Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 4, No. 1, 2025

Overview of Secure Time Synchronization in NTP

Huaqiang Wang

Department of Electronic Information, Southwest Minzu University, Chengdu 610225, China

* **Corresponding author**: Liangfu Peng (Email: pengliangfu@126.com)

Abstract: This paper provides an overview of the fundamental principles, historical evolution, and current security challenges of the Network Time Protocol (NTP), along with a detailed exploration of various enhancement methods proposed to address existing vulnerabilities. Initially, the paper analyzes the working mechanisms of NTP and its version progression, highlighting the protocol's critical role in ensuring consistent time synchronization and event ordering within distributed systems. Subsequently, it delves deeply into typical security threats faced by NTP, including packet spoofing, replay attacks, Denial of Service (DoS) attacks, as well as vulnerabilities inherent in the protocol's implementation. Moreover, the paper summarizes multiple security enhancement approaches proposed by researchers globally in recent years, including the Autokey protocol featuring authentication and key exchange mechanisms, authentication models leveraging Elliptic Curve Digital Signature Algorithm (ECDSA) and Secure Sockets Layer (SSL) certificates, the standardized Network Time Security (NTS) protocol, and security enhancement techniques integrating Chinese cryptographic standards and digest authentication mechanisms. Additionally, the paper highlights the shortcomings of existing approaches in practical application scenarios, particularly emphasizing the insufficient security research on the broadcast mode of NTP. It further points out the necessity of systematic design and verification of security strategies related to broadcast mode, aiming to meet synchronization security requirements across broader and more complex network environments.

Keywords: NTP, NTS, Security, Broadcast mode, Authentication.

1. Introduction

Time synchronization plays a critical role in distributed systems, ensuring coordinated system operation, maintaining event order consistency, and enabling the implementation of security policies. Due to frequency drift in internal hardware clocks—affected by temperature variations and oscillator inaccuracies—different devices cannot maintain consistent system time. This can result in timing errors of several tens of microseconds per second. Therefore, synchronization mechanisms are essential to maintain time consistency across network nodes.

Time synchronization approaches are generally categorized into external synchronization—such as using the Universal Time Coordinated (UTC) standard time—and internal synchronization, which involves mutual calibration among nodes. Existing methods include hardware-based synchronization, satellite navigation-based synchronization, and software-based synchronization. Although the first two offer high precision, they are costly, complex to deploy, and unsuitable for large-scale distributed systems. In contrast, software-based synchronization offers advantages such as low cost and strong scalability. Among these, the Network Time Protocol (NTP), which enables large-scale time coordination over networks, has become one of the core technologies for achieving high-precision time synchronization.

In 1985, Professor Mills from the University of Delaware first proposed the concept of the Network Time Protocol (NTP) to achieve high-precision time synchronization [1]. The protocol calibrates time between hosts by measuring the Round-Trip Delay and Clock Offset. NTP has undergone several versions of evolution: NTPv1 [2], as the initial version, primarily demonstrated the feasibility of time synchronization over unreliable networks; NTPv2 [3] introduced symmetric and broadcast modes, supported multiple time distribution structures, and added authentication and control message mechanisms for improved management and monitoring. NTPv3 [3] optimized the clock filtering algorithm and error evaluation model, enhancing synchronization accuracy and robustness, while improving compatibility with various network protocols. The latest version, NTPv4 [5], released in June 2010, further refines the synchronization algorithm, enhances cross-platform adaptability and protocol compatibility, and supports more complex network environments and multi-tier time server architectures.

Since data packets in network time services are typically transmitted in plaintext over the Internet, they are vulnerable to various security threats in the absence of effective protection mechanisms. For example, attackers may eavesdrop on, tamper with, or replay NTP packets sent by clients, crafting forged packets to launch spoofing attacks against clients or servers, or even initiate Denial of Service (DoS) attacks that lead to network congestion. Such attacks can result in users receiving incorrect or invalid time information, causing not only economic losses but also posing serious risks to the operation of critical systems [6]. Therefore, stringent security requirements are essential for network time service systems.

First, clock synchronization between clients and servers is closely tied to identity authentication, requiring efficient encryption and authentication processes to be completed within very short time windows. Second, since the NTP packet format is publicly known, attackers can easily intercept and forge packets, using replay attacks to disrupt the time synchronization process and further increase system risk. In addition, the NTP service architecture is hierarchical: time information is transmitted from primary time sources through multiple layers of servers before reaching the client. Only by ensuring the trustworthiness of all servers along the transmission path can the authenticity and reliability of the

time data received by the client be guaranteed.

