stability and application performance is maintained.

React JS is not a framework but a library that brings together objects and focuses on a particular thing to get it done properly. It is the view controller in MVC architecture. Any developer who knows JavaScript can easily understand React, learn its basics and build an overwhelming app.

JSX

JSX stands for JavaScript XML. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React.

It is faster than normal JavaScript as it performs optimizations while translating to regular JavaScript. It makes easier for us to create templates. Instead of separating the markup and logic in separated files, React uses components for this purpose. We will learn about components in details in further articles.

Components

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and returns HTML via a render function.

Components come in two types, Class components and Function components, in this tutorial we will concentrate on Class components.

Unidirectional data flow and Flux

React implements one-way data flow which makes it easy to reason about your app. Flux is a pattern that helps keeping your data unidirectional.

Why we use React JS

The main objective of React JS is to develop User Interfaces (UI) that improves the speed of the apps. It uses virtual DOM (JavaScript object), which improves the performance of the app. The JavaScript virtual DOM is faster than the regular DOM. We can use React JS on the client and server-side as well as with other frameworks. It uses component and data patterns that improve readability and helps to maintain larger apps.

ADVANTAGES OF REACT JS

- It facilitates the overall process of writing components
- It boosts productivity and facilitates further maintenance
- It ensures faster rendering
- It guarantees stable code
- It is SEO friendly
- It comes with a helpful developer toolset
- There is React Native for mobile app development
- It is focused and easy-to-learn
- It is backed by a strong community
- Uses virtual DOM which is a JavaScript object. This will improve apps performance, since JavaScript virtual DOM is faster than the regular DOM.
- Can be used on client and server side as well as with other frameworks.

4.6 SOFTWARE DESCRIPTION

4.6.1 MICROSOFT VISUAL STUDIO

In this project the Microsoft visual studio is used as the code editor.

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle.

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, Type Script, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past. The most basic edition of Visual Studio, the Community edition, is available free of charge. The slogan for Visual Studio Community edition is "Free, fully-featured IDE for students, open-source and individual developers". The currently supported Visual Studio version is 2019

Visual Studio is an Integrated Development Environment(IDE) developed by Microsoft to develop GUI(Graphical User Interface), console, Web applications, web

apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for mac OS.

Evolution of Visual Studio: The first version of VS(Visual Studio) was released in 1997, named as Visual Studio 97 having version number 5.0. The latest version of Visual Studio is 15.0 which was released on March 7, 2017. It is also termed as Visual Studio 2017. The supported .Net Framework Versions in latest Visual Studio is 3.5 to 4.7. Java was supported in old versions of Visual Studio but in the latest version doesn't provide any support for Java language.

Features

Code editor. Like any other IDE, it includes a code editor that supports syntax highlighting and code completion using IntelliSense for variables, functions, methods, loops, and LINQ queries.

- Debugger.
- Designer.
- Other tools.
- Extensibility.
- Previous products.
- Community.

4.6.2 Node JS

In this project node JS is used for API generation. Node JS is open-source, cross-platform, back-end, JavaScript runtime environment that executes JavaScript code outside a web browser. Node JS lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to

produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node JS represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts. Node JS allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionalities.

Why use node.js

Node.js is built upon Google Chrome's V8 runtime written in C++, built for multiple operating systems and super fast. Both V8 and Node are frequently updated, with performance boosters, security patches, and support for modern JavaScript features. The use of JavaScript also means that transforming JSON data the most common data interchange format on the Web is fast by default. The single threaded, event-driven architecture of Node.js allows it to handle multiple simultaneous connections efficiently. Most of the popular web platforms create an additional thread for each new request, using up RAM for the whole time it takes to process it. Node, on the other hand, operates on a single thread, making use of the event loop and call backs for I/O operations, delegating tasks such as database operations as soon as possible. This allows it to handle hundreds of thousands or even a million concurrent connections. What's more, Node.js embraces app scalability from the get-go, with powerful features such as the Cluster module enabling load balancing over multiple CPU cores. Robust tools, such as the PM2 process manager, make it even easier to monitor, optimize and deploy Node applications.

