

A Multi-Layer Security System for Data Access Control, Authentication, and Authorization

by

Tamanna Kaiser
22141060

Rafa Siddiqua
22141064

Md. Main Uddin Hasan
22141049

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
May 2022

© 2022. Brac University
All rights reserved.

Declaration

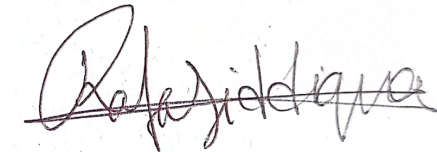
It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Tamanna Kaiser
22141060



Rafa Siddiqua
22141064



Md. Main Uddin Hasan
22141049

Approval

The thesis titled “A Multi-Layer Security System for Data Access Control, Authentication, and Authorization” submitted by

1. Tamanna Kaiser (22141060)
2. Rafa Siddiqua (22141064)
3. Md. Main Uddin Hasan (22141049)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 29, 2022.

Examining Committee:

Supervisor:
(Member)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Muhammad Iqbal Hossain, PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Multi-Factor-Authentication is one of the most used services nowadays by all kinds of people, especially by many organizations. People use this service to authorize their stored data and to access it without any security disturbance. As the use of different storage systems for various types of data is increasing, we need to focus on security. Any kind of security threat can be a great threat to any company. While checking the most recent surveys of different security issues we find that 94% of organizations are moderate to extremely concerned about data security. According to research from Intel, insider threats are responsible for an incredible 43 percent of all breaches. Half are intentional and half are accidental. [62][42] In this paper, we are proposing a fully secure data flow for data security with data encryption, IAM, IDAAS, AAAS, MFA, and SAML to prevent unauthorized data access and insecure data storage. With these models, we can control access and authorization to secure both data storage and stored data access.

Here, this system focuses on secure authentication, authorization, and access data control by using a multi-layer security system. In this multi-layer security system, there will be Multi-Factor-Authentication along with Two-Factor-Authentication via email or phone. To ensure security, this system has a key-exchange system where the primary key and secondary key will be generated as a One-Time-Password for super admin and co-admin. Then the system will check the OTP in the Key Exchange process. In the future, the system will also be adding biometric authentication in this system for the co-admins. There will be two biometric options which are Irish Scanner and Fingerprint so that we can ensure the high-level authentication security for this system.

Keywords: TFA, MFA, Real-time computing, Multi-layered, Security System , Key-exchange, KSU key exchange, session timeout, bio-metric Super-admin, Co-admin, Verification, Primary server, Backup server, Data server, URL, Launch, Request, OTP, actor.

Dedication

This work is dedicated primarily to all the support and encouragement that our parents have given us so far. We also devote our work to our supervisor and co-supervisor, who are indispensable for developing final products.

Acknowledgement

We are very grateful to our supervisor, Dr. Sadia Hamid Kazi, Chairperson and Associate Professor, BRAC University, Dhaka, and Co-supervisor Dr. Muhammad Iqbal Hossain, Assistant Professor, BRAC University, Dhaka for enabling us to work on an interesting topic. They have allowed us to explore the subject of our thesis by putting themselves in the background, and we could not have asked for more. We would also like to thank our other course's faculty and our friends whose support has kept us and every member of our team moving forward for having the dedication to work until the very end.

Contents

Declaration	i
Approval	ii
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vi
List of Figures	x
List of Tables	xii
1 Introduction	1
1.1 Aim of the project	1
1.2 Background	2
1.2.1 Data Breach	2
1.2.2 Insufficient Identity, Credential, Access and Key Management	2
1.2.3 Insider Threats, Unauthorized Access	3
1.2.4 Data privacy and confidentiality	3
1.3 Motivation & Research Objective	3
2 Literature Review	5
2.1 Related Work	6
2.2 Methodology	7
2.2.1 IAM	7
2.2.2 IDAAS	9
2.2.3 AAAS	10
2.2.4 MFA	12
2.2.5 SAML	13
2.2.6 AES Encryption	14
2.2.7 Key Exchange	15
3 System Requirements Specification	16
3.1 Functional Requirements	16
3.2 Non-Functional Requirements	16
3.3 Software Components	17

3.4	Organizational Security Impact	19
3.5	Ethical Consideration	20
3.6	Risk Management & Safety	20
3.7	Economical Impact	20
3.8	Economic Feasibility	21
4	KSU Key Exchange Algorithm	23
4.1	Introducing the KSU Key Exchange Algorithm	23
4.2	KSU Key Exchange Algorithm Diagram & Analysis	24
4.3	KSU Key Exchange Algorithm Cyclomatic Complexity	25
4.4	Comparison with other key exchange algorithm	33
4.5	Nobility of KSU Algorithm	34
4.6	KSU Key Exchange Algorithm Security Against Common Attacks . .	34
5	System Analysis	36
5.1	System Requirements	36
5.2	System Design	36
5.2.1	WorkFlow	37
5.2.2	USE CASE	38
5.2.3	Sequence Diagram	39
5.3	System Architecture	40
5.4	System Architecture Analysis	44
5.4.1	Login with TFA or MFA	44
5.4.2	Creating Roles and Actors	44
5.4.3	Creating Task and controlling	44
5.4.4	Task viewing by the user	45
5.4.5	Data entry Clerk	45
6	System Security Analysis	46
6.1	Security	46
6.2	Result analysis	47
6.3	System Vulnerability Analysis	48
6.4	System Model Comparison	48
6.5	Nobility of the system	49
7	System Implementation	50
7.1	Implementation showcasing	50
7.1.1	Super Admin Activity	50
7.1.2	Co-Admin Activity	51
7.1.3	Task Publishing	53
7.2	Code Explanation	54
7.2.1	Request and Key Generation	54
7.2.2	Request approval status	55
7.2.3	Request Confirmation and Key verification	56
8	Conclusion	57
8.1	Advantages	57
8.2	Limitations	58
8.3	Future Work	58

8.4 Conclusion	59
Bibliography	64

List of Figures

2.1	IAM-phases [48]	8
2.2	IAM Components [48]	9
2.3	IDAAS as a model [39]	10
2.4	AAAS Model [39]	11
2.5	MFA [24]	13
2.6	The SAML Authentication Process [4]	14
2.7	AES Design [45]	15
2.8	Key Exchange Between Two Parties [60]	15
3.1	PHP [47]	17
3.2	Phpmyadmin [18]	18
3.3	Diagram.net [1]	18
3.4	Laravel [41]	18
3.5	CSS 3 [22]	18
3.6	HTML [3]	19
3.7	Javascript [6]	19
3.8	XAMPP [56]	19
4.1	KSU Key-Exchange Algorithm Diagram	24
4.2	Flowchart-Initialization	26
4.3	Flowchart-check data and request update part 1	27
4.4	Flowchart-check data and request update part 2	28
4.5	Flowchart-flowchart-check data and request update part 3	29
4.6	Flowchart-check key part 1	30
4.7	Flowchart-check key part 2	31
4.8	Flowchart-flowchart-check key part 3	32
4.9	Flowchart-update table part 1	32
4.10	Flowchart-flowchart-update table part 2	33
5.1	Workflow Diagram for the Security System	37
5.2	USE-CASE Diagram for the Security System	38
5.3	Sequence Diagram for the Security System	39
5.4	Part One (SuperAdmin)	40
5.5	Part Two (Co-Admin)	41
5.6	Part One Part Three (User)	42
5.7	Part four (Data Entry Clerk, including the database)	43
6.1	Security Layers of the System	47
7.1	Super Admin Dashboard	50

7.2	User or admin creation	51
7.3	Task Creation	51
7.4	Task Interface	52
7.5	Task Assign to user and request send	52
7.6	Key Generation Database	52
7.7	Key Exchange Input UI for verification	53
7.8	Key Exchange Verification Success	53
7.9	After Successful Verification	53
7.10	After Publishing Task to the user	54
7.11	Send Request and Key Generation Part 1	54
7.12	Send Request and Key Generation part 2	55
7.13	Task Approve and Approve Status part 1	55
7.14	Task Approve and Approve Status part 2	55
7.15	Request Confirmation Status and key verification part 1	56
7.16	Request Confirmation Status and key verification part 2	56

List of Tables

3.1	Operation Estimated Cost Project	21
3.2	Estimated Hardware Cost Project	21
3.3	Estimated Software Cost Project	21
3.4	Estimated Maintenance Cost of the Project	22
4.1	KSU Key Exchange Algorithm	23

Chapter 1

Introduction

In this age of modern technology, every user is becoming more concerned about security and privacy. Every system is considered a secure system when it can fulfill the following criteria: accountability, accessibility, authentication, authorization, confidentiality, and reliability. This system proposed a multi-layer-security service based on authorization and authentication. Because the security of this system is built on identity verification processes, Two-Factor-Authentication, Multi-Factor-Authentication, and key exchange algorithms, KSU , and key hashing are important components. In this system, primary and secondary one-time-passwords will be generated and those will be secured through the hash function. To run the system, there will be several defined roles for individuals, and all those individuals will manage and support the infrastructure. Furthermore, because this service will demand email for those one-time-keys, they will be provided to the co-admins through email for verification, which will help the user to launch the significant task. As a result, the user may be able to obtain the key and gain access to the system. This system is fully designed to give full security to any system by providing secure access with authentication, authorization, and the newly introduced key-exchange algorithm, KSU.

1.1 Aim of the project

Multi-Layer-Security is the technology to secure and protect users' data which is stored in an architecture via system providers. People are using different system providers to store their data, and their main concern is the security of their data nowadays. Authentication has all the security needed, but still, there are some issues people are facing while using different security services. From some of the recent surveys, we know there are huge cybersecurity threats that we have to solve. The Different Security Alliance is already researching this security issue, and they have some good models for service security. We analyzed and discovered that these models might be used in tandem to avoid data leaks. Mis-configuration was identified as the most critical security issue by organizations (68 percent), followed by unauthorized access (58 percent), unprotected interfaces (52 percent), and account hijacking (50 percent). [62] To address these security concerns, we suggest a comprehensive approach that can mitigate the risk of all of the vulnerabilities raised in the surveys. Furthermore, insider threats are a serious security concern for any organization. Keeping that in mind, we want to work with Identity Access Man-

agement (IAM), which includes Identity as a Service (IDAAS) and Multi-factor Authentication (MFA) for access control. Then for authorization, we like to work with Authorization as a Service (AAAS). Next, for data security, we want to include Advanced Encryption Standard (AES) and for secure authorization, we will use Security Assertion Markup Language (SAML). To conclude, this proposed model is unique because we are giving options to the users to choose between layers and take as much service as they require based on their system.

1.2 Background

When establishing a multi-factor authentication system, developers must be aware of the risks associated with unlawful access to personal data, the risks associated with a lack of scalable identity access, insider threats, and the necessity of data privacy and confidentiality.

1.2.1 Data Breach

Unauthorized access to confidential data, burglary, information leakage, and unauthorized broadcast of intelligence information are all examples of data breaches.[15] Data breaches may reveal personal information such as credit card numbers, Social Security numbers, personal health information (PHI), personally identifiable information (PII), or business information such as client lists, software source codes, trade secrets, or intellectual property. Sensitive data and information are transported from an enterprise to a third-party due to a data breach. This will result in a loss of control over the data in terms of storage and privacy [20].

1.2.2 Insufficient Identity, Credential, Access and Key Management

A lack of scalable identity access management systems, the inability to apply multi factor authentication, the usage of weak passwords, and the poor management of keys and certificates may all lead to a slew of dangers. According to the CSA, hostile actors acting as legitimate users, operators, or developers have the ability to access, edit, and delete data, as well as issue control plans and execute administrative responsibilities. They can monitor data transmissions or transmit malicious software that seems to be from a reputable source.[51] As a result, insufficient identity, credential, or key management might allow for illicit data access and potentially catastrophic damage to enterprises or end users. To communicate with other programs, most online apps must employ numerous application program interfaces (API). They're called application fundamentals as they help developers leverage online services like Facebook and Pinterest, as well as services like Google Maps and Dropbox, and any code that has been made available for use by multiple programmers and corporations. API keep the digital world connected. For example, Facebook's developer platform has over 10,000 apps that all use Facebook API [31].

API, on the other hand, offer significant danger if not utilized safely. An unsecured API exposes a database to malicious mining by cyber-criminals. The service providers give APIs to software developers in order for them to create interfaces

that allow them to communicate with the system services.[13] Another layer on top of the foundation adds to the security complexity, making it easier for security flaws to penetrate the system. Clear-text verification or content transmission, unsuitable approvals, unidentified access, reusable login information, or text categorization can all obstruct services and customer access, limit monitoring and logging capabilities, create unknown services, and API roles and responsibilities, and eventually lead to service repudiation and denial. [37]

1.2.3 Insider Threats, Unauthorized Access

Threats posed by trusted insiders are just as significant in the system as they are on-premise. Insiders can be current or former workers, contractors, or a trusted business partner, anyone who does not have to breach a company's defenses in order to get access to its systems. An insider does not need to be malevolent to cause harm; they might accidentally jeopardize data and systems. According to the 2018 Cost of Insider Threats study conducted by the Ponemon Institute, 64 percent of all reported insider events were the result of employee or contractor carelessness. This might involve mis-configured system servers, keeping sensitive data on a personal device, or being a victim of a phishing attack. [30]

1.2.4 Data privacy and confidentiality

Data protection is the process of preserving a company's data in any scenario, irrespective of where it is stored, whether it is at rest or in motion, and whether it is managed internally or by a third party. When storing private or confidential data on the storage server, users must maintain data confidentiality. To keep data private, authentication and access control mechanisms are utilized. Improving system dependability and trustworthiness might help with data confidentiality, authentication, and access control issues. [25] Customers do not trust service providers,

and it is incredibly difficult for storage providers to prevent any insider threats, therefore keeping sensitive data directly in the system is extremely dangerous. Simple encryption suffers from the key management issue and is unable of supporting complicated requirements such as inquiry, simultaneous change, and fine-grained authorisation.[51]

1.3 Motivation & Research Objective

In the research we conducted to find the security issues in today's system, we found that there are so many failings in terms of maintaining the proper authorization, authentication, and data access control together. Some systems are providing only data security and some are ensuring proper authentication, but it is found that there are no such models which will be secure enough to provide all three of the concerns, which are authorization, authentication, and data access control, that we need to provide altogether to make a fully secure system.

In this paper, we are implementing a system security model where this system

ensures access control, authorization, and authentication together. In this way, this stored data will be secured by encryption and access will be controlled by authorities. Specific data will be accessed by the specific authority fixed by the administrator. It will ensure data security and prevent any kind of unauthorized access, as well as prevent insider threats. The objectives of this research are:

- Security threats and issues.
- Prevent data breaches.
- Reduce unauthorized access.
- Develop a full secure control panel for organizations.
- Give authorized access control and management.
- Prevent insider threats.
- Secure data privacy and confidentiality.
- Use IAM, IDAAS, AAAS, MFA, TFA and SAML together to protect the data in the system architecture
- Secure data access and authentication with KSU Key-exchange algorithm for authorization.
- Ensure full system security, including data encryption, identity access and management, and storage security.

Chapter 2

Literature Review

According to different Web Services [19], one of the world's largest security service providers, multifactor authentication is a great method for safeguarding system accessibility because it adds an extra layer of security to the traditional username and password authentication. They implemented techniques that allow users to employ knowledge (username and password) factors as the first level of authentication and then supplement it with possession factors, such as having the user produce an authentication code from an MFA-capable device. This is true for AWS accounts, as well as individual accounts on Identity Access Manager (IAM). This is applicable for different accounts, individual Identity Access Manager (IAM) accounts, and service APIs. One-time passwords, hardware, virtual MFA-enabled devices, and SMS authentication are all alternatives to MFA. A user's account settings and resources are more secure as a result of these several factors [23]. Given the industry buzz, it's a positive move for security services providers to embrace MFA, but it's clear that the majority of the attention is on possession considerations. The only issue with this is that if a user loses their MFA-enabled device, their credentials are in danger.

The main drawback is that if a user loses their MFA-enabled device, their credentials are at risk of being stolen. There isn't a single inherence factor accessible for use among the service providers. This necessitates evaluating and analyzing the inclusion of inherence factors (what a user is) in order to improve security and solve the issue of potential loss of possession elements. It would also be important to figure out why companies haven't included biometrics in MFA, for example. One of the major threats to security is that data owners have no control over their own data after it is stored on the server. Therefore, there is a need to protect data in untrusted object environments. Every system does not guarantee such security factors as confidentiality, integrity, identification, etc. [43], [61]. Different cryptographic techniques can be used to protect data [9], [55]. Authentication plays an important role in protecting against unauthorized access to stored data. This is the first step toward information security.

2.1 Related Work

The authentication method described by Jing. [50] is based on the SAML protocol. This author investigated the security difficulties with bidirectional authentication and proposed a solution that includes a link between the certificate authority and the certificate authority's challenge-response. The session key is distributed to service providers and users. It offers some methods for ensuring the session's protection and security, as well as resolving some issues with data transfer. It is possible to avoid replay assaults in resource transformation by using consistent identity resource utilization, in detailed. [48] have created a methodology for establishing a safer identity management (IAM) system in security. It was accomplished by combining multiple authentication mechanism technologies into IAM. It displays the validation of user identities as well as the concealment of the user's actual identities. Various security measures can be provided in identity management systems to accomplish system security. Wang [53] proposed an approach for ensuring the security of users, services, and information access in a system. In a manual authentication strategy, there is a risk that third-party providers will misuse login information. If we adopt a manual identity management solution, there is a possibility that third-party vendors would use user credentials to launch malicious attacks on the server. As an outcome, a single-sign-on (SSO) approach is implemented, allowing users to use their one-time password (OTP) to access apps and resources at any time. It demonstrated how the OpenID, OAuth, and SAML protocols are utilized to facilitate authentication [35].Jiang et al. [49] examined and identified a set of security difficulties, identity threats, and limits in the environment, with an emphasis on identity and access management and security services. This study examines various protocols and their regularly used mechanisms from various perspectives. We can see that these protocols are only used to hide the identities of end-users communicating over the system, not for the information that is carried over the system, by analyzing numerous criteria such as Infrastructure as a service (IaaS) [40], [28].

Existing techniques are unable to concentrate on authentication validity as data flows across the service environment [58], [46], and [54]. To provide security for identities, information, and data flow, the proposed architecture employs the SAML protocol, robust authentication servers, IDS servers, and Penta Tope-based Elliptic Curve Cryptography [59]. To ensure the security of identities, information, and data flow, the proposed system employs the SAML protocol, strong authentication servers, IDS servers, and Penta Tope-based Elliptic Curve Cryptography [7] to send messages in an encrypted format. to secure the present system's encrypted data similarity search algorithms [57], which cyber attackers can successfully breach utilizing known cipher assaults. Similarly, this section covers a variety of alternative ways for security performance monitoring, multiple apps, and service platforms. Patel Y et al. [52] advocate the implementation of a multi-level authentication system to secure the network from illegal access and the implementation of security steps to prevent user data from being stored in any storage system. It provides security services in a system context. It consists of two schemas, Site 1 and Site 2. Site 1 consists of an application server and a database server, as well as a web interface via which users can submit or retrieve data from any safe system.

Banyal R K et al. [19] created a multi-factor authentication framework for various service providers, and a valid ticket was issued for the authorized service user by the ticket database. This framework includes an authentication mechanism that may operate with standard authentication techniques. A system validates this structure by authenticating the user based on a variety of factors including login, passwords, and captcha. Pippal S et al. [17] found a novel solution, a shared database structure method, that enables a bigger number of renters (enterprises) per database server owing to the database single feature, which satisfies database needs from many groups in different places. Authentication and authorization are critical prerequisites. Due to client architecture, some traditional authentication methods are addressed. Authentication ensures that the data presented reflects a request to be authorized by a particular object [11], [36]. The majority of web-based systems and services use a basic login strategy to achieve identification and authentication needs. There are various methods to individually authenticate the person [21] [12]

2.2 Methodology

The purpose of the proposed model is to secure systems storage, data, and access management with IAM, IDAAS, AAAS, MFA, SAML, and AES Encryption. This system follows the IAM, or Identity and Access Management, framework to ensure data access control so that critical information is only accessed by the people it is authorized to. The system makes sure that all the data of the users is protected and prevents unwanted data access. Here, the system service will work as IDaaS, or Identity as a Service. This service allows its users to manage their identities in a secure authentication and authorization process. It will be built and hosted in the system for the user's security. In this system, Authentication as a Service ensures the authentication of the system. It makes sure the users are originally authenticated to get the data access. To provide security, it uses MFA, or Multi-Factor-Authentication, to ensure the AaaS.

In this system, multi-factor authentication is employed, and co-admins must submit two or more verification factors in order to access the data and design. The system employs the industry-standard SAML, or Security Assertion Markup Language, which enables providers to authenticate users and transfer an authentication token to another user in order for that user to get access. In this system, SAML will generate and pass two keys known as primary and secondary keys between two co-admins. After that, the key exchange verification will happen, and after successful key insertion, the co-admins will get access. The random keys will be 8 bit long, and a total of four keys will be generated to ensure the highest security of the system. In this way, data access, authorization, and authentication will be provided in this system.

