PGP Based RSA Encryption for MFA in Web Applications

MSc Research Project
Cyber Security

Ragul Murugesan
Student ID: 23154870

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……Ragul Murugesan…………………………………………

Student ID: x23154870……………………………………………………………………………

Programme: …MSc Cyber Security……………………………… **Year:** …2024………….

Module: …MSc Research Project………………………………………………………….………

Supervisor: ……… Michael Prior…………………………………………………………….………

Submission Due Date: ………12th December 2024……………………………………………………….………

Project Title: PGP Based RSA Encryption for MFA in Web Applications………………

Word Count: ………6377…………… **Page Count**……………19….……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Ragul Murugesan ……………………………………………………………………………………………………

Date: 10th December 2024 ……………………………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

PGP Based RSA Encryption for MFA in Web Applications

Ragul Murugesan
X23154870

Abstract

This thesis explores a novel approach to two-factor authentication (2FA) through encryption/decryption-based methodologies, utilizing the robustness of public key cryptography to secure user authentication. Traditional 2FA uses methodologies like SMS or authenticator applications, which are variably prone to attack via SIM swapping, interception, and other forms of compromise. It relies on asymmetric encryption that individual pairs generate their own personal public/private key sets, ensuring that only the legitimate user can decrypt the authentication challenge proposed by the server. This implies that one-time passwords are replaced by means of a PGP-based challenge-response process, enhancing the underlying security without even leaving a slim chance for other vulnerable channels. Such techniques include key generation, message signing, and verification proving the above system to be resilient and another method apart from the usual or standard forms of the 2FA methods. This gives strong protection against phishing and replay attacks. Hence, this encryption-based 2FA mechanism is preferred in applications that require high security and user data privacy.

Keywords: Encryption, Decryption, OpenPGP, 2FA, Authentication, Comparing.

1 Introduction

This has been evident through increased interest in secure methods of authentication. Among effective solutions to encrypt the data, one of the robust ways to deliver measures for the security of data transmission and authentication of users' identity by using asymmetric encryption is Pretty Good Privacy. Traditional 2FA relies upon exogenous communication channels, such as SMS or email, which introduce vulnerabilities to interception and spoofing attacks that may compromise user accounts. This is where PGP-based encryption overcomes the security shortcomings of others to become a more secure alternative to traditional 2FA. This study investigates the effectiveness of PGP as an authentication tool, focusing on methodologies to generate, encrypt, and verify credentials securely with a unique passphrase that is only used as decrypt verification which then revels the data. By implementing a challenge-response mechanism that verifies possession of a private key without exposing it, this approach minimizes reliance on vulnerable external channels. As this the start of a new implementation as an optional 2FA model that can be used to authenticate users of such services in a secure way rather then the traditional methods that used in regular basis. The strength of using PGP is that it is secure in most possible way cause the output is always encrypted. This research aims to demonstrate the superiority of encryption based authentication in safeguarding user access and data integrity.

1.1 Pretty Good Privacy

Pretty Good Privacy (PGP) is a widely recognized encryption program that seamlessly provides cryptographic privacy and authentication for securing communication and data. It uses a hybrid approach of symmetric and asymmetric cryptography for message encryption, decryption, and signing. The underlying model of PGP is based on a public-private key pair in which everyone has the public key for encryption, while the private one remains secret and known to the user for decryption or signing. Besides this, PGP also allows checking on the integrity of data by allowing verification of the authenticity of messages and files through digital signatures. It is the thesis that, considering all the shortcomings of traditional password methods and one-time passwords, proposes a new solution using the capabilities provided by PGP. The paper continues with a PGP-based challenge-response authentication model wherein the authenticity of the user gets established through the user's private key signing of a challenge received from the server, matched by the server via the user's public key without the transmission of credentials. The idea will further ensure security: no password storing, less possibility of phishing, keeping the private keys with the users themselves. This concept enhances the added robustness to the usability of the authentication systems. This methodology uses the same general approach of asymmetric and symmetric cryptography to perform in-transit and at-rest encryption of both emails and files. Additionally, it provides a way to maintain authenticity while data is in transit. In encrypting, a passphrase was provided to complement it with added security; hence, in decryption, the private key would not be sufficient. It needs the particular passphrase set up by the user for successful authentication.

1.2 Symmetric Encryption Technology

Symmetric encryption(Encryption — OpenPGP for Application Developers, n.d.) is a foundational element of cryptography, where the same key is used for both encryption and decryption. In the context of PGP, symmetric encryption is used for encrypting large datasets efficiently. Notable characteristics include:

- **Key Sharing Challenge**: One of the primary concern is the secure distribution of the symmetric key between parties. If there is no implementation of secure channels, then interception may occur.
- **Efficiency**: Symmetric encryption is highly efficient making it suitable for encrypting substantial amounts of data with minimal computational overhead. This is ensured with much lower computational overhead, hence much faster processing compared to asymmetric encryption.
- **Usage in PGP**: PGP applies symmetric encryption to wrap the real message or information. For this, a session key is generated temporarily, which is used for the information encryption. Further, asymmetric encryption is performed on this to make it secure while transmission. This approaches efficiency with safety.

1.3 Asymmetric Key Encryption

Asymmetric encryption, a cornerstone of PGP, employs a pair of mathematically related keys a public key for encryption and a private key for decryption. It is the dual key approach that dismisses any thought of sharing sensitive keys are pretty safe indeed. While in symmetric encryption one same key is used by both parties in an asymmetric system private keys remain secret while public keys are shared without caution.

