

Securing Precision Time Synchronization with NTS for PTP and CSPTP

Karen O'Donoghue

5 May 2026

WSTS 2026

Network Time Security options – before 2020

Precision Time Protocol (PTP)

- IEEE Std 1588TM-2008
 - Annex K – Group source authentication, message integrity, and replay attack protection (defined as Experimental, flaws identified)

Network Time Protocol (NTP)

- RFC 5905 – NTPv4
 - Documents the symmetric key authentication scheme from NTPv3
 - Optional Message Authentication Code (MAC)
 - Pre-shared key scheme for server authentication (scaling issues)
- RFC 5906 - Autokey
 - Provides authentication of time servers using PKI (known flaws)
 - Concerns about design and limited deployment

Network Time Security (NTS)

- RFC 8915
- Published in September 2020
 - <https://www.rfc-editor.org/rfc/rfc8915.pdf>
- Multiple interoperable implementations
 - Open Source
 - Commercial product support
- Public NTS enabled NTP servers
- Exploring the use of NTS pools
 - <https://datatracker.ietf.org/doc/draft-ietf-ntp-nts-keyexchange-pool/>

Stream: Internet Engineering Task Force (IETF)
RFC: [8915](#)
Category: Standards Track
Published: September 2020
ISSN: 2070-1721
Authors: D. Franke D. Sibold K. Teichel M. Dansarie R. Sundblad
Akamai PTB PTB Netnod

RFC 8915 Network Time Security for the Network Time Protocol

Abstract

This memo specifies Network Time Security (NTS), a mechanism for using Transport Layer Security (TLS) and Authenticated Encryption with Associated Data (AEAD) to provide cryptographic security for the client-server mode of the Network Time Protocol (NTP).

NTS is structured as a suite of two loosely coupled sub-protocols. The first (NTS Key Establishment (NTS-KE)) handles initial authentication and key establishment over TLS. The second (NTS Extension Fields for NTPv4) handles encryption and authentication during NTP time synchronization via extension fields in the NTP packets, and holds all required state only on the client via opaque cookies.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

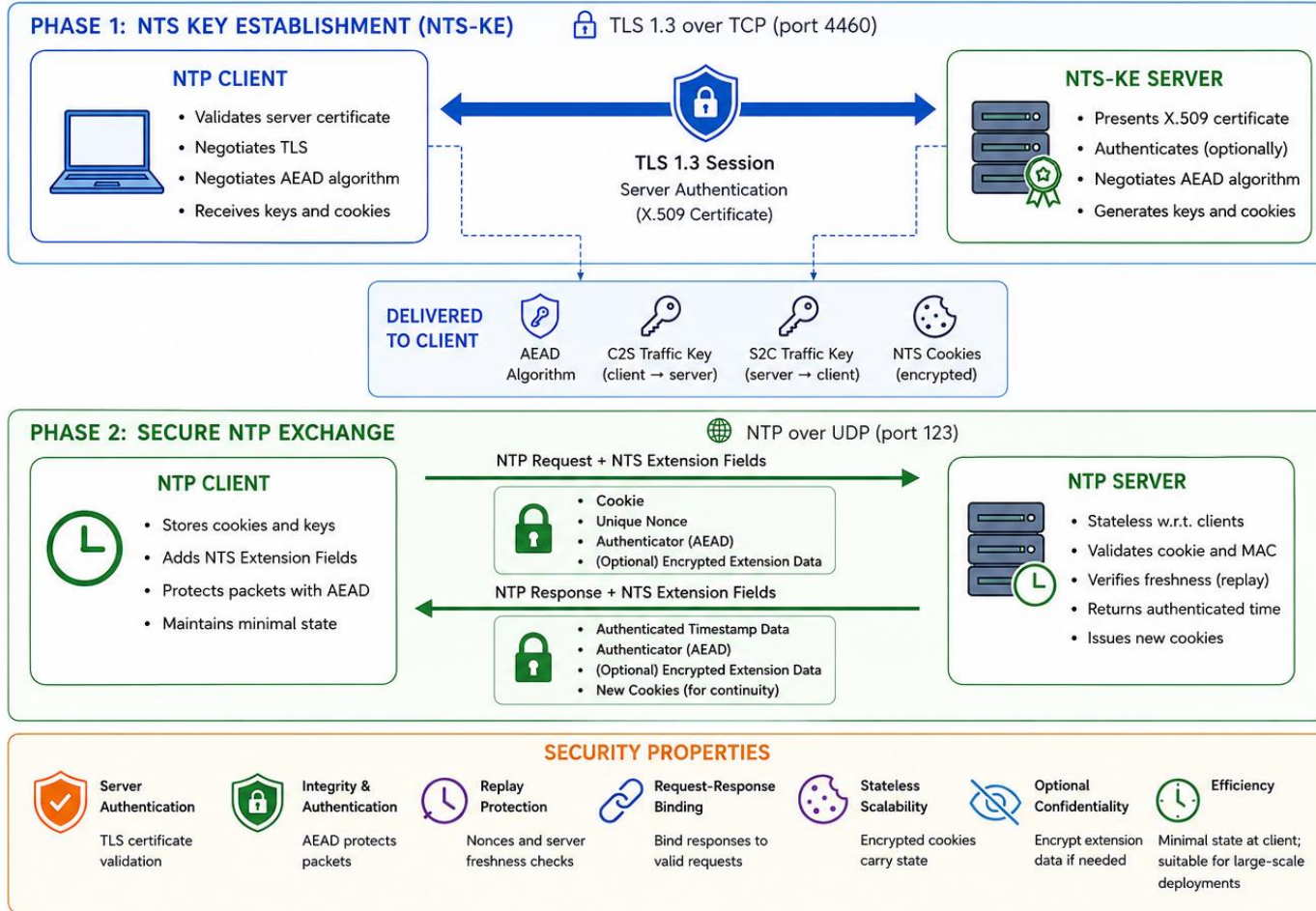
Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8915>.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

RFC 8915 – NETWORK TIME SECURITY (NTS) ARCHITECTURE

NTS secures NTP by combining a TLS-based key establishment with AEAD-protected NTP exchanges.



Current PTP Security

- IEEE Std 1588™ – 2019
- As amended by IEEE Std 1588d™-2023 (GDOI Key Management)
 - Note: There are other non-security related amendments as well.



IEEE 1588 Security Approach

IEEE 1588 security includes a set of mechanisms and tools that can be used together or individually.



Individual mechanisms are optional.

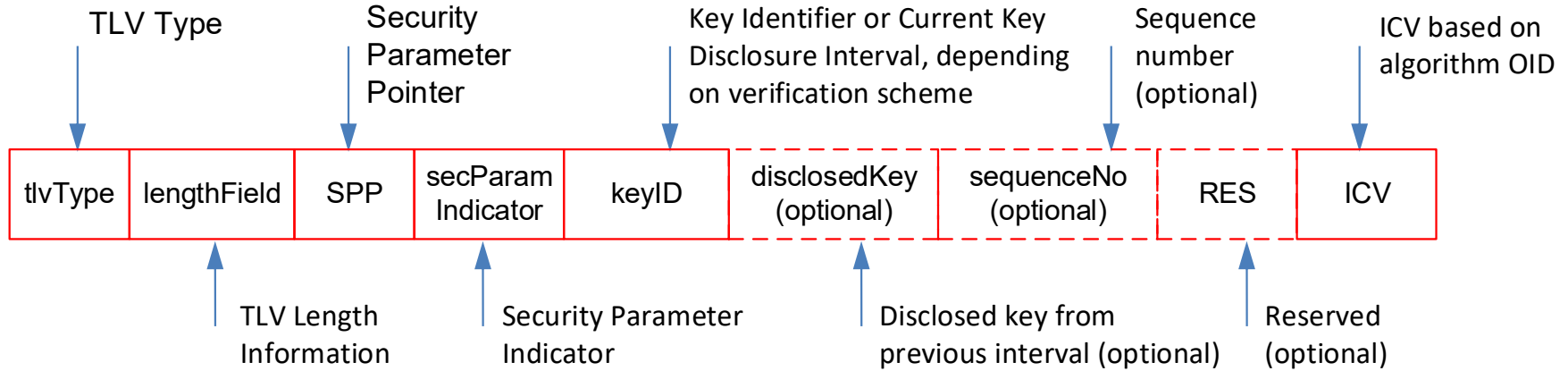


The specific mechanisms chosen may vary by application and environment.

IEEE 1588 Security



- The multi-pronged approach:
 - PTP Integrated Security Mechanisms (Prong A)
 - External Transport Security Mechanisms (Prong B)
 - Architecture Guidance (Prong C)
 - Monitoring and Management Guidance (Prong D)



AUTHENTICATION TLV

- From: IEEE Std 1588™ – 2019

Possible Key Management options for PTP

- Static key distribution by some form of out of band communication
 - Good for getting started but not a long-term or large-scale solution
- GDOI
- TESLA – more detailed specification needed (may be eventually deprecated if no active interest is identified)
- Network Time Security (NTS) – under development

