Cyber Intelligence: Enabling Secure Communication in Legacy Systems to Prevent Phishing and MITM Attacks

Mrs. Thamizharasi M
Professor of Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
[email]@rajalakshmi.edu.in

Guruprasath P
Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
221801014@rajalakshmi.edu.in

Harshini MD
Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
221801017@rajalakshmi.edu.in

Abstract— **Secure communication is essential in today's digitally connected world, especially in light of the increase in phishing attacks that target outdated software that lacks built-in cyber security protections. This study introduces "Project-R," a new secure communication system that allows encrypted data transmission between endpoints by utilizing 4096-bit asymmetric encryption. In order to accomplish end-to-end encryption without changing the network layer and maintain compatibility with legacy systems, the project makes use of a public-private key scheme. Project-R minimizes the requirement for specialized user expertise and the risks associated with human mistake by automating key generation, maintenance, and verification through the use of a server-client architecture. Private keys securely handle decryption on the client side, while public keys are generated and distributed by the server to facilitate the seamless transmission of encrypted data across established protocols. The unique feature of Project-R is its ability to integrate at the application layer, avoiding common network changes and providing an easily navigable, scalable solution that can be used with Software as a Service (SaaS) models. Cloud-based architecture solves scalability issues by offering variable subscription access. Furthermore, the automation of the system lowers the overhead associated with training, enabling users to effectively build and maintain secure communication channels even with less cyber security knowledge. By requiring integrity checks during data transmission, which guarantees data authenticity and prohibits unwanted modification, the project significantly reduces the risks posed by phishing.**

Preliminary findings show that this encryption architecture, which uses high-bit encryption keys, improves communication confidentiality and data integrity by making brute-force attacks computationally impractical. Project-R has demonstrated promise as a workable method for safely transferring sensitive data while preserving compatibility with legacy applications and providing a simplified, low-intervention user experience through real-world simulations.

Keywords*— Secure communication, asymmetric encryption, public and private keys, phishing prevention, cloud scalability, legacy systems, application layer security.*

I. Introduction

The cyber security landscape is facing increasing issues due to the exponential expansion of digital communication, especially for older systems that lack built-in security features. Conventional methods of protecting these systems usually require expensive software overhauls or intricate changes to the network layer, which are unfeasible for many enterprises. This paper introduces "Project-R," a secure communication system meant to guarantee data integrity and confidentiality in legacy applications through the use of 4096-bit public and private key encryption. Project-R implements strong data encryption to prevent unwanted parties from accessing sensitive information, in response to the growing sophistication of phishing assaults.

Based on a client-server architecture, Project-R automates key distribution and creation to speed up the encryption and decryption procedure. The server creates a distinct public key for every client when connection is established, allowing encrypted data transfer without the need for manual key management—a frequent cause of human mistake in cyber security procedures. With the matching private key, the client may safely decode the received data, minimizing user intervention and offering a smooth experience. Because the system's architecture integrates at the application layer, it can remain compatible with the current infrastructure without requiring changes at the network level. Together with subscription-based access through cloud platforms, this adaptability solves scaling issues and makes the system more widely available to a variety of customers, from small businesses to major companies.

Integrity checks are incorporated into Project-R to identify and stop phishing efforts, adding another level of data authenticity. It minimizes operational complexity and lowers the need for user training by automating security procedures. By means of comprehensive testing and practical simulations, Project-R showcases a dependable and intuitive method for secure communication, striking a balance between robust encryption, ease of use, and compatibility with legacy systems. This study highlights Project-R's promise as a scalable, affordable solution for secure digital communication by analysing its design, security protocols, and performance results.

Figure 1: Tech Stacks

II. RELATED WORKS

In the paper "Real-time client-side phishing prevention add-on," G. Armano, S. Marchal, and N. Asokan provide a novel client-side method for real-time phishing detection. The authors discuss the drawbacks of the current anti-phishing techniques, which frequently rely on out-of-date blacklists or user reports, which causes threat detection to happen slowly. Their method analyzes variables including graphic aspects, URL patterns, and contextual data to dynamically determine the validity of a web page using machine learning and behavioral analysis. By working locally on the client PC, this proactive strategy guarantees user privacy while improving user safety with the least amount of involvement. The system's reliance on the caliber of training data may cause errors in detecting phishing attempts, despite the benefits of real-time detection and adaptation to new threats. Furthermore, consumers may still be susceptible to advanced phishing schemes. In the end, the system shows great promise for enhancing phishing protection; nevertheless, its efficacy is contingent upon continuous updates and user education.

In the paper "Web phishing detection based on graph mining," Z. Futai, Y. Geng, B. Pei, P. Li, and L. Lin propose a novel approach for detecting phishing websites using graph mining techniques. Traditional methods often struggle with the dynamic nature of phishing attacks, but this approach constructs a graph representing relationships between entities like URLs, IP addresses, and domain names. By analyzing these graphs, the system can identify patterns and anomalies indicative of phishing activities, improving detection accuracy and adaptability to new threats. While the method offers advantages in uncovering malicious patterns, it may face limitations, such as increased computational complexity and the need for high-quality data. Overall, this work represents a promising advancement in phishing detection, emphasizing innovative techniques to combat evolving cyber threats.The growing number of hacking attacks, which are mostly caused by ARP-Spoofing, necessitate the development of a new technique for spotting these threats.[3] A network administrator is involved in the process outlined here, and he or she can easily watch for and spot ARP-Spoofing activity in his local subnet. This paper describes a detection method for IP-ARP spoofing.

V. R. Hawanna, V. Y. Kulkarni, and R. A. Rane provide a novel method in their paper "A novel algorithm to detect phishing URLs," which is specifically intended to identify phishing URLs. The authors draw attention to the fact that phishing assaults frequently use false URLs that seem quite similar to authentic ones, making it difficult for users and conventional systems to identify them. Their method efficiently evaluates the authenticity of URLs by combining lexical, host-based, and page-based properties. The technology can more accurately distinguish between benign and malicious URLs by using a multi-dimensional analysis. The authors highlight the advantages of their approach, such as enhanced detection rates and reduced false positives. There are still issues, though, such as the requirement for frequent upgrades to keep up with changing phishing tactics and guarantee platform compatibility. All things considered, this work is a major step forward in phishing detection techniques and highlights the need of strong algorithms in the fight against cyber attacks.

