LONTAR KOMPUTER VOL. 16, NO. 1 APRIL 2025
DOI : 10.24843/LKJITI.2025.v16.i01.p06
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

p-ISSN 2088-1541
e-ISSN 2541-5832

Development of Secure API to Support ICD-10 Based Electronic Medical Records Interoperability

I G N Lanang Wijayakusuma[a1], Made Sudarma[b2], I Ketut Gede Darma Putra [b3], Oka Sudana[b4], Minho Jo[b5], I Putu Winada Gautama[b6]

[a]Fakultas Teknik, Universitas Udayana
Bukit Jimbaran, Indonesia
[1]lanang_wijaya@unud.ac.id
[2]msudarma@unud.ac.id
[3]ikgdarmaputra@unud.ac.id
[4]agungokas@unud.ac.id

[b]Fakultas MIPA, Universitas Udayana
Bukit Jimbaran, Indonesia
[6]winadagautama@unud.ac.id

[c]Departement of Computer and Information Science, Korea University
Sejong Metropolitan City, South Korea
[6]minhojo@korea.ac.kr

Abstract

Previous research in 2021 and 2022 has yielded a revolutionary health examination system. This system seamlessly integrates the World Health Organization's International Classification of Diseases-10 (ICD-10) data, ensuring diagnoses align with global standards and thereby enhancing the quality of healthcare provision. A pivotal achievement is the creation of a sophisticated doctor's examination interface, designed for precision and efficiency. Complementing this interface, a search engine autonomously generates relevant keywords, successfully passing the rigorous black-box test, which attests to its robustness and reliability in retrieving critical medical information. A new challenge arises in enabling seamless access to the stored medical record data for various stakeholders, including the Ministry of Health, BPJS, insurance companies, and other relevant entities. To address this, the research team has devised the Application Programming Interface (API). Functioning as a crucial bridge, this API facilitates interoperability among diverse systems. Adherence to the stringent security standards set by the Open Web Application Security Project (OWASP) ensures that the exchange of medical data occurs within a secure environment. Consequently, sensitive patient information can be shared across platforms without compromising confidentiality or integrity.

Keywords: *API, ICD-10, Health Infrastructure, Medical Records, OWASP*

1. Introduction

In light of the COVID-19 pandemic, the healthcare industry has experienced a significant surge in the use of information technology. As a result, many organizations have created a variety of health applications. To promote seamless information sharing between these applications, the usage of application programming interfaces (APIs) has become increasingly important[1], [2], [3]. Healthcare professionals have embraced APIs because they streamline diagnosis coordination, data reporting, and insurance reimbursement. Furthermore, APIs are easy to use and can be implemented across multiple hospitals to expedite the development of comprehensive electronic medical records[4], [5], [6], [7].

Developing a highly secure API is crucial, as it serves as the primary gateway for unauthorized individuals to access sensitive information. Authentication procedures are typically stored in sessions and cookies to ensure that access is granted only to authorized users, which must be protected at all times. Secure Application Programming Interfaces (APIs) are now widely adopted across various sectors, such as insurance[8], banking[9], [10], [11], [12], government[13], [14], academia [15], and healthcare[16], [17], [18]. In the healthcare sector, various techniques, such as [19], [20], [21], [22], [23], [24], [25], [26], [27] and the OWASP methodology [28], are available to secure APIs. It is essential to create a secure API as it serves as the main point of entry for unauthorized individuals to gain access to sensitive data. Generally, organizations store authentication procedures in sessions and cookies to ensure that they grant access only to authorized users, which must be safeguarded at all costs. Several techniques, such as encryption and the OWASP approach, can be utilized to secure APIs in the healthcare sector[29], [30], [31].

Furthermore, the study focuses on the Internet of Medical Things (IoMT), which integrates medical devices and applications through network technologies to connect healthcare information systems. The research implemented a hybrid security solution (SFTSDH = SF + TSD + H) using the Spring Framework, Services for Sensitive Data, and HTTP security methods. This solution addresses identity verification, secure data collection, and exchange in health applications, incorporating features such as identity brokering, OAuth2, multifactor authentication, and access control [32], [33]. The study extended the security solution to create a digital infrastructure for electronic health (eHealth) research and innovation, specifically in the context of an electronic coaching (eCoaching) prototype system. The implemented solution effectively secured the eCoaching APIs against attacks and demonstrated resilience under a load of approximately 1000 concurrent users in the digital health infrastructure. Qualitative comparisons with other security solutions, including SF security and third-party security, indicated that the SFTSDH solution showed promising outcomes [33].

It has been observed that articles associated with eHealth research highlight the security and privacy requirements for healthcare systems. These requirements apply to both on-premises and cloud-based Electronic Health Record (EHR) security. The articles discuss user authentication, governing amenability, authorized access, secrecy, ethical consent, legal issues and the relevance of data access, data ownership, data uniformity, data separation, security audits, archiving, and requirements for third-party certificates such as SAS70 Type II, PCI DSS Level 1, ISO 27001, and FISMA. They also cover protection against external security threats such as DoS, DDoS, MITM, IP spoofing, security policies, security protocols, and database access management. The identified vital security terms can be classified into four categories: authentication, authorization, encryption, and external security threats. Authentication is used to verify personal identity and is related to verifying credentials. It determines whether the person is a legitimate user. Authorization is a mechanism to determine whether a particular service is available only to authenticated users. It verifies that users have the right to access resources such as records, databases, and files. Typically, authorization is performed after authentication to verify a user's privileges. Encryption utilizes an algorithm to encrypt data and then employs a key to decrypt it, ensuring the secure transmission of information to the intended recipient. External threats are the possibility that people outside the system may use malicious software, hacking, disruption, or social engineering to exploit system vulnerabilities[34], [35], [36].

