iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82541

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Secure PHI Access Simulator Using Hybrid Encryption and RBAC

Manas Piyush[1], Mohammad Hasan[2], Md Suhail[3], Rahul Sardar[4], Dr. Bidisha Bhabani[5]
[1, 2, 3, 4]*Department of Computer Science and Engineering, JIS University, Agarpara, Kolkata—700109, India*
[5]*Assistant Professor, Dept. of CSE, JIS University*

Abstract: ***Digital patient record management has become standard in modern healthcare, yet widespread adoption has brought unprecedented challenges in maintaining the confidentiality of Protected Health Information (PHI). A fundamental architectural flaw persists across most deployed systems: access restrictions operate only at the user-interface layer, leaving the underlying stored data completely unprotected against direct database intrusion. This work proposes a cryptography-first design philosophy through the development of a web-based PHI Access Simulator. The simulator binds AES-256-GCM data encryption to RSA-2048 key encapsulation, with a Role-Based Access Control (RBAC) layer that governs key issuance based on verified job roles. Three access scenarios—clinical, administrative, and unauthorized—were systematically evaluated. Results confirm that cryptographic enforcement, rather than visual concealment, produces mathematically verifiable access boundaries that remain intact even under direct database compromise.***

Keywords: Protected Health Information (PHI); Electronic Health Records (EHR); AES-256-GCM; RSA-2048; Hybrid Cryptography; Role-Based Access Control (RBAC); Principle of Least Privilege; Data-Centric Security.

I. INTRODUCTION

Modern hospitals operate almost entirely on digital infrastructure. The convenience of electronic patient records accelerates clinical decision-making and facilitates interdepartmental coordination, but this connectivity simultaneously widens the attack surface available to malicious actors. Stolen medical records are particularly valuable on illicit markets because they consolidate insurance details, personal identifiers, and prescription histories in a single file—information that enables identity fraud far beyond what a compromised credit card allows.

Security architects have historically responded to these threats by fortifying the network boundary. Sophisticated firewall configurations, intrusion detection systems, and tiered authentication mechanisms constitute the traditional defensive posture. This approach carries a critical structural weakness: every protection it offers is rendered worthless the instant a threat actor gains authenticated access. A phishing attack that harvests a staff member's credentials, a session token intercepted over an unsecured connection, or a backup tape removed from a server room all produce the same outcome—unrestricted access to plaintext patient records.

The logical response to this vulnerability is to shift the protection mechanism from the network perimeter to the data itself. When PHI fields are cryptographically locked prior to storage, the possession of database access grants an attacker nothing more than ciphertext. This paper documents the design and experimental validation of a browser-based simulator that applies exactly this principle, coupling RBAC-governed key distribution with hybrid cryptographic protection of patient data at rest.

II. OBJECTIVES

This project addresses three interrelated engineering goals:

1) Design a clinically realistic hospital-dashboard prototype in which PHI confidentiality is enforced through cryptographic key control rather than through application-layer visibility rules.

2) Validate the Principle of Least Privilege by constructing a formal RBAC permission matrix that translates occupational boundaries into cryptographic access boundaries, provably limiting each role to its minimum required data.

3) Implement and demonstrate a two-algorithm cryptographic pipeline in which AES-256-GCM handles high-throughput PHI encryption while RSA-2048 manages session-key encapsulation, operating entirely within a standard browser environment.

III. LITERATURE REVIEW

The evolution of healthcare data security can be traced through three identifiable phases, each representing a qualitative shift in the threat model and the defensive response.

A. *Network-Perimeter Dominance*

The earliest digital health systems adopted security architectures borrowed wholesale from general-purpose enterprise computing. Perimeter firewalls, role-separated network segments, and credential-based authentication formed the standard protective layer. Regulatory frameworks such as HIPAA codified these controls into legal requirements. Despite compliance, subsequent research repeatedly exposed a foundational weakness: credential theft through social engineering grants adversaries access that is indistinguishable from legitimate user activity, after which no further barrier impedes database access [1].