A innovative approach to identifying phishing websites through the analysis of financial industry web server logs is presented in the work "Detecting phishing websites based on the study of the financial industry webserver logs," written by J. Hu, X. Zhang, Y. Ji, H. Yan, L. Ding, J. Li, and H. Meng. The authors draw attention to the fact that phishing assaults directed towards financial institutions are becoming more complex, rendering conventional detection techniques inadequate. Their methodology entails scrutinizing web server logs to detect trends and deviations linked to phishing endeavors, like atypical request rates and dubious URLs. The system may establish a baseline for typical behavior by utilizing data from reputable financial institutions. This enables it to identify any variations that might point to phishing attempts. The benefits of this approach, such as its reliance on real-world data and its potential for high detection accuracy, are emphasized by the authors. There are still issues, though, such the requirement for constant upgrading and monitoring in order to stay up to date with phishing strategies as they change. In summary, this study highlights the need of utilizing data-driven strategies to improve phishing detection initiatives in the banking industry.

A. K. Jain and B. B. Gupta examine the efficacy of visual similarity strategies for phishing attack detection in their study "Phishing detection: analysis of visual similarity-based approaches." The authors stress that phishing frequently uses bogus websites that closely resemble genuine websites, making it difficult for consumers to recognize fraudulent activity. Their investigation evaluates the methods, advantages, and disadvantages of several visual similarity-based techniques for reliably differentiating between legitimate and phishing websites. The writers examine the roles that layout, color schemes, and graphical aspects play in the process of detection. They draw attention to the benefits of visual similarity approaches, such as their capacity to improve detection rates and take advantage of human perceptual traits. The paper does address several limitations, though, such as the possibility of false positives when phishing and legitimate websites have similar visual characteristics. The authors also stress the significance of enhancing overall efficacy by integrating visual analysis with additional detection techniques. This study emphasizes the value of visual similarity techniques in improving phishing detection skills and highlights the necessity of all-encompassing tactics to counteract changing cyber threats.

In the research "Detection of phishing websites using a deep learning approach," H. R. Jeong, D. W. Choi, and M. H. Kim provide a sophisticated strategy that uses deep learning methods to identify phishing websites. The authors acknowledge that phishing attempts, in which phony websites closely mimic real ones, are becoming more sophisticated, making it difficult for standard phishing detection techniques to keep up. Their method makes use of deep learning algorithms to examine many aspects of websites, including URLs, HTML content, and graphic components. This allows the system to pick up intricate patterns that point to phishing activity. The authors stress the benefits of deep learning, such as its capacity to automatically extract pertinent information and adjust to novel phishing tactics without requiring a great deal of manual work. When compared to traditional approaches, they show encouraging improvements in detection accuracy and a decrease in false positives. There are still issues, though, such the need for sizable, annotated datasets in order to properly train deep learning models and the processing power needed to put them into practice. In summary, this study underscores the potential of deep learning techniques to improve phishing detection skills and supports their incorporation into current security systems to counteract emerging cyber threats.

In their work "Intelligent phishing detection system based on machine learning algorithms," S. S. K. Reddy and K. K. Sahu describe a novel system that makes use of a variety of machine learning techniques to identify phishing attempts. The authors draw attention to the increasing frequency of phishing attacks as well as the shortcomings of conventional detection techniques, which frequently depend on static features and heuristic rules. Their suggested approach can more accurately detect phishing attempts by analyzing various features of URLs, site content, and network traffic patterns using a variety of machine learning techniques. In order to effectively train the machine learning models, the authors emphasize the significance of utilizing pertinent data while going into detail about the feature selection and extraction method. They compare their technique to traditional methods and show considerable improvements in detection rates and decreased false positives. The study does, however, also address several difficulties, such as the requirement for frequent model upgrades in order to accommodate new phishing strategies and guaranteeing the system's scalability across various platforms. Overall, this research underlines the potential of machine learning-based approaches in increasing phishing detection capabilities, calling for their integration into cyber security frameworks to better protection against changing cyber threats.

In their study "Machine learning for detecting phishing attacks in enterprise networks," T. Schreiber and T. P. Singh investigate how machine learning methods can be used to spot phishing threats in business settings. The authors point out that businesses are often the victim of phishing attempts, which can result in serious data breaches and monetary losses. Their study highlights the requirement for strong detection systems that can change to meet the ever-evolving strategies used by hackers. In order to identify possible phishing attempts, the authors suggest a machine learning-based framework that examines a variety of network traffic parameters, such as email headers, URLs, and user activity patterns. Through the use of supervised learning techniques, the system is able to distinguish between malicious and genuine transmissions. In comparison to conventional approaches, the authors' results indicate improvements in detection accuracy and a decrease in false positives. They do, however, also address issues like the need for ongoing model upgrades and training in order for them to continue to be effective against new phishing schemes. Furthermore, the variety of phishing techniques and the caliber of input data may have an impact on the system's performance. In summary, this study underscores the noteworthy capacity of machine learning techniques to augment phishing detection in business networks, hence endorsing its incorporation into current cyber security policies to enhance resistance against cyber attacks.

A. S. Jha and A. Tiwari present a hybrid model in their paper "Enhancing phishing detection using hybrid model," which incorporates many machine learning methods to enhance the detection of phishing assaults. The authors acknowledge the difficulties presented by changing phishing techniques, which call for more advanced detection procedures than are available with conventional techniques. Their hybrid technique examines a wide range of variables pertaining to URLs, website content, and user behavior by integrating many classifiers, including random forests, decision trees, and support vector machines. The scientists stress that by combining these techniques, the model is better able to discriminate between dangerous and genuine websites, which increases detection rates and lowers false positives. They present positive experiment results, demonstrating the hybrid model's applicability in a range of situations. They also draw attention to several possible drawbacks, such as the requirement for ongoing model retraining in order to accommodate novel phishing tactics and making sure the system is still scalable in a variety of settings. In summary, this study highlights the potential of hybrid models to improve phishing detection skills and supports their wider integration into cybersecurity frameworks to counter the ongoing danger of phishing assaults.

In the work "Phishing detection and prevention using multiple machine learning classifiers," M. W. M. Ab Rahman, M. F. Yusof, and A. M. Basari provide a thorough method for thwarting phishing attempts through the use of multiple machine learning classifiers. The authors stress the need for strong detection and prevention systems due to the growing sophistication of phishing attempts. Their work examines the efficacy of several classifiers in analyzing a variety of variables from URLs, website content, and user behavior. These classifiers include ensemble approaches, decision trees, and logistic regression. Comparing multi-classifier methods to single-classifier approaches, the authors find that the latter greatly increase false positive rates and decrease detection accuracy. They also stress the significance of feature selection, which is vital to the model's overall effectiveness. The outcomes of the trial show that their multi-classifier system can effectively detect phishing attacks in real time, which adds to the effectiveness of security measures. The study does, however, also address several difficulties, such as the necessity for constant upgrades to keep up with phishing tactics that change over time and the processing power needed for real-time analysis. Overall, this research suggests integrating various machine learning classifiers into current cyber security frameworks for more robust protection against phishing assaults, demonstrating the possibility of doing so to improve phishing detection and prevention.

