*I.J. Wireless and Microwave Technologies*, **2024, 5, 1-11**
Published Online on October 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2024.05.01

# Securing Musical Streams: Leveraging ElGamal Encryption in REST API Frameworks for PWAs

**Timothy John Pattiasina***
Department of Information System, Faculty of Information Technology, Institut Informatika Indonesia (IKADO) Surabaya, East Java, 60189, Indonesia
E-mail: temmy@ikado.ac.id
ORCID iD: https://orcid.org/0000-0003-0351-9441
*Corresponding Author

Received: 04 February, 2024; Revised: 20 May, 2024; Accepted: 12 July, 2024; Published: 08 October, 2024

**Abstract:** Subscription music platforms, like many web applications, increasingly rely on Progressive Web Apps (PWAs) to enhance user experience. These PWAs function by exchanging data with servers or REST APIs. However, the current reliance on REST APIs poses significant security risks due to vulnerabilities in data transmission. To address this issue, this research integrates the El Gamal cryptographic algorithm into the architecture of a subscription music platform. By incorporating the El Gamal cryptographic algorithm, this research endeavors to fortify the security posture of data exchanged between users and servers through REST APIs. This encryption method was selected for its robust resistance to various cryptographic attacks, providing a strong defense against unauthorized interception and tampering of sensitive information. To evaluate the efficacy of the El Gamal integration, a rigorous white box testing regimen was employed, encompassing metrics such as cyclomatic complexity and basic path testing. These assessments comprehensively examined the code's structure and execution to identify potential vulnerabilities and ensure the correct implementation of the cryptographic algorithm. The findings of the white box testing unequivocally demonstrated the successful integration of El Gamal cryptography on both the client and server components of the subscription music platform, effectively safeguarding the confidentiality and integrity of data transmitted via REST APIs. This research contributes to the advancement of secure communication protocols within web applications, particularly subscription-based platforms. Through the implementation of robust encryption, the study enhances data integrity and confidentiality, ultimately building user trust.

**Index Terms:** Subscription Music Platform, Progressive Web Apps, REST API, El Gamal, White Box Testing.

## 1. Introduction

Rapid technological advancements in recent years have brought significant innovations in presentation and program design, particularly in building simple mobile apps and websites. These advancements have enabled the development of more reliable and efficient computing systems [1,2]. With various development methods and techniques available today, developers can quickly create programs or applications that yield impressive results [3,4]. Despite these advancements, some websites, such as music subscription platforms, often overlook critical security elements, considering them insignificant [5]. For instance, Spotify, a music subscription website, misused encryption technology in communicating its subscription pages, which contain sensitive information such as personal details and credit card information used for payment purposes. Protecting this information from theft by malicious parties is crucial [6]. In 2018, Indonesia experienced at least 3,127,497 cases of user data theft, underscoring the importance of robust security measures. Beyond the convenience and benefits provided by computers and the internet, there is a dark side that can significantly impact human life [7].

Currently, an increasing number of developers are applying the concept of Progressive Web Apps (PWA) to develop website-based programs that provide fast, optimal services and user experiences [8,9]. PWAs operate like native applications, offering notifications and user experiences similar to those of native applications. By utilizing the offline-first concept and communicating data through the Representational State Transfer Application Programming Interface (REST API), PWAs ensure that pages running on the browser do not display network error messages or white screens, even on very poor networks [10,11]. However, the problem arises with the open nature of the request and response communication between the client and the server, which allows malicious parties to intercept and steal information or carry out man-in-the-middle attacks. Currently, website communication security is ensured by using SSL

(Secure Socket Layer) on pages accessed via HTTPS [12]. While the SSL protocol secures data traffic during the communication building process (session layer), the data can still be exposed during the data presentation process (application layer) [13].

This study addresses the security concerns associated with obtaining password information in web applications, particularly in the context of REST API data communication. Unlike conventional methods, this study employs the ElGamal encryption algorithm to encrypt all REST API data requests and responses, ensuring enhanced data security. The ElGamal algorithm, initially designed for digital signatures, has been adapted for data encryption and decryption, offering robust security measures [14]. Compared to existing studies [15,16,17], this approach offers several advantages. Firstly, by encrypting all data requests and responses using ElGamal encryption, the system ensures that only authorized requests can access and read the data transmitted by the server. Additionally, the server can only decrypt information from clients that adhere to the same encryption protocol, minimizing spam requests and enhancing overall data integrity. The use of ElGamal encryption, a type of public-key cryptography, further strengthens security measures. Public-key cryptography, utilizing a public key provided by the owner and a private key kept secret, allows for secure message transmission without the need to share private keys [18]. This method ensures that encrypted messages can only be decrypted with the recipient's private key, enhancing data confidentiality and security.

The experiment results confirm the effectiveness of the ElGamal cryptographic algorithm in securing data communication within the subscription music platform system. The implementation of ElGamal encryption in the PWA web application utilizing REST API demonstrates its viability and efficacy in real-world scenarios, highlighting its potential for enhancing data security in web-based applications.

## 2. Related Works

This section provides an in-depth review of current cryptographic algorithms used to ensure secure online data communication. The review focuses on the specific cryptographic algorithms in use, emphasizing their role in facilitating secure data exchanges on online platforms. The evaluation includes an analysis of the algorithms' effectiveness in protecting the integrity and confidentiality of information during online transactions. Additionally, the strengths, weaknesses, and challenges associated with these cryptographic algorithms are explored to offer a comprehensive understanding of their impact on online data security.

Mousavi et al. [19] proposed a novel hybrid cryptographic algorithm that integrates the Rivest Cipher (RC4), Elliptic-Curve Cryptography (ECC), and Secure Hash Algorithm (SHA-256) to protect sensitive information in IoT-based smart irrigation systems. In this approach, the ECC algorithm encrypts the RC4 key, and the output is transformed using SHA-256 for hashing and generating cryptic data. The SHA-256 algorithm is used to encrypt RC4-based ciphertext, thereby enhancing data integrity. Analysis and simulation results show that the proposed scheme is resilient against various known attacks, including the Man-in-the-Middle (MiM) attack, and performs better than other cryptographic algorithms. The results confirm the effectiveness of the proposed model and its robustness in maintaining confidentiality through secrecy analysis. In another study by the same authors [20], symmetric, asymmetric, and hybrid encryption algorithms were examined to enhance IoT security. The use of asymmetric key encryption ensures secure communication among multiple users, eliminating the need to distribute keys through insecure channels. The comparison of algorithms based on security factors reveals that ECC outperforms others in terms of performance. ECC generates smaller, faster, and more reliable cryptographic keys while reducing memory requirements and the time needed for encryption/decryption.