How does node.js works

Node.js is an epitome of an exceptionally customizable and scalable tech. The server engine utilizes an event-based, non-blocking I/O model. This makes the adaptation of Java script easier to the machine language providing execution of the code superfast. Thanks to Java script and Node.js, the code operates faster in server-to-client direction. This enhances the performance ability of the web applications to the next level. To be more precise, web application development in

Node.js ensures a steady and secure non-blocking I/O model, simplifying the code beautifully.

Node.js runs over Google's V8 JavaScript engine, where web applications are event-based in an asynchronous manner. Node.js platform uses a "*single-threaded event loop*". So, how exactly does Node.js handle concurrent requests along with a single-threaded model? Well! *Multi-threaded request-response* architecture is an event loop which is much slower and unable to handle multiple concurrent threads at a time.

The platform does not follow a similar request/response multi-threaded stateless model; instead, it goes by a simplified single-threaded event loop model. As per Node.js developers, a specific library called "*Libuv*" provides this mechanism known as an event loop. This Node.js processing model is majorly based on JavaScript event-based model along with the *callback mechanism*.

How is node.js being used in web development?

Node.js uses JavaScript, a programming language that makes the web app development simplest for developers for all the browsers. Generally, to build server-side applications, JavaScript required to consider additional programming languages such as Python, PHP, Ruby, etc. But while creating the web app with Node.js, they do not need support from other languages.

Why Node.js is so popular?

Node.js is the part of the JavaScript that is the main language to build the web application. Nowadays, JavaScript is the only option with the developers to develop web apps in the browser. Node.js is an advanced JavaScript that has revolutionized server-based development.

One of the biggest reasons why Node.js is so popular is because it uses JavaScript as its main language to build web applications. And to be honest, JavaScript is now the only choice to develop web applications in the browser. What

more! A new and robust framework is introduced quite frequently to woo developers. With Node.js, JavaScript has literally revolutionized on the server. The language is common to most web developers, and is certainly driving the world today. And according to experts, the trend will not fade away soon. Since JavaScript is a language that most developers know or have used at some point of time, the transition from another web technology to Node.js is a breeze. This makes it a preferred choice among web developers.

Node.js is primarily a JavaScript runtime that is powered by V8, developed by Google for use in Chrome. V8 has the ability to compile and execute JavaScript at lightning fast speed, mainly because it compiles JavaScript into a native machine code. In addition to this, Node.js has a magical event loop, which is a single thread performing all I/O operations in an asynchronous manner. In Node.js, when an application has to perform I/O operation, it sends asynchronous tasks and callback action to the event loop, and then continues to perform the rest of the program. On completion of sync operation, the event loop automatically returns to the task to execute callback. This is much unlike traditional looping, which consumes a lot of memory and is exceptionally difficult to execute. Thus, reading/writing to file system, network connections, and to the database is executed very fast in Node. It allows developers to build fast and highly scalable network applications that are capable of handling bulk amounts of simultaneous connections having high throughput.

Typically, Node.js uses a basic event-driven architecture. This means everything executed on it, including every single operation and call, is a series of asynchronous call back. This enables Node to run on a single thread as unlike other web technologies where a new thread is spawned for every client request. This not only makes it light in weight, but also forms the foundation of Node's non blocking I/O feature.

Traditionally built applications become less adaptive and rigid over time as new requirements are fed. Eventually, they start creaking under the stress they were not built for. However, developing new services using Node.js is comparatively easier. With Node, a bunch of small applications is built instead of a single, large application.

This allows making changes or adding a new functionality with much ease, without having to make changes deep inside the code-base.