2.2.1 IAM

Identity and Access Management (IAM) is a security discipline that enables the right individuals to access the right data at the right time and for the right reasons. IAM solves the mission-critical requirement of making proper resource access across increasingly varied technological settings.

IAM often consists of the following characteristics:

- Single Access Control [48] Interface: IAM solution provides a clear and consistent access control system for all secure service platforms. All services can utilize the same interface.
- Extra Security: Users can specify extra security for important applications.
- Resource-level Access Control: Business owners can define responsibilities and provide people access to resources at various granularity levels. [26]

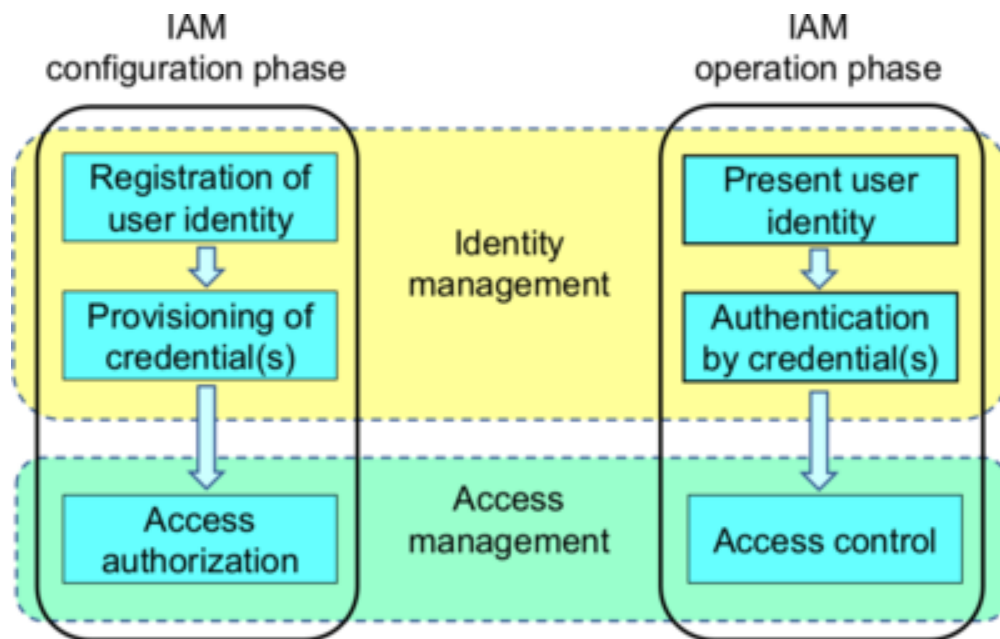


Figure 2.1: IAM-phases [48]

IAM Domain:

- Identification: Defines the process whereby a user, software, or device asserts its identity.
- Authentication: A user or application must submit a set of credentials, such as a password or an authentication token, to get access to a system.
- Authorization: Following authentication, the system determines if the user has permission to do the desired activity by comparing it to a stored access control matrix.
- Access Governance: The processes and procedures for requesting, approving, granting, managing, and auditing access.
- Accountability: The user or program gaining system access must be held responsible for the acts performed within the system.

There are more factors to consider for comprehensive identity and IAM coverage in the service. Management of Privileged Access (PAM) It explains the methods for

granting privileged access to information resources. This might imply granting system administrators access to the underlying operating systems deployed on virtual instances in protected systems.[14]

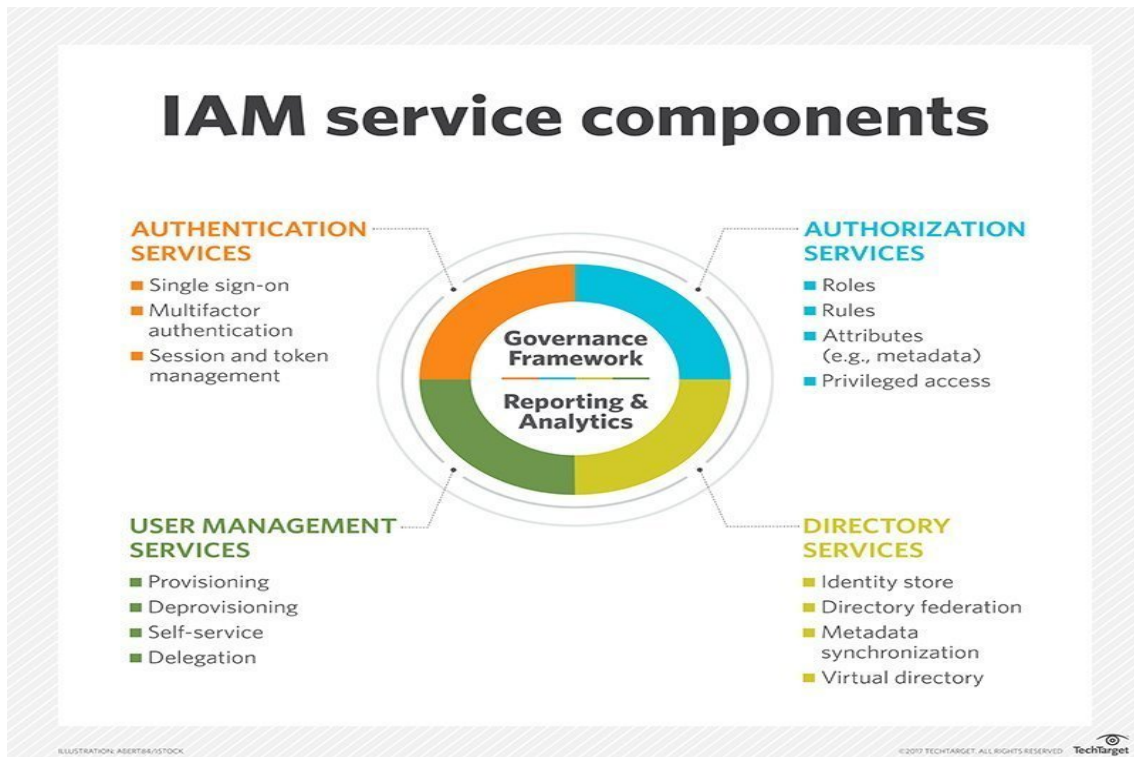


Figure 2.2: IAM Components [48]

Identity and Access Management (IAM) enables administrators to regulate who has access to certain resources. With an IAM architecture in place, IT managers can monitor user access to information in their businesses. The IAM solutions offer role-based access control, which allows system administrators to regulate access to systems or networks based on user roles inside the organization. The following should be done by IAM systems: capturing, recording user login data, managing the company database of user identities, and organizing access privileges assigning and deleting. This means that IAM systems should provide an overview and visibility of all aspects of the user base of the firm to a centralized directory service. In this model, it will be used to manage data access control for the service users. By using this, the company or organizer user can assign the authority to access a specific amount of data.

2.2.2 IDAAS

Identity as a Service (IDaaS) is a SaaS-based IAM product that enables companies to give safe access to an increasing number of software and SaaS services via single sign-on (SSO via SAML or OIDC), authentication, and access restrictions.

The core aspects of IDaaS are -

IGA: The two most critical components of IDaaS are the provisioning of users to securely access apps and password reset capabilities.

Access: User authentication, SSO, and authorization are all supported by federation standards such as SAML.

Intelligence: Monitoring and reviewing of identity access logs.

An identity service’s goal is to ensure that users are who they claim to be and to provide them with the appropriate forms of access to software applications, data, or other services at the appropriate time. If the infrastructure for this is created on-site, the company must determine what to do when an issue emerges. IDaaS can

be used for a variety of different purposes. One such application is adaptive multi-factor authentication. This is a feature that allows users to send several factors to access the network, boosting security over single-factor authentication, and access is allowed dynamically based on the risk of the user. Single sign-on is another application. This enables users to log into the network perimeter once and access any section of the company’s permitted constellation of applications and resources with a single effort. [61]

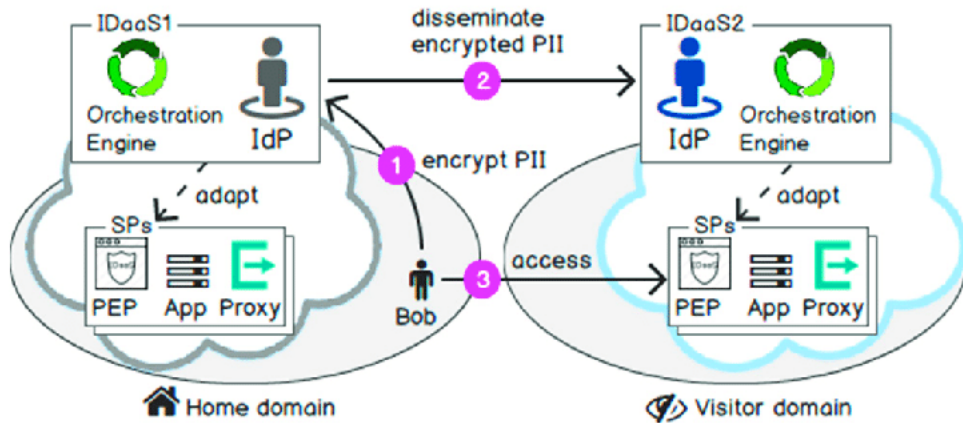


Figure 2.3: IDAAS as a model [39]

Identity as a service is authentication designed and managed by a third-party supplier to provide users with appropriate access to appropriate resources. In this layer 1 security paradigm, system designers wish to leverage this service in conjunction with IAM to allow experts to manage data access control. IDAAS improves cybersecurity while saving time through faster login and fewer password resets. The user will feel secure with this service regarding the authorization issue. It will assign an identity here in order to access the data.

2.2.3 AAAS

Users can access the UIDAAAaaS engine as a standalone service. This mechanism consists of three algorithms: APG, AKG, and Auth V. The user first requests the new user page and enters their credentials. Users initially request access to various services. Authentication consists of two phases. The first step is to create a new user, and the second is to update an existing user’s information. APG generates a password for the user and sends this to the user’s registered email, which is also

stored on a remote server, once the user enters their login credentials. AKG generates an Auth_key server and keeps it in an encrypted format on the server. When new users access their email accounts, they can see their username and password. The user inputs the user ID and password from the login page. It uses the authentication key generation technique to generate a user Auth key. AKG creates 4,444 Auth key users during the login procedure. Users can access services if their Auth key matches the server's Auth key; otherwise, their request is refused. UIDAaaS Algorithms under Development: The UIDAaaS (User Identity-based Authentication as a Service) algorithm suite consists of three algorithms:

- Authentication Password Generation (APG)
- Authentication Key Generation (AKG)
- Authentication Verification (Auth V).

The APG and AKG algorithms are proposed for use in the verification process for the system's storage as a key generation method that is also available as a service. APG delivers the password to the registered email address of the system user. CU can access any CSP service with the aid of an id and password. Following authentication, CU may modify the passwords to his liking. The suggested verification algorithm is made available as a separate Authentication as a Service (AaaS) from a CSP.

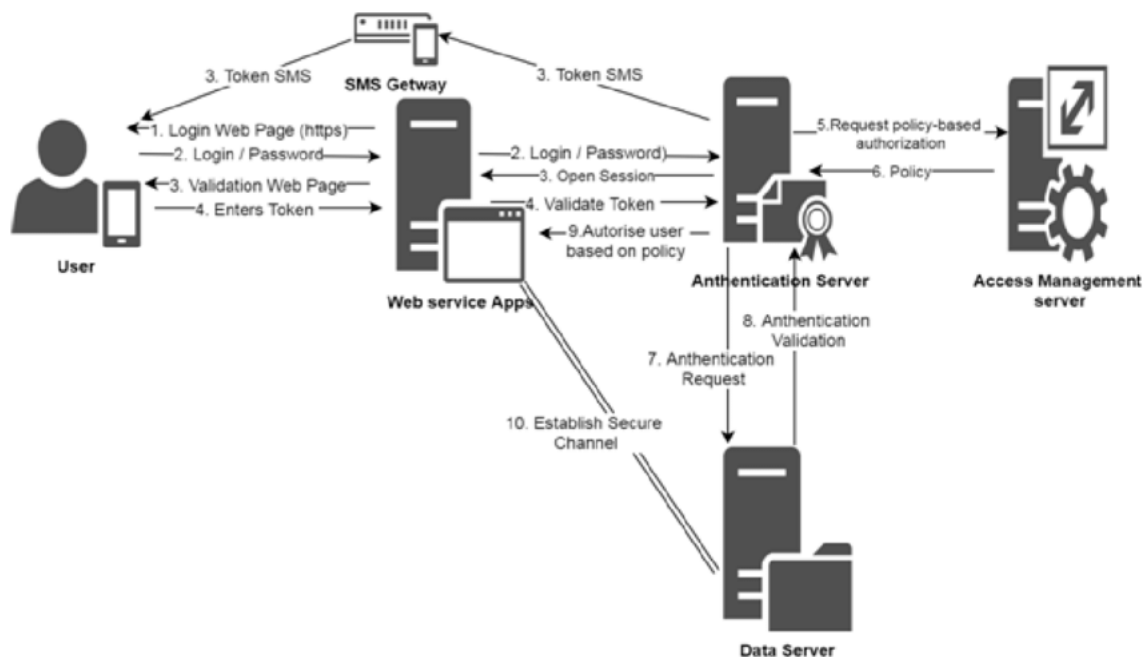


Figure 2.4: AAAS Model [39]

Authentication as a service allows corporations to simply implement multi-factor authentication to safeguard access to any service, from any device, from anywhere. This service is used in this tier to securely authenticate an organization's personnel through multi-factor authentication (MFA). Authentication as a service uses multi-factor authentication (MFA) for login security, which combines two or more authentication aspects for identity verification. It might be as follows:

recognizable, like a one-time pass-code (OTP) or the answer to a secret question owned by users, such as a smartphone the users have something unique about them, such as a fingerprint or face scan,

To accomplish multi-factor authentication, the authentication method must include at least two distinct technologies from at least two distinct technology groups. As a consequence, although using a PIN in conjunction with a password is not considered MFA, using a PIN in conjunction with face recognition as a second factor is. It is also permissible to use more than two authentication methods. However, most people desire frictionless authentication, which is the ability to be confirmed without having to go through additional security processes.[16]

Authentication as a service also supports cutting-edge technologies such as FIDO open standards, biometrics (including facial recognition and fingerprint scans), out-of-band authentication (such as Cronto), QR-like codes, and next-generation hardware. So, this is the future of system security. [32]

2.2.4 MFA

MFA is a type of authentication that requires a user to provide two or more verification factors in order to get access to a resource.

MFA's primary benefit is that it strengthens the security of any company by forcing users to identify themselves using more than simply a login and password. While usernames and passwords are essential, they are subject to brute force assaults and are easily stolen by third parties. Enforcing the usage of an MFA element, such as a fingerprint or physical hardware key, boosts your organization's confidence in its capacity to defend itself from fraudsters.

MFA functions by demanding more verification data (factors). One-time passwords (OTP) are one of the most common MFA challenges that users confront. OTP are four to eight-digit codes delivered to customers by email, SMS, or a variety of mobile apps. On a regular basis or when an authentication request is submitted, OTP generate a new code. The code is largely dependent on a seed price given to the user when they initially register, as well as a few extra components, which may be as simple as an incremental counter or a time value.

The majority of MFA authentication techniques use one of three types of additional information:

- Personal information, such as a password or PIN.
- Possession of an item, such as tablet or smartphone
- Inherent, such as fingerprint recognition or voice recognition

When it comes to authentication factors, identity as a service (IDaaS) solutions like One-Login offer many more MFA authentication choices and can interface with apps outside of the Microsoft ecosystem more simply. [8]

MFA is a kind of authentication in which the user is required to provide two or more verification factors in order to get access to a resource such as an application, an online account, or a VPN. MFA is a necessary component of a successful identity and access management (IAM) policy. Here, in this model, users can choose between two-factor authentication and multi-factor authentication. Users' companies



Figure 2.5: MFA [24]

or organizations can choose the service as per their needs. If the company needs to secure its data access highly, we would suggest going for multi-factor authentication. If they do not need high security, then they can always go for two-factor authentication, which will help them to use less storage and pay less. Most companies nowadays want to store their data in storage. Among various pieces of data, there is financial data that needs more than two-factor authentication. So, in that case, the model would suggest they go for MFA, which requires them to use what they have, what they know, and what they are. This authentication service can be the best way to secure unauthorized access and prevent insider threats.

2.2.5 SAML

SAML stands for Security Assertion Markup Language. It is an open standard based on XML that allows two parties to exchange identity data: an identity provider (IDP) and a service provider (SP).

Identity Provider: Authenticates the user and transmits the user's identity and MFA to the service provider.

Service Provider: Trusts the identity provider and provides the requested user access to the resource.

SAML operates by sharing user information between the identity and service provider, such as logins, authentication statuses, IDs, and other important aspects. As a consequence, the authentication method is simplified and safe because the user only has to log in once with a single set of login credentials. As a result, when a user attempts to visit a site, the identity provider authenticates the user to the service provider, who then grants access to the user. SAML employs a claims-based authentication methodology. When a user tries to visit a website, the service provider seeks authentication from the identity provider. The identity provider's SAML assertion is then used by the service provider to allow the user access. Let's look at an example to see how the workflow works. [8]

The user runs their browser and navigates to the web application of the service provider, which authenticates with an identity provider.

- The web application responds with a SAML request.
- SAML inquiries are delivered to the identity provider via the browser.

- The identity provider parses the SAML request.
- The identity provider verifies the user’s identity by asking a login and password or another kind of verification. NOTE: If the user is already approved, this step will be skipped.
- The identity provider generates the SAML response, which is then transmitted to the user’s browser.
- The browser sends the generated SAML response to the service provider’s web application, which verifies it..
- If the verification is successful, the user has access to the online service.

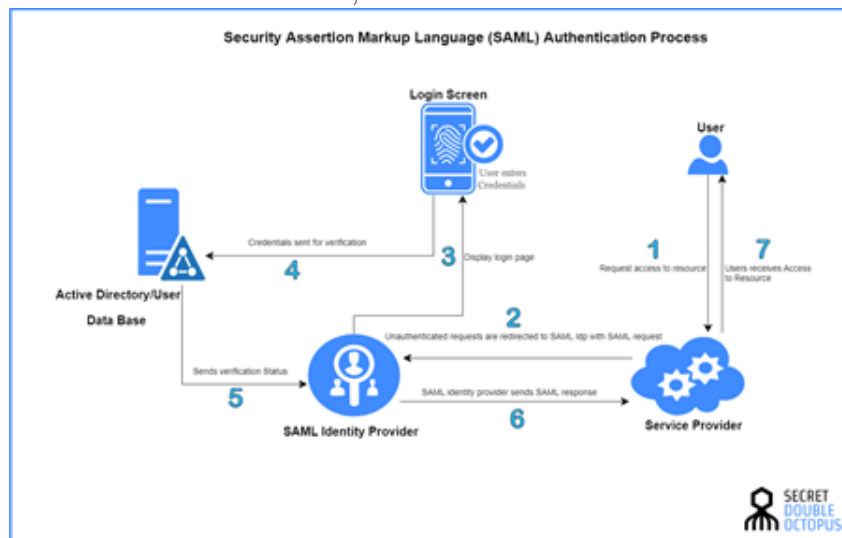


Figure 2.6: The SAML Authentication Process [4]

SAML is a free and open authentication protocol. Based on the Extensible Markup Language (XML) architecture, web applications use SAML to exchange authentication data between two parties: the identity provider (IdP) and the service provider (SP). It will offer further protection to the authentication at this layer. It will aid in the exchange of authentication credentials between IT partners. As a result, the authentication data flow will be safeguarded, and identity access management will be handled safely in the system.