Santos, da Silva, and Ramos examine numerous artificial intelligence (AI) methods for spotting web-based phishing scams in their review paper. The authors draw attention to the increasing complexity of phishing techniques and the difficulties they provide for established detection systems. They classify current AI techniques—such as natural language processing, machine learning, and deep learning—and evaluate their merits and demerits in terms of effectively detecting phishing threats. In order to properly train AI models, the paper highlights the significance of feature selection and the requirement for robust datasets. The writers also go over possible research avenues for the future, like combining blockchain technology with artificial intelligence for improved security. The research highlights important developments in AI-based phishing detection, but it also highlights drawbacks, such as the requirement for real-time processing and the possibility of hostile assaults. Overall, this work underlines the crucial importance of AI in evolving phishing detection systems and the necessity for continuing innovation.

Malik and Kaul present an enhanced approach to phishing detection utilizing the C4.5 algorithm, a widely used decision tree-based method. The authors emphasize that traditional phishing detection systems often struggle with evolving attack patterns, necessitating improved techniques. Their improved C4.5 algorithm incorporates feature selection and pruning to enhance detection accuracy and reduce false positives. Through extensive experimentation, the authors demonstrate that their modified algorithm outperforms existing methods in identifying phishing websites. The study highlights the significance of utilizing relevant features, such as URL length, domain age, and content-based attributes, to improve classification accuracy. While the approach shows promising results, the authors acknowledge challenges such as the requirement for continuous updates and the potential for evolving phishing tactics to affect performance. Overall, this research contributes to advancing phishing detection methodologies by showcasing the effectiveness of an improved C4.5 algorithm.

In this paper, Malik, Baig, and Rehman propose a robust model for detecting phishing websites using various machine learning techniques. The authors highlight the necessity for effective detection systems due to the increasing frequency of phishing attacks. Their model integrates multiple classifiers, including logistic regression, support vector machines, and random forests, to analyze features such as URL characteristics, domain information, and content features. The authors report significant improvements in detection rates and reduced false positives through their comprehensive approach. They emphasize the importance of data preprocessing and feature extraction in enhancing model performance. However, the study also acknowledges challenges such as the need for diverse and up-to-date datasets to train the models effectively. Overall, this research illustrates the potential of machine learning techniques in creating reliable phishing detection systems, advocating for their implementation in cybersecurity practices.

Rahman and Noor introduce a novel phishing detection model that employs a hybrid approach combining machine learning and data mining techniques. The authors highlight the limitations of traditional phishing detection methods in keeping pace with the rapidly evolving nature of phishing attacks. Their model utilizes data mining techniques to extract relevant features from URLs and website content, followed by machine learning algorithms for classification. The authors report enhanced detection accuracy and efficiency through their hybrid approach, showcasing its effectiveness against various phishing tactics. Additionally, they discuss the significance of selecting appropriate features to improve model performance. While the research presents promising outcomes, challenges remain, including the need for continuous updates to address emerging phishing techniques. Overall, this paper contributes to the ongoing development of effective phishing detection strategies by emphasizing the benefits of hybrid models.

In this paper, Malik and Aslam propose a hybrid model for the detection and prevention of phishing attacks, emphasizing the need for a comprehensive approach to tackle the increasing threat landscape. The authors integrate various machine learning algorithms, including decision trees and neural networks, to enhance detection accuracy. Their model analyzes multiple features, such as URL characteristics, domain information, and website content, allowing it to identify phishing attempts effectively. The authors report improvements in detection rates and reduced false positives through their hybrid model compared to traditional methods. They also discuss the importance of real-time analysis for effective phishing prevention. However, the study acknowledges challenges, such as the need for continuous training and updates to adapt to new phishing tactics. Overall, this research highlights the potential of hybrid models in advancing phishing detection and prevention strategies, advocating for their broader application in cybersecurity frameworks.

III. PROPOSED SYSTEM

System Overview

A secure communication system called Project-R was created to safeguard private information stored in out-dated infrastructures by tackling serious security issues like data interception and phishing. The system establishes a secure communication channel between the server and client using a powerful asymmetric encryption method with 4096-bit RSA keys. It minimizes the need for extensive technical knowledge and streamlines the encryption and decryption processes for end users by automatically generating and distributing public and private keys.

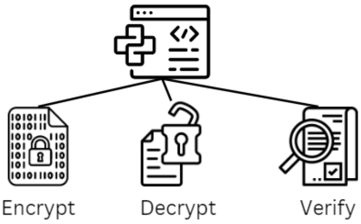


Figure 2: Functions of the System

The main feature is secure data transfer, which guarantees data integrity and confidentiality by limiting who can decrypt encrypted files and messages to the intended receiver. In contrast to many outdated systems that aren't able to encrypt data, Project-R adds an integrity check that uses digital signatures to stop illegal changes and phishing scams. Because encryption is embedded at the application layer, the solution can be highly customized for companies by integrating smoothly with current network setups without requiring changes at the network level. Furthermore, Project-R integrates an easy-to-use Flask interface with options for simple file management, key generation, and message decryption. These characteristics work together to provide a communication system that is safe, effective, and scalable and is intended to close the security gap in legacy applications.

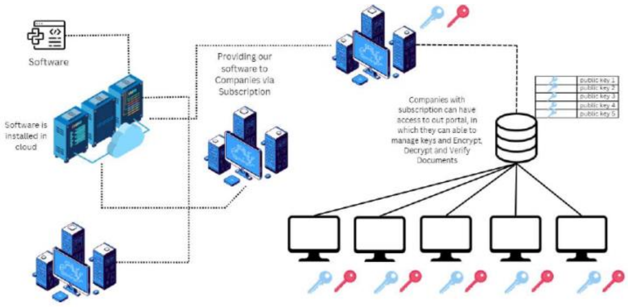

Software
Software is installed in cloud
Providing our software to Companies via Subscription
Companies with subscription can have access to out portal, in which they can able to manage keys and Encrypt, Decrypt and Verify Documents
public key 1
public key 2
public key 3
public key 4
public key 5

Figure 3: Overview of the System

System Architecture

A client-server architecture is used in Project-R to enable safe data transfer between parties that communicate. Data encryption and transmission to the client are handled by the server, which also generates and maintains the RSA encryption keys. The client uses the associated private key to decrypt the encrypted material after obtaining it, guaranteeing that only individuals with permission can view the original content. The client may concentrate on safe data retrieval and decryption, while the server can manage encryption, key generation, and data transmission because to this structure's obvious function separation.