Healthcare research shows various studies associated with security and protection in electronic health records (EHRs), secure health monitoring framework, security conventions, and endorsement plans, security contemplations in medical services applications, strategies for medical services security, and interoperability, healthcare cloud, big data security, medical services security consistence, security execution, security difficulties, and success factors. However, studies related to security solutions at the Microservice Solution Architecture Level (MSSA) are limited[16], [17], [37].

The main contribution of this research lies in the development of an Application Programming Interface (API) that is equipped with end-to-end encryption, a departure from the conventional API approach. Notably, all credentials and requests have been placed in the API header, a feature that yields several advantages, including enhanced security, ease of use, standardization, customization, better control, and regulatory compliance[38].

2. Research Methods

This chapter examines the research phases and methodologies employed in the conducted study. The subsequent content elaborates on research techniques used to develop the Secure API.

2.1. Secure API Overview

In the dynamic realm of healthcare informatics, the protection of Electronic Medical Records (EMRs) is paramount. Our academic exploration introduces Secure API, a pioneering solution meticulously crafted to fortify the sanctity of medical data, setting new standards in security for electronic health systems. Figure 1 shows how the secure API works :

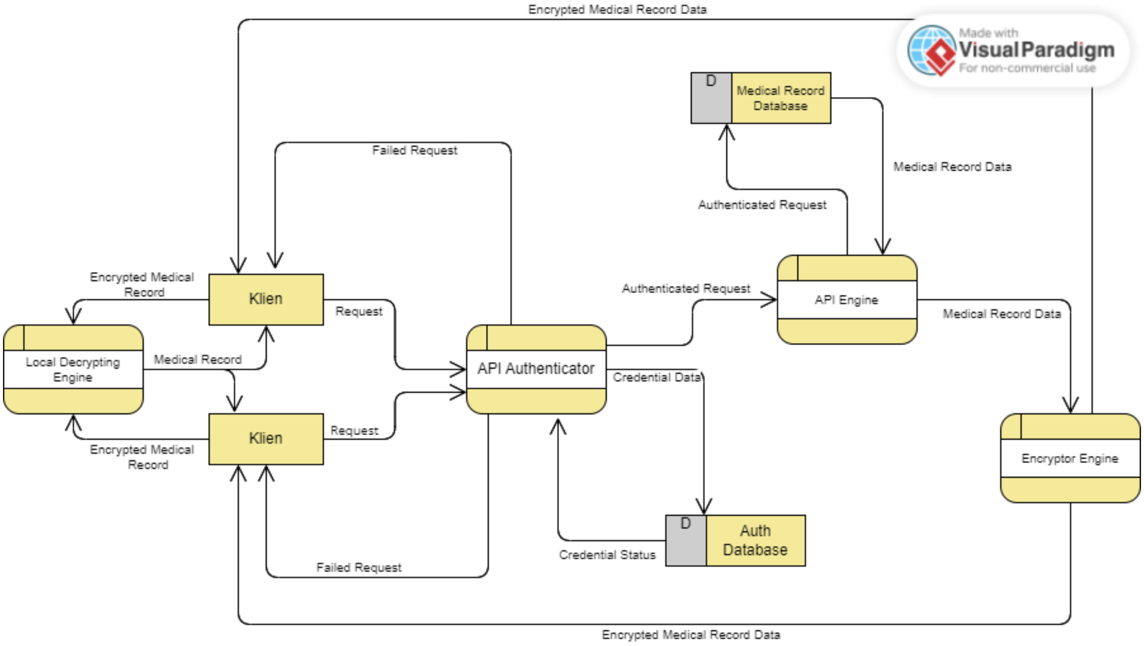


Figure 1. Secure API Overview

As illustrated in the image above, the API developed in this study consists of three key modules: the end-to-end encryption module, the credential authenticator module, and the database module. In practice, the health database administrator will manually generate both the end-to-end encryption key and the credential key. Subsequently, these keys will be disseminated to health service facilities after the registration procedure. Should this API be implemented, it could significantly reduce patient treatment timelines and provide all healthcare facilities with a comprehensive record of patient medical data.

There are some well-established security features implemented in the design of a secure API for electronic medical records, as shown in Table 2 :

Table 1. Established Security Features Implemented

Security Features	Description
Bearer token of Cloudflare under attack mode	Bearer tokens from Cloudflare are used in the API to limit active request time, minimizing the risk of a DDOS attack. It validates user requests and blocks suspicious traffic, such as traffic from virtual private servers (VPS) or automated crawling bots.
API Key	The database admin generates the API key through a one-way hashing process to ensure client access rights. The proposed design will provide a database administrator with an API key upon registration. But the authentication procedure is outside the scope of this research.
End-to-End Encryption Using AES-256-CTR	The encryption system that we use for end-to-end encryption is based on the AES-256-CTR encryption algorithm. This encryption module is installed on the medical record database side, while the decryption module is installed on each health facility system. The decryption module will be installed during the registration process for the health service facility. The secret key for the decryption module is generated when the API key is created, and each client has a unique secret key.
Cloudflare web application firewall (WAF) custom rules	To further protect the API server, WAF custom rules are used to block requests from specific countries and URLs, so only validated countries and URLs can access the API.