B. *Cryptographic Approaches and the Hybrid Consensus*

Recognition of perimeter limitations directed academic attention toward protecting the data directly. Symmetric algorithms, most prominently AES standardized under NIST FIPS 197 [3], deliver the throughput needed for large medical datasets but introduce a key-distribution problem when deployed across heterogeneous clinical teams. Asymmetric algorithms, particularly RSA [4], resolve distribution elegantly through public-key infrastructure but impose computational costs that make direct encryption of large payloads impractical. The resolution that has since become standard in the literature is hybrid encryption: AES protects the bulk payload at speed, while RSA secures only the compact symmetric key, preserving the advantages of both paradigms [3][4].

C. *Role-Based Governance and Least Privilege*

Assigning data permissions to named individuals proved administratively unsustainable in large organizations. The NIST RBAC standard (INCITS 359-2004) [2] introduced role abstraction as the governing principle: permissions attach to occupational roles, and individuals inherit those permissions through role assignment. This model maps naturally onto hospital organizational structures where job functions are well-defined and stable. Contemporary EHR security literature consistently identifies RBAC as the optimal access-governance framework for healthcare environments [1][2].

IV. RESEARCH GAP

A systematic examination of production EHR deployments reveals a persistent discrepancy between the security models recommended in the literature and those actually implemented. The prevailing commercial pattern exercises access control at the presentation tier: role verification determines which data fields are rendered in the user interface, while the storage tier retains all records as unprotected plaintext. This design hides data rather than protecting it. The practical consequence of this gap is that any compromise that bypasses the front-end application—a SQL injection attack, an administrator credential leak, a stolen database backup, or a malicious insider with direct query access—immediately yields all stored patient data without restriction. The application's role enforcement becomes entirely irrelevant once an adversary is operating at the database tier. This project addresses that gap by demonstrating that access control must be applied at the storage tier through cryptographic key governance. When decryption material is issued conditionally on verified role membership, unauthorized data access remains computationally infeasible regardless of the attack vector used to reach the database, because the encryption layer is structurally independent of the application layer.

V. PROPOSED METHODOLOGY

The simulator architecture integrates two interdependent modules that together enforce access control at the cryptographic level.

A. *Hybrid Cryptographic Module*

PHI fields designated as high-sensitivity—clinical diagnosis and physician notes—are transformed into ciphertext using AES-256-GCM before being committed to storage. This algorithm was selected on three grounds: a 256-bit key length positions the system well beyond the reach of foreseeable brute-force attacks; the Galois/Counter Mode of operation appends an authentication tag to every encrypted block, enabling detection of any post-encryption tampering; and its stream-cipher-like throughput makes it suitable for the data volumes characteristic of healthcare record systems.

Each AES session key is subsequently encrypted under a 2048-bit RSA public key, yielding a compact encrypted key blob that is stored alongside the ciphertext record. Reconstruction of the plaintext requires the corresponding RSA private key, which the RBAC module releases only under prescribed role conditions.

B. RBAC Authorization Module

Three role categories are defined in the access-control matrix, each carrying distinct key-issuance rules:

1) Clinical Staff (Nurse) — The RSA private key is issued upon role verification. AES-GCM decryption proceeds on all PHI fields. Financial billing data is excluded from the clinical view in accordance with the Least Privilege requirement.
2) Administrative Staff (Billing Clerk) — Billing-tier fields are returned in plaintext. The RSA private key is withheld; clinical ciphertext is forwarded annotated with a DECRYPTION BLOCKED status flag.
3) Unrecognized User — The RBAC evaluation finds no matching role entry and terminates the pipeline immediately. A 403 Forbidden response is returned with zero data exposure across all sensitivity tiers.

C. Integrated Request Processing

Each access request traverses a fixed sequence: role identification and matrix evaluation by the RBAC module; conditional RSA key issuance or denial; AES-GCM decryption for authorized clinical requests or ciphertext passthrough with status annotation for administrative requests; and controlled data delivery to the view layer with visual theming that reflects the access outcome.

VI. EXPERIMENTAL SETUP

A. Implementation Platform

The prototype was realized as a single self-contained HTML file to eliminate infrastructure dependencies and maximize reproducibility across environments. The implementation stack comprised standard HTML5 for dashboard structure and vault-status display; Tailwind CSS (CDN delivery) for responsive layout and color-coded role feedback; and JavaScript, which hosts the RBAC logic, the AES-256-GCM and RSA-2048 cryptographic simulation, and the dynamic DOM rendering of decrypted or blocked output.