Given the security challenges currently facing network time services, it is necessary to design efficient and practical security protection models based on the operational mechanisms of the NTP protocol. Enhancing the system's resilience to attacks while maintaining service performance is of great significance for improving the reliability of network time synchronization services and strengthening the time security of critical applications. Therefore, a thorough analysis of the evolution of NTP security mechanisms and the characteristics of existing solutions can provide valuable theoretical foundations and technical references for future research.

2. Research Status

In the field of NTP security risk analysis, Martin [7] revealed a vulnerability in the NTP daemon (ntpd) that allows a man-in-the-middle attacker to forge malicious packets, forcing the client to shift its clock beyond an acceptable threshold. This can result in abnormal system time, affecting critical applications and services that rely on accurate timestamps. The U.S. National Vulnerability Database (NVD) reported in 2014 that versions of ntpd prior to 4.2.8 contained multiple stack-based buffer overflow vulnerabilities, which attackers could exploit to execute arbitrary code or crash the service—seriously undermining the security and reliability of the NTP protocol [8].

Malhotra et al [9], discovered that even when symmetric key-based authentication is enabled, NTP's broadcast mode still suffers from critical security flaws. Attackers can craft syntactically valid but unauthenticated broadcast packets (e.g., with forged MACs) and repeatedly send them to clients. This causes clients to discard packets continuously, eventually lacking enough valid samples to compute the time offset—resulting in a denial-of-service (DoS) attack on the synchronization process. They also proposed a replay attack that can lock a client's clock to a fixed time, highlighting the insufficiency of symmetric-key-based security in broadcast scenarios.

Rascagneres [10] disclosed a DoS vulnerability in NTP's timestamp verification mechanism. Attackers can send specially crafted NTP requests to prevent the server from responding to legitimate clients, thus exploiting logical flaws in timestamp validation and causing service denial.

Diao Zaoxiang et al [11], systematically analyzed the security vulnerabilities of the NTP protocol in both LAN and IP network environments using symmetric and asymmetric cryptographic mechanisms. They pointed out weaknesses in the synchronization algorithms, authentication procedures, and client behavior monitoring. Attackers could exploit these flaws through ARP spoofing, server impersonation, distributed denial-of-service (DDoS) attacks, and DNS hijacking to induce time drift or service disruption. Experimental results demonstrated that symmetric authentication could allow up to 4 seconds of time deviation, while asymmetric mechanisms still showed high attack success rates.

Furthermore, Malhotra et al [12], argued that the NTP datagram specification (RFC 5905) lacks comprehensive consideration of security requirements across different operational modes. They identified several structural vulnerabilities exploitable by remote attackers, such as control interface information leakage and weak authentication, enabling offline attacks like time tampering.

To address security concerns in NTP, Professor Mills [13] proposed the Autokey protocol, which transmits authentication data via NTP extension fields. During the key exchange phase, Autokey supports multiple identity authentication schemes to verify the server's legitimacy, and employs Message Authentication Codes (MAC) during the synchronization phase to ensure data integrity and authenticity. Autokey is compatible with the standard NTP protocol, easy to deploy, and imposes minimal impact on synchronization precision, thus providing a relatively practical security enhancement. However, it suffers from significant security weaknesses, including susceptibility to man-in-the-middle attacks and fragile authentication mechanisms. These issues mainly stem from the use of cryptographically weak algorithms, particularly the broken Message-Digest Algorithm 5 (MD5), which undermines its overall security capability [14].

In response to the widespread lack of authentication mechanisms or the reliance on weak cryptographic algorithms in existing NTP implementations, Dowling et al. [15] proposed the Authenticated Network Time Protocol (ANTP). This protocol combines asymmetric key exchange with symmetric encryption, effectively defending against time desynchronization attacks without sacrificing synchronization accuracy or introducing server-side state burdens. ANTP was implemented and validated in OpenNTPD, demonstrating its security and scalability. However, it requires additional deployment overhead.

In 2017, Malhotra et al [16], identified flaws in NTP's datagram protocol, highlighting that its control query interface may leak sensitive information. Such information can be exploited by attackers to launch off-path attacks against NTP. Based on these vulnerabilities, the authors designed and implemented an encryption model to enhance the security of NTP communications.