The EMR data is encrypted using AES-256 in Counter mode (CTR), which provides high throughput and parallelism support, making it suitable for real-time health information systems. The encryption process takes place at the server level before data transmission, ensuring:

a. Data Confidentiality: No intermediate system, including the API gateway, can access plaintext.
b. Client-Specific Keys: Each healthcare client has a unique symmetric key, generated upon API key issuance and securely delivered during the registration phase.
c. Decryption Locality: Only authorized client systems can decrypt the data using the pre-installed local decryption engine

The authentication process employs API keys hashed using cryptographic hash functions, and these keys are :

a. Generated by the Database Admin
b. Uniquely mapped to client credentials and secret keys.
c. Immutable and irretrievable even by the system admin

To mitigate Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks, bearer tokens are issued during session initiation. Requests without valid tokens or those with high-frequency abnormal behavior are dropped. The token lifespan is limited to:

a. Reduce surface area for brute-force attempts
b. Prevent automated bots or crawlers (common from VPS IP pools)

Cloudflare custom web application firewall (WAF) rules are applied to filter allowed countries (geo-based blocking), whitelist specific endpoints/URLs, and detect patterns associated with malicious payloads (e.g., SQL injection, XSS).

The design implements Zero Trust security practices by assuming no request is trusted by default, even from within the network. It re-authenticates each session and token using time-sensitive credentials and enforces least-privilege access based on API key roles.

With all of this safety features implemented, we design the API to comply with HIPAA (USA): via end-to-end encryption and access control, GDPR (EU): by ensuring data minimization, encryption at rest and in transit and Permenkes No. 24/2022 (Indonesia): Supports "Satu Sehat" interoperability and secure third-party data access.

2.1. Use Case Diagram

Figure 2 illustrates the users of the secure API, comprising three entities: health facility administrators, doctors, and patients. Initially, the health facility administrator conducts admissions by either searching for or inputting patient data to consolidate all records of a specific patient into one comprehensive medical file. Following that, the physician evaluates the patient using subjective and objective assessment techniques, as well as planning strategies, while considering the patient's medical background. Lastly, patients have the option to access and retrieve their health history data.

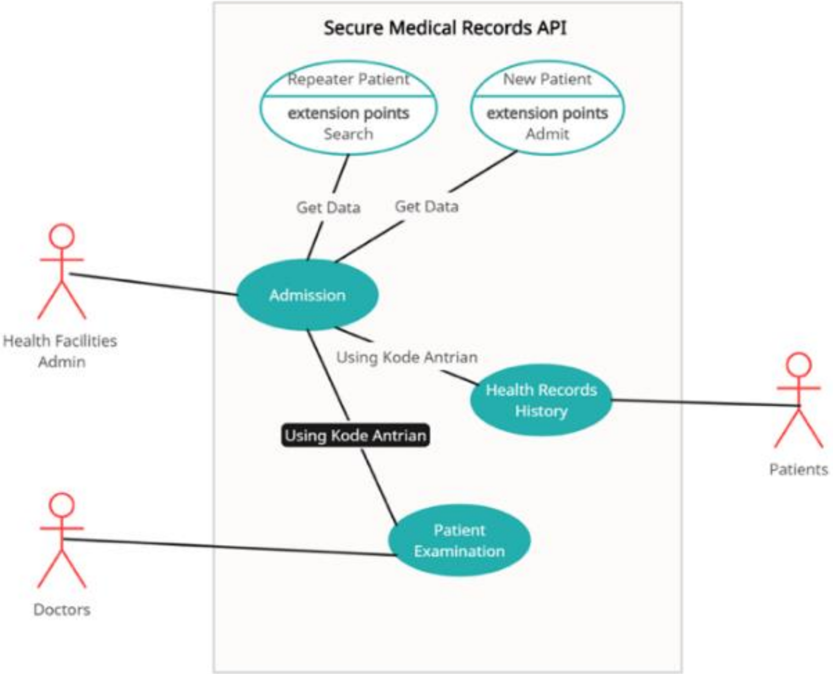


Figure 2. Secure API Use Case Diagram

2.2. Blackbox Testing

The purpose of testing is to validate that the software's functionalities, inputs, and outputs align with the specified requirements. Black Box Testing is a method used for testing software without examining its internal workings. This approach is favored as it is widely used for testing applications, and its implementation is straightforward. It enables software developers to generate input conditions that encompass all functional requirements. Such testing is critical to guarantee that the software functions in line with the design specifications[39], [40], [41].