B. Synthetic Patient Dataset

A representative patient record was constructed for Manas Piyush (Patient ID: 47392) and partitioned into two sensitivity tiers. The open-access tier contained demographic and vital data—blood pressure 120/90, temperature 98.6°F, billing code EMS356, and a charge of
₹2,000.00—fields whose exposure carries limited re-identification risk. The encrypted PHI tier comprised a clinical diagnosis of Severe Asthma (High Priority) and physician monitoring notes, both stored exclusively as AES-256-GCM ciphertext in the simulated vault.

C. Baseline Condition

Prior to any role activation, the dashboard displays the record in its data-at-rest state: open-access fields are visible in plaintext while PHI-tier fields appear as base64-encoded ciphertext. This baseline confirms that encryption is active before any access event occurs, establishing the protective state that the RBAC module conditionally relaxes.

VII. SYSTEM ARCHITECTURE AND DATA FLOW

Every access request is routed through three sequential processing stages, illustrated in Fig. 1.

Stage 1 – RBAC Validation: The inbound role token is evaluated against the access-control matrix. Unrecognized tokens terminate the pipeline at this stage with a 403 response; recognized tokens generate an access-granted signal that advances the request to Stage 2.

Stage 2 – Hybrid Decryption: For clinical roles, the RSA-2048 private key is issued to the decryption routine, which unwraps the encrypted AES session key and applies AES-256-GCM decryption to the PHI payload. For administrative roles, the RSA key is withheld and the ciphertext is forwarded with a BLOCKED status flag.

Stage 3 – Controlled Data View: The output layer renders plaintext fields or annotated ciphertext according to the status flag received from Stage 2. Dashboard coloring—green for clinical access, amber for administrative access, red for denial—provides immediate visual confirmation of the access outcome.

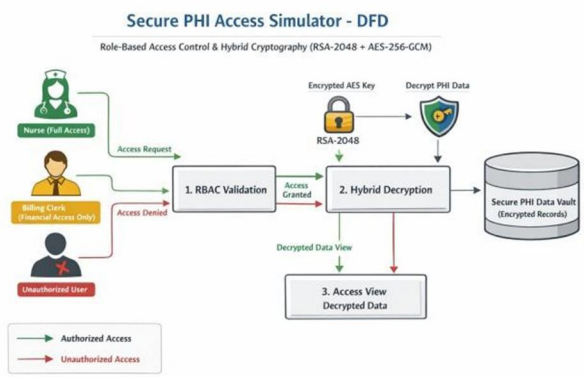


Fig. 1: Secure PHI Access Simulator — Data Flow Diagram (DFD)

VIII. EXECUTION PROCEDURE

The simulator operates without a web server, database engine, or compiled runtime. The four-step execution procedure is as follows:

1) Step 1 – Preparation: Save the simulator source as simulator.html on any local machine running a contemporary operating system.

2) Step 2 – Launch: Open simulator.html in a current browser (Chrome v90+, Edge v90+, Firefox v88+, or Safari v14+). An internet connection is required for Tailwind CSS rendering; the cryptographic logic functions offline.

3) Step 3 – Baseline Verification: Before activating any role, inspect the PHI-tier fields to confirm ciphertext is displayed, establishing the encrypted baseline state.

4) Step 4 – Role Simulation: Activate each of the three role buttons in sequence and document the access-output panel response for each scenario.

IX. RESULTS AND ANALYSIS

A. Clinical Role (Nurse)

Activating the Nurse role caused the RSA private key to be issued to the decryption routine. The encrypted AES session key was successfully unwrapped, and AES-256-GCM decryption was applied to the PHI payload. Both protected fields—the Severe Asthma diagnosis and the physician monitoring notes—were rendered in readable form within a green-bordered output panel. The billing code field was absent from the clinical view, confirming that Least Privilege applies symmetrically: clinical staff are restricted from financial data just as administrative staff are restricted from clinical data.

B. Administrative Role (Billing Clerk)

Activating the Billing Clerk role returned the patient name, identifier, and billing charge (₹2,000.00) as plaintext. The RSA private key was not released for this role; the clinical ciphertext was passed through to the view layer annotated with an RSA/AES DECRYPTION BLOCKED status. The amber-themed output panel confirmed partial, role-bounded access, demonstrating that a legitimate credentialed user cannot breach data boundaries defined in the RBAC matrix.