1.3.1 RSA (Rivest-Shamir-Adleman)

Process: (*RSA Encryption*, n.d.)RSA operates by generating key pairs from two large prime numbers, creating a secure encryption-decryption mechanism. Since these are keys computationally infeasible to reverse-engineer, the protection then becomes strong.
Strength: The difficulty of factoring large integers underpins RSA's security, making it a reliable choice for encryption. RSA has good resistance to attack, provided the key size is big enough-for example, 2048 bits or greater.
Application in PGP: RSA is frequently used in PGP for encrypting session keys, ensuring that only the intended recipient can decrypt the data. Its widespread adoption makes it a benchmark for public-key cryptography.

1.3.2 ECC (Elliptic Curve Cryptography)

Process: ECC leverages the properties of elliptic curves over finite fields to produce compact yet highly secure key pairs. This reduces the computational load while maintaining strong encryption.(*OpenPGP: Key Generation (Elliptic Curve)*, n.d.)
Strength: It offers comparable strength like RSA but with much smaller keysizes, which helps in better performance and resource efficiency. For example, a 256-bit key
in ECC would be as good as 3072-bit strength in RSA.
Application in PGP: New implementation of the PGP is more into applying ECC, owing to its efficiency and good security profile that can be performed on constrained-resource systems.

1.3.3 DSA (Digital Signature Algorithm)

Process: DSA focuses on generating and verifying digital signatures to authenticate messages. It further builds up the trust because, in DSA, it is assured that the message comes from a valid source.
Strength: It ensures message integrity and authenticity, providing cryptographic proof that a message has not been altered. It prevents unauthorized changes and provides non-repudiation.
Application in PGP: DSA along with the Secure Hash Algorithm is used for data
encryption and authentication in PGPsystems.It provides detection against message
modifications.

2 Related Work

An Evaluation and Implementation of Pretty Good Privacy (PGP) in Wireless Network Security" focuses on the use of PGP to secure wireless networks. It evaluates how PGP's encryption techniques, including public and symmetric key cryptography, enhance data confidentiality and prevent unauthorized access. This paper discusses PGP's adaptability to different wireless infrastructures and highlights the challenges, like key management, and the security benefits it brings to wireless communication.

(Fan et al., 2024) "Unveiling the Darkness: Analysing Organised Crime on the Wall Street Market Darknet Marketplace Using PGP Public Keys" explores PGP's role in darknet marketplaces, specifically as a method for user authentication and transaction reputation. The paper demonstrates that PGP keys, shared among users on platforms like Wall Street Market, help identify potential organized groups by linking vendors and buyers via shared authentication data. This study supports the understanding of PGP in maintaining privacy, trust, and authentication in anonymous environments, relevant to using PGP for secure and verified user interactions in a login system.

(Raghavan Nair, 2023)"Data Security at Cloud Storage using PGP in conjunction with IPsec VPN" presents a layered security model for cloud storage using PGP and IPsec VPN to secure data at rest and in transit. The research proposes using PGP for encrypting data in the fog computing environment before sending it to the cloud, combining symmetric AES and asymmetric RSA encryption for robust protection. IPsec VPN provides a secure transmission tunnel, enhancing data privacy and integrity. This approach is relevant to PGP-based secure login systems by demonstrating PGP’s strength in data confidentiality and secure transmission across networks.

(Sunil et al., 2023)“Exploring the Use of Pretty Good Privacy (PGP) in Wireless Network Security" discusses the application of PGP in securing wireless communications by encrypting data to prevent eavesdropping and ensuring authentication. PGP’s dual use of public and private keys enables secure data exchange and authentication over Wi-Fi, providing confidentiality and integrity. The study highlights PGP’s effectiveness in protecting data in transit across wireless networks, which complements secure user login systems that rely on PGP's authentication and encryption mechanisms to prevent unauthorized access and data breaches.

(Syed et al., 2023)"Information Security using GNU Privacy Guard" explores the use of GNU Privacy Guard (GNUPG), an open-source implementation of PGP, for secure data communication. GNUPG leverages both symmetric and asymmetric encryption to ensure data confidentiality and integrity, with applications in encrypting files, emails, and creating digital signatures. The paper details GNUPG’s key management, highlighting challenges like key verification through a web of trust.

(Ibrokhimov et al., 2019)"Multi-Factor Authentication in Cyber Physical System: A State of Art Survey" reviews the evolution from single-factor to multi-factor authentication (MFA) and highlights MFA's role in enhancing cybersecurity. The paper discusses various MFA techniques, including biometric verification, threshold cryptography, and cloud-based authentication, and emphasizes that combining multiple independent authentication factors provides stronger security against unauthorized access. This aligns with PGP-related authentication by underscoring the importance of multi-layered security approaches, especially in systems requiring secure and verified user identity management.

(Saxena et al., 2021) ProtonMail Advance Encryption and Security explores ProtonMail's encryption protocols, focusing on end-to-end encryption and zero-access encryption, where even ProtonMail servers cannot access user data. The system relies on public-private key cryptography to secure emails, aligning with PGP principles to ensure only intended recipients can decrypt messages. By using a combination of AES and RSA, ProtonMail emphasizes user privacy, data integrity, and confidentiality, similar to a PGP-based authentication system, where encryption and secure key management are essential for protecting sensitive communication.

(Borradaile et al., 2021)The Motivated Can Encrypt (Even with PGP) studies the adoption of PGP by activists with high motivation due to privacy concerns, despite the usability challenges that typically hinder PGP use. The paper explores long-term PGP use by activists who face potential surveillance threats, finding that those with a strong motivation for privacy can overcome usability issues. This relates to a PGP-based authentication system by demonstrating that motivated users may successfully adopt complex encryption solutions, highlighting the importance of user motivation and usability design in secure systems.

(Bruseghini et al., 2022)Victory by KO: Attacking OpenPGP Using Key Overwriting presents vulnerabilities in OpenPGP's key management through Key Overwriting (KO) attacks, where an attacker can overwrite certain fields within OpenPGP's encrypted key packets, potentially allowing full recovery of private keys. The paper highlights the lack of cryptographic binding between fields in OpenPGP’s key structure, enabling cross-algorithm attacks that can misinterpret keys or replace public parameters with adversarial ones. This research underscores the importance of robust key validation mechanisms, which is relevant to secure PGP-based authentication systems by demonstrating the need for fortified protections against key tampering.

