Network Time Protocols
Internet-Draft
Intended status: Standards Track
Expires: 16 February 2026

M. Langer
PTB
R. Bermbach
Ostfalia University

15 August 2025

NTS4PTP - Network Time Security for the Precision Time Protocol
draft-ietf-ntp-nts-for-ptp-02

Abstract

This document specifies an automatic key management service for the integrated security mechanism (prong A) of IEEE Std 1588邃「-2019 (PTPv2.1) described there in Annex P. This key management follows the immediate security processing approach of prong A and extends the NTS Key Establishment protocol defined in IETF RFC 8915 for securing NTPv4. The resulting NTS for PTP (NTS4PTP) protocol provides a security solution for all PTP modes and operates completely independent of NTPv4. It also provides measures against known attack vectors targeting PTP.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 16 February 2026.

Copyright Notice

Table of Contents

1. Introduction

In its Annex P the IEEE Std 1588-2019 ([IEEE1588-2019], Precision Time Protocol version 2.1, PTPv2.1) defines a comprehensive PTP security concept based on four prongs (A to D). Prong A incorporates an immediate security processing approach and specifies in section 16.14 an extension to secure PTP messages by means of an AUTHENTICATION TLV (AuthTLV) containing an Integrity Check Value (ICV). For PTP instances to use the securing mechanism, a respective key needs to be securely distributed among them. Annex P gives requirements for such a key management system and mentions potential candidates without further specification, but allows other solutions as long as they fulfill those requirements.

Since many time server appliances support both, the Precision Time Protocol (PTP) and the Network Time Protocol (NTP), it should be easier for the manufacturer of these devices and the network operator if PTP and NTP use a key management system based on the same technology. The Network Time Security (NTS) protocol was specified by the Internet Engineering Task Force (IETF) to protect the integrity of NTP messages [RFC8915]. Its NTS Key Establishment sub-protocol is secured by the Transport Layer Security (TLS 1.3, IETF RFC 8446 [RFC8446]) mechanism. TLS is used to protect numerous popular network protocols, so it is present in many networks.

This document specifies an automatic key management service, NTS for PTP, short NTS4PTP, for the immediate security processing in prong A. The solution [Langer_et_al._2022], [Langer_et_al._2020] is based on and expands the NTS Key Establishment protocol defined in IETF RFC 8915 [RFC8915] for securing NTP, but works completely independent of NTP. In addition, this document introduces a new sub-protocol, the NTS Time Server Registration (NTS-TSR) protocol, defining the communication between PTP unicast servers (grantors) with the NTS-Key Establishment server (NTS-KE server). (In NTS for NTP the specification of the communication between NTS time server and NTS-KE server has been left open.) Figure 1 depicts the participants of the NTS4PTP protocol and the sub-protocols they use.

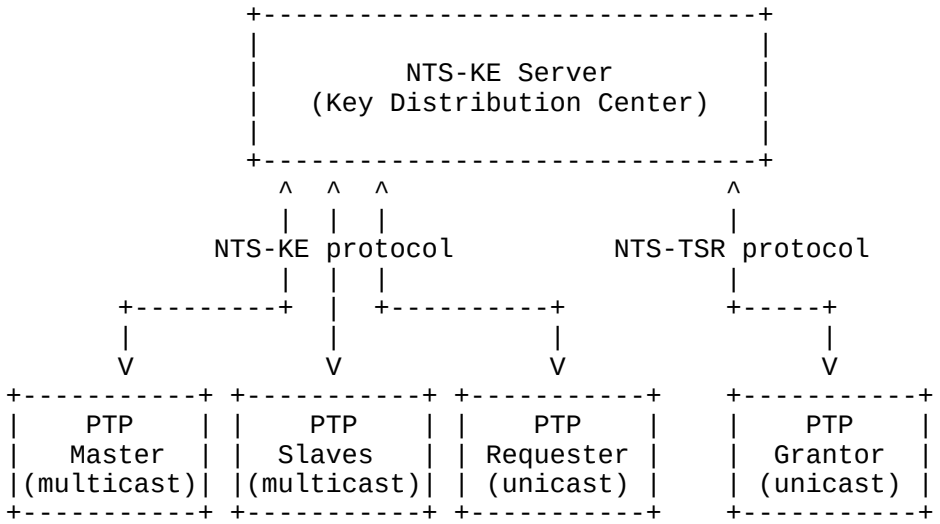


Figure 1: Communication of PTP instances with the NTS-KE server using the NTS- KE and NTS-TSR sub-protocols

For PTP multicast communication the PTP grandmaster as well as all participating PTP slaves use the NTS-KE protocol to obtain the security association (SA), i.e. key, lifetime etc. for a specific group. PTPv2.1 does not know groups, but distinguishes between PTP domains and profiles in order to separate different PTP networks from each other. NTS4PTP allows the administrator to freely define groups, be it using domains and profiles or any other method to assign the logically separated PTP networks to their own SA (see first paragraph of Section 2.3). For such PTP multicast or mixed multicast/unicast communication, NTS4PTP defines the group-based mode, short GrM.

For securing a PTP unicast communication a potential grantor (time server) uses the NTS-TSR protocol to register with the NTS-KE server. Thereby, ticket key, lifetime etc. for encrypting a so-called ticket are exchanged. A potential PTP unicast client (requester) then again uses the NTS-KE protocol to obtain the security association, i.e. unicast key, lifetime etc., as well as an encrypted ticket for the unicast communication with the specific grantor from the NTS-KE server. Thereafter, the ticket is transported from requester to grantor attached to a PTP signaling message [IEEE1588-2019] to establish a so-called unicast contract for delivering PTP time information. The (ticket key-) encrypted ticket holds all necessary information for the grantor to identify the requester as well as the (unicast) key used to secure and check the PTP messages between them. For this PTP unicast communication (also called negotiated PTP unicast), NTS4PTP defines the ticket-based mode, short TiM.

Though the key management for PTP is based on the NTS Key Establishment (NTS-KE) protocol for NTP, it works completely independent of NTP. The key management system uses the procedures described in IETF RFC 8915 for the NTS-KE protocol and expands it with new NTS messages for PTP. It may be applied in a key establishment server that already manages NTP but can also be operated only handling key establishment for PTP. Even when the PTP network is isolated from the Internet, a key establishment server can be installed in that network providing the PTP instances with necessary key and security parameters.

The NTS-KE server may often be implemented as a separate unit. It also may be collocated with a PTP instance, e.g., the Grandmaster. In the latter case communication between the NTS-KE server program and the PTP instance program needs to be implemented in a secure way if TLS communication (e.g., via local host or inter-process communication) is not or cannot be used.

Using the expanded NTS Key Establishment protocol and the newly defined NTS Time Server Registration protocol for the NTS key management for PTP, NTS4PTP provides the two principle approaches specified in this document:

1. Group-based mode (GrM)

* suitable for the PTP multicast and mixed multicast/unicast communication model,
* definition of one or more security groups in the PTP network,
* designed to secure 1:n communication

2. Ticket-based mode (TiM)

* suitable for the PTP unicast communication model between a PTP requester and grantor,
* designed to secure 1:1 communication

For these modes, the NTS key management for PTP defines six new NTS messages, see Figure 2. All messages are constructed from specific records as described in (see Section 4):

* PTP Key Request message (use in GrM and TiM, see Section 3.1)
* PTP Key Response message (use in GrM and TiM, see Section 3.2)
* PTP Registration Request message (use in TiM, see Section 3.3)
* PTP Registration Response message (use in TiM, see Section 3.4)
* PTP Registration Revoke message (use in TiM, see Section 3.5)

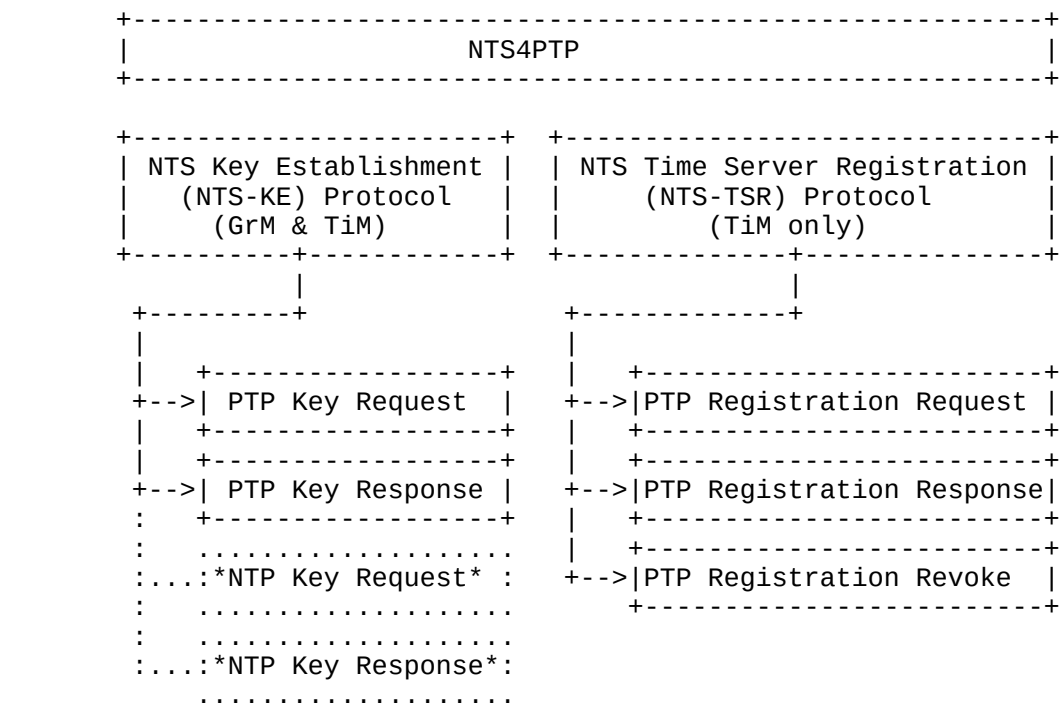


Figure 2: The new messages of NTS4PTP

1.1. Security Goals and Limitations

The security measures described in section 16.14 of the PTPv2.1 standard [IEEE1588-2019] focus in particular on the requirements that a key management system for PTP must meet to enable the protection of PTP messages. The application of the exchanged security parameters by the key management system is currently not sufficiently specified in IEEE Std 1588-2019 to protect the PTP messages against replay attacks, start-up replay and spoofing attacks. Therefore, this document describes its own mechanisms for applying the exchanged security parameters in PTP to secure the PTP messages. Specification gaps in section 16.14 of the PTPv2.1 standard do not affect NTS4PTP.

However, it should be emphasized that complete protection of the Precision Time Protocol is not technically feasible, since time distribution is primarily based on one-way time transmission. This technology is fundamentally vulnerable to delay attacks and cannot be prevented by any cryptographic means. The PTPv2.1 standard describes

other mechanisms in Annex P, such as network redundancy and monitoring, to mitigate these attacks (see also [Langer_2023], section 4.3.2 ff, and [Langer_et_al._2019]). Nevertheless, these measures are outside the scope of a cryptographic security solution.

NTS4PTP provides authenticity and integrity of PTP messages as well as protection against attacks such as packet manipulation and replay (see also [Langer_et_al._2022]). NTS4PTP does not provide protection against delay attacks.

1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.3. Terms and Abbreviations

Term	Description
AEAD	Authenticated Encryption with Associated Data [RFC5116]
AES	Advanced Encryption Standard, also: Rijndael
Authentication TLV (AuthTLV)	PTPv2.1 extension that provides authenticity and integrity protection for PTP messages [IEEE1588-2019]
ALPN	Application-Layer Protocol Negotiation [RFC7301]
CMAC	Cipher-based Message Authentication Code, see also MAC
Container, Container records	Container records (short: container) comprise a set of NTS records in its record body that serve a specific purpose, e.g., the Current Parameters container record.
CSPRNG	Cryptographically Secure Pseudorandom Number Generator
DoS	Denial of Service

DDoS	Distributed Denial of Service
GMAC	Galois Message Authentication Code, see also MAC
Grace Period	Defines a period of time during which security parameters are accepted for a short time after their lifetime has expired
Group	NTS4PTP uses the term to describe PTP entities in a PTP multicast setup (e.g., master, slave, ...) that are authorized for a common security association to secure and verify PTP messages between them.
GrM	Group-based mode of NTS4PTP
Group Key	Key used for authentication of PTP messages in group-based mode (GrM)
HMAC	Hash-based Message Authentication Code, see also MAC
ICV	Integrity Check Value, result of a cryptographic function used to detect unauthorized modifications of a PTP message, field in the Authentication TLV
IEEE 802.3	Standards collection defining the physical layer and data link layer's media access control (MAC) of wired Ethernet, transport mode in PTP
IP, IPv4, IPv6	Internet Protocol, network layer communications protocol, version 4 or version 6, part of the Internet protocol suite
IV	Initialization Vector, for example used with some MAC algorithms
Lifetime	Specifies the validity period of the security parameters in seconds, which is counted down
MAC address	Medium Access Control address, unique identifier used as a network address within a network segment
MAC algorithm	Message Authentication Code, short piece of

	information used for authenticating and integrity-checking of a message
NTP	Network Time Protocol [RFC5905]
NTS4PTP	NTS for PTP, variant of NTS to provide key management to PTP
NTS	Network Time Security [RFC8915]
NTS-KE	Network Time Security Key Establishment protocol
NTS-TSR	Network Time Security Time Server Registration protocol
OCSP	Online Certificate Status Protocol [RFC6960]
PKI	Public Key Infrastructure
PortIdentity	Specifies a specific PTP port
PTP	Precision Time Protocol [IEEE1588-2019]
Record, NTS record	Special NTS type-length-value data structure defining specific parameters; records build the respective NTS messages (differs from the TLV format of PTP)
SA	Security Association, description of the set of security parameters necessary to provide security services (e.g., authentication and integrity) between different entities sharing the same SA
SAD	Security Association Database
sdoId	Standards Development Organization Identifier, attribute for providing isolation of PTP Instances using different PTP profiles; in NTS4PTP it may form the group number in combination with the PTP domain number
SPP	Security Parameter Pointer
TCP	Transmission Control Protocol, part of the Internet protocol suite

Ticket	NTS record which contains the encrypted security parameters that a grantor needs for a secured PTP unicast connection to the requester
Ticket Key	Encryption key for the ticket, negotiated between NTS-KE server and grantor during the registration process (different from unicast key)
TC, Transparent Clock	Device in a PTP network with multiple PTP ports (switch) which measures its transit time and provides it in a correction field of the PTP message
TiM	Ticket-based mode for NTS4PTP
TLS	Transport Layer Security [RFC8446]
TLV	Data set containing a type, length, and value field. Used in PTPv2.1 [IEEE1588-2019], compare to Authentication TLV and Ticket TLV
UDP	User Datagram Protocol, part of the Internet protocol suite
Unicast Key	Used to secure the PTP messages between requester and grantor (different from ticket key)
Update Period	During the update period new security parameters are available at the NTS-KE server, resp. grantors should re-register with the NTS-KE server
X.509	Standard to form X.509 certificates [ITU-T_X.509]

Table 1: Terms and abbreviations

2. Key Management for PTP Using Network Time Security

After the rundown of the different PTP instances and the sub-protocols they use for communication with the NTS Key Establishment (NTS-KE) server in the introduction, the following sections specify the setup and use of TLS 1.3 to secure the communication with the NTS-KE server, before the message exchange for both, the group-based mode as well as the ticket-based mode is described in detail in Section 2.3 and Section 2.4. More general topics as key update, authentication and authorization etc. are covered in Section 2.5.

2.1. Setup of a TLS Communication Channel with the NTS-KE Protocol

TLS is a layer five protocol that runs on TCP over IP. Therefore, PTP implementations that support NTS-based key management need to support TCP and IP (at least on a separate management port).

A PTP instance wanting to request a key using the NTS-KE protocol defined in [RFC8915], first starts a TLS 1.3 connection to the NTS-KE server.

The PTP instance connects to the NTS-KE server on the NTS TCP port (port number 4460). Then both parties perform a TLS handshake to establish a TLS 1.3 communication channel. The details of the TLS handshake are specified in IETF RFC 8446 [RFC8446].

Implementations MUST conform to the rules stated in Section 3 "TLS Profile for Network Time Security" of IETF RFC 8915 [RFC8915].

The client starts the TLS 1.3 handshake with a ’Client Hello’ message to the NTS-KE server containing the Application Layer Protocol Negotiation (ALPN) [RFC7301] extension containing "ntske/1", which refers to the NTS Key Establishment as the subsequent protocol. The server responds with a "Server Hello" message sending its certificate and feasible cipher suites as well as requesting the client’s certificate using a TLS ’CertificateRequest’. (The latter does not conflict to the procedure in NTS for NTP.)

Afterwards, the client authenticates the server using the root CA certificate or by means of the Online Certificate Status Protocol (OCSP, IETF RFC 6960) [RFC6960]. In the same way, the server authenticates the client, if it had sent its certificate (which is always necessary with NTS4PTP, in contrast to NTS for NTP.) After the authentication procedure both, client and server agree on the cipher suite and then establish a secured channel that ensures authenticity, integrity and confidentiality for subsequent NTS messages.

Once the TLS session is established, the PTP instance will ask for a key as well as the associated security parameters using the new NTS message PTP Key Request (see Section 3.1). The NTS-KE server will respond with a PTP Key Response message (see Section 3.2).

NTS for NTP postulates in IETF RFC 8915 [RFC8915] that after completion of a request/response sequence the TLS session is to be closed. For NTS4PTP the same procedure can be used. Additionally, the NTS-KE server may keep the TLS session open until a short timeout configured by the admin expires or the ’close notify’ arrives. This allows the PTP instance to make another NTS request without starting a new TLS handshake. Finally, the NTS-KE server also sends a ’close notify’ to the PTP instance and closes the TLS channel.

With the key and other information received, the PTP instance can take part in the secured PTP communication in the different modes of operation.

After the reception of the first set of security parameters the PTP instance may resume the TLS session according to IETF RFC 8446 [RFC8446], Section 4.6.1, allowing the PTP instance to skip the TLS version and algorithm negotiations. If TLS Session Resumption ([RFC8446], Section 2.2) is used and supported by the NTS-KE server, a suitable lifetime (max. 24 hrs) for the TLS session key MUST be defined to not open the TLS connection for security threats. If the NTS-KE server does not support TLS resumption, a full TLS handshake MUST be performed.

As the TLS session provides authentication, but not authorization additional means have to be used for the latter (see Section 2.5.5.4).

2.2. Setup of a TLS Communication Channel with the NTS-TSR Protocol

As already mentioned and shown in Figure 1 and Figure 2, the new NTS Time Server Registration protocol is used for registering a grantor with the NTS-KE server (ticket-based mode only). Thereby, the new messages PTP Registration Request message, PTP Registration Response message and PTP Registration Revoke message are applied.

The setup of the TLS channel in this ticket-based mode (TiM) is handled in the same way as described above for the NTS-KE protocol, see Section 2.1. The only difference lies in the ALPN used, which is now "ntstsr/1".

Once the TLS session is established, the grantor will register with the NTS-KE server using the NTS message PTP Registration Request (see Section 3.3). The NTS-KE server will respond with a PTP Registration Response message (see Section 3.4) containing ticket key, lifetime etc.

When the PTP Registration Request message was responded with a PTP Registration Response, the TLS session is closed.

Also using a TLS connection with the NTS-KE server a grantor can cancel its registration with a PTP Registration Revoke message (see Section 3.5).

2.3. NTS Message Exchange for Group-Based Mode

As described in Section 2.1, a PTP instance wanting to join a secured PTP communication in the group-based modes contacts the NTS-KE server starting the establishment of a secured TLS connection using the NTS-KE protocol (ALPN: ntske/1). Then, the client continues with a PTP Key Request message (see Section 3.1), asking for for a security association of a specific group (GrM-SA) as shown in Figure 3. The NTS-KE server identifies the respective sub-protocol by means of the ALPN and analyzes the contents of the Next Protocol Negotiation record. If it is PTP the server examines whether the client had sent its certificate and that it is valid. Finally, it checks whether the client is authorized to join the requested group. If everything is ok the NTS-KE server generates the respective PTP Key Response message (see Section 3.2) for the requesting client with all the necessary data to join the group communication. Else, it contains a respective error code if the PTP instance is not allowed to join the group. This procedure is necessary for all parties, which are or will be members of that PTP group including the Grandmaster and other special participants, e.g., Transparent Clocks. As mentioned above, this not only applies to the multicast communication model but also to mixed multicast/unicast communication (former hybrid mode) where the explicit unicast communication uses the multicast group key received from the NTS-KE server. The group number for both modes is defined by the administrator, as described in Section 4.2.2.

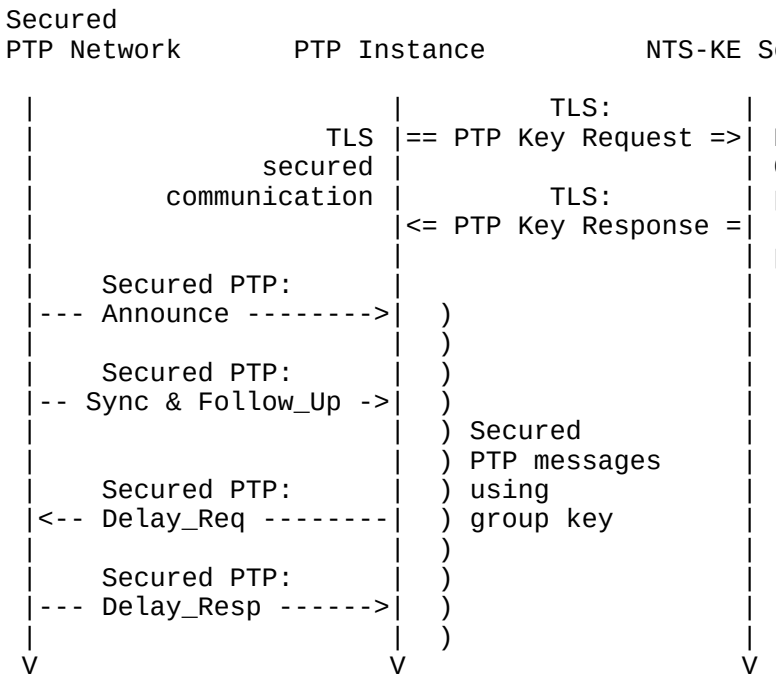


Figure 3: Message exchange for the group-based mode (GrM)

After the NTS Key Establishment messages for the group-based mode (GrM) have been exchanged, the secured PTP communication can take place using the security association(s) communicated. The participants of the PTP network are now able to use the group key to verify secured PTP messages of the corresponding group or to generate secured PTP messages itself. In order to do this, the PTP node applies the group key together with the MAC algorithm to the PTP packet to generate the ICV transported in the AUTHENTICATION TLV of the PTP message.

The key management for this mode works relatively simple and needs only the above mentioned two NTS messages: PTP Key Request and PTP Key Response.

2.4. NTS Message Exchange for the Ticket-Based Mode

The ticket-based mode (TiM) for negotiated unicast connections ensures end-to-end security between the two PTP communication partners, requester and grantor, and is therefore only suitable for PTP unicast where no group binding exists. Thus, this model scales excellently with the number of connections. TiM also allows free MAC algorithm negotiation.

In PTP unicast mode using unicast message negotiation ([IEEE1588-2019], Section 16.1) any potential instance (the grantor) which can be contacted by other PTP instances (the requesters) needs to register upfront with the NTS-KE server as depicted in Figure 4. For the registration, again a TLS channel has to be set up using the new NTS Time Server Registration sub-protocol with the ALPN "ntstsr/1" as described in Section 2.2. This also ensures the mutual authentication of grantor and NTS-KE server.

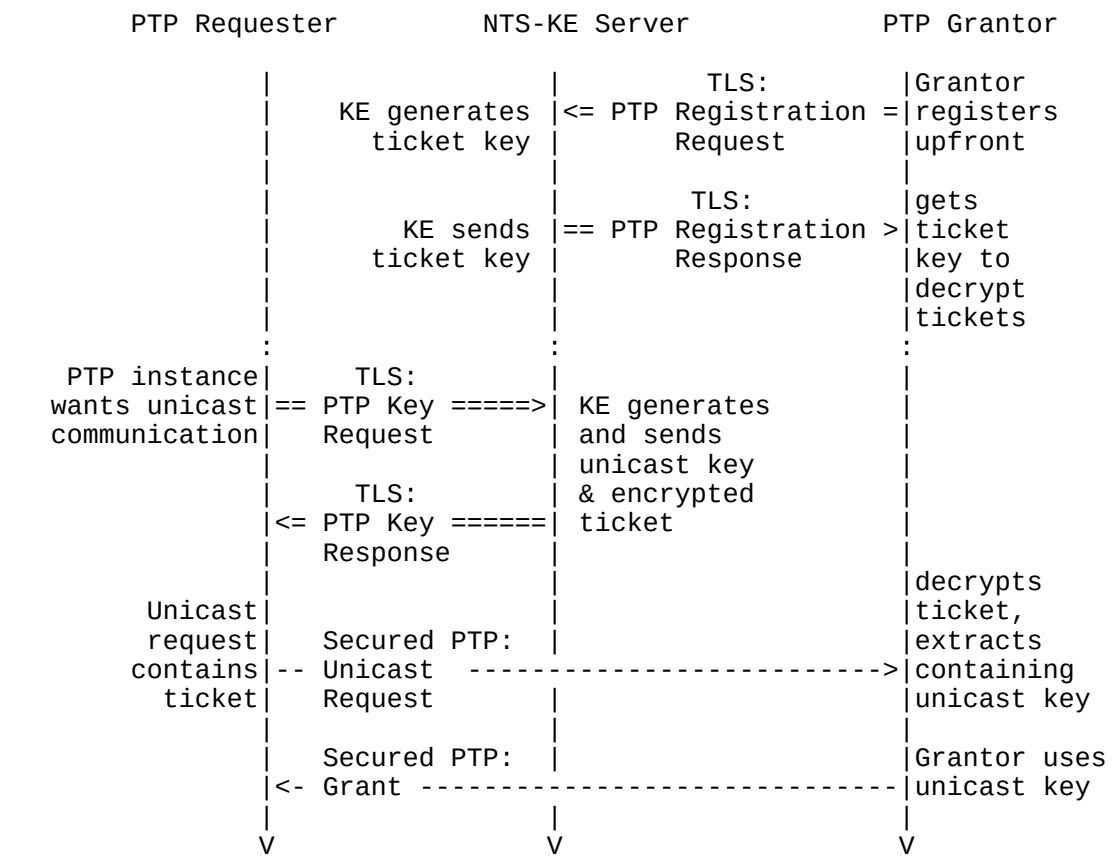


Figure 4: Message exchange for ticket-based unicast mode (TiM)

(Note: As any PTP instance may request unicast messages from any other instance the terms requester and grantor as used in the standard suit better than talking about slave respectively master. In unicast PTP, the grantor is typically a PTP port in the MASTER state, and the requester is typically a PTP port in the SLAVE state. However, all PTP ports are allowed to grant and request unicast PTP message contracts regardless of which state they are in. A PTP port in MASTER state may be requester, a port in SLAVE state may be a grantor.)

The registration of a PTP grantor is performed via a PTP Registration Request message (see Section 3.3). The NTS-KE server answers with a PTP Registration Response message (see Section 3.4). If no delivery of security data is possible for whatever reason, the PTP Registration Response message contains a respective error code.

With the reception of the PTP Registration Response message, the grantor holds a ticket key known only to the NTS-KE server and the registered grantor. With this ticket key it can decrypt cryptographic information contained in a so-called ticket which enables secure unicast communication.

After the end of the registration process (phase 1), phase 2 begins with the PTP key request of the client (here called requester). Similar to the group-based mode, the requester wanting to start a secured PTP unicast communication with a specific grantor contacts the NTS-KE server sending a PTP Key Request message (see Section 3.1) as shown in Table 2, again using the TLS-secured NTS Key Establishment protocol. The NTS-KE server performs the authentication check of the client and then answers with a PTP Key Response message (see Section 3.2) with all the necessary data to begin the unicast communication with the desired partner or with a respective error code if unicast communication with that instance is unavailable. Though the message types are the same as in GrM the content differs.

In TiM the PTP Key Response message includes the TiM-SA with a unicast key to secure the PTP message exchange with the desired grantor. In addition, it contains the above mentioned (partially) encrypted ticket which the requester later (phase 3) transmits in the AUTHENTICATION TLV (see Section 5.1.3) with the secured PTP message to the grantor.

After the NTS Key Establishment messages for the PTP unicast mode have been exchanged, finally, the secured PTP communication (phase 3) can take place using the security association(s) communicated. A requester may send a (unicast key-) secured PTP signaling message containing the received encrypted ticket, asking for a grant of a so-called unicast contract which contains a request for a specific PTP message type, as well as the desired frame rate.

The grantor receiving the PTP message decrypts the received ticket with its ticket key and extracts the containing security parameters, for example the unicast key used by the requester to secure the PTP message and the requester’s identity. In that way the grantor can check the received message, identify the requester and can use the unicast key for further secure PTP communication with the requester until the unicast key expires.

A grantor that supports unicast and provides sufficient capacity will acknowledge the request for a unicast contract with a PTP unicast grant.

If a grantor is no longer at disposal for unicast mode during the lifetime of registration and ticket key, it sends a TLS-secured PTP Registration Revoke message (see Section 3.5, not shown in Figure 4) to the NTS-KE server, so requesters no longer receive security associations (key etc.) in PTP Key Response messages for this grantor. Instead, the NTS-KE server sends response messages with respective error codes.

For addressing a grantor, the requesting instance simply may use the grantor’s IP, MAC address or PortIdentity attribute.

2.5. General Topics

This section describes more general topics like key update and key generation as well as discussion of the time information on the NTS-KE server, the use of certificates and topics concerning upfront configuration.

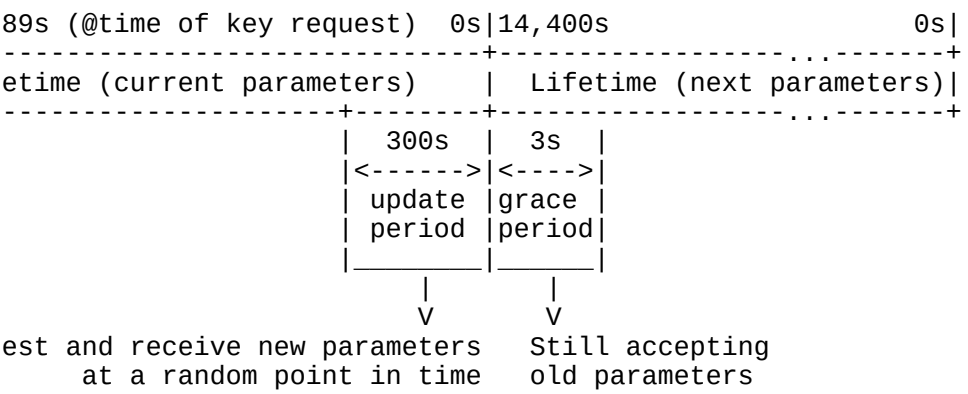
2.5.1. Key Update Process

The security parameters update process is an important part of NTS4PTP. It keeps the keys up to date, allows for both, runtime security policy changes and easy group control. The rotation operation allows uninterrupted PTP operation in GrM as well as in TiM.

The update mechanism is based on the Validity Period record in the NTS response messages, which includes the three values lifetime, update period and grace period, see Figure 5. The lifetime parameter specifies the validity period of the security parameters (e.g., security association (SA) and ticket) in seconds, which is counted down. This value can range from a few minutes to a few days. Due to the design of the replay protection, a maximum lifetime of many days is possible, but SHOULD NOT exceed 24h, see Section 4.2.16 and Section 5.2. After the validity period has expired, the security parameters may no longer be used to secure PTP messages and must be deleted soon after.

New security parameters are available on the NTS-KE server during the update period, a time span before the expiry of the lifetime. The length of the update period is therefore always shorter than the full lifetime and is typically in the range of a few minutes.

The grace period also helps to ensure uninterrupted key rotation. This value defines a period of time after the lifetime expiry during which the expired security parameters continue to be accepted. The grace period covers a few seconds at most and is only intended to compensate for runtime delays in the network during the update process. A maximum grace period of 5 seconds is recommended. The respective values of the three parameters are defined by the administrator and can also be specified by a corresponding PTP profile.



Example:
----
lifetime (full): 14,400s = 4h
update period: 300s = 5min
grace period: 3s

Figure 5: Example of the parameter rotation using lifetime, update period and grace period in group-based mode

As the value for lifetime is specified in seconds which denote the remaining time and is decremented down to zero, hard adjustments of the clock used have to be avoided. Therefore, the use of a monotonic clock is recommended. Requests during the currently running validity period will receive respectively adapted count values.

The Validity Period record (see Section 4.2.16) with its parameters lifetime, update period and grace period is contained in a so-called Current Parameters container record. Together with other security parameters this container record is always present in a PTP Key respectively Registration Response message. During the update period the response message additionally comprises the Next Parameters container record, which holds the new lifetime etc. starting at the end of the current lifetime as well as the other security parameters of the upcoming lifetime cycle.

Any PTP client sending a PTP Key Request to the NTS-KE server, be it in GrM to receive the group SA or be it in TiM asking for TiM-SA (unicast key etc. and encrypted ticket), will receive the Current Parameters container record where lifetime includes the remaining time to run rather than the full. Requesting during the update period the response includes also the new lifetime value etc. in the Next Parameters container record. The new lifetime is the full value of the validity starting at the end of the current lifetime and update period. After the old lifetime has expired, only the new parameters (including lifetime, update period and grace period) have to be used. Merely during the grace period, the old SA will be accepted to cope with smaller delays in the PTP communication.

All PTP clients are obliged to connect to the NTS-KE server during the update period to allow for uninterrupted secured PTP operation. To avoid peak load on the NTS-KE server all clients SHOULD choose a random starting time during the update period.

In TiM the unicast grantors execute the NTS-TSR protocol to register with the NTS-KE server. The rotation sequence (see Figure 6) and the behavior of the PTP Registration Response message is almost identical to the NTS-KE protocol. The main difference here is that the update period has to start earlier so that a grantor has re-registered before requesters ask for new security parameters at the NTS-KE server.

As the difference between the start of the requester’ s update period and the beginning of the update period of the grantor is not communicated, the grantor should contact the NTS-KE server directly after the start of its update period. However, since the rotation periods occur at different times for multiple grantors, no load peaks occur here either.

If a grantor does not re-register in time, requesters asking for a key etc. may not receive a Next Parameters container record, as no new SA is available at that point. So, requesters need to try again later.

As PTP unicast contracts in TiM run independently of the update cycle, a special situation may occur. If the remaining lifetime is short, the grantor decides whether it grants any contract longer than the remaining lifetime or not. If a unicast contract is to be extended within the update period and the requester already owns the new TiM-SA with the ticket, it MAY already apply the upcoming security parameters here. This allows the requester to negotiate the full time for the unicast contract with the grantor.

If a grantor has revoked his registration with a PTP Registration Revoke message, requesters will receive a PTP Key Response message with an error code when trying to update for a new TiM-SA. No immediate key revoke mechanism exists. The grantor SHOULD NOT grant respective unicast requests during the remaining lifetime of the revoked key.

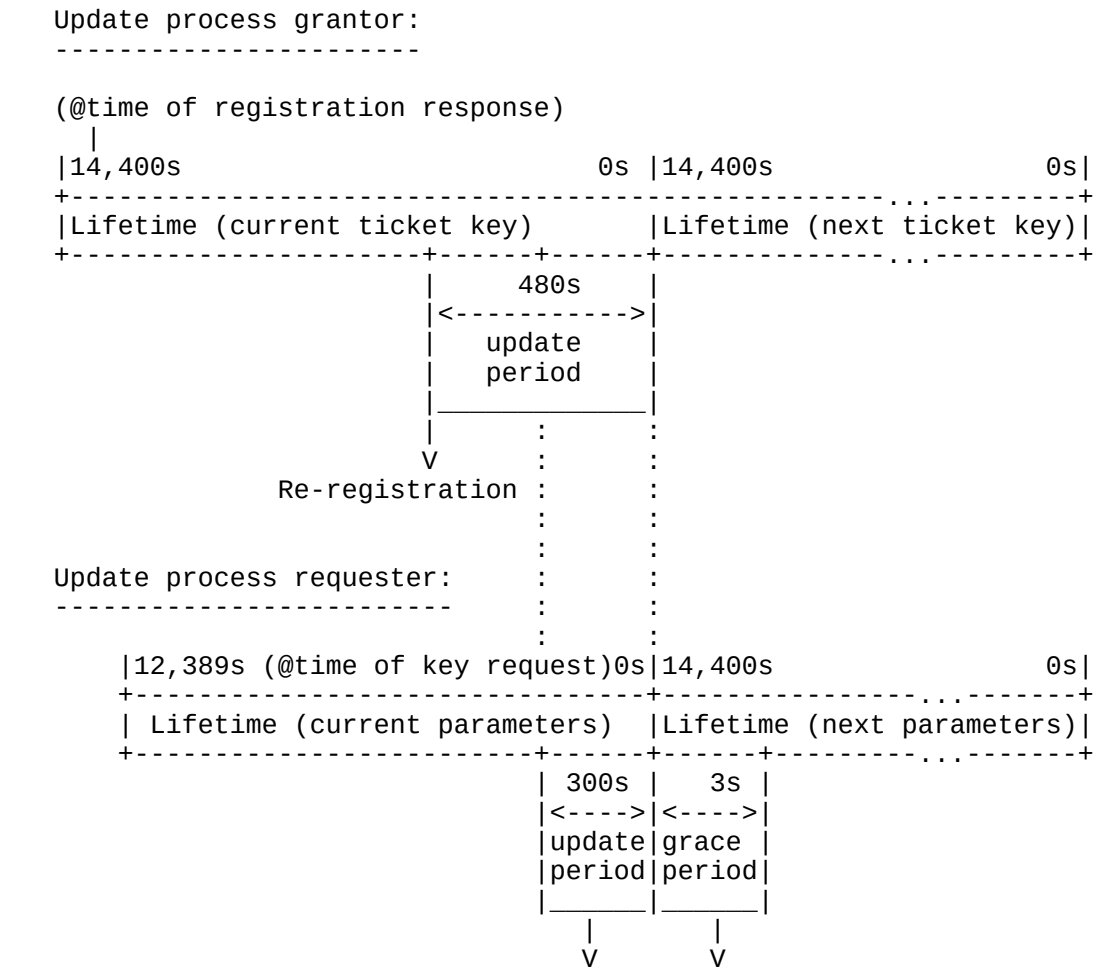


Figure 6: Example of the parameter rotation using lifetime and update period in ticket-based mode

2.5.2. Key Generation

In all cases keys obtained by a secure random number generator SHALL be used. The length of the keys depends on the cryptographic algorithm used (see also last subsection in Section 6.2).

2.5.3. Time Information of the NTS-KE server

As the NTS-KE server embeds time duration information in the respective messages, its local time should be accurate to within a few seconds compared to the controlled PTP network(s). To avoid any dependencies, it should synchronize to a secure external time source, for example an NTS-secured NTP server. The time information is also necessary to check the lifetime of certificates used.

2.5.4. Certificates

The authentication of the TLS communication parties is based on certificates issued by a trusted Certificate Authority (CA) that are utilized during the TLS handshake. In classical TLS applications only servers are required to have them. For the key management system described here, the PTP nodes also need certificates to allow only authorized and trusted devices to get the group key and join a secure PTP network. (As TLS only authenticates the communication partners, authorization has to be managed by external means, see the topic "Authorization" in Section 2.5.5.4.) The verification of a certificate always requires a loose time synchronicity, because they have a validity period. This, however, reveals the well-known start-up problem, since secure time transfer itself requires valid certificates. (See the discussion and proposals on this topic in IETF RFC 8915 [RFC8915], Section 8.5 "Initial Verification of Server certificates" which applies to client and server certificates in the PTP key management system, too.)

Furthermore, some kind of Public Key Infrastructure (PKI) is necessary, which may be conceivable via the Online Certificate Status Protocol (OCSP, IETF RFC 6960) [RFC6960] or other means as well as offline via root CA certificates.

The TLS communication parties must be equipped with a private key and a certificate in advance. The certificate contains a digital signature of the CA as well as the public key of the sender. The key pair is required to establish an authenticated and encrypted channel for the initial TLS phase. Distribution and update of the certificates can be done manually or automatically. However, it is important that they are issued by a trusted CA instance, which can be either local (private CA) or external (public CA).

For the certificates the standard for X.509 [ITU-T_X.509] certificates are to be used. Additional data in the certificates like domain, sdoId and/or GrM group attributes may help in authorizing. In that case it should be noted that using the PTP device in another network then implies to have a new certificate, too. Working with certificates without authorization information would not have that disadvantage, but more configuring at the NTS-KE server would be necessary: which domain, sdoId and/or GrM group attributes belong to which certificate.

As TLS is used to secure both sub-protocols, the NTS-KE and the NTS-TSR protocol, a comment on the security of TLS seems reasonable. A TLS 1.3 connection is considered secure today. However, note that a DoS (Denial of Service) attack on the key server can prevent new connections or parameter updates for secure PTP communication. A hijacked key management system is also critical, because it can completely disable the protection mechanism. A redundant implementation of the key server is therefore essential for a robust system. A further mitigation can be the limitation of the number of TLS requests of single PTP nodes to prevent flooding. But such measures are out of the scope of this document.

2.5.5. Upfront Configuration

All PTP instances as well as the NTS-KE server need to be configured by the network administrator. This applies to several fields of parameters.

2.5.5.1. Security Parameters

The cryptographic algorithm and associated parameters (the so-called security association(s) SA) used for PTP keys are configured by network operators at the NTS-KE server. PTP instances that do not support the configured algorithms cannot operate with the security. Since most PTP networks are managed by a single organization, configuring the cryptographic algorithm (MAC) for ICV calculation is practical. This prevents the need for the NTS-KE server and PTP instances to implement an NTS algorithm negotiation protocol.

For the ticket-based mode the AEAD algorithms need to be specified which the PTP grantors and the NTS-KE server support and negotiate during the registration process. Optionally, the MAC algorithm may be negotiated during a unicast PTP Key Request to allow faster or stronger algorithms, but a standard algorithm supported by every instance should be defined. Eventually, suitable algorithms may be defined in a respective PTP profile.

2.5.5.2. Key Lifetimes

Supplementary to the above mentioned SAs the desired key rotation periods, i.e., the lifetimes of keys respectively all security parameters need to be configured at the NTS-KE server. This applies to the lifetime of a group key in the group-based mode as well as the lifetime of ticket key and unicast key in the ticket-based mode (typically for every unicast pair in general). In addition, the corresponding update periods and grace periods need to be defined. Any particular lifetime, update period and grace period is configured as time spans specified in seconds.

2.5.5.3. Certificates

The network administrator has to supply each PTP instance and the NTS-KE server with their X.509 certificates. The TLS communication parties must be pre-equipped with a private key and a certificate containing the public key (see Section 2.5.4).

2.5.5.4. Authorization

The certificates provide authentication of the communication partners. Normally, they do not contain authorization information. Authorization decides, which PTP instances are allowed to join a group (in any of the group-based modes) or may enter a unicast communication in the ticket-based mode and request the respective SA(s) and key.

As mentioned, members of a group (multicast communication model, mixed multicast/unicast communication model) may be identified by their domain and their sdoId or a self-defined scheme. So, PTP domain and sdoId may be attributes in the certificates of the potential group members supplying additional authorization. If not contained in the certificates extra authorization means are necessary. (See also the discussion on advantages and disadvantages on certificates containing additional authorization data in Section 2.5.4.)

In TiM, any authenticated grantor that is an authorized GrM group member may request a registration for unicast communication at the NTS-KE server (implicit authorization). If no group authorization is available (e.g., unicast only operation) another authentication scheme is necessary.

In the same way, any requester may request security parameters for a unicast connection with a specific grantor. Only authentication at the NTS-KE server using its certificate and membership in the GrM group is needed (implicit authorization). If a unicast communication

is not desired by the grantor, it should not grant a specific PTP unicast request. Again, if no group authorization is available (e.g., unicast only operation) another authentication scheme is necessary.

Authorization can be executed at least in some manual configuration. Probably the application of a standard access control system like Diameter, RADIUS or similar would be more appropriate. Also role-based access control (RBAC), attribute-based access control (ABAC) or more flexible tools like Open Policy Agent (OPA) could help administering larger systems. But details of the authorization of PTP instances lie out of scope of this document.

2.5.5.5. Transparent Clocks

In GrM, Transparent Clocks (TC) need to be supplied with respective certificates for authentication, too. They need to request for the relevant GrM-SA(s) at the NTS-KE server to allow secure use of the correction field in a PTP message and generation of a corrected ICV.

In addition, authorization of TCs for the respective GrM groups is paramount. Otherwise the security can easily be broken with attackers pretending to be TCs in the path.

Transparent clocks may notice that the communication runs secured. In GrM they request a group key from the NTS-KE server. Afterwards they can check the ICV of incoming messages, fill in the correction field and generate a new ICV for outgoing messages.

2.5.5.6. Start-up considerations

At start-up of a single PTP instance or the complete PTP network, some issues have to be considered.

At least loose time synchronization is necessary to allow for authentication using the certificates. See the discussion and proposals on this topic in IETF RFC 8915 [RFC8915], Section 8.5 "Initial Verification of Server certificates" which applies to client and server certificates in the PTP key management system, too.

To avoid peak loads on the NTS-KE server, PTP instances SHALL contact the NTS-KE server at a random time after start-up, similar to a PTP key re-request during an update period. Every grantor must register with the NTS-KE server before requesters can request a TiM-SA.

To avoid start-up replay attacks starting PTP instances should follow the procedure described in Section 5.2.

3. NTS Messages for PTP

This section describes the structure of the specific NTS messages for the PTP key management. Table 2 to Table 10 specify which records the messages are composed of. The Mode column indicates the intended use of the particular record for the respective PTP communication mode. The next column informs whether the respective record is mandatory or optional. The reference column in the tables refer to the specific subsections of the record specification. The right column shows typical values as an example.

More details especially on the records the messages are built of and their types, sizes, requirements and restrictions are given in Section 4.

The NTS messages MUST contain the records given for the particular message, though not necessarily in the same sequence indicated. Only the End of Message record MUST be the final record.

3.1. PTP Key Request Message

Table 2 shows the record structure of a PTP Key Request message. The message starts with the NTS Next Protocol Negotiation record, which in this application always holds PTPv2.1. The following Association Mode record describes the mode how the PTP instance wants to communicate: In GrM the desired group number is given. In TiM the Association Mode contains the identification of the desired grantor, for example IPv4 and its IP address.

PTP Key Request (NTS-KE protocol)

NTS Record Name	Mode	Use	Reference	Exemplary body contents
NTS Next Protocol Negotiation	GrM / TiM	mandatory	Section 4.2.7	PTPv2.1
Association Mode	GrM / TiM	mandatory	Section 4.2.2	(Association Type || Association Value)
Supported MAC Algorithms	TiM	optional	Section 4.2.12	CMAC || HMAC
Source	TiM	mandatory	Section	{binary data}

PortIdentity			4.2.11	
End of Message	GrM / TiM	mandatory	Section 4.2.4	(no record body)

Table 2: Record structure of the PTP Key Request message

Only in TiM, an optional record may follow. It offers the possibility to choose from additional MAC algorithms and presents the supported algorithms from which the NTS-KE server may choose. Again, only in ticket-based mode, the Source PortIdentity record gives the data of the identification of the applying requester, for example IPv4 and its IP address. The messages always end with an End of Message record.

3.2. PTP Key Response Message

Table 3 shows the record structure of a PTP Key Response message from the NTS-KE server (NTS-KE protocol). The message starts with the NTS Next Protocol Negotiation record which in this application always holds PTPv2.1.

PTP Key Response (NTS-KE protocol)

NTS Record Name	Mode	Use	Reference	Exemplary body contents
NTS Next Protocol Negotiation	GrM / TiM	mandatory	Section 4.2.7	PTPv2.1
Current Parameters	GrM / TiM	mandatory	Section 4.2.3	set of records {...}
Next Parameters	GrM / TiM	mandatory (only during update period)	Section 4.2.6	set of records {...}
End of Message	GrM / TiM	mandatory	Section 4.2.4	(no record body)

Table 3: Record structure of the PTP Key Response message.

The following Current Parameters record is a container record holding, in separate records, all the security data required to join and communicate in the secured PTP communication during the current validity period. Table 5 shows the records incorporated in this container record, again with example contents in the right-hand column. For more details on the records included in the Current Parameters container record see Section 4.2.3.

If the request lies inside the update period, a Next Parameters container record is additionally appended in the PTP Key Response message giving all the security data needed for the upcoming validity period. Its structure follows the same composition as the Current Parameters container record. If that specific client is to be excluded from the group in the upcoming SA period no Next Parameters container SHALL be sent. In the event of an error, e.g., the requested grantor is not available, both parameters container records are removed and a single error record is inserted (see Table 4). The messages always end with an End of Message record.

PTP Key Response with Error (NTS-KE protocol)

NTS Record Name	Mode	Use	Reference	Exemplary

				body contents
NTS Next Protocol Negotiation	GrM / TiM	mandatory	Section 4.2.7	PTPv2.1
Error	GrM / TiM	mandatory	Section 4.2.5	Not authorized
End of Message	GrM / TiM	mandatory	Section 4.2.4	(no record body)

Table 4: Record structure of the PTP Key Response message in case of an error.

The structure of the respective container records (Current Parameters and Next Parameters) used in the PTP Key Response message is given below:

Current/Next Parameters container - PTP Key Response (NTS-KE protocol)

NTS Record Name	Mode	Use	Reference	Exemplary body contents
Security Association	GrM / TiM	mandatory	Section 4.2.10	data set {...}
Validity Period	GrM / TiM	mandatory	Section 4.2.16	{1560s||300s||3s}
PTP Time Server	TiM	mandatory	Section 4.2.9	data set {...}
Ticket	TiM	mandatory	Section 4.2.13	data set {...}

Table 5: Record structure of the container records

3.3. PTP Registration Request Message

The PTP Registration Request message (NTS-TSR protocol) starts with the NTS Message Type record containing the message type as well as the message version number, here always 1.0, see Table 6. (As the message belongs to the NTS-TSR protocol, no NTS Next Protocol Negotiation record is necessary.)

PTP Registration Request (NTS-TSR protocol)

NTS Record Name	Mode	Use	Reference	Exemplary body contents
NTS Message Type	TiM	mandatory	Section 4.2.8	PTP Registration Request||v1.0
PTP Time Server	TiM	mandatory	Section 4.2.9	data set {...}
AEAD Algorithm Negotiation	TiM	mandatory	Section 4.2.1	{AEAD_512||AEAD_256}
Supported MAC Algorithms	TiM	mandatory	Section 4.2.12	{CMAC||HMAC}
End of Message	TiM	mandatory	Section 4.2.4	(no record body)

Table 6: Record structure of the PTP Registration Request message

The PTP Time Server record presents all known network addresses of this grantor that are supported for a unicast connection. The following AEAD Algorithm Negotiation record indicates which algorithms for encryption of the ticket the grantor supports.

Then the next record (not optional as in PTP Key Request) follows, presenting all the grantor’s supported MAC algorithms. The Supported MAC Algorithms record contains a list of supported MAC algorithms by the grantor that are feasible for calculating the ICV when securing the PTP messages in TiM. The message always ends with an End of Message record.

3.4. PTP Registration Response Message

The PTP Registration Response message (NTS-TSR protocol) from the NTS-KE server starts with the NTS Message Type record containing the message type as well as the message version number, here always 1.0, see Table 7. (As the message belongs to the NTS-TSR protocol, no NTS Next Protocol Negotiation record is necessary.)

PTP Registration Response (NTS-TSR protocol)

NTS Record Name	Mode	Use	Reference	Exemplary body contents
NTS Message Type	TiM	mandatory	Section 4.2.8	PTP Registration Response||v1.0
Current Parameters	TiM	mandatory	Section 4.2.3	set of records {...}
Next Parameters	TiM	mandatory (only during update period)	Section 4.2.6	set of records {...}
End of Message	TiM	mandatory	Section 4.2.4	(no record body)

Table 7: Record structure of the PTP Registration Response message.

As in the NTS-KE protocol, the following Current Parameters record is a container record containing in separate records all the necessary parameters for the current validity period. Table 9 shows the records contained in that container record, again with exemplary contents in the right column. For more details on the records contained in the Current Parameters container record see Section 4.2.3.

PTP Registration Response with error (NTS-TSR protocol)

NTS Record Name	Mode	Use	Reference	Exemplary body contents
NTS Message Type	TiM	mandatory	Section 4.2.8	PTP Registration Response||v1.0

Error	TiM	mandatory	Section 4.2.5	Not authorized
End of Message	TiM	mandatory	Section 4.2.4	(no record body)

Table 8: Record structure of the PTP Registration Response message in case of an error.

The structure of the respective container records (Current Parameters and Next Parameters) used in the PTP Registration Response message is given below:

Current/Next Parameters container - PTP Registration Response (NTS-TSR protocol)

NTS Record Name	Mode	Use	Reference	Exemplary body contents
AEAD Algorithm Negotiation	TiM	mandatory	Section 4.2.1	AEAD_AES_SIV_CMAC_512
Validity Period	TiM	mandatory	Section 4.2.16	{2460s||400s||3s}
Ticket Key ID	TiM	mandatory	Section 4.2.15	278
Ticket Key	TiM	mandatory	Section 4.2.14	{binary data}

Table 9: Record structure of the container records in the PTP Registration Response message

If the registration request lies inside the update period a Next Parameters container record is additionally appended giving all the security data needed in the upcoming validity period. Its structure follows the same composition as the Current Parameters container record. (If the respective grantor has not registered yet for the upcoming SA period or has revoked its service, no Next Parameters container will be sent.) In case of an error, both parameters container records are removed and a single error record is inserted (see Table 8). The messages always end with an End of Message record.

3.5. PTP Registration Revoke Message

The PTP Registration Revoke message (NTS-TSR protocol) from the grantor starts with the NTS Message Type record containing the message type as well as the message version number, here always 1.0, see Table 10. (As the message belongs to the NTS-TSR protocol, no NTS Next Protocol Negotiation record is necessary.)

PTP Registration Revoke (NTS-TSR protocol)

NTS Record Name	Mode	Use	Reference	Exemplary body contents
NTS Message Type	TiM	mandatory	Section 4.2.8	PTP Registration Revoke||v1.0
Source PortIdentity	TiM	mandatory	Section 4.2.11	{binary data}
End of Message	TiM	mandatory	Section 4.2.4	(no record body)

Table 10: Record structure of the PTP Registration Revoke message

The second record contains the Source PortIdentity which identifies the grantor wishing to discontinue its unicast support. Together with the Subject Key Identifier (SKI) of the client certificate (verified during the TLS connection establishment) and the Source PortIdentity given in the NTS message, the NTS-KE server can uniquely identify the grantor if the PTP device communicates with the NTS-KE server via a management port running multiple grantors. The message always ends with an End of Message record.

4. NTS Records for PTP

The above sections have described the principle communication sequences and structure for the new NTS messages. All messages follow the "NTS Key Establishment Process" stated in the first part (up to the description of Figure 3) of Section 4 of IETF RFC 8915 [RFC8915] with the exception that registration requests use the ALPN "ntstsr/1" instead of the ALPN "ntske/1/1" and do not include a Next Protocol Negotiation record:

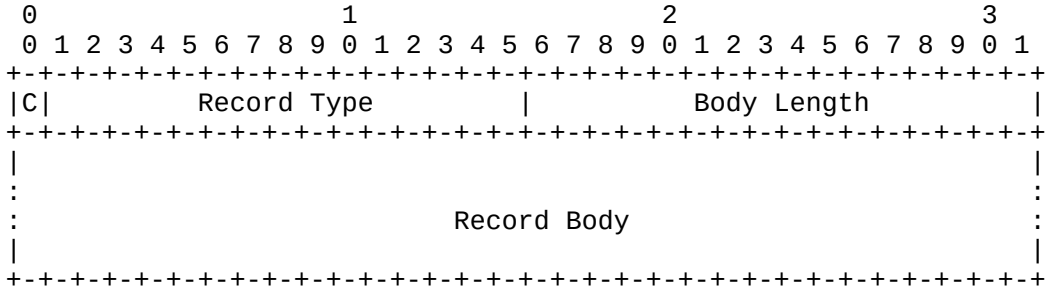


Figure 7: NTS-KE record format

All NTS messages consist of a sequence of records, each containing a Critical Bit C, the Record Type, the Body Length and the Record Body, see Figure 7. The Critical Bit determines the disposition of unrecognized Record Types. Implementations which receive a record with an unrecognized Record Type MUST ignore the record if the Critical Bit is 0 and MUST treat it as an error if the Critical Bit is 1. The Record Type number is a 15-bit integer. The semantics of record types 07 are specified in [RFC8915]. Additional type numbers as defined in this document SHALL be tracked through the IANA Network Time Security Key Establishment Record Types registry. The Body Length specifies the length of the Record Body field, in octets, as a 16-bit integer. Record bodies MAY have any representable length and need not be aligned to a word boundary. The syntax and semantics of the field Record Body SHALL be determined by the Record Type. All fields of an NTS-KE record are in network byte order.

More details on record structure as well as the specific records used here are given in this section and respective subsections. Container records (short: container) themselves comprise a set of records in the record body that serve a specific purpose, e.g., the Current Parameters container record.

The records contained in a message may follow in arbitrary sequence (though nothing speaks against using the sequence given in the record descriptions), only the End of Message record MUST be the last one in the sequence indicating the end of the current message. Container records do not include an End of Message record.

4.1. Overview of the NTS Records

In Table 11 below, this section lists all NTS records from which the messages are constructed. In addition to the NTS records already defined for NTP in IETF RFC 8915 (see [RFC8915], Section 7.6.), additional records are required and their type numbers have to be defined by the IANA. The detailed structure and respective content of the records is given in Section 4.2. In addition to the record number the sub-protocol it is used with, Table 11 indicates where it can be found in [RFC8915] or in this document.

NTS Record Types	Description	Record Used in Protocol	Reference
0	End of Message	NTS-KE / NTS-TSR	[RFC8915] in Section 4.1.1. This document in Section 4.2.4
1	NTS Next Protocol Negotiation	NTS-KE	[RFC8915] in Section 4.1.2. This document in Section 4.2.7
2	Error	NTS-KE / NTS-TSR	[RFC8915] in Section 4.1.3. This document in Section 4.2.5
3	Warning	NTS-KE	[RFC8915] in Section 4.1.4. Not used for PTP
4	AEAD Algorithm Negotiation	NTS-TSR	[RFC8915] in Section 4.1.5. This document in Section 4.2.1
5	New Cookie for NTPv4	NTS-KE	[RFC8915] in Section 4.1.6. Not used for PTP
6	NTPv4 Server Negotiation	NTS-KE	[RFC8915] in Section 4.1.7. Not used for PTP
7	NTPv4 Port Negotiation	NTS-KE	[RFC8915] in Section 4.1.8. Not used for PTP

8 - TBD	(Reserved for NTP)		
TBD01	Association Mode	NTS-KE	Section 4.2.2
TBD02	Current Parameters	NTS-KE / NTS-TSR	Section 4.2.3
TBD03	Next Parameters	NTS-KE / NTS-TSR	Section 4.2.6
TBD04	NTS Message Type	NTS-TSR	Section 4.2.8
TBD05	PTP Time Server	NTS-KE / NTS-TSR	Section 4.2.9
TBD06	Security Association	NTS-KE	Section 4.2.10
TBD07	Source PortIdentity	NTS-KE / NTS-TSR	Section 4.2.11
TBD08	Supported MAC Algorithms	NTS-KE / NTS-TSR	Section 4.2.12
TBD09	Ticket	NTS-TSR	Section 4.2.13
TBD10	Ticket Key	NTS-TSR	Section 4.2.14
TBD11	Ticket Key ID	NTS-TSR	Section 4.2.15
TBD12	Validity Period	NTS-KE / NTS-TSR	Section 4.2.16
TBD13 - 16383	Unassigned		
16384 - 32767	Reserved for Private or Experimental Use		[RFC8915]

Table 11: NTS Key Establishment and Time Server Registration record types registry

4.2. Detailed Description of the NTS Records

The following subsections describe the specific NTS records used to construct the NTS messages for the PTP key management system in detail. They appear in alphabetic sequence of their individual names. See Section 3 for the application of the records in the respective messages.

Note: For easier editing of the content, most of the descriptions in the following subsections are written as bullet points.

4.2.1. AEAD Algorithm Negotiation

This record is used in the NTS-TSR protocol.

This record is required in Ticket-based mode (TiM) and enables the negotiation of the AEAD algorithm needed to encrypt and decrypt the encrypted payload of the ticket (see Section 6.1). The negotiation takes place between the PTP grantor and the NTS-KE server by using the NTS registration messages. The structure and properties follow the record defined in IETF RFC 8915 [RFC8915], Section 4.1.5.

Content and conditions:

* The record has a Record Type number of 4 and the Critical Bit MAY be set.
* The Record Body contains a sequence of 16-bit unsigned integers in network byte order:
  Supported AEAD Algorithms = {AEAD 1 || AEAD 2 || …}

* Each integer represents a numeric identifier of an AEAD algorithm registered by the IANA in [IANA_AEAD].
* Duplicate identifiers SHOULD NOT be included.
* Grantor and NTS-KE server MUST support at least the AEAD_AES_SIV_CMAC_256 algorithm.
* A list of recommended AEAD algorithms is shown in the following Table 12.
* Other AEAD algorithms MAY also be used.

Numeric ID	AEAD Algorithm	Use	Key Length (Octets)	Reference
15	AEAD_AES_SIV_CMAC_256	mand.	32	[RFC5297]
16	AEAD_AES_SIV_CMAC_384	opt.	48	[RFC5297]
17	AEAD_AES_SIV_CMAC_512	opt.	64	[RFC5297]

Table 12: Recommended AEAD algorithms

PTP Registration Request message:

* In a PTP Registration Request message, this record MUST be contained exactly once.
* In that message at least one algorithm MUST be included, e.g., the AEAD_AES_SIV_CMAC_256 algorithm.
* If multiple AEAD algorithms are supported, the grantor SHOULD put the algorithm identifiers in descending priority in the Record Body.
* Strong algorithms with higher bit lengths SHOULD have higher priority.
* The NTS-KE server SHOULD choose the highest priority AEAD algorithm from the request message that grantor and NTS-KE server support.
* The NTS-KE server MAY ignore the priority and choose a different algorithm that grantor and NTS-KE server support.

PTP Registration Response message:

* In a PTP Registration Response message, this record MUST contain exactly one AEAD algorithm.
* This record MUST be contained exactly once in the Current Parameters container record and exactly once in the Next Parameters container record (if available).
* The selected AEAD algorithm in the Current Parameters container record MAY differ from the selected AEAD algorithm in the Next Parameters container record.

4.2.2. Association Mode

This record is used in the NTS-KE protocol.

This record enables the NTS-KE server to distinguish between the operation modes (GrM/TiM) of a PTP Key Request message. A GrM request carries a group number, while a TiM request contains an identification attribute of the desired grantor (e.g., IP address or PortIdentity).

Content and conditions:

* In a PTP Key Request message, this record MUST be contained exactly once.
* The record has a Record Type number of TBD01 and the Critical Bit MAY be set.
* The Record Body SHALL consist of one association tuple concatenating an Association Type and an Association Value:

Field	Octets	Offset
Association Type	2	0
Association Value	A	2

Table 13: Association Mode record

* The Association Type is a 16-bit unsigned integer.
* The length of Association Value depends on the value of Association Type.
* All data in the fields are stored in network byte order.
* The type numbers for Association Type, together with the length and content of Association Value, are shown in the table below, with further details given subsequently.

Description	Assoc. Type Number	Association Mode	Association Value Content	Assoc. Value Octets
Group	0	GrM	Group Number	4
IPv4	1	TiM	IPv4 address of the target port	4
IPv6	2	TiM	IPv6 address of the target port	16
802.3	3	TiM	MAC address of the target port	6
PortIdentity	4	TiM	PortIdentity of the target PTP entity	10
	5 - 32767	Unassigned		
	32768 - 65565	Reserved for Private or Experimental Use		

Table 14: Association Types

Group:

* This Association Type allows a PTP instance to join a GrM group.
* The Association Value is a 32-bit unsigned integer in network byte order.
* The group number can be freely specified by the administrator.

IPv4:

* This Association Type allows a requester to establish a PTP unicast connection to the desired grantor.
* The Association Value contains the IPv4 address of the target PTP entity.

* The total length is 4 octets.

IPv6:

* This Association Type allows a requester to establish a PTP unicast connection to the desired grantor.
* The Association Value contains the IPv6 address of the target PTP entity.
* The total length is 16 octets.

802.3:

* This Association Type allows a requester to establish a PTP unicast connection to the desired grantor.
* The Association Value contains the MAC address of the Ethernet port of the target PTP entity.
* The total length is 6 octets.
* This method supports the IEEE 802.3 mode in PTP, where no UDP/IP stack is used.

PortIdentity:

* This Association Type allows a requester to establish a PTP unicast connection to the desired grantor.
* The Association Value contains the PortIdentity of the target PTP entity.
* The total length is 10 octets.
* The PortIdentity consists of the attributes clockIdentity (8 octet array) and portNumber (16-bit unsigned integer), see [IEEE1588-2019], Sections 5.3.5 and 7.5: *PortIdentity = {clockIdentity || portNumber}*

4.2.3. Current Parameters

This record is used in the NTS-KE and NTS-TSR protocol.

This record is a simple container that can carry an arbitrary number of NTS records. It holds all security parameters relevant for the current validity period. The content as well as further conditions are defined by the respective NTS messages. The order of the included records is arbitrary and the parsing rules are so far identical with the NTS message. One exception: An End of Message record SHOULD NOT be present and MUST be ignored. When the parser reaches the end of the Record Body quantified by the Body Length, all embedded records have been processed.

Content and conditions:

* The record has a Record Type number of TBD02 and the Critical Bit MAY be set.
* The Record Body is defined as a set of records and MAY contain the records shown in Table 15 respectively in Table 16.
* The NTS-KE server MUST NOT add any other records and the client MUST ignore other records.

PTP Key Response message:

NTS Record Name	Communication Type	Use	Reference
Security Associations	GrM / TiM	mandatory	Section 4.2.10
Validity Period	GrM / TiM	mandatory	Section 4.2.16
PTP Time Server	TiM	mandatory	Section 4.2.9
Ticket	TiM	mandatory	Section 4.2.13

Table 15: Current Parameters container for PTP Key Response message

* In a PTP Key Response message, this record MUST be contained exactly once.
* The records Security Association and Validity Period MUST be contained exactly once.
* Additionally, the records PTP Time Server and Ticket MUST be included exactly once if the client had sent a PTP Key Request message for TiM and MUST NOT be included if the client had sent a PTP Key Request message for a multicast group in GrM.

PTP Registration Response message:

* In a PTP Registration Response message, the Current Parameters container record MUST be contained exactly once.
* The Record Body MUST contain the following records exactly once:

NTS Record Name	Use	Reference
AEAD Algorithm Negotiation	mandatory	Section 4.2.1
Validity Period	mandatory	Section 4.2.16
Ticket Key ID	mandatory	Section 4.2.15
Ticket Key	mandatory	Section 4.2.14

Table 16: Current Parameters container for PTP Registration Response Message

4.2.4. End of Message

This record is used in the NTS-KE and NTS-TSR protocol.

The End of Message record is defined in IETF RFC 8915 [RFC8915], Section 4 and 4.1.1.

Content and conditions:

* The record has a Record Type number of 0 and a zero-length body.
* The Critical Bit MUST be set.
* This record MUST occur exactly once as the final record of every NTS message.
* This record SHOULD NOT be included in the container records and MUST be ignored if present.

4.2.5. Error

This record is used in the NTS-KE and NTS-TSR protocol.

The Error record is defined in IETF RFC 8915 [RFC8915], Section 4.1.3. In addition to the Error codes 0 to 2 specified there the following Error codes were added:

Error Code	Description
0	Unrecognized Critical Record
1	Bad Request
2	Internal Server Error
TBD-E01	Not Authenticated
TBD-E02	Not Authorized
TBD-E03	Algorithms Not Supported
TBD-E04	Grantor Not Registered
TBD-E - 32767	Unassigned
32768 - 65535	Reserved for Private or Experimental Use

Table 17: Error Codes

Content and conditions:

* The record has a Record Type number of 2 and body length of two octets consisting of an unsigned 16-bit integer in network byte order, denoting an error code.
* The Critical Bit MUST be set.
* The Error code TBD-E01 "Not Authenticated" is sent by the NTS-KE server if the requesting client is not authenticated by its certificate.
* The Error code TBD-E02 "Not Authorized" is sent by the NTS-KE server if the requesting client is not authorized to join the desired multicast group or if a grantor is prohibited to register with the NTS-KE server.
* The Error code TBD-E03 "Algorithms Not Supported" is sent by the NTS-KE server if the security algorithms presented by the requesting client or the requesting grantor are not supported.
* The Error code TBD-E04 "Grantor Not Registered" is sent by the NTS-KE server when the requester asks for the Security Association for a grantor that is not registered with the NTS-KE server.
* The Error record MUST NOT be included in any of the messages PTP Key Request, PTP Registration Request and PTP Registration Revoke.

4.2.6. Next Parameters

This record is used in the NTS-KE and NTS-TSR protocol.

This record is a simple container that can carry an arbitrary number of NTS records. It holds all security parameters relevant for the upcoming validity period. The content as well as further conditions are defined by the respective NTS messages. The order of the included records is arbitrary and the parsing rules are so far identical with the NTS message. One exception: An End of Message record SHOULD NOT be present and MUST be ignored. When the parser reaches the end of the Record Body quantified by the Body Length, all embedded records have been processed.

Content and conditions:

* The record has a Record Type number of TBD03 and the Critical Bit MAY be set.
* The Record Body is defined as a set of records.
* The structure of the record body and all conditions MUST be identical to the rules described in Section 4.2.3 of this document.
* Outside the update period, this record MUST NOT be included.
* In both, the PTP Key Response and PTP Registration Response message, this record SHOULD be contained exactly once during the update period.
* In GrM, this record MAY also be missing if the requesting client is to be explicitly excluded from a multicast group after the security parameter rotation process by the NTS-KE server.
* More details are described in Section 2.5.1.

4.2.7. NTS Next Protocol Negotiation

This record is used in the NTS-KE protocol.

The Next Protocol Negotiation record is defined in IETF RFC 8915 [RFC8915], Section 4.1.2:

> _"The Protocol IDs listed in the client's NTS Next Protocol Negotiation record denote those protocols that the client wishes to speak using the key material established through this NTS-KE server session. Protocol IDs listed in the NTS-KE server's response MUST comprise a subset of those listed in the request and denote those protocols that the NTP server is willing and able to speak using the key material established through this NTS-KE server session. The client MAY proceed with one or more of them. The request MUST list at least one protocol, but the response MAY be empty."_

Content and conditions:

* The record has a Record Type number of 1 and the Critical Bit MUST be set.
* The Record Body consists of a sequence of 16-bit unsigned integers in network byte order.
  Record body = {Protocol ID 1 || Protocol ID 2 || ⋯}
* Each integer represents a Protocol ID from the IANA "Network Time Security Next Protocols" registry (tbd.) as shown in the table below.
* For NTS request messages for PTPv2.1 (NTS-KE protocol merely), only the Protocol ID for PTPv2.1 SHOULD be included.
* This prevents the mixing of records for different time protocols.

Protocol ID	Protocol Name	Reference
0	Network Time Protocol version 4 (NTPv4)	[RFC8915], Section 7.7
TBD-P01	Precision Time Protocol version 2.1 (PTPv2.1)	This document
TBD-P - 32767	Unassigned	
32768 - 65535	Reserved for Private or Experimental Use	

Table 18: NTS next protocol IDs

Possible NTP/PTP conflict:

* The support of multiple protocols in this record may lead to the problem that records in NTS messages can no longer be assigned to a specific time protocol.
* For example, an NTS request could include records for both NTP and PTP.
* However, NTS for NTP does not use NTS message types and the End of Message record is also not defined for the case of multiple NTS requests in one TLS message.
* This leads to the mixing of the records in the NTS messages.
* A countermeasure is the use of only a single time protocol in the NTS Next Protocol Negotiation record that explicitly assigns the NTS message to a specific time protocol.
* When using NTS-secured NTP and NTS-secured PTP, two separate NTS requests i.e., two separate TLS sessions MUST be made.

4.2.8. NTS Message Type

This record is used in the NTS-TSR protocol.

This record enables the distinction between different NTS message types and message versions for the NTS-TSR protocol. It MUST be included exactly once in each NTS message in the NTS-TSR protocol.

Content and conditions:

* The record has a Record Type number of TBD04 and the Critical Bit MUST be set.
* The Record Body MUST consist of three data fields:

Field		Octets	Offset
Message Type		2	0
Message Version	Major version	1	2
Message Version (cont.)	Minor version	1	3

Table 19: Content of the NTS Message Type record

* The Message Type field is a 16-bit unsigned integer in network byte order, denoting the type of the current NTS message.
* The following values (tbd. by IANA) are defined for the Message Type:

Message Type (value)	NTS Message (NTS-TSR protocol)
TBD-T00	PTP Registration Request
TBD-T01	PTP Registration Response
TBD-T02	PTP Registration Revoke
TBD-T - 32767	Unassigned
32768 - 65535	Reserved for Private or Experimental Use

Table 20: NTS Message Types for the NTS-TSR protocol

* The Message Version consists of a tuple of two 8-bit unsigned integers in network byte order:
  NTS Message Version = {major version || minor version}
* The representable version is therefore in the range 0.0 to 255.255 (e.g., v1.4 = 0104h).
* All NTS messages for PTPv2.1 described in this document are in version number 1.0.
* Thus the Message Version MUST match 0100h.

4.2.9. PTP Time Server

This record is used in the NTS-KE and NTS-TSR protocol.

The PTP Time Server record is used exclusively in TiM (PTP unicast connection) and signals to the client (PTP requester) for which grantor the security parameters are valid. This record is used both, in the NTS-KE protocol in the PTP Key Response, and in NTS-TSR protocol in the PTP Registration Request message.

Content and conditions:

* The record has a Record Type number of TBD05 and the Critical Bit MAY be set.
* The record body consists of a set of association values constructed of the data tuple which forms the record body of the Association Mode record described in Section 4.2.2 (Association Mode).
* The structure of the record body and all conditions MUST be identical to the rules described in Section 4.2.2 (Association Mode) of this document, with the following exceptions:

* In a PTP Key Response message, this record MUST be contained exactly once within a container record (e.g., Current Parameters container record).
* The PTP Time Server record contains a list of all available addresses of the grantor assigned by the NTS-KE server.
* This MUST be one of the following association types: IPv4, IPv6, MAC address or the PortIdentity of the grantor.
* As the record is only used in TiM, it MUST NOT be of the association type Group.
* The list SHALL contain only one of each association type.
* This allows the client to change the PTP transport mode (e.g., from IPv4 to IEEE 802.3) without performing a new NTS request.
* The list in the PTP Time Server record MUST contain at least the PortIdentity.

* In a PTP Registration Request message, this record MUST be included exactly once.

* The grantor MUST enter all network addresses that are supported for a unicast connection.
* This can be an IPv4, IPv6, MAC address, as well as the PortIdentity.
* The list in the PTP Time Server record MUST NOT contain the Association Type number 0 (multicast group) and MUST contain at least the PortIdentity.
* The PortIdentity is especially needed by the NTS-KE server to identify the correct PTP instance (the grantor) in case of a PTP Registration Revoke message and enables a requester to more easily identify a grantor, e.g., in the SAD.

4.2.10. Security Association

This record is used in the NTS-KE protocol.

This record contains the information "how" specific PTP message types must be secured. It comprises all dynamic (negotiable) values necessary to construct the AUTHENTICATION TLV (IEEE Std 1588-2019, Section 16.14.3). Static values and flags, such as the secParamIndicator, are described in more detail in Section 5.1.

Content and conditions:

* The record has a Record Type number of TBD06 and the Critical Bit MAY be set.
* The Record Body is a sequence of various parameters in network byte order and MUST be formatted according to the following table:

Field	Octets	Offset
Integrity Algorithm Type	2	0
Key ID	4	2
Key Length	2	6
Key	K	8

Table 21: Security Association record

* In a PTP Key Response message, the Security Association record MUST be included exactly once in the Current Parameters container record.

* In a PTP Key Response message during the update period, the Security Association record MUST be included exactly once in the Current Parameters container record.
* The Next Parameters container record MUST be present only during the update period.
* In TiM, the Security Association record MUST be included exactly once in the encrypted Ticket as well.

Integrity Algorithm Type

* This value is a 16-bit unsigned integer in network byte order.
* The possible values are equivalent to the MAC algorithm types from the table in Section 4.2.12.
* The value used depends on the negotiated or predefined MAC algorithm.

Key ID

* The key ID is a 32-bit unsigned integer in network byte order.
* The field length is oriented towards the structure of the AUTHENTICATION TLV.
* The generation and management of the key ID is controlled by the NTS-KE server.
* The NTS-KE server MUST ensure that every key ID is unique.
  - Previous key IDs SHOULD NOT be reused for a certain number of rotation periods or a defined period of time (see Section 6.2).

Key Length

* This value is a 16-bit unsigned integer in network byte order, denoting the length of the key in octets.

Key

* The value is a sequence of octets with a length of Key Length.
* This symmetric key is needed together with the MAC algorithm to calculate the ICV.
* It can be both a group key (GrM) or a unicast key (TiM).

4.2.11. Source PortIdentity

This record is used in the NTS-KE and NTS-TSR protocol.

This record contains a PTP PortIdentity and serves as an identifier. In a PTP Key Request message, it enables the unique assignment of the NTS request to the PTP instance of the sender, since the request may have been sent to the NTS-KE server via a management port.

The PortIdentity is embedded in the PTP Key Response message within the ticket to bind it to the PTP requester. Grantors can verify that the ticket comes from the correct sender when it is received and before it is decrypted, to prevent possible crypto-performance attacks. In a PTP registration Revoke message this record enables the assignment of the grantor at the NTS-KE server to revoke an existing registration. This is necessary because requesting PTP devices may have multiple independent PTP ports and possibly multiple registrations with the KE.

Content and conditions:

* The record has a Record Type number of TBD07 and the Critical Bit MAY be set.
* The record contains the PTP PortIdentity of the sender in network byte order, with a total length of 10 octets.
* In a PTP Key Request message, this record MUST be included exactly once if the client intends a unicast request in TiM and MUST NOT be included if the client intends to join a group in GrM.
* In the messages PTP Key Response, PTP Registration Response and PTP Registration Revoke message, this record MUST be included and exactly once.
* The PortIdentity consists of the attributes clockIdentity and portNumber:
  PortIdentity = {clockIdentity || portNumber}
* The clockIdentity is an 8-octet array and the portNumber is a 16-bit unsigned integer (source: [IEEE1588-2019], Sections 5.3.5 and 7.5)

4.2.12. Supported MAC Algorithms

This record is used in the NTS-KE and NTS-TSR protocol.

This record allows free negotiation of the MAC algorithm needed to generate the ICV. Since multicast groups are restricted to a shared algorithm, this record is used mandatorily in a PTP Registration Request message and MAY be used (optionally) in a PTP Key Request message.

Content and conditions:

* The record has a Record Type number of TBD08 and the Critical Bit MAY be set.
* The Record Body contains a sequence of 16-bit unsigned integers in network byte order.
  Supported MAC Algorithms = {MAC 1 || MAC 2 || …}
* Each integer represents a MAC Algorithm Type defined in the table below.

* Duplicate identifiers SHOULD NOT be included.
* Each PTP node MUST support at least the HMAC-SHA256-128 algorithm.

MAC Algorithm Types	MAC Algorithm	ICV Length (octets)	Reference
TBD-A00	HMAC-SHA256-128	16	[fiPS-PUB-198-1], [IEEE1588-2019]
TBD-A01	HMAC-SHA256	32	[fiPS-PUB-198-1]
TBD-A02	AES-CMAC	16	[RFC4493]
TBD-A03	AES-GMAC-128	16	[RFC4543]
TBD-A04	AES-GMAC-192	24	[RFC4543]
TBD-A05	AES-GMAC-256	32	[RFC4543]
TBD-A - 32767	Unassigned		
32768 - 65535	Reserved for Private or Experimental Use		

Table 22: MAC Algorithms

No other MAC algorithms than the algorithms in the Table above MUST be used.

In Group-based mode (GrM):

* This record is not necessary, since all PTP nodes in a multicast group MUST support the same MAC algorithm.
* Therefore, this record SHOULD NOT be included in a PTP Key Request message and the NTS-KE server MUST ignore this record if the Association Type in the Association Mode record is 0 (= GrM group).
* Unless this is specified otherwise by a PTP profile, the HMAC-SHA256-128 algorithm SHALL be used by default.

In Ticket-based mode (TiM):

* In a PTP Key Request message, this record MAY be contained if the requester wants a unicast connection (TiM) to a specific grantor.

* The requester MUST NOT send more than one record of this type.
* If this record is present, at least one MAC algorithm MUST be included.
* If multiple MAC algorithms are supported, the requester SHOULD put the desired algorithm identifiers in descending priority in the record body.
* Strong algorithms with higher bit lengths SHOULD have higher priority.
* In a PTP Registration Request message, this record MUST be present and the grantor MUST include all supported MAC algorithms in any order.
* The NTS-KE server selects the algorithm after receiving a PTP Key Request message in unicast mode.
* The NTS-KE server SHOULD choose the highest priority MAC algorithm from the request message that grantor and requester support.
* The NTS-KE server MAY ignore the priority and choose a different algorithm that grantor and requester support.
* If the MAC Algorithm Negotiation record is not within the PTP Key Request message, the NTS-KE server MUST choose the default algorithm HMAC-SHA256-128.

Initialization Vector (IV)

* If GMAC is to be supported as a MAC algorithm, then an Initialization Vector (IV) must be constructed according to IETF RFC 4543 [RFC4543], Section 3.1.
* Therefore, the IV MUST be eight octets long and MUST NOT be repeated for a specific key.
* This can be achieved, for example, by using a counter.

4.2.13. Ticket

This record is used in the NTS-KE protocol.

This record contains the parameters of the negotiated AEAD algorithm chosen between the grantor and the NTS-KE server, as well as an encrypted security association. The record contains all the necessary security parameters that the grantor needs for a secured PTP unicast connection to the requester. The ticket is encrypted by the NTS-KE server with the symmetric ticket key which is also known to the grantor. The requester is not able to decrypt the encrypted security association within the ticket.

Content and conditions:

* The record has a Record Type number of TBD09 and the Critical Bit MAY be set.

* The Record Body consists of several data fields and MUST be formatted as follows.

Field	Octets	Offset
Ticket Key ID	4	0
Source PortIdentity	10	4
Nonce Length	2	14
Nonce	N	16
Encrypted SA Length	2	N+16
Encrypted Security Association	E	N+18

Table 23: Structure of a Ticket record

* In a PTP Key Response message, this record MUST be included exactly once each in the Current Parameters container record and a potential Next Parameters container record if the requesting client wants a unicast communication to a specific grantor in TiM.

Ticket Key ID

* This is a 32-bit unsigned integer in network byte order, denoting the key ID of the ticket key.
* The value is set by the NTS-KE server and is valid for the respective validity period.
* See also Section 4.2.15 for more details.

Source PortIdentity

* This 10-octet long field contains the identical Source PortIdentity of the PTP client from the PTP Key Request message.

Nonce Length

* This is a 16-bit unsigned integer in network byte order, denoting the length of the Nonce field in octets.

Nonce

* This field contains the Nonce needed for the AEAD operation.

* The length and conditions attached to the Nonce depend on the AEAD algorithm used.
* More details and conditions are described in Section 6.1.

Encrypted SA Length

* This is a 16-bit unsigned integer in network byte order, denoting the length of the Encrypted Security Association field in octets.

Encrypted Security Association

* This field contains the output of the AEAD operation ("Ciphertext") after the encryption process of the respective Record Body of the respective Security Association record.
* The plaintext of this field is described in Section 4.2.10.
* More details about the AEAD process and the required input data are described in Section 6.1.

4.2.14. Ticket Key

This record is used in the NTS-TSR protocol.

This record contains the ticket key, which together with an AEAD algorithm is used to encrypt and decrypt the ticket payload (content of the Encrypted Security Association field in the Ticket record).

Content and conditions:

* The record has a Record Type number of TBD10 and the Critical Bit MAY be set.
* The Record Body consists of a sequence of octets holding the symmetric key for the AEAD function.
* The generation and length of the key MUST meet the requirements of the associated AEAD algorithm.
* In a PTP Registration Response message, this record MUST be included exactly once each in the Current Parameters container record and the Next Parameters container record.

4.2.15. Ticket Key ID

Used in NTS-TSR protocol.

The Ticket Key ID record is a unique identifier that allows a grantor to identify the associated ticket key. The NTS-KE server is responsible for generating this key ID, which is also unique to the PTP network and incremented at each rotation period. The associated key is known only to the NTS-KE server and grantor, and is generated and exchanged during the registration phase of the grantor. All tickets generated by the NTS-KE server for the corresponding grantor in this validity period using the same ticket key ID.

Content and conditions:

* The record has a Record Type number of TBD11 and the Critical Bit MAY be set.
* The Record Body consists of a 32-bit unsigned integer in network byte order.
* The generation and management of the ticket key ID is controlled by the NTS-KE server.
* The NTS-KE server must ensure that every ticket key has a unique number.
  - The value is implementation dependent and MAY be an enumeration.
  - Previous IDs SHOULD NOT be reused for a certain number of rotation periods or a defined period of time.
* In a PTP Registration Response message, this record MUST be included exactly once in the Current Parameters container record and once in the Next Parameters container record.
* The Ticket record MUST be present in TiM and MUST NOT be present in GrM.

4.2.16. Validity Period

Used in NTS-KE and NTS-TSR protocol.

This record contains the validity information of the respective security parameters (see also Section 2.5.1).

Content and conditions:

* In a PTP Key Response as well as in the PTP Registration Response message, this record MUST be included exactly once each in the Current Parameters container record and a potential Next Parameters container record.
* The record has a Record Type number of TBD12 and the Critical Bit MAY be set.
* The Record Body MUST consist of three data fields:

Field	Octets	Offset
Lifetime	4	0
Update Period	4	4
Grace Period	4	8

Table 24: Structure of a Validity Period record

Lifetime

* The Lifetime is a 32-bit unsigned integer in network byte order.
* If this record is within a Current Parameters container record, it shows the remaining lifetime of the security parameters for the current validity period in seconds.
* If this record is within a Next Parameters container record, it shows the total lifetime of the security parameters for the next validity period in seconds.
* The counting down of the Next Parameters lifetime starts as soon as the remaining lifetime of the Current Parameters reaches 0s.
* The key lifetimes SHOULD NOT exceed 24 hours, a lifetime of 1 hour is recommended.
* The maximum value is set by the NTS-KE administrator or the PTP profile.
* In conjunction with a PTP unicast establishment in TiM, the lifetime of the unicast key (within the Security Association record), the ticket key and registration lifetime of a grantor with the NTS-KE server MUST be identical.

Update Period

* The Update Period is a 32-bit unsigned integer in network byte order.
* It specifies how many seconds before the lifetime expires the update period starts.
* Unlike the lifetime, this is a fixed value that is not counted down.
* The Update Period value MUST NOT be greater than the full Lifetime.
* Recommended is an Update Period of 120s-300s if the full Lifetime is 900s or longer.
* If the value of the Update Period in the Current Parameters container record is greater than the Lifetime, then the key update process has started.

* The presence or absence of the Next Parameters container record is specified in Section 4.2.6.

Grace Period

* The Grace Period is a 32-bit unsigned integer in network byte order.
* It defines how many seconds expired security parameters SHOULD still be accepted.
* This allows the verification of incoming PTP messages that were still on the network and secured with the old parameters.
* The Grace Period value MUST NOT be greater than the Update Period.
* Recommended is a Grace Period of 0 to 5 seconds.

Notes:

* Requests during the currently running lifetime will receive respectively adapted count values.
* The lifetime is a counter that is decremented and marks the expiration of defined parameters when the value reaches zero.
* The realization is implementation-dependent and can be done for example by a secondly decrementing.
* It MUST be ensured that jumps (e.g., by adjustment of the local clock) are avoided.
* The use of a monotonic clock is suitable for this.
* Furthermore, it is to be considered which consequences the drifting of the local clock can cause.
* With sufficiently small values of the lifetime (<12 hours), this factor should be negligible.

5. Additional Security Measures

As mentioned in Section 1.1 the PTPv2.1 standard [IEEE1588-2019] does not supply provisions against certain attacks, such as replay, start-up replay and address spoofing. In addition to providing an automatic key management solution, this document also addresses measures to fend off such attacks.

It should be emphasized that some other attack vectors, such as those based on message delay, cannot be countered by cryptographic means. Therefore, other measures such as redundancy and monitoring should be used, which are outside the scope of this document.

5.1. AUTHENTICATION TLV Parameters

The AUTHENTICATION TLV is the heart of the integrated security mechanism (prong A) for PTP. It provides data for the processing of the security means. The structure is shown in Table 25 below (see also figure 49 of [IEEE1588-2019]).

TLV Field	Use	Description
tlvType	mandatory	TLV Type
lengthField	mandatory	TLV Length Information
SPP	mandatory	Security Parameter Pointer
secParamIndicator	mandatory	Security Parameter Indicator
keyID	mandatory	Key Identifier or Current Key Disclosure Interval, depending on verification scheme
disclosedKey	optional	Disclosed key from previous interval
sequenceNo	optional	Sequence number
RES	optional	Reserved
ICV	mandatory	ICV based on algorithm OID

Table 25: Structure of the AUTHENTICATION TLV

The tlvType is AUTHENTICATION (0x8009) and lengthField holds the length of the total TLV in octets. When using the AUTHENTICATION TLV with NTS4PTP key management, the keyID is provided by the NTS-KE server in the PTP Key Response message. Due to the one-octet size limitation, the SPP is unused in NTS4PTP and remains 0, while the SA is identified by the keyID.

The secParamIndicator field and its flags indicate which of the following optional fields are present.

The disclosedKey field is only to be used with delayed security processing approach like a TESLA-based solution [RFC4082]. Therefore, it is always missing in NTS4PTP.

NTS4PTP uses the optional fields sequenceNo and RES to transport data for the additional security measures mentioned. Thus, the secParamIndicator MUST have a value of 0x03.

The field keyID identifies the key currently to be used.

The ICV field contains the integrity check value of the particular PTP message calculated using the integrity algorithm defined by the key management. The length depends on the MAC algorithm used.

The following Section 5.1.1 and Section 5.1.2 describe the usage of the sequenceNo field and the RES field in a way that has been proposed to IEEE1588 and is currently under discussion there.

5.1.1. The sequenceNo Field

In order to allow the flexible use of different security solutions, the field sequenceNo is structured as shown in Table 26.

sequenceNo Field	Octets	Offset
keyMgmtType	1	0
seqNoLength	1	1
SeqID	S	2

Table 26: Structure of the sequenceNo field in the AUTHENTICATION TLV

The keyMgmtType field specifies the key management solution used; for this document it is the number 0x01, identifying NTS4PTP. The seqNoLength field contains the number S of octets allocated to the following SeqID field. S MUST be even. The field SeqID holds the respective sequence number used for the key management solution. This number should not be confused with the sequenceID in the PTP header.

5.1.2. The RES Field dataBlocks Field

The former RES field, now called dataBlocks (see Table 27), is used to transport additional data required by security solutions. It can contain more than one data block, each with its own type, length and data field. A dataBlockCnt field indicates the number of dataBlocks present. As the octets in the dataBlocks (RES) field MUST be even, possibly a padding octet can follow.

dataBlocks Field	Octets	Offset	Remarks
dataBlockCnt	1	0	= N, number of data blocks
block1Type	1	1	type of data block 1
payLoadLength	2	2	length of its data = PD1
payLoadData	PD1	4	block 1 data
block2Type	1	4+PD1	if BC > 1, …
payLoadLength	2	5+PD1	if BC > 1, …
payLoadData	PD2	7+PD1	if BC > 1, …
...	...	...	...
blockNType	1		if BC = N …
payLoadLength	2		if BC = N …, …
payLoadData	PDN		if BC = N …
padding	0/1		

Table 27: Structure of the dataBlocks field (former RES field) in the AUTHENTICATION TLV

5.1.3. The NTS4PTP Data in the dataBlocks Field

In addition to the sequenceNo field, NTS4PTP also uses a data block in the dataBlocks field (formerly RES). As described in Section 5.1.2, keyMgmtType indicates the security solution, in this case NTS4PTP with the number 0x01. The payLoadLength field specifies the length of the NTS4PTP data (payLoadData) in this data block. The payLoadData is structured as shown in Table 28. Some of the NTS4PTP

payloadData fields are only included in special situations.

Field	Octets	Offset	Remarks
ntsParamIndicator	1	0	
uniqueIdentifier	16	1	
sourceAddress	SrA	17	
ticket	T	4+17+SrA	see Section 4.2.13
icvAlgoParam	IAP	17+SrA+T	

Table 28: Structure of an NTS4PTP data block in the dataBlocks field of the AUTHENTICATION TLV

The ntsParamIndicator field (see Table 29) is always present and defines which of the other fields are contained in the NTS4PTP payLoadData.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
res.	res.	res.	IcvAP	Tick	SrcA high	SrcA low	UID

Table 29: Structure of the ntsParamIndicator Field

* Bit 0 (UID) is 1 if a unique identifier (uniqueIdentifier field) is present. If the bit is 0, there is no such uniqueIdentifier field in the NTS4PTP payLoadData.
* Bits 2 and 1 (SrcA) indicate whether the sourceAddress field is present and its size. If the bits are 00, there is no sourceAddress field in the NTS4PTP payLoadData. Otherwise, the bits specify the address mode in the sourceAddress field: IPv4 (01: 4 octets), MAC (10: 6 octets) or IPv6 (11: 16 octets).
* Bit 3 (Tick) defines whether a ticket is included (1) for the transmission of encrypted security information from a PTP requester to a PTP grantor in unicast mode (TiM, ticket-based mode). If the bit is 0, no Ticket is included in the NTS4PTP payLoadData.

* Bit 4 (IcvAP) indicates an included icvAlgoParam field (1) for use with security algorithms for generating the ICV that require an initialization vector (IV) or a special nonce. If the bit is 0, no icvAlgoParam field is included in the NTS4PTP payLoadData.

The uniqueIdentifier field contains a 16-octet number that is used, for example, to mitigate start-up replay attacks.

The sourceAddress field has a length of SrA octets and contains the address information of the sending PTP instance to mitigate address spoofing attacks. Depending on the address mode the length SrA is 4 (IPv4), 6 (MAC) or 16 octets (IPv6).

The Ticket field has a length of T octets and contains the body of a ticket record (see Section 4.2.13) in the ticket-based mode of NTS4PTP (TiM). It is only present when a requester signals to a grantor that it wants a PTP unicast communication.

The icvAlgoParam field has a length of IAP octets and contains an initialization vector (IV) or a special nonce for security algorithms that require such data to generate the ICV, such as GMAC or AEAD. The length N is determined by the algorithm or specified by the administrator. IAP must be even.

5.2. Replay Protection

Since the size of the sequence numbers in the PTP header is limited to 16 bits, which is much too small, NTS4PTP uses its own sequence numbers SeqID in 32 bits in the sequenceNo field (see Section 5.1.1), not to be confused with the sequenceID of PTP in the header.

A size of 4 octets for the SeqID is large enough to avoid overflow during typical key lifetimes. The PTP sequence numbers are not affected and are used as usual.

(At a high continuous rate of 1,000 messages/sec, in 24h a value of 86.4 million would be reached and a potential overflow could occur after more than 49 days. Extreme message rates above 16,000 per second would require shorter key lifetimes).

Each connection between two PTP communication partners has its own sequence of SeqIDs. SeqIDs are only reset to zero when a new Security Association (SA) is started with a new key. This applies to both, multicast (GrM, group-based mode) and unicast (TiM, ticket-based mode) connections. In TiM, the SeqID is reset when the requester sends the first (unicast key-) secured PTP message, usually a signaling message, containing the received, partially encrypted ticket and asking for a so-called unicast contract, which contains a request for a specific PTP message type as well as the desired frame rate.

With each message exchange, the SeqID is incremented so that the receiving PTP instance can verify the correct order and replay is mitigated. Overflow is very unlikely, see above.

5.3. Start-up Replay Protection

The uniqueIdentifier field is used to fend off start-up replay attacks. It is required at device start-up to verify the current SeqID.

When a slave PTP instance is reset or restarted, it does not know the current SeqID. It connects to the NTS-KE server via TLS to obtain the current SA. The instance then sends any PTP request message (e.g., DelayReq, PDelayReq or Signaling message) to its master. In the AuthTLV, it includes a zero in the SeqID field and a 128-bit nonce as a so-called uniqueIdentifier. With the response message (Delay_Resp, Pdelay_Resp, Signaling response) it receives back the same nonce in the uniqueIdentifier field and the current SeqID of the responder in the SeqID field. This allows the slave to be sure that the message is the response to its request (challenge-response mechanism). The integrity of the message exchange is protected by the respective ICV. This mitigates replay attacks, especially at start-up. The PTP sequence numbers are not affected and are used as usual.

The unique identifier data is also used in any bi-directional PTP communication, providing additional security to this data exchange. Thus potentially limiting the effect of delay attacks to RTT/2.

5.4. Address Spoofing Protection

The sourceAddress field contains the address information of the sending PTP instance to prevent spoofing attacks. Depending on the address mode the length SrA is 4 (IPv4), 6 (MAC) or 16 octets (IPv6). For spoofing protection, this field is always present.

In principle, the ICV only secures the PTP packet. The Ethernet/UDP/IP headers are not secured. Without further checks, an attacker can manipulate the IP/MAC addresses of secured PTP packets and try to influence the BMCA with manipulated Announce messages. NTS4PTP solves this problem with an address field in the payLoadData of the NTS4PTP data block in the AuthTLV. The address information and the SourcePortID are secured and must be checked in every communication.

6. Additional Mechanisms

This section provides information about the use of the negotiated AEAD algorithm as well as the generation of the security policy pointers.

6.1. AEAD Operation

General information about AEAD:

* The AEAD operation enables the integrity protection and the optional encryption of the given data, depending on the input parameters.
* While the structure of the AEAD output after the securing operation is determined by the negotiated AEAD algorithm, it usually contains an authentication tag in addition to the actual ciphertext.
* The authentication tag provides the integrity protection, whereas the ciphertext represents the encrypted data.
* The AEAD algorithms supported in this document (see Section 4.2.1) always return an authentication tag with a fixed length of 16 octets.
* The size of the following ciphertext is equal to the length of the plaintext.
* The concatenation of authentication tag and ciphertext always form the unit "Ciphertext":
  Ciphertext = {authentication tag || ciphertext}
* Hint: The term "Ciphertext" is distinguished between upper and lower case letters.
* The following text always describes "Ciphertext".
* Separation of the information concatenated in Ciphertext is not necessary at any time.
* Six parameters are relevant for the execution of an AEAD operation:
  - AEAD (...): is the AEAD algorithm itself
  - A: Associated Data
  - N: Nonce
  - K: Key
  - P: Plaintext
  - C: Ciphertext

* The protection and encryption of the data is done as follows: C = AEAD (A, N, K, P)
* Therefore, the output of the AEAD function is the Ciphertext.
* The verification and decryption of the data is done this way: P = AEAD (A, N, K, C)
* The output of the AEAD function is the Plaintext if the integrity verification is successful.

AEAD algorithm and input/output values for the Ticket record:

* AEAD (…):
  - The AEAD algorithm that is negotiated between grantor and NTS-KE server during the registration phase.
  - A list of the AEAD algorithms considered in this document can be found in Section 4.2.1.
* Associated Data:
  - The Associated Data is an optional AEAD parameter and can be of any length and content, as long as the AEAD algorithm does not give any further restrictions.
  - In addition to the Plaintext, this associated data is also included in the integrity protection.
  - When encrypting or decrypting the Security Association record, this parameter MUST remain empty.
* Nonce:
  - Corresponds to the value from the Nonce field in the Ticket (Section 4.2.13).
  - The requirements and conditions depend on the selected AEAD algorithm.
  - For the AEAD algorithms defined in Section 4.2.1 (with numeric identifiers 15, 16, 17), a cryptographically secure random number MUST be used.
  - Due to the block length of the internal AES algorithm, the Nonce SHOULD have a length of 16 octets.
* Key:
  - This is the symmetric key required by the AEAD algorithm.
  - The key length depends on the selected algorithm.
  - When encrypting or decrypting the Security Association record, the ticket key MUST be used.
* Plaintext:
  - This parameter contains the data to be encrypted and secured.
  - For AEAD encryption, this corresponds to the Record Body of the Security Association record with all parameters inside.
  - This is also the output of the AEAD operation after the decryption process.
* Ciphertext:
  - Corresponds to the value from the Encrypted Security Association field in the Ticket (Section 4.2.13).

- The Ciphertext is the output of the AEAD operation after the encryption process.
- This is also the input parameter for the AEAD decryption operation.

6.2. SA/SP Management

This section describes the requirements and recommendations attached to SA/SPP management.

Requirements for the Security Association Database management:

* The structure and management of the Security Association Database (SAD) are implementation-dependent both on the NTS-KE server and on the PTP devices.
* An example of this, as well as other recommendations, are described in Annex P of IEEE Std 1588-2019 ([IEEE1588-2019].
* A PTP device MUST contain exactly one SAD and Security Policy Database (SPD).

Key/Key ID generation:

The generation of the keys MUST be performed by using a Cryptographically Secure Pseudo Random Number Generator (CSPRNG) on the NTS-KE server (see also Section 2.5.2). The length of the keys depends on the MAC algorithm used. The generation and management of the key ID is also controlled by the NTS-KE server. The NTS-KE server MUST ensure that every key ID is unique at least within an SA with multiple parameter sets. The value of the key ID is implementation dependent and MAY be either a random number, a hash value or an enumeration. Key IDs of expired keys MAY be reused but SHOULD NOT be reused for a certain number of rotation periods or a defined period of time. Before reusing a key ID, the NTS-KE server MUST be ensured that the key ID is no longer in use in the PTP network (e.g., within Next Parameters).

7. IANA Considerations

TBD

8. Security Considerations

TBD

9. Acknowledgements

TBD

10. References

10.1. Normative References

[fiPS-PUB-198-1]
National Institute of Standards and Technology (NIST), "The Keyed-Hash Message Authentication Code (HMAC)", NIST fiPS PUB 198-1, 2008.

[IANA_AEAD]
Internet Assigned Numbers Authority (IANA, "Authenticated Encryption with Associated Data (AEAD) Parameters", IANA AEAD Parameters (2022), December 2022, <https://www.iana.org/assignments/aead-parameters/aead-parameters.xhtml>.

[IEEE1588-2019]
Institute of Electrical and Electronics Engineers - IEEE Standards Association, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Standard 1588-2019, 2019.

[ITU-T_X.509]
International Telecommunication Union (ITU), "Information technology Open systems interconnection The Directory: Public-key and attribute certificate frameworks", ITU-T Recommendation X.509 (2008), November 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4543] McGrew, D. and J. Viega, "The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH", RFC 4543, DOI 10.17487/RFC4543, May 2006, <https://www.rfc-editor.org/info/rfc4543>.

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, DOI 10.17487/RFC5116, January 2008, <https://www.rfc-editor.org/info/rfc5116>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <https://www.rfc-editor.org/info/rfc6960>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <https://www.rfc-editor.org/info/rfc7301>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8915] Franke, D., Sibold, D., Teichel, K., Dansarie, M., and R. Sundblad, "Network Time Security for the Network Time Protocol", RFC 8915, DOI 10.17487/RFC8915, September 2020, <https://www.rfc-editor.org/info/rfc8915>.

10.2. Informative References

[Langer_2023]
Langer, M., "Secured Time Synchronization Using Packet-Based Time Protocols", Dissertation, Technical University Braunschweig, Germany, June 2023, <https://leopard.tu-braunschweig.de/servlets/MCRFileNodeServlet/dbbs_derivate_00053439/Diss_Langer_Martin.pdf>.

[Langer_et_al._2019]
Langer, M., Teichel, K., Heine, K., Sibold, D., and R. Bermbach, "Guards and Watchdogs in One-Way Synchronization with Delay-Related Authentication Mechanisms", 2019 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS), Portland, Oregon, USA, DOI 10.1109/ISPCS.2019.8886633, September 2019, <https://ieeexplore.ieee.org/document/8886633>.

[Langer_et_al._2020]
Langer, M., Heine, K., Sibold, D., and R. Bermbach, "A Network Time Security Based Automatic Key Management for PTPv2.1", 2020 IEEE 45th Conference on Local Computer Networks (LCN), Sydney, Australia, DOI 10.1109/LCN48667.2020.9314809, November 2020, <https://ieeexplore.ieee.org/document/9314809>.

[Langer_et_al._2022]
Langer, M. and R. Bermbach, "A comprehensive key management solution for PTP networks", Computer Networks, Volume 213 (2022), 109075, DOI 10.1016/j.comnet.2022.109075, June 2022, <https://www.sciencedirect.com/science/article/pii/S1389128622002158>.

[RFC4082] Perrig, A., Canetti, R., Ed., Song, D., Tygar, D., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, DOI 10.17487/RFC4082, December 2018, <https://www.rfc-editor.org/info/rfc4082>.

[RFC4493] Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The AES-CMAC Algorithm", RFC 4493, DOI 10.17487/RFC4493, June 2006, <https://www.rfc-editor.org/info/rfc4493>.

[RFC5297] Harkins, D., "Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES)", RFC 5297, DOI 10.17487/RFC5297, October 2008, <https://www.rfc-editor.org/info/rfc5297>.

Authors' Addresses

Martin Langer
Physikalisch-Technische Bundesanstalt (PTB)
Bundesallee 100
38116 Braunschweig
Germany
Email: martin.langer@ptb.de

Rainer Bermbach
Ostfalia University of Applied Sciences
Salzdahlumer Strae 46/48
38302 Wolfenbttel
Germany
Email: r.bermbach@ostfalia.de