C. Unauthorized User

Activating the Unauthorized User role produced an immediate 403 Forbidden response. The RBAC evaluation found no matching matrix entry and halted the pipeline before any data—including fields classified as open-access—was released. The red-flashing dashboard confirmed that the system defaults to complete data withholding rather than partial disclosure when confronted with an unrecognized identity.

D. PHI Simulator — AES Model vs PHI Simulator — Hybrid Model

Fig. 2 provides a structured comparison between the PHI Simulator — AES Model—the foundational version employing single-layer AES-GCM encryption—and the PHI Simulator — Hybrid Model—the advanced version incorporating a full hybrid pipeline of AES-256-GCM and RSA-2048. The comparison quantifies improvements across six technical dimensions and visualizes the security profile enhancement through a radar analysis.

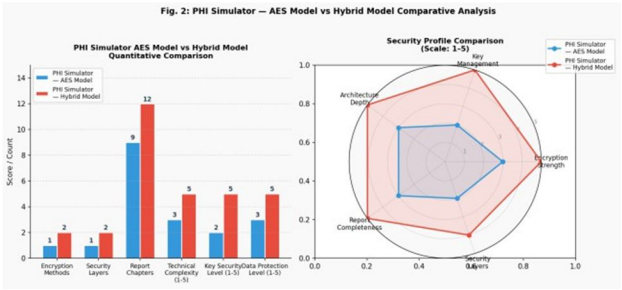


Fig. 2: PHI Simulator — AES Model vs PHI Simulator — Hybrid Model — Comparative Analysis

X. FUTURE SCOPE

Several enhancements are required before this architecture could be deployed in a production clinical environment:

1) Secure Key Infrastructure: RSA private keys must be held in a Hardware Security Module (HSM) or a secrets management platform such as HashiCorp Vault, rather than being exposed to browser-side JavaScript.

2) Persistent Storage Layer: Integration with an encrypted relational or document-oriented database (PostgreSQL or MongoDB) to store and retrieve thousands of patient records durably across sessions.

3) Multi-Factor Identity Verification: Supplementing role assertion with time-based one-time passwords or hardware authentication tokens to strengthen identity assurance before key issuance.

4) Immutable Audit Trail: Recording every access event—successful or denied—to an append-only log structure to support regulatory compliance, forensic investigation, and anomaly detection.

5) Managed Cloud Deployment: Hosting the system on HIPAA-eligible infrastructure (AWS, Google Cloud, or Azure) with encryption-at-rest for storage volumes and TLS 1.3 for all data in transit.

6) Emergency Override Mechanism: A break-glass workflow permitting temporary escalated access during life-critical emergencies, with mandatory alerting and post-event review to prevent abuse.

XI. CONCLUSION

Healthcare data security cannot rely on visual concealment as its primary mechanism. When PHI records are stored in unprotected plaintext, any breach that reaches the database layer—regardless of the sophistication of the surrounding perimeter controls—immediately yields complete access to every patient record in the system. This structural vulnerability is not an implementation failure; it is the inevitable consequence of an architecture that conflates hiding data with protecting data.

The PHI Simulator — Hybrid Model presented in this paper demonstrates an alternative architecture in which protection is embedded in the data itself. By issuing RSA-2048 decryption keys exclusively to verified RBAC roles, the system ensures that clinical PHI remains computationally opaque to any party—internal or external—who does not hold the appropriate role-assigned credentials. The experimental results confirm that this guarantee is preserved across three distinct access scenarios: a nurse obtains full clinical access, a billing clerk is cryptographically confined to financial data, and an unauthorized user is denied any data whatsoever.

This work contributes a practical architectural template for organizations seeking to advance beyond perimeter-centric security toward a data-centric security posture—a transition that grows increasingly urgent as both the sophistication of external attacks and the frequency of insider-threat incidents continue to rise.