(Wueppelmann, 2015) PGP Auth: Using Public Key Encryption for Authentication on the Web presents a novel approach to web authentication using PGP as an alternative to traditional passwords. The PGP Auth system leverages public-key encryption to authenticate users securely, addressing vulnerabilities like phishing and password leaks. By requiring only a public key for login, PGP Auth enables secure account access without transmitting sensitive passwords. This method supports multiple device linking through key signing, relevant to secure login systems using PGP by showcasing a streamlined and user-friendly application of public-key cryptography for user authentication.

2.1 Comparison Table of Reviewed Literature Papers.

Serial No	Title	Authors	Advantages	Disadvantages
1	An Evaluation and Implementation of Pretty Good Privacy (PGP) in Wireless Network Security	Dr. Arvind Hans, M. Vigenesh, Dr.Yuvaraj. S, Suvarna R. Bhagwat, Hamza Zaki, Dr. Chiging Yamang	Enhances data confidentiality in wireless networks; adapts to various wireless infrastructures.	Challenges in key management.

2	Analysing Organised Crime on Darknet Marketplace Using PGP Public Keys	Shiying Fan, Paul Moritz Ranly, Lukas Graner, Inna Vogel, Martin Steinebach	Links vendor and buyer profiles; supports identification of organized groups.	Requires detailed analysis and cross-referencing for results.
3	Data Security at Cloud Storage using PGP in conjunction with IPsec VPN	Regin Raghavan Nair	Robust encryption with PGP and secure data transmission via IPsec VPN.	Relies on complex configuration for integration.
4	Exploring the Use of Pretty Good Privacy (PGP) in Wireless Network Security	Sunil. MP, Saket Mishra, Srisathirapathy *S*, Mahesh A., Vishal Sharma, Girija Shankar	Prevents eavesdropping in wireless communications; ensures authentication.	Usability concerns due to technical complexity.
5	Information Security using GNU Privacy Guard	Dabeeruddin Syed , Abdullah Hussein Al Ghushami, Ameema Zainab, Shafi Muhammad Abdulhamid , Mohammed Salem Daen A Al Kuwari	Open-source implementation; strong encryption and digital signatures.	Challenging key verification through the web of trust.
6	Multi-Factor Authentication in Cyber Physical System: A State of Art Survey	Sanjar Ibrokhimov, Kueh Lee Hui, Ahmed Abdulhakim Al-Absi, hoon jae lee, Mangal Sain	Highlights the strength of layered authentication approaches.	Requires specialized hardware for biometric factors.
7	ProtonMail Advance Encryption and Security	Kumkum Saxena, Dev Rajdev, Divesh Bhatia, Manav Bahi	End-to-end encryption; zero-access encryption ensures privacy.	Limited to email communication.
8	The Motivated Can Encrypt (Even with PGP)	Glencora Borradaile, Kelsy Kretschmer, Michele Gretes, Alexandria LeClerc	Shows that motivated users adopt secure systems despite usability issues.	Adoption depends on user motivation and education.
9	Victory by KO: Attacking OpenPGP Using Key Overwriting	Lara Bruseghini, Daniel Huigens, Kenneth G. Paterson	Exposes vulnerabilities to enhance security standards.	Demonstrates potential weaknesses in OpenPGP's key structure.
10	PGP Auth: Using Public Key Encryption for Authentication on the Web	Derek Wueppelmann	Eliminates passwords; secures authentication via PGP public keys.	Usability improvements needed for widespread adoption.

Table 1: Comparison of concepts

2.2 Literature Gap

Most of the discussed methods in the above studies does not focus on an authentication system for a website or an application which uses asymmetric encryption to authenticate users when logging in to their respective accounts where MFAs like google authenticator, SMS, Email OTPs are used to login to enhance security and usability.

Proposed Solution: By implementing PGP authentication as an MFA technique to create a hybrid, layered authentication model that would is encrypted inform of jumbled text that could be only decrypted with the respective private_key and passphrase.

3 Research Questions and Objectives

3.1 Research Question:

1. In what ways can PGP encryption technique enhance the security and reliability of user authentication systems?

3.2 Research Objective:

- To conceptualize and develop an authentication system that leverages PGP for secure challenge-response verification.
- To identify and address potential challenges in integrating PGP-based mechanisms into modern authentication workflows.
- To assess user adaptability and system scalability when deploying cryptographic methods in conventional environments.

3.3 Research Niche:

The niche of this research lies at the authentication process using a Public/Private key as an encryption and decryption technique for users to login to their respective account of a web application. The study addresses a new approach to authenticate users when logging into a web application account and it is compared to other traditional 2FA's (Multi-Factor Authentication). By focusing on the following areas this research aims to contribute significantly to the field of cybersecurity:

1. Use of PGP: Implementation of Pretty Good Privacy for user authentication.

Effectiveness of Encryption: Types of different encryption techniques used to evaluate login attempts on the user.

4 Research Methodology

The methodology approached in this paper is focused on identifying a solution which is suitable to be integrated into an existing infrastructure of an organization to add an extra layer of security to critical data owned by an organization. The Multi Factor authentication is primarily one of the used techniques to authenticate a process in digital world due to its scalability, availability and cost effectiveness, but it comes with other downfall such as concern of security of the critical data of any kind and dependency to ensure data security. To the use of otherwise an optional and robust solution using a combination of Symmetric and Asymmetric key encryption method. This section provides a comprehensive review of the cryptographic techniques employed, including Advanced Encryption Standard (AES) and public key cryptography algorithms like RSA and ECC. While every other Multi-Factor Authentication is unique in its own way, this research will focus on contributing more by adding encryption as security for a 2FA to verify a user login.