Another key benefit of Node.js is language re-use. Other web technologies like Spring or ASP.NET requires developers to have knowledge in another language to write code on the server-side, be it VB.NET, Java, or C#. This means all functions necessarily include two languages – one on the client-side and the other on server-side. On the contrary, Node uses only JavaScript for the client and server side. Thus, web developers have to interact only in a single language across all layers. Adding to it, this can be re-used over and PayPal uses Node.js, and has reported doubling the number of requests per-second and reducing the response time by 35%. On the other hand, Wal Mart, the retail giant had a superb experience with Node.js in 2013, when they put all their mobile-based traffic via Node.js on Black Friday, the busiest shopping time of the year. Amazingly, on Black Friday, Wal Mart servers did not go over a mere 1% CPU utilization and yet they deploy with 200,000,000 users online. LinkedIn, the global networking site, moved from Ruby to Node to handle their mobile traffic, and reduced the number of servers to 30 to 3, i.e. almost 90% reduction. The new system was up to 20 times faster. All these figures clearly indicate performance capability of Node.js. over again for future communications.

Features of Node.js

Following are some of the important features that make Node.js the first choice of software architects.

- Asynchronous and Event Driven – All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- Very Fast – Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.

- Single Threaded but Highly Scalable – Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- No Buffering – Node.js applications never buffer any data. These applications simply output the data in chunks.
- License – Node.js is released under the MIT license

CHAPTER – 5

RESULTS AND DISCUSSIONS

5.1 FINAL RESULTS OBTAINED

A web application using a java script framework react JS was successfully developed, to enter the patient details to perform diagnosis for diabetes.

The below screenshot shows the web application for the project

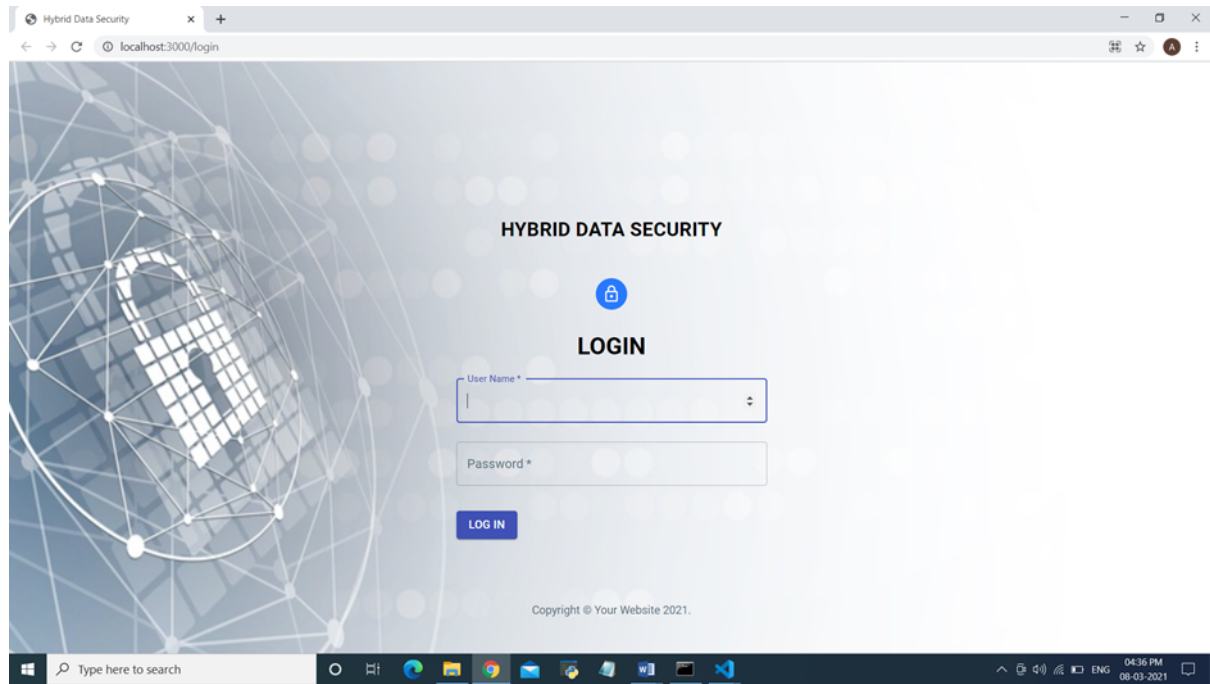


Fig 5.1: Login Page

The user enters his/her login credentials to enter into the webpage.

