Workgroup:	Network Time Protocol	
Internet-Draft:	draft-langer-ntp-nts4ptp-01	
Published:	8 March 2021	
Intended Status:	Standards Track	
Expires:	9 September 2021	
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 described there in Annex P. It implements a key management for immediate security processing complementing the exemplary GDOI proposal in P.2.1.2.1. The key management service is based on 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 9 September 2021.

Copyright Notice

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

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. This document outlines how the NTS Key Establishment protocol of IETF RFC 8915 can be expanded for use as a PTP key management mechanism [Langer_et_al._2020] for immediate security processing complementing the exemplary GDOI proposal in the IEEE Std 1588-2019 [IEEE1588-2019]. As a key establishment server for NTP should be implemented stateless which is not necessary for PTP systems, suitable new NTS messages are to be defined in this document.

Though the key management for PTP is based on the NTS Key Establishment 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 (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 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 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:

- 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

- secured (end-to-end) PTP unicast communication between 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.

This document describes the structure and usage of these two approaches 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 4 depicts details of a TICKET TLV needed to transport encrypted security information in PTP unicast requests. The following Section 5 mentions specific parameters used in the PTP AUTHENTICATION TLV when working with the NTS4PTP key management system. Section 6 and Section 7 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. Figure 1 describes the principle sequence which can be used for PTP multicast as well as PTP unicast operation.

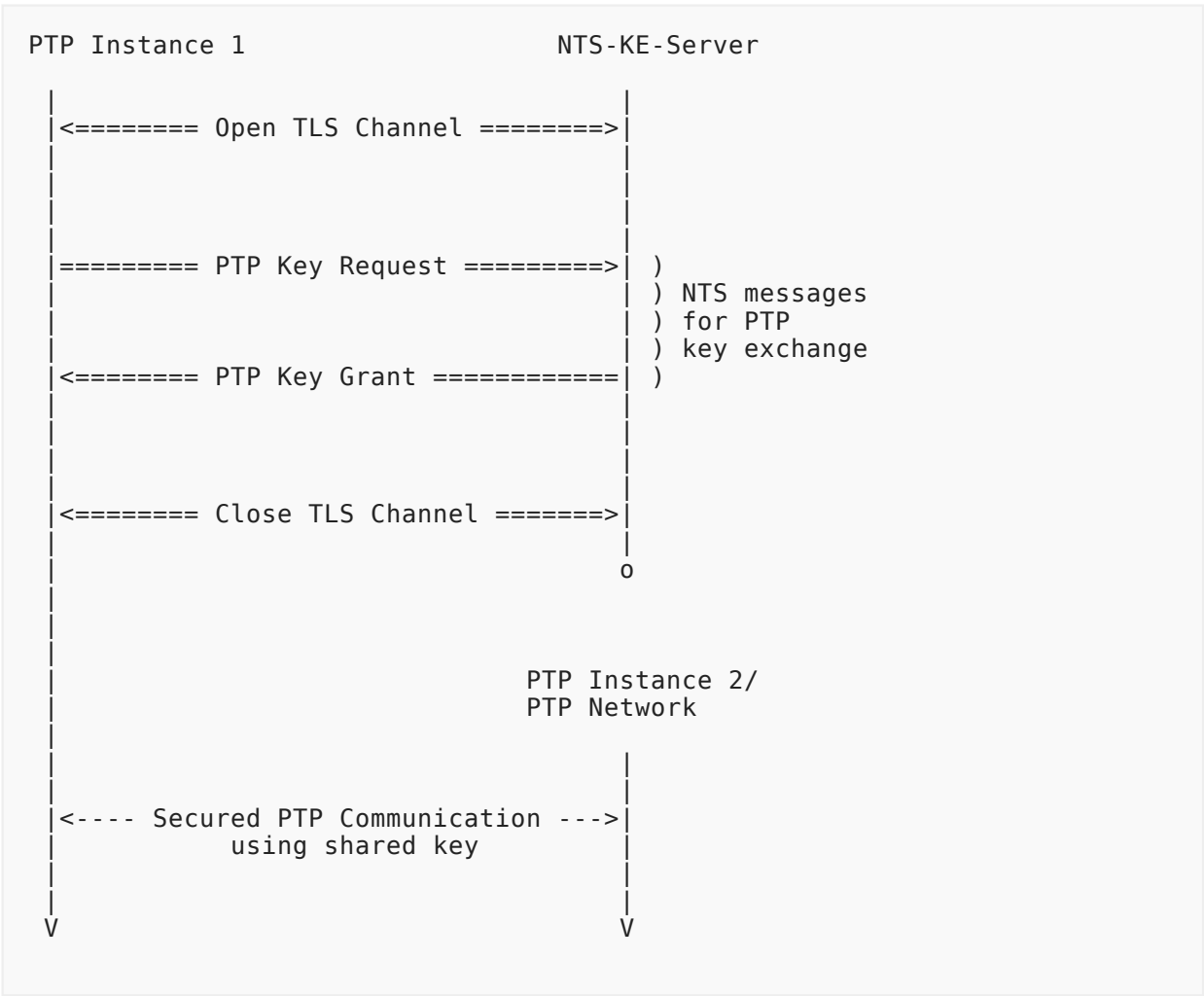

Figure 1: NTS Key distribution sequence

The client connects to the 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 chapter 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 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 KE server) using a digital X.509 certificate,
- verification of the TLS client (PTP instance) using its digital X.509 certificate and
- the encryption of the subsequent information exchange between the TLS communication partners.

TLS therefore enables peer authentication by certificates and provides authenticity, message integrity and confidentiality of following data transmitted over the TLS channel.

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). The NTS application of the KE server will respond with either a PTP Key Grant message (see Section 2.3.2), or a PTP Refusal message (see Section 2.3.3). All messages are constructed from specific records as described in Section 3.2.

When the Key Request message was responded with a PTP Key Grant or a PTP Refusal the TLS session will be closed with a close notify TLS message 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 can resume the TLS session by including a TLS session ID, allowing the PTP instance to skip the TLS version and algorithm negotiations. If resuming is used, a suitable lifetime for the TLS session key must be defined to not open the TLS connection for security threats.

As the TLS session provides authentication, but not authorization additional means has 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 and the ticket-based approach 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 KE server inside a secured TLS connection with a PTP Key Request message (see Section 2.3.1) as shown in Figure 2. The KE server answers with a PTP Key Grant message (see Section 2.3.2) with all the necessary data to join the group communication or with a PTP Refusal message (see Section 2.3.3) 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 KE server. The group number for both modes is primarily generated by a concatenation of the PTP domain number and the PTP profile (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.