Köğçe et al [17], proposed integrating the Secure Sockets Layer (SSL) certificate infrastructure into the NTP protocol to provide both source authentication and message integrity. Although the Rivest–Shamir–Adleman (RSA) algorithm offers strong security, its computational overhead may impair synchronization accuracy. In contrast, their use of the Elliptic Curve Digital Signature Algorithm (ECDSA) achieves equivalent security strength with shorter key lengths, significantly reducing both computational and communication costs—thus improving system performance and deployment efficiency.

Li Xin et al [18], conducted a detailed analysis of typical attack methods targeting NTP and proposed a series of practical countermeasures to address the security threats in network time synchronization. Their work emphasized the importance of regularly updating NTP software versions, disabling unnecessary ports, establishing robust authentication mechanisms, and building a trusted time source hierarchy. To mitigate DoS attacks and signal spoofing, they suggested implementing hardware redundancy, packet filtering, and multi-source time verification mechanisms to enhance system robustness. Furthermore, they advocated the integration of encrypted transport technologies such as Internet Protocol Security (IPSec) and Media Access Control Security (MACsec) to protect time synchronization data during transmission. They also proposed the construction of a time security monitoring platform for real-time detection and response to abnormal behaviors, providing theoretical support and systematic guidance for enhancing NTP's security in

practical applications.

Lan Li [19], combined formal modeling with system reliability analysis to systematically examine the security vulnerabilities of railway time synchronization networks using the Autokey mechanism. Employing Colored Petri Nets, Stochastic Petri Nets, and Semi-Markov processes, they modeled potential attack paths and state transition probabilities. The study identified the risk of man-in-the-middle attacks and further proposed a multi-layer system reliability evaluation framework based on weighted radar charts and cloud models to locate weak points in time synchronization systems.

As concerns over NTP security grew, the IETF introduced Network Time Security (NTS) as a security enhancement to the protocol [20]. NTS aims to provide cryptographic authentication between NTP clients and servers, ensuring the authenticity and integrity of time synchronization packets while defending against replay attacks. The latest official NTS standard, RFC 8915, was released in September 2020 and systematically defines security strategies for the client/server mode of NTP. Earlier versions, such as NTS-06, used custom handshake protocols and embedded specific message types in NTP extension fields to implement security features. While effective against amplification attacks, these designs suffered from issues such as asymmetric packet sizes, induced time offsets, IP fragmentation, and privacy concerns. Moreover, the custom handshake protocol revealed multiple security vulnerabilities. To address these issues, NTS version 17 and beyond adopted the standard Transport Layer Security (TLS) handshake, replacing the earlier custom design. This transition significantly improved communication security, resolved fragmentation and replay issues, and reduced processing latency.

Langer et al [21], conducted comparative experiments to evaluate the performance differences between NTP and NTS. They noted that while NTS improves protocol security, it introduces additional computational and communication overhead.

Chen Xi et al [22], proposed incorporating MD5 and Secure Hash Algorithm 1 (SHA-1) into NTP implementations to verify data integrity and enhance tamper resistance. On this basis, they further suggested hashing critical NTP data fields to improve attack resistance and synchronization security. Experimental results demonstrated that this security-enhanced mechanism maintains millisecond-level synchronization precision while improving the overall security of the synchronization process.

3. Literature Review

With the widespread adoption of network time synchronization in critical infrastructure, the security of the Network Time Protocol (NTP) has increasingly attracted attention from both academia and industry. Existing research has focused on the various security threats faced by NTP and the corresponding protection mechanisms, leading to a relatively systematic body of theoretical and practical work. As the dominant time synchronization protocol in use today, NTP has long lacked strong identity authentication and integrity verification mechanisms in open networks. This has made it vulnerable to a range of attacks, including message forgery, replay attacks, delay attacks, and denial-of-service (DoS) attacks. To address these risks, researchers have proposed a variety of security enhancements, primarily focusing on message authentication, anti-replay mechanisms, and secure key exchange protocols.

Representative approaches include: integrating SSL certificate systems with the Elliptic Curve Digital Signature Algorithm (ECDSA) to provide identity authentication and message integrity while minimizing synchronization latency [17]; employing formal modeling tools such as Colored Petri Nets and semi-Markov processes to analyze potential attack paths and assess system reliability, thereby identifying structural vulnerabilities [19]; improving synchronization accuracy and tamper-resistance by establishing trusted chains of time sources and multi-source time verification mechanisms [18]; adopting the Network Time Security (NTS) protocol to introduce standardized TLS handshake mechanisms, achieving message authentication and replay protection [20]; and applying hash-based integrity checks to critical NTP fields to enhance anti-tampering capabilities without compromising synchronization precision [22].

4. Conclusion

While these efforts have yielded significant results in the client-server (unicast) mode, they fall short of addressing all real-world synchronization scenarios. In particular, there is a lack of systematic design and validation for securing NTP's broadcast mode, which is widely used in large-scale distributed environments due to its efficient and connectionless nature. However, the absence of interaction mechanisms makes broadcast mode more susceptible to replay and spoofing attacks. Therefore, future research on NTP security should pay greater attention to the unique characteristics and challenges of the broadcast mode, in order to meet the security demands of more complex and large-scale network environments.

References

[1] D. Mills. RFC 958, Network Time Protocol (NTP) [S]. September, 1985.

[2] D. Mills. RFC 1059, Network Time Protocol (Version 1) Specification and Implementation [S]. July, 1988.

[3] D. Mills. RFC 1119, Network Time Protocol (Version 2) Specification and Implementation [S]. September, 1989.

[4] D. Mills. RFC 1305, Network Time Protocol (Version 3) Specification, Implementation and Analysis [S]. March, 1992.

[5] D. Mills, J. Martin, Ed, et al. RFC 5905, Network Time Protocol Version 4: Protocol and Algorithms Specification [S]. June, 2010.

[6] Dong P, Wei G. Research of security network timing service technique [J]. Journal of Time & Frequency, 2018.

[7] Martin Prpic. MITM attacker can force ntpd to make a step larger than the panic threshold [EB/OL]. https://bugzilla. redhat. com/show_bug. cgi? id=CVE-2015-5300. 2015-10-03

[8] National Vulnerability Database. CVE-2014-9295: Multiple stack-based buffer overflows in ntpd in ntp before 4.2.8 [OL], 2014.

[9] Malhotra A, Goldberg S. Attacking ntp's authenticated broadcast mode [J]. ACM SIGCOMM Computer Communication Review, 2016, 46(2): 12-17.

[10] Paul Rascagneres. Vulnerability Spotlight: Exploiting Network Time Protocol Origin Timestamp Check Denial of Service Vulnerability [EB/OL]. https://blog.talosintelligence.com/ntpd-dos.

[11] Zao-Xiang D, Xiao-Ning Z, Shu-Jun W, et al. The Vulnerability of NTP Under Forged Server Attack [J]. Electronic Information Warfare Technology, 2016.

[12] Malhotra A, Gundy M V, Varia M, et al. The Security of NTP's Datagram Protocol [C]// Springer, Cham. Springer, Cham, 2017.

[13] Mills D L. The Autokey security architecture, protocol and algorithms [J]. Network Working Group, University of Delaware, Technical Report, 2006: 06-1

[14] Xie T, Feng D. How to find weak input differences for MD5 collision attacks [J]. 2009

[15] Dowling B, Stebila D, Zaverucha G. Authenticated network time synchronization [C]// 25th Security Symposium. 2016: 823-840.

[16] Malhotra A, Van Gundy M, Varia M, et a1. The security of ntp's datagram protocol [C]. International Conference on Financial Cryptography and Data Security. Springer, Cham, 2017: 405- 423.

[17] Köğçe M, Şişeci N E. A new approach to security of NTP via SSL certificates [C]//2019 1stInternational Informatics and Software Engineering Conference (UBMYK). IEEE, 2019: 1-5.

[18] Xin L, Yingxin G. Security Analysis and Suggestions on Network Time Synchronization Based on NTP [J]. Modern Transmission, 2019(3):3.

[19] Li L, Youpeng Z. Vulnerability Analysis of Railway Time Synchronization Network Protocol Based on Stochastic Petri Net [J]. Journal of the China Railway Society, 2017.

[20] Franke D, Sibold D, Teichel K, et al. RFC 8915: Network time security for the network time protocol [J]. 2020.

[21] Langer M, Bermbach R, Teichel K, et al. Performance comparison between network time security protocol drafts: Improvements and accuracy of the latest NTS draft [C]//2019 Joint Conference of the IEEE International Frequency Control Symposium and European Frequency and Time Forum (EFTF/IFC). IEEE, 2019: 1-7.

[22] Xi C, Wenchi Z, Ming M. Research on secure NTP method based on message digest encryption [J]. GNSS World of China, 2021, 46(05): 84-91.