AlRikabi and Hazim [21] conducted a study on the heightened importance and necessity of data security in daily life. They addressed the risk of the confidentiality of files stored in systems vulnerable to hacking. To mitigate these risks, various methods are employed to secure different forms of data, including text, pictures, and videos. Their paper focuses on embedding a protected image within another image by transforming its format using the DWT wavelet transform. The technique involves zeroing specific sites and storing their contents, followed by mathematical processing using the exponential function to produce a fully encrypted image. The essential image is concealed behind the encrypted image. The proposed system includes two algorithms—one for encoding and hiding, and another for efficiently restoring and decoding the main image to its original state.

Qazi et al. [22] focused on addressing security concerns in Wireless Sensor Networks (WSN) by providing authentication and data encryption for node-to-node communication. Their proposed scheme ensures secure node-to-node communication and conserves memory space on nodes using the Elliptic Curve Digital Signature (ECDSA) cryptographic scheme. This scheme measures key generation time, the count of hello messages, and packet size effectively. Additionally, the Algorithm for Secure Communication in Wireless Networks (ASCW) facilitates key management with an acceptable key length. ASCW enhances communication security at the node level, contributing to overall network security. It also mitigates risks and security threats on the network by incorporating an authentication mechanism. A physical testbed was designed based on specified requirements. Experimental results indicate that ASCW is a suitable and innovative approach for securing data on nodes during communication in WSNs.

A literature review by Imam et al. [23] addresses challenges related to throughput, latency, energy consumption, and security on the Internet of Things (IoT). Wireless technologies are categorized based on mechanisms at the physical layer, media access control layer, and network layer. The essentials of wireless communication and considerations concerning these layers are summarized according to IoT criteria. Security threats and vulnerabilities in standards against potential attacks at these layers are outlined. An integrated approach is emphasized to provide solutions at each layer. The research presents a concise analysis of available wireless communication standards and their key features, exploring the scope and challenges of IoT applications, identifying potential directions for future research, and detailing the third-generation partnership project for low-power wide-area solutions designed to meet IoT requirements.

Another related study [24] reviewed current security issues, wireless communication techniques, and technologies aimed at securing IoT. The study's objective was to enhance security at an individual technique level and identify loopholes to address IoT network security comprehensively. The research examines previous contributions to understand better and addresses all security perspectives simultaneously. Detailed analysis of countermeasures and challenges from security perspectives is undertaken, in line with current industry trends. Blockchain, machine learning, fog, and edge computing are considered potential solutions for securing IoT. Machine learning, especially when integrated with end-to-end security, emerges as a promising candidate. This comprehensive review provides thorough understanding and knowledge to define security strategies for successful IoT implementation.

Tepetes et al. [25] introduced "Sheltered," a hybrid application designed for secure and private video calls between pairs of users over the internet. A secure exchange of unique call IDs is facilitated through an RSA-based scheme. They also created a chat system based on the anonymous connection of two users to a temporary chat room, with no registration or authorization needed. Communication within these chat rooms is encrypted using the AES encryption standard and is not stored after the chat session concludes. As a hybrid application, 'Sheltered' is a web application packaged into a lightweight native application container, granting access to native platform features and device hardware. Developed using web technologies like JavaScript, HTML5, CSS, NodeJS, React, and ExpressJS, 'Sheltered' also utilizes the Ionic framework to wrap native code into a JavaScript API, creating a corresponding mobile app.

Table.1 Comparison of Findings from Related Works

| Study | Methodology | Advantages | Challenges |
|---|---|---|---|
| Mousavi et al. [19] | Hybrid algorithm: RC4, ECC, SHA-256 | Resilient against MiM attacks, superior performance, maintains confidentiality | Implementation complexity |
| AlRikabi and Hazim [21] | Data embedding with DWT wavelet transform | Effective data concealment, protects against detection and discrimination | Limited data type versatility |
| Qazi et al. [22] | ECDSA cryptographic scheme for WSNs | Efficient key management, node-level security, conserves memory space | Potential memory constraints |
| Imam et al. [23] | Comprehensive review of IoT security challenges at physical, MAC, and network layers | Integrated security approach, broad coverage of IoT standards and challenges | Lacks specific implementation details |
| Tepetes et al. [25] | Hybrid application for secure video calls and chat using RSA and AES | Practical real-time communication solutions, secure exchange of call IDs, anonymous temporary chat rooms | Primarily applicable to video calls and chat systems |
| Current Research | Elgamal cryptography for securing personal information on a subscription music platform | Ensures confidentiality of personal and credit card details, encrypts REST API requests and responses | Context-specific application to subscription music platforms |

Based on several previous studies, this research lays the foundation for developing a system to secure personal information and credit card details on a subscription music platform using cryptography [26]. This information is considered crucial and requires confidentiality. In its development, every request and data response from the REST API are encrypted using El Gamal cryptography.

## 3. Proposed Methodology

Architectural design encompasses the development and functioning of a system, focusing on the intricate interactions between the system and its users, as well as the connections between the system and its underlying database. The integration of a robust database is a fundamental aspect of system creation, ensuring efficient data management and retrieval. In our proposed system, we explore the dynamic relationships among the user, Content Delivery Network (CDN) server, and REST API server. The CDN server is responsible for delivering content swiftly and reliably to users, while the REST API server handles data requests and responses, ensuring secure and seamless communication between the user interface and the database. This architecture is designed to optimize performance, enhance user experience, and maintain data security. To illustrate, the relationships among the user, Content Delivery Network (CDN) server, and REST API server were depicted below.