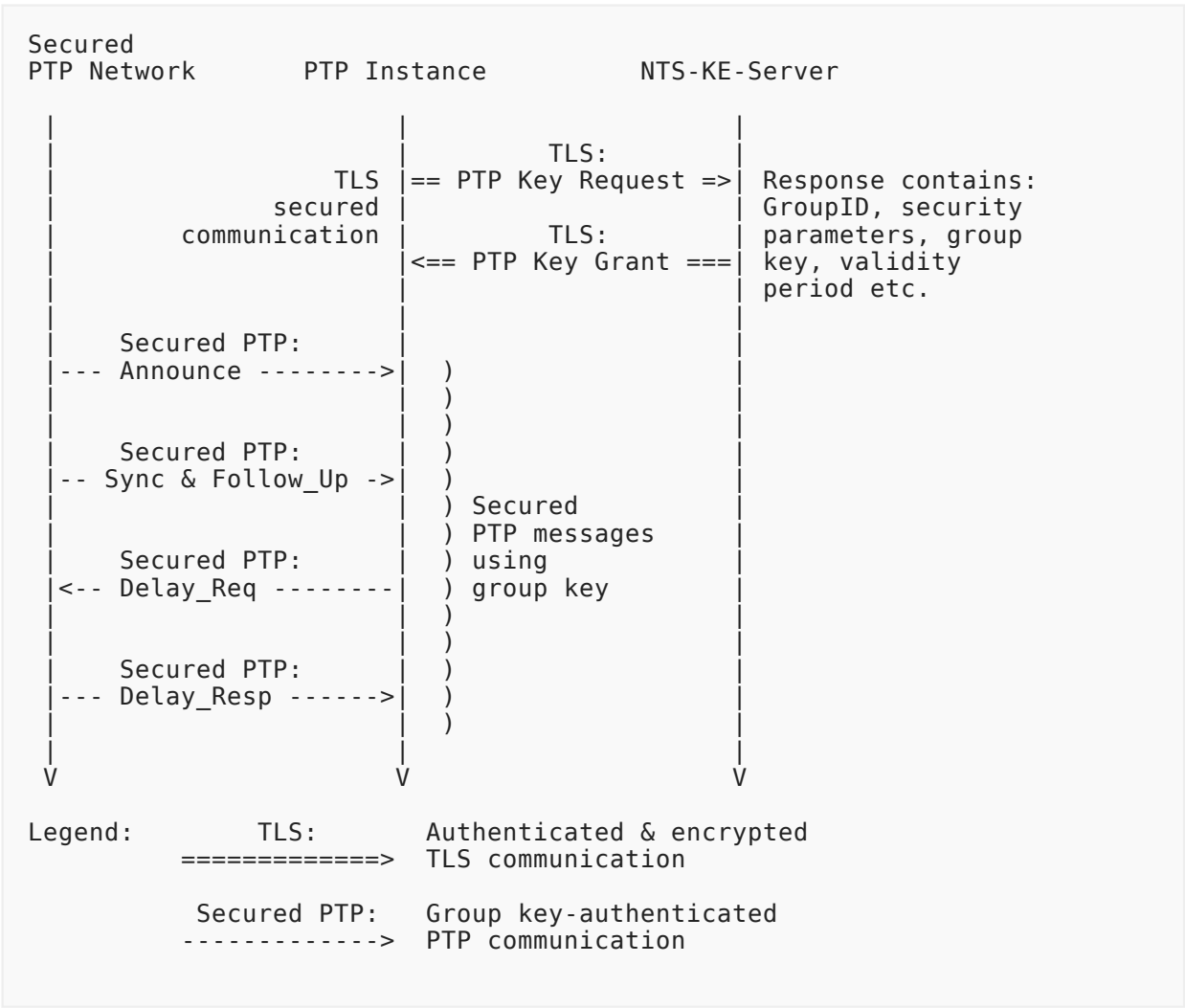

Figure 2: Message exchange for the group-based approach

This 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 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 key management for these modes works relatively simple and needs only the above mentioned three NTS messages: PTP Key Request, PTP Key Grant or PTP Refusal. The group number used for addressing is automatically derived from the configured attributes domain number and sdoID.

Additionally, besides multicast and hybrid mode, a (multicast) group of two PTP instances can be configured, practically implementing a special unicast communication.

The key management for these modes works relatively simple and needs only the above mentioned three NTS messages: PTP Key Request, PTP Key Grant or PTP Refusal. The group number used for addressing is automatically derived from the configured attributes PTP domain number and sdoId. For Go2, the attribute subGroup is additionally required.

2.1.2. NTS Message Exchange for the Ticket-based Approach

In (native) PTP unicast mode using unicast message negotiation ([IEEE1588-2019], 16.1) any potential instance (the grantor) which can be contacted by other PTP instances (the requesters) needs to register upfront with the KE server as depicted in Figure 3.