2.2.6 AES Encryption

To secure sensitive information, the US government chose the Advanced Encryption Standard (AES) as a symmetric block cipher. The National Institute of Standards and Technology (NIST) highlighted the need for a replacement for the Data Encryption Standard (DES), which was becoming vulnerable to brute-force assaults, in 1997. Around the world, AES is used to secure critical data in software and hardware. It is crucial for federal computer security, cybersecurity, and data protection. [38]

In this model, we used AES, or The Advanced Encryption Standard, also known by its original name Rijndael, which is a US-developed technique for electronic

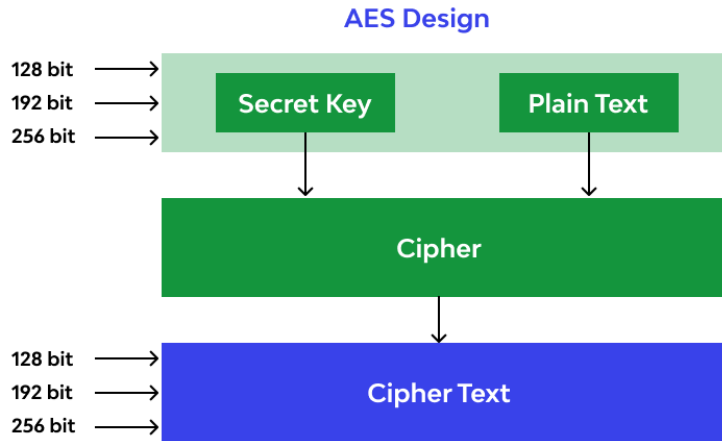


Figure 2.7: AES Design [45]

data encryption. We required one of the best data encryption solutions for data protection. Encrypting data before storing it ensures the confidentiality of the user's data. In this proposed model, the client's data will be encrypted by AES and then stored in storage. When an authorized person tries to access this data store, it will check the IAM and authentication and then decrypt the specific data authorized for that user. [10]

2.2.7 Key Exchange

In the key exchange method, two parties exchange their encrypted keys to get access to the data. This is one of the security protocols for ensuring a secure system environment. It also makes sure that unwanted persons can not get access to the data. This type of key exchange is made with a cryptography algorithm. [44]

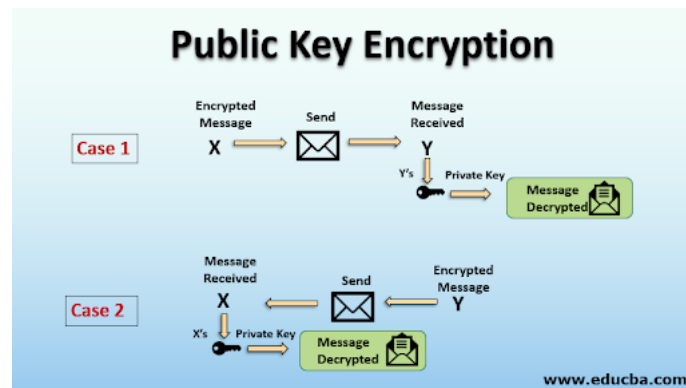


Figure 2.8: Key Exchange Between Two Parties [60]

If two parties want to exchange the keys between them, then only they can access the data. These keys must be encrypted. The message will be encrypted from the sender's side, and it will be decrypted again from the receiver's side. [5]

Chapter 3

System Requirements Specification

3.1 Functional Requirements

Every system or application needs to fulfill functional requirements, which include system requirements, user requirements, and business requirements. This will describe this system's behavior and its output.

- This system has an 8-bit One-Time-Password key-exchange verification system while logging in for both users and admins, which will be providing authorization, authentication, and data access control.
- While using this multi-layered security system in a time of key exchange, both primary and secondary keys for co-admin 1 and co-admin 2 will be provided via email.
- After verification, the task URL will be provided by email.
- The system also has a session time out system which will be active in hours. At that particular time, the user needs to finish its given task. Once the time ends, the URL will not be active anymore.
- In the event of an intruder attack, the system will be shut down instantly, and the attacker's laptop or computer's Mac address will be banned.

3.2 Non-Functional Requirements

To judge the operation of the system, explain some of the non-functional requirements below which fulfill the requirements of this system and provide the best services. By using non-functional requirements, customers can judge the system's operations through these constraints or requirements imposed on the system.

- **Portability:** It's easy to convert a system running on one platform to run on another.
- **Reliability:** The ability of a system to operate continuously and in a user-acceptable manner when operating inside the environment for which it was intended.

- **Availability:** The service should be accessible at all times via a search engine and should be restricted mostly by the server downtime upon which system is executing.
- **Maintainability:** A private service is used to administer the server, and authorization and authentication will be ensured by the system.
- **Security:** High access security in a layer-based system to ensure data security, authorization, and authentication.
- **User friendly:** The system should be simple to use.
- **Performance:** The performance should be quick.
- **Safety:** The system should be efficient enough that it does not hang when the supervisor, super admin, or co-admin runs it regularly.
- **Privacy:** The system's data should not be divulged to anyone.

This system is mainly focused on software or application-based components rather than device drivers or operating systems, as this system has a minimal amount of use for both device drivers or operating systems. The operating systems are as usual, like keyboards, monitors, disk drivers, etc.

3.3 Software Components

PHP: This system backend work is done by PHP. Using PHP, this system's super-admin admin and user dashboard create, task form create, task assign, primary and secondary key generate, key verification, request accept, task confirmation status update, task launch different work has been done.

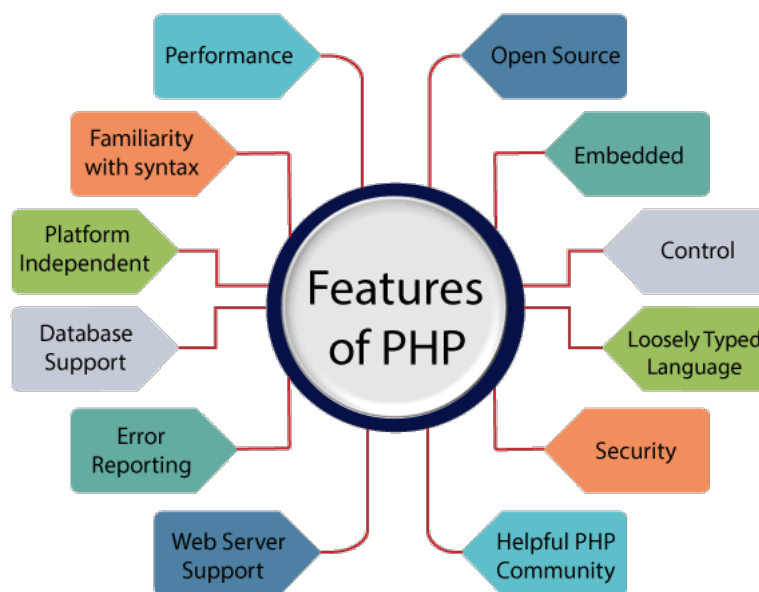


Figure 3.1: PHP [47]

Phpmyadmin: In these systems, there will be a need for three types of databases: super-admin, co-admin, and user. As this system is still a prototype, the attributes for super-admin and co-admin details are limited. While developing this system, the attributes will increase according to its needs. The user database table has a primary and secondary key generation option from where the system will check if the One-time-password has been cross-verified and then the super-admin can launch the task. Also, tasks can be updated and edited by the superadmin.



Figure 3.2: Phpmyadmin [18]

Diagram.net: For this system's design flowchart, workflow diagrams, in short, all types of diagrams have been drawn by the draw.net online software tool.



Figure 3.3: Diagram.net [1]

Laravel: This system's framework is done by Laravel as it follows a model-view-controller design pattern. Along with this, Laravel uses an MVC architecture system and CMS to reduce the complexity of this system's website.



Figure 3.4: Laravel [41]

CSS: Cascading Style Sheets, or CSS, were used to style and layout the web pages of this system's website. It can be used to change the size, spacing, color, and font of material, as well as add ornamental elements and different sorts of animation, or split content into columns.



Figure 3.5: CSS 3 [22]

HTML: For providing tasks URL links through email, to create different attributes, structure section, tags, Hyper Text Markup Language (HTML) is used in this system.



Figure 3.6: HTML [3]

JavaScript: Javascript has been used for different complex features in this system, like cross key verification and a hash function to send email as a confirmation.



Figure 3.7: Javascript [6]

XAMPP: As this system is a prototype, it's been hosted on a local server through Xampp as a test base before releasing this system to the main server.



Figure 3.8: XAMPP [56]

3.4 Organizational Security Impact

This system will prevent consumer mistrust. Day by day, this generation is fully converting to servers where the system stores all the data. There are lots of providers who sell or misuse the client's data. As a result, clients or consumers mistrust the providers. But the system providers have zero tolerance for any kind of data breach. The system providers fully believe in privacy no matter what type of data the system has. It has seen many different cases where providers neglect the smaller privacy issues, which leads to bigger losses for the organization. The system providers will always be aware of smaller ones, even if the vulnerabilities are minor ones.

Sometimes black hat hackers try to break the system to get money from the clients, which will be a great loss and a psychological toll on the organization. This system ensures that this type of case will never happen by using the primary and secondary 8-bit OTP key exchange systems. While upgrading this system along with the time and requirements, there will never be a security gap from the system provider's side.

3.5 Ethical Consideration

No Internal Data Breaches- No internal data breaches from system providers.

Reliability: In these systems, there is MFA and TFA for authentication or authorization so that the customers can fully rely on the system.

Privacy and Confidentiality: While using this system, any organization's sensitive data and information will be stored with full confidentiality so that no third party can get access to those data.

Piracy and sabotage: No data or information will ever be shared with any other rival organizations or competitors.

Liability: The system providers will never guarantee anything that cannot be done by the system. System providers will only take money for the services that are provided to the system clients. There will be no hidden costs for anything.

3.6 Risk Management & Safety

As a system provider, it will make sure that there will be no third-party involvement while monitoring and controlling the data. Sometimes there can be server issues for which the workflow can be hampered and data can be lost. To prevent these types of issues, this system will ensure a backup server so that there will be a proper flow of work with no data loss.

In this system, there will be multiple bio-metric authentication systems so that if any person finds any problem with one bio-metric system, they can use another one while logging in. To ensure proper time management, this system will be using a time countdown system for session login. There will be one-way communication between the primary and secondary servers, so there will be no chance for trade information to be leaked from this system.

3.7 Economical Impact

This system is a layer-based system, so clients can easily customize and take the security services as per their requirements and as per what they pay. So, there will be flexibility in the economic sector. This system offers a customized and layer-based

security system, so from this point of view, it will be software that may require some extra expense to provide the clients with the best service. Clients of this system will handle any economic transaction or any data regarding the economy. It will be kept as very sensitive data, which can be accessed by only the super admin.

3.8 Economic Feasibility

Financial analysis includes a method and an attempt to ensure that it is repeatable and likely to be completed. This feasibility basically evaluates the new process performance, evaluating the current system’s expenses and revenues as well as the proposed service’s advantages, and if it would exceed its lifetime costs. In other words, it must be an accurate cost-benefit analysis before any action is taken. To calculate the total cost of ownership, costs must be estimated for:

Total Estimated Cost For the Project:

Operation cost:

Server	10k- 12k Per Month
Storage	100k Lease
Network Connections	10k Yearly basis
Employee Cost	4 Developers - per month 50k, 1 Project Manager-per month 40k, 2 System Designer - per month 50k, 1 Security Specialist- per month 60k, 1 Network Specialist- per month 50k, 1 Documents Writer - per month 30k.
Infrastructure costs	60k-80k Per month

Table 3.1: Operation Estimated Cost Project

Hardware Cost:

Hardware Specification	500k - 600k
Hardware-Maintenance	100k Yearly basis
New or retired hardware costs (along with future hardware)	100k Yearly basis

Table 3.2: Estimated Hardware Cost Project

Software Cost:

License Cost	60k Yearly basis
Purchase Cost	100k Yearly basis

Table 3.3: Estimated Software Cost Project

Maintenance Cost:

Application Update Cost	50k per month
System Update Cost (software version update and purchase)	100k Per month

Table 3.4: Estimated Maintenance Cost of the Project

Chapter 4

KSU Key Exchange Algorithm

4.1 Introducing the KSU Key Exchange Algorithm

KSU is an algorithm for key exchange algorithms. Using this algorithm in a system, two parties can complete the authentication process with four keys consisting of eight bits each. Through this protocol, 8 bits of a primary key and 8 bits of a secondary key will be generated for each system admin, and each admin will provide their primary and another admin's secondary key for the authentication process. After checking this key exchange process, admins can proceed to the system. This key exchange algorithm ensures the authentication and security of the system. The security of the KSU algorithm is mainly based on the four randomly generated secret keys and the key exchange verification between two parties.

KSU Key Exchange Algorithm:

Let,

Ka= primary key for Co-admin 1

Kb= secondary key for Co-admin 1

Kx= primary key for Co-admin 2

Ky= secondary key for Co-admin 2

After requesting access from Co-admin 1 2

The system will generate → for Co-admin 1 → Ka and Kb keys

The system will generate → for Co-admin 2 → Kx and Ky keys

The System sends the email → Co-admin 1 → Ka and Ky keys

The System sends the email → Co-admin 2 → Kx and Kb keys

Both Co-admins will sign in with the keys they have given in their emails.

System Generated Keys		System Key Exchange Verification	
Admin 1	Admin 2	Admin 1	Admin 2
Ka (8 bits)	Kx (8 bits)	Ka (8 bits)	Kx (8 bits)
Kb (8 bits)	Ky (8 bits)	Ky(8 bits)	Kb (8 bits)
16 bits	16 bits	Total 32 bits	

Table 4.1: KSU Key Exchange Algorithm

The system will verify → Co-admin 2 → Kx and Kb keys

The system will complete the verification process and Co-admin will proceed to the system.

$K_a = 8$ bits
 $K_b = 8$ bits
 $K_x = 8$ bits
 $K_y = 8$ bits
 $\text{Co-admin1} = K_a + K_y = (8+8)$ bits = 16 bits
 $\text{Co-admin2} = K_x + K_b = (8+8)$ bits = 16 bits
 Total: 16 bits + 16 bits = 32 bits.
 It will take $2^{32} = 4,294,967,296$ attempts to break these keys.

4.2 KSU Key Exchange Algorithm Diagram & Analysis



Figure 4.1: KSU Key-Exchange Algorithm Diagram

In this diagram above, a few steps of the process of the KSU key exchange algorithm are shown. This process includes the request being sent into the system and then

requesting processing with This requires the generation of two separate keys of 16 bits each for two actors (Co-Admin) for separate requests. Each key will have two parts (Ka, Kb Kx, Ky). There will be an exchange of each key's 2nd part with other co-admins, and co-admins will enter the 2 parts of keys (for Co-admin 1 Ka Ky, and for Co-admin 2 Kx Kb). Then there will be a verification check between these four parts of the two keys after they are entered with the generated keys stored in the system's database. After the confirmation of the KSU key exchange process, there will be a button called "Publish" in this system's super-admin interface that will be active, and the super-admin will publish the task. This can also be the accessing process of some top secure facilities and data centers, or for any sort of highly secure and authorized place.

4.3 KSU Key Exchange Algorithm Cyclomatic Complexity

This system's complexity is calculated by cyclomatic complexity. As it can be used as a quality matrix, used to measure the minimum effort with the best area concentration for testing , it can guide the testing process. [2] It is easy to implement.

$$V(G) = E - N + 2 \text{ [2]}$$

where,

E = the number of edges in the control flow graph.

N = the number of nodes in the control flow graph.

P = the number of connected components.

Edge=98

Nodes=98

p=2

So, cyclomatic complexity $V(G) = E - N + 2P = 98 - 98 + 2 * 2 = 4$

Flow chart:

In the flowchart for code of key-exchange verification , login authentication and authorization are given below:

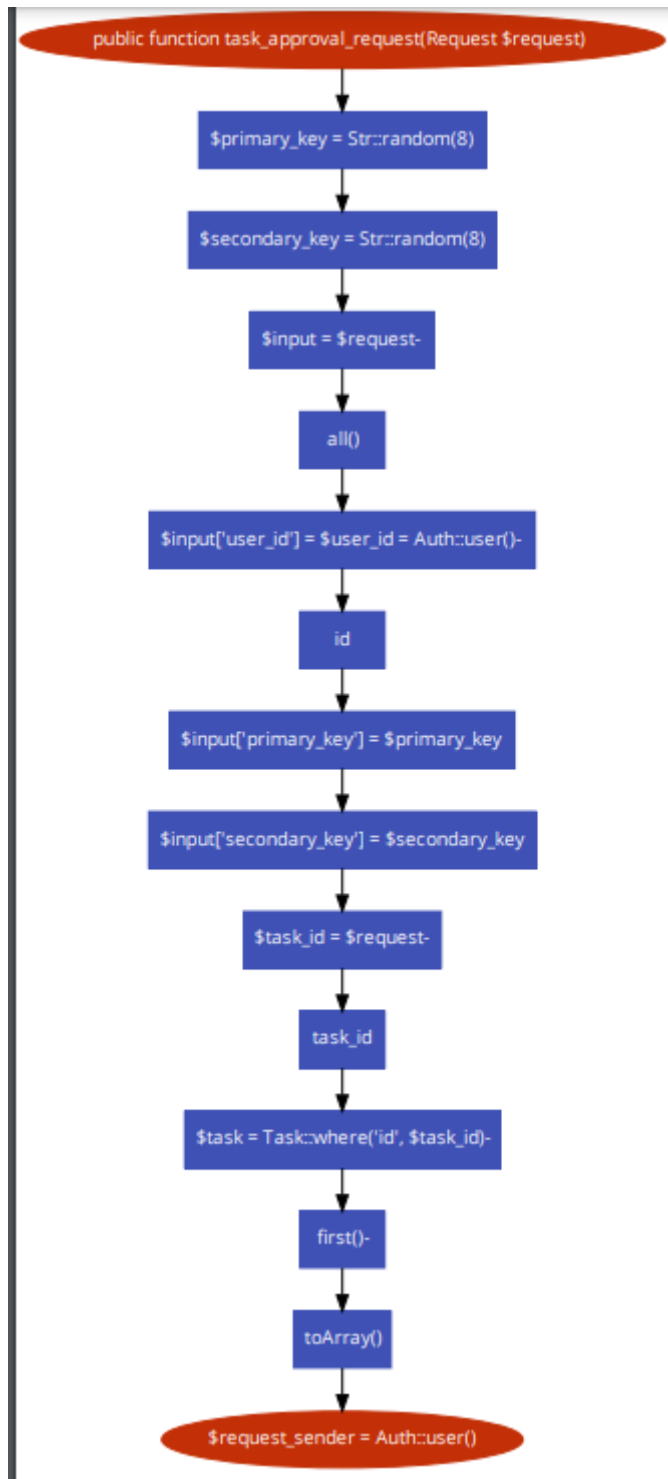


Figure 4.2: Flowchart-Initialization

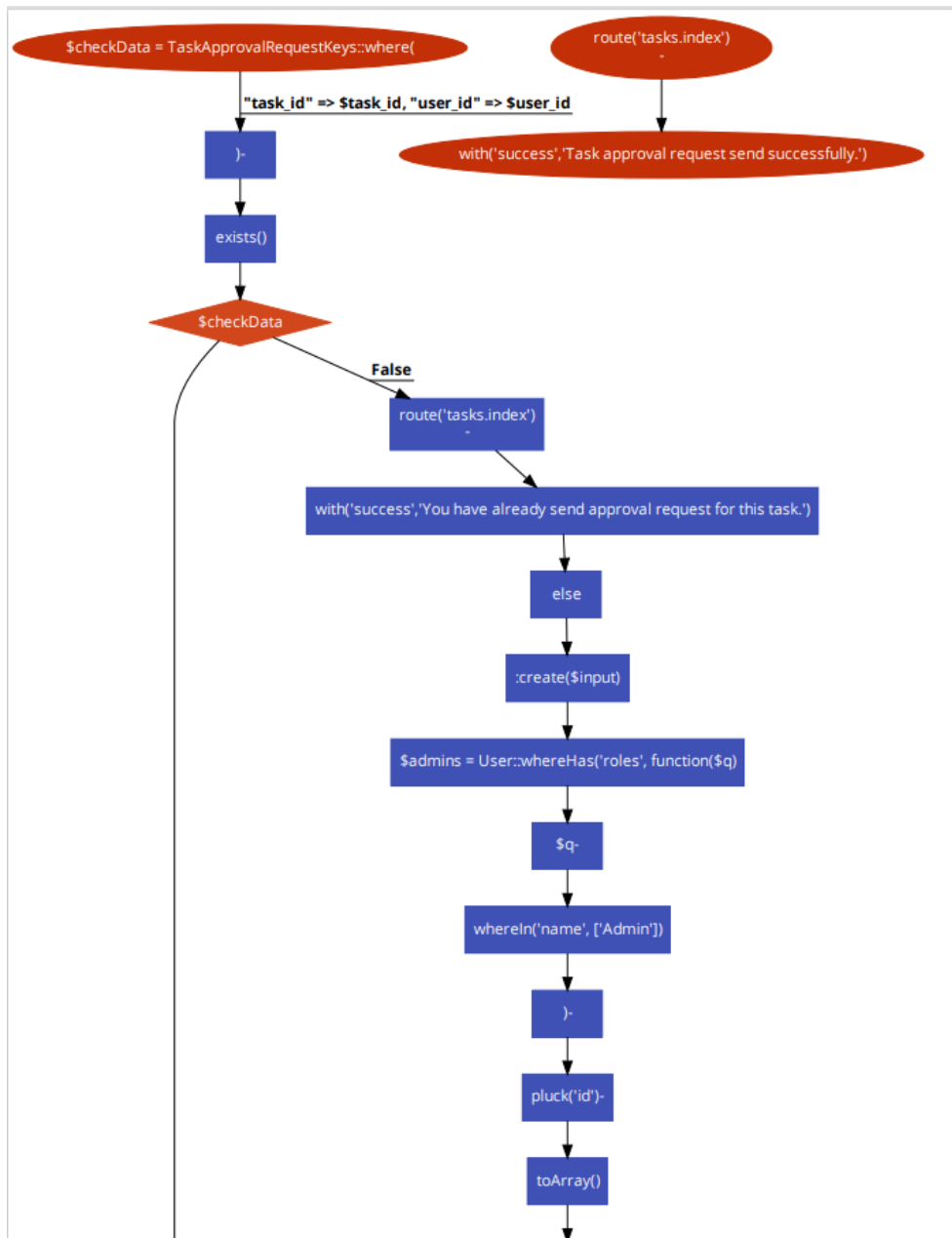


Figure 4.3: Flowchart-check data and request update part 1

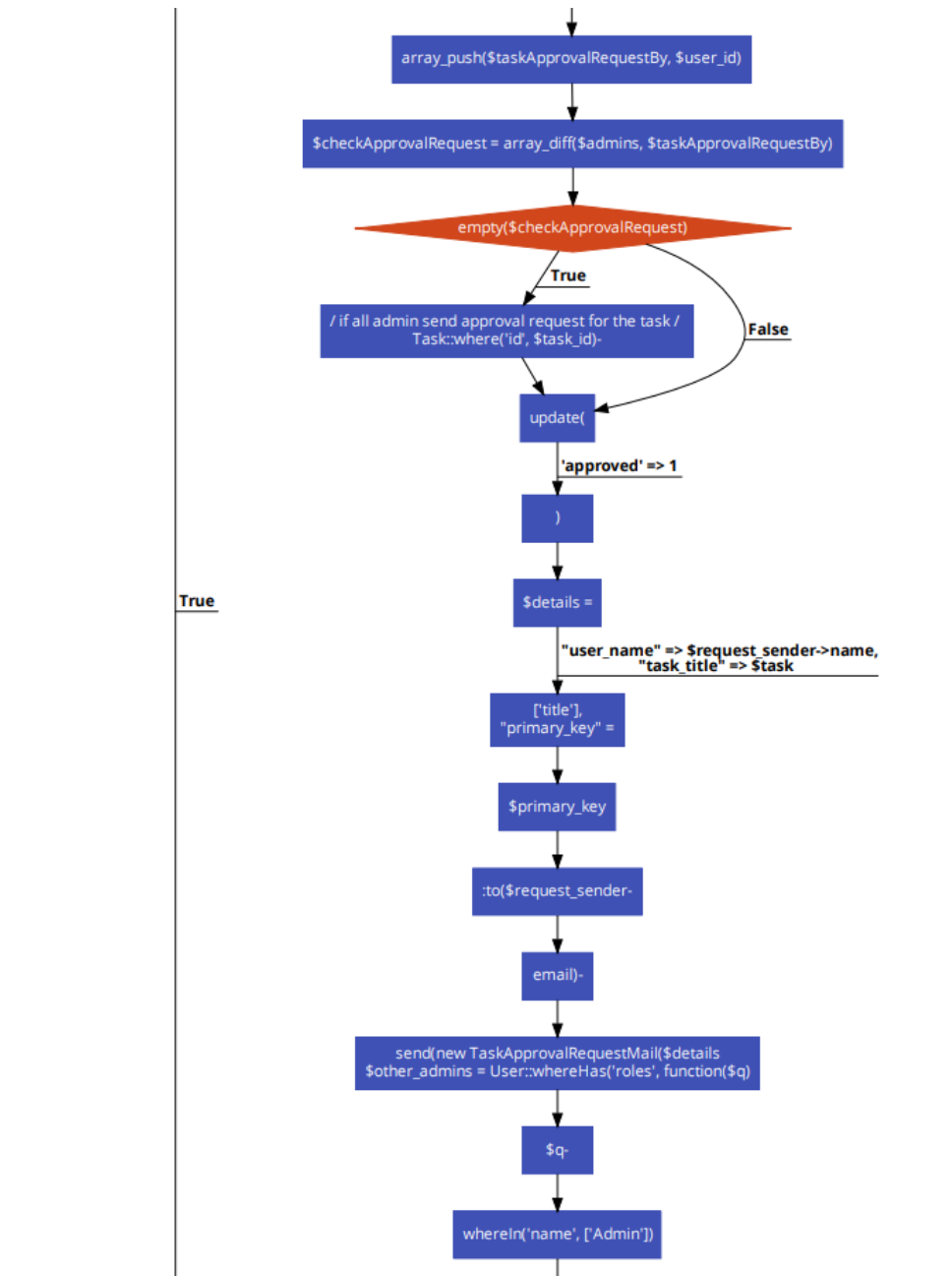


Figure 4.4: Flowchart-check data and request update part 2

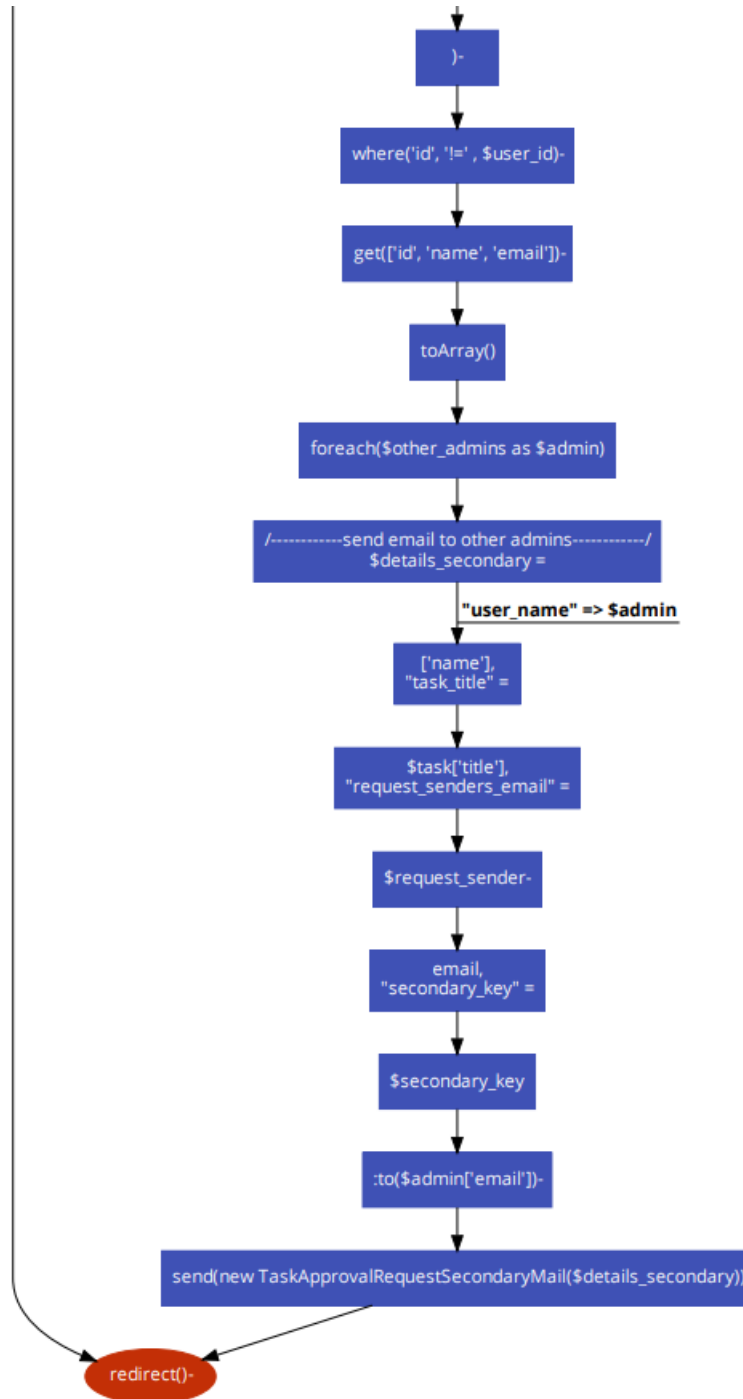


Figure 4.5: Flowchart-flowchart-check data and request update part 3

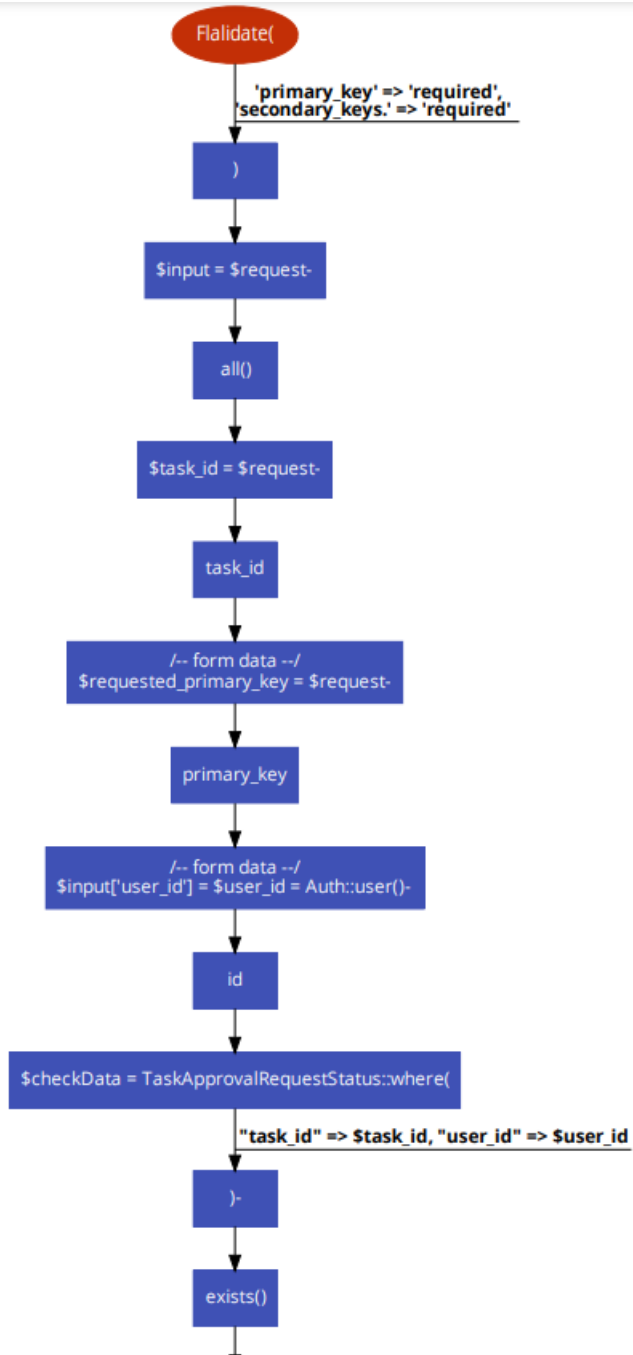


Figure 4.6: Flowchart-check key part 1

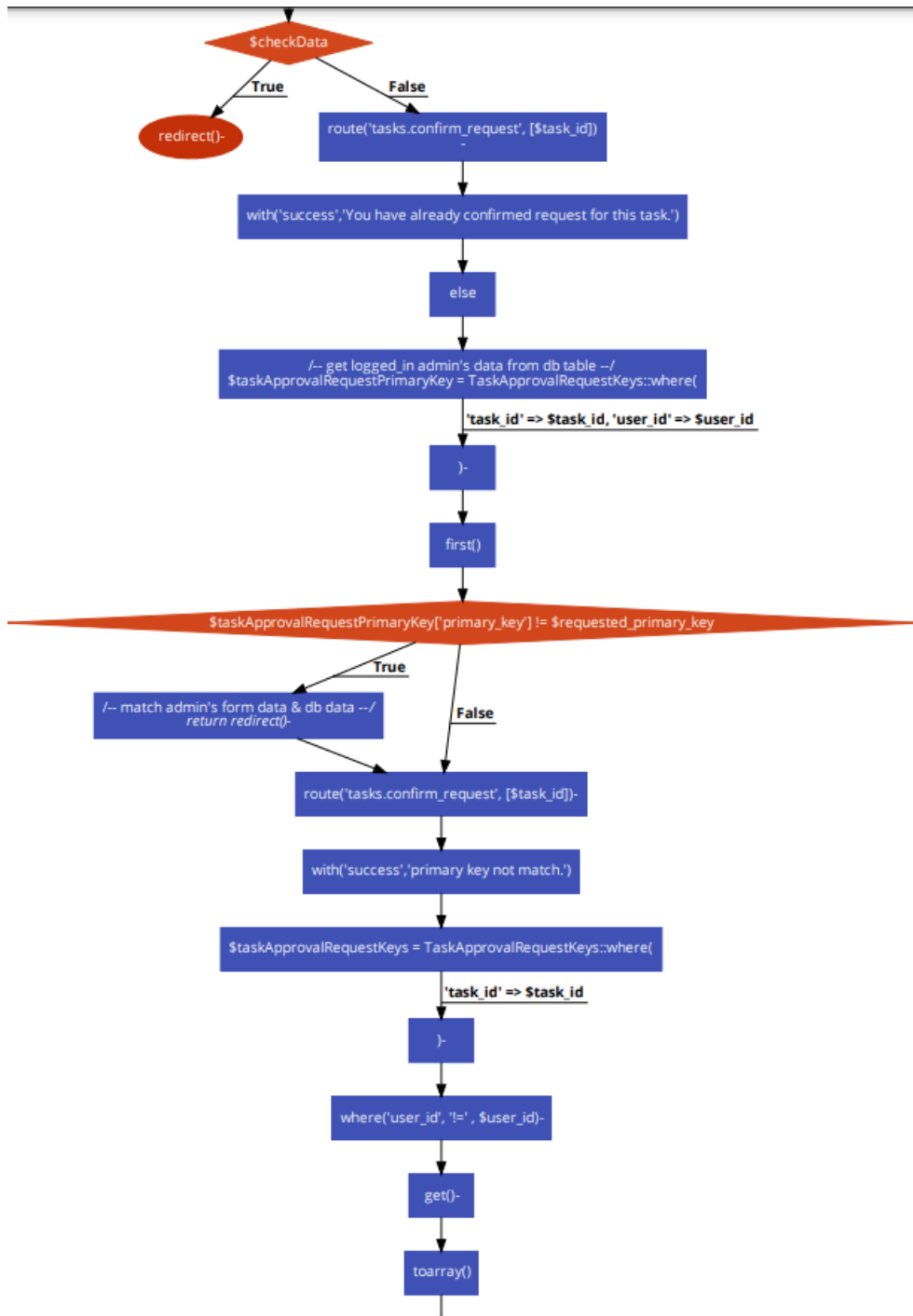


Figure 4.7: Flowchart-check key part 2

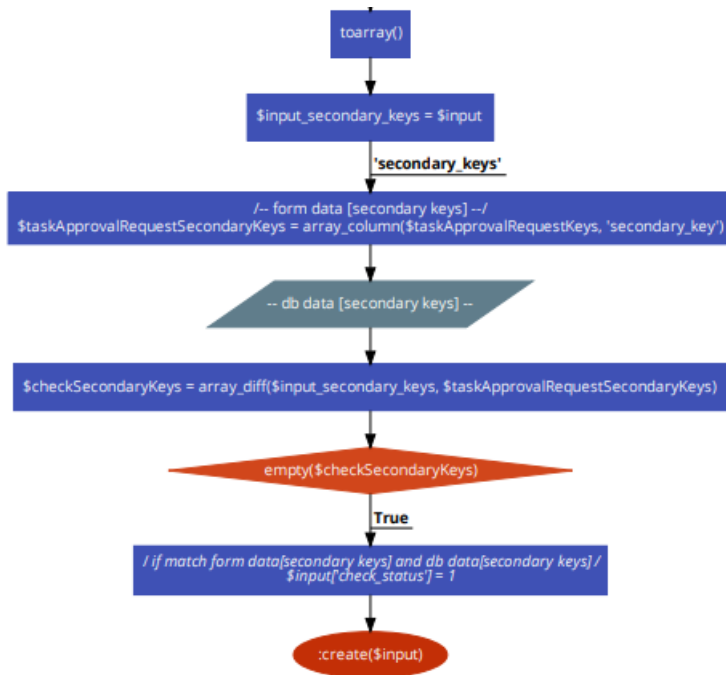


Figure 4.8: Flowchart-flowchart-check key part 3

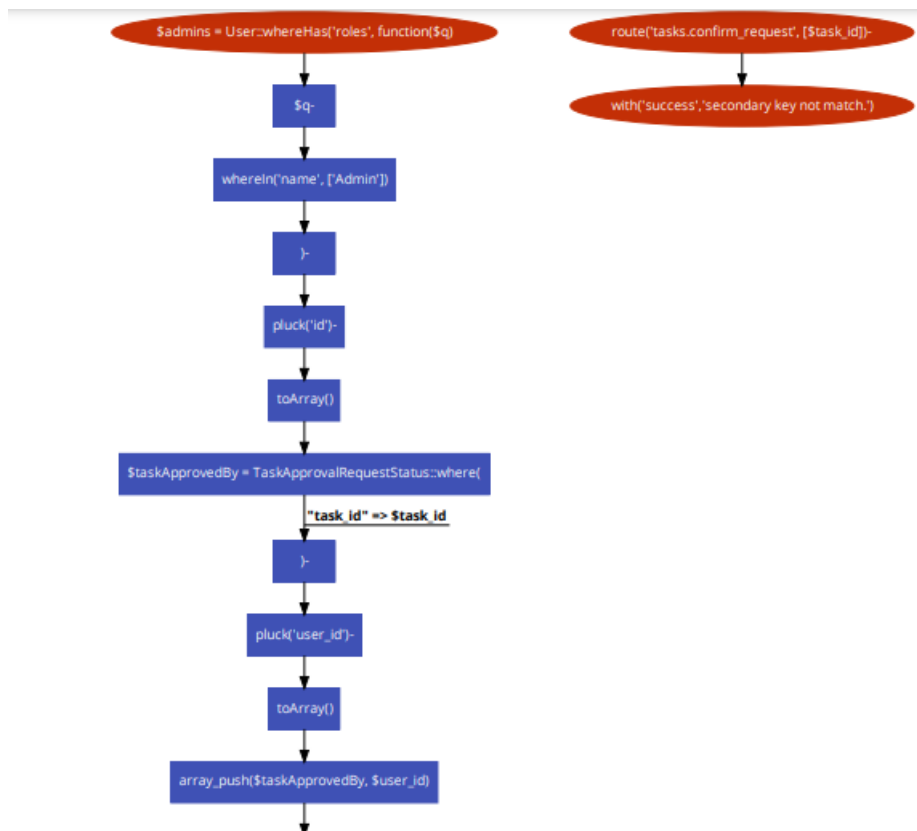


Figure 4.9: Flowchart-update table part 1

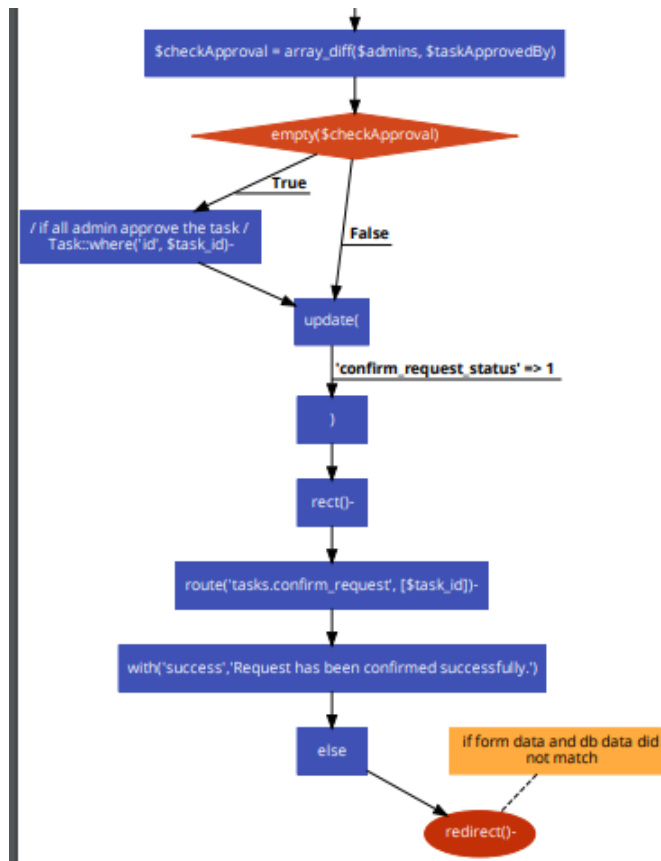


Figure 4.10: Flowchart-flowchart-update table part 2

4.4 Comparison with other key exchange algorithm

The top two key exchange algorithms are RSA and the Diffie-Hellman Algorithms. Both of them have some limitations. The proposed and implemented key exchange algorithm, KSU, is more secure than the RSA and Diffie-Hellman Algorithms. How KSU key exchange is a unique algorithm and why it is more secure than RSA and Diffie-Hellman are given below.