Once the user has successfully logged in, the home page is displayed to the user. The patient enters his/her medical details to be used to perform diagnosis for diabetes. On clicking the save location button, the location is saved. The below screenshot shows the form onto which the user uploads the medical data for diagnosis of diabetes. This form has the very confidential information which the patient is looks to be saved with encryption.

Hybrid Data Security

Patient Details

Log Out

Patient Registration Form

Patient FirstName * kishan

Patient LastName * r

Gender Male

Age * 23

Phone Number * 423353334

E-Mail Address * kishan@gmail.com

PAN Card Number * 343535353

Pregnancies * 0

BMI * 80

Blood Glucose Level * 67

Blood Pressure Level * 26

Skin Thickness * 50

Insulin * 26

Purpose For Visit * checkup

Address *

Fig 5.2: Registration Form

After filling the detail the encrypted result of the filled data is displayed and stored in the server securely using the hybrid secure algorithms. Once the user hits the submit button all the personal data of the user and the medical report is encrypted and securely stored in the mongo database which is efficient data base storage.

Hybrid Data Security

Patient Details

Log Out

BMI * 80

Blood Glucose Level * 67

Blood Pressure Level * 26

Skin Thickness * 50

Insulin * 26

Purpose For Visit * checkup

Address * omr

City * chennai

State/Province/Region * tamilnadu

Zip / Postal code * 600003

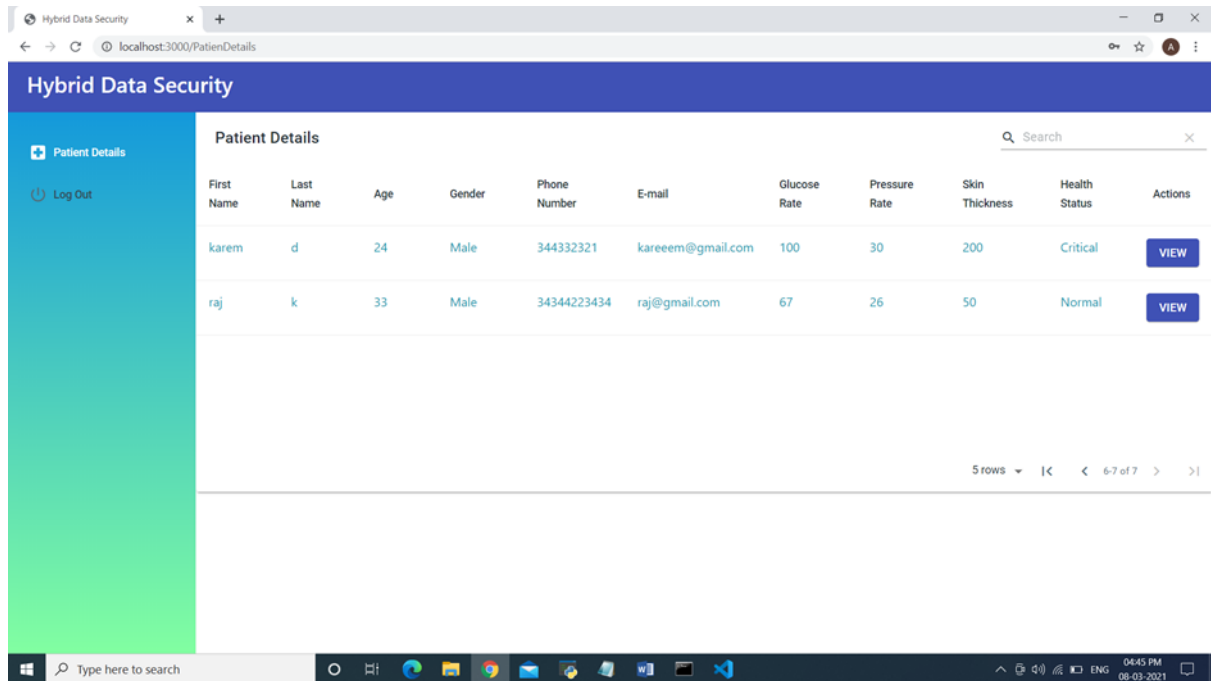
Country * india

SAVE DETAILS

