Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 02 | Feb-2026
ISSN (O) 3107-6696

Secure Tamper-Resistant chat Application using Digital Signature Algorithm (DSA)

[1]G Sivasangari, [2]Dr. V. Malathi

Department of Computer Science with Cybersecurity, Dr. N.G.P Arts and Science College, Coimbatore, India, sivasangari24.06@gmail.com

Department of Computer Science with Cybersecurity, Dr. N.G.P Arts and Science College, Coimbatore, India, malathi.v@drngpasc.ac.in

---

Abstract - Chat applications are widely used in personal, academic, and professional communication, but many existing systems lack proper security mechanisms to protect message from tampering, fake sender identities, and replay attacks. Messages can be altered during transmission, leading to serious security risks and misinformation. To address these issues, this paper presents a Secure Tamper-Resistant Chat Application developed using Java and Digital Signature Algorithms. In the proposed system, each message is digitally signed by the sender and verified by the receiver to ensure both message integrity and sender authenticity. Any modified or forged message is automatically rejected, ensuring secure and reliable communication through a client-server architecture.

Keywords: Secure Chat Application, Digital Signature Algorithms, Message Integrity, Sender Authentication, Tamper Resistance, Java, Client–Server Architecture

1. Introduction

Chat applications have become one of the most common ways of communication in personal, academic, and professional environments. They allow users to exchange messages quickly and easily over the internet. However, many existing chat systems mainly focus on speed and usability, while security is often given less importance. Because of this, such applications are vulnerable to various threats, including message tampering, fake sender identities, and replay attacks, which can lead to misinformation and loss of trust among users.

To address these security challenges, this paper proposes a Secure Tamper-Resistant Chat Application developed using Java and Digital Signature Algorithms. In the proposed system, every message is digitally signed by the sender and verified at the receiver side to ensure that the message is original and sent by an authenticated user. If any message is altered or forged during transmission, it is automatically detected and rejected. By using a client–server architecture and digital signature verification, the system provides secure, reliable, and trustworthy real-time communication.

2. Security Challenges in Chat Applications

Chat applications are widely used because they provide fast and convenient communication, but they also face several security challenges. Most traditional chat systems are designed with a focus on performance and user experience, while security aspects are often treated as secondary requirements. This makes chat applications an attractive target for attackers who attempt to exploit vulnerabilities in message transmission and user authentication.

One of the major security challenges is **message tampering**, where an attacker modifies the content of a message while it is being transmitted over the network. Since chat messages usually pass through multiple network layers, there is a possibility that unauthorized entities may intercept and alter the data. Such modifications can lead to misinformation, misunderstandings, and serious consequences in sensitive communication environments.

Another important challenge is **fake sender identity or impersonation attacks**. In this type of attack, an adversary pretends to be a legitimate user and sends messages on their behalf. Without proper authentication mechanisms, the receiver cannot easily verify whether the message was actually sent by the claimed sender. This reduces trust in the communication system and can be exploited for fraud or social engineering attacks.

Replay attacks also pose a significant threat to chat applications. In a replay attack, old messages are captured and resent to the receiver, causing confusion or misleading communication. Since many chat systems do not maintain proper verification of message freshness, such attacks are difficult to detect. These security challenges highlight the need

for strong mechanisms that can ensure message integrity, sender authenticity, and protection against tampering, which motivates the use of digital signature–based solutions.

3. Digital Signature Algorithms: An Overview

Digital Signature Algorithms (DSA) are cryptographic techniques used to verify the authenticity and integrity of digital messages. A digital signature ensures that a message has been sent by a legitimate sender and has not been altered during transmission. It works using asymmetric cryptography, where each user is assigned a pair of keys: a private key for signing messages and a public key for verifying signatures.

The digital signature process generally involves three main steps: key generation, message signing, and signature verification. During key generation, a public–private key pair is created for the user. When a message is sent, the sender uses the private key to generate a digital signature based on the message content. At the receiver side, the public key of the sender is used to verify the signature and confirm the authenticity of the message.

Digital signatures provide important security properties such as message integrity, authentication, and non-repudiation. If a message is modified even slightly, the verification process fails, indicating tampering. Because of these strong security guarantees, digital signature algorithms are widely used in secure communication systems and form a reliable foundation for building tamper-resistant chat applications.

4. Related Work and Existing Secure Chat Systems

Several secure chat systems have been proposed and developed to address the security issues present in traditional messaging applications. Most existing systems focus on encryption techniques, such as symmetric and asymmetric encryption, to protect message confidentiality. End-to-end encryption is widely used to ensure that only the sender and receiver can read the message content, preventing unauthorized access during transmission.