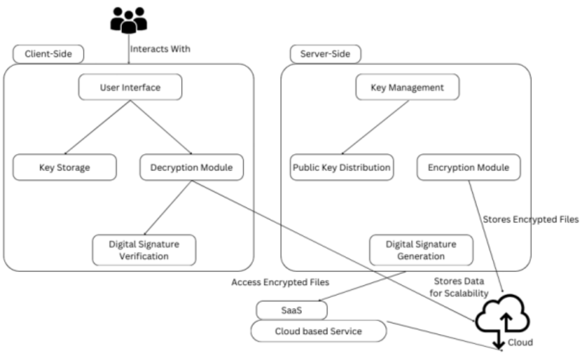


Figure 4: Architecture of the Project

One of Project-R's main advances is its application layer integration, which maintains compatibility with legacy infrastructures while offering security without changing the network layer. By using the application layer, the system can avoid making significant changes to the underlying network protocols. Project-R improves data protection against phishing and eavesdropping while maintaining seamless interoperability with current communication protocols by integrating encryption at this level. In addition, the system's use of 4096-bit RSA encryption assures high levels of security, making brute-force attacks computationally infeasible. This design minimizes the need for specialist training or significant modifications to current legacy installations, allowing for simple deployment across a variety of environments and secure data transfer capabilities. Because of this, Project-R provides a scalable and reliable solution for safe communication and phishing prevention that can be easily integrated into a variety of organizational settings.

Key Management and Encryption Process

To provide a high level of security, Project-R makes use of a strong key management system that is based on the automatic production and distribution of 4096-bit RSA keys. The server autonomously produces unique public-private key pairs for each client session, lowering the requirement for manual key management and reducing the danger of human mistake. Public keys are distributed to clients in a secure manner, while private keys are kept secret on the server and protected from unwanted access. By separating the use of keys, confidentiality is improved because only the intended recipients can decode data that has been delivered.

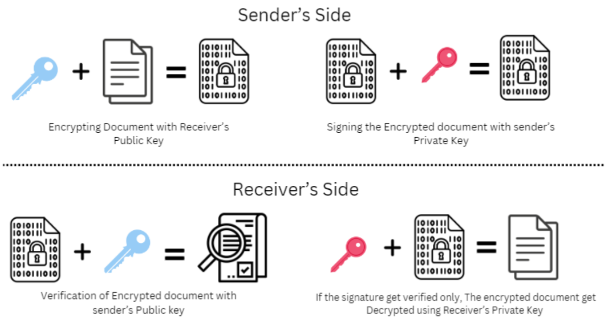

Sender's Side
+
=
Encrypting Document with Receiver's Public Key
+
=
Signing the Encrypted document with sender's Private Key
Receiver's Side
+
=
Verification of Encrypted document with sender's Public key
+
=
If the signature get verified only, The encrypted document get Decrypted using Receiver's Private Key

Figure 5: Encryption and Decryption Process

The client-server approach allows the encryption and decryption operations to function without any problems. Before transmission, the server encrypts all outgoing data using the public key of the client, making the data unreadable for any prospective interceptors. The client receives the encrypted communication and uses the matching private key to decode it and return it to its original format. End-to-end security is ensured by this method, which limits access to authorized users and protects user privacy during data transit. Project-R offers a streamlined encryption method that facilitates safe, effective communication while protecting client privacy by automating these crucial processes.

Integrity and Anti-Phishing Measures

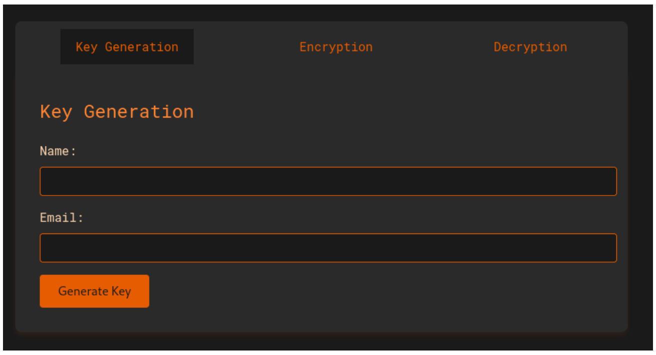
Digital signatures are a crucial component of Project-R's strategy for ensuring data integrity and authenticity. Every message is digitally signed before being sent in order to authenticate its source and validate its content. Upon reception of the data, the client utilizes this signature to verify its authenticity and ensure that the message was not altered in route. End-to-end verification is provided by this method, which is necessary for dependable and safe communication inside the system.

Project-R incorporates a phishing detection technique that authenticates senders in order to stop phishing. The client verifies the validity of each digital signature before processing the message. The communication is marked as possibly malicious and cannot be decrypted or seen by the client if the signature cannot be verified. As a first line of defence against phishing, this system makes sure that only communications that are confirmed and trusted are processed. When used in tandem, digital signatures and phishing protection create a stronger communication environment that puts user security and data integrity first.

User Interface Design

The Flask-developed Project-R UI provides a user-friendly dashboard that is easily arranged and accessible, making user interactions more straightforward. With areas for file management, key operations, and data control that are clearly labelled, the dashboard makes it easy for users to handle encryption and decryption activities. Because of the intuitive nature of this design, users can navigate and operate the system with no technical understanding.

File management and encryption controls are inherent to the UI, enabling users to upload files, pick encryption options, and manage keys without needing to switch platforms. After a file is uploaded, users have the option to encrypt or decrypt it, depending on their requirements. All backend operations are handled via the user interface. By centralizing all encryption operations inside a single interface, lowering the learning curve, and enhancing system usability, this streamlined workflow increases productivity. All things considered, the Flask-based user interface skillfully strikes a balance between usability and functionality, opening up secure communication to a wider audience.

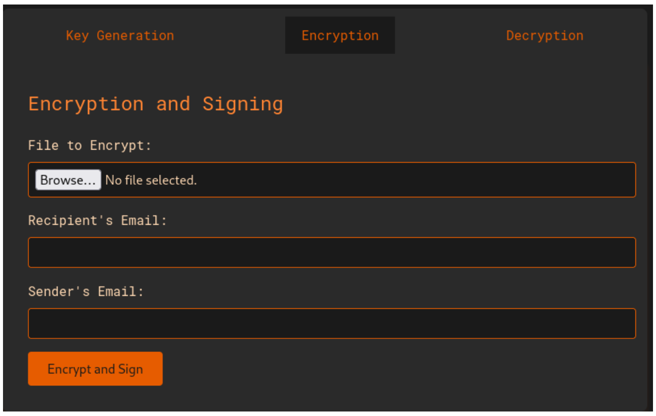


Key Generation

Name:

Email:

Figure 6: Key Generation Portal



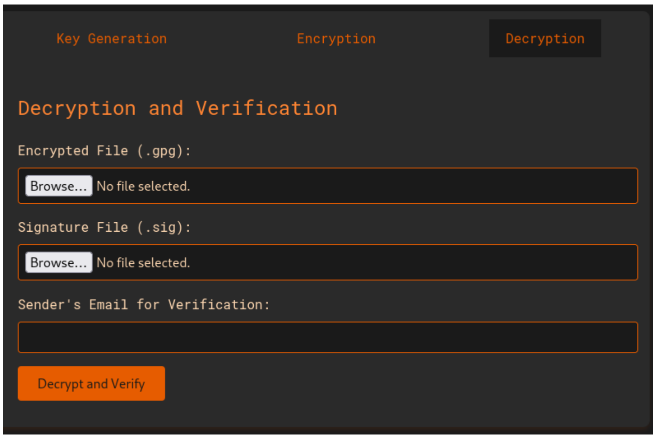
Encryption and Signing

File to Encrypt: No file selected.