U2fSdGVkXV+IUNGQf... Patient Given Informations are save successfully !!!!!!!!!!!!!!! ... UJbiI2SA4c6GTxKL2xbqwQvnVjeFUB0IzOIjgIzVqM+4EGLHwRXBmxanUyd59mhbcBEIO

Fig 5.3: Details Encrypted And Saved

Now all the patient personal data and the medical report is saved in database with disease prediction which determines whether the patient is normal or critical. And then doctor login with his login ID in the home page where he can see the patient data along with the status of disease predicted whether the patient is normal or critical by machine learning algorithm known as logistic regression.



The screenshot shows a web application titled "Hybrid Data Security". On the left is a sidebar with a "Patient Details" section and a "Log Out" button. The main area displays a table of patient details. The table has columns for First Name, Last Name, Age, Gender, Phone Number, E-mail, Glucose Rate, Pressure Rate, Skin Thickness, Health Status, and Actions. Two patients are listed: "karem d" (Age 24, Male, Glucose 100, Pressure 30, Skin 200, Critical) and "raj k" (Age 33, Male, Glucose 67, Pressure 26, Skin 50, Normal). Each row has a "VIEW" button. At the bottom right of the table, it says "5 rows" and "6-7 of 7".

First Name	Last Name	Age	Gender	Phone Number	E-mail	Glucose Rate	Pressure Rate	Skin Thickness	Health Status	Actions
karem	d	24	Male	344332321	kareem@gmail.com	100	30	200	Critical	VIEW
raj	k	33	Male	34344223434	raj@gmail.com	67	26	50	Normal	VIEW

Fig 5.4: Doctor Web Page

Here doctor checks whether the health status is normal or critical depending on he will advice the patient to take the treatment.