PTP Security - Current work items

1

Address recently identified PTP Vulnerabilities, fix identified errors

2

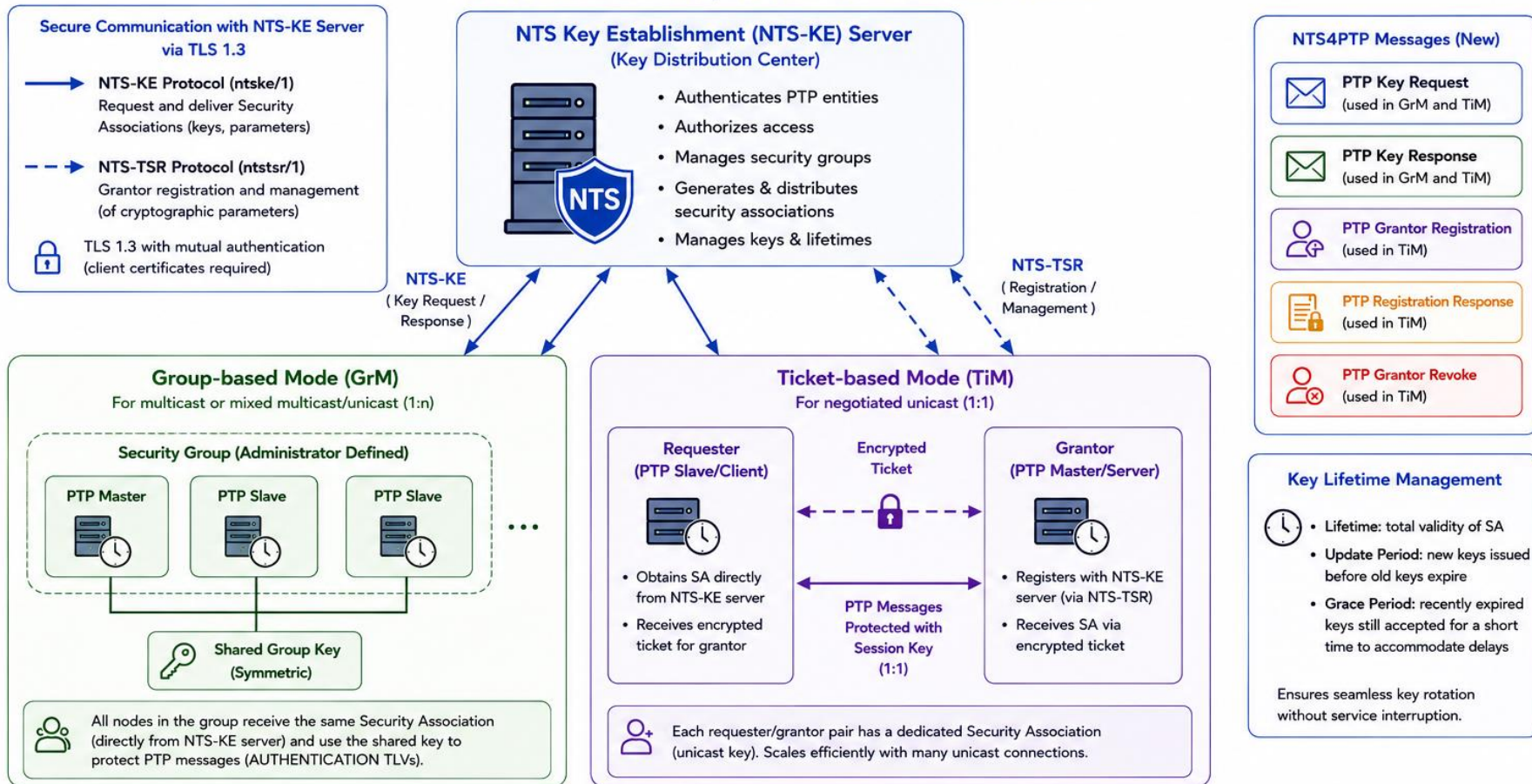
Add NTS as a key management scheme (NTS for PTP)

3

Add Data Sets for PTP Security

Proposed NTS4PTP Architecture Overview

Network Time Security for the Precision Time Protocol



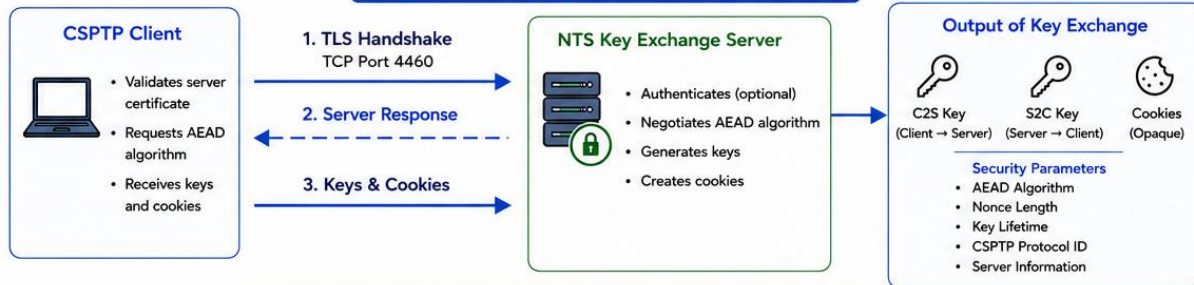
CSPTP (with NTS based security)