REFERENCES

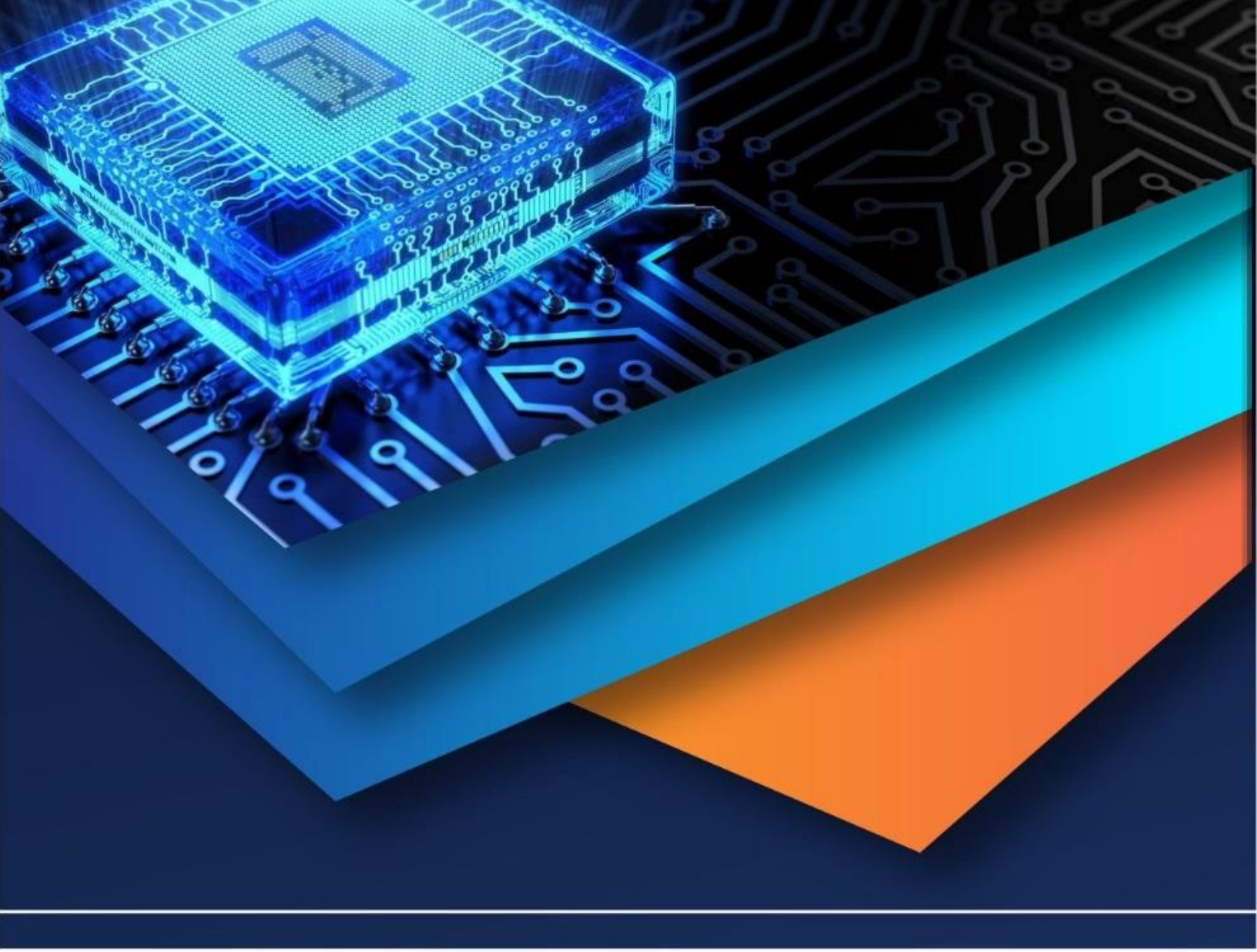
[1] U.S. Department of Health and Human Services, “Summary of the HIPAA Security Rule,” Office for Civil Rights, Washington, D.C. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/security/index.html

[2] National Institute of Standards and Technology, “Role-Based Access Control,” ANSI INCITS 359-2004, Gaithersburg, MD: NIST, 2004.

[3] National Institute of Standards and Technology, “Announcing the Advanced Encryption Standard (AES),” Federal Information Processing Standards Publication 197, Gaithersburg, MD: NIST, Nov. 2001.

[4] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Communications of the ACM, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[5] Mozilla Developer Network, “Web Cryptography API,” MDN Web Docs. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Crypto_API



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY