JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2026, VOL 9, NO 2

Security Challenges In Healthcare Cloud Apis: A Systematic Review

Brahmanand Reddy Bhavanam

Info Way Solutions LLC, USA

Abstract
Cloud computing has a substantial influence on the healthcare ICT infrastructure because of its flexible, scalable, and cost-effective features; it also plays an important role in electronic health record management, clinical workflow, and the interoperability of disparate healthcare systems. The use of cloud-based Application Programming Interfaces in a healthcare system raises security and compliance risks due to sensitive protected health information and strict data protection requirements. This systematic review describes the various privacy and security challenges and vulnerabilities associated with healthcare cloud application programming interfaces and identifies the most important security areas that need consideration, including authentication protocols, data encryption protocols, and secure data transmission protocols. This article also explains the fundamental building blocks of healthcare cloud APIs and reviews their unique privacy and security challenges for real-time access and interoperability, as well as informed consent workflows. The article analyzes the potential attack surfaces for healthcare cloud APIs, such as man-in-the-middle attacks, distributed denial of service, and unauthorized access, and their impact on healthcare operations and patient safety. Additional threats include issues with verifying and managing medical device access (like credential management, MFA, token-based authentication, role-based access control, and attribute-based access control), using encryption for stored and transmitted data, managing encryption keys, and how end-to-end encryption affects performance in complex systems. Existing security standards include general and sector-specific recommendations for security hygiene, such as security-by-design, continuous security monitoring, incident reporting, and regulatory compliance. Emerging security concerns for APIs include artificial intelligence for threat detection, the adoption of zero-trust architecture, and the use of quantum-resistant encryption mechanisms in light of quantum computing developments. Recommendations are made for healthcare organizational leadership, cloud service providers, and policymakers to improve API security posture, prompt security innovation and address security challenges while maintaining operational efficiency and privacy compliance.

Keywords: Healthcare Cloud Apis, Authentication Vulnerabilities, Data Encryption, Api Security Frameworks, HIPAA Compliance.

1. Introduction

The adoption of cloud computing platforms is a game changer in the overall IT infrastructure landscape in healthcare. Factors such as operational flexibility, resource utilization, and cloud elasticity are driving the market forward. Healthcare institutions can use these infrastructure models to work around the limitations

of on-premises hardware. They can exploit elastic provisioning, interoperability between previously separate clinical systems, and the ability to store and process the huge quantities of new health data now being generated by electronic health records, medical imaging systems, and connected medical devices. With this technological progress, a new security model for Application Programming Interfaces is emerging. Characteristics of cloud APIs, such as being exposed to networking (computing protocols), the use of distributed authentication, and the many mission-critical healthcare use cases that they cater to, create a different attack surface from that of in-house software and systems, many of which are legacy applications running in enterprise data centers. Combining private health information with a publicly available public-facing interface merits serious consideration of the security aspects of the introduction of cloud-based APIs. Health information is valuable to attackers because it is archived, holistic, and applicable in cases of identity theft and fraud in medical insurance [2].
We carried out a systematic review of the security issues related to healthcare cloud application programming interfaces (APIs). This review addresses three security domains that constitute the foundation of secure healthcare API design and implementation. The first domain is Authentication: the most primary access control mechanism to health-related resources, with different architectural patterns, protocol choices, and design decisions rendering it more or less effective for protecting healthcare resources against unauthorized access while not obstructing legitimate clinical workflows. The second domain has been described as Data Encryption Standards that include the mechanism to select encryption algorithms and key management to be used in providing the confidentiality and integrity of required protected health information throughout its lifecycle in cloud computing environments. The third domain is Secure Transmission Protocols, which are the technical protocols and communication security technology layers that are used to protect data in transit between the healthcare system and the cloud. The review focuses on these three related domains, which acknowledge being a part of broader security domains and have a good balance of depth of technical analysis while being relevant to health organizations to implement in practice.
Healthcare cloud API security exists in the context of a complicated regulatory environment of privacy and security legislation, which creates requirements and restrictions on those who hold protected health information (PHI) and therefore defines the technical and operational environment in which they provide services. In the United States, the Health Insurance Portability and Accountability Act (HIPAA) mandates a thorough set of privacy and security requirements that govern the manner in which healthcare organizations and their business associates handle electronic protected health information (ePHI) and their associated legal obligations. The EU General Data Protection Regulation specifies comparable obligations for data subjects, transparency about processing, and stricter consent. It affects how healthcare APIs use cross-border data flows and user access rights. The US Health Information Technology for Economic and Clinical Health Act strengthens enforcement of healthcare privacy rules and extends security rules to other areas of the rapidly growing digital health environment, such as cloud computing and health information exchanges. Cross-regulatory and pan-regulatory requirements create a compliance baseline that extends beyond technical implementation. Healthcare organizations must be able to provide evidence of technical safeguards as well as recordkeeping requirements and security testing in case of possible data breach notifications. This regulatory backdrop makes security a requirement for APIs, but it must both technically reduce threats and comply with regulatory demands by enabling security designs that can be audited and verified while still supporting the operational needs of clinical care delivery.