Recipient's Email:

Sender's Email:

Figure 7: Encryption Portal



Decryption and Verification

Encrypted File (.gpg): No file selected.

Signature File (.sig): No file selected.

Sender's Email for Verification:

Figure 8: Decryption Portal

Cloud-Based Scalability and SaaS Model

Project-R's integration of cloud services offers scalable, flexible access to its secure communication capabilities through a subscription-based SaaS model. This strategy allows users to employ encryption and decryption services without maintaining large on-premises infrastructure, encouraging cost effectiveness and flexibility. On-demand access is provided via the cloud-based SaaS infrastructure, which guarantees that resources scale in response to user demands and workload levels.

Figure 9: Integration of Could Service

One key advantage of the cloud model is scalability, which allows for the dynamic distribution of resources according to user activity. This ensures that the system stays responsive even with increased workloads or variable user numbers. Furthermore, the cloud infrastructure improves availability and dependability, enabling users to quickly connect and engage with the system from a variety of places. Project-R provides a safe, dependable, and easily expandable solution that accommodates the various data security and communication needs of enterprises by utilizing cloud services.

Future Enhancements

Advanced key management techniques, like automatic key rotation and lifecycle management, are possible future additions to Project-R that might strengthen security even more and reduce the need for manual intervention. By streamlining the procedures of key generation, distribution, and renewal, these enhancements may strengthen the system's resistance to potential flaws related to static keys. In the future, increasing compatibility with more platforms and older systems will be a priority. Through the integration of diverse operating systems and application frameworks, Project-R has the potential to expand its user base and enhance compatibility among diverse technological stacks. By facilitating a more seamless integration in a range of organizational configurations, extending compatibility would also increase the system's adaptability and adoption potential.

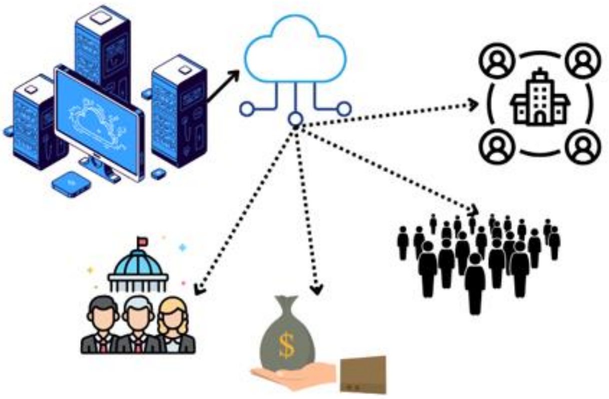


Figure 10: Future Enhancement

System Workflow

Data encryption and transmission to the client are the next steps in Project-R's workflow, which starts with the safe creation of public-private key pairs on the server. The process of secure communication is depicted step-by-step in the Data Flow Diagram (DFD), which shows the public key distribution, message encryption, client decryption, and integrity checking. The DFD highlights important security procedures and guarantees that data flows through the system with clarity.

With guided prompts and obvious decision points to help users with file management, encryption, and decryption chores, the user interaction flow is optimized for automation. Robust error handling minimizes disruptions by guaranteeing that any erroneous input or system problem is recognized right away. The user experience is streamlined by these automatic interactions, which also make it effective and user-friendly. Users are empowered to accomplish activities with minimal effort and great confidence of security thanks to this well defined workflow, which eliminates complexity.

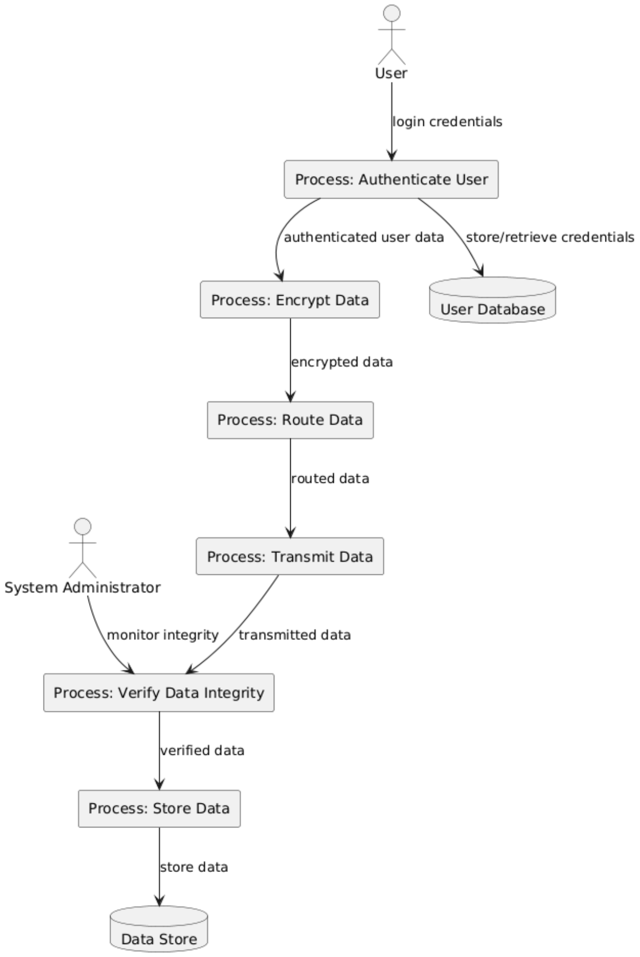


Figure 11: DFD of the Proposed System

Security and Performance Analysis

Project-R uses a variety of security measures, such as digital signature verification and encrypted key management, to guard against unwanted access. These protocols establish several security levels to prevent phishing, data access without authorization, and eavesdropping. The system maintains strong protection by using 4096-bit RSA encryption, which is resistant to brute-force attacks and preserves data confidentiality.