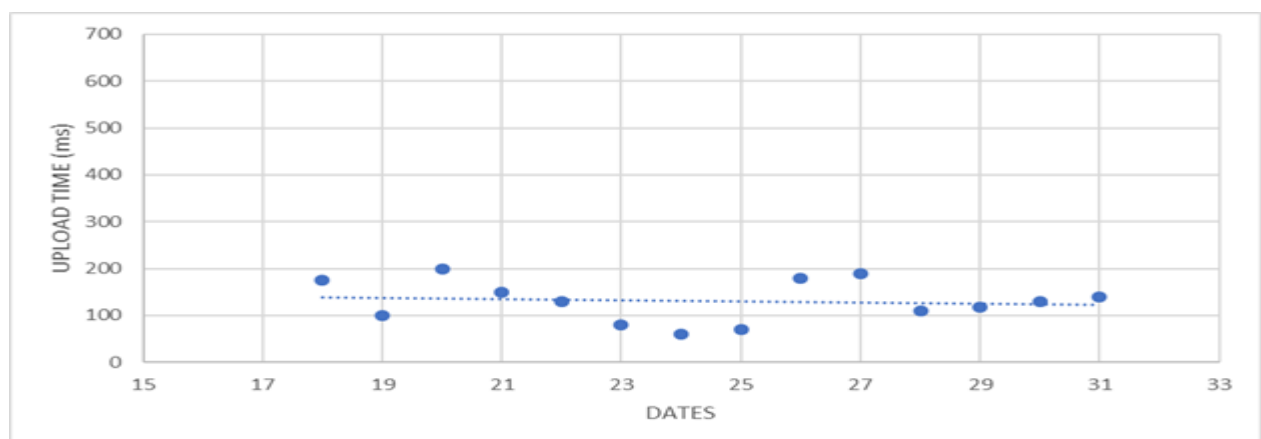


Fig 5.5: Data Query takes 100 to 200ms

Thus, from the above results and discussion, it is clear that we have efficiently made a project for successfully securing the data to be transmitted through wireless communications with a web application developed using a java script framework React JS to encrypt and send medical data of a patient through wireless communication. Thus, we have successfully implemented the scope of the project.

5.2 DISCUSSIONS

In the existing system, presents a project to securely transmit sensor data over a wireless communication network. The sensor data is transmitted in packets that is secured using the encryption algorithm. A linear encryption scheme is used to secure the data packets sent through wireless communication network. In the proposed system, we have overcome the disadvantages of the existing system by the implementation of a web application using a java script framework react JS to upload the medical data for diagnosis of diabetes and securely transmit the medical data using novel algorithms based on the combinations of JWT, block chain hash algorithm and cipher AES algorithm. This system is mainly applicable in the medical and banking industry.

FACTORS	EXISTING SYSTEM	PROPOSED SYSTEM
ALGORITHM	Linear encryption scheme	JWT, SHA-256, CRYPTO AES ALGORITHM
ATTACKS	Stealthy Attacks on Remote State Estimation	Packet Sniffing, Rouge Access Point, other wireless data transfer attacks.

EFFICIENCY	Uses only a single encryption algorithm	Uses three algorithms
------------	---	-----------------------

Table 5.1 Comparison of system

CHAPTER – 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Thus a medical application is developed to perform diagnosis and to predict if the patient is diabetic or normal using machine learning algorithm. So the chances of making mistakes will be reduced as the diagnoses is first done by the trained machine learning model by using logistic regression. It not only does the diagnoses of diabetes but also securely encrypts the patient personal data with two level of encryption such as crypto AES algorithm and SHA 256 so those data are not available to the person outside it will be available to the patient and doctor only so privacy and confidentiality is ensured against the data. After all both the patient data and diagnosed information will be stored in mongo database which is an efficient database storage. For developing medical web application we use react JS which is known for its efficiency. And an end to end data security is provided using hybrid secure algorithm which ensures security over the wireless data transfer server to the database and a separate patient and doctor login credentials are provided for data transmission and access. Thus this project is used to provide a solution to perform diagnosis for diabetics and securely transmit data through a wireless communication medium using a web application developed using a java script framework react JS. And by using this project, we can protect data from unauthorized access. Thus, this project provides an affordable and efficient means to protect and preserve data as well as diagnosing diabetes.

6.2 FUTURE WORK

In the coming future, we review the application of the project to determine technology in the fields that require data security and privacy. In the banking and medical field, they are more chance to develop or convert this project in many ways. Thus, this project has an efficient scope in coming future where data can be transmitted securely over a wireless communication.

REFERENCES

- [1] J. Andrić, D. Oreški and T. Kišasondi, "Analysis of phishing attacks against students," 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia,, pp. 1423-1429,2016
- [2] M. Masoumi, "Novel Hybrid CMOS/Memristor Implementation of the AES Algorithm Robust Against Differential Power Analysis Attack," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 7, pp. 1314-1318, 2020
- [3] N. M. M. Alhag and Y. A. Mohamed, An Enhancement of Data Encryption Standards Algorithm (DES),International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum,, pp. 1-6,2018
- [4] T. V. Sathyanarayana and L. M. I. Sheela, "Data security in cloud computing, International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), Chennai, India, pp. 822-827, 2013
- [5] Xu, L., Sun, S., Yuan, X., Liu, J. K., Zuo, C., & Chungen, X. (Accepted/In press). Enabling authorized encrypted search for multi-authority medical databases, Volume 9, pp 534-546,2016
- [6] Rahimli, Ailar. A Review of Cloud Computing Technology Solution for Healthcare System. Research Journal of Applied Sciences, Engineering and Technology.vol. 8. 2150-2153, 2014