4.1 Pretty Good Privacy

(*How PGP Works*, n.d.)Pretty Good Privacy (PGP) is a hybrid encryption system that merges the advantages of symmetric and asymmetric cryptographic approaches to deliver a secure, efficient, and versatile encryption method. At its core, PGP operates by encrypting the main content of the communication using a symmetric encryption algorithm like AES, ensuring speed and minimal computational overhead. To secure the symmetric key itself, PGP utilizes asymmetric encryption, employing algorithms such as RSA or ECC to encrypt the symmetric key before it is transmitted to the recipient. This two-step process addresses the inherent weaknesses of each encryption type when used alone, creating a robust mechanism for secure data exchange.

In addition to encryption, PGP also incorporates digital signatures, which are created using the sender's private key. These signatures are crucial for ensuring the authenticity and integrity of the message, as they allow the recipient to verify that the message has not been tampered with and that it originated from the stated sender. The hybrid approach enables PGP to cater to diverse use cases, ranging from individual email encryption to securing data in large-scale enterprise applications.

Furthermore, PGP's adaptability to modern digital infrastructure makes it a preferred solution for protecting sensitive information. Its compatibility with various communication platforms and protocols enables seamless integration into both legacy systems and contemporary cloud-based applications. By providing a comprehensive suite of encryption, authentication, and verification tools, PGP addresses the multifaceted challenges of cybersecurity in the digital age.

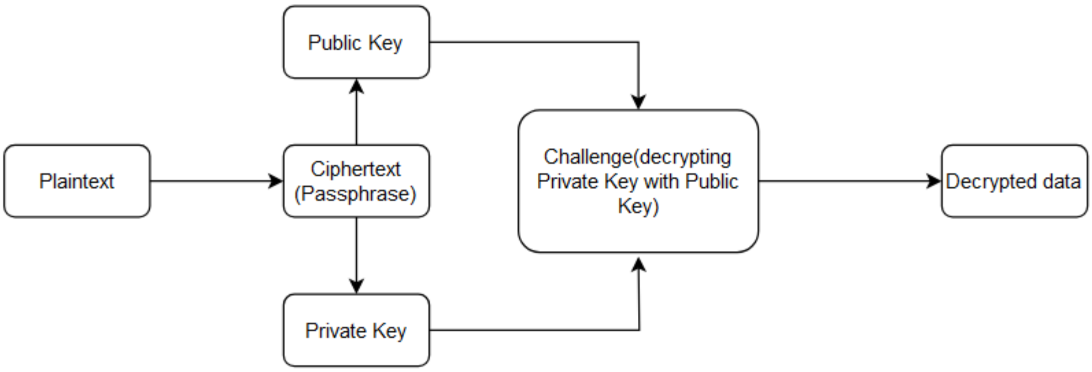


Figure 3.1 PGP working methodology

4.2 Encryption Techniques

This section details the encryption techniques employed in the project, with a specific focus on RSA (Rivest-Shamir-Adleman) encryption. Unlike other symmetric encryption methods such as AES or elliptic curve-based systems, RSA was chosen for its widespread adoption, robust security, and suitability for encrypting sensitive keys and authenticating data. In the context of this project, RSA serves as the backbone for encrypting the symmetric keys that secure the actual content of communications. By leveraging the strengths of RSA, the methodology ensures both the confidentiality of data during transmission and the authenticity of the sender.

4.2.1 Advanced Encryption Technique

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm widely recognized for its efficiency and security. (*Advanced Encryption Standard (AES) - GeeksforGeeks,* n.d.)While not the primary method used in this project, AES is worth noting due to its importance in modern cryptography. Key features of AES include:

- Block Cipher: AES encrypts data in fixed-size blocks (typically 128 bits), ensuring a consistent and predictable encryption process.
- Key Sizes: AES supports 128-bit, 192-bit, and 256-bit keys, offering scalable levels of security depending on the use case.
- Performance: Known for its speed, AES is highly efficient in both software and hardware implementations, making it ideal for encrypting large datasets.
- Adoption: AES is used globally in a variety of industries, including finance and government, for its proven reliability and strength against brute-force attacks.

While AES is not directly implemented in this project, understanding its role in symmetric encryption provides valuable context for comparing different encryption techniques.

4.2.2 RSA Encryption

RSA encryption forms the core of this project’s encryption methodology. As an asymmetric encryption algorithm, (*RSA Encryption*, n.d.)RSA uses a pair of keys a public key for encryption and a private key for decryption—to ensure secure communication. The following highlights its key features:

- **Key Pair Generation**: RSA generates large prime numbers to create a public-private key pair, with security strength depending on the key size (commonly 2048-bit or 4096-bit).
- **Encryption and Decryption**: The sender encrypts the symmetric key with the recipient’s public key.The recipient uses their private key to decrypt the symmetric key, enabling access to the encrypted message.
- **Digital Signatures**: RSA allows for the creation of digital signatures, where a hashed message is encrypted with the sender’s private key, providing both integrity and authenticity.
- **Security:** RSA’s security is based on the computational difficulty of factoring large composite numbers, making it highly resilient against conventional cryptographic attacks.

In this project, RSA is employed for encrypting symmetric keys and verifying the authenticity of communications, ensuring both confidentiality and trust in the system.

4.2.3 Elliptic Curve Encryption

Elliptic Curve Cryptography (ECC), while not implemented as part of this project, represents an important advancement in public-key cryptography. (*OpenPGP: Key Generation (Elliptic Curve)*, n.d.)ECC uses mathematical properties of elliptic curves over finite fields to achieve encryption, offering comparable security to RSA with significantly smaller key sizes. Key features include:

- **Efficiency**: ECC provides strong security with smaller key sizes (e.g., a 256-bit ECC key offers equivalent security to a 3072-bit RSA key), making it ideal for resource-constrained environments.
- **Applications**: ECC is widely used in mobile and IoT devices where computational power and memory are limited.
- **Performance**: ECC enables faster encryption and decryption processes compared to RSA in some scenarios, reducing computational overhead.