2. Healthcare Cloud API Architecture and Threat Landscape

2.1 Cloud API Architecture in Healthcare Systems

Healthcare API infrastructure may be implemented as layered presentation, business logic, and data persistence architectures. A presentation layer can be either a direct interface to clients or an API gateway service. The business logic layer performs clinical workflows, including authorizations to access protected health information in a data persistence layer, which is distributed in a systematic way. Through centralized routing, cross-domain security policy, and protocol translation between heterogeneous systems, an API gateway architecture serves as the entry point for requests. Back-end services provide domain-specific functionality such as clinical data, patient identity resolution, and clinical decision support [3]. The integration points of this architecture usually consist of a variety of health information systems, such as electronic health record systems, health information exchange systems, laboratory information systems, radiology information systems, and pharmacy management systems, each of which contributes specific data elements related to health that need to be aggregated, normalized, and made available through a standard set of application programming interfaces (APIs). The architecture requirement of maximum data privacy distinguishes healthcare from other domains. Getting data in real time without delays because of proprietary concerns, could affect clinical results. Therefore, there are specific API architectures meant to support a reduced time latency (i.e., the time delay between a request and a response) and a high level of availability. Performance optimizations must not compromise security controls, and the implementation of interoperability standards such as Health Level Seven (HL7) and Fast Healthcare Interoperability Resources (FHIR) creates structural constraints on an API specification that defines the data models, resources, and methods for exchanging information across institutional boundaries. In healthcare, consent management systems must handle complicated consent processes and follow data protection rules, which include checking a patient's consent in real time when an API call is made, keeping records of actions taken, following rules about how long to keep data, and adapting security measures as needed.

2.2 Primary Threat Vectors and Attack Surfaces

Another threat model for healthcare cloud API communications is man-in-the-middle attacks. Because health information data must be transferred over a network, attacks could be performed to place themselves as a man-in-the-middle between the real healthcare entities and the cloud service endpoints to view, collect, alter, or delete sensitive health information while in transit. Threat actors would focus on how effectively transport layer security is configured, the strength of certificate validation, and how endpoint validation is configured. Health organizations that do not implement mutual authentication, accept self-signed certificates, or support backwards compatibility with deprecated and obsolete algorithms in legacy medical devices are vulnerable to this attack. Session hijacking scenarios could occur due to weaknesses in the generation, transmission, and verification of session tokens, which could allow an attacker with access to authentication credentials to impersonate a legitimate user to manipulate patient data without knowing the account passwords. DDoS attacks on APIs hosted in the cloud could put the availability of these services at risk, which healthcare delivery systems rely on for timely patient data access during office visits. DDoS attack vectors leverage the finite nature of computation and network resources by artificially generating requests that exceed the defined threshold for the targeted resource. At the application layer, resource exhaustion attacks can also take place, targeting resource-intensive API requests like expensive database calls or cryptographic operations. Attackers can leverage these expensive operations to reduce the threshold of requests that need to be sent to exhaust backend services or delay overall application performance. Attempts to gain unauthorized access to a healthcare cloud API may employ multiple hacking techniques to bypass authentication and authorization controls and mechanisms. These may include exploiting a weakness in the API's implementation of the identity verification logic, social engineering legitimate users to share their credentials, and all other techniques that circumvent the credential verification step of authentication. Privilege escalation vulnerabilities include the ability of an attacker to manipulate an application's authorization logic flaws, with their authorized credentials, to have access to more data or functionality than they are entitled to. Insider threat vulnerabilities are unique to the healthcare industry, where numerous clinicians are authorized users of API-provided patient data accessed in the course of their care. It is possible for them to be abused by insider threats (such as requesting temporary access to a resource and then using that access for their gain or to exfiltrate data) and may be difficult to detect, as normal and abnormal behavior can be highly contextual depending on use cases, patient conditions, or emergency access situations.

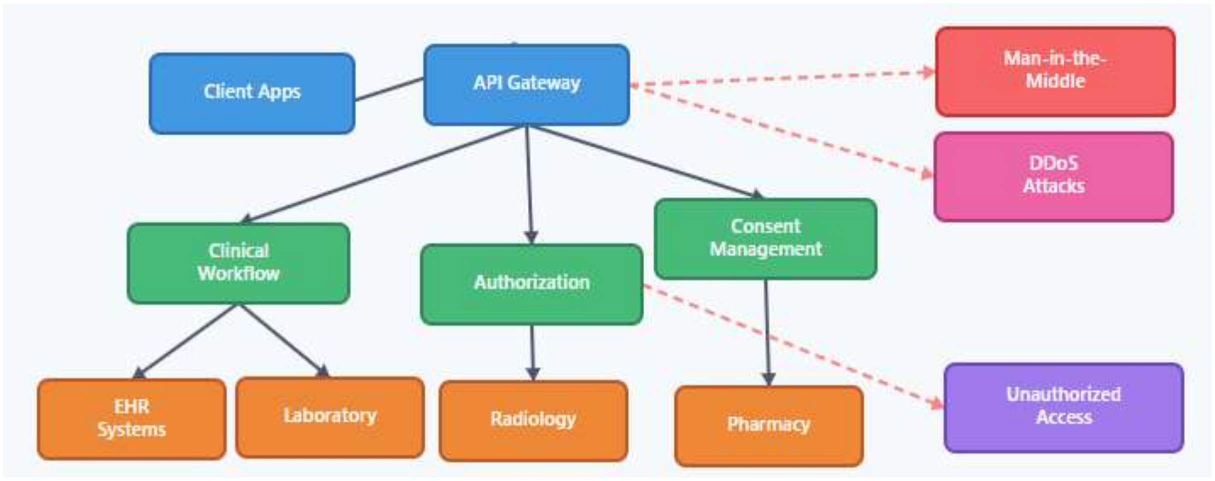


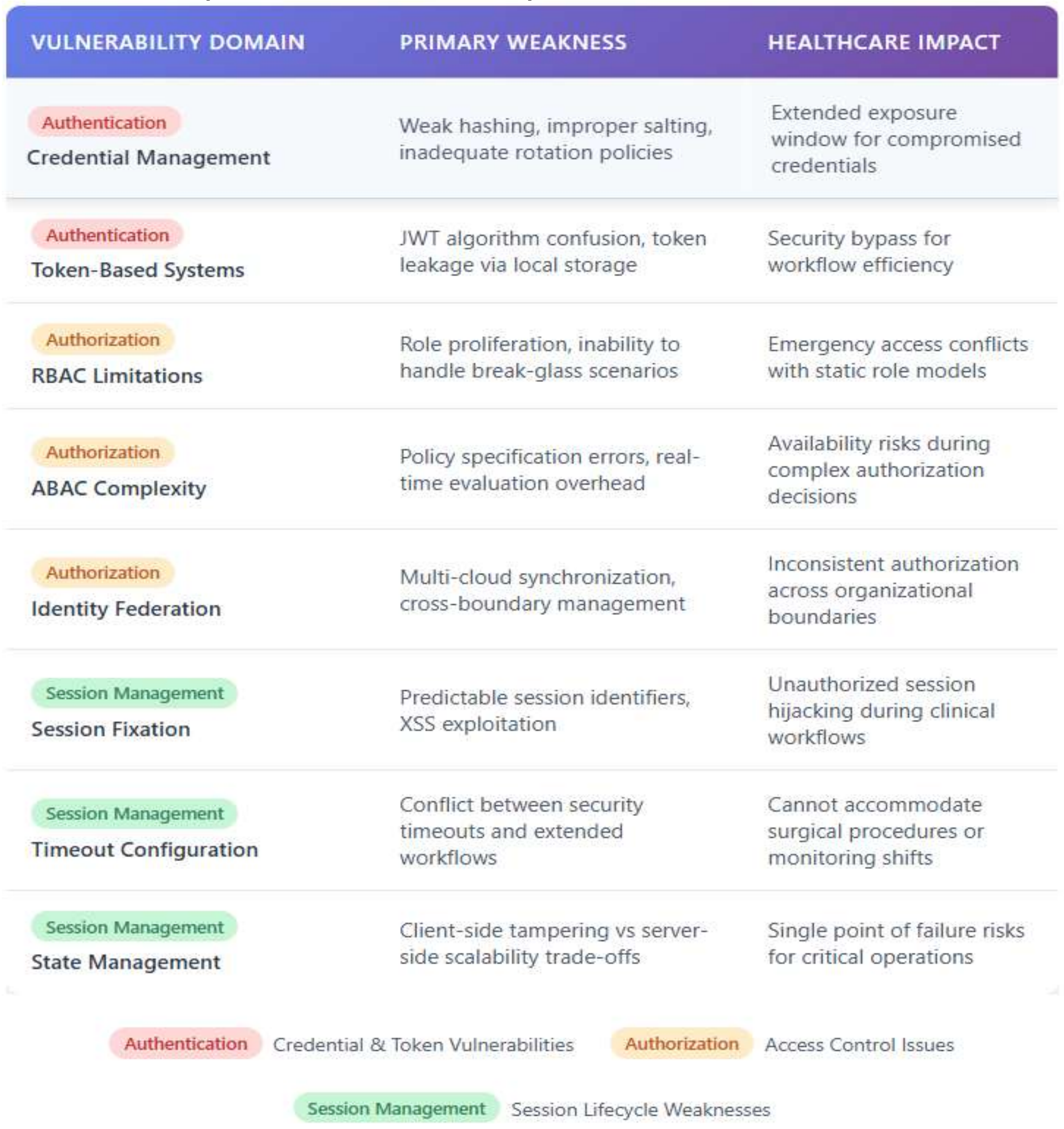
Fig 1: Healthcare Cloud API Architecture and Threat Vector Mapping [3, 4]

3. Authentication and Access Control Vulnerabilities

Healthcare cloud APIs are vulnerable to weak credential management practices, poor multi-factor authentication implementation, and architectural weaknesses in token-based authentication systems. Credential weaknesses include password composition rules, improper credential storage, and inadequate credential recycling. Also, problems like easy-to-guess password rules, poor protection of saved user passwords (without adding extra security), and slow updates to passwords that leave stolen login details exposed for longer are widely known issues. In healthcare, multi-factor authentication can also create challenges of interoperability because of a tension between security compliance on one hand and clinician workflow on the other. Some token-based authentication implementations that are based on the JSON Web Token specifications are vulnerable to algorithm confusion attacks, where the attacker can modify the token header to downgrade the algorithm from an asymmetric to a symmetric algorithm, creating a valid token because the symmetric key was shared in advance between the two ends. Such confusion occurs, for example, when clinicians switch off the security safeguards for the sake of expediting patient care. Token leakage can happen when authentication tokens are stored in the browser's local storage, which can be attacked through cross-site scripting; when tokens are sent over unsafe connections, making them easy to intercept; and when users interact with application monitoring systems that log tokens without hiding them.

Healthcare cloud API authorization frameworks still have the limitations of customary access control methods. Furthermore, identity management across federated multi-cloud infrastructures across organizations and jurisdictional boundaries remains challenging. However, role-based access control systems make it easier for administrators by allowing them to manage permissions based on user roles instead of individual users, which cuts down on the number of decisions needed for access in healthcare cloud situations. This proliferation is managed poorly in the healthcare domain because the clinical roles are both fine-grained and dynamic, and legitimate access is context-sensitive. Break-glass scenarios, where clinical care requires emergency access to patient information, do not align with the standard RBAC roles. Temporary privilege delegation scenarios, when another clinician covers for a clinician for a patient who is not normally in their scope, do not fit either. Research scenarios, in which access decisions are based on approved protocols, also do not fit. ABAC models address many of the limitations of RBAC. They allow for fine-grained access control that considers a user's attributes, a resource's attributes, and the environment's conditions to determine whether a user can access a resource. However, ABAC has its own limitations, including the complexity of the policy specification, difficulties with attribute provisioning, and availability issues due to the need for real-time evaluation of complex policy logic over dynamic collections of attributes.

Session management security for healthcare cloud application programming interfaces (APIs) is the technical and policy-related use of programming methodologies and practices for managing the session lifecycle from session establishment to session termination. An example of a session security attack is the session fixation attack, which exploits vulnerabilities in session identifier generation and assignment mechanisms. An attacker can use social engineering or cross-site scripting to manipulate the web application to create a session with a predictable session ID, and wait for the user to log in to the web application. Session timeout policies are constrained by a conflict between security principles, which require short session lifetimes to minimize the exposure time for session hijacking, and functionality that requires long sessions to enable extended clinical workflows such as surgical procedures or patient monitoring shifts. Other considerations of session security include session state management, which must consider trade-offs between session state stored on the client, where it is accessible to tampering even with strong cryptographic protection, versus that stored on the server, where it is subject to scalability and to a single point of failure. The API must consider how to address interactive user sessions from clinicians, programmatic sessions from background jobs, and mobile clinical workflows with limited network availability and inconsistent connectivity.



VULNERABILITY DOMAIN	PRIMARY WEAKNESS	HEALTHCARE IMPACT
Authentication Credential Management	Weak hashing, improper salting, inadequate rotation policies	Extended exposure window for compromised credentials
Authentication Token-Based Systems	JWT algorithm confusion, token leakage via local storage	Security bypass for workflow efficiency
Authorization RBAC Limitations	Role proliferation, inability to handle break-glass scenarios	Emergency access conflicts with static role models
Authorization ABAC Complexity	Policy specification errors, real-time evaluation overhead	Availability risks during complex authorization decisions
Authorization Identity Federation	Multi-cloud synchronization, cross-boundary management	Inconsistent authorization across organizational boundaries
Session Management Session Fixation	Predictable session identifiers, XSS exploitation	Unauthorized session hijacking during clinical workflows
Session Management Timeout Configuration	Conflict between security timeouts and extended workflows	Cannot accommodate surgical procedures or monitoring shifts
Session Management State Management	Client-side tampering vs server-side scalability trade-offs	Single point of failure risks for critical operations

Authentication	Credential & Token Vulnerabilities
Authorization	Access Control Issues
Session Management	Session Lifecycle Weaknesses

Table 1: Authentication and Access Control Vulnerability Summary [5, 6]

4. Data Encryption and Secure Transmission Protocols

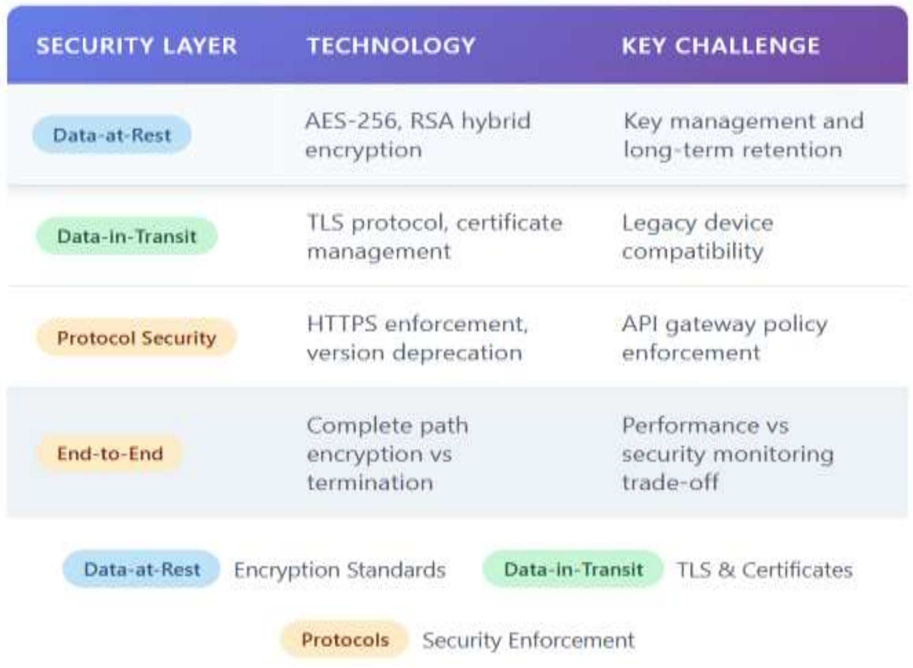
4.1 Encryption Standards and Implementation

Data-at-rest encryption for healthcare cloud computing is the employment of a cryptographic algorithm to protect protected health information from unauthorized exposure in the event that physical security or storage media controls have been circumvented. This method can subsequently be deployed as a compensating control to transform that data into a form unintelligible without knowledge of decryption keys. The Advanced Encryption Standard (AES) algorithm, which uses a 256-bit key, has become the main choice for securely encrypting healthcare data because it works efficiently, is hard to break, and is supported by laws and security guidelines. Rivest-Shamir-Adleman asymmetric encryption is the most widely used algorithm for public key cryptography, including hybrid encryption schemes that rely on RSA to encrypt symmetric keys that encrypt bulk data. Managing keys securely in hybrid encryption keeps the encryption and decryption processes separate; similarly, asymmetric encryption is used to share and protect data when multiple authorized parties need to access and decrypt it without sharing the secret key. Cryptographic algorithms without strong key protection provide no security benefit. To address this threat, healthcare cloud infrastructure stores all keys in hardware security modules or cloud-native key management services with access control and audit logging. Keys must be periodically rotated to enforce limits on how much data is encrypted under any one key. Deprecated keys must be made unrecoverable by securely destroying any copies. In the healthcare use case, data is almost always expected to be kept encrypted for decades, with keys changed periodically, and there are requirements that keys must be escrowed or there must be some form of key recovery mechanism to permit access to data after an investigation or litigation. Finally, the key needs business continuity support for key backup and key disaster recovery while still maintaining confidentiality of the key and preventing extraction of the key.
Data-in-transit encryption uses cryptography to protect healthcare data transmitted between API clients and cloud services. It ensures that network traffic is not subject to eavesdropping, tampering or man-in-the-middle attacks, which can lead to the unauthorized exposure and modification of data over the network. Transport Layer Security protocols are one of the main underlying technologies protecting API traffic. Cryptographic handshake protocols are used to verify the identities of the people communicating, agree on encryption methods and key exchange techniques, and create session keys for future messages. Certificate management is more complex, requiring obtaining a digital certificate from a trusted certificate authority, keeping the private key private, validating the authenticity of the certificate's revocation status, and renewing it before expiration, while maintaining uninterrupted and active protection. Choosing a cipher suite for transport layer security (TLS) means finding a good mix between strong security against attacks and the need for good performance, especially when dealing with older medical devices or other systems that can't be updated to use the latest security methods. Because healthcare cloud APIs are distributed, security and encryption of the API's multiple connections between client applications and API gateways, between API gateways and backend microservices, and potentially the API's participation with third-party services to provide laboratory results or insurance eligibility verification can be complicated, with each connection requiring its certificate and selection of a supported protocol while still having an overall cohesive security policy.

4.2 Secure Communication Protocols

Protocols related to healthcare cloud APIs include those for communication security, version control, and protocol enforcement. These protocols establish a minimum security posture for exchanging healthcare information across organizational boundaries or interfaces with cloud service providers. For example, HTTPS enforcement seeks to ensure that all communications with the API are conducted via the HTTP protocol protected with TLS, with no fallback to unprotected HTTP communication, guarding against network packet interception of sensitive healthcare information. HTTPS redirection and enforcement and HTTP Strict Transport Security are server- and client-side capabilities, respectively, for forcing browsers and applications to use encrypted HTTP. Security policies at an API gateway can provide protocol enforcement, enforce strong cipher suites, control certificate validation requirements, and enable defense

in depth, where gateway-level security complements application-level policies and provides unified security posture management in virtualized microservices environments. Protocol version deprecation requires that insecure protocol versions like SSL and early TLS versions with known vulnerabilities, like insecure cipher suite support and poor protocol design characteristics that make downgrade attacks possible, be disabled. A challenge for healthcare organizations is achieving compliance with these practices, especially given the presence of existing legacy medical devices, third-party products, and partner organizations that cannot be easily updated due to lengthy approval processes and limited resources from medical device manufacturers.
The end-to-end encrypted health cloud API architecture increases the cost of some cryptographic operations such as cipher suite negotiation during the exchange of the initial message in the SSL/TLS handshake protocol, symmetric encryption and decryption of message payloads, and Message Authentication Code (MAC) based integrity check of message payloads, whilst attempting to achieve security guarantees in those API communications while striving to satisfy system-level architectural requirements for latency, observability and other implementation complexity factors, to avoid impacting on the performance of time-sensitive clinical query operations or the ability to process transactions, for example for high-volume processing of insurance claims submissions and returning or distributing laboratory results. Performance considerations can also be a factor at the architectural layer. For instance, end-to-end encryption provides constant security along the entire request path but does preclude intermediate inspection for security monitoring and performance optimization purposes. An API gateway or load balancer can stop the encrypted data flow at a certain point, which lets it check the traffic and direct it based on content, but this means that unencrypted data is still visible within the secure network area. There can also be challenges in microservices architectures, where a single API request traverses multiple internal services with different security levels, trust boundaries, security domains, and threat models. Third-party integration adds another level of complexity. Healthcare APIs exchange protected health information with other systems, such as laboratory information systems, pharmacy benefits managers, pharmacy networks, health information exchanges, payer sites, and provider sites. These systems might have different security protocols, certificate authorities, and encryption algorithms. The process requires protocol translation, certificate chain validation across organizational boundaries, and mutual authentication.



SECURITY LAYER	TECHNOLOGY	KEY CHALLENGE
Data-at-Rest	AES-256, RSA hybrid encryption	Key management and long-term retention
Data-in-Transit	TLS protocol, certificate management	Legacy device compatibility
Protocol Security	HTTPS enforcement, version deprecation	API gateway policy enforcement
End-to-End	Complete path encryption vs termination	Performance vs security monitoring trade-off

Data-at-Rest	Encryption Standards
Data-in-Transit	TLS & Certificates
Protocols	Security Enforcement

Table 2: Encryption and Secure Transmission Protocol Overview [7, 8]

5. Security Frameworks, Best Practices, and Mitigation Strategies

5.1 Existing Security Frameworks and Standards

Industry security frameworks, such as the Open Web Application Security Project API Security Top 10, offer lists of controls that healthcare organizations can apply in systematically addressing API security threats, often based on industry experience and security research. The OWASP API Security Top 10 is a widely accepted list of the ten biggest security problems that can affect APIs, such as when an API doesn't properly check if a user has permission to access a specific object or resource; issues with user login security; revealing too much information in API responses; and not having enough resources or limits to prevent denial-of-service attacks. [9] The National Institute of Standards and Technology's cybersecurity framework is risk-based and subdivided into five functions: identify, protect, detect, respond, and recover. This framework provides a healthcare organization with a process for identifying information assets to be protected and control measures to be implemented based on its risk appetite. International Organization for Standardization standard 27001 establishes controls for managing information security. It focuses on governance, risk assessments, and selected controls. Healthcare standards build on this base with additional controls regarding these healthcare regulation requirements. The Health Information Trust Alliance Common Security Framework combines the specific rules from the HIPAA Security Rule, HITECH Act, and NIST publications into one set of controls, which are grouped by areas like access control and audit logging. Cloud Security Alliance white papers for healthcare provide guidance on cloud adoption by healthcare organizations, shared responsibility of cloud service providers and healthcare organizations, cross-border data flow policies on data residency, and technical security controls for securing multi-tenant cloud environments.

5.2 Best Practices for API Security

Secure API design and development involves incorporating security measures during the design and implementation stages of a system, before the software is deployed for use, which is more cost-effective than making changes after deployment. Secure coding guidelines for a specific programming language help prevent common mistakes, like stopping injection attacks by using parameterized queries, avoiding cross-site scripting through output encoding, and preventing buffer overflow errors by checking limits. Input validation and verification check all the information that the API gets from clients (like format, data type, and acceptable values) to reject any requests that have mistakes, harmful elements, or invalid values before they reach the main business logic. Rate limiting and throttling protect an API from resource exhaustion by limiting the number of requests a client can make within a given time. Continuous security monitoring collects and analyzes security events in real time to detect incidents, policy violations and anomalous behaviors that indicate an attack. Security information and event management (SIEM) solutions gather and connect logs from different parts of an API system to find attacks that happen through a series of normal events. Anomaly detection uses behavioral analytics and machine learning to establish what normal usage patterns of an API look like and to identify unusual usage, which could suggest stolen credentials or insider threats. Regulatory compliance standards may require organizations to demonstrate compliance with security and privacy controls by carrying out regular assessments and continuously monitoring them. Security assessment involves checking for weaknesses in systems using automated scans and manual checks (vulnerability assessment), estimating how likely threats are and their possible effects (risk assessment), and testing how well security measures work by simulating an attack (penetration testing). To reduce operational burdens, compliance automation tools provide evidence of security control operations and allow organizations to produce compliance reports documenting both gaps and adherence.