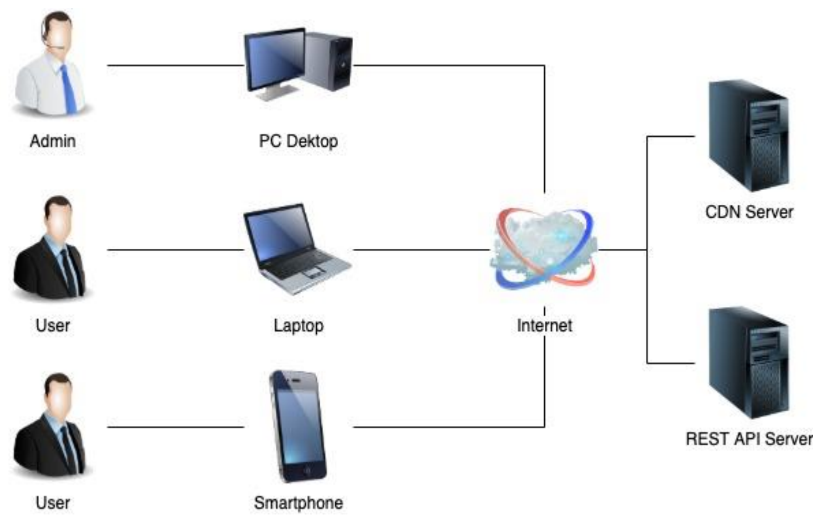


Fig. 1. System Architecture Design.

As illustrated in Figure 1, the system users comprise regular users and administrators. Consequently, it is necessary to allocate access rights to restrict each user's capabilities. Users can perform actions such as registration, login, music playback, profile management, and payments. In contrast, admin users can log in and manage music. All users can access the website using various devices such as computers, laptops, and smartphones. The page content and other necessary files, such as CSS, JavaScript, images, and music, will be loaded through the CDN Server, while requests for data will be made to the REST API Server.

The use case diagram serves to identify the primary functions and services that the system is designed to offer [27]. Figure 2 below presents the use case diagram depicting the key interactions within the primary system. It outlines the various actions or functionalities that users, whether they are regular users or administrators, can engage with. These actions encompass activities such as user registration, login, music playback, profile management, and payment processing.

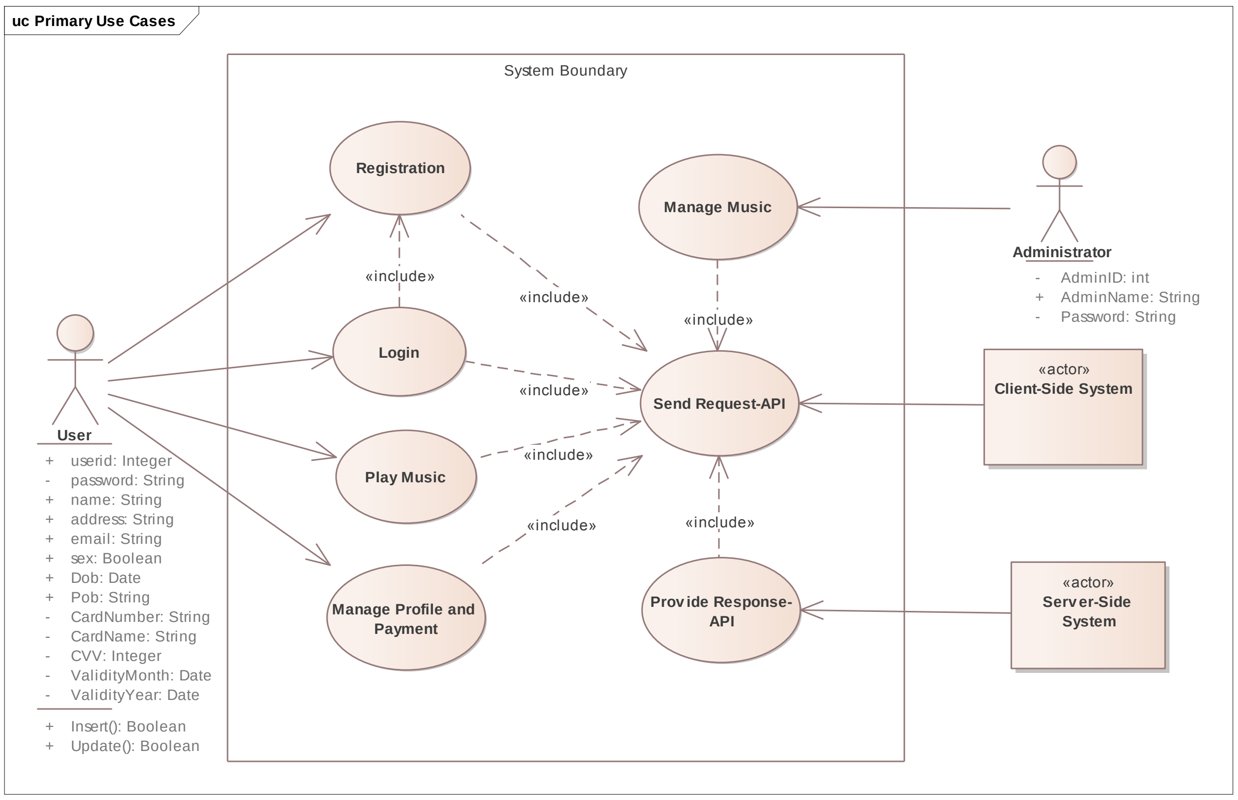


Fig. 2. Primary System Use Case Diagram.

Moreover, the diagram illustrates the connections and interactions between different system components, providing a high-level overview of how users and system elements collaborate. As the system evolves, this use case diagram serves as a valuable reference for understanding the fundamental functionalities it supports. This visual representation aids in the comprehensive exploration of potential user-system interactions, offering insights into the scope and features of the primary system. It is a pivotal tool in refining the system's design and ensuring that it aligns seamlessly with user requirements and expectations.

There are four types of actors: users, administrators, client-side systems, and server-side systems. Each actor type has specific use cases aligned with their respective needs. In this context, the user actor refers to general users or members subscribing to the service. Users possess various functionalities within the system, including registration, login, music playback, and management of profiles and payments. Conversely, the administrator actor is responsible for managing the music within the subscription-based platform, having the authority to oversee and control the available music content.