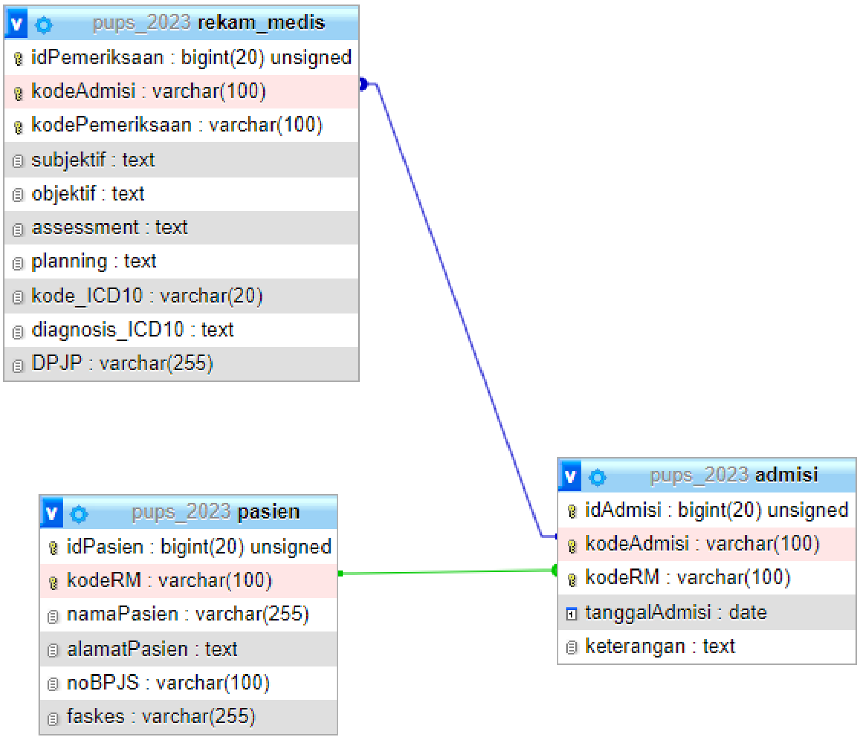
3. Results and Discussion

This section presents a comprehensive analysis of the Secure API development for electronic medical records interoperability. Additionally, it discusses the testing and analysis of the Secure API results in a scholarly manner.

3.1. Secure API Implementation

Based on Minister of Health Regulation No. 24 of 2022 concerning electronic medical records, the required variables can be described in the database design used to store medical record data. This database will later evolve into a medical record database that can be used to share data among healthcare institutions. The following is the database schema designed for this research :

p-ISSN 2088-1541
e-ISSN 2541-5832



pups_2023 rekam_medis
idPemeriksaan : bigint(20) unsigned
kodeAdmisi : varchar(100)
kodePemeriksaan : varchar(100)
subjektif : text
objektif : text
assessment : text
planning : text
kode_ICD10 : varchar(20)
diagnosis_ICD10 : text
DPJP : varchar(255)

pups_2023 admisi
idAdmisi : bigint(20) unsigned
kodeAdmisi : varchar(100)
kodeRM : varchar(100)
tanggalAdmisi : date
keterangan : text

pups_2023 pasien
idPasien : bigint(20) unsigned
kodeRM : varchar(100)
namaPasien : varchar(255)
alamatPasien : text
noBPJS : varchar(100)
faskes : varchar(255)

Figure 3. Database Schema of Electronic Medical Records

The database schema in Figure 3 consists of three core tables that work in tandem to form a patient's medical record. The medical record table houses SOAP (Subjective, Objective, Assessment, and Plan) data and ICD-10 (International Classification of Diseases, 10th Revision) diagnoses for each patient. The patient table contains personal details of the patients, while the admission table stores information about their admittance, such as where and when they received health services.

The admission table is linked to the Admission code as the primary key, the patient table with the RM code as the primary key, and the medical record table with the Admission code and Examination code as composite keys. This schema will serve as the primary database, collecting medical record data from patients across various healthcare facilities in Bali.

Once the database schema has been established, the API will be divided into three primary components: the patient API, admission API, and medical record API. Each API offers two key functions: retrieving and inserting data. The accompanying image illustrates the request process that users will follow, typically within their respective healthcare facility's internal system.

The API workflow image in Figure 4 illustrates that the API is designed to connect existing healthcare facilities in Bali. The primary requirement for patient medical record data to be connected is that each health service facility must already have a cloud-based electronic medical record system, which will be the system that shares data via this API.

The process that occurs is that the internal system of the health service facility makes a request, either to display the patient's medical record data or to store it. The patient medical record data in this API design consists of admissions, personal data, and patient examination results. This request will be validated by the API authenticator module, which will issue a success or failure response depending on the credentials sent in the request header. If all credentials are met, the API will return an encrypted response containing medical record data, which must be decrypted within each health service facility's internal system.