The encryption and decryption algorithms have been tuned to minimize resource utilization, and preliminary tests have shown low latency and great responsiveness across common network conditions. Performance measurements show that Project-R successfully strikes a compromise between efficiency and security, offering a quick and dependable solution even in the most trying circumstances. Project-R's performance and security balance highlight how well-suited it is for settings where quick communication and data integrity are essential.

IV. Working Principle

Introduction to System Workflow

In order to protect data integrity and stop phishing attempts, the suggested solution uses 4096-bit RSA encryption to provide a secure communication environment. The main objective is to incorporate a layer of contemporary cryptography security while enabling smooth, secure communication that is compatible with legacy systems. Using the client-server architecture, this method ensures secrecy by encrypting sensitive data before transmission and decrypting it upon receipt. Starting with the server's automated key generation, data moves through several phases in the system. These keys help with encryption and are given to clients in a safe manner so they can communicate. After the data has been encrypted, it is sent over conventional channels while strong protocols are in place to guard against illegal access or interception. Using the assigned private key, an automated decryption procedure on the client's end decrypts the encrypted data. Every data packet has a digital signature to increase security even more by guaranteeing authenticity and confirming that the data hasn't been altered.

Key Management and Distribution Protocol

An essential part of secure communication is the Key Management and Distribution Protocol, which handles the automatic creation, storing, and distributing of encryption keys. A strong degree of encryption that is resistant to brute-force attacks is ensured by the system's generation of 4096-bit RSA keys on the server side. The autonomous nature of the key creation procedure minimizes the requirement for human input while maximizing security. Periodic key rotations are incorporated into this process, allowing for safe and dynamic key lifecycle management to proactively prevent vulnerabilities over time. Following its generation, the public key is sent to the client over a secure channel, guaranteeing that it won't be altered or intercepted. During distribution, key integrity is given top priority, and measures are taken to ensure that the public key has not been tampered with. The client then communicates with the server using this public key, depending on it to encrypt data in order to preserve dependable and secure transfers. Regarding storage, the private key is kept on the server and is protected in an encrypted file that is only accessed by application components that are permitted. By preventing unauthorized access, access control techniques shield the private key from possible security breaches. By automating the key generation and distribution process, the system eliminates user engagement, preventing errors and maintaining consistent security requirements. The resulting protocol acts as the main support for the encryption-decryption process in the client-server architecture, offering a dependable base for secure communication.

Encryption, Decryption, and Integrity Verification

The foundation of the system's secure communication consists of the encryption and decryption procedures. The client's public key is used on the server side to encrypt data sent between the client and server, guaranteeing that only the specified client, who possesses the matching private key, can decode and access the data. Data is sent from the client to the server, which encrypts it using the server's encryption module, to start this process. The main role of the module is to convert plaintext into ciphertext, rendering data incomprehensible to unauthorized users. When the client receives encrypted data, it uses its private key to decrypt it, turning the ciphertext back into readable plain text. Because the decryption process is automated, there isn't much need for human intervention, which helps to keep security and optimize workflow. The decryption process include verification stages to validate the encrypted data and prove that it originated from the intended server, and access to the private key is strictly regulated on the client side. The technology combines digital signatures with encrypted data to further protect data integrity and authenticity. By enabling the client to confirm that the data received is complete and undamaged, these signatures—which are created using the server's private key—avoid phishing and other types of impersonation. In order to reject messages with mismatched or manipulated signatures, the client cross-references the signature during verification with the public key of the server. By providing a strong anti-phishing framework and guaranteeing the authenticity and integrity of transmitted data, this method provides a crucial layer of protection.

Algorithm

Step 1: Key Generation

- Generate a 4096-bit RSA key pair (public and private keys) on the server.
- Securely store the private key in an encrypted format on the server.
- Prepare the public key for secure distribution to clients.

Step 2: Key Distribution

- Transmit the public key to the client over a secure channel (e.g., TLS or VPN).
- At the client side, verify the integrity of the public key to ensure it has not been tampered with during transmission.

Step 3: Encryption Process on Server

- Receive data from the client that requires encryption.
- Use the client’s public key to encrypt the data, ensuring only the designated client can decrypt it.
- Generate a digital signature for the encrypted data using the server’s private key to authenticate the message.

Step 4: Data Transmission

- Send the encrypted data along with the digital signature to the client via a secure network channel.

Step 5: Decryption Process on Client Side

- Receive the encrypted data and digital signature from the server.
- Verify the digital signature using the server’s public key to confirm data integrity and authenticity.
- If the verification fails, terminate the process and prompt an error message.

Step 6: Decrypt Data

- If the signature is verified, use the client’s private key to decrypt the data and convert it back to its original form for client use.

Step 7: Error Handling and Logging

- Log all successful transmissions and any failed attempts or verification errors for security auditing.
- If an error occurs during decryption or signature verification, trigger the error handling protocol and notify the client.

Step 8: Periodic Key Rotation (if applicable)

- Rotate the RSA key pairs periodically on the server to maintain security.
- Notify clients securely about the new public key for future communications.