- The RSA system requires a third party to verify the reliability of the public key, but in KSU, no third party is involved as both primary and secondary keys are generated randomly by the system.[29]
- In RSA, high processing is required for the decryption on the end user's side but in KSU, no high processing is required.[29]
- Unlike RSA, in KSU the keys are generated from the system side so it doesn't slow the data transfer rate even if the data is large. [29]
- Diffie-Hellman's one of the biggest limitations is the lack of authentication process. KSU ensures highly-secure authentication. [27]
- As there is a lack of authentication process in Diffie-Hellman there can be man-in-the-middle attacks as a result the system can be vulnerable. In KSU there

is no possibility of a man-in-the-middle attack as both primary and secondary keys are randomly generated per session by the system and these keys are hidden with a hash function in the system database.

- KSU is inexpensive in terms of resources and CPU Performance time compared to Diffie-Hellman as KSU is not computationally intensive.
- With KSU both authentication and encryption are possible as a result it will be one of the best and most secure key exchange algorithms.

4.5 Nobility of KSU Algorithm

KSU algorithm is a newly introduced algorithm in order to ensure data access and authentication security. In this key exchange algorithm the system will randomly generate four keys in every session. In every other key exchange algorithm there is either a public key or a private key and among them one key is fixed in many systems whereas in this system the keys will be generated randomly in every session. There is no fixed key. These four keys won't be used by only one person. It will be used parallel in two co-admins interfaces by them. Both of them will use their own primary key and another's secondary key for the verification. In this way if both of the co-admins can verify each other only then both of them can access the system. Even if one co-admin verifies himself and another can't verify himself, the system can not be accessed by any of them. So, for accessing the system both co-admins have to get verified with the keys they are provided from the system. For this reason, it can be said that this key is a newly introduced unique approach to ensure the secure authentication and data access control.

4.6 KSU Key Exchange Algorithm Security Against Common Attacks

This system is secure enough to defend against dictionary attacks or brute force attacks as-

- The system uses multi-factor authentication.
- This system will use biometrics in the future.
- The system will use a timer to countdown with a time limitation for any particular task for the authorized person
- In the future, this system will be more secure with a session timeout for each user.
- The system will generate random keys each time the admin/user tries to get access.
- Each key will consist of 8 bits, and there will be four randomly generated keys, meaning a total of 32 bits of keys are being used here for the key exchange verification.

- Normally, every new character of the key adds 95 possible letters, numbers, and special characters to find the right one. So, it will be tough to find the 32 bits key.
- It would take on average $2^{32}=4,294,967,296$ attempts to crack the key for each session. This system is more secure because each time the admin/user tries to get access, there will be newly generated keys that will never be used again.

This key exchange protocol will also defend against Men-in-the-Middle and Denial-of-Service attacks as this protocol will generate entirely new keys every time. So, these attacks can not be possible here. Also, this system will defend against most of the attacks such as Malware, Phishing, SQL Injection, Password attacks, Cross-site Scripting, and Internet of things attacks.

Chapter 5

System Analysis

5.1 System Requirements

To enable a system, there are always a number of requirements that will need to be present in the host of the system to run the system smoothly. This is a very important milestone for the proposed system to work. Under this scope of work, the system will be required to be analyzed in detail for functions, documents, actors, attributes, diagrams, hardware quality, performance, and configuration. It is also needed to check the software license, validity, quality, and other infrastructure of this relevant security system. This system will run on any browser and device with an internet connection. But it may vary from user to user. If this system is used for highly secure facilities, then it will need to have access to MFA checking devices and other hardware that may help the system to run smoothly in that particular system.

To control the system, there will be a few roles for actors whose roles will be fixed, and those actors will handle and maintain the system. Also, as this system will require email or SMS options, there is a need for configuration for those features. So that the user may be able to get the key to access the system. A key exchange algorithm with an AES data transmission system will also be needed, including a hash function to make it more secure. In the future, with more resources, this system could be the leading security system with quality development.

5.2 System Design

During this stage, the comprehensive functional range defining and developing for the developed system tasks is carried out in accordance with the conventional software strategy. This is a critical stage in the development of any system. Given the eventual development and deployment scope, the proposed system design should be sufficiently resilient, scalable, user-friendly, and inter-operable. At this system-

designing stage, there will be a showcasing of design-related tasks and various standard system designs of diagrams. At this stage, there will be the identification of modules by specifying technical diagrams such as use case diagrams, activity diagrams, sequence diagrams, data flow diagrams, class diagrams, and window navigation diagrams. For the convenience of the system, except for use cases, activity diagrams, and sequence diagrams, all other types of diagrams have been included

with the system's architecture, and later on, the system architecture with its analysis and detailed process will be presented. The USE CASE, workflow, and sequence diagrams will be shown here separately.

5.2.1 WorkFlow

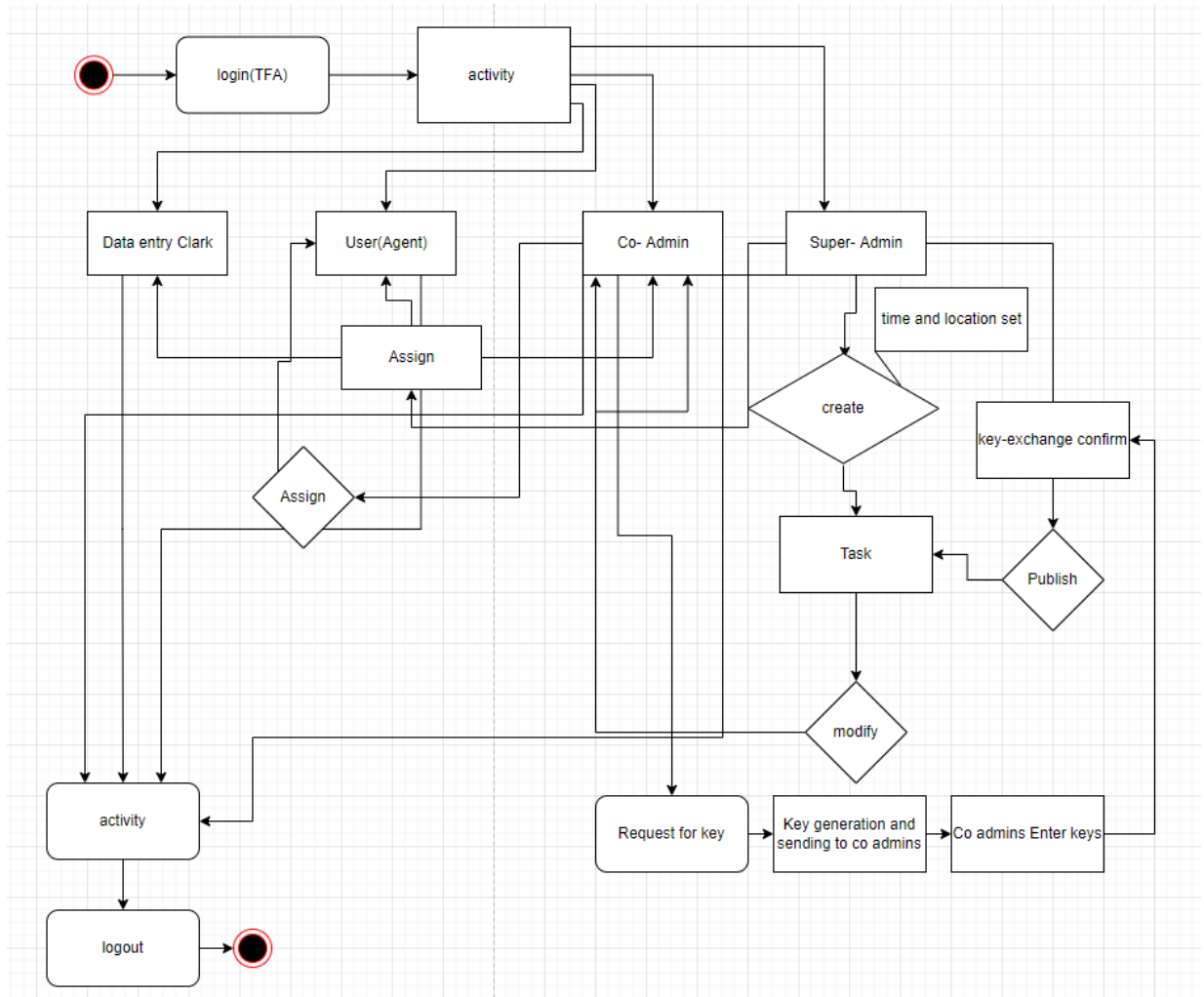


Figure 5.1: Workflow Diagram for the Security System

5.2.2 USE CASE

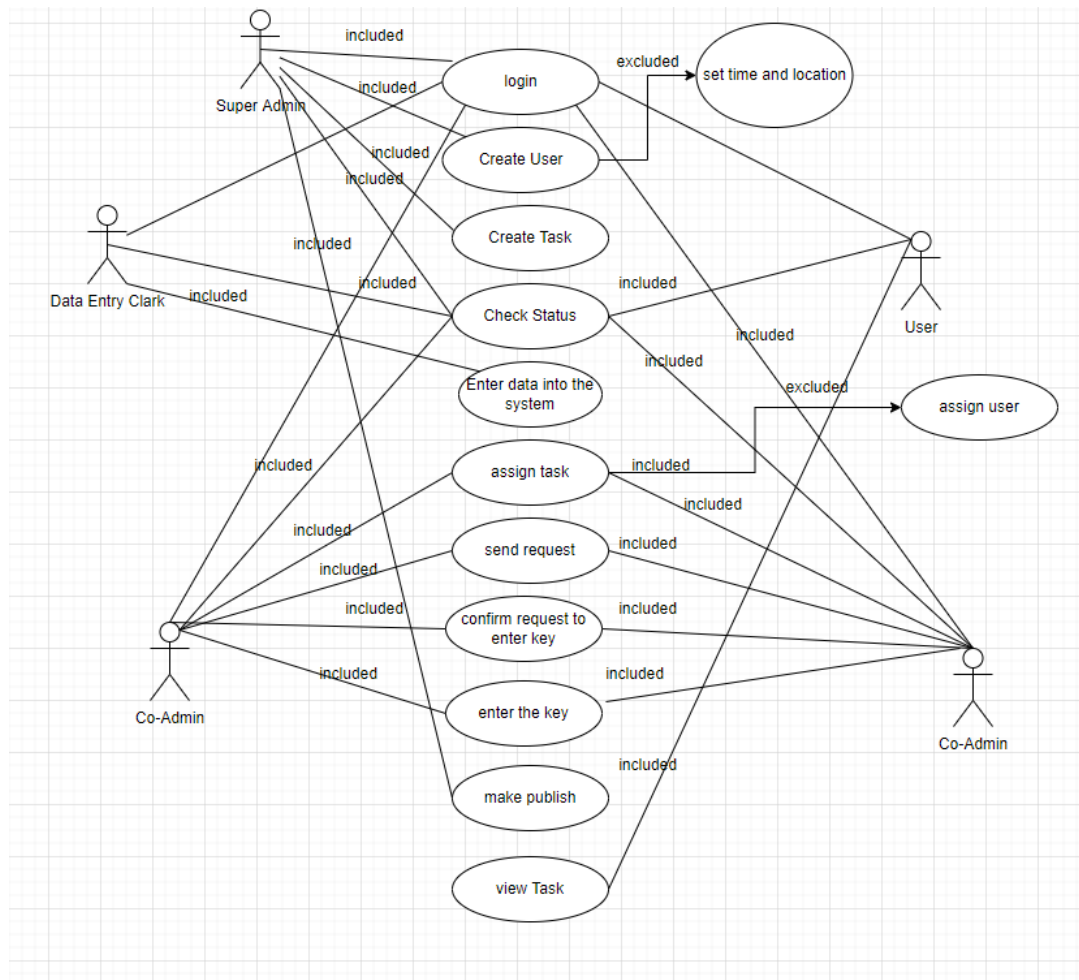


Figure 5.2: USE-CASE Diagram for the Security System

5.2.3 Sequence Diagram

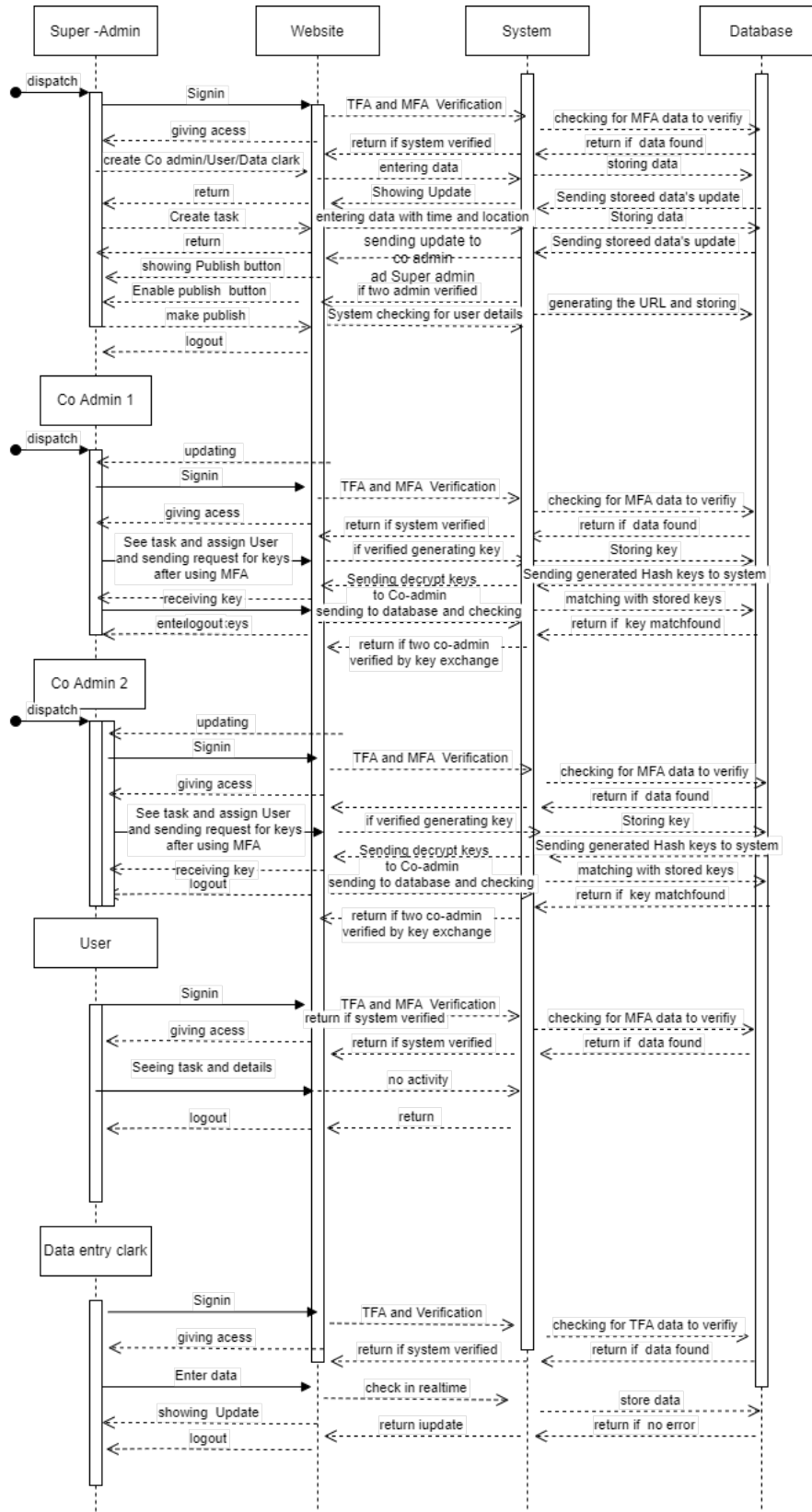


Figure 5.3: Sequence Diagram for the Security System

5.3 System Architecture

In every system implementation, it is very important to have a proper plan or architecture for the whole system to ensure the quality, security, and integrity of the system. Without the proper workflow, it is very difficult to maintain system development. Here, the details of the system's architecture will be shown with proper direction and workflow.

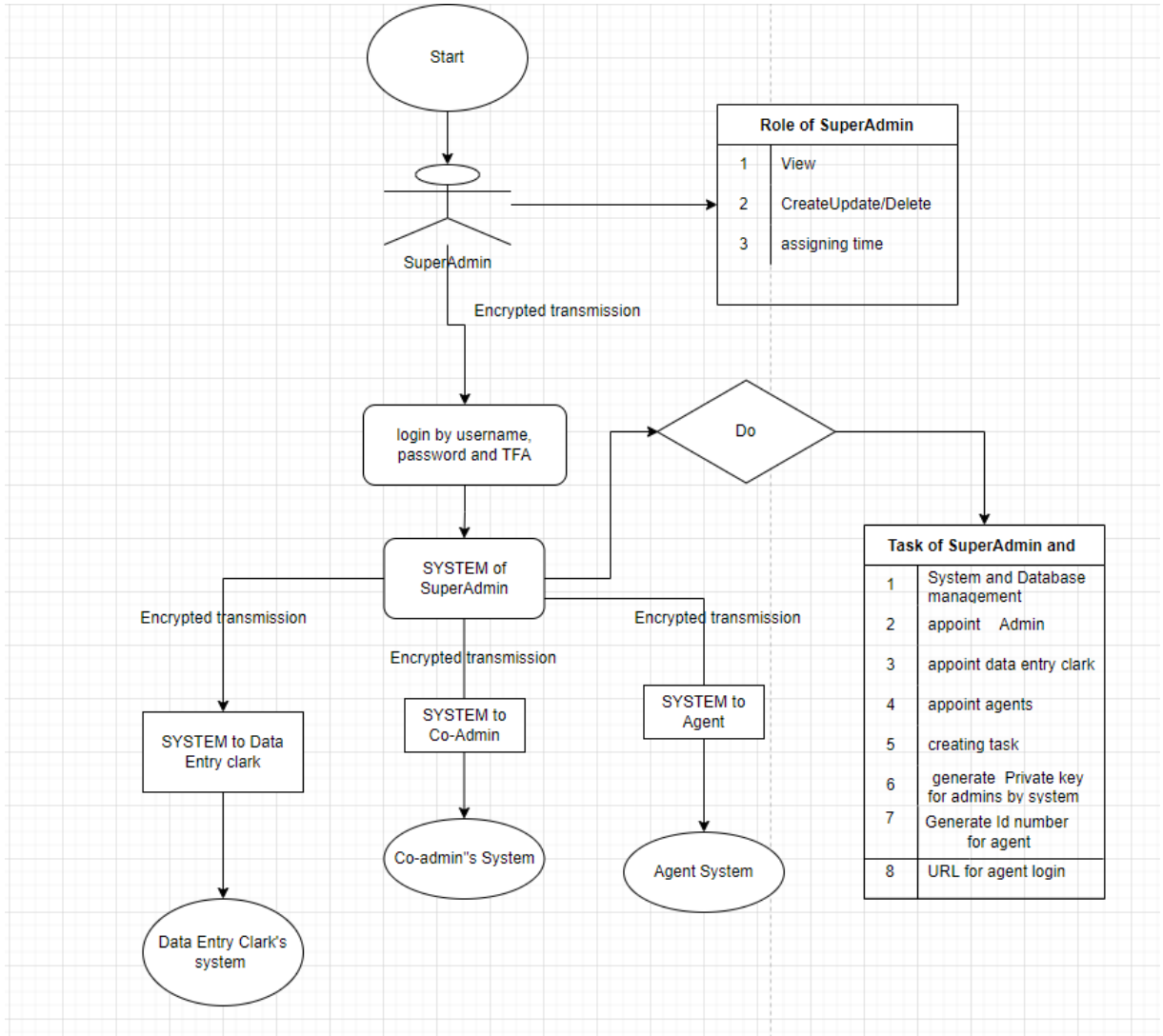


Figure 5.4: Part One (SuperAdmin)

