Workgroup:	Network Time Protocol	
Internet-Draft:	draft-langer-ntp-nts-for-ptp-04	
Published:	23 August 2022	
Intended Status:	Standards Track	
Expires:	24 February 2023	
Authors:	M. Langer	R. Bermbach
	Ostfalia University	*Ostfalia University*

NTS4PTP - Key Management System for the Precision Time Protocol Based on the Network Time Security Protocol

Abstract

This document defines a key management service for automatic key management for the integrated security mechanism (prong A) of IEEE Std 1588[TM]-2019 (PTPv2.1) described there in Annex P. It implements a key management for the immediate security processing approach and offers a security solution for all relevant PTP modes. The key management service for PTP is based on and extends the NTS Key Establishment protocol defined in IETF RFC 8915 for securing NTP, but works completely independent from NTP.

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 24 February 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions

with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Notational Conventions

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.

2. Key Management Using Network Time Security

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.

This document defines such a key management service for automatic key management 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 from NTP.

Many networks include both, PTP and NTP at the same time. Furthermore, many time server appliances that are capable of acting as the Grandmaster of a PTP network are also capable of acting as an NTP server. For these reasons, it is likely to be easier both, for the time server manufacturer 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. For example, HTTPS, the predominant secure web protocol uses TLS for security. Since many PTP capable network appliances have management interfaces based on HTTPS, the manufacturers are already implementing TLS.

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 and expands it with new NTS messages for PTP. It may be applied in a Key Establishment server (NTS-KE server) that already manages NTP but can also be operated only handling KE 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) is not or cannot be used.

Using the expanded NTS Key Establishment protocol for the NTS key management for PTP, NTS4PTP provides two principle approaches specified in this document.

1. Group-based approach (GrBA, multicast)

- definition of one or more security groups in the PTP network,
- very suitable for PTP multicast mode and mixed multicast/unicast mode,
- suitable for unicast mode in small subgroups of very few participants (Group-of-2, Go2) but poor scaling and more administration work,

2. Ticket-based approach (TiBA, unicast)

- secured (end-to-end) PTP unicast communication between a PTP requester and grantor,
- no group binding necessary,
- very suitable for native PTP unicast mode, because of good scaling,
- a bit more complex NTS message handling.

For these modes, the NTS key management for PTP defines six new NTS messages which will be introduced in the sections to come:

- PTP Key Request message (see Section 2.3.1)
- PTP Key Response message (see Section 2.3.2)
- PTP Registration Request message (see Section 2.3.3)
- PTP Registration Response message (see Section 2.3.4)
- PTP Registration Revoke message (see Section 2.3.5)
- Heartbeat message (see Section 2.3.6)

This document describes the structure and usage of the two approaches GrBA and TiBA in their application as a key management system for the integrated security mechanism (prong A) of IEEE Std 1588-2019. Section 2.1 starts with a description of the principle key distribution mechanism, continues with details of the various group-based options (Section 2.1.1) and the ticket-based unicast mode (Section 2.1.2) before it ends with more general topics in Section 2.2 for example the key update process and finally an overview of the newly defined NTS messages in Section 2.3. Section 3 gives all the details necessary to construct all records forming the particular NTS messages. Section 5 depicts details of a TICKET TLV needed to transport encrypted security information in PTP unicast requests. The following Section 6 mentions specific parameters used in the PTP AUTHENTICATION TLV when working with the NTS4PTP key management system. Section 7 and Section 8 discuss IANA respectively security considerations.

2.1. Principle Key Distribution Mechanism

A PTP instance requests a key from the server referred to as the Key Establishment server, or NTS-KE server using the NTS-KE protocol defined in [RFC8915], see Section 1.3. Figure 1 describes the principle sequence which can be used for PTP multicast as well as PTP unicast operation.

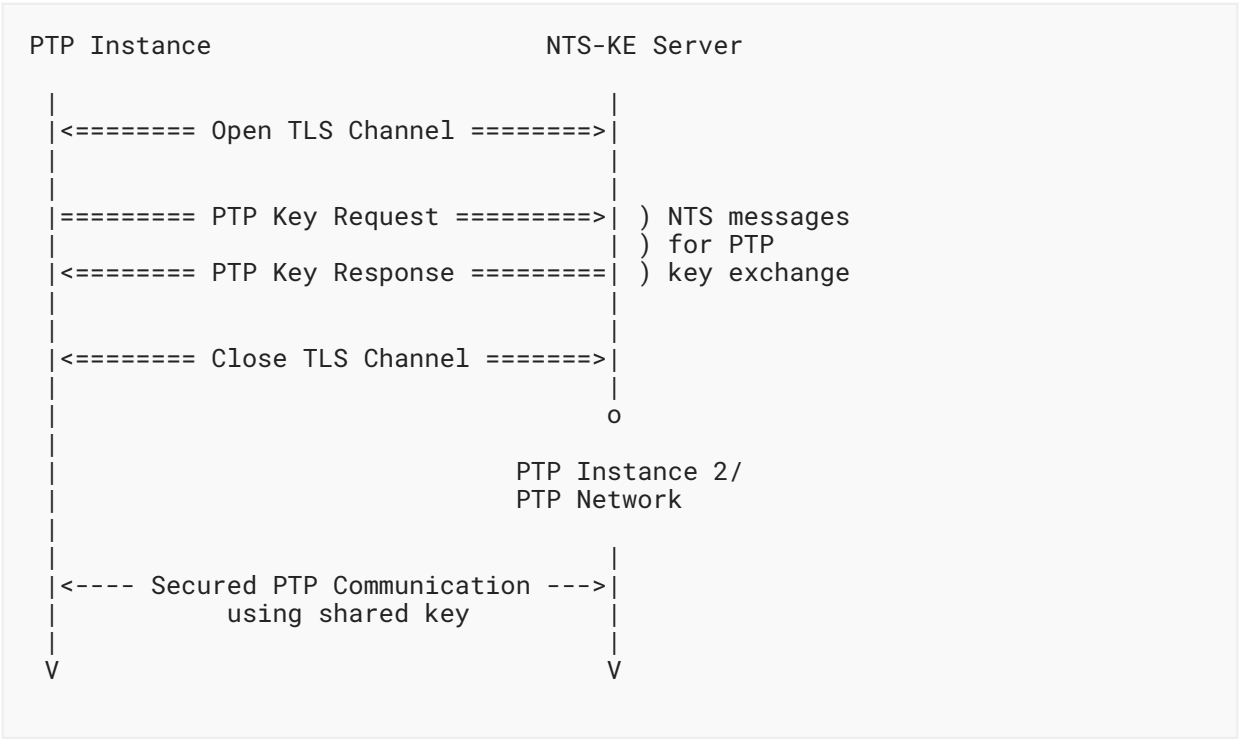

Figure 1: NTS key distribution sequence

The PTP instance client 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. No earlier TLS versions are allowed. 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]:

> *"Network Time Security makes use of TLS for NTS key establishment.*
>
> *Since the NTS protocol is new as of this publication, no backward-compatibility concerns exist to justify using obsolete, insecure, or otherwise broken TLS features or versions.*
>
> *Implementations MUST conform with RFC 7525* [RFC7525] *or with a later revision of BCP 195.*
>
> *Implementations MUST NOT negotiate TLS versions earlier than 1.3* [RFC8446] *and MAY refuse to negotiate any TLS version that has been superseded by a later supported version.*

> *Use of the Application-Layer Protocol Negotiation Extension* [RFC7301] *is integral to NTS, and support for it is REQUIRED for interoperability ... "*

The client starts the TLS handshake with a 'Client Hello' message that must contain two TLS extensions. The first extension is the Application Layer Protocol Negotiation [RFC7301] (ALPN with "ntske/1", which refers to the NTS Key Establishment as the subsequent protocol.) The second extension is the Post-Handshake Client Authentication, which the client uses to signal the TLS server that the client certificate can be requested after the TLS handshake. Afterwards, the client authenticates the NTS-KE server using the root CA certificate or by means of the Online Certificate Status Protocol (OCSP, IETF RFC 6960). Both, client and server agree on the cipher suite and then establish a secured channel that ensures authenticity, integrity and confidentiality for subsequent messages. In the process, the NTS-KE server acknowledges the ALPN and expects a message from the NTS-KE protocol.

Thus, the TLS handshake accomplishes the following:

- Negotiation of TLS version (only TLS 1.3 allowed), and
- negotiation of the cipher suite for the TLS session, and
- authentication of the TLS server (equivalent to the NTS-KE server) using a digital X.509 certificate,
- and the encryption of the subsequent information exchange between the TLS communication partners.

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).

Once the TLS session is established, the PTP instance will ask for a PTP key as well as the associated security parameters using the new NTS message PTP Key Request (see Section 2.3.1). Then the server requests the client's X.509 certificate (via TLS Certificate Request) and verifies it upon receipt. In NTS for NTP this was unnecessary, in NTS4PTP the clients MUST be authenticated, too. The NTS application of the NTS-KE server will respond with a PTP Key Response message (see Section 2.3.2). If no delivery of security data is possible for whatever reason, the PTP Key Response message contains a respective error code. All messages are constructed from specific records as described in Section 3.2.

When the PTP Key Request message was responded with a PTP Key Response, the TLS session will be closed with a 'close notify' TLS alert from both parties, the PTP instance and the key server.

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 2.2.5.4).

As mentioned above, the NTS key management for PTP supports two principle methods, the group-based approach (GrBA) and the ticket-based approach (TiBA) which are described in the following sections below.

2.1.1. NTS Message Exchange for Group-based Approach

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, asking for a specific group (see Section 2.3.1) as shown in Figure 2. After receiving the message, the NTS-KE server requests the client's certificate and performs an authorization check. The NTS-KE server then replies with a PTP Key Response message (see Section 2.3.2) 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 multicast mode but also to mixed multicast/unicast mode (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 primarily generated by a concatenation of the PTP domain number and the PTP profile identifier (sdoId), as described in Section 3.2.2.

Additionally, besides multicast and mixed multicast/unicast mode, a group of two (or few more) PTP instances can be configured, practically implementing a special group-based unicast communication mode, the group-of-2 (Go2) mode.

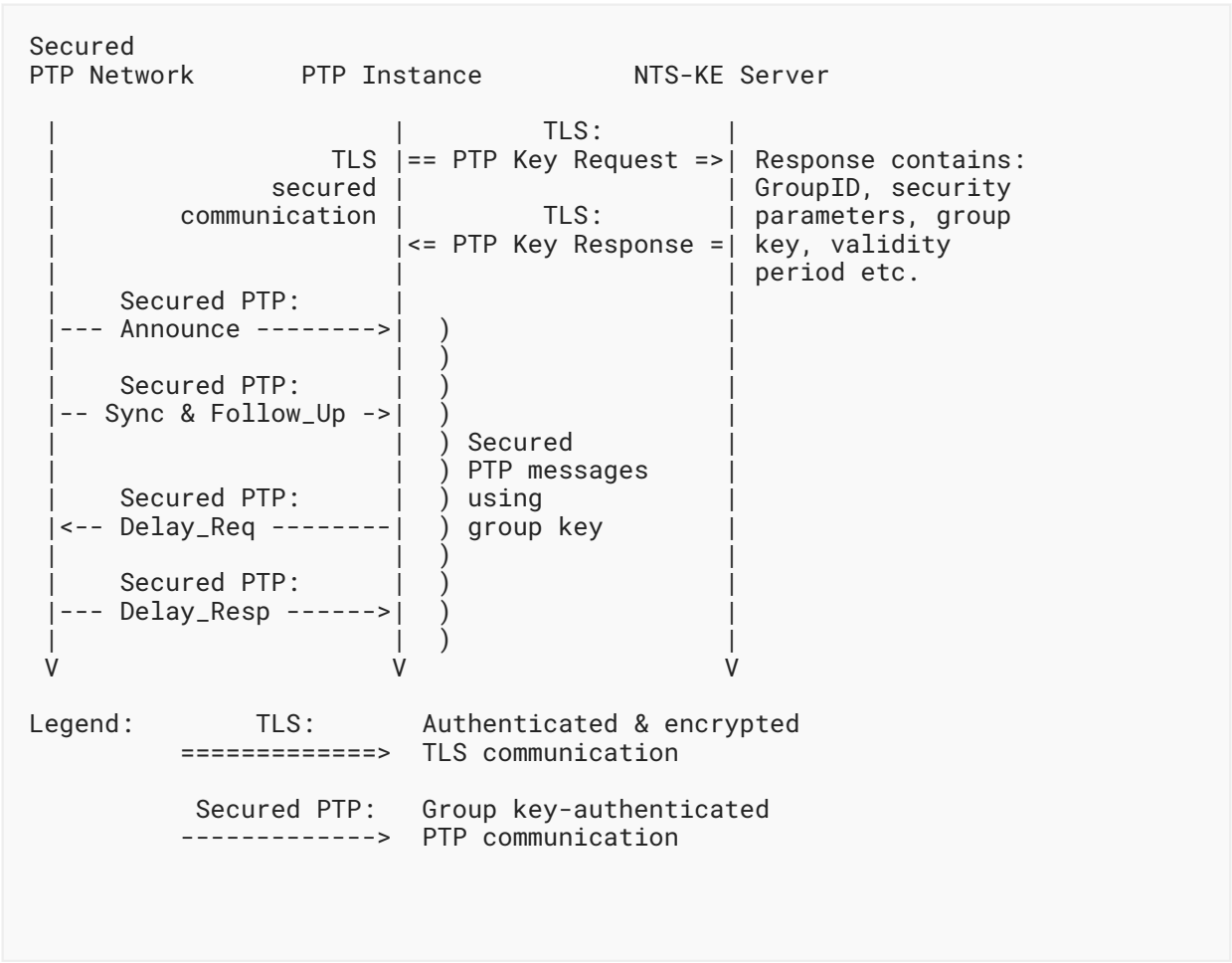

Figure 2: Message exchange for the group-based approach

This Go2 mode requires additional administration in advance defining groups-of-2 and supplying them with an additional attribute in addition to the group number mentioned for the other group-based modes - the subGroup attribute in the Association Mode record (see Section 3.2.2) of the PTP Key Request message. So, addressing for Go2 is achieved by use of the group number derived from domain number, sdoId and the additional attribute subGroup. Communication in that mode is performed using multicast addresses. If the latter is undesirable, unicast addresses can be used but the particular IP or MAC addresses of the communication partners need to be configured upfront, too.

In spite of its specific name, Go2 allows more than two participants, for example additional Transparent Clocks. All participants in that subgroup need to be configured respectively. (To enable the NTS-KE server to supply the subgroup members with the particular security data the respective certificates may reflect permission to take part in the subgroup. Else another authorization method is to be used.)

Having predefined the Go2s the key management for this mode of operation follows the same procedure (see Figure 2) and uses the same NTS messages as the other group-based modes. Both participants, the Group-of-2 requester and the respective grantor need to have received their security parameters including key etc. before secure PTP communication can take place.

After the NTS key establishment messages for these group-based modes 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 these modes works relatively simple and needs only the above mentioned two NTS messages: PTP Key Request and PTP Key Response.

2.1.2. NTS Message Exchange for the Ticket-based Approach

The scaling problems of the group-based approach are solved by the ticket-based approach (TiBA) for unicast connections. TiBA 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. Therefore, this model scales excellently with the number of connections. TiBA also allows free MAC algorithm and server negotiation, eliminating the need for the administrator to manually prepare the table of acceptable unicast masters at each individual PTP node. In addition, this allows optional load control by the NTS-KE server.

In (native) 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 3.

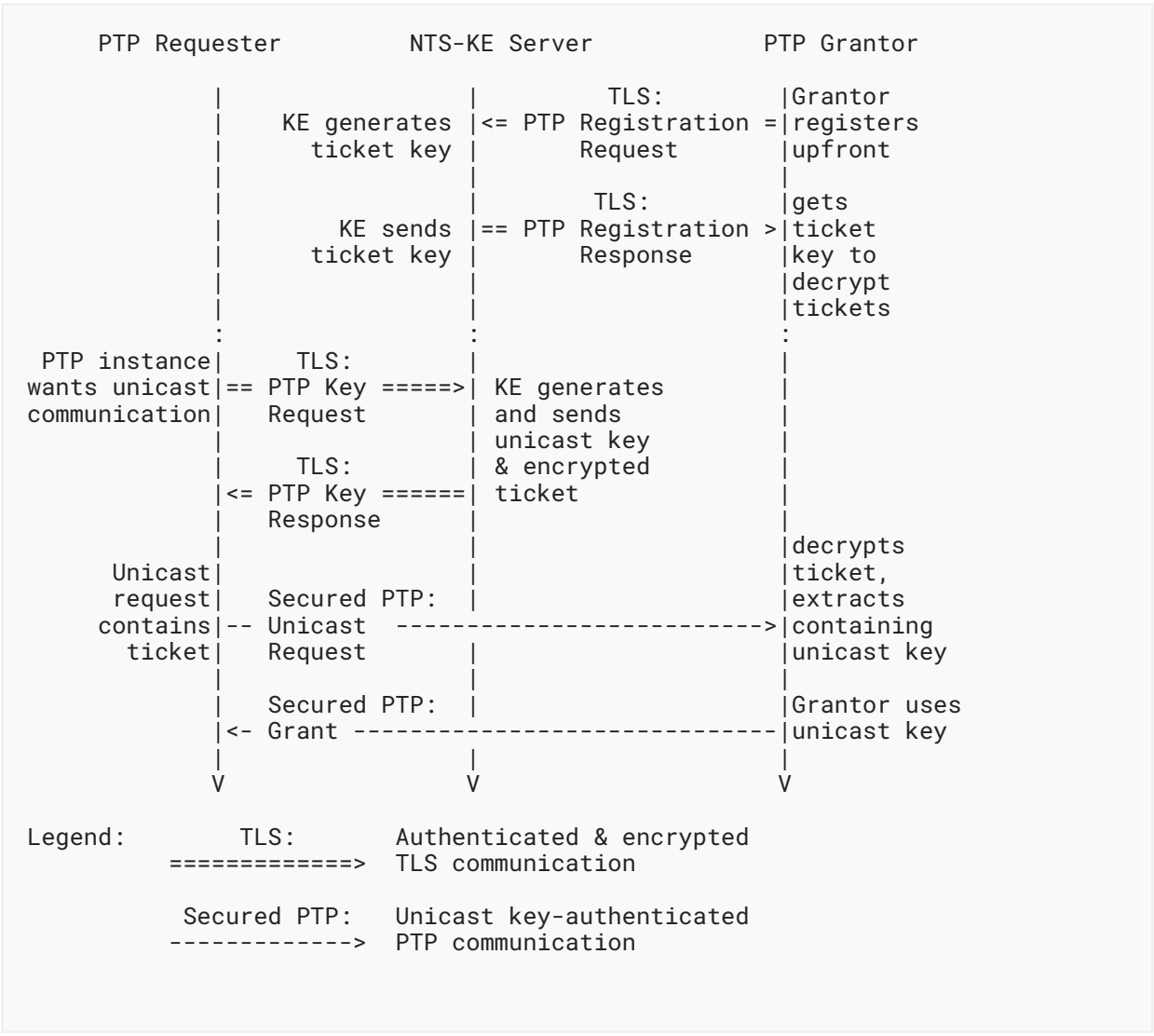

Figure 3: Message exchange for ticket-based unicast mode

(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.)

Since the registration of unicast grantors is not provided for in the NTS-KE protocol, a new sub-protocol is needed, the NTS Time Server Registration (NTS-TSR) protocol. NTS-TSR does not conflict with NTS for NTP, and the original procedure for NTS-secured NTP remains unchanged. All NTS requests still arrive at the NTS-KE server on port 4460/TCP, whether a simple client or a time server connects. The authentication of the NTS-KE server by the querying partner already takes place when the TLS connection is established. In doing so, it chooses the NTS protocol to be

used by selecting the ALPN [RFC7301]. If the ALPN contains the string "ntske/1", the NTS Key Establishment protocol is executed after the TLS handshake (see group-based approach). If it contains "ntstsr/1" instead, the NTS Time Server Registration protocol is executed. (Unlike the NTS-KE protocol, requesting grantors are already authenticated during the TLS handshake.)

The registration of a PTP grantor is performed via a PTP Registration Request message (see Section 2.3.3). The NTS-KE server answers with a PTP Registration Response message (see Section 2.3.4). 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 key request of the client (now called requester). Similar to the group-based approach, a PTP instance (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 2.3.1) as shown in Figure 7, 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 2.3.2) 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 GrBA the content differs.

The PTP Key Response message includes 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 a special Ticket TLV (see Section 5) 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 2.3.5, not shown in Figure 3) to the NTS-KE server, so requesters no longer receive PTP Key Response messages for this grantor.

The Heartbeat message (see Section 2.3.6, not shown in Figure 3) allows grantors to send messages to the NTS-KE server at regular intervals during the validity of the current security data and signal their own functionality. Optionally, these messages can contain status reports, for example, to enable load balancing between the registered time servers or to provide additional monitoring.

With its use of two protocols, the NTS-KE and the NTS-TSR protocol, this unicast mode is a bit more complex than the Group-of-2 approach and eventually uses all six new NTS messages. However, no subgroups have to be defined upfront. Addressing a grantor, the requesting instance simply may use the grantor's IP, MAC address or PortIdentity attribute.

2.2. 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.

2.2.1. 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 multicast as well as unicast mode.

The update mechanism is based on the Validity Period record in the NTS response messages, which includes the three values lifetime, update period (UP) and grace period (GP), see Figure 4. 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 up to 388 days is possible, but should not be exploited). 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. To ensure uninterrupted rotation for unicast connections, it is also necessary to ensure that the update period is greater than the minimum unicast contract time.

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. The respective values of the three parameters are defined by the administrator and can also be specified by a corresponding PTP profile.

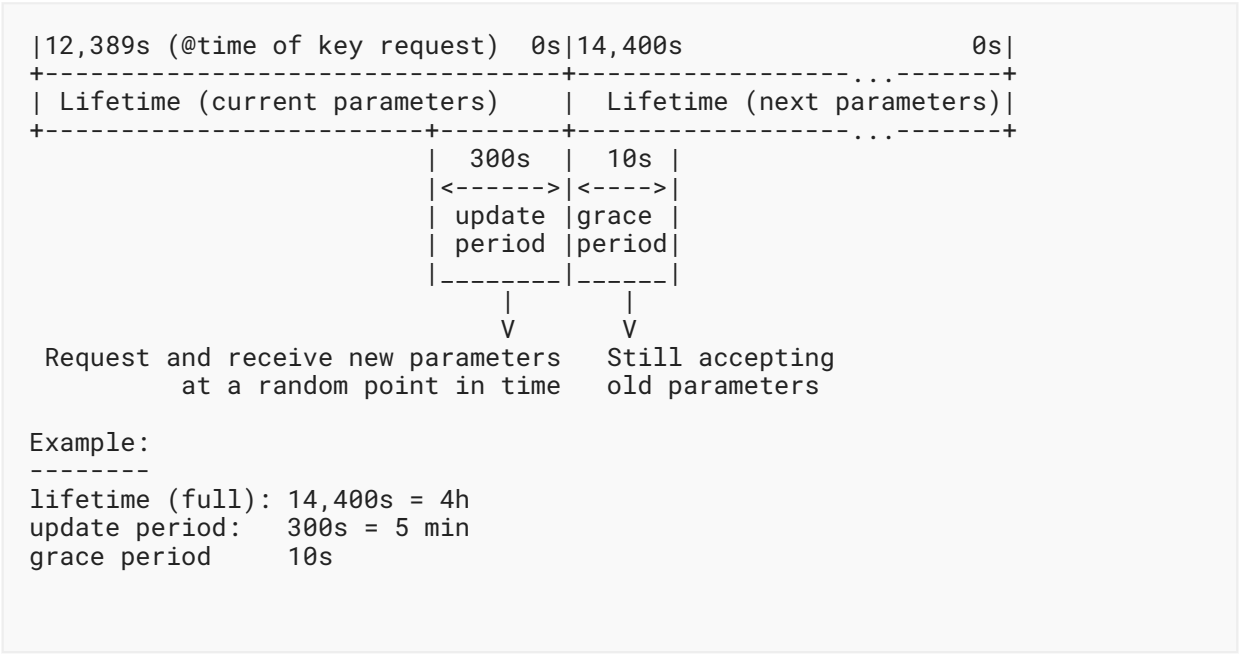

Figure 4: Example of the parameter rotation using lifetime, update period and grace period in group-based mode

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 3.2.18) with its parameters lifetime, update time 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 GrBA to receive the group SA or be it in TiBA asking for the unicast 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 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 TiBA the unicast grantors execute the NTS-TSR protocol to register with the NTS-KE server. The rotation sequence (see Figure 5) 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 in their update period.

As unicast contracts in TiBA run independently of the update cycle, a special situation may occur. If the remaining lifetime is short, it may be necessary to select a shorter time for the unicast contract validity period because the unicast contract cannot run longer than the lifetime. If a unicast contract is to be extended within the update period and the requester already owns the new ticket, it can already apply the upcoming security parameters here. This corresponds to some kind of negative grace period (pre-validity use of upcoming security parameters) and 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 unicast key. No immediate key revoke mechanism exists. The grantor SHOULD not grant respective unicast requests during the remaining lifetime of the revoked key.

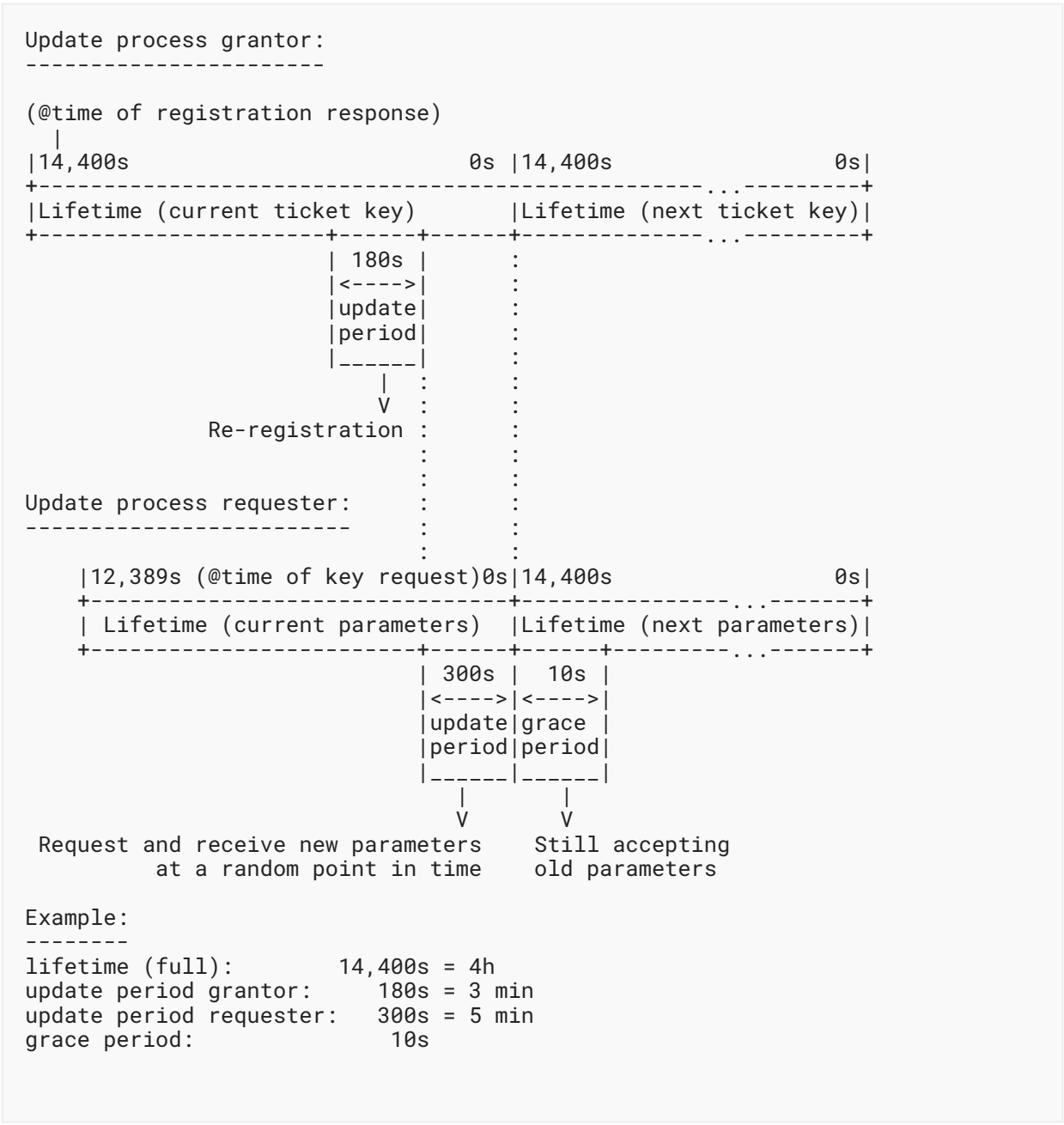

Figure 5: Example of the parameter rotation using lifetime and update period in ticket-based mode

2.2.2. Key Generation

In all cases keys obtained by a secure random number generator SHALL be used. The length of the keys depends on the MAC algorithm (see also last subsection in Section 4.2) respectively the AEAD algorithm utilized.

2.2.3. 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.

2.2.4. 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 2.2.5.4.) 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) 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 MUST be used. Additional data in the certificates like domain, sdoId and/or subgroup 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 subgroup 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.

2.2.5. 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.

2.2.5.1. 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 approach 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 protocol supported by every instance should be defined. Eventually, suitable algorithms may be defined in a respective PTP profile.

2.2.5.2. 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 approach as well as the lifetime of ticket key and unicast key in the ticket-based unicast approach (typically for every unicast pair in general or eventually specific for each requestor-grantor pair). 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.

2.2.5.3. 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 equipped with a private key and a certificate containing the public key in advance (see Section 2.2.4).

2.2.5.4. 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 approach and request the respective SA(s) and key.

As mentioned, members of a group (multicast mode, mixed multicast/unicast mode) are identified by their domain and their sdoId. 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 2.2.4.)

If the special Group-of-2 mode is used, the optional subGroup parameter (i.e., the subgroup number) needs to be specified at all members of respective Go2s, upfront. To enable the NTS-KE server to supply the subgroup members with the particular security data their respective certificates may reflect permission to take part in the subgroup. Else another authorization method is to be used.

In native unicast mode, any authenticated grantor that is member of the group used for multicast may request a registration for unicast communication at the NTS-KE server. If it is intended for unicast, this must be configured locally. If no group authorization is available (e.g., pure unicast operation) another authentication scheme is necessary.

In the same way, any requester (if configured for it locally) may request security data for a unicast connection with a specific grantor. Only authentication at the NTS-KE server using its certificate and membership in the group used for multicast is needed. If a unicast communication is not desired by the grantor, it should not grant a specific unicast request. Again, if no group authorization is available (e.g., pure unicast 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.

2.2.5.5. Transparent Clocks

Transparent Clocks (TC) need to be supplied with respective certificates, too. For group-based modes they must be configured for the particular PTP domain and sdoId and eventually for the specific subgroup(s) when using Group-of-2. They need to request for the relevant group key(s) at the NTS-KE server to allow secure use of the correctionfield in a PTP message and generation of a corrected ICV. If TCs are used in ticket-based unicast mode, they need to be authorized for the particular unicast path.

Authorization of TCs for the respective groups, subgroups and unicast connections is paramount. Otherwise the security can easily be broken with attackers pretending to be TCs in the path. Authorization of TCs is necessary too in unicast communication, even if the normal unicast partners need not be especially authorized.

Transparent clocks may notice that the communication runs secured. In the group-based approaches multicast mode and mixed multicast/unicast mode they construct the GroupID from domain and sdoId and request a group key from the NTS-KE server. Similarly, they can use the additional subgroup attribute in Go2 mode for a (group) key request. Afterwards they can check the ICV of incoming messages, fill in the correction field and generate a new ICV for outgoing messages. In ticket-based unicast mode a TC may notice a secured unicast request from a

requester to the grantor and can request the unicast key from the NTS-KE server to make use of the correction field afterwards. As mentioned above upfront authentication and authorization of the particular TCs is paramount not to open the secured communication to attackers.

2.2.5.6. 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.

Similarly, to a key re-request during an update period, key requests SHOULD be started at a random point in time after start-up to avoid peak load on the NTS-KE server. Every grantor must register with the NTS-KE server before requesters can request a unicast key (and ticket).

2.3. Overview of NTS Messages and their Structure for Use with PTP

Section 2.1 described the principle communication sequences for PTP Key Request, PTP Registration Request and corresponding response messages. All messages follow the "NTS Key Establishment Process" stated in the first part (until the description of Figure 3 starts) of Section 4 of IETF RFC 8915 [RFC8915]:

> *"The NTS key establishment protocol is conducted via TCP port 4460. The two endpoints carry out a TLS handshake in conformance with Section 3, with the client offering (via an ALPN extension* [RFC7301]*), and the server accepting, an application-layer protocol of "ntske/1". Immediately following a successful handshake, the client SHALL send a single request as Application Data encapsulated in the TLS-protected channel. Then, the server SHALL send a single response. After sending their respective request and response, the client and server SHALL send TLS "close_notify" alerts in accordance with Section 6.1 of RFC 8446* [RFC8446].
>
> *The client's request and the server's response each SHALL consist of a sequence of records formatted according to* Figure 6*. The request and a non-error response each SHALL include exactly one NTS Next Protocol Negotiation record. The sequence SHALL be terminated by a "End of Message" record. The requirement that all NTS-KE messages be terminated by an End of Message record makes them self-delimiting.*
>
> *Clients and servers MAY enforce length limits on requests and responses, however, servers MUST accept requests of at least 1024 octets and clients SHOULD accept responses of at least 65536 octets.*
>
> *The fields of an NTS-KE record are defined as follows:*
>
> - *C (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 (see Section 4.1.3).*

- *Record Type Number: A 15-bit integer in network byte order. The semantics of record types 0-7 are specified in this memo. Additional type numbers SHALL be tracked through the IANA Network Time Security Key Establishment Record Types registry.*
- *Body Length: The length of the Record Body field, in octets, as a 16-bit integer in network byte order. Record bodies MAY have any representable length and need not be aligned to a word boundary.*
- *Record Body: The syntax and semantics of this field SHALL be determined by the Record Type.*

For clarity regarding bit-endianness: the Critical Bit is the most-significant bit of the first octet. In the C programming language, given a network buffer `unsigned char b[]` containing an NTS-KE record, the critical bit is `b[0] >> 7` while the record type is `((b[0] & 0x7f) << 8) + b[1]`."

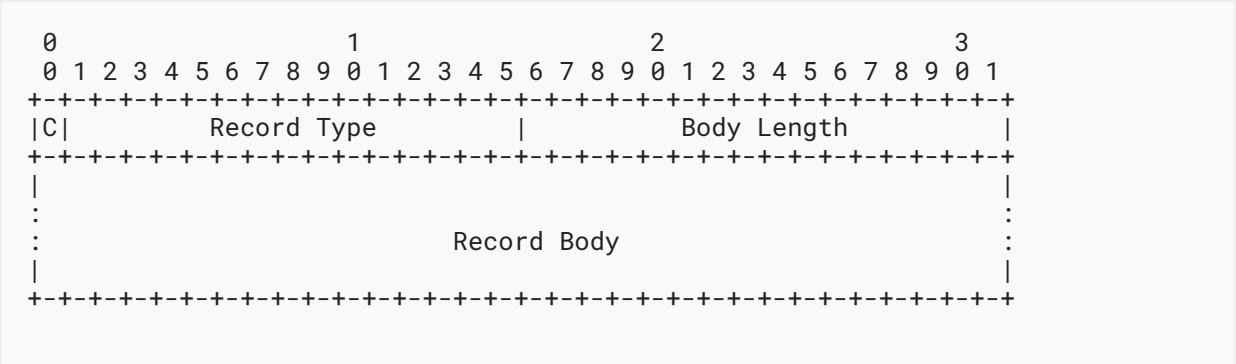

Figure 6: NTS-KE record format

Thus, 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 6. More details on record structure as well as the specific records used here are given in Section 3 and respective subsections there. So-called 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 has to be the last one in the sequence indicating the end of the current message. Container records do not include an End of Message record.

The NTS key management for PTP is based on six new NTS messages:

- PTP Key Request message (see Section 2.3.1)
- PTP Key Response message (see Section 2.3.2)
- PTP Registration Request message (see Section 2.3.3)
- PTP Registration Response message (see Section 2.3.4)
- PTP Registration Revoke message (see Section 2.3.5)
- Heartbeat message (see Section 2.3.6)

The following sections describe the principle structure of those new NTS messages for the PTP key management. More details especially on the records the messages are built of and their types, sizes, requirements and restrictions are given in Section 3.2.

2.3.1. PTP Key Request Message

```
PTP Key Request (NTS-KE protocol)
+==================================+==========================+
| Record                           | Exemplary body contents  |
+==================================+==========================+
| NTS Next Protocol Negotiation    | PTPv2.1                  |
+----------------------------------+--------------------------+
| Association Mode                 | {Assoc.Type||Assoc.Val.} |
+----------------------------------+--------------------------+
| Supported MAC Algorithms (opt.)  | CMAC                     |
+----------------------------------+--------------------------+
| Source PortIdentity (unicast only)| {binary data}           |
+----------------------------------+--------------------------+
| End of Message                   |                          |
+==================================+==========================+
```

Figure 7: Structure of a PTP Key Request message

Figure 7 shows the record structure of a PTP Key Request message. In the right column typical values are shown as examples. Detailed information on types, sizes etc. is given in Section 3.2. 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 the group-based approach the desired group number (plus eventually the subgroup attribute) is given. For ticket-based unicast communication the Association Mode contains the identification of the desired grantor, for example IPv4 and its IP address.

Only in TiBA, 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 unicast 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.

2.3.2. PTP Key Response Message

Figure 8 shows the record structure of a PTP Key Response message from the NTS-KE server (NTS-KE protocol). In the right column typical values are shown as examples. Detailed information on types, sizes etc. is given in Section 3.2. The message starts with the NTS Next Protocol Negotiation record which in this application always holds PTPv2.1.

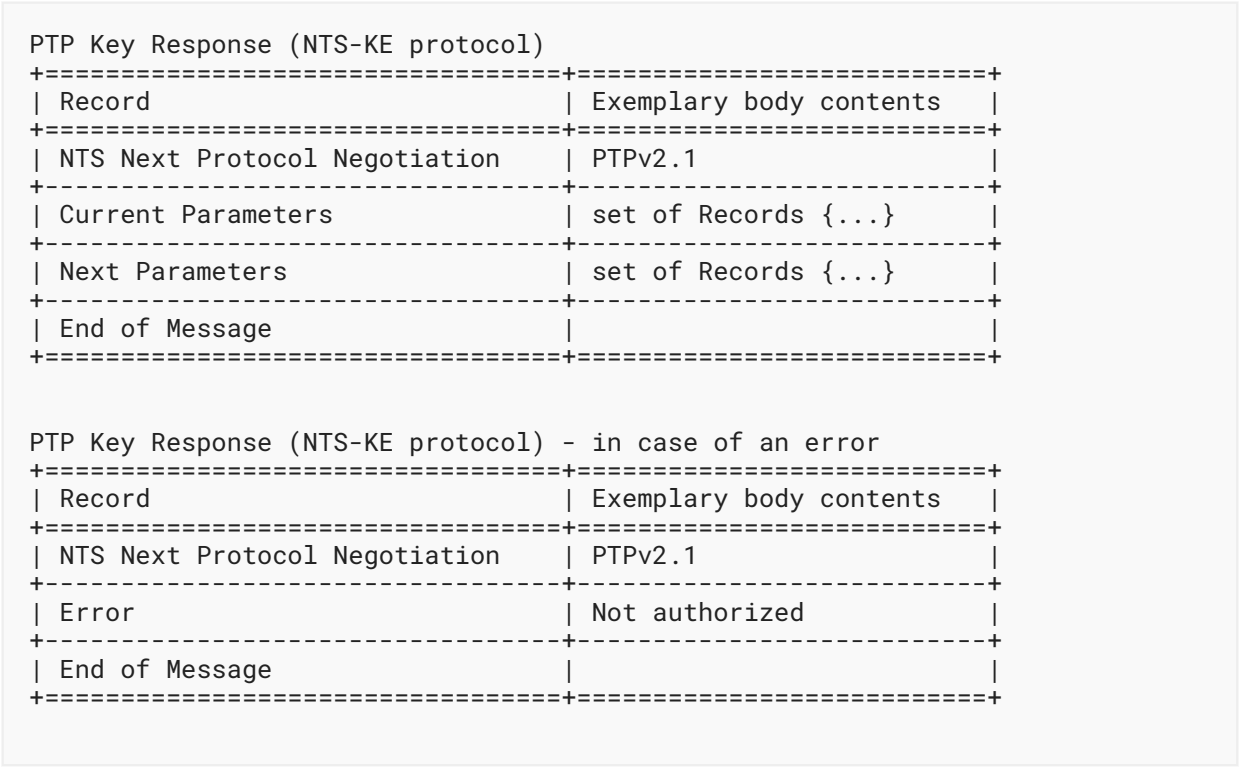

Figure 8: Structure of a PTP Key Response message.

The following Current Parameters record is a container record containing in separate records all the security data needed to join and communicate in the secured PTP communication during the current validity period. Figure 9 gives an example of data contained in that record. For more details on the records contained in the Current Parameters container record see Section 3.2.3.

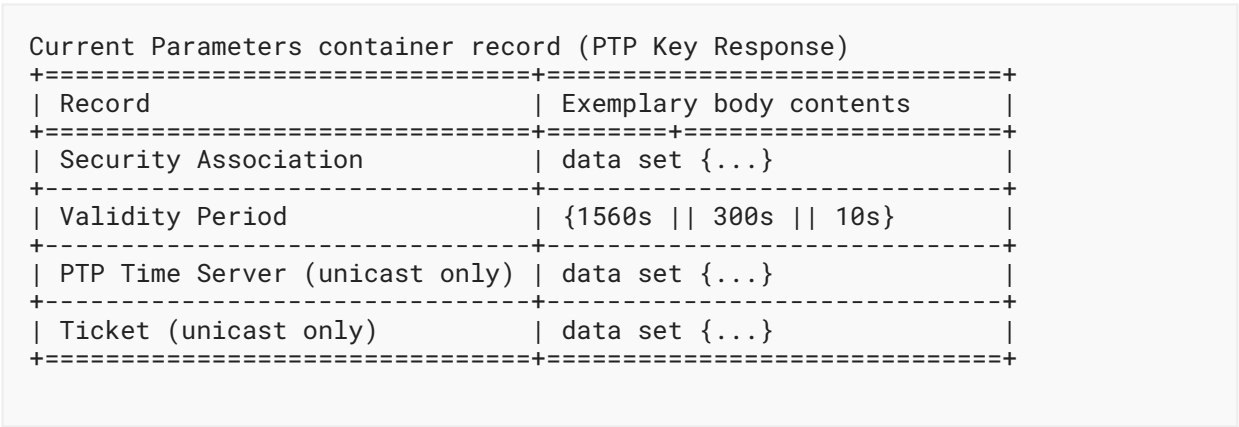

Figure 9: Exemplary contents of a Current Parameters container record of a PTP Key Response message

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 in the upcoming validity period. Its structure follows the same composition as the Current Parameters container record. In case of an error, both parameters container records are removed and a single error record is inserted (see the lower part of Figure 8). The messages always end with an End of Message record.

2.3.3. PTP Registration Request Message

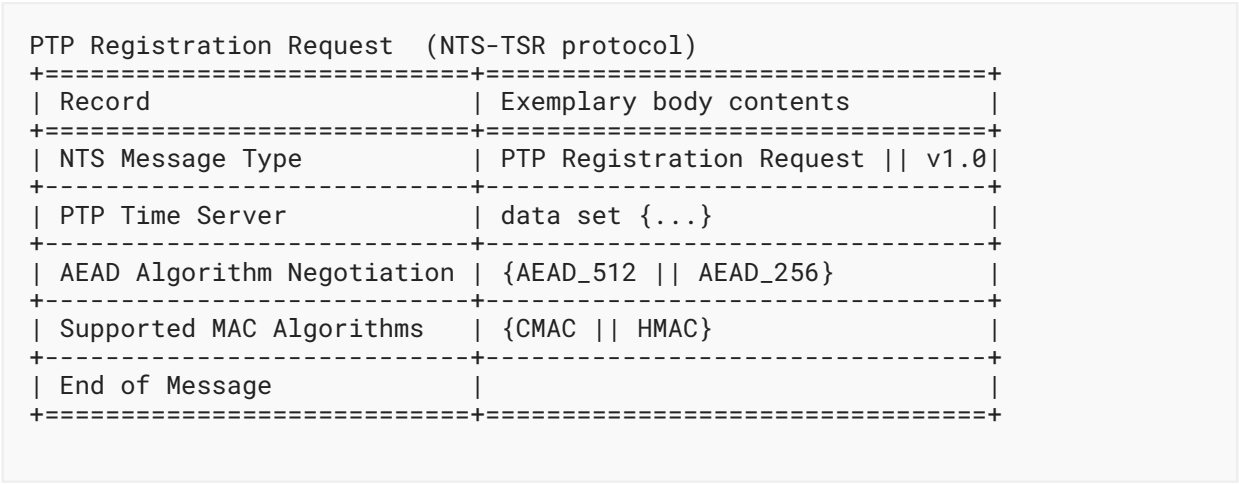

Figure 10: Structure of a PTP Registration Request message

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 Figure 10. (As the message belongs to the NTS-TSR protocol, no NTS Next Protocol Negotiation record is necessary.)

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 and comprises the MAC algorithms supported by the grantor that are feasible for calculating the ICV when securing the PTP messages in TiBA. The message always ends with an End of Message record.

2.3.4. PTP Registration Response Message

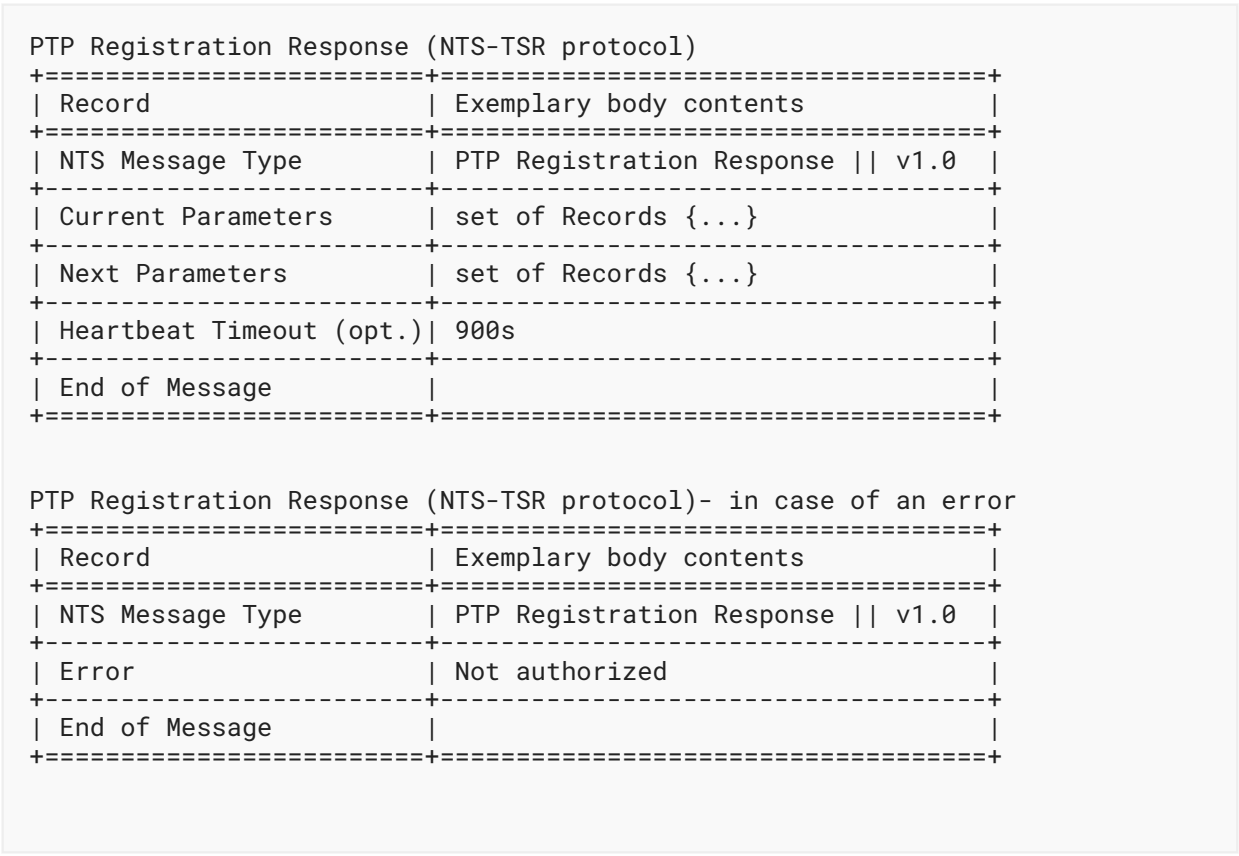

Figure 11: Structure of a PTP Registration Response message

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 Figure 11. (As the message belongs to the NTS-TSR protocol, no NTS Next Protocol Negotiation record is necessary.)

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. Figure 12 gives an example of data contained in that record. For more details on the records contained in the Current Parameters container record see Section 3.2.3.

```
Current Parameters container record (PTP Registration Response)
+================================+================================+
| Record                         | Exemplary body contents        |
+================================+================================+
| AEAD Algorithm Negotiation     | AEAD_AES_SIV_CMAC_512          |
+--------------------------------+--------------------------------+
| Validity Period                | {2460s || 400s || 10s}         |
+--------------------------------+--------------------------------+
| Ticket Key ID                  | 278                            |
+--------------------------------+--------------------------------+
| Ticket Key                     | {binary data}                  |
+================================+================================+
```

Figure 12: Exemplary contents of a Current Parameters container record of a PTP Registration Response message in the NTS-TSR protocol

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. In case of an error, both parameters container records are removed and a single error record is inserted (see the lower part of Figure 11).The messages always end with an End of Message record.

2.3.5. PTP Registration Revoke Message

```
PTP Registration Revoke (NTS-TSR protocol)
+================================+================================+
| Record                         | Exemplary body contents        |
+================================+================================+
| NTS Message Type               | PTP Registr. Revoke || v1.0    |
+--------------------------------+--------------------------------+
| Source PortIdentity            | {binary data}                  |
+--------------------------------+--------------------------------+
| End of Message                 |                                |
+================================+================================+
```

Figure 13: Structure of a 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 Figure 13. (As the message belongs to the NTS-TSR protocol, no NTS Next Protocol Negotiation record is necessary.)

The second record contains the Source PortIdentity which identifies the grantor wanting to stop its unicast support. This allows the NTS-KE server to 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.

2.3.6. Heartbeat Message

```
Heartbeat (NTS-TSR protocol)
+================================+============================+
| Record                         | Exemplary body contents    |
+================================+============================+
| NTS Message Type               | Heartbeat || v1.0          |
+--------------------------------+----------------------------+
| Status (optional)              | server load: low           |
+--------------------------------+----------------------------+
| End of Message                 |                            |
+================================+============================+
```

Figure 14: Structure of a Heartbeat message in the NTS-TSR protocol

The Heartbeat message (NTS-TSR protocol) from the grantor to 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 Figure 14. (As the message belongs to the NTS-TSR protocol, no NTS Next Protocol Negotiation record is necessary.)

The second record contains the optional Status record which allows the grantor to present various status updates to the NTS-KE server. The message always ends with an End of Message record.

Heartbeat messages provide grantors with the ability to send messages to the NTS-KE server at regular intervals to signal their own functionality. These messages can optionally also contain one or multiple status records (see Figure 14), for example to improve load balancing between the registered time servers or to provide additional monitoring. The NTS-KE server MUST accept Heartbeat messages from a grantor if they have been previously requested by the NTS-KE server in the Registration Response message. However, the NTS-KE server MAY discard heartbeat messages if they arrive more frequently than specified by the heartbeat timeout (see Section 2.3.6). If the NTS-KE server receives heartbeat messages from a grantor even though this is not requested, the NTS-KE server SHOULD discard these messages and not process them further. Processing of the status information is optional and the status records MAY be ignored by the NTS-KE server. If the Grantor sends heartbeat messages to the NTS-KE server, the frames SHOULD NOT exceed the maximum transmission unit (MTU, 1500 octets for Ethernet).

3. NTS Messages for PTP

This section covers the structure of the NTS messages and the details of the respective payload. The individual parameters are transmitted by NTS records, which are described in more detail in Section 3.2. In addition to the NTS records defined for NTP in IETF RFC8915, further records are required, which are listed in Table 1 below and begin with Record Type 1024 (compare IETF RFC 8915 [RFC8915], Section 7.6. Network Time Security Key Establishment Record Types Registry).

NTS Record Types	Description	Record Used in Protocol	Reference
0	End of Message	NTS-KE/NTS-TSR	[RFC8915], Section 4.1.1; this document, Section 3.2.4
1	NTS Next Protocol Negotiation	NTS-KE	[RFC8915], Section 4.1.2; this document, Section 3.2.8
2	Error	NTS-KE/NTS-TSR	[RFC8915], Section 4.1.3; this document, Section 3.2.5
3	Warning	NTS-KE	[RFC8915], Section 4.1.4; not used for PTP
4	AEAD Algorithm Negotiation	NTS-TSR	[RFC8915], Section 4.1.5; this document, Section 3.2.1
5	New Cookie for NTPv4	NTS-KE	[RFC8915], Section 4.1.6; not used for PTP
6	NTPv4 Server Negotiation	NTS-KE	[RFC8915], Section 4.1.7; not used for PTP
7	NTPv4 Port Negotiation	NTS-KE	[RFC8915], Section 4.1.8; not used for PTP
8 - 1023	Reserved for NTP		
1024	Association Mode	NTS-KE	This document, Section 3.2.2
1025	Current Parameters	NTS-KE/NTS-TSR	This document, Section 3.2.3
1026	Heartbeat Timeout	NTS-TSR	This document, Section 3.2.6
1027	Next Parameters Container	NTS-KE/NTS-TSR	This document, Section 3.2.7
1028	NTS Message Type	NTS-TSR	This document, Section 3.2.9
1029	PTP Time Server	NTS-KE/NTS-TSR	This document, Section 3.2.10
1030	Security Association	NTS-KE	This document, Section 3.2.11

NTS Record Types	Description	Record Used in Protocol	Reference
1031	Source PortIdentity	NTS-KE/NTS-TSR	This document, Section 3.2.12
1032	Status	NTS-TSR	This document, Section 3.2.13
1033	Supported MAC Algorithms	NTS-KE/NTS-TSR	This document, Section 3.2.14
1034	Ticket	NTS-TSR	This document, Section 3.2.15
1035	Ticket Key	NTS-TSR	This document, Section 3.2.16
1036	Ticket Key ID	NTS-TSR	This document, Section 3.2.17
1037	Validity Period	NTS-KE/NTS-TSR	This document, Section 3.2.18
1038 - 16383	Unassigned		
16384 - 32767	Reserved for Private or Experimental Use		[RFC8915]

Table 1: NTS Key Establishment and Time Server Registration record types registry

3.1. NTS Message Types

This section repeats the composition of the specific NTS messages for the PTP key management in overview form. The specification of the respective records from which the messages are constructed follows in Section 3.2. The reference column in the tables refer to the specific subsections.

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.

PTP Key Request (NTS-KE protocol)

NTS Record Name	Mode*	Use	Reference
NTS Next Protocol Negotiation	GrBA / TiBA	mandatory	This document, Section 3.2.8
Association Mode	GrBA / TiBA	mandatory	This document, Section 3.2.2

NTS Record Name	Mode*	Use	Reference
Supported MAC Algorithms	TiBA	optional	This document, Section 3.2.14
Source PortIdentity	TiBA	mandatory	This document, Section 3.2.12
End of Message	GrBA / TiBA	mandatory	This document, Section 3.2.4

Table 2: Record structure of the PTP Key Request message

* The Mode column refers to the intended use of the particular record for the respective PTP communication mode.

PTP Key Response (NTS-KE protocol)

NTS Record Name	Mode	Use	Reference
NTS Next Protocol Negotiation	GrBA / TiBA	mandatory	This document, Section 3.2.8
Current Parameters	GrBA / TiBA	mandatory	This document, Section 3.2.3
Next Parameters Container	GrBA / TiBA	mandatory (only during update period)	This document, Section 3.2.7
End of Message	GrBA / TiBA	mandatory	This document, Section 3.2.4

Table 3: Record structure of the PTP Key Response message. In case of an error, both parameters container records are removed and a single error record is inserted.

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
Security Association	GrBA / TiBA	mandatory	This document, Section 3.2.11
Validity Period	GrBA / TiBA	mandatory	This document, Section 3.2.18
PTP Time Server	TiBA	mandatory	This document, Section 3.2.10
Ticket	TiBA	mandatory	This document, Section 3.2.15

Table 4: Record structure of the container records

PTP Registration Request (NTS-TSR protocol)

NTS Record Name	Mode	Use	Reference
NTS Message Type	TiBA	mandatory	This document, Section 3.2.9
PTP Time Server	TiBA	mandatory	This document, Section 3.2.10
AEAD Algorithm Negotiation	TiBA	mandatory	This document, Section 3.2.1
Supported MAC Algorithms	TiBA	mandatory	This document, Section 3.2.14
End of Message	TiBA	mandatory	This document, Section 3.2.4

Table 5: Record structure of the PTP Registration Request message

PTP Registration Response (NTS-TSR protocol)

NTS Record Name	Mode	Use	Reference
NTS Message Type	TiBA	mandatory	This document, Section 3.2.9
Current Parameters	TiBA	mandatory	This document, Section 3.2.3
Next Parameters	TiBA	mandatory (only during update period)	This document, Section 3.2.7
Heartbeat Timeout	TiBA	optional	This document, Section 3.2.6
End of Message	TiBA	mandatory	This document, Section 3.2.4

Table 6: Record structure of the PTP Registration Response message. In case of an error, both parameters container records are removed and a single error record is inserted.

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
AEAD Algorithm Negotiation	TiBA	mandatory	This document, Section 3.2.1
Validity Period	TiBA	mandatory	This document, Section 3.2.18

NTS Record Name	Mode	Use	Reference
Ticket Key ID	TiBA	mandatory	This document, Section 3.2.17
Ticket Key	TiBA	mandatory	This document, Section 3.2.16

Table 7: Record structure of the container records in the PTP Registration Response message

PTP Registration Revoke (NTS-TSR protocol)

NTS Record Name	Mode	Use	Reference
NTS Message Type	TiBA	mandatory	This document, Section 3.2.9
Source PortIdentity	TiBA	mandatory	This document, Section 3.2.12
End of Message	TiBA	mandatory	This document, Section 3.2.4

Table 8: Record structure of the PTP Registration Revoke message

Heartbeat Message (NTS-TSR protocol)

NTS Record Name	Mode	Use	Reference
NTS Message Type	TiBA	mandatory	This document, Section 3.2.9
Status	TiBA	optional	This document, Section 3.2.13
End of Message	TiBA	mandatory	This document, Section 3.2.4

Table 9: Record structure of the Heartbeat message in the NTS-TSR protocol

3.2. 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 3.1 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.

3.2.1. AEAD Algorithm Negotiation

Used in NTS-TSR protocol

This record is required in unicast mode and enables the negotiation of the AEAD algorithm needed to encrypt and decrypt the ticket. 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. (https://www.iana.org/assignments/aead-parameters/aead-parameters.xhtml)
- 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 10.
- Other AEAD algorithms MAY also be used.

Numeric ID	AEAD Algorithm	Use	Key Length (Octets)	Reference
15	AEAD_AES_SIV_CMAC_256	mand.	16	[RFC5297]
16	AEAD_AES_SIV_CMAC_384	opt.	24	[RFC5297]
17	AEAD_AES_SIV_CMAC_512	opt.	32	[RFC5297]
32 - 32767	Unassigned			
32768 - 65535	Reserved for Private or Experimental Use			[RFC5116]

Table 10: AEAD algorithms

- In a PTP Registration Request message, this record MUST be contained exactly once.
- In that message at least the AEAD_AES_SIV_CMAC_256 algorithm MUST be included.
- 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.
- In a PTP Registration Response message, this record MUST be contained exactly once in the Current Parameters container record and exactly once in the Next Parameters container record.
- The Next Parameters container record MUST be present only during the update period.
- 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.
- In a PTP Registration Response message, this record MUST contain exactly one AEAD algorithm.

- The selected algorithm MAY differ in the corresponding Current Parameters container record and Next Parameters container record.

3.2.2. Association Mode

Used in NTS-KE protocol

This record enables the NTS-KE server to distinguish between a group based request (multicast, mixed multicast/unicast, Group-of-2) or a unicast request. A multicast request carries a group number, while a unicast request contains an identification attribute of the 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 1024 and the Critical Bit MAY be set.
- The Record Body SHALL consist of two data fields:

field	Octets	Offset
Association Type	2	0
Association Value	A	2

Table 11: Association

- 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 of Association Type as well as the length and content of Association Value are shown in the following table and more details are given below.

Description	Assoc. Type Number	Association Mode	Association Value Content	Assoc. Value Octets
Group	0	Multicast / Unicast*	Group Number	5
IPv4	1	Unicast	IPv4 address of the target port	4
IPv6	2	Unicast	IPv6 address of the target port	16
802.3	3	Unicast	MAC address of the target port	6
PortIdentity	4	Unicast		10

Description	Assoc. Type Number	Association Mode	Association Value Content	Assoc. Value Octets
			PortIdentity of the target PTP entity	

Table 12: Association Types

Unicast*: predefined groups of two (Group-of-2, Go2, see Group entry below)

Group:

- This association type allows a PTP instance to join a PTP multicast group.
- A group is identified by the PTP domain, the PTP profile (sdoId) and a sub-group attribute (see table below).
- The PTP domainNumber is an 8-bit unsigned integer in the closed range 0 to 255.
- The sdoId of a PTP domain is a 12-bit unsigned integer in the closed range 0 to 4095:
  - The most significant 4 bits are named the majorSdoId.
  - The least significant 8 bits are named the minorSdoId.
  - Reference: IEEE Std 1588-2019, Section 7.1.1

  sdoId = {majorSdoId || minorSdoId}

- The subGroup is 16-bit unsigned integer, which allows the division of a PTP multicast network into separate groups, each with individual security parameters.
- This also allows manually configured unicast connections (Group-of-2), which can include transparent clocks as well.
- The subGroup number is defined manually by the administrator.
- Access to the groups is controlled by authorization procedures of the PTP devices (see Section 2.2.5.4).
- If no subgroups are required (= multicast mode), this attribute MUST contain the value zero.
- The group number is eventually formed by concatenation of the following values:
  group number = {domainNumber || 4 bit zero padding || sdoId || subGroup}

This is equvalent to:

Bits 7 - 4	Bits 3 - 0	Octets	Offset
domainNumber (high)	domainNumber (low)	1	0
zero padding	majorSdoId	1	1
minorSdoId (high)	minorSdoId (low)	1	2
subgroup (high)	subGroup (low)	2	4

Table 13: Group Association

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 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 and portNumber:
  PortIdentity = **{clockIdentity || portNumber}**
- The clockIdentity is an 8 octet array and the portNumber is a 16-bit unsigned integer.
- Source: IEEE Std 1588-2019, Sections 5.3.5 and 7.5

3.2.3. Current Parameters

Used in 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 1025 and the Critical Bit MAY be set.
- In a PTP Key Response message, this record MUST be contained exactly once.
- The Record Body is defined as a set of records and MAY contain the following records:

NTS Record Name	Comunication Type	Use	Reference
Security Associations (one or more)	Multicast / Unicast	mandatory	This document, Section 3.2.11
Validity Period	Multicast / Unicast	mandatory	This document, Section 3.2.18
PTP Time Server	Unicast	mandatory	This document, Section 3.2.10
Ticket	Unicast	mandatory	This document, Section 3.2.15

Table 14: Current Parameters container for PTP Key Response message

- 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 wants a unicast connection and MUST NOT be included if the client wants to join a multicast group.
- 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:

- In a PTP Registration Response message, the Current Parameters Container record MUST be contained exactly once.
- The record body MAY contain the following records:

NTS Record Name	Use	Reference
AEAD Algorithm Negotiation	mandatory	This document, Section 3.2.1
Validity Period	mandatory	This document, Section 3.2.18
Ticket Key ID	mandatory	This document, Section 3.2.17
Ticket Key	mandatory	This document, Section 3.2.16

Table 15: Current Parameters container for PTP Registration Response Message

3.2.4. End of Message

Used in NTS-KE and NTS-TSR protocol

The End of Message record is defined in IETF RFC8915 [RFC8915], Section 4:

> *"The record sequence in an NTS message SHALL be terminated by an "End of Message" record. The requirement that all NTS-KE messages be terminated by an End of Message record makes them self-delimiting."*

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 request and response, NTS registration revoke and heartbeat message.
- This record SHOULD NOT be included in the container records and MUST be ignored if present.
- See also: IETF RFC8915, Section 4.1.1

3.2.5. Error

Used in NTS-KE and NTS-TSR protocol

The Error record is defined in IETF RFC8915 [RFC8915], Section 4.1.3. In addition to the Error codes 0 to 2 specified there the following Error codes 3 to 4 are defined:

Error Code	Description
0	Unrecognized Critical Record
1	Bad Request
2	Internal Server Error
3	Not Authorized
4	Grantor not Registered
5 - 32767	Unassigned
32768 - 65535	Reserved for Private or Experimental Use

Table 16: 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 3 "Not Authorized" is sent by the NTS-KE server if the requester is not authorized to join the desired multicast group or if a grantor is prohibited to register with the NTS-KE server.
- The Error record MUST NOT be included in a PTP Registration Request message.
- The Error code 4 "Grantor not Registered" is sent by the NTS-KE server when the requester wants to establish a unicast connection to a grantor that is not registered with the NTS-KE server.
- The Error record MUST NOT be included in a PTP Key Request message.

3.2.6. Heartbeat Timeout

Used in NTS-TSR protocol

This record provides the NTS-KE server the capability to monitor the availability of the registered grantors. If this optional record is used, the registered grantors SHOULD send an NTS Heartbeat message to the NTS-KE server before the timeout expires.

Content and conditions:

- The record has a Record Type number of 1026 and the Critical Bit SHOULD NOT be set.
- The Record Body consists of a 16-bit unsigned integer in network byte order and denotes the heartbeat timeout in seconds..
- The timeout set by the NTS-KE server MUST NOT be less than 1s and MUST be less than the lifetime set in the Validity Period record.
- The timeout starts at the NTS-KE server with the generation of the Registration Response message.
- Grantors that receive an invalid value as a heartbeat timeout MUST ignore this record and MUST NOT send heartbeat messages.
- Grantors that receive a valid value SHOULD send a heartbeat message to the NTS-KE server before the timeout has elapsed.
- The grantors SHOULD keep the heartbeat intervals and MAY also send heartbeat messages more frequently.
- After transmitting a heartbeat from the grantor to the NTS-KE server, both sides reset the timeout to the start value and let the time count down again.
- If this timeout is exceeded without receiving a heartbeat message or several heartbeats are missing in a row, the NTS-KE server MAY delete the grantor from its registration list, so that a new registration of the grantor is necessary.
- Grantors that are not (or no longer) registered with a NTS-KE server MUST NOT send heartbeat messages and NTS-KE servers MUST discard heartbeat messages from non-registered grantors.
- NTS-KE servers MAY respond in such cases with a Registration Response message containing error code 4 "Grantor not Registered".

3.2.7. Next Parameters

Used in 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 1027 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 3.2.3 of this document.
- In both the PTP Key Response and PTP Registration Response message, this record MUST be contained exactly once during the update period.
- Outside the update period, this record MUST NOT be included.
- In GrBA mode, 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 2.2.1.

3.2.8. NTS Next Protocol Negotiation

Used in NTS-KE protocol

The Next Protocol Negotiation record is defined in IETF RFC8915 [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 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
1	Precision Time Protocol version 2.1 (PTPv2.1)	This document
2 - 32767	Unassigned	
32768 - 65535	Reserved for Private or Experimental Use	

Table 17: 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.

3.2.9. NTS Message Type

Used in 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 1028 and the Critical Bit MUST be set.
- The Record Body MUST consist of three data fields:

Field	Octets	Offset
Message Type	2	0

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

Table 18: 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 are defined for the Message Type:

Message Type (value)	NTS Message (NTS-TSR protocol)
0	PTP Registration Request
1	PTP Registration Response
2	PTP Registration Revoke
3	Heartbeat
4 - 32767	Unassigned
32768 - 65535	Reserved for Private or Experimental Use

Table 19: 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.

3.2.10. PTP Time Server

Used in NTS-KE and NTS-TSR protocol

The PTP Time Server record is used exclusively in TiBA mode (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 1029 and the Critical Bit MAY be set.
- The record body consists of a tuple of two 8-bit unsigned integers in network byte order.
- The structure of the record body and all conditions MUST be identical to the rules described in Section 3.2.2 (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 can be an IPv4, IPv6, MAC address, as well as the PortIdentity of the grantor.
- This allows the client to change the PTP transport mode (e.g., from IPv4 to 802.3) without performing a new NTS request.
- The list in the PTP Time Server record MUST NOT contain the Association Type number 0 (multicast group) and MUST contain at least one entry.
- The NTS-KE server SHOULD provide the grantor addresses requested by the client in the PTP Key Request message, but MAY also assign a different grantor to the client.

- 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.

3.2.11. Security Association

Used in NTS-KEprotocol

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 6.

Content and conditions:

- The record has a Record Type number of 1030 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
Security Parameter Pointer	1	0
Integrity Algorithm Type	2	1
Key ID	4	3
Key Length	2	7

Field	Octets	Offset
Key	K	9

Table 20: Security Association record

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

Security Parameter Pointer

- The Security Parameter Pointer (SPP) is an 8-bit unsigned integer in the closed range 0 to 255.
- This value enables the mutual assignment of SA, SP and AUTHENTICATION TLVs.
- The generation and management of the SPP is controlled by the NTS-KE server (see Section 4.2).

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 3.2.14.
- 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.
  - The value can be either a random number or an enumeration.
  - Previous key IDs SHOULD NOT be reused for a certain number of rotation periods or a defined period of time (see Section 4.2).

Key Length

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

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 (GrBA mode) or a unicast key (TiBA mode).