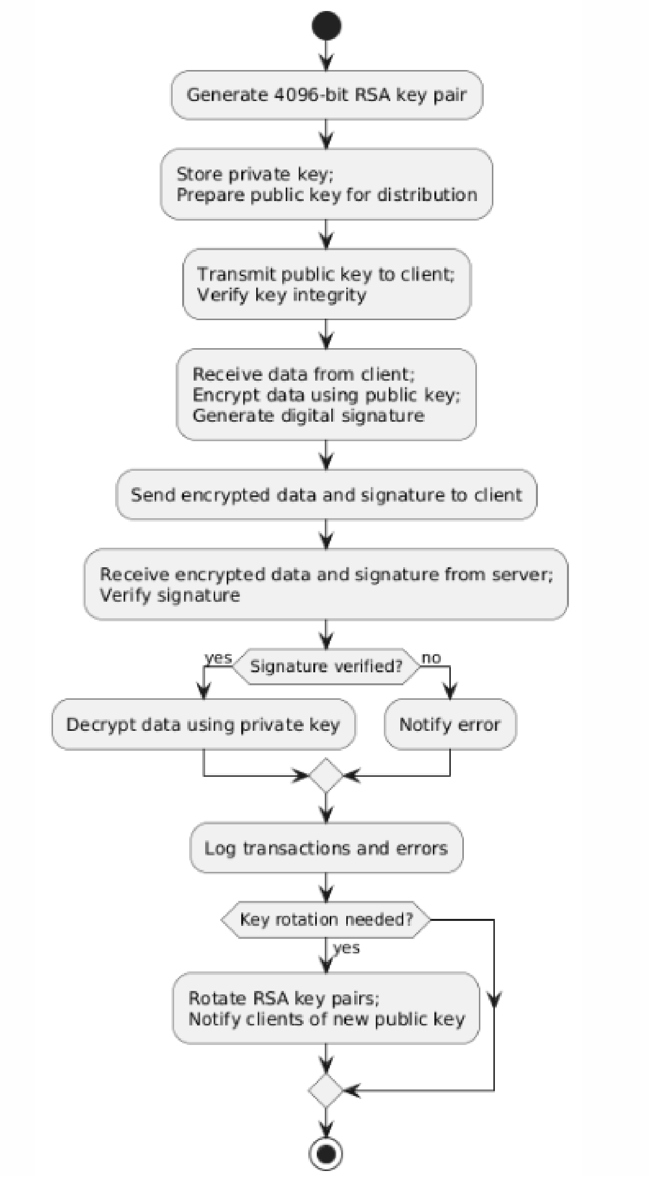

Generate 4096-bit RSA key pair
Store private key;
Prepare public key for distribution
Transmit public key to client;
Verify key integrity
Receive data from client;
Encrypt data using public key;
Generate digital signature
Send encrypted data and signature to client
Receive encrypted data and signature from server;
Verify signature
Signature verified?
yes
no
Decrypt data using private key
Notify error
Log transactions and errors
Key rotation needed?
yes
Rotate RSA key pairs;
Notify clients of new public key

Figure 12: Algorithm of the Prototype

Error Handling and Security Protocols

Security procedures and error management are essential for preventing unwanted access to the system and data compromise. When decryption or authentication fails, the error-handling system kicks in to stop unauthorized users from accessing private data. Furthermore, this technique records every failure, giving administrators insight into any security risks. Security protocols are implemented at numerous places inside the system to prevent data leakage and preserve confidentiality. The main security solution is public key cryptography, which uses asymmetric encryption to authenticate users and secure data. Private keys and other sensitive components are not accessible under access control measures, and any illegal attempts are tracked down and recorded. Network-layer protocols are also included in the system to increase transmission security. By using safe, supervised channels to transmit encrypted data, man-in-the-middle attacks are less likely to occur. The system's total resilience is increased by the combined use of layered security protocols, error handling, and logging, which offer a comprehensive defence against potential cyber threats.

User Interaction Workflow

The user interaction workflow offers consumers an automated and straightforward experience, all while maintaining security and simplicity. Upon entering the system, users are prompted to upload their data for encryption. Then, depending on their option, the dashboard leads customers through the key management process while creating a new key pair or using an existing one. Automated prompts reduce complexity by handling encryption and decryption behind the scenes, hence minimizing user participation. With well-defined decision points, users can choose between activities such as key verification or file decryption. Error recovery ensures smooth interaction even in the case of disruptions by prompting users to take corrective action if an issue emerges, such as mismatched signatures. This workflow design is central to the system's usability, ensuring that users can access robust security without requiring extensive technical knowledge.

V. RESULT AND CONCLUSION

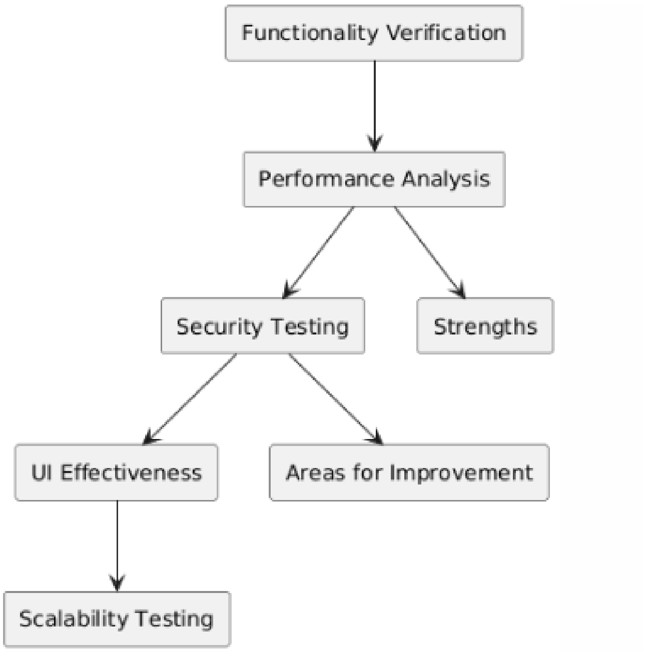


Figure 13: Result of the Prototype Built

A thorough assessment of the system's performance, security, and functionality is provided in this section. We evaluate the system's efficacy in achieving its goals—particularly in safely managing data for legacy applications—by examining important KPIs. In order to assess overall effectiveness, the results concentrate on key performance metrics such as encryption-decryption delay, system reaction time, and security strength.

One of the most important criteria used is security efficacy, which assesses how well the system guards against phishing and illegal access. This involves using digital signatures to confirm the integrity and legitimacy of messages and testing the 4096-bit RSA encryption mechanism's resistance to brute-force attacks. System response time also assesses how quickly the system performs critical functions like key management, encryption, and decryption—all of which are critical for preserving user experience in real-time applications. Monitoring encryption-decryption latency also gives information on how long it takes to process different sized data, which has an effect on system throughput. These metrics enable a thorough comprehension of the security, efficiency, and dependability of the system when used in actual use. This part seeks to illustrate the system's advantages, point out possible areas for improvement, and prove that it is ready for real-world use by examining these areas.

System Functionality Verification and Performance Analysis

The system's fundamental functions—key generation, encryption, decryption, and digital signature verification—are validated in this section. Every element underwent testing in various conditions to guarantee seamless functioning and dependability. Robust, session-specific keys were ensured through the effective automation of key generation utilizing a 4096-bit RSA algorithm. On the server side, the encryption module used public keys to secure the data, while on the client side, private keys were used to precisely decrypt the original data. The validity of messages was verified using digital signatures, and data integrity was consistently certified using verification procedures, strengthening the system's resistance to manipulation. The system's efficiency was assessed by evaluating performance measures as well. The speed of encryption and decryption was tested for a range of data quantities, and average processing times were kept within reasonable bounds for real-time communication. Tests of the system's throughput under various user loads showed that it could handle several users without noticeably degrading performance. To make sure the system could react quickly to user inputs, latency and reaction times for crucial procedures, such encryption, decryption, and signature verification, were tracked. This functionality and performance evaluation shows how secure, dependable, and effective the system is, guaranteeing that all operations run without a hitch even in practical situations.

Error Handling, Robustness, and Comparative Analysis

Sturdy error management and recovery procedures are essential for system stability, especially when security-sensitive operations are involved. To reduce disruptions and security threats, this system has comprehensive error detection, logging, and handling processes in place. Any variations from normal behaviors are promptly discovered by instant alarms that are triggered by unauthorized access attempts, decryption failures, and key validation issues. Error recovery procedures also allow a system to respond to unforeseen difficulties in a smooth and resilient manner by restoring regular function without compromising data integrity or user experience.