While encryption provides confidentiality, many existing chat systems do not fully address message integrity and sender authentication at the application level. Some systems rely on basic authentication methods such as usernames and passwords, which are vulnerable to impersonation attacks. In addition, not all systems provide effective mechanisms to detect message tampering or replay attacks, making them less reliable for secure communication.

Recent research has explored the use of digital signatures in secure messaging systems to improve authenticity and integrity. Digital signature–based approaches allow receivers to verify the sender and ensure that messages have not been modified. However, many of these approaches are complex or lack clear implementation in real-time chat environments. This survey highlights the need for a simple and efficient digital signature–based chat system that can provide strong tamper resistance while maintaining ease of use and real-time communication.

5. Proposed Secure Tamper-Resistant Chat Architecture

The proposed secure chat system is designed using a client–server architecture to support real-time communication between users. In this architecture, each user acts as a client that can send and receive messages through a centralized server. The server is responsible for managing user connections and forwarding messages, while security-related verification is mainly performed at the client side to ensure message authenticity.

Each user in the system is assigned a unique public–private key pair. The private key is securely stored on the sender's device and is used to digitally sign outgoing messages. The corresponding public key is shared with other users or stored in a trusted directory, allowing receivers to verify the digital signature of incoming messages. This key-based design ensures that only authenticated users can send valid messages.

The architecture ensures tamper resistance by verifying every message before it is displayed to the receiver. If a message is modified during transmission or sent by an unauthorized user, the digital signature verification fails and the message is rejected. By combining digital signature verification with a structured client–server model, the proposed architecture provides secure, reliable, and trustworthy chat communication.

6. Message Signing and Verification Methodology

In the proposed system, message security is achieved through a digital signing and verification process. When a user composes a message, the system first prepares the message content for transmission. Before sending the message, a digital signature is generated using the sender's private key. This signature is uniquely linked to the message content, ensuring that any modification to the message will invalidate the signature.

After the message is signed, both the message and its digital signature are sent to the server. The server acts as an intermediate entity that forwards the signed message to the intended receiver without altering its content. The server does not need access to the private keys, which helps maintain the confidentiality and security of the signing process.

At the receiver side, the digital signature is verified using the sender's public key. The verification process checks whether the received message matches the signature. If the verification is successful, the message is considered authentic and is displayed to the user. If the verification fails, the message is identified as tampered or forged and is rejected. This methodology ensures secure and reliable message exchange in the chat application.

Figure 1 Architecture of Secure Tamper-Resistant Chat Application

7. Security Analysis and Discussion

The proposed secure chat system provides strong protection against message tampering by using digital signature verification. Since each message is signed using the sender's private key, any modification to the message content during transmission will result in a signature mismatch. This ensures that tampered messages are immediately detected and rejected, maintaining message integrity throughout the communication process.

The system also effectively addresses impersonation attacks by verifying the sender's identity through public key authentication. Only users who possess valid private keys can generate correct digital signatures, making it difficult for attackers to forge messages. This mechanism increases trust between communicating users and ensures that messages are exchanged only between authenticated participants.

In addition, the system can prevent replay attacks by associating each message with unique identifiers or timestamps. This allows the receiver to detect and discard old or duplicated messages. While digital signature verification introduces slight computational overhead, the added security benefits significantly outweigh the performance cost, making the proposed approach suitable for secure real-time chat applications.

Figure 2 shows the digital signature based message signing and verification workflow.

8. Conclusion and Future Work

This paper presented a survey and design overview of a Secure Tamper-Resistant Chat Application using Digital Signature Algorithms in Java. The study highlighted the major security challenges present in existing chat systems and demonstrated how digital signatures can effectively ensure message integrity, authenticity, and resistance to tampering. By integrating digital signature verification into a client–server chat architecture, the proposed approach provides a secure and reliable communication platform.

The use of digital signatures allows the receiver to verify both the origin and originality of each message before accepting it. This significantly reduces the risk of message tampering, impersonation, and replay attacks. Although the verification process introduces a small computational overhead, the improvement in security and trust makes the approach suitable for secure real-time communication environments.

Future work can focus on enhancing the system by integrating advanced security features such as blockchain-based message storage, multi-factor authentication, and secure file sharing. In addition, the system can be extended to support mobile platforms and large-scale deployments, further improving its practicality and real-world applicability.

9. References

[1] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[2] 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.

[3] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering: Design Principles and Practical Applications*. Hoboken, NJ, USA: Wiley, 2010.

[4] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1996.

[5] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Pearson Education, 2017.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[7] Oracle, "Java Cryptography Architecture (JCA) Reference Guide," Oracle Corporation. [Online]. Available: https://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html

[8] IEEE Computer Society, "IEEE Standard for Secure Message Authentication," *IEEE Std 1363*, 2018.