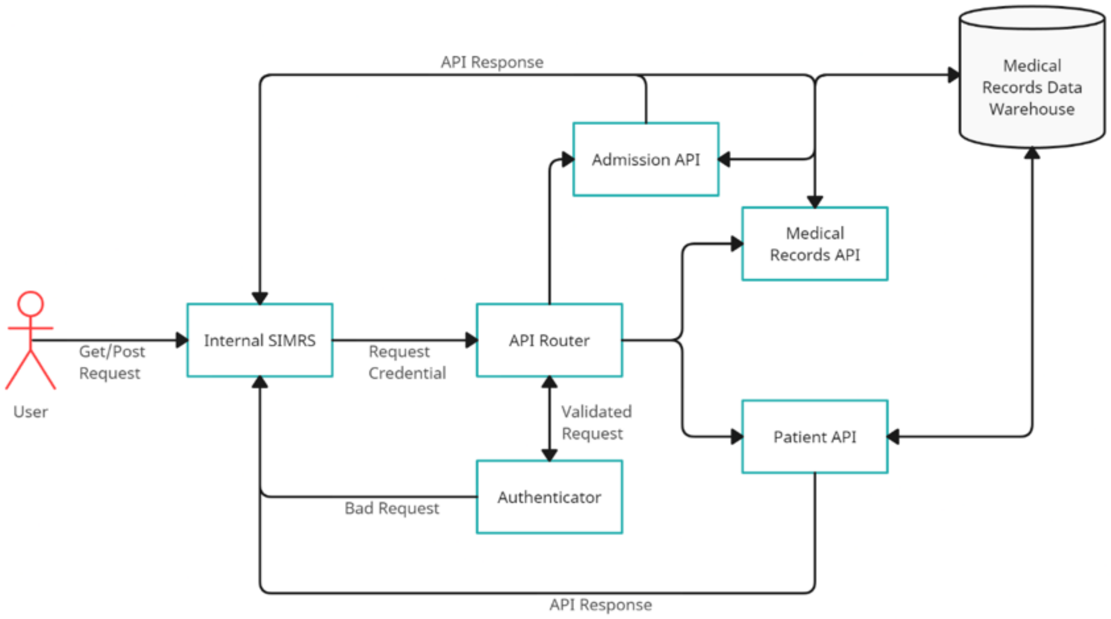


Figure 4. Secure API Workflow

3.2. Blackbox Testing

The designed API is tested in the initial stage using black box testing. The test includes: test credentials and test end-to-end encryption. The credential test is designed to verify whether the API has provided the correct response and whether the entered credentials are valid. The end-to-end encryption test verifies whether the data sent in response is encrypted and whether the client-side decryption module can perform decryption, ensuring that the data received by the user is complete and accurate medical record data. The testing is conducted on a cloud server, with endpoint https://tempatkitaproject.com/pups_testing/test/v1/. Table 2 explains the details of the testing scenarios that were run :

Table 2. Testing Scenario

Invalid API Key Scenario	
API Key	Invalid inserted
JSON Request Body	{"type":"pasien","flag":"insert","data":{"kodeRM":"PUPS\/UNI-000000","NIK":"5171630603890006","namaPasien":"Andy Nathaniel","alamatPasien":"Jalan Kartini Gang XII No. 52, Dauh Puri Kaja, Denpasar Utara","noBPJS":"618129461823866","faskes":"Klinik Bakti Rahayu"}}
HTTP Response	{"response_code":403,"message":"Invalid Credential","details":"WjNLKzlhZGhFYjZ0cTdPbFdxSHVtSmhu","time_stamp":1710497983}
Invalid JSON Body Request	
API Key	b733ed4837304f6dbc030ad7fdbe720a
JSON Request Body	{"flag":"insert","data":{"kodeRM":"PUPS\/UNI-000000","NIK":"5171630603890006","namaPasien":"Andy Nathaniel","alamatPasien":"Jalan Kartini Gang XII No. 52, Dauh Puri Kaja, Denpasar Utara","noBPJS":"618129461823866","faskes":"Klinik Bakti Rahayu"}}