- [7] Y. Li, H. Wang and K. Sun, "Personal Information in Passwords and Its Security Implications," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 10, pp. 2320-2333, Oct. 2017
- [8] M. Mangia, A. Marchioni, F. Pareschi, R. Rovatti and G. Setti, "Chained Compressed Sensing: A Blockchain-Inspired Approach for Low-Cost Security in IoT Sensing," in IEEE Internet of Things Journal, vol. 6, no. 4, pp. 6465-6475, Aug. 2019
- [9] E. Bertino, "Data Security and Privacy: Concepts, Approaches, and Research Directions, IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, pp. 400-407, 2016
- [10] H. T. Yew, M. F. Ng, S. Z. Ping, S. K. Chung, A. Chekima and J. A. Dargham, "IoT Based Real-Time Remote Patient Monitoring System," 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), Langkawi, Malaysia, pp. 176-179, 2020
- [11] A. N. Asat, A. F. Mahat, R. Hassan and A. S. Ahmed, "Development of dengue detection and prevention system (Deng-E) based upon open data in 6th International Conference on Electrical Engineering and Informatics (ICEEI), Langkawi, Malaysia, pp. 1-6, 2017
- [12] Xu Ruzhi, Guo jian and Deng Liwu, "A database security gateway to the detection of SQL attacks, 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, China, pp. V3-537-V3-540, 2010
- [13] N. Agrawal and S. Tapaswi, "Defense Mechanisms Against DDoS Attacks in a Cloud Computing Environment: State-of-the-Art and Research Challenges," in IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3769-3795, 2019
- [14] H. Rady, H. Hossam, M. S. Saied and H. Mostafa, "Memristor-Based AES Key Generation for Low Power IoT Hardware Security Modules," 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), Dallas, TX, USA, pp. 231-234, 2019

[15] S. Khadka, "Privacy, security and storage issues in medical data management," Third Asian Himalayas International Conference on Internet, Kathmundu, Nepal, pp. 1-5,2012

[16] L. Chen and H. Chen, "Ensuring Dynamic Data Integrity with Public Auditability for Cloud Storage," International Conference on Computer Science and Service System, Nanjing, China, pp. 711-714,2012

[17] A. Razaque et al., "Secure data sharing in multi-clouds,International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, pp. 1909-1913,2016

[18] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "Coding schemes for securing cyber-physical systems against stealthy data injection attacks," IEEE Trans. Control Network Syst., vol. 4, no. 1, pp. 106–117, 2016.

[19] D. Shi, Z. Guo, K. H. Johansson, and L. Shi, "Causality countermeasures for anomaly detection in cyber-physical systems," IEEE Trans. Autom. Control, vol. 63, no. 2, pp. 386–401, 2017.

[20] X. Ren and Y. Mo, "Secure detection: Performance metric and sensor deployment strategy," IEEE Trans. Signal Process., vol. 66, no. 17, pp. 4450–4460, 2018.

APPENDIX

A.SAMPLE CODE

MACHINE LEARNING CODE

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import roc_curve
from sklearn.svm import SVC
import requests
```

```

import sys
import os
import glob
from subprocess import run,PIPE
import pandas as pd
import json
#import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import roc_auc_score
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score
#from ubidots import ApiClient
import json
import urllib.request
import urllib.parse
from flask import jsonify
from flask_cors import CORS, cross_origin
import requests
from flask import Flask, app,request
app = Flask(__name__)
cors = CORS(app)
app.config['CORS_HEADERS'] = 'Content-Type'
face_flag = 0
@app.route('/heart', methods=['GET','POST'])
@cross_origin()

```