Although ECC was not the focus of this project, its potential for improving encryption efficiency in constrained environments makes it a valuable alternative for future exploration in projects requiring lightweight cryptographic solutions.

5 Design Specification

The techniques and methodology that were put to use to realize the project, along with the rationale for choosing them, are explained in detail in this part of the report. No matter how much we consider conventional security through any of the most commonly heard latest encryption techniques, whether it be ECC or even one of the advanced kinds like homomorphic encryption, problems such as complexity, performance, usability, and interoperability issues are likely to be created. In light of this, the view was to make use of: A better approach is developed by incorporating different industry-wide techniques. An implicit reasoning that lies at the core of the selection is the established prevalence of the selected techniques, namely AES and RSA, thereby enabling the proposed mechanism to be supported by existing software and hardware infrastructure. The essence of the approach lay in proposing a methodology that could be fitted into the existing system without performance or compatibility issues.

The design specification for the PGP system is divided into several critical processes, each contributing to the overall security and functionality of the encryption framework. The focus is on creating a system that ensures confidentiality, authenticity, and integrity while maintaining efficiency and ease of use.

Key Generation:

- Public-Private Key Pairs: Each user generates a public-private key pair using RSA, with key sizes typically set to 4096 bits to ensure a strong security baseline. These keys are used for encrypting symmetric keys and for digital signature creation.

- Symmetric Keys: Symmetric keys are randomly generated for each message or session. These keys are used to encrypt the actual message content, ensuring efficient data handling.

Encryption Process:

- Message Encryption: The message content is encrypted using a symmetric encryption algorithm. This ensures the encryption process is fast and suitable for large data sizes.
- Symmetric Key Encryption: The symmetric key is encrypted using the recipient's public RSA key. This ensures that only the recipient can decrypt the symmetric key with their private key, adding a layer of security to the process.

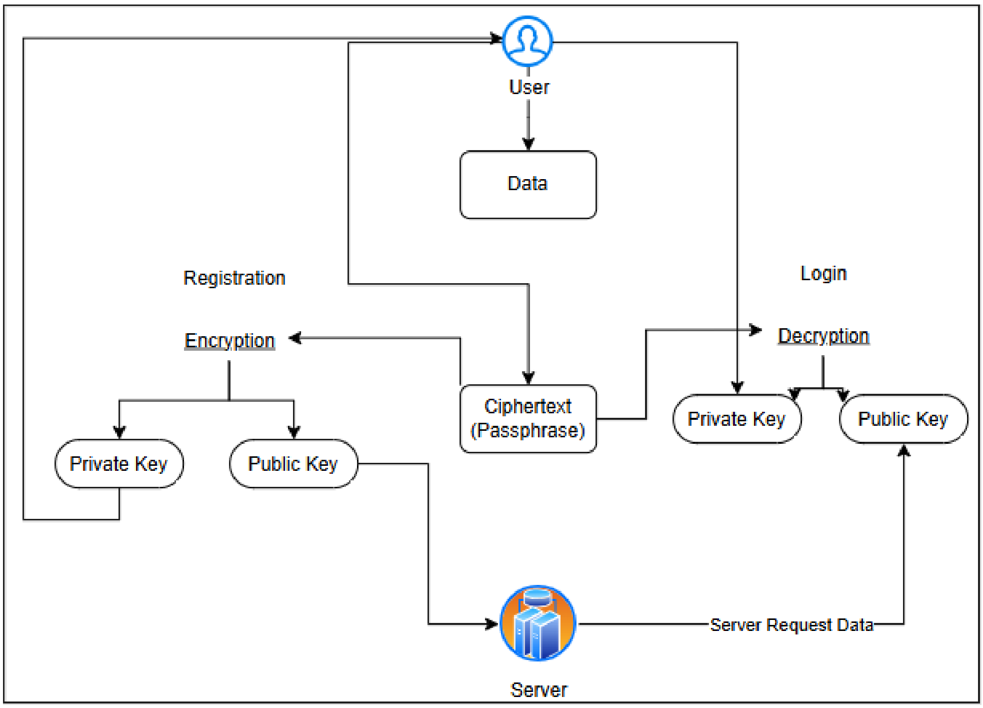


Figure 6.1 Encryption/Decryption Process

Decryption Process:

- Symmetric Key Decryption: Upon receiving the encrypted message and the encrypted symmetric key, the recipient uses their private RSA key to decrypt the symmetric key.
- Message Decryption: The decrypted symmetric key is then used to decrypt the message content, ensuring the original plaintext is restored securely.

Digital Signature Creation:

- Hashing the Message: A hash of the message is created using a secure hashing algorithm, such as SHA-256. The hash ensures message integrity.
- Signing the Hash: The hash is encrypted with the sender's private key to create a digital signature. This signature is appended to the message, allowing the recipient to verify the authenticity of the sender.

Digital Signature Verification:

- Hash Validation: The recipient decrypts the digital signature using the sender's public key to retrieve the original hash.

- Integrity Check: The recipient generates a new hash of the received message and compares it with the decrypted hash. If the hashes match, the message is verified to be authentic and untampered.