- A starting CSPTP draft has been selected from multiple proposals.
- Discussions on updates to that draft are ongoing.
- Plan to incorporate security in the initial version of standard.
- Initial proposal on the use of NTS for CSPTP has been submitted
- Preliminary "proof of concept" implementation work done (Hackathon project @ IETF)
 - <https://wiki.ietf.org/en/meeting/123/hackathon>

Possible NTS for CSPTP Architecture

Adapting Network Time Security (RFC 8915) for Client Server Precision Time Protocol

PHASE 1 – NTS KEY EXCHANGE (TLS over TCP)



Key Design Highlights



correctionField is NOT authenticated (may change in network). Client validates it within reasonable bounds.



Follow_Up messages require additional validation to ensure consistency with Sync.

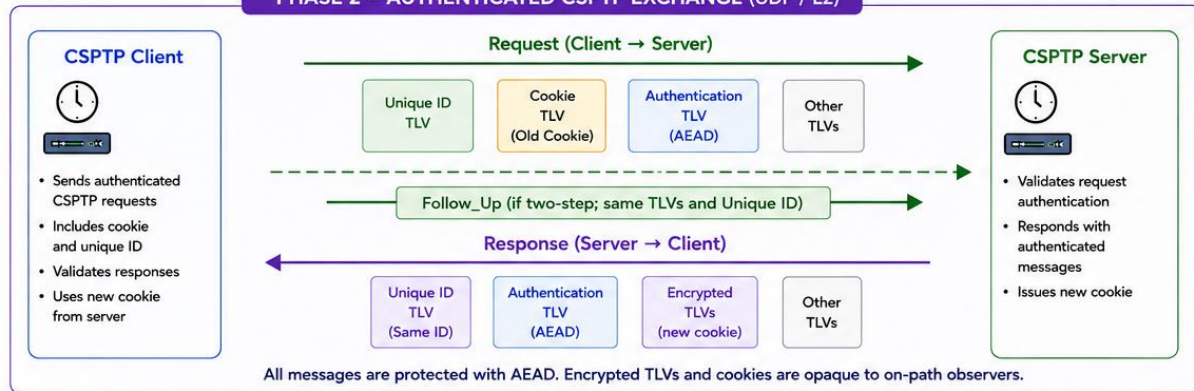


Cookies provide stateless authentication and privacy.



AEAD provides integrity, authentication and replay protection.

PHASE 2 – AUTHENTICATED CSPTP EXCHANGE (UDP / L2)



TLVs Used

Unique ID TLV

Links all messages in a request/response exchange (Sync and Follow_Up).

Cookie TLV

Carries opaque cookie identifying the security context (not encrypted).

Authentication TLV

Contains nonce and AEAD ciphertext to authenticate the CSPTP message.

Security Properties



Server Authentication



Message Integrity



Replay Resistance



Privacy Protection



Stateless Scalability

Threats Addressed



Spoofing



Modification



Replay



Delay Attacks* (Mitigated)

* Delay attacks cannot be fully prevented; clients should reject measurements with excessive path delay.

Help Wanted!



- Implementation experience:
 - Identify existing implementations and products?
 - Any errors in the specification? What was confusing or hard to implement?
 - Any open source repos to help others?
- Testing and Measurements
 - Test performance impacts in different configuration.
 - Any internet scale experiments that could be conducted? (think Ripe Atlas)
- Deployment projects
 - What were the road bumps along the way? What is missing in the existing specifications? (e.g. generic interface to NTS key management server)
- Operational experience
 - Identify existing deployments? Any feedback.

Final Remarks

- A great deal has been accomplished in the last five years.
- Solutions are available **now** to address many PTP security concerns.
- IEEE 1588k will fix some existing problems and add more functionality for PTP.
- NTS is gaining traction in the NTP community.
- NTS can be applied to both PTP and the emerging CSPTP.
- Reuse helps implementation and deployment.
- Ongoing work to specify the details.
- **Feedback from implementation, experiments, and deployment experiences is really helpful.**

Acknowledgements and Thanks...



- In particular, David Venhoek, Steffen Fries, Martin Langer, Rainer Bermbach, Doug Arnold, and Dieter Sibold.
- All IEEE 1588 Contributors
- All NTP NTS Contributors
- All errors and omissions are mine.

Thank you!

Any
Questions?

The logo for Meinberg, featuring the word "MEINBERG" in a stylized, italicized blue font. The letters are bold and slanted to the right. The logo is contained within a white rectangular box with rounded corners and a blue border. The background of the slide is dark blue with a faint geometric pattern of lines and dots.

The Synchronization Experts.

- Karen O'Donoghue
- karen.odonoghue@meinberg-usa.com