```

def login():
    if request.method == 'POST':
        # print("called")
        print(request.json['data_set'])
        result = input(request.json['data_set'])
        return result

def data_analysis():
    global regressor
    dataset=pd.read_csv('diabetes3.csv')
    dataset.head()

columnlist=['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','Age',
'Outcome']

for i in columnlist:
    labelencoder_X=LabelEncoder()
    dataset[i]=labelencoder_X.fit_transform(dataset[i])
print(dataset)
X=dataset.iloc[:,0:-1].values
X.shape
Y=dataset.iloc[:,7:].values
Y.shape

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=0)

regressor = LogisticRegression()
H=regressor.fit(X_train,Y_train)
Y_pred = regressor.predict(X_test)
Y_pred

classifier = SVC(kernel = 'linear', C = 1, random_state = 0, probability = True)
classifier.fit(X_train, Y_train)

```

```

probs = classifier.predict_proba(X_test)
# print(probs)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
auc = roc_auc_score(Y_test, probs)
print('AUC: %.3f % auc)
# clf = svm.SVC(kernel='linear', C=1.0)
Y_pred = regressor.predict(X_test)
Y_pred
print("ACC:", roc_auc_score(Y_test, Y_pred))

return regressor
def input(s):
    data_analysis()
    patient_condition='Not found'
    x = list(map(float, s))
    print(x)
    # prediction
    # 102 0 0 160000 0 1 0 0 --0(Normal)
    # 102 0 0 60000 0 1 0 0 --1(Critical)
    Y_pred = regressor.predict([x])
    # Y_pred
    print(Y_pred)
    if Y_pred[0] == 1:
        a = 1
        # patient_condition='Patient is critical'
        patient_condition='Critical'

```

```

        print("Patient is critical")
    else:
        b = 0
        # patient_condition='Patient is normal'
        patient_condition='Normal'
    print(patient_condition)
    return ( {"data" :patient_condition})
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5008)

```

WEB APPLICATION CODE

```

import React, { useState, useEffect } from "react";
import Grid from '@material-ui/core/Grid';
import { makeStyles } from '@material-ui/core/styles';
import Typography from '@material-ui/core/Typography';
import TextField from '@material-ui/core/TextField';
import FormControlLabel from '@material-ui/core/FormControlLabel';
import Checkbox from '@material-ui/core/Checkbox';
import Select from '@material-ui/core/Select';
import InputLabel from '@material-ui/core/InputLabel';
import MenuItem from '@material-ui/core/MenuItem';
import FormControl from '@material-ui/core/FormControl';
import Box from '@material-ui/core/Box';
import Button from '@material-ui/core/Button';
import FormGroup from '@material-ui/core/FormGroup';
import axios from 'axios';
import Snackbar from '@material-ui/core/Snackbar';
import MuiAlert from '@material-ui/lab/Alert';

```

```

import sha256 from "sha256"
import initJWTService from 'jwt-service';
import { CircularProgress } from '@material-ui/core';
import SavedPatientDetails from '../SavedPatientDetails/SavedPatientDetails'
function Alert(props) {
    return <MuiAlert elevation={6} variant="filled" {...props} />;
}
const useStyles = makeStyles((theme) => ({
    formControl: {
        margin: theme.spacing(1),
        minWidth: 600,
    },

```

B. Plagiarism Report

hybrid

ORIGINALITY REPORT

3%

SIMILARITY INDEX

2%

INTERNET SOURCES

2%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

ieeexplore.ieee.org

Internet Source

1%

2

Lei Xu, Shifeng Sun, Xingliang Yuan, Joseph K. Liu, Cong Zuo, Chungun Xu. "Enabling Authorized Encrypted Search for Multi-Authority Medical Databases", IEEE Transactions on Emerging Topics in Computing, 2021

Publication

1%

3

www.jgenng.com

Internet Source

1%

4

www.ijitee.org

Internet Source

<1%

5

orca.cf.ac.uk

Internet Source

<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off