Design Components:

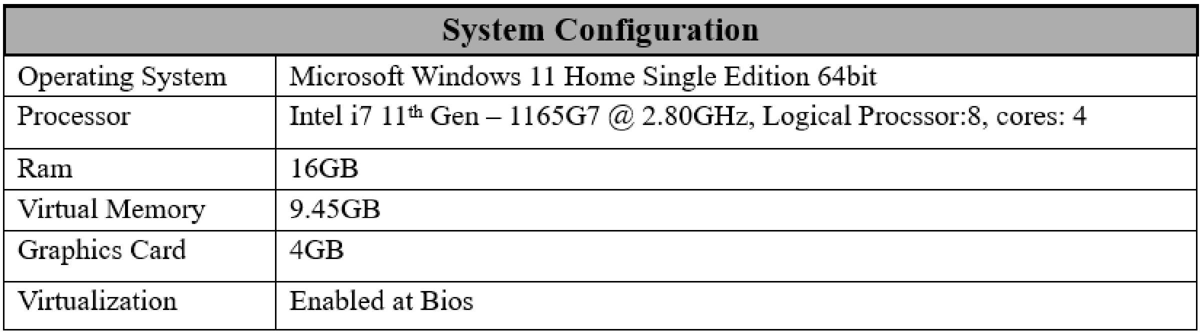
Key Exchange Mechanism: Public keys are exchanged through a secure channel or stored on a trusted key server, ensuring they are accessible to intended recipients without compromise.

Storage and Management:

- Private keys are stored locally and encrypted with a passphrase to prevent unauthorized access.
- Symmetric keys are generated dynamically for each communication session, limiting their exposure and reducing the risk of compromise.

6 Implementation

The implementation of the PGP-based challenge-response authentication system is designed to enhance the security and usability of modern authentication mechanisms. This section outlines the architecture, development, and integration of the proposed solution, detailing the roles of the frontend and backend components, the cryptographic operations, and the workflow for registration, login, and authentication.



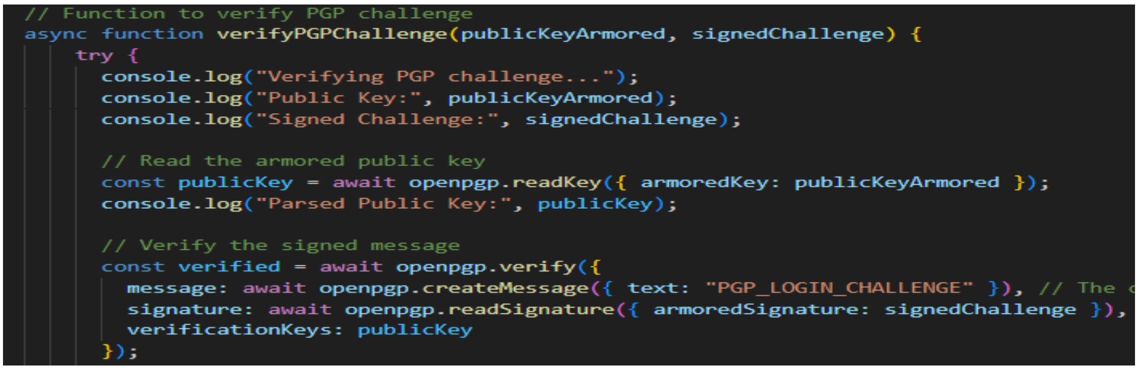
System Configuration	
Operating System	Microsoft Windows 11 Home Single Edition 64bit
Processor	Intel i7 11th Gen – 1165G7 @ 2.80GHz, Logical Procssor:8, cores: 4
Ram	16GB
Virtual Memory	9.45GB
Graphics Card	4GB
Virtualization	Enabled at Bios

Figure 7.1 System Specification

1. System Architecture

The system consists of two primary components:

- **Frontend**: A web-based user interface built with HTML, CSS, and (*Node.Js — Run JavaScript Everywhere*, n.d.)JavaScript, responsible for user interactions, PGP key management, and cryptographic operations.
- **Backend**: A Node.js server with a (*MySQL :: MySQL Workbench*, n.d.)MySQL database, managing user data, challenges, and verification logic. The server is responsible for storing public keys and issuing cryptographic challenges.



```
// Function to verify PGP challenge
async function verifyPGPChallenge(publicKeyArmored, signedChallenge) {
    try {
      console.log("Verifying PGP challenge...");
      console.log("Public Key:", publicKeyArmored);
      console.log("Signed Challenge:", signedChallenge);

      // Read the armored public key
      const publicKey = await openpgp.readKey({ armoredKey: publicKeyArmored });
      console.log("Parsed Public Key:", publicKey);

      // Verify the signed message
      const verified = await openpgp.verify({
        message: await openpgp.createMessage({ text: "PGP_LOGIN_CHALLENGE" }), // The
        signature: await openpgp.readSignature({ armoredSignature: signedChallenge }),
        verificationKeys: publicKey
      });
```

Figure 7.2 PGP Key Verification Snippet

2. PGP Key Management

Key management is a cornerstone of the system, ensuring that private keys remain secure and only public keys are stored on the server.

- **Key Generation:** The OpenPGP.js library is used to generate a public-private key pair on the client-side during registration. The public key is sent to the server for storage, while the private key remains with the user, protected by a passphrase.
- **Key Storage**: The private key is offered to users as a downloadable file and optionally stored locally in secure storage mechanisms like the browser's localStorage.

3. Registration Workflow

During registration, the system collects the user's basic details and generates their PGP key pair:

- The user provides a username, email, and password.
- The frontend generates a PGP key pair using OpenPGP.js.
- The private key is encrypted with a passphrase (e.g "TestPassphrase") and made available for download.
- The public key, along with the hashed password, is sent to the backend and stored in the MySQL database.