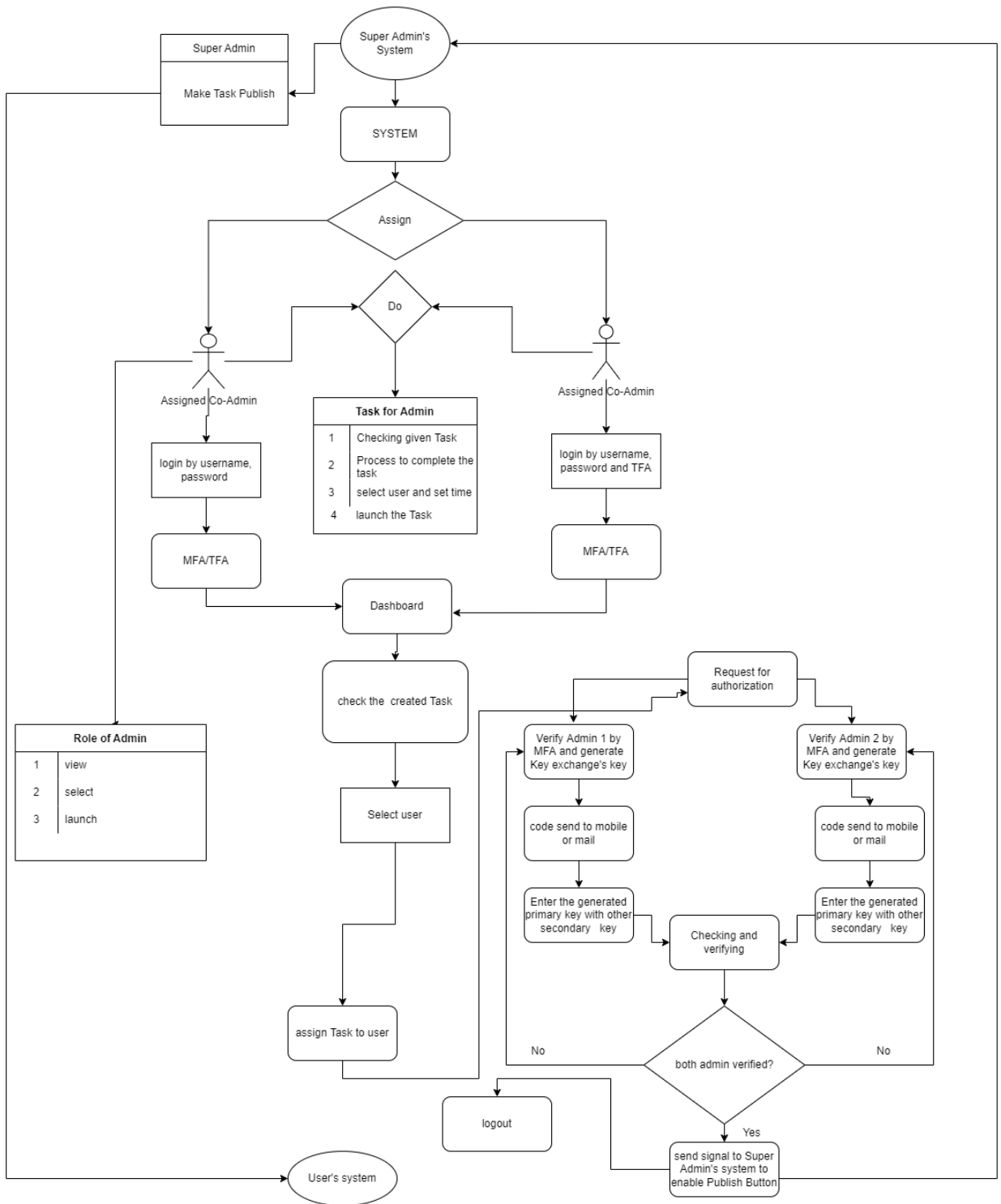


Figure 5.5: Part Two (Co-Admin)

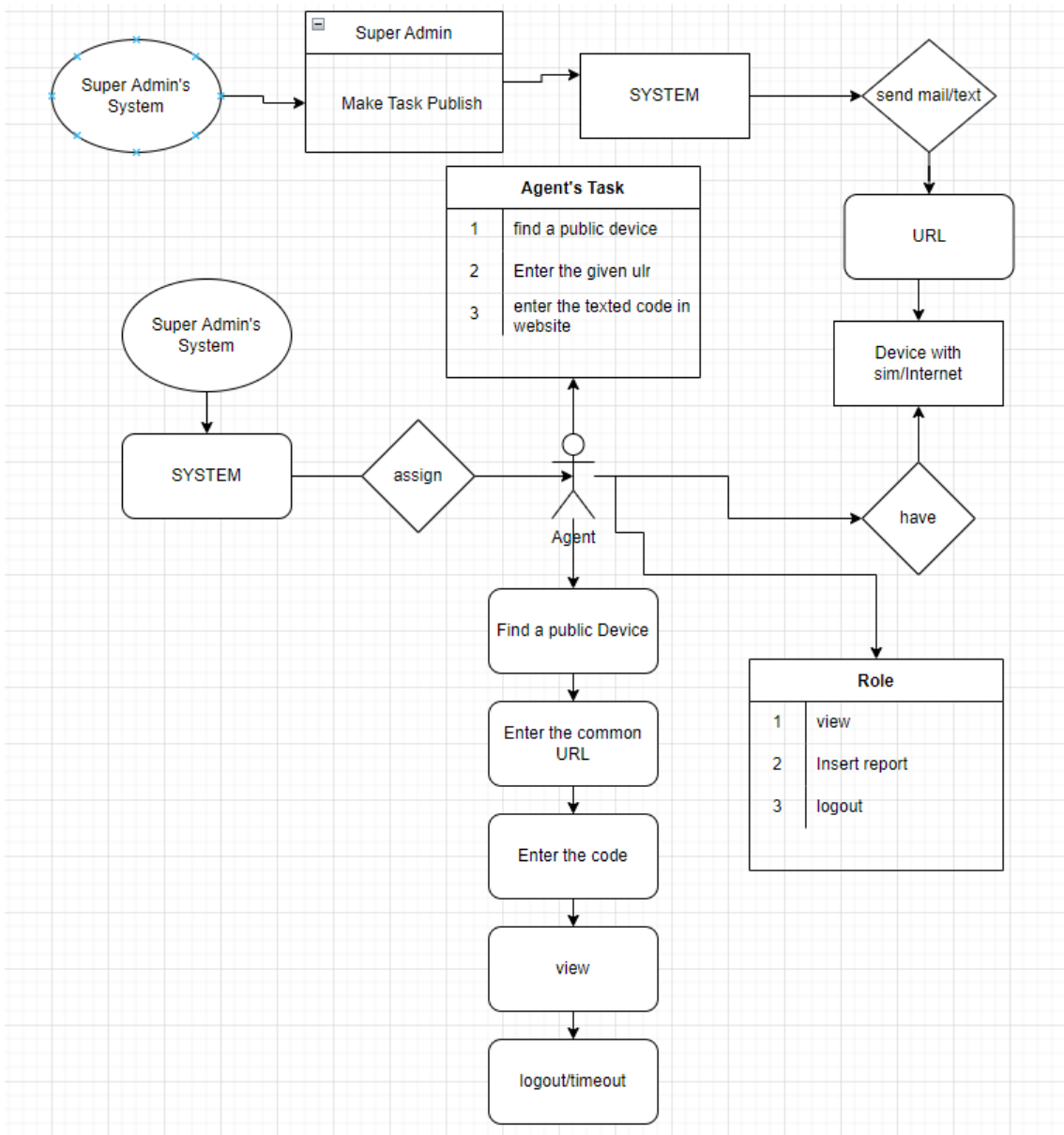


Figure 5.6: Part One Part Three (User)

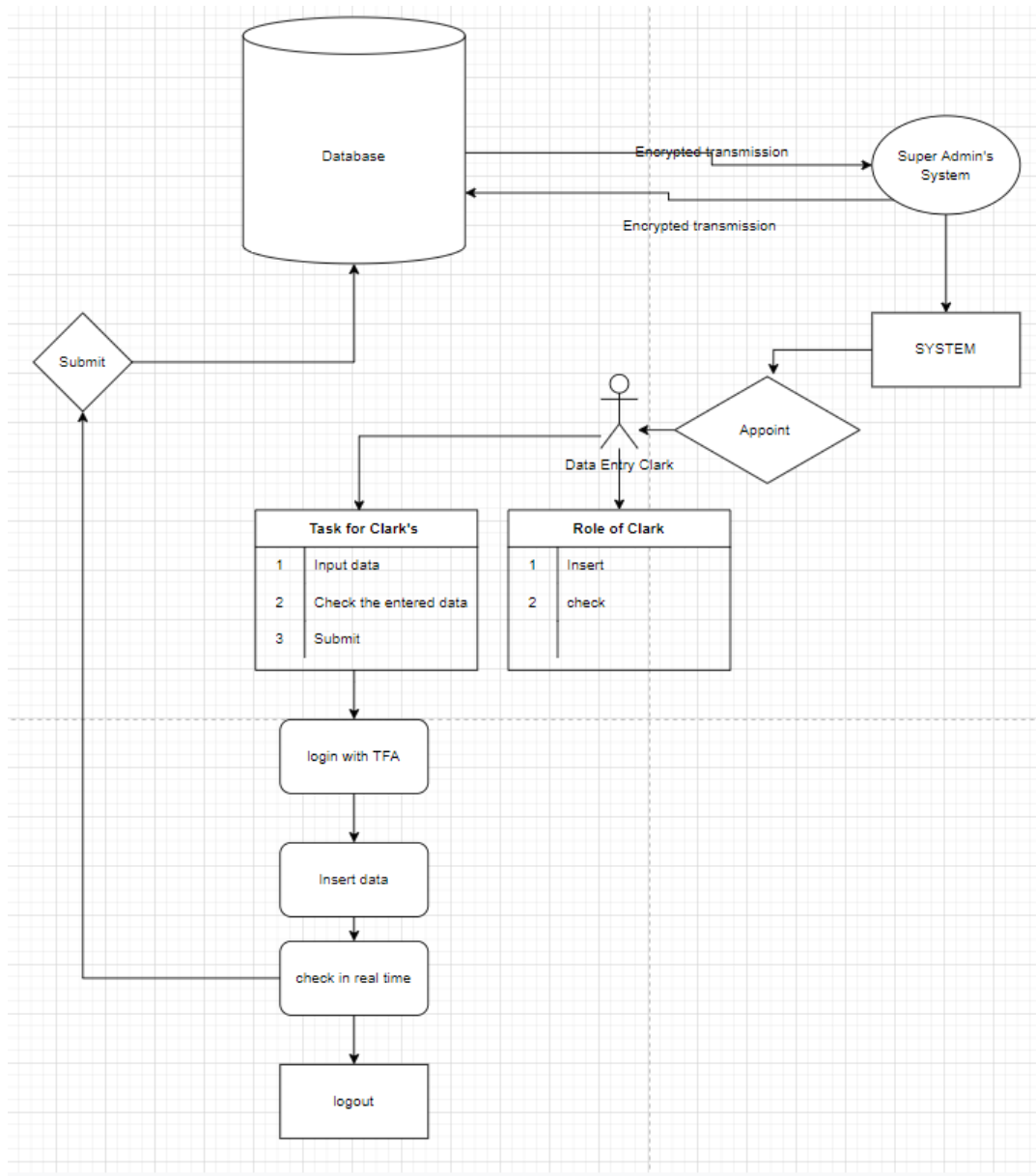


Figure 5.7: Part four (Data Entry Clerk, including the database)

These four initial parts will be joined and will act like one single system's architecture.

5.4 System Architecture Analysis

In system security, it is always a concern about authentication and authorization with the proper access method. Nowadays, people spend more money and time to secure their secret stuff from the rest of the world. That is why security systems are getting more complicated and step-based. In this paper, the proposed system is also a security system that is based on the data access control of users with the system of authentication and authorization. This system is a multi-layered security system. This system is a bit complex, but as there is a saying, the complex is the simple form for the person who practices it. That means if anyone uses this system on a daily basis, then it won't be a complex system for that user, but it will be a complex system for hackers or new users also.

5.4.1 Login with TFA or MFA

This multi-layered security system, this system includes Super-Admin's user id and password with Two-Factor-Authentication or Multi-Factor-Authentication. In this system's scenario, there will be a super admin shown in part one of the system architecture. The Super-Admin will sign in to the system by using the user id or email and password. During that process, the system will ask for TFA or MFA-related data access. If this is TFA, then the system will send an OTP to the user's mobile, or if this system is MFA-based, then the system will ask for the bio-metric data the user gave before and store it in the database. If the credentials match, then Super Admin will get access. By the same method of authentication, the two co-admins of this system and the user of the system will get access. For data entry clerks, there will be only TFA-based authentication with a login system.

5.4.2 Creating Roles and Actors

At this stage, after the super admin gets access to the system, the super admin will create roles such as Co-admins, user/agents, and Data-Entry-Clerk, and assign permission to perform limited tasks to their roles. After that, Super-Admin will create those roles' actors. The Super-Admin will create exactly two co-admins, a few data entry clerks, and n number of users for n number of places they want to operate with this system. The forms of these three actors will have their details, including username, email, name, and password. The users' form will have a location too.

5.4.3 Creating Task and controlling

At this point, Super-Admin will create tasks for the users with details of the task and the proper direction to do the task. The task will be location-based, and Super-Admin will assign the location. Along with that, he will also assign two types of time. One type of time will be for the user to see the task. For example, after 30 to 40 minutes, the window with the task details the user was seeing will be closed and the task will be automatically removed from the server or just from the user's interface. He will not be able to view the task details anymore. Another type of time will be set for finishing the task and reporting back. After creating the whole task, Super-Admin will submit it and the task will automatically pass to the two

co-admins. Then two co-admins will view the task and will discuss it. After that, they will assign a user, who is located where the task is assigned to do. The system will also give them an update regarding the schedule of that user, whether that user was assigned to a task before which is still incomplete or not. Then they will either assign a new user or just force assign the task to the user, which will be in the user's queue. After the assigning process, the co-admin will see a button "send a request for authentication." Then Co-Admins will be asked to give their bio-metric information, such as fingerprints, pulse, or anyone or more forms of the bio-metric verification system. If the system verifies the two co-admins with MFA, then the system will generate two keys. Each key will have two parts. This system will have a key exchange algorithm that has been mentioned before, namely the KSU key-exchange algorithm. After the key exchange algorithm process, co-admins will receive emails with two parts of keys according to the key exchange process. Then co-admin will click the button "confirm request" and a window will popup in co-admins' interface where they will insert the received keys. If anyone gives any part of the key wrong, then the system will not go further and, after a limited attempt, they will not be able to verify anymore. But if they insert the correct keys and the system verifies them, then there is a button in the super-admin's interface called "Make Publish." This button will be enabled and the super admin will publish the task. Then there will be a one time generated URL of the system's website which will be sent to the user's email. After clicking the link, the user will be redirected to a login panel.

5.4.4 Task viewing by the user

After logging into the system, the user will view the task list and check the newly created task. After he clicks the link, the countdown of the total time to completion will begin, and after he clicks the task view button, the countdown of the task details visibility will begin. After seeing the task, the user will logout and the task will be gone. There will be just the option to report back regarding the task. And after the total time finishes, the user will not be able to view the task or there will be no completed task history in their interface. Till the new link arrives, users won't be able to see the website link either, as the link was generated for the particular task.

5.4.5 Data entry Clerk

There is no MFA-based login system for this actor. This actor will just login with the TFA login system. He will have manual data and will be assigned by the super admin to insert those data into the system's database. He will just have the insertion permission and then have a view in real time to see what he entered. After the cross check, he will enter the data he inserted. That data will be viewed by super admin only. So this is the initial operation of this system. With more research analysis, this system will develop more and could be a new and more secure system to do high level confidential tasks or regular organizational operations.

Chapter 6

System Security Analysis

6.1 Security

As this system security is based on authentication and authorization processes, TFA, MFA, and key exchange algorithms, AES encryption, and key hashing are major factors in this system. First of all, to access this system, all types of actors will need a password and username. This will be the entry level security, or basic layer. Then in the next layer to give access, this system will ask the actor to give the security code or OTP if the system is TFA based. The code will go to the users' mobile or email. Then the actor will enter the code to get access. If the system is MFA based, then bio-metrics credential readers such as fingerprint machines, pulse readers, blood sample matches, or iris scanners will be installed with the system, and the system will ask the user to give fingerprints and follow. When the user gives the MFA credentials, the system will save the given credentials for this session. After that, the user will get a code in the next layer to verify their status as valid users of the system. Then the actor will enter the code to get authorization and he can perform what his role allowed him as a categorized authorization. All the data transmitted from website to system to database will be transmitted with AES encryption. Also, the generated code for TFA will be under the hash function.

During the performance of a particular actor (two co-admins), they need to authenticate again before sending a request to launch a task. The system will ask them to provide bio-metrics credentials again, and after they provide them, the system will cross check them with the database's data and also match them with the session's temporarily saved data while co-admins were accessing the system. After the authentication level is reached, the system will perform based on the KSU key exchange algorithm and generate 4 parts of the key, which will be stored in the database after being hashed. But the user will receive the code in decrypted form. These keys will be generated for just one task. So, in the next task, there will be another combination of generated keys. And after the proper confirmation, the task or co-admins' task will be published by the super-admin.

So, from the observation of this system and key factors, there will be three layers of security if the system is based on TFA and KSU key exchange algorithms, and there will be four layers of security if the system is MFA based, as there will be TFA after MFA's part. algorithms So, based on the history of this factor and the newly introduced KSU algorithms, this system is very secure.

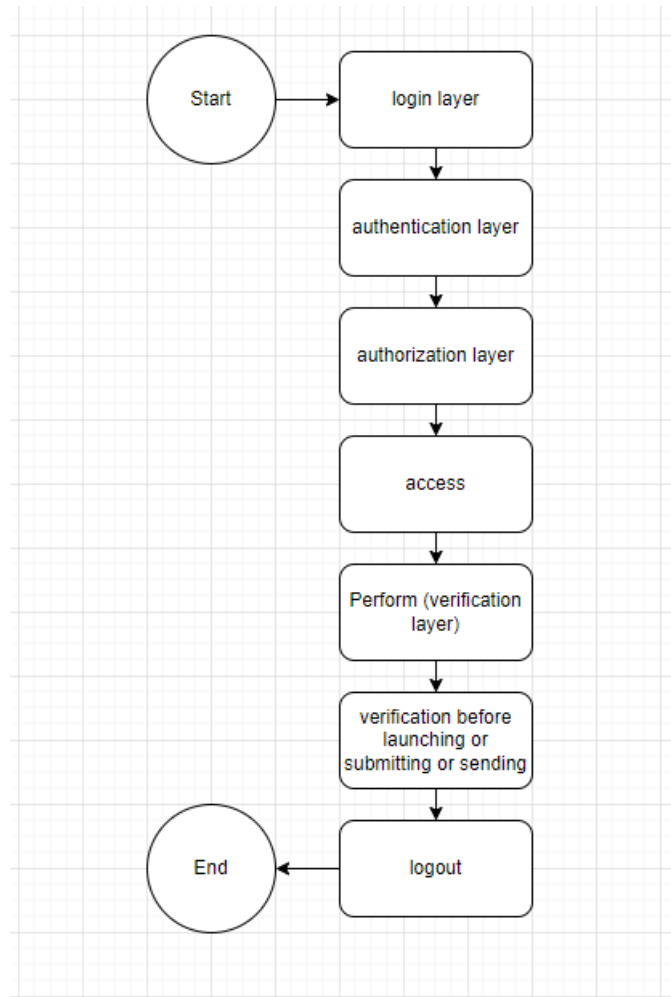


Figure 6.1: Security Layers of the System

6.2 Result analysis

This system is very new and there is a newly developed algorithm, but this system has TFA, MFA, AES encryption, and HASH function, which are very strong factors in any security system. It can be said that this system is also very secure. Based on the current information regarding general login systems, TFA, and MFA and AES and Hash, there can be a probability calculation based on the probability of attack prevention of the mentioned factors.

The general login system's probability of success against any attack in general is 20%. [57] The TFA system's probability of success against any attack in general is 65%. [12] The MFA system's probability of success against any attack in general is 99%. [19] The KEY exchange system's Diffie-Hellman probability of success against any attack in general is 81.9%. [9] The general login system's probability of success against any attack in general is 98.5%. [38] These are the few probabilities of the main factors of the system. From these, it can be calculated that $((20+65+99+81.9+98.5) / 500) * 100 = 72.88\%$, which may be the probability of this system's success rate against all types of cyber attack. Since the AES encryption and MFA method are here, the actual test result may vary from the probability-based result. It may come to a 92% to 95% success rate against cyber attacks.

6.3 System Vulnerability Analysis

This system ensured full authorization and authentication checks. As a result, there is a very small chance of any attack. If an intruder tries to do a man-in-the-middle attack, he will not get any of the data access because of this system's layer-based security. First of all, the two co-admins will send a request for the data access, and the system will generate four keys, which will be provided to the co-admins, and they will enter those keys along with bio-metric authentication. So, even if an intruder sits in the middle and gets access to the keys, he will not get access to any data. Also, this system will generate different keys for every session. So, it is very difficult to break down this system's security with a man-in-the-middle attack.

A dictionary attack will not be possible in this system. In a dictionary attack, an intruder tries to break the system with common dictionary words, but in this system, the keys will be generated randomly every time the co-admins try to access the data. There will be no use of any common dictionary word, so no dictionary attack is possible here. This system generates four keys, and each key will consist of eight bits. So, a total of 32 bits of keys. After that, these keys will be encrypted with the SHA-256 hash function, which will encrypt the key with 256 bits. As a result, a brute force attack will be really tough to break the system as there will be high computation of each bit of each key. Even if an intruder tries to use a brute force attack, it will take a lot of time. Also, per session keys are being generated randomly so with a brute force attack this system can not be fully compromised.

This system cannot be hacked with malware or phishing attacks because with malware the full system cannot be compromised as it is a layer-based system. If a phishing attack happens, an intruder may get the keys of the co-admins but they can't access the system as when they try to log in with that key, they have to give the credentials of the co-admins as well as verify the identity with bio-metrics. This system is also secure in terms of identity access control and authorization. To sum up it can be said that this system is secure enough to defend against the most common attacks of this time. As a result, this proposed model can be said to be one of the most secure systems in terms of authorization, authentication, and data access control.