The next step involves creating a use case diagram that explains the sequence of activities related to the implementation of ElGamal cryptography. This diagram provides an overview of the expected functionality and illustrates the interactions between system components. Below is the use case diagram for the implementation of ElGamal cryptography:

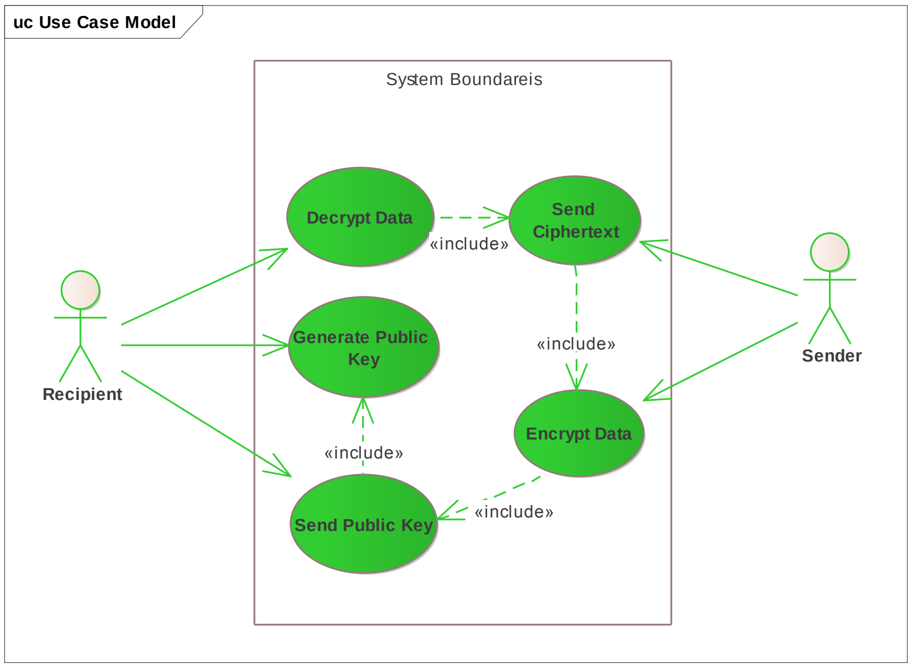


Fig. 3. ElGamal Implementation Use Case Diagram

In Figure 3 above, the recipient is an actor who serves as the data receiver. The recipient has the ability to generate a public key, send the public key, and decrypt the data. On the other hand, the sender is an actor who functions as the data sender, having the capability to encrypt the data and send the ciphertext.

The class diagram depicts the structure and description of classes, packages, and objects, along with their relationships, providing a broad overview of a system by showcasing entities and their associations [28]. The relationship between classes is called multiplicity or cardinality. Additionally, a class diagram portrays the state (attributes/properties) of a system while also offering services to manipulate that state (methods/functions). Here is a class diagram of a subscription-based music platform:

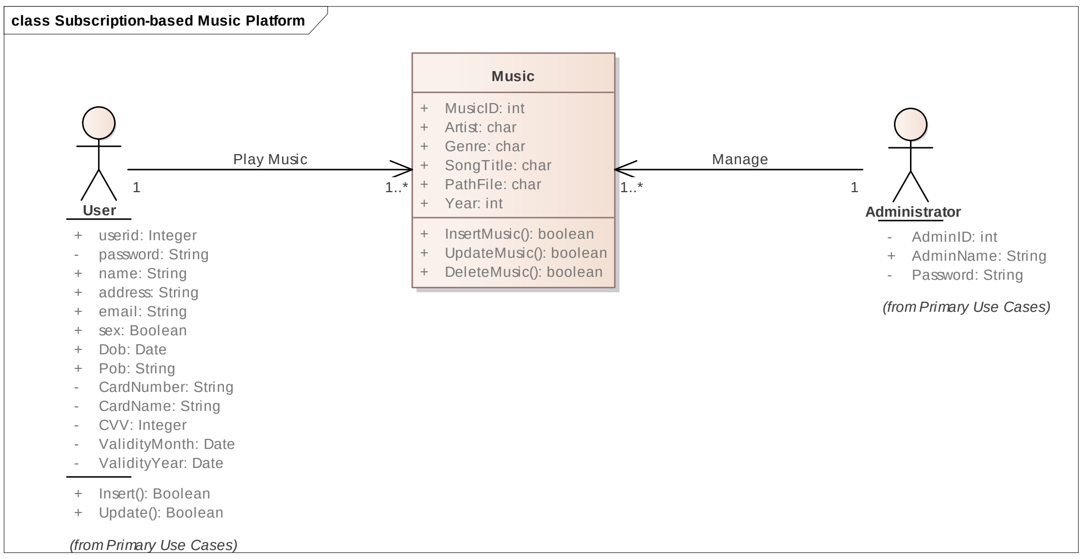


Fig. 4. System Class Diagram

In Figure 4, there are three entities: user, music, and admin. The user entity has attributes such as user ID, name, address, city, email, gender, place of birth, date of birth, password, card number, cardholder name, validity month, validity year, and CVV. Furthermore, the user entity has functions for inserting an account and updating an account. The music entity possesses attributes including music ID, title, artist, genre, year, and file location. Additionally, the music entity has functions for inserting music, updating music, and deleting music. The administrator entity comprises attributes such as admin ID, admin name, and password.

The entire data communication in the system will be carried out using API requests. The API request process is performed on the client side to receive and send data to the REST API server.

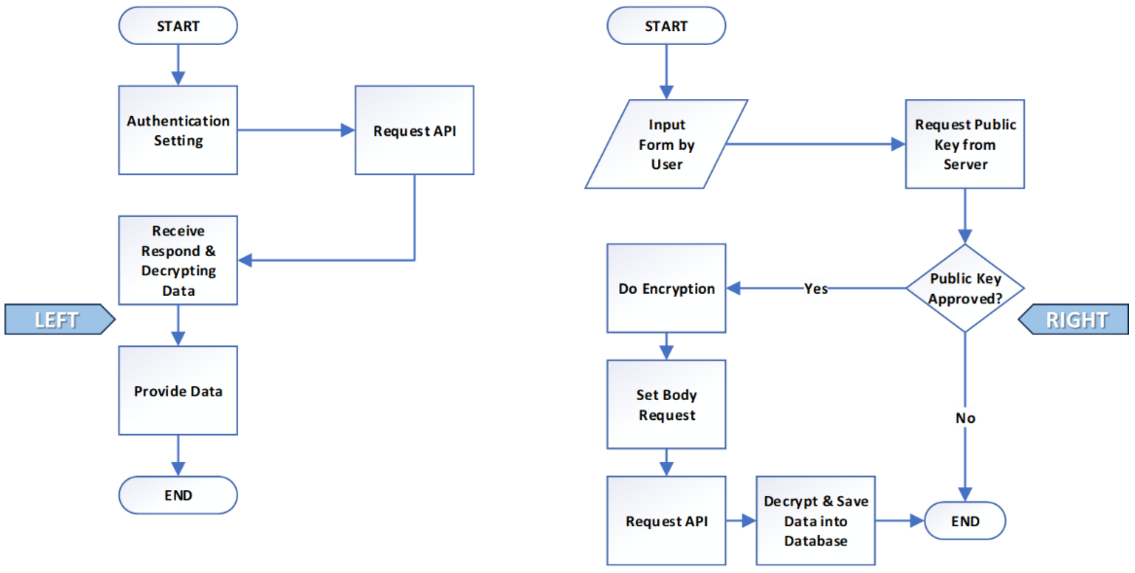


Fig. 5. API Request Send & Receive Data Process

Examining the left flowchart illustrated in Figure 5 reveals the sequence of steps involved in receiving data, as outlined below:

1. Set Authentication in the request header with a public key.
2. Client-side makes a request using the GET method on the URL to be addressed.
3. After receiving a response in the form of ciphertext from the server, the system on the client-side will decrypt it into plaintext.
4. Presenting data to the user as needed.

In the flowchart positioned to the right of Figure 5, the procedure for the client-side request process during the transmission of data to the server unfolds as follows:

1. The user completes and submits the form.
2. The client initiates an API request to the server using the GET method to acquire the public key.
3. If the public key is successfully received, the client encrypts the form data using the obtained public key, converting it into ciphertext. However, if the public key retrieval fails, the client displays a message indicating the failure.
4. The client configures the request body with the ciphertext.
5. The client sends a request to the server using the POST method, specifying the appropriate URL for data submission.
6. Following the transmission to the server, the server receives the ciphertext and decrypts it into plaintext.
7. The server processes the plaintext data and stores it in the database.

It's important to note that this description assumes the use of asymmetric encryption where the public key is used for encryption on the client side and the private key is used for decryption on the server side [29]. Additionally, the specific implementation details may vary depending on the technology stack and the design of the API.

Specifically, during key generation, a large prime number $p$ and its primitive root $g$ are selected, followed by the computation of the public and private keys. In the encryption process, each character of the plaintext is converted to its ASCII value, encrypted using randomly selected integers and the public key, and combined to form ciphertext. Below is the pseudocode for the encryption process, as shown in Table 1.

Table 2. ElGamal Key Generation and Encryption Pseudocode

**Pseudocode**

```
function encrypt(message, privateKey, p):
    ciphertext = ""
    for each character in message:
        ASCII_value = convert_to_ASCII(character)
        k = generate_random_integer()
        c1 = g^k mod p
        c2 = (ASCII_value * y^k) mod p
        encrypted_character = combine(c1, c2)
        ciphertext += encrypted_character
    return ciphertext
```

The decryption process (Table. 2) for the ElGamal cryptographic algorithm involves several key steps to recover the original plaintext message from ciphertext. After receiving the encrypted message, the server or recipient first splits each encrypted character into its components $c1$ and $c2$. Using the recipient's private key $x$ and the prime modulus $p$, $c1$ is raised to the power of $x \ modulo \ p$ *(1)* to compute $c1^x \ mod \ p$ *(2)*. Next, the modular inverse of *(2)* is calculated to derive $m$, which represents the original ASCII value of the plaintext character. This ASCII value is then converted back into its corresponding character form, and this process is repeated for each character in the ciphertext. By systematically applying these decryption steps, the recipient effectively reverses the encryption process, ensuring that the original plaintext message is accurately reconstructed and can be comprehended by the intended recipient.

Table 3. Pseudocode for Decryption Process of ElGamal Algorithm.

**Pseudocode**

```
function decrypt(ciphertext, publicKey):
    plaintext = ""
    for each encrypted_character in ciphertext:
        c1, c2 = split(encrypted_character)
        c1_x = c1^privateKey mod p
        m = (c2 * modular_inverse(c1_x, p)) mod p
        character = convert_to_character(m)
        plaintext += character
    return plaintext
```

## 4. Results & Discussion

As a result, the user interface of the subscription-based music platform, designed to interact with every user. It covers various functionalities, including the registration process, login, music playback, music management, and the implementation of the account information management page. The intended system implementation can be seen in Figures 6 and 7 below:

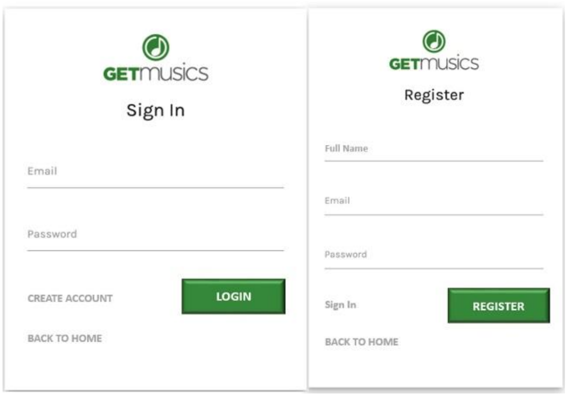
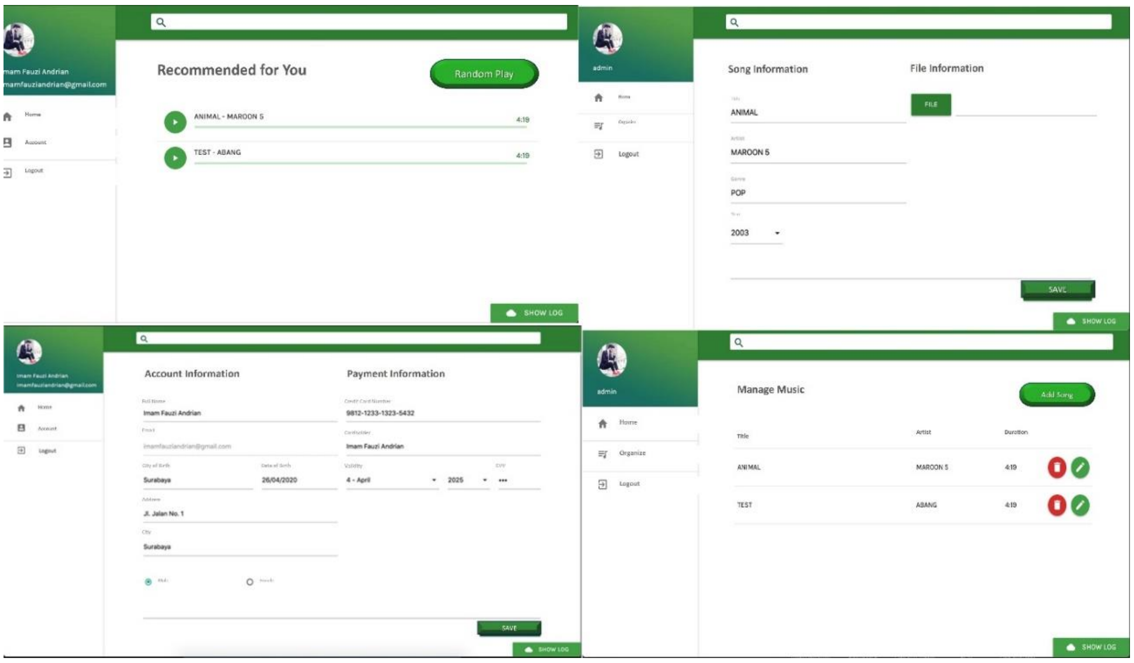