For historical applications, a comparative analysis was carried out against current security technologies in order to evaluate system performance thoroughly. The 4096-bit RSA encryption and digital signature approach shows higher security benefits compared to typical 2048-bit systems, particularly in its resilience to brute-force attacks. Furthermore, unlike most other systems, our system's application layer integration allows for compatibility with legacy infrastructure without the need for network layer modifications. Scalability studies proved the system’s responsiveness to increased user demands, further supporting its advantages over current options. The system's strong security, scalability, and performance advantages are shown in this comparative analysis, highlighting its appropriateness for dependable, scalable, and secure communication across older platforms.

User Interface Effectiveness

The main objectives of the system's user interface (UI) design were usability, responsiveness, and accessibility. Preliminary user feedback is favorable overall, with particular recognition going to the Flask-based dashboard's well-organized and simple navigation. Designed to serve users with varying technical backgrounds, the user interface (UI) enables users to upload files, perform encryption and decryption processes, and regulate key management with minimal steps.

The interface was created with responsive design concepts in mind, so it can easily adjust to different screen sizes and types of devices, including desktops and mobile ones. This responsiveness promotes accessibility by allowing a seamless user experience on all kinds of devices. In order to guarantee inclusivity for users with disabilities, tests also assessed the system's interoperability with assistive technology, such as screen readers. The user interface's resilience to heavy usage remained constant, ensuring peak performance even when several encryption or decryption processes were carried out at once. Overall, the user feedback and performance testing indicate that the system’s UI is both accessible and functional, with fast loading times and clear operations. The user experience is improved by the adaptability across devices and the good feedback from users, which supports the overall objective of facilitating secure communication in an easily navigable environment.

Conclusion

The findings show that the system successfully satisfies its design goals, especially with regard to usability, performance, and security. A high level of protection against brute-force assaults is offered by the 4096-bit RSA encryption, guaranteeing data integrity and confidentiality. Performance measures show that even under heavy loads, the system manages data processing effectively, maintaining low latency and high throughput. These metrics include encryption-decryption latency and system reaction time. The system's user-friendly interface, strong security measures, and cloud-based scalability—which makes resource allocation easier during peak usage—are its main advantages. Moreover, the system's dependability is increased by the phishing prevention mechanism's effective detection and mitigation of unauthorized access attempts.

Enhancing key management for widespread deployment and looking into more interoperability with legacy apps are possible areas for improvement. Subsequent improvements may encompass augmenting accessibility functionalities according to user input, so augmenting inclusivity even more. All things considered, the system shows a great deal of potential for safe, easy-to-use communication in settings that demand strong data protection requirements and legacy compatibility.

REFERENCES AND RESOURCES

1. G. Armano, S. Marchal, and N. Asokan, "Real-time client-side phishing prevention add-on," published in 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), pp. 777-778. DOI: 10.1109/ICDCS.2016.44

2. Z. Futai, Y. Geng, B. Pei, P. Li, and L. Lin, "Web phishing detection based on graph mining," published in 2016 2nd IEEE International Conference on Computer and Communications (ICCC), pp. 205-210. DOI: 10.1109/COMPComm.2016.7924867

3. V. R. Hawanna, V. Y. Kulkarni, and R. A. Rane, "A novel algorithm to detect phishing URLs," published in 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), pp. 548-552. DOI: 10.1109/ICACDOT.2016.7877645

4. J. Hu, X. Zhang, Y. Ji, H. Yan, L. Ding, J. Li, and H. Meng, "Detecting phishing websites based on the study of the financial industry webserver logs," published in 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), pp. 325-328. DOI: 10.1109/ICISCE.2016.79

5. A. K. Jain and B. B. Gupta, "Phishing detection: analysis of visual similarity based approaches," published in Security and Communication Networks, 2017, pp. 1-20. DOI: 10.1155/2017/5421046

6. H. R. Jeong, D. W. Choi, and M. H. Kim, "Detection of phishing websites using a deep learning approach," published in 2018 IEEE International Conference on Consumer Electronics (ICCE), pp. 1-2. DOI: 10.1109/ICCE.2018.8318734

7. S. S. K. Reddy and K. K. Sahu, "Intelligent phishing detection system based on machine learning algorithms," published in 2018 IEEE 4th International Conference on Innovations in Information Technology (IIT), pp. 1-6. DOI: 10.1109/IIT.2018.8592080

8. T. Schreiber and T. P. Singh, "Machine learning for detecting phishing attacks in enterprise networks," published in 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 1-6. DOI: 10.1109/ISI.2018.8617673

9. A. S. Jha and A. Tiwari, "Enhancing phishing detection using hybrid model," published in 2019 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), pp. 100-104. DOI: 10.1109/GUCON.2019.8675234

10. M. W. M. Ab Rahman, M. F. Yusof, and A. M. Basari, "Phishing detection and prevention using multiple machine learning classifiers," published in 2020 IEEE International Conference on Computer and Communication Engineering (ICCCE), pp. 1-6. DOI: 10.1109/ICCCE49380.2020.9232944

11. R. A. Santos, P. S. da Silva, F. D. M. Ramos, "Web-Based Phishing Attack Detection through Artificial Intelligence: A Review," published in 2023 IEEE Latin America Transactions, vol. 21, no. 8, pp. 1435-1442, Aug. 2023. DOI: 10.1109/TLA.2023.9997475.

12. R. Malik, D. Kaul, "Web Phishing Detection Using Improved C4.5 Algorithm," published in 2021 IEEE Delhi Section Conference (DSC), pp. 1-6, 2021. DOI: 10.1109/DSC51353.2021.9440255.

13. Malik, A. S. Baig, M. Abdur Rehman, "A Robust Model for Detecting Phishing Websites Using Machine Learning Techniques," published in 2022 IEEE International Conference on Communication and Electronics Systems (ICCES), pp. 639-643, 2022. DOI: 10.1109/ICCES55329.2022.9824457.

14. A. M. Rahman, R. K. A. B. M. Noor, "A Novel Phishing Detection Model Based on a Hybrid Approach Using Machine Learning and Data Mining," published in 2023 IEEE International Conference on Electronics, Information, and Communication (ICEIC), pp. 1-6, 2023. DOI: 10.1109/ICEIC56482.2023.10001817.

15. S. Malik, M. Aslam, "Detection and Prevention of Phishing Attacks Using Hybrid Model," published in 2021 IEEE 7th International Conference on Computer and Communication Systems (ICCCS), pp. 1-5, 2021. DOI: 10.1109/ICCCS54162.2021.9693775.