HTTP Response	{"response_code":400,"message":"Missing \"type\"in request structure","details":"c1VNbXhmVjdURFRyY3ROZ1NxY1dXZE1T","time_stamp":1710498026}
Valid Request	
API Key	b733ed4837304f6dbc030ad7fdbe720a
JSON Request Body	{"type":"rekam_medis","flag":"insert","data":{"kodeAdmisi":"","kodePemeriksaan":"PUPS\/UNI-000000","tanggalPemeriksaan":"2024-03-15","waktuPemeriksaan":"17:47:44","subjektif":"SUBJEKTIF TESTING","objektif":"OBJEKTIF TESTING","assessment":"ASSESSMENT TESTING","planning":"PLANNING TESTING","kode_ICD10":"A00.0","diagnosis_ICD10":"Cholera due to Vibrio cholerae 01, biovar cholerae \u00b7 Certain infectious and parasitic diseases","DPJP":"DPJP TESTING"}}
HTTP Response	{"response_code":200,"message":"Proses Insert Pemeriksaan Berhasil","details":"d0RWTkppWERLbzJSVnBJOEhYRHVUbjhn","time_stamp":1710498081}
End-to-End Encryption Testing	
API Key	b733ed4837304f6dbc030ad7fdbe720a
JSON Request Body	{"type":"rekam_medis","flag":"get","data":{"kodePemeriksaan":"PUPS\/PMRK1-000003"}}
Encrypted Response	{"response_code":200,"message":"Data Successfully Retrieved","details":"Y25FcTNhYWpHY1hnRmZUc3VzUjV2QzlaYkMrNllGUHBhNzVEY1l0cFN1dUMrVytZNWw1YTdUZ0E2TlVMQzc3OGo3ZzcwM3A0Nmx5S2czRzErRlZYS1piZFIwdTZ1WXQ3VUlQejVuNzRpVEZRekdmeU1NaWdSMVVJL2xtcCs2bndUZFlqMlI1aVZ3ZHVGYnpSaCtDaDh1OE9UL0l5aEJIVU1OL0Z2ZkhReVZUWENraXJpWVJkSjNHUlZRTDFoZlJ2UkJKbDZMR2IjR1lWQXBKS0pZNlczNnpBWlpibGZ5ajFRbXp6bmNRazhzaTVON1Vjd3E3R0RRREdkd09uQ1dCZzkxTDdyQUZKcUJiOEcwbzgzWHRGaUtmOSt0MW9Jb2ZBZStaS2NEbS9yY0pjcldtZ3dXSFAxaU5mbGJ0aTRaZDhSakNMa1V6eFU5NTJlMGxZZXBkN1RueERJSFBVbTNGeHYya2hXT3o3bThQZEsxYWxhdExSY1pjZlZwV0RCRXI3a0V1N05mdVVTdytzZks5MHpaVkRicWVhejJ1aGxzOWdaOXFmOTR1TTAxdURiVUFsK2w5OE5Iekc0Um1CVG1KaDR1clVvZ0lDNWw0TnlpbUlmV3ZWclVWNktsOHE4dmY2SzRwcnpWZUNEM3luQ2FoU2JCZzBwMEF6TVZxMHQyZEgreTlZK0o1aUNVV1ErRVRIQlk1T2dCVjN3NmV1OW9ZUW05Q0d4cFhJUFZqY1MwbWlVQWNHUzBxb0pnTTRpdWYzNkFzQlArY2hOYjIxcnlBWlNMSzBSVzNrcjBzbG5ZVm9GN2FFNjVrQ3h0SFpMSExNUnp0MkM2YlpDLzVkYklvSFBwaHlYa1ZYTnY1cHhJOFhIS1I6VVkvZVh4NEJicTN5TUhPUmJIaHV3M011a0lTTlllSnRrV1lXd2ZEZmlHYXlUZHlWcml5eUs1aW9QU1lJbTFnQ0Z5dG8zMHNwTEJsOVo2Y3p5WW89","time_stamp":1710498961}
Decrypted Medical Record	{"response_code":200,"message":"Data Successfully Retrieved","details":{"idPemeriksaan":"5","kodeAdmisi":"PUPS/ADM1-000003","kodePemeriksaan":"PUPS/PMRK1-000003","tanggalPemeriksaan":"2023-10-14","waktuPemeriksaan":"16:40:48","subjektif":"SUBJEKTIF TESTING PUPS/PMRK1-000003","objektif":"OBJEKTIF TESTING PUPS/PMRK1-000003","assessment":"ASSESSMENT TESTING PUPS/PMRK1-000003","planning":"PLANNING TESTING PUPS/PMRK1-000003","kode_ICD10":"A00.0","diagnosis_ICD10":"Cholera due to Vibrio cholerae 01, biovar cholerae · Certain infectious and parasitic diseases","DPJP":"DPJP TESTING PUPS/PMRK1-000003"},"time_stamp":1710498961}

The testing results above indicate that all API scenarios have executed successfully. The expected functions are all running normally, indicating that the secure API has been successfully implemented.

In testing security features, such as API keys generated by the admin and end-to-end encryption by the decryption module installed at the health service facility, several crucial aspects were taken into account.

First, tests are conducted to ensure that each API key generated by the administrator is unique and is only provided to authorized parties. This is important to prevent unauthorized access to patient systems and data. Testing also includes validation of each API request to ensure that only valid and authorized requests are accepted by the system.

Second, end-to-end encryption utilizes the AES-CTR-256 cryptographic algorithm to safeguard data in transit and when stored in the database. Testing is conducted to ensure that the encryption and decryption processes operate smoothly without compromising system performance. Additionally, testing was conducted to assess the reliability of the decryption module in restoring encrypted data to its original form without compromising or altering the information.

During testing, various attack scenarios can also be considered, such as brute force attacks against encryption keys or attempts to access data unauthorizedly through API key manipulation. These tests help ensure that the security system can effectively address potential security threats.

The results of this testing demonstrate that the implemented API has been thoroughly tested and functions as expected, ensuring that patient data remains safe and protected.

Table 3. Comparison of Present Research with Previous Research

Features	Present Research	Previous Research
Platform	Multi-platform	Web based
Data Coverage	Cover multiple internal systems of healthcare facilities	Online cover specific healthcare facilities that use the system
Data Types	Includes subjective, objective, assessment, and planning (SOAP) data, in addition to ICD-10 diagnosis data	The data is not specified in SOAP, but already contains ICD-10 diagnosis data
Interface	Multiple interfaces can be implemented	Web-based, but already user-friendly
Security	More robust, with OWASP principles implemented	Not using the OWASP principle yet

4. Conclusion and Future Works

The discussion above demonstrates the successful implementation of a Secure API that incorporates ICD-10 diagnoses for patient medical record data. Health facilities in Bali can now utilize this API, tailor-made for collaborative use, allowing them to exchange patient medical record data seamlessly. This method is expected to enhance the accuracy and speed of patient treatment by providing comprehensive and instantaneous access to health history data.

The API has undergone black-box testing, which confirms that it provides appropriate responses for all request scenarios. Additionally, the end-to-end encryption module has been implemented effectively. However, it's worth noting that the API has only been tested within a black box setting thus far. It is hoped that the API can be tested in a native environment with security tests and real patient examination data input in the future.