Fig. 6. User Login & Registration Page



ANIMAL - MAROON 5 4:19
TEST - ABANG 4:19

ANIMAL
MAROON 5
POP
2003

Imam Fauzi Andrian
imamfauziandrian@gmail.com
Surabaya 26/04/2020
Jl. Jalan No. 1
Surabaya
9812-1233-1323-5432
Imam Fauzi Andrian
4 - April 2025 •••

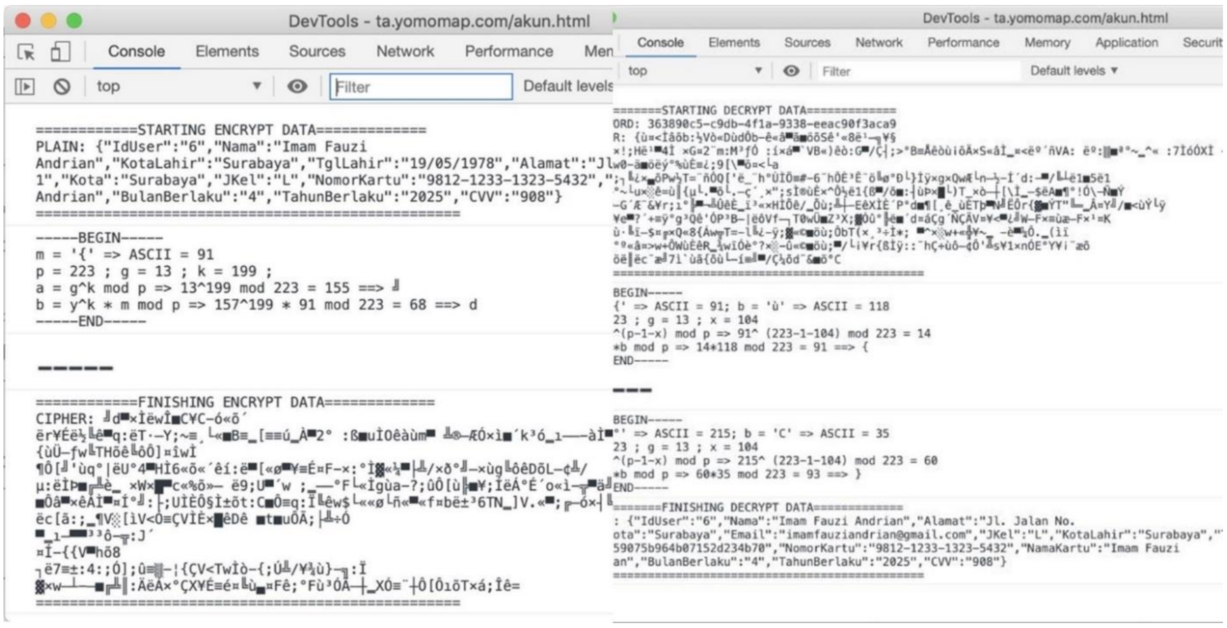
| Title | Artist | Duration |
|---|---|---|
| ANIMAL | MAROON 5 | 4:19 |
| TEST | ABANG | 4:19 |

Fig. 7. Music Management & User Profile Page

The robust ElGamal cryptographic algorithm has been meticulously integrated into each and every page of the subscription music platform website. This strategic implementation stands as a testament to the platform's unwavering commitment to safeguarding the integrity and confidentiality of the information embedded within each page. This security measure becomes particularly pronounced during data communication facilitated by the REST-API, where the ElGamal algorithm plays a pivotal role in fortifying the security infrastructure on both the client and server sides.

The meticulous implementation of the ElGamal cryptographic algorithm is not merely a blanket security measure but is intricately woven into the fabric of data communication processes across the entire website. This ensures a comprehensive and consistent application of security protocols, creating a robust defense mechanism against potential threats. By embedding the ElGamal algorithm within the REST-API framework, the subscription music platform guarantees that every interaction involving data, whether initiated by the client or processed on the server side, is fortified with a layer of cryptographic protection, enhancing the overall security posture of the platform. This strategic integration exemplifies a proactive stance toward cybersecurity, underscoring the platform's commitment to providing users with a secure and trustworthy environment for their interactions [30].

Once the system is developed, various tests are conducted to ensure its proper functionality. Below are some of the tests performed on the encryption and decryption processes using ElGamal cryptography:



```
=============STARTING ENCRYPT DATA=============
PLAIN: {"IdUser":"6","Nama":"Imam Fauzi
Andrian","KotaLahir":"Surabaya","TglLahir":"19/05/1978","Alamat":"Jl
1","Kota":"Surabaya","JKel":"L","NomorKartu":"9812-1233-1323-5432","
Andrian","BulanBerlaku":"4","TahunBerlaku":"2025","CVV":"908"}
==================================================

------BEGIN------
m = '{' => ASCII = 91
p = 223 ; g = 13 ; k = 199 ;
a = g^k mod p => 13^199 mod 223 = 155 ==> ╝
b = y^k * m mod p => 157^199 * 91 mod 223 = 68 ==> d
------END------

------

=============FINISHING ENCRYPT DATA=============
CIPHER: ╝d▀×İëwÎ■C¥C-ó«õ´
[illegible]
==================================================
```

```
=======STARTING DECRYPT DATA=============
ORD: 363890c5-c9db-4f1a-9338-eeac90f3aca9
R: {ù¤<Îâõb:½Vò«DùdÔb-ê«â▀ã■õõSê'«8ë¹¬¥§
[illegible]
==================================================

BEGIN------
{' => ASCII = 91; b = 'ù' => ASCII = 118
23 ; g = 13 ; x = 104
^(p-1-x) mod p => 91^ (223-1-104) mod 223 = 14
*b mod p => 14*118 mod 223 = 91 ==> {
END------

---

BEGIN------
°' => ASCII = 215; b = 'C' => ASCII = 35
23 ; g = 13 ; x = 104
^(p-1-x) mod p => 215^ (223-1-104) mod 223 = 60
*b mod p => 60*35 mod 223 = 93 ==> }
END------
=======FINISHING DECRYPT DATA=============
: {"IdUser":"6","Nama":"Imam Fauzi Andrian","Alamat":"Jl. Jalan No.
ota":"Surabaya","Email":"imamfauziandrian@gmail.com","JKel":"L","KotaLahir":"Surabaya","
59075b964b07152d234b70","NomorKartu":"9812-1233-1323-5432","NamaKartu":"Imam Fauzi
an","BulanBerlaku":"4","TahunBerlaku":"2025","CVV":"908"}
==================================================
```

Fig. 8. Data Encryption & Decryption Test

Figure 8 illustrates the client-side encryption process for data sent to the server via the REST API. Each character is encrypted and combined to form ciphertext. This encryption occurs for every API request, which is then processed by the server. Decryption on the client-side meticulously processes each character within the received ciphertext, applying decryption and skillfully combining the results to reconstruct the original plaintext. This ensures data integrity and security during transfer.

White-box testing revealed the robustness of the ElGamal cryptographic algorithm implementation on the subscription-based music platform. All identified paths through the encryption and decryption processes were thoroughly tested. Cyclomatic complexity was measured for both functions, identifying areas where code complexity could lead to vulnerabilities. For example, the encryption process (Figure 8) involves conditional statements and loops that handle character transformation into ciphertext. Basic path testing ensured that every independent path through the encryption and decryption code is executed at least once. Test cases verified that each character is correctly encrypted and combined into ciphertext for every API request, and accurately decrypted and reconstructed into the original plaintext.

To assess the completeness of the tested features, a feature completeness matrix was employed. This matrix provides a comprehensive evaluation of the implementation's coverage and functionality. For the ElGamal cryptographic algorithm, the encryption and decryption processes were evaluated against four criteria: accuracy (weighted 40%), efficiency (weighted 20%), reliability (weighted 20%), and security (weighted 20%). Each feature was scored out of 10 for each criterion. The encryption process received scores of 9/10 for accuracy, 8/10 for efficiency, 9/10 for reliability, and 9/10 for security. These scores were then weighted and summed, resulting in a total completeness score of 8.8/10 (3.6 for accuracy, 1.6 for efficiency, 1.8 for reliability, and 1.8 for security). The decryption process received identical scores and achieved a completeness score of 8.8/10 as well. These high scores indicate successful implementation and functionality for the majority of features. Cyclomatic complexity analysis and basic path testing further ensured all possible paths were tested and optimized, confirming the robustness, reliability, and secure data communication of the ElGamal cryptographic algorithm implementation within the subscription-based music platform.

## 5. Conclusion

In conclusion, this study successfully implemented ElGamal cryptography within a subscription-based music platform utilizing a REST API on a PHP and JavaScript web platform. This integration demonstrably enhanced data security, particularly for sensitive information like credit card details, by leveraging custom-developed client-side and server-side modules for ElGamal encryption and decryption. This approach offered a robust defense against potential security breaches and was applicable to various web systems, including Progressive Web Apps (PWAs). White box testing was instrumental in validating the implementation, ensuring that all identified paths through the encryption and decryption processes were thoroughly tested. Metrics such as cyclomatic complexity and basic path testing provided insights into critical areas of the code, enhancing the reliability and security of the encryption framework. Future research should explore further advancements in data security by integrating additional private key cryptographic methods, such as Caesar cryptography or a signcryption scheme, to create a layered defense against evolving threats. Additionally, implementing a royalty module could significantly benefit music creators by tracking and managing royalties based on music streaming and usage, ensuring fair compensation. By incorporating such a module, the subscription music platform could evolve into a comprehensive solution that fosters a sustainable ecosystem, benefiting both subscribers and music creators.

## References

[1] M. Banafaa, I. Shayea, J. Din, M. H. Azmi, A. Alashbi, Y. I. Daradkeh, and A. Alhammadi. "6G mobile communication technology: Requirements, targets, applications, challenges, advantages, and opportunities," *Alexandria Engineering Journal*, pp. 245-274, 2023. doi: 10.1016/j.aej.2022.08.017.

[2] H. Wang, H. Ning, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand. "A Survey on the Metaverse: The State-of-the-Art, Technologies, Applications, and Challenges," in *IEEE Internet of Things Journal*, pp. 14671-14688, 2023, doi: 10.1109/JIOT.2023.3278329.

[3] T. J. Marion and S. K. Fixson. "The transformation of the innovation process: How digital tools are changing work, collaboration, and organizations in new product development." *Journal of Product Innovation Management*, pp. 192-215, 2021. doi: 10.1111/jpim.12547.

[4] A. Haleem, M. Javaid, M. A. Qadri, R. Suman. "Understanding the role of digital technologies in education: A review," *Sustainable Operations and Computers*, pp. 275-285, 2022. doi: 10.1016/j.susoc.2022.05.004.

[5] J. W. Morris. "Music Platforms and the Optimization of Culture," *Social Media + Society*, 2020. doi: 10.1177/2056305120940.

[6] F. Widi, A. Qahar, and A. Aswari. "Legal protection against personal data in online loan transactions," *Golden Ratio of Law and Social Policy Review (GRLSPR)*,pp. 7-26, 2021. doi: 10.52970/grlspr.v1i1.152.

[7] B. Tumalun. "Upaya Penanggulangan Kejahatan Komputer Dalam Sistem Elektronik Menurut Pasal 30 Undang-Undang Nomor 11 Tahun 2008," *Lex Et Societatis*, pp. 24-31, 2018. doi: 10.35796/les.v6i2.19950.

[8] B. Susanto, G. Virginia, U. Proboyekti, J. C. D. Ester. "Progressive Web App Implementation in Omah Wayang Klaten Website," in *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI*, Springer Nature Singapore, 2023, pp. 333-348. doi: 10.1007/978-981-99-0835-6_24.

[9] P. H. Putri, L. N. Hasanah. "Srikandi Health: Development of a progressive web apps-based health information system as a solution for ease of monitoring and management of anemia," In *AIP Conference Proceedings*, AIP Publishing, 2023, Vol. 2491. doi: 10.1063/5.0105495.

[10] A. I. Khan, A. Al-Badi, and M. Al-Kindi. "Progressive Web Application Assessment Using AHP," *Procedia Computer Science*, pp. 289-294, 2019. doi: 10.1016/j.procs.2019.08.041.

[11] A. Biørn-Hansen, T. A. Majchrzak, and T. M. Grønli, "Progressive Web Apps: The Possible Web-native Unifier for Mobile Development," in *Proceedings of the 13th International Conference on Web Information Systems and Technologies*, pp. 344–351, SciTePress, 2017. doi:10.5220/0006353703440351.

[12] R. Dastres and M. Soori. "Secure Socket Layer (SSL) in the Network and Web Security," *International Journal of Computer and Information Engineering*, In press, pp.330-333, 2020. https://hal.science/hal-03024764.

[13] A. Alabduljabbar, R. Ma, S. Choi, R. Jang, S. Chen, and D. Mohaisen. "Understanding the security of free content websites by analyzing their SSL certificates: a comparative study, " in *Proceedings of the 1st Workshop on Cybersecurity and Social Sciences*, May 2022, pp. 19-25. doi: 10.1145/3494108.3522769.

[14] D. Kumbhakar, K. Sanyal, and S. Karforma. "An optimal and efficient data security technique through crypto-stegano for E-commerce," *Multimed Tools Appl*, pp. 21005–21018, 2023. doi:10.1007/s11042-023-14526-7.

[15] F. Mallouli, A. Hellal, N. S. Saeed, and F. A. Alzahrani. "A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms," In *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)*, Jun. 2019, pp.173-176, doi: 10.1109/CSCloud/EdgeCom46520.2019.

[16] O. A. Imran, S. F. Yousif, I. S. Hameed, W. N. Abed, and A. T. Hammid. "Implementation of El-Gamal algorithm for speech signals encryption and decryption," *Procedia Computer Science*. pp. 1028-1037, 2020. doi: 10.1016/j.procs.2020.03.402.

[17] E. R. Arboleda. "Secure and Fast Chaotic El Gamal Cryptosystem," *International Journal of Engineering and Advanced Technology (IJEAT)*. pp. 1693-1699, 2019. doi: 10.35940/ijeat.2249-8958.

[18] A. Braeken. "Public key versus symmetric key cryptography in client–server authentication protocols," *International Journal of Information Security*, pp. 103-114, 2022. doi: 10.1007/s10207-021-00543-w.

[19] S. K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari. "Improving the security of internet of things using cryptographic algorithms: a case of smart irrigation systems," *Journal of Ambient Intelligence and Humanized Computing*, Feb. 2021, pp. 2033-2051. doi: 10.1007/s12652-020-02303-5.

[20] S. K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari. "Security of internet of things based on cryptographic algorithms: a survey", *Wireless Networks*, Feb. 2021. pp. 1515-1555. doi: 10.1007/s11276-020-02535-5.

[21] H. T. S. ALRikabi, and H. T. Hazim. "Enhanced data security of communication system using combined encryption and steganography," *iJIM*, p. 145, 2021. doi: 10.3991/ijim.v15i16.24557.

[22] R. Qazi, K. N. Qureshi, F. Bashir, N. U. Islam, S. Iqbal, and A. Arshad. "Security protocol using elliptic curve cryptography algorithm for wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 547–566. 2021. doi: 10.1007/s12652-020-02020-z.

[23] R. Imam, Q. M. Areeb, A. Alturki, and F. Anwer. "Systematic and critical review of rsa based public key cryptographic schemes: Past and present status," *IEEE Access*, pp. 155949-155976, 2021. doi: 10.1109/ACCESS.2021.3129224.

[24] N. A. Khan, A. Awang, and S. A. A. Karim. "Security in Internet of Things: A review," *IEEE access*, pp.104649-104670, 2022. doi: 10.1109/ACCESS.2022.3209355.

[25] K. Tepetes, E. Papaioannou, C. Kaklamanis .SHELTERED”: A HYBRID PRIVACY-PRESERVING APPLICATION FOR SECURE VIDEO CALLS OVER IP, *INTED2023 Proceedings*, pp. 563-573, 2023. doi: 10.21125/inted.2023.0202.

[26] M. Z. Gunduz, and R. Das. "Cyber-security on smart grid: Threats and potential solutions," *Computer networks*, p. 107094, 2020. doi: 10.1016/j.comnet.2019.107094.

[27] E. R. Aquino, P. De Saqui-Sannes, and R. A. Vingerhoeds, “A Methodological Assistant for Use Case Diagrams,” in *Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development*, 2020, pp. 227–236. doi: 10.5220/0008938002270236.
[28] A. Tazin and M. M. Kokar, “Composition of UML Class Diagrams Using Category Theory and External Constraints,” *Journal of Software Engineering and Applications*, vol. 15, no. 12, pp. 436–468, 2022, doi: https://doi.org/10.4236/jsea.2022.1512025.
[29] S. Zarni, "Performance Comparison of Asymmetric Cryptography (Case study- Mail message)," *APTIKOM Journal on Computer Science and Information Technologies*, pp. 15-21, 2019. doi: 10.11591/APTIKOM.J.CSIT.147.
[30] B. Krishna, S. Krishnan, and M. P. Sebastian. "Understanding the process of building institutional trust among digital payment users through national cybersecurity commitment trustworthiness cues: a critical realist perspective", *Information Technology & People*, 2023. doi:10.1108/ITP-05-2023-0434.

## Authors’ Profiles

**Timothy John Pattiasina:** PhD student at State University of Malang (UM) Indonesia, Lecturer of the Department of Information System, Faculty of Information Technology Institut Informatika Indonesia (IKADO) Surabaya. Areas of scientific interests: Computer Science, Computer Security, Fuzzy, Game & Augmented Reality.

**How to cite this paper:** Timothy John Pattiasina, "Securing Musical Streams: Leveraging ElGamal Encryption in REST API Frameworks for PWAs", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.14, No.5, pp. 1-11, 2024. DOI:10.5815/ijwmt.2024.05.01