3.2.12. Source PortIdentity

Used in 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 1031 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 TiBA mode and MUST NOT be included if the client intends to join a multicast group/Go2 (= GrBA mode).
- In a PTP Registration Revoke message, this record MUST be included 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)

3.2.13. Status

Used in NTS-TSR protocol

The Status record is an optional record that represents the current load of the sender. It allows the NTS-KE server to improve load balancing when assigning grantors to the requesting PTP clients in TiBA mode. The content of the record is designed in such a way that it can also transmit other information (e.g., manufacturer-related information).

Content and conditions:

- The record has a Record Type number of 1032 and the Critical Bit SHOULD NOT be set.
- The Record Body MUST consist of two data fields:

Field	Octets	Offset
Status Type	2	0

Field	Octets	Offset
Status Data	D	2

Table 21: Structure of the Status record

- The Status Type is a 16-bit unsigned integer, denoting the content of the Status Data field.
- The Status Data field is a sequence of octets in network byte order whose length, content and structure is determined by the Status Type field.
- The following values are currently set:

Status Type	Status Data length	Description
0	1 octet (unsigned int)	grantor load
1 - 32767	Unassigned	
32767 - 65535	Reserved for Private or Experimental Use	

Table 22: Values for Status Data

- The following values apply to Status Type 0:

Status Type	Status Data value	Description
0	0x01	grantor load: 0% to 24%
0	0x02	grantor load: 25% to 49%
0	0x03	grantor load: 50% to 74%
0	0x04	grantor load: 75% to 84%
0	0x05	grantor load: 85% to 94%
0	0x06	grantor load: 95% to 100%

Table 23: Values for Status Type 0

- In a Heartbeat message this record MAY be contained once or several times.
- If multiple status records are included, the status type MUST NOT occur twice.
- The NTS-KE server MAY use the status record for optimizations and MAY also ignore them.

3.2.14. Supported MAC Algorithms

Used in 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 1033 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
0	HMAC-SHA256-128	16	[fiPS-PUB-198-1], [IEEE1588-2019]
1	HMAC-SHA256	32	[fiPS-PUB-198-1]
2	AES-CMAC	16	[RFC4493]
3	AES-GMAC-128	16	[RFC4543]
4	AES-GMAC-192	24	[RFC4543]
5	AES-GMAC-256	32	[RFC4543]
6 - 32767	Unassigned		
32768 - 65535	Reserved for Private or Experimental Use		

Table 24: MAC Algorithms

In GrBA mode:

- 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 massage and the NTS-KE server MUST ignore this record if the Association Type in the Association Mode record is 0 (= multicast group).
- Unless this is specified otherwise by a PTP profile, the HMAC-SHA256-128 algorithm SHALL be used by default.

In TiBA mode:

- In a PTP Key Request message, this record MAY be contained if the requester wants a unicast connection (TiBA mode, not Go2) to a specific grantor.
- The requester MUST NOT send more than one record of this type.

- If this record is present, at least the HMAC-SHA256-128 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.

3.2.15. Ticket

Used in NTS-KE protocol

This record contains the parameters of the selected AEAD algorithm, 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 1034 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

Field	Octets	Offset
Encrypted SA Length	2	N+16
Encrypted Security Association	E	N+18

Table 25: 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 the Next Parameters container record if the requesting client wants a unicast communication to a specific grantor in TiBA mode.
- The Next Parameters container record MUST be present only during the update period.

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 3.2.17 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.

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 4.1.

Encrypted SA Length

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

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 3.2.11.
- More details about the AEAD process and the required input data are described in Section 4.1.

3.2.16. Ticket Key

Used in 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 1035 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.
- The Next Parameters container record MUST be present only during the update period.

3.2.17. 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 1036 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 either a random number, a hash value or an enumeration.
  - Previous IDs SHOULD NOT be reused for a certain number of rotation periods or a defined period of time.
- In a PTP Key Response message, this record MUST be included exactly once each in the Current Parameters container record and the Next Parameters container record if a unicast connection in TiBA mode is to be established.
- If the requester wishes to join a multicast group, the Ticket Key ID record MUST NOT be included in the container records.

- 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 Next Parameters container record MUST be present only during the update period.
- The Ticket record MUST be present in TiBA mode and MUST NOT be present in GrBA mode.

3.2.18. Validity Period

Used in NTS-KE and NTS-TSR protocol

This record contains the validity information of the respective security parameters (see also Section 2.2.1).

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 the Next Parameters container record.
- The record has a Record Type number of 1037 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 26: 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 maximum value is set by the NTS-KE administrator or the PTP profile.
- In conjunction with a PTP unicast establishment in TiBA mode, 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 3.2.7.

Grace Period

- The Grace Period is a 32-bit unsigned integer in network byte order.
- It defines how many seconds expired security parameters MUST 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 5 to 10 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.

4. Additional Mechanisms

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

4.1. 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 3.2.1) 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 3.2.1.
- 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 3.2.15).

    - The requirements and conditions depend on the selected AEAD algorithm.
    - For the AEAD algorithms defined in Section 3.2.1 (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 3.2.15).
    - The Ciphertext is the output of the AEAD operation after the encryption process.
    - This is also the input parameter for the AEAD decryption operation.

4.2. SA/SP Management

This section describes the requirements and recommendations attached to SA/SP management, as well as details about the generation of identifiers.

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).
- For multicast and Group-of-2 connections, SPPs MUST NOT occur more than once in the SAD of a PTP device.
- For unicast connections, SPPs MAY occur more than once in the SAD of a PTP device.
- The NTS-KE server MUST ensure that SPPs can be uniquely assigned to a multicast group or unicast connection.
- This concerns both the NTS-KE server and all PTP devices assigned to the NTS-KE server.

SPP generation:

> The generation of the SPP always takes place on the NTS-KE server and enables the identification of a corresponding SA. The value of the SPP can be either a random number or an enumeration. An SPP used in any multicast group MUST NOT occur in any other multicast group or unicast connection. If a multicast group or unicast connection is removed by the NTS-KE server, the released SPPs MAY be reused for new groups or unicast connections. Before reusing an SPP, the NTS-KE server MUST ensure that the SPP is no longer in use in the PTP network (e.g., within Next Parameters). In different PTP devices, an SPP used in a unicast connection MAY also occur in another unicast connection, as long as they are not used in multicast groups.

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 2.2.2). 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).

5. New TICKET TLV for PTP Messages

Once a PTP port is registered as a grantor for association in unicast mode another PTP port (requester) can associate with it by first requesting a key from the NTS-KE server with Association Type in the Association Mode record set to one of the values 1 to 4 (IPv4, IPv6, 802.3 or PortIdentity), and Association Values to the related address of the desired grantor. After the reception of a PTP Key Response message during the NTS-KE protocol the requester obtains the unicast key and the Ticket record containing the Record Body of the Security Association record (see Section 2.1.2 and Section 3.2.15). The ticket includes the identification of the requester, the Encrypted SA along with the unicast key as well as the lifetime in the Validity record.

To provide the grantor with the security data, the requester sends a secured unicast request to the grantor, e.g., an Announce request (= Signaling message with a REQUEST_UNICAST_TRANSMISSION TLV with Announce as messageType in the TLV), which is secured with the unicast key.

To accomplish that, the requester sends a newly defined TICKET TLV with the Ticket embedded and the AUTHENTICATION TLV with the PTP unicast negotiation message. The TICKET TLV must be positioned before the AUTHENTICATION TLV to include the TICKET TLV in the securing by the ICV. The receiving grantor decrypts the Ticket (actually the encrypted security association) from the TICKET TLV getting access to the information therein. With the contained unicast key, the grantor checks the requester identity and the authenticity of the request message.

Thereafter, all secured unicast messages between grantor and requester will use the unicast key for generating the ICV in the AUTHENTICATION TLV for authentication of the message until the unicast key expires.

If the requester's identity does not match with the Source PortIdentity field in the Ticket or the ICV in the AUTHENTICATION TLV is not identical to the generated ICV by the grantor, then the unicast request message MUST be denied.

The TICKET TLV structure is given in Table 27 below.

field	Octets	Offset
tlvType	2	0
lengthfield	2	2
Ticket record	T	4

Table 27: Structure of the TICKET TLV

To comply with the TLV structure of IEEE Std 1588-2019 ([IEEE1588-2019], Section 14.1) the TICKET TLV is structured as presented in Table 27 with a newly defined tlvType, a respective length field and the Ticket record (see Section 3.2.15) containing the encrypted security association. Eventually the Ticket TLV may be defined externally to IEEE 1588 SA, e.g., by the IETF. Then the structure should follow IEEE Std 1588-2019 ([IEEE1588-2019], Section 14.3) to define a new standard organization extension TLV as presented in Table 28 below.

field	Octets	Offset
tlvType	2	0
lengthfield	2	2
organizationId	3	4
organizationSubType	3	7
Ticket record	T	10

Table 28: Structure of an organization extension TLV form for the TICKET TLV

The TICKET TLV will be added to the PTP message preceding the AUTHENTICATION TLV as shown in figure 48 of IEEE Std 1588-2019 ([IEEE1588-2019], Section 16.14.1.1).

6. AUTHENTICATION TLV Parameters

The AUTHENTICATION TLV is the heart of the integrated security mechanism (prong A) for PTP. It provides all necessary data for the processing of the security means. The structure is shown in Table 29 below (compare to figure 49 of [IEEE1588-2019]).

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 29: Structure of the AUTHENTICATION TLV

The tlvType is AUTHENTICATION and lengthfield gives the length of the TLV. When using the AUTHENTICATION TLV with NTS key management, the SPP and keyID will be provided by the NTS-KE server in the PTP Key Response message

The optional disclosedKey, sequenceNo, and RES fields are omitted. So all of the flags in the SecParamIndicator MUST be FALSE.

ICV field contains the integrity check value of the particular PTP message calculated using the integrity algorithm defined by the key management.

7. IANA Considerations

Considerations should be made ...

...

8. Security Considerations

...

9. Acknowledgements

The authors would like to thank ...

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.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <https://www.rfc-editor.org/info/rfc7525>.

[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_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>.

Authors' Addresses

Martin Langer
Ostfalia University of Applied Sciences
Salzdahlumer Straße 46/48
38302 Wolfenbüttel
Germany
Email: mart.langer@ostfalia.de

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