Table 3: Security Frameworks and Best Practices [9, 10]

Framework/Practice	Purpose
OWASP API Security Top 10	Critical API vulnerabilities
NIST Cybersecurity Framework	Risk-based security approach
HITRUST CSF	Healthcare compliance harmonization
Secure Coding & Input Validation	Prevent injection attacks
SIEM & Anomaly Detection	Real-time threat monitoring
Penetration Testing	Vulnerability identification

Conclusion

The systematic review of the literature indicates that there are several security vulnerabilities pertaining to healthcare cloud APIs, which affect all types of authentication, encryption, and data transport mechanisms, and the confidentiality, availability, and integrity of PHI. Impersonators could execute attacks using weak multi-factor authentication, token-based authentication, or malformed session identifiers. It is found that the authorization configuration in the healthcare cloud APIs does not support the complexity of clinical workflows and the emergency access process, which are intrinsic to the use of RBAC and ABAC. Data at rest requires implementing decryption processing and key management, data in transit requires setting transport layer security, and cryptographic overhead must be considered. Time-sensitive clinical activity requirements may conflict with the system's performance requirements. Man-in-the-middle attacks, distributed denial of service, and attempts to gain access are encouraged by the intersection of valuable, internet-accessible APIs and healthcare data. HIPAA, GDPR, and related privacy regulations further compound vulnerabilities by increasing network complexity. More studies are needed on security technologies like AI-based threat detection (which helps find hidden attacks), zero-trust technology (which verifies users no matter where they are), and quantum-resistant cryptography (to protect against future quantum computers that could break current encryption). Healthcare organizations should also work on developing a security program that includes secure API design practices, continuous monitoring, security testing, and automation to comply with regulations, including in hybrid and multi-cloud environments, which add security complexity. Fulfilling requirements for data protection in health care can be achieved by improving the security features of cloud service providers focusing on health care, better transparency about shared responsibility models, and improving encryption, access control, and audit logging solutions. Regulatory frameworks can be developed to support cloud-native models and technologies. Such measures can ease healthcare innovation while not compromising on the security and privacy of patients' health data, thus improving public trust in digital health systems.

References

[1] Jigna J. Hathaliya and Sudeep Tanwar, "An exhaustive survey on security and privacy issues in Healthcare 4.0," Computer Communications, vol. 153, pp. 311-335, 2020. Available: https://doi.org/10.1016/j.comcom.2020.02.018

[2] Clemens Scott Kruse et al., "Security Techniques for the Electronic Health Records," Journal of Medical Systems, vol. 41, no. 8, article 127, August 2017. Available: https://doi.org/10.1007/s10916-017-0778-4

[3] Hassan Takabi et al., "Security and Privacy Challenges in Cloud Computing Environments," IEEE Security & Privacy (Volume: 8, Issue: 6), 2010. Available: https://ieeexplore.ieee.org/document/5655240

[4] Duane Bender and Kamran Sartipi, "HL7 FHIR: An Agile and RESTful approach to healthcare information exchange", Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems, 2013. Available: https://doi.org/10.1109/CBMS.2013.6627810

[5] José Luis Fernández-Alemán et al., "Security and privacy in electronic health records: A systematic literature review," Journal of Biomedical Informatics, Volume 46, Issue 3, 2013. Available: https://doi.org/10.1016/j.jbi.2012.12.003

[6] Assad Abbas et al., "A cloud-based health insurance plan recommendation system: A user-centered approach," Future Generation Computer Systems, vol. 43-44, 2015. Available: https://doi.org/10.1016/j.future.2014.08.010
[7] Adil Hussain Seh et al., "Healthcare Data Breaches: Insights and Implications," Healthcare, vol. 8, no. 2, article 133, 2020. Available: https://doi.org/10.3390/healthcare8020133
[8] Mazhar Ali et al., "SeDaSC: Secure Data Sharing in Clouds," IEEE Systems Journal (Volume: 11, Issue: 2), 2015. Available: https://ieeexplore.ieee.org/document/7008450
[9] Rahul Johari and Pankaj Sharma, "A Survey on Web Application Vulnerabilities (SQLIA, XSS) Exploitation and Security Engine for SQL Injection," International Conference on Communication Systems and Network Technologies, 2012. Available: https://ieeexplore.ieee.org/document/6200667
[10] Abdul Razaque et al., "State-of-the-Art Review of Information Diffusion Models and Their Impact on Social Network Vulnerabilities," Journal of King Saud University - Computer and Information Sciences, Volume 34, Issue 1, 2022. Available: https://doi.org/10.1016/j.jksuci.2019.08.008