References

[1] J. N. Olayinka, O. B. Akawa, E. K. Ogbu, A. T. Eduviere, R. I. Ozolua, and M. Soliman, “Apigenin attenuates depressive-like behavior via modulating monoamine oxidase A enzyme activity in chronically stressed mice,” *Current Research in Pharmacology and Drug Discovery*, vol. 5, 2023, doi: 10.1016/j.crphar.2023.100161.

[2] P. Ramos, “Compatibility studies of selected mucolytic drugs with excipients used in solid dosage forms: Thermogravimetry analysis,” *Farmacia*, vol. 69, no. 3, 2021, doi: 10.31925/farmacia.2021.3.22.

[3] W. Wang, C. Ren, H. Song, S. Zhang, and P. Liu, “FGL_Droid: An Efficient Android Malware Detection Method Based on Hybrid Analysis,” *Security and Communication Networks*, vol. 2022, 2022, doi: 10.1155/2022/8398591.

[4] M. L. Braunstein *et al.*, “The development and electronic delivery of case-based learning using a fast healthcare interoperability resource system,” *JAMIA Open*, vol. 2, no. 4, 2019, doi: 10.1093/jamiaopen/ooz055.

[5] K. D. Mandl, D. Gottlieb, and A. Ellis, “Beyond One-Off Integrations: A Commercial, Substitutable, Reusable, Standards-Based, Electronic Health Record-Connected App,” *Journal of Medical Internet Research*, vol. 21, no. 2, 2019, doi: 10.2196/12902.

[6] M. Mugisha *et al.*, “Integration of international classification of diseases version 11 Application Program Interface (API) in the Rwandan Electronic Medical Records (openMRS): Findings from two district Hospitals in Rwanda,” in *Studies in Health Technology and Informatics*, 2020. doi: 10.3233/SHTI200549.

[7] I. G. N. L. Wijayakusuma and S. C. Yowani, “WHO ICD-10 BASED ONLINE DISEASE DIAGNOSIS FOR GENERATING DIGITAL MEDICAL RECORD APPLICATION DEVELOPMENT,” *SINTECH (Science and Information Technology) Journal*, vol. 5, no. 1, 2022, doi: 10.31598/sintechjournal.v5i1.1040.

[8] J. M. Carson, P. Chen, and J. F. Outreville, “Foreign Direct Investment Affect the Supply of Life Insurance in Developing Countries ?,” *Journal of Insurance Issues*, vol. 44, no. 1, 2021.

[9] T. Chadaeva, “The Impact of Open Banking on the US Finance Industry,” *Russia and America in the 21st Century*, no. 3, 2019, doi: 10.18254/s207054760007179-3.

[10] V. Stefanelli and F. Manta, “Digital Financial Services and Open Banking Innovation: Are Banks Becoming ‘invisible’?,” *Global Business Review*, 2023, doi: 10.1177/09721509231151491.

[11] M. Ul Alam, M. A. K. Azad, and M. S. Ali, “Best Practices to Secure API Implementations in Core Banking System (CBS) in Banks,” in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference, CCWC 2022*, 2022. doi: 10.1109/CCWC54503.2022.9720840.

[12] A. Bris *et al.*, “KNIGHTS, RAIDERS, AND TARGETS - THE IMPACT OF THE HOSTILE TAKEOVER - COFFEE,JC, LOWENSTEIN,L, ROSEACKERMAN,S,” *Journal of Bank Finance*, vol. 37, no. 1, 2021.

[13] R. Buckley *et al.*, “Governing FinTech 4.0: BigTech, Platform Finance, and Governing FinTech 4.0: BigTech, Platform Finance, and Sustainable Development,” *Fordham Journal of Corporate & Financial Law*, vol. 27, no. 1, 2022.

[14] D. W. Arner, R. P. Buckley, K. Charamba, A. Sergeev, and D. A. Zetzsche, “BigTech and Platform Finance: Governing FinTech 4.0 for Sustainable Development,” *SSRN Electronic Journal*, 2021, doi: 10.2139/ssrn.3915275.

[15] N. A. Prasetyo and Y. Saintika, “Integration between Moodle and Academic Information System using Restful API for Online Learning,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 7, no. 2, 2021, doi: 10.26555/jiteki.v7i2.21816.

[16] K. Abouelmehdi, A. Beni-Hessane, and H. Khaloufi, “Big healthcare data: preserving security and privacy,” *Journal of Big Data*, vol. 5, no. 1, 2018, doi: 10.1186/s40537-017-0110-7.

[17] J. Kwon and M. E. Johnson, “Meaningful healthcare security: Does meaningful-use attestation improve information security performance?,” *MIS Quarterly*, vol. 42, no. 4, 2018, doi: 10.25300/MISQ/2018/13580.

[18] H. Huang, T. Gong, N. Ye, R. Wang, and Y. Dou, “Private and Secured Medical Data Transmission and Analysis for Wireless Sensing Healthcare System,” *IEEE Transaction on Industrial Informatics*, vol. 13, no. 3, 2017, doi: 10.1109/TII.2017.2687618.

[19] P. Sarosh, S. A. Parah, and G. M. Bhat, “An efficient image encryption scheme for healthcare applications,” *Multimedia Tools and Applications*, vol. 81, no. 5, 2022, doi: 10.1007/s11042-021-11812-0.

[20] Priyanka and A. K. Singh, “A survey of image encryption for healthcare applications,” *Evolutionary Intelligence*, vol. 16, no. 3, 2023, doi: 10.1007/s12065-021-00683-x.

[21] N. M. Hamed and A. A. Yassin, “Secure Patient Authentication Scheme in the Healthcare System Using Symmetric Encryption,” *Iraqi Journal for Electrical and Electronic Engineering*, vol. 18, no. 1, 2022, doi: 10.37917/ijeee.18.1.9.

[22] K. Munjal and R. Bhatia, “A systematic review of homomorphic encryption and its contributions in healthcare industry,” *Complex and Intelligent Systems*, vol. 9, no. 4, 2023, doi: 10.1007/s40747-022-00756-z.

[23] P. Sharma, N. R. Moparthi, S. Namasudra, V. Shanmuganathan, and C. H. Hsu, “Blockchain-based IoT architecture to secure healthcare system using identity-based encryption,” *Expert Systems,* vol. 39, no. 10, 2022, doi: 10.1111/exsy.12915.

[24] M. K. Hasan *et al.*, “Lightweight Encryption Technique to Enhance Medical Image Security on Internet of Medical Things Applications,” *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3061710.

[25] S. Das and S. Namasudra, “A Novel Hybrid Encryption Method to Secure Healthcare Data in IoT-enabled Healthcare Infrastructure,” *Computers and Electrical Engineering*, vol. 101, 2022, doi: 10.1016/j.compeleceng.2022.107991.

[26] A. Ali *et al.*, “An Industrial IoT-Based Blockchain-Enabled Secure Searchable Encryption Approach for Healthcare Systems Using Neural Network,” *Sensors*, vol. 22, no. 2, 2022, doi: 10.3390/s22020572.

[27] M. Singh and A. K. Singh, “Security of Medical Images Using a Key-Based Encryption Algorithm in the RDWT-RSVD Domain: SeMIE,” *Journal of Database Management*, vol. 34, no. 2, 2023, doi: 10.4018/JDM.318413.

[28] S. Schmeelk and L. Tao, “A Case Study of Mobile Health Applications: The OWASP Risk of Insufficient Cryptography,” *Journal of Computer Science Research*, vol. 4, no. 1, 2022, doi: 10.30564/jcsr.v4i1.4271.

[29] I. G. Anugrah and M. A. R. I. Fakhruddin, “Development Authentication and Authorization Systems of Multi Information Systems Based REst API and Auth Token,” *INNOVATION RESEARCH JOURNAL*, vol. 1, no. 2, 2020, doi: 10.30587/innovation.v1i2.1927.

[30] A. Elanda and R. L. Buana, “Analisis Keamanan Sistem Informasi Berbasis Website Dengan Metode Open Web Application Security Project (OWASP) Versi 4: Systematic Review,” *CESS (Journal of Computer Engineering, System and Science)*, vol. 5, no. 2, 2020, doi: 10.24114/cess.v5i2.17149.

[31] M. Mehrtak *et al.*, “Security challenges and solutions using healthcare cloud computing,” *Journal of medicine and life*, vol. 14, no. 4. 2021. doi: 10.25122/jml-2021-0100.

[32] B. Gao, F. Liu, S. Du, and F. Meng, “An OAuth2.0-Based Unified Authentication System for Secure Services in the Smart Campus Environment,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-319-93713-7_73.

[33] A. Chatterjee, M. W. Gerdes, P. Khatiwada, and A. Prinz, “SFTSDH: Applying Spring Security Framework with TSD-Based OAuth2 to Protect Microservice Architecture APIs,” *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3165548.

[34] N. A. J. De Witte *et al.*, “Online consultations in mental healthcare during the COVID-19 outbreak: An international survey study on professionals’ motivations and perceived barriers,” *Internet Interventions*, vol. 25, 2021, doi: 10.1016/j.invent.2021.100405.

[35] K. Bennett, A. J. Bennett, and K. M. Griffiths, “Security considerations for e-mental health interventions,” *Journal of Medical Internet Research*, vol. 12, no. 5, 2010, doi: 10.2196/jmir.1468.

[36] L. Yin, A. Zhang, X. Ye, and X. Xie, “Security-aware department matching and doctor searching for online appointment registration system,” *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2904724.

[37] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, “Security and Privacy in the Medical Internet of Things: A Review,” *Security and Communication Networks*, vol. 2018. 2018. doi: 10.1155/2018/5978636.

[38] S. K. Woody, D. Burdick, H. Lapp, and E. S. Huang, “Application programming interfaces for knowledge transfer and generation in the life sciences and healthcare,” *npj Digital Medicine*, vol. 3, no. 1. 2020. doi: 10.1038/s41746-020-0235-5.

[39] M. Y. P. Mahendra, I. N. Piarsa, and D. Putra Githa, “Geographic Information System of Public Complaint Testing Based On Mobile Web (Public Complaint),” *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, 2018, doi: 10.24843/lkjiti.2018.v09.i02.p04.

[40] N. M. D. Febriyanti, A. A. K. O. Sudana, and I. N. Piarsa, “Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen,” *Jurnal Ilmiah Teknologi dan Komputer*, vol. 2, no. 3, 2021.

[41] C. S. Kruse, B. Smith, H. Vanderlinden, and A. Nealand, “Security Techniques for the Electronic Health Records,” *Journal of Medical Systems*, vol. 41, no. 8, 2017, doi: 10.1007/s10916-017-0778-4.