6.4 System Model Comparison

For securing a system from various kinds of cyber and physical threats, there are so many methods or models of security systems. which are very good in their own situation and perspective. Some models of security systems use only login systems or at least login systems with TFA or MFA [34]. Very few systems use MFA, and even fewer systems use encryption decryption for authentication. Only a very few of them use layered architecture for security [63]. But those layered architectures are not good enough as they don't provide internal performance authentication for security. which may become a good target for man-in-the-middle attack, or brute force attack, or other types of attacks.[39]

AMDH service providers [33] offer multi-layered security services that are intended to mitigate, postpone, or avoid threats. This service takes a holistic approach from the consumer to the staff by implementing access control at each level. They attempt

to control data access not only by controlling the server able to host applications but also by configuring the laptop and server, the host firewall, switch configurations, routing configuration, connectivity, the firewalls that secure the facilities they access, detection systems, DDOS protection, tracking, and correlation of potential attacks. By monitoring server data, the organizers attempted to ensure the environment. For secure remote access, this system employs a firewall and other safeguards. They also use regular upgrades on devices, servers, applications, and equipment.

Here in the paper, this proposed system is very much well secured as this has three layers of security for TFA based and four layers of security for MFA based system. This only just for authentication and authorization purpose. Each login system for each type of actor is very secure with general login system with TFA or MFA TFA. Also this system will transmit all data from user to user so user to system or system to system in AES transmission mode. Also all kind of pin and password or authentication key will be hashed with SHA-256 Hashing algorithm before storing in database. Also there will be the newly introduced KSU key exchange algorithm, which will be used for authentication and giving authorization before doing or sending any task or also it can be used in login layer with MFA to secure any access point. With this type of strong layer security it will be very much difficult for a third party or any unauthorized person to access in this system. This system maybe look complex and may the installation price defer from for the client system's requirement and very much easy to use and will be very fast if the requirements of this system meets before using this. With proper maintenance, updates and properly maintain tools setup this system will become a headache for hackers to break it. From various calculation and perspective this can be said that this proposed Multi-Layer Security System will be much better than the existing security system for data access control, authentication and authorization.

6.5 Nobility of the system

In this system, there is a unique approach which makes this system different from other models, which is multi layer based security and the newly introduced key exchange algorithm. This multi layer security system can be customized or modified by the organization or the clients according to their needs. And the key exchange algorithm is based on four randomly generated keys for every session. For each needed minimum, there will be two keys, which will be provided to their email. First of all, the two co-admins will send a request for the data access, and the system will generate four keys, which will be provided to the co-admins. First, the co-admin will enter the primary key 1 and secondary key 2, and the second co-admin will enter the primary key 2 and secondary key1 for the verification of the process. Every time the co-admins try to access the system or send a request, they have to go through this key exchange verification process. This key exchange verification will ensure the security of the authorization and authentication. This is the unique property that makes this system different from other models and makes the whole system or data secure.

Chapter 7

System Implementation

7.1 Implementation showcasing

This system has been implemented initially in small scale where the system will run all the basic layer security and the system also runs the newly introduced KSU algorithm for the verification process. The system features and the screenshots of the system has given below with the explanation of the implementation.

7.1.1 Super Admin Activity

Super Admin can create admin, co-admin, or user roles. The picture below shows the super admin dashboard where he can see all the users, tasks and the roles of the system. He can edit and add or delete users or roles for the system.

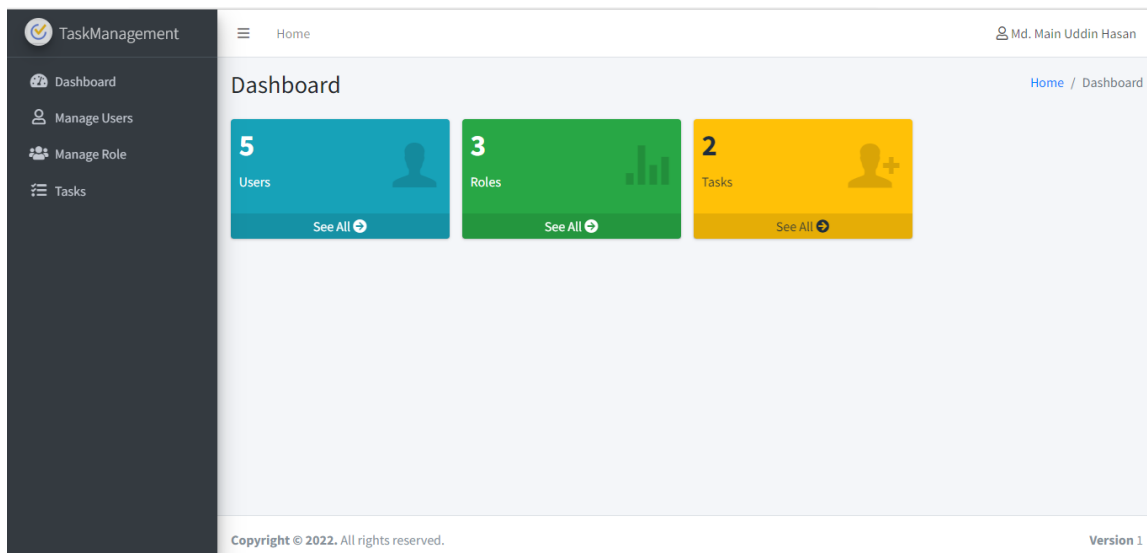


Figure 7.1: Super Admin Dashboard

Super-admin has access to all the information of co-admins and users. He can check the email, contact no, roles and the action of the co-admins and users.

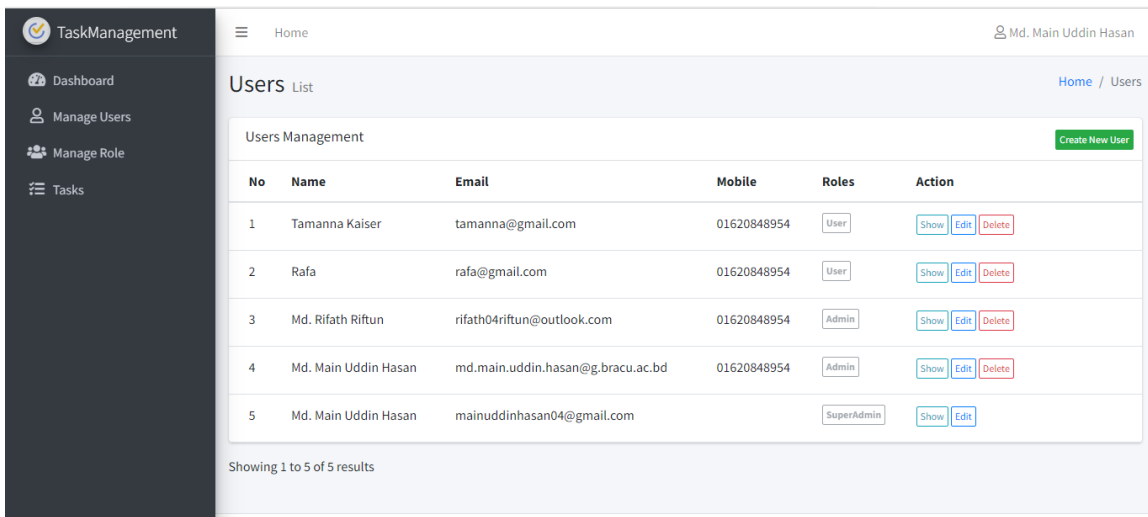


Figure 7.2: User or admin creation

Super-Admin has access to see all the information, including co-admins or users assigned tasks. After co-admins assign any task for the users and sent approval request to the super admin. Then super admin will approve the request only after the key exchange verification of the co-admins. Next, super admin will publish the task to the users which was assigned by the co-admins.

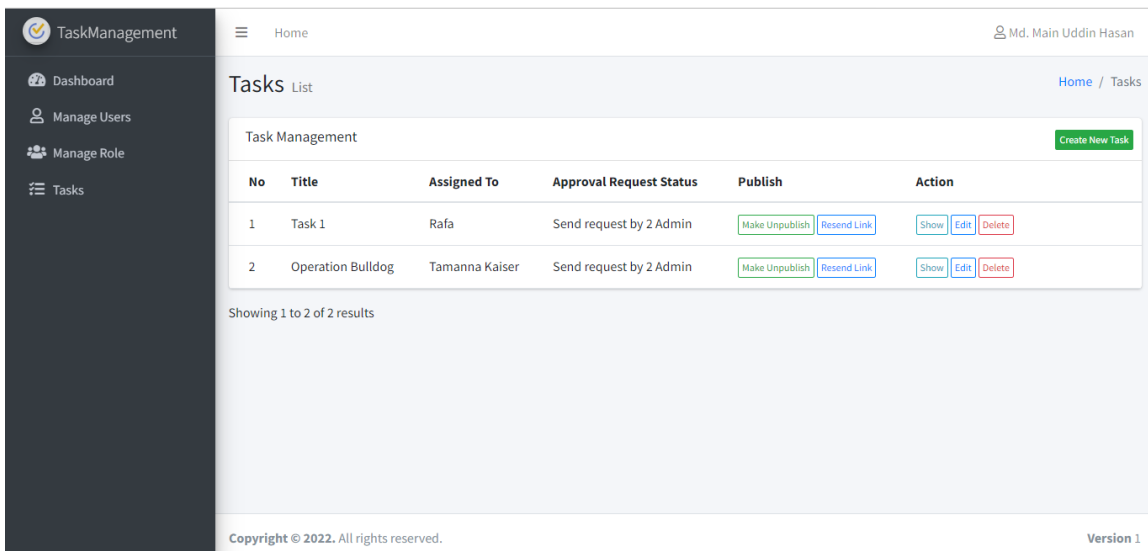


Figure 7.3: Task Creation

7.1.2 Co-Admin Activity

Co-admin will assign the tasks to the users. They can create new task and can assign the task to the user.

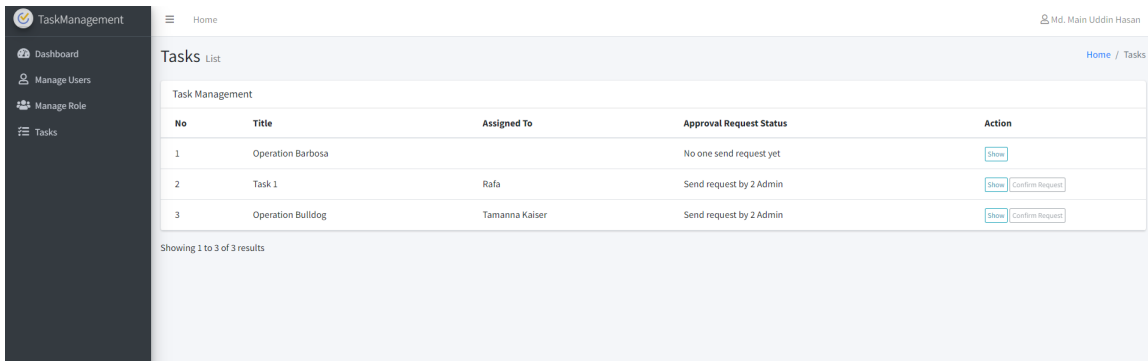


Figure 7.4: Task Interface

After assigning tasks, co-admins will send the request to the super admin. So that super admin can approve the request and make it publish for the users.

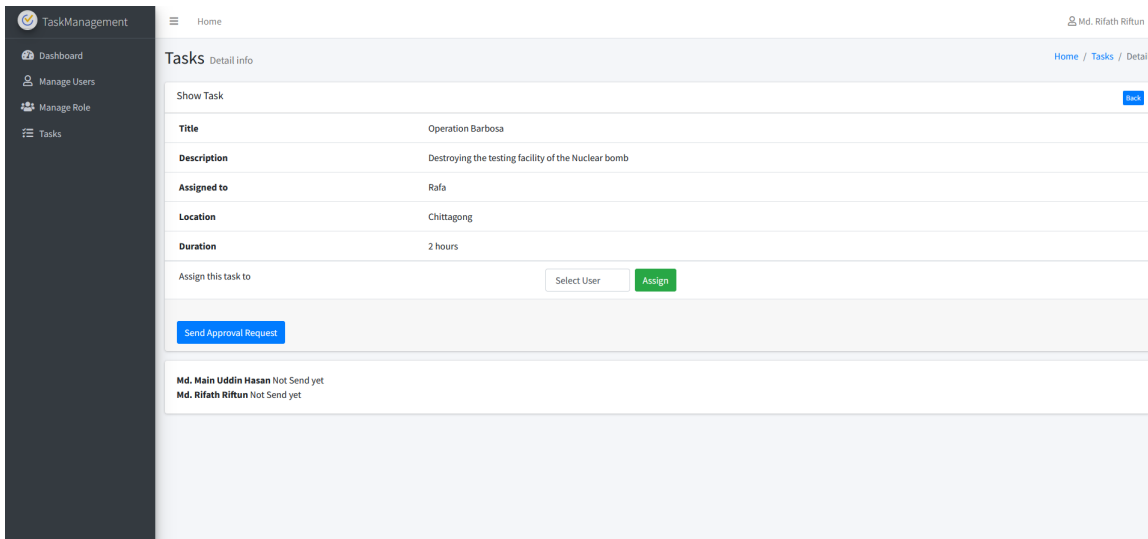


Figure 7.5: Task Assign to user and request send

After sending the request , the primary and secondary keys will be generated and will be sent to the two co-admins via email. Each co-admins will get their own primary key and the secondary key of the other co-admin.

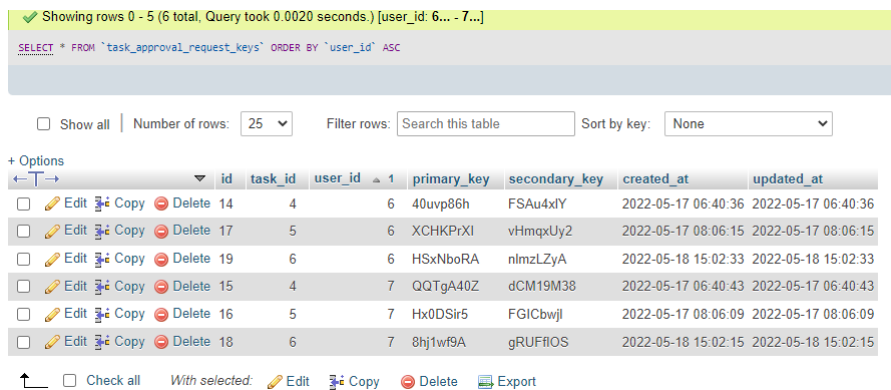


Figure 7.6: Key Generation Database

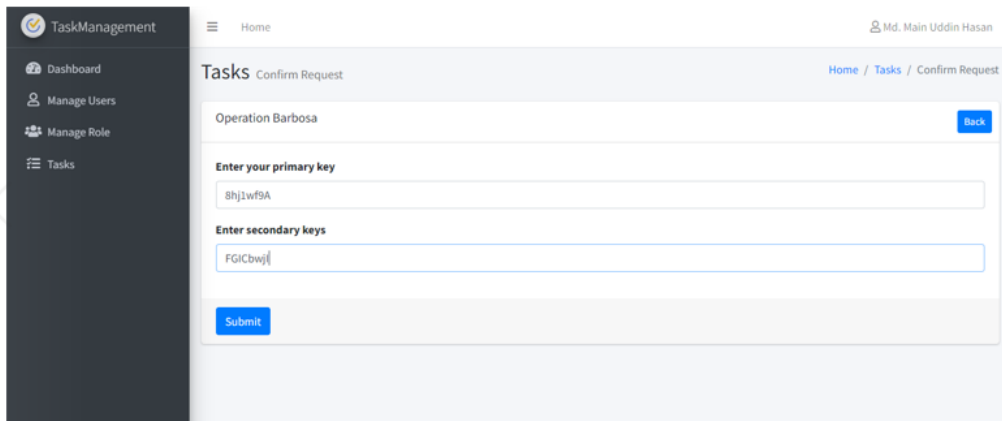


Figure 7.7: Key Exchange Input UI for verification

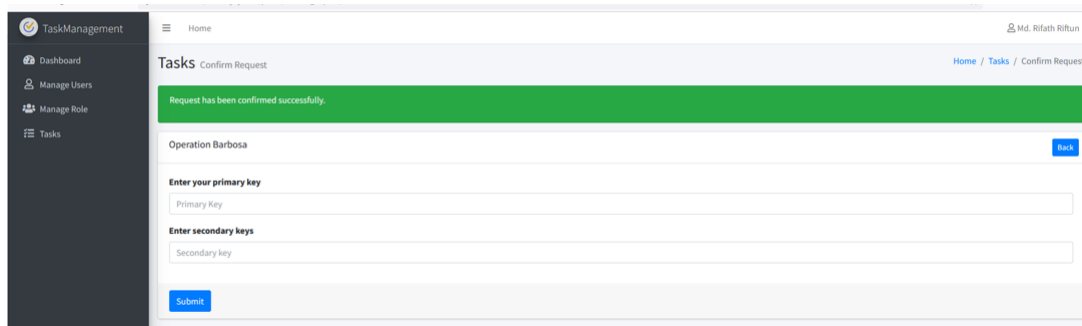


Figure 7.8: Key Exchange Verification Success

7.1.3 Task Publishing

After the verification, the co-admins request will be shown to the super admin and after getting the request super admin will launch or publish the assigned task to the users as the publish button will be visible to the super admin.

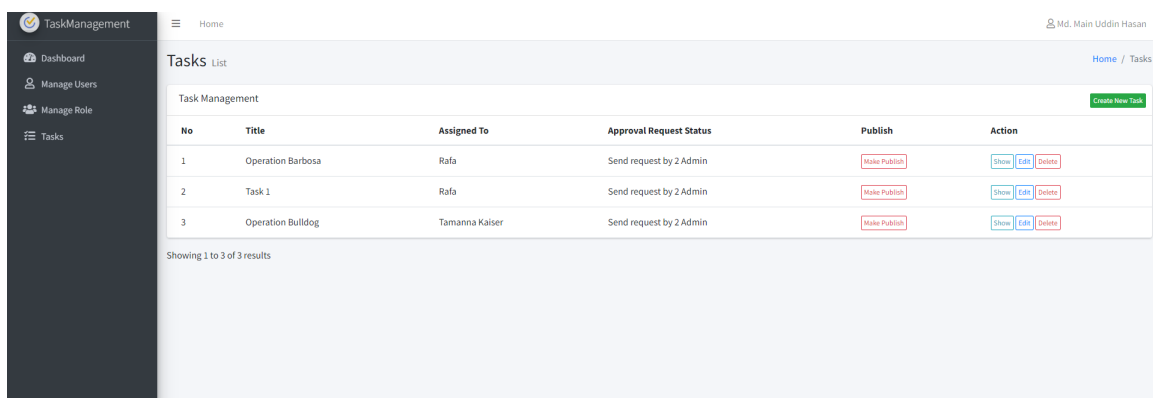


Figure 7.9: After Successful Verification

After the visibility of the publish button super admin will publish the tasks. Then this task will be shown to the users it is assigned to.