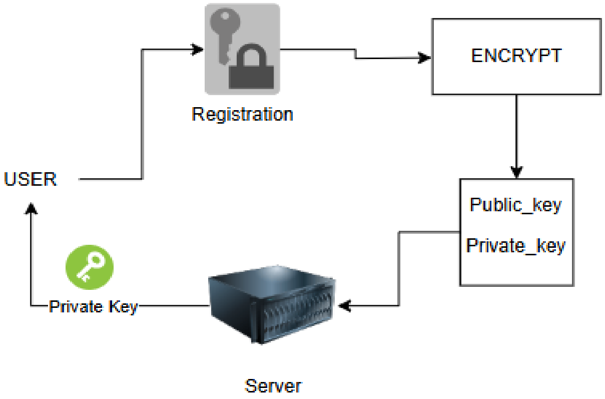


Figure 7.1 Registration Workflow

4. Login and Challenge-Response Authentication Workflow

The login process uses a challenge-response mechanism to verify the user's identity without transmitting sensitive credentials:

- Login Request: The user enters their username and password. The backend verifies the password and retrieves the user's stored public key.
- Challenge Issuance: Upon successful password verification, the backend generates a unique cryptographic challenge (e.g., a random string) and sends it to the client.
- Challenge Signing: The frontend decrypts the user's private key (prompting for their passphrase) and uses it to sign the challenge.
- Verification: The signed challenge is sent back to the backend, where it is verified against the user's public key. If the signature is valid, the user is authenticated.

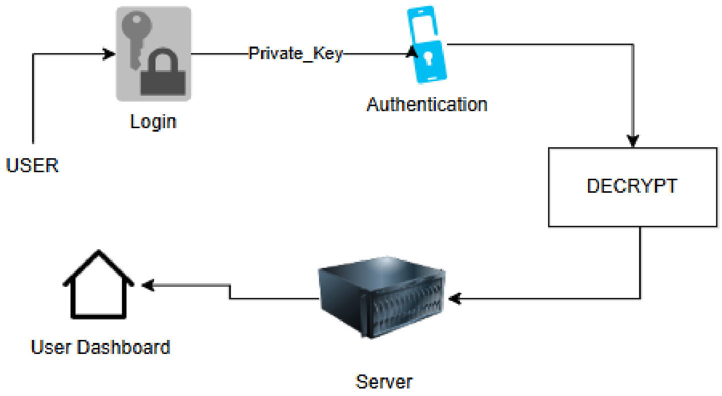


Figure 7.2 Login Authentication Workflow

5. Error Handling

- Error handling ensures the robustness of the system:
- Frontend: Alerts the user of invalid credentials, incorrect private key passphrase, or server errors.
- Backend: Logs all failed login attempts, invalid private key usage, and unverified challenges for monitoring and debugging purposes.

6. Security Considerations

The implementation includes several measures to enhance security:

- Private Key Protection: The private key is never transmitted to the server, ensuring end-to-end security.
- Public Key Storage: Only the public key and hashed password are stored on the backend, reducing the risk of sensitive data leaks.
- Challenge Expiry: Challenges are time-limited to mitigate replay attacks.
- HTTPS: All communication between the client and server is encrypted using HTTPS.

7 Evaluation

This section evaluates the PGP based authentication response system implemented in the project by comparing it with other majorly used Multi-Factor Authentication(MFA) methods. The analysis highlights the results in Advantages and Disadvantages in relation to security, practical application and usability.

7.1 Case Study 1 (Comparing with Other MFA methods)

1. **Passwords with SMS/Email Based OTPs**

Advantages

- Resistance to Phishing Attacks: Unlike OTPs that are sent over insecure channels (SMS, Email),PGP based authentication sensitive data(private_key) is accessible only through the device itself, mitigating phishing risks.
- No Dependency on Third Parties: The system eliminates to rely on telecommunication networks or email service, which reduces exposure to delivery delays, interception.
- Enhanced Cryptographic Security: PGP uses robust public-key cryptography, making it harder to compromise compared to Traditional OTPs which can be interrupted.

Disadvantages

- User Key Management: User must securely manage their private_keys, which introduces complexity compared to simple methods of receiving OTPs.

- Initial Setup Complexity: Generating and storing PGP Key requires more effort compared to setting up OTP based Authentication.

2. **Authenticator Apps (e.g Google Authenticator)**

   Advantages:
- Decentralized Approach: Unlike authenticator apps which are depend over centralized backend servers for key synchronization, PGP keys are managed by the users independently.
- Key based Authentication: PGP allows for secure cryptographic authentication that does not need regular device synchronization reducing the risks associated with server side breaches.

  Disadvantages:
- Learning Curve: Authenticator apps are usually user friendly and require less technical knowledge, while PGP needs familiarity with key management and encryption concepts for implementing.
- Lack of Convenience for Mobile users: The lack of mobile app integration can make PGP less convenient for users that rely on phones for authentication.

3. **Hardware Tokens (e.g YubiKey, FIDO)**

   Advantages:
- Cost Efficiency: PGP system can be implemented without the use of physical devices, reducing cost for both user and organization.
- Flexibility: The keys can be used across multiple devices are it doesn’t need special hardware. It eliminates physical risks by which key can be digitally stored.

  Disadvantages:
- Software Reliance: Because the system's security relies on the hardware and software used to handle PGP keys it is vulnerable to keylogging and malware assaults.
- Lack of Physical Security: Hardware tokens provide a tangible layer of security that is absent in PGP based systems.

7.2 Case Study 2

Advantages of the Proposed PGP-Based Authentication System

1. High Security:
   - Ensures that private keys are kept safe and arent shared over the network by using asymmetric cryptography.
   - Removes password based weaknesses including credential stuffing and bruteforce attacks.
2. Decentralized and User-Controlled:
   - Centralized storage is not relied in this case which reduces risk of mass breaches.
   - Also puts users in control of their own key which heightens privacy.
3. Cost Effective Implementation:
   - Does not require specialized hardware or rely on third part services to authenticate.

Disadvantages of the Proposed Systen

1. Usability Challenges:

   Key management should be understood by users and encryption concepts which maybe a barrier for non technical users.
2. Dependency on User Devices:

   Without a backup recovery from a hacked device or deleted private key may be challenging or impossible.
3. Limited Mobile Integration:

Engagement by consumers used to app-based MFA solutions may be discouraged by the absence of easy mobile app integration.

7.3 Discussion

This PGP-based authentication system tries to solve several well-known security weaknesses in most of the conventional MFA methods. At least in theory, this could be a strong and decentralized approach. Usability and teaching key management to the user are potential obstacles to real-world use. This works fine in security-critical situations when the field requires elevated privacy, but further efforts need to be taken regarding ease of access by users and incorporating the solution to levels where mobile and hardware adoption can become seamless.

8 Conclusion and Future Work

The contribution of this research was to illustrate how PGP can be used to improve security and reliability in user authentication by designing an implementation of the PGP-based challenge-response authentication system. No passwords or OTPs are required, yet the system allows for users' secure verification through cryptographically signed challenges. The main observations with this system include high resistance to phishing attacks, a tendency for authentication processes to be decentralized, and no third-party dependence. Such features make this PGP-based authentication a very powerful alternative to traditional ones in highly secure environments. It also expressed important implications, most especially the absolute need for user-managed key management to provide for decentralized authentication and prevent massive security breaks. Results indicate that PGP is capable of a strong authentication method that is cryptographic sound, but usability issues stand as a limitation, and for non-technical users, it is a challenge to manage keys, seamlessly integrating with mobile platforms. Notwithstanding these limitations the current work forms a feasible and important part of incorporating PGP encryption into modern authentication schemes. It creates a foundation for further research on usability improvements, the incorporation of mobile technology, and hardware-based key storage, therefore forming a landmark contribution in securing yet user-friendly authentication systems.

Future work in the PGP-based authentication system should go toward further ease of use and availability, such as developing user-friendly interfaces and automating key management tools in order to lower the technical barrier for non-expert users. Yet another promising line of development is mobile-phone integration, including developing special applications that will allow seamless integration of PGP authentication into a smartphone ecosystem and further investigating biometric authentication methods to increase ease of use without compromising security. Hardware-based key storage solutions, including hardware security modules or secure USB devices, will further provide a higher degree of physical security for private keys. Effective and user-oriented key recovery methods will also be key in addressing those problems emanating from key loss or device failure. This will ultimately permit scalability and resilience testing through the system's implementation in various real-life situations, such as corporate environments; work on its interoperability with more cryptographic protocols will further boost its usability.

References

Advanced Encryption Standard (AES) - GeeksforGeeks. (n.d.). Retrieved November 9, 2024, from https://www.geeksforgeeks.org/advanced-encryption-standard-aes/

Borradaile, G., Kretschmer, K., Gretes, M., & LeClerc, A. (2021). The Motivated Can Encrypt (Even with PGP). *Proceedings on Privacy Enhancing Technologies*, *2021*(3), 49–69. https://doi.org/10.2478/popets-2021-0037

Bruseghini, L., Huigens, D., & Paterson, K. G. (2022). Victory by KO: Attacking OpenPGP Using Key Overwriting. *Proceedings of the ACM Conference on Computer and Communications Security*, 411–423. https://doi.org/10.1145/3548606.3559363

Encryption — OpenPGP for application developers. (n.d.). Retrieved December 9, 2024, from https://openpgp.dev/book/encryption.html

Fan, S., Ranly, P. M., Graner, L., Vogel, I., & Steinebach, M. (2024). Analysing Organised Crime on Darknet Martketplace Using PGP Public Keys. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3664476.3670464

How PGP works. (n.d.). Retrieved November 9, 2024, from https://users.ece.cmu.edu/~adrian/630-f04/PGP-intro.html

Ibrokhimov, S., Hui, K. L., Abdulhakim Al-Absi, A., Lee, H. J., & Sain, M. (2019). Multi-Factor Authentication in Cyber Physical System: A State of Art Survey. *International Conference on Advanced Communication Technology, ICACT*, *2019-February*, 279–284. https://doi.org/10.23919/ICACT.2019.8701960

MySQL :: MySQL Workbench. (n.d.). Retrieved December 5, 2024, from https://www.mysql.com/products/workbench/

Node.js — Run JavaScript Everywhere. (n.d.). Retrieved December 5, 2024, from https://nodejs.org/en

OpenPGP: Key generation (Elliptic Curve). (n.d.). Retrieved December 9, 2024, from https://asecuritysite.com/ecc/openpgp01

Raghavan Nair, R. (2023). *Data Security at Cloud Storage using PGP in conjunction with IPsec VPN*.

RSA encryption. (n.d.). Retrieved December 9, 2024, from https://www.comparitech.com/blog/information-security/rsa-encryption/

Saxena, K., Rajdev, D., Bhatia, D., & Bahl, M. (2021). ProtonMail: Advance Encryption and Security. *Proceedings - International Conference on Communication, Information and Computing Technology, ICCICT 2021*. https://doi.org/10.1109/ICCICT50803.2021.9510041

Sunil, M. P., Mishra, S., Srisathirapathy, S., Bhandari, M. A., Sharma, V., & Sahoo, G. S. (2023). Exploring the Use of Pretty Good Privacy (PGP) in Wireless Network Security. *2023 3rd International Conference on Smart Generation Computing, Communication and Networking, SMART GENCON 2023*. https://doi.org/10.1109/SMARTGENCON60755.2023.10442370

Syed, D., Al-Ghushami, A. H., Zainab, A., Abdulhamid, S. M., & Al-Kuwari, M. S. D. A. (2023). Information Security using GNU Privacy Guard. *2023 IEEE 13th Annual Computing and Communication Workshop and Conference, CCWC 2023*, 295–300. https://doi.org/10.1109/CCWC57344.2023.10099196

Wueppelmann, D. (2015). *PGP Auth: Using Public Key Encryption for Authentication on the Web* [Carleton University]. https://doi.org/10.22215/ETD/2015-11139