Tasks List Home / Tasks

Task Management Create New Task

No	Title	Assigned To	Approval Request Status	Publish	Action
1	Operation Barbosa	Rafa	Send request by 2 Admin	Make Unpublish Resend Link	Show Edit Delete
2	Task 1	Rafa	Send request by 2 Admin	Make Unpublish Resend Link	Show Edit Delete
3	Operation Bulldog	Tamanna Kaiser	Send request by 2 Admin	Make Unpublish Resend Link	Show Edit Delete

Showing 1 to 3 of 3 results

Figure 7.10: After Publishing Task to the user

7.2 Code Explanation

7.2.1 Request and Key Generation

After co-admins request the system will generate keys and send it to their email.

```
public function task_approval_request(Request $request)
{
    $primary_key = Str::random(8);
    $secondary_key = Str::random(8);

    $input = $request->all();
    $input['user_id'] = Auth::user()->id;
    $input['primary_key'] = $primary_key;
    $input['secondary_key'] = $secondary_key;

    $task_id = $request->task_id;
    $task = Task::where('id', $task_id)->first()->toArray();
    $request_sender = Auth::user();

    $checkData = TaskApprovalRequestKeys::where(["task_id" => $task_id, "user_id" => $user_id]->exists());

    if($checkData){
        return redirect()->route('tasks.index')
            ->with('success', 'You have already send approval request for this task.');
```

```
    }else{
        TaskApprovalRequestKeys::create($input);

        /* --update task table if all admin send approval request for the task-- */
        $admins = User::whereHas('roles', function($q){$q->whereIn('name', ['Admin']);})->pluck('id')->toArray();

        $taskApprovalRequestBy = TaskApprovalRequestKeys::where(["task_id" => $task_id])->pluck('user_id')->toArray();
        array_push($taskApprovalRequestBy, $user_id);

        $checkApprovalRequest = array_diff($admins, $taskApprovalRequestBy);

        if(empty($checkApprovalRequest)){ /* if all admin send approval request for the task */
            Task::where('id', $task_id)->update(['approved' => 1]);
        }
        /* --/update task table if all admin send approval request for the task-- */

        /*-----send email to request sender-----*/
        $details = [
            "user_name" => $request_sender->name,
            "task_title" => $task['title'],
            "primary_key" => $primary_key
        ];
        Mail::to($request_sender->email)->send(new TaskApprovalRequestMail($details));
        /*-----/send email to request sender-----*/

        $other_admins = User::whereHas('roles', function($q){$q->whereIn('name', ['Admin']);})->where('id', '!=', $user_id)->get(['id', 'name', 'email'])->toArray();
```

Figure 7.11: Send Request and Key Generation Part 1

```

$other_admins = User::whereHas('roles', function($q){$q->whereIn('name', ['Admin']);})->where('id', '!=', $user_id)->get(['id', 'name', 'email']->toArray());

foreach($other_admins as $admin){
    /*-----send email to other admins-----*/
    $details_secondary = [
        "user_name" => $admin['name'],
        "task_title" => $task['title'],
        "request_senders_email" => $request_sender->email,
        "secondary_key" => $secondary_key
    ];
    Mail::to($admin['email']->send(new TaskApprovalRequestSecondaryMail($details_secondary));
    /*-----/send email to other admins-----*/
}

return redirect()->route('tasks.index')
->with('success','Task approval request send successfully.');
```

Figure 7.12: Send Request and Key Generation part 2

7.2.2 Request approval status

The system will update the table after sending the request

```

public function task_approve(Request $request)
{
    $input = $request->all();
    $input['user_id'] = Auth::user()->id;
    $input['approved'] = 1;

    $task_id = $request->task_id;

    $checkData = TaskApprovedBy::where(["task_id" => $task_id, "user_id" => $user_id])->exists();

    if($checkData){
        return redirect()->route('tasks.index')
        ->with('success','You have already approve this task.');
```

Figure 7.13: Task Approve and Approve Status part 1

```

public function approve_status($task_id)
{
    $count_data = TaskApprovedBy::where(["task_id" => $task_id, 'approved' => 1])->get('user_id')->count();
    return $count_data;
}

public function send_approval_request_status($task_id)
{
    $count_data = TaskApprovalRequestKeys::where(["task_id" => $task_id])->get('user_id')->count();
    return $count_data;
}

public function check_approval_request_status($task_id, $user_id)
{
    $check_approval_request_status = TaskApprovalRequestStatus::where(["task_id" => $task_id, 'user_id' => $user_id])->first();
    return $check_approval_request_status;
}
```

Figure 7.14: Task Approve and Approve Status part 2

7.2.3 Request Confirmation and Key verification

The system will show the request confirmation and the key verification process will be finished.

```
public function confirm_request_status(Request $request)
{
    $request->validate([
        'primary_key' => 'required',
        'secondary_keys.*' => 'required'
    ]);

    $input = $request->all();

    $task_id = $request->task_id;          /*-- form data --*/
    $requested_primary_key = $request->primary_key; /*-- form data --*/
    $input['user_id'] = $user_id = Auth::user()->id;

    $checkData = TaskApprovalRequestStatus::where(["task_id" => $task_id, "user_id" => $user_id])->exists();

    if($checkData){
        return redirect()->route('tasks.confirm_request', [$task_id])
            ->with('success', 'You have already confirmed request for this task.');
```

Figure 7.15: Request Confirmation Status and key verification part 1

```
    }else{
        /*-- get logged in admin's data from db table --*/
        $taskApprovalRequestPrimaryKey = TaskApprovalRequestKeys::where(["task_id" => $task_id, 'user_id' => $user_id])->first();
        if($taskApprovalRequestPrimaryKey['primary_key'] != $requested_primary_key){ /*-- match admin's form data & db data --*/
            return redirect()->route('tasks.confirm_request', [$task_id])->with('success', 'primary key not match.');
```

Figure 7.16: Request Confirmation Status and key verification part 2

Chapter 8

Conclusion

8.1 Advantages

The system's main advantage is that it focuses on data authorization, authentication, and access control. This system is here to guarantee that by using the system, a client can easily get a fully secure system. where they can assign a supervisor, super admin, and co-admin. Only these authorized people can access the data after getting the primary and secondary key exchanges. The data will be shown to the person it is assigned to. So the system is focusing on authorization, authentication, and data access control, which are the advantages of the system.

- Can be customized as per customer or organization requirements.
- Intruders cannot breach the whole system at once as it is a multi-layered based security system.
- It has a built-in key exchange system so employers or admins can access only the authorized data of the server after the verification.
- Data access is highly secured, so it is impossible for anyone to get access without a proper credential.
- It will have a backup server, so there will be no chance of data loss.
- It will have a one-way connection between the primary and secondary servers. So, it reduces the chances of an external or internal data breach.
- It will have a session time-out system for which data will be more secure from unwanted authorization.
- In the future, a time countdown will be added that will remind the users of the time limitation for accessing the data.
- To secure this application soon, a biometrics authorization system login will be added for the super admin and co-admin.
- Each primary and secondary OTP will consist of 8-bit. So, if anyone tries to break into the system, they have to work with 32 bits which will be very tough and time consuming.

8.2 Limitations

Every system has some limitations such as ours. Though this system ensures the best security system yet there are a few limitations which are being discussed below. There will be so many future upgrades by which the limitations will be reduced and it will be one of the best authorization, authentication, and data access control models.

- After installing the system, the maintenance cost can be high for some organizations as it has different layers-based security so it will have different maintenance costs for each layer.
- When the system will add a biometric login system for super admin and co-admin, the system will need an extra server for storing the data of biometrics for which the clients have to pay some extra cost.
- This system may require regular updates. If any organization does not update the system regularly then there can be a security gap which will lead to unwanted data breaches and authentication.
- For future developments, the system will require a large amount of investment to provide a secure system for the clients.

8.3 Future Work

While developing this system we have many plans for the entire application or project for the future. By developing those certain applications the system can be made more secure and suitable for work and time efficient.

- Session time out for the end user will be added, where the system will set the time in minutes instead of hour for the users to finish their work.
- There will be a countdown system to remind the user of time limitations.
- There is a future plan to add an encrypted biometric system like fingerprint or iris scanner for the admin side to make sure that rather than super-admin and co-admin, anyone can not access data from the main and secondary server.
- Super admin , Co-admin, and User will use mfa supported login system.
- Device: there will be encrypted mobile devices(carriable) like cell-phones,tablets for super admin.So that Super-admin can access data from anywhere.
- System will have a backup or secondary server. In the future from the secondary server the co-admins will do their task,super admin will handle the main database and will pass the task to the secondary data base.After passing the task in the secondary server There will be no connection between those two servers. There will be one way communication (tunneling) from main server to secondary server

- In future this system will have some testing activities like Software performance testing, Usability testing, Accessibility testing, Security testing which will deliver the application developments and requirements.
- System going to be immediately shutdown in case of any intruder attack and the Mac address of the intruder laptop or pc will be blocked.

8.4 Conclusion

As we found the security issues while researching we tried to implement a fully secure system to ensure that there are no vulnerabilities which we found in so many models. After the implementation of this system it is nearly shown that this system can handle all the security issues of a system which can include authorization, authentication and data access control. It is a system of a combination of full security where the data is authorized for certain people, when these people are trying to access the data they have to go through with a full secure authentication process. After the authentication the super admin can control the data access as a result no unwanted person can access the data or get the authority to get authenticated in the system. Moreover, this system is ensuring the full secure model which was not found in so many existing models. This system is a unique approach with a newly introduced key exchange algorithm to ensure the full security of the data and the system.

Thus, this research paper represents a full layer-by-layer security approach to provide full data and access control layer security. In this way, the service can be fully secured for the future. Here is the multi-layer security system to make the data storage and workplace more secure and suitable. Layer security is going to protect the systems most valuable information of the technology environment where a breach or cyberattack could occur, with access control authentication by using two-factor-authentication and multi-factor-authentication along with biometric authentication. By enabling this, we can reduce the risk of different types of cyber attack. Along with Two-factor-authentication, MFA requires multiple forms of verification to access the application, account, or the network. For example, after entering one's password, one will be prompted to enter a one-time code sent via email, text message or push notification. These additional forms of authentication prevent hackers from exploiting weak or compromised end-user credentials from the system. To conclude, it can be said that this system model will be one of the best approaches for ensuring data security, authorization and authentication.

Bibliography

- [1] H. Nielsen, "Connection between the regge trajectory universal slope α and the transverse momentum distribution of partons in planar feynman diagram model," *Physics Letters B*, vol. 35, no. 6, pp. 515–518, 1971.
- [2] M. Shepperd, "A critique of cyclomatic complexity as a software metric," *Software Engineering Journal*, vol. 3, no. 2, pp. 30–36, 1988.
- [3] R. J. Nemiroff and J. T. Bonnell, "Astronomy picture of the day: [Http://antwrp.gsfc.nasa.gov/apod/astropix.html](http://antwrp.gsfc.nasa.gov/apod/astropix.html)," in *American Astronomical Society Meeting Abstracts*, vol. 187, 1995, pp. 05–06.
- [4] T. de Witte, S. Suci, M. Peetermans, *et al.*, "Intensive chemotherapy for poor prognosis myelodysplasia (mds) and secondary acute myeloid leukemia (saml) following mds of more than 6 months duration: A pilot study by the leukemia cooperative group of the european organisation for research and treatment in cancer (eortc-lcg)," 1995.
- [5] L. D. Stein, "Web security," *Addison-Wesley, Massachusetts*, vol. 26, pp. 1–4, 1998.
- [6] V. Batagelj, "Logo to svg," in *Proceedings of the 8th European Logo Conference, Linz, Austria*, 2001, pp. 21–25.
- [7] C.-L. Lin and T. Hwang, "A password authentication scheme with secure password updating," *Computers & Security*, vol. 22, no. 1, pp. 68–72, 2003.
- [8] J. Rosenberg and D. Remy, *Securing web services with WS-security: Demystifying WS-security, WS-policy, SAML, XML signature, and XML encryption*. Pearson Higher Education, 2004.
- [9] B. A. Forouzan, "Cryptography and network security. special indian," *Tata Mc-Graw-Hill. ISBN*, vol. 13, pp. 978–, 2007.
- [10] S. Heron, "Advanced encryption standard (aes)," *Network Security*, vol. 2009, no. 12, pp. 8–12, 2009.
- [11] P. Mell and T. Grance, "The nist definition of cloud computing. national institute of standards and technology," *Information Technology Laboratory, Version*, vol. 15, no. 10.07, p. 2009, 2009.
- [12] S. Lee, I. Ong, H.-T. Lim, and H.-J. Lee, "Two factor authentication for cloud computing," *Journal of information and communication convergence engineering*, vol. 8, no. 4, pp. 427–432, 2010.
- [13] A. Sirisha and G. G. Kumari, "Api access control in cloud using the role based access control model," in *Trendz in Information Sciences & Computing (TISC2010)*, IEEE, 2010, pp. 135–137.

- [14] M. Ates, S. Ravet, A. M. Ahmat, and J. Fayolle, "An identity-centric internet: Identity in the cloud, identity as a service and other delights," in *2011 Sixth International Conference on Availability, Reliability and Security*, IEEE, 2011, pp. 555–560.
- [15] A. Behl, "Emerging security challenges in cloud computing: An insight to cloud security challenges and their mitigation," in *2011 World Congress on Information and Communication Technologies*, IEEE, 2011, pp. 217–222.
- [16] R. H. Khan, J. Ylitalo, and A. S. Ahmed, "Openid authentication as a service in openstack," in *2011 7th International Conference on Information Assurance and Security (IAS)*, IEEE, 2011, pp. 372–377.
- [17] S. Pippal, V. Sharma, S. Mishra, and D. S. Kushwaha, "An efficient schema shared approach for cloud based multitenant database with authentication and authorization framework," in *2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, IEEE, 2011, pp. 213–218.
- [18] K. S. Gonzales, "Technology tools for improving online learning environments," 2012.
- [19] R. K. Banyal, P. Jain, and V. K. Jain, "Multi-factor authentication framework for cloud computing," in *2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation*, IEEE, 2013, pp. 105–110.
- [20] R. Rai, G. Sahoo, and S. Mehruz, "Securing software as a service model of cloud computing: Issues and solutions," *arXiv preprint arXiv:1309.2426*, 2013.
- [21] P. S. S.L.Mewada U.K. Singh, "Security enhancement in cloud computing (cc)," *International Journal of Scientific Research in Computer Science and Engineering*, vol. 1, pp. 31–37, 1 Jan. 2013, ISSN: 2347-2693. [Online]. Available: https://www.isroset.org/journal/IJSRCSE/full_paper_view.php?paper_id=39.
- [22] L.-H. Chang, S. Behl, and T.-H. Shieh, "W-revised: An amazing tool for creating customized websites," in *The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, IEEE, 2014, pp. 465–470.
- [23] W. Liu, A. S. Uluagac, and R. Beyah, "Maca: A privacy-preserving multi-factor cloud authentication system utilizing big data," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2014, pp. 518–523.
- [24] S. Mahnken, "Today's authentication options: The need for adaptive multifactor authentication," *Biometric Technology Today*, vol. 2014, no. 7, pp. 8–10, 2014.
- [25] Y. Sun, J. Zhang, Y. Xiong, and G. Zhu, "Data security and privacy in cloud computing," *International Journal of Distributed Sensor Networks*, vol. 10, no. 7, p. 190 903, 2014.
- [26] A. Alabaichi and A. I. Salih, "Enhance security of advance encryption standard algorithm based on key-dependent s-box," in *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, IEEE, 2015, pp. 44–53.

- [27] S. Boni, J. Bhatt, and S. Bhat, “Improving the diffie-hellman key exchange algorithm by proposing the multiplicative key exchange algorithm,” *International Journal of Computer Applications*, vol. 130, no. 15, 2015.
- [28] D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman, and D. Woods, “Cloud-trust—a security assessment model for infrastructure as a service (iaas) clouds,” *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 523–536, 2015.
- [29] S. A. Jaju and S. S. Chowhan, “A modified rsa algorithm to enhance security for digital signature,” in *2015 international conference and workshop on computing and communication (IEMCON)*, IEEE, 2015, pp. 1–5.
- [30] M. Kazim and S. Y. Zhu, “A survey on top security threats in cloud computing,” 2015.
- [31] R. Rai, G. Sahoo, and S. Mehfuz, “Exploring the factors influencing the cloud computing adoption: A systematic study on cloud migration,” *SpringerPlus*, vol. 4, no. 1, pp. 1–12, 2015.
- [32] Y. Shah, V. Choyi, and L. Subramanian, “Multi-factor authentication as a service,” in *2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, IEEE, 2015, pp. 144–150.
- [33] J. B. Hong and D. S. Kim, “Towards scalable security analysis using multi-layered security models,” *Journal of Network and Computer Applications*, vol. 75, pp. 156–168, 2016.
- [34] R. Malathi *et al.*, “An integrated approach of physical biometric authentication system,” *Procedia Computer Science*, vol. 85, pp. 820–826, 2016.
- [35] R. Parsamehr and S. F. H. Nezhad, “Mutual authentication protocol to share files in cloud storage,” in *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, IEEE, 2016, pp. 153–158.
- [36] S. Rajani, V. Ghorpade, and M. Dhange, “Multi-factor authentication as a service for cloud data security,” *Int J Comput Sci Eng*, vol. 4, pp. 43–46, 2016.
- [37] P. Samarati, S. D. C. di Vimercati, S. Murugesan, and I. Bojanova, *Cloud security: Issues and concerns*. Wiley Chichester, 2016.
- [38] A. M. Abdullah *et al.*, “Advanced encryption standard (aes) algorithm to encrypt and decrypt data,” *Cryptography and Network Security*, vol. 16, pp. 1–11, 2017.
- [39] P. Ping, Z. Xuan, and M. Xinyue, “Research on security test for application software based on spn,” *Procedia engineering*, vol. 174, pp. 1140–1147, 2017.
- [40] J. Shen, D. Liu, Q. Liu, X. Sun, and Y. Zhang, “Secure authentication in cloud big data with hierarchical attribute authorization structure,” *IEEE Transactions on Big Data*, vol. 7, no. 4, pp. 668–677, 2017.
- [41] L. Shtika, “Challenges and benefits of developing an open-source & full-stack” conference management” framework,” 2017.
- [42] A. Singh and K. Chatterjee, “Cloud security issues and challenges: A survey,” *Journal of Network and Computer Applications*, vol. 79, pp. 88–115, 2017.

- [43] T. H. Vo, W. Fuhrmann, and K.-P. Fischer-Hellmann, "How to adapt authentication and authorization infrastructure of applications for the cloud," in *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, IEEE, 2017, pp. 54–61.
- [44] K. Zkik, M. Tebaa, T. Tachihante, and G. Orhanou, "A new authentication and homomorphic encryption as a service model for preserving privacy in clouds.," *J. Comput. Sci.*, vol. 13, no. 12, pp. 702–717, 2017.
- [45] A. K. Agrahari, M. Sheth, and N. Praveen, "Comprehensive survey on image steganography using lsb with aes," *Int. J. Appl. Eng. Res.*, vol. 13, no. 8, pp. 5841–5844, 2018.
- [46] A. Choudhary, I. Gupta, V. Singh, and P. K. Jana, "A gsa based hybrid algorithm for bi-objective workflow scheduling in cloud computing," *Future Generation Computer Systems*, vol. 83, pp. 14–26, 2018.
- [47] J. Hordeaux, Q. Wang, N. Katz, E. L. Buza, P. Bell, and J. M. Wilson, "The neurotropic properties of aav-php. b are limited to c57bl/6j mice," *Molecular Therapy*, vol. 26, no. 3, pp. 664–668, 2018.
- [48] I. Indu, P. R. Anand, and V. Bhaskar, "Identity and access management in cloud environment: Mechanisms and challenges," *Engineering science and technology, an international journal*, vol. 21, no. 4, pp. 574–588, 2018.
- [49] H. Jiang, M. Xie, B. Kang, C. Li, and L. Si, "Id-based public auditing protocol for cloud storage data integrity checking with strengthened authentication and security," *Wuhan University Journal of Natural Sciences*, vol. 23, no. 4, pp. 362–368, 2018.
- [50] D. Jing, J. Yan, A. Fujiang, and Z. Ying, "An improved uniform identity authentication method based on saml in cloud environment," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, IEEE, 2018, pp. 533–536.
- [51] Q. K. Kadhim, R. Yusof, H. S. Mahdi, S. S. A. Al-Shami, and S. R. Selamat, "A review study on cloud computing issues," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1018, 2018, p. 012 006.
- [52] N. Veeraragavan, "Design and implementation of authentication as a service (aaas) in windows azure cloud platform," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1142, 2018, p. 012 016.
- [53] C. Wang, K. Ding, B. Li, *et al.*, "An enhanced user authentication protocol based on elliptic curve cryptosystem in cloud computing environment," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.
- [54] T. Xiang, X. Li, F. Chen, Y. Yang, and S. Zhang, "Achieving verifiable, dynamic and efficient auditing for outsourced database in cloud," *Journal of Parallel and Distributed Computing*, vol. 112, pp. 97–107, 2018.
- [55] S. S. Chauhan, "Conversion of stream cipher into block cipher," in *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM)*, Amity University Rajasthan, Jaipur-India, 2019.

- [56] S. CHEN, “Developing a corporate website for anlu aima electric bicycle shop,” 2019.
- [57] C. Guo, P. Tian, and C.-C. Chang, “Privacy preserving weighted similarity search scheme for encrypted data,” *IET Information Security*, vol. 13, no. 1, pp. 61–69, 2019.
- [58] Y. Liu, S. Xiao, H. Wang, and X. A. Wang, “New provable data transfer from provable data possession and deletion for secure cloud storage,” *International Journal of Distributed Sensor Networks*, vol. 15, no. 4, p. 1 550 147 719 842 493, 2019.
- [59] V. Nikhila and C. Rupa, “Intensifying multimedia information security using comprehensive cipher,” in *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*, IEEE, vol. 1, 2019, pp. 1–4.
- [60] T. Senthilnathan, P. Prabu, R. Sivakumar, and S. Sakthivel, “An enhancing reversible data hiding for secured data using shuffle block key encryption and histogram bit shifting in cloud environment,” *Cluster Computing*, vol. 22, no. 5, pp. 12 839–12 847, 2019.
- [61] T. H. Vo, W. Fuhrmann, K.-P. Fischer-Hellmann, and S. Furnell, “Identity-as-a-service: An adaptive security infrastructure and privacy-preserving user identity for the cloud environment,” *Future Internet*, vol. 11, no. 5, p. 116, 2019.
- [62] S. Kumarasinghe, M. Shafana, and M. Ahamed Sabani, “A prototypical adoption security model for major vulnerabilities in cloud computing,” 2021.
- [63] G. Zhai, J. Zhou, and X. Yang, “Communications in computer and information science,” *Digital TV and Wireless Multimedia Communication*, vol. 815, pp. 128–137,