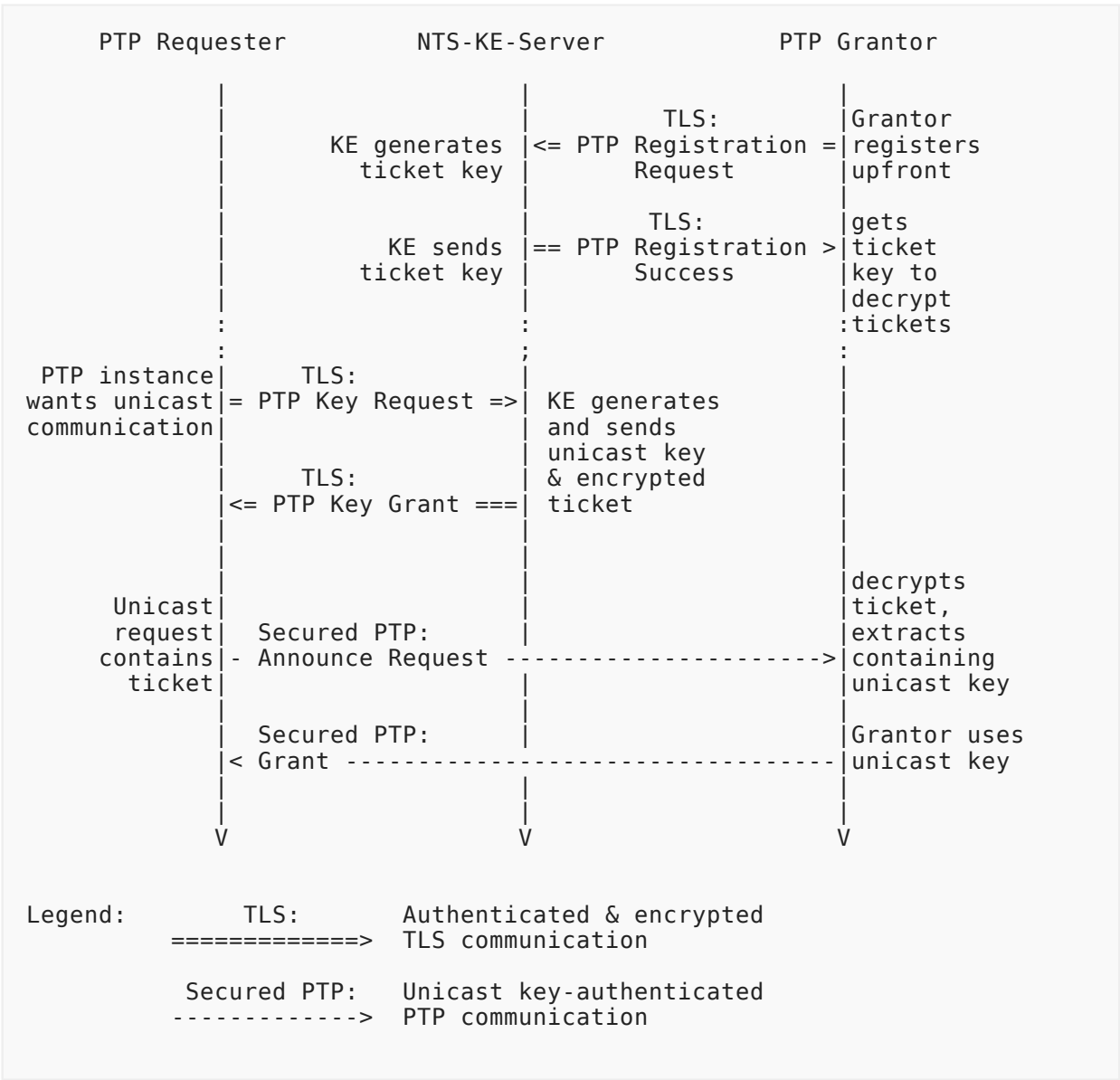

Figure 3: Message exchange for ticket-based unicast mode

(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 resp. 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.)

This registration is performed via a PTP Registration Request message (see Section 2.3.4). The KE server answers with a PTP Registration Success message (see Section 2.3.5) or a PTP Refusal message (see Section 2.3.3).

With the reception of the PTP Registration Success message the grantor holds a ticket key known only to the 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.

As with the group-based approach, a PTP instance (the requester) wanting to start a secured PTP unicast communication with a specific grantor contacts the KE server sending a PTP Key Request message (see Section 2.3.1) as shown in Figure 3 using the TLS-secured NTS Key Establishment protocol. The KE server answers with a PTP Key Grant message (see Section 2.3.2) with all the necessary data to begin the unicast communication with the desired partner or with a PTP Refusal message (see Section 2.3.3) if unicast communication with that instance is unavailable.

The PTP Key Grant message includes a unicast key to secure the PTP message exchange with the desired grantor. In addition, it contains the above mentioned encrypted ticket which the requester transmits in a special Ticket TLV (see Section 4) with the secured PTP message to the grantor. 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.

After the NTS key establishment messages for the PTP unicast mode have been exchanged the secured PTP communication can take place using the Security Association(s) communicated.

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 2.3.6) to the KE server, so requesters no longer receive PTP Key Grant messages for this grantor.

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 KE server, the use of certificates and topics concerning upfront configuration.

2.2.1. Key Update Process

All keys are equipped with parameters for a specific lifetime. Thereafter new key material has to be used. The value in the Lifetime record given by the KE server in the respective NTS messages is specified in seconds which denote the remaining time until the key expires and are decremented down to zero. So hard adjustments of the clock used have to be avoided. Therefore the use of a monotonic clock is recommended. Requests during the currently running lifetime will receive respectively adapted count values.

The receiving instances may concede a Grace Time in the range of, for example 5 - 10 seconds where an old key is still accepted to handle internal delays gracefully. The Grace Time may be defined in a PTP profile. Additionally, the KE server can optionally be configured to inform about a grace time value generally to be used.

New security parameters will be available after the Time until Update (TuU). The Time until Update given by the KE server is specified in seconds which are decremented down to zero. After that point in time until the end of the Lifetime of an associated key the PTP instances should connect to the KE server again, to receive new security parameters. The actual point in time, when a PTP instance asks for new data, should be selected randomly in the update period - the time after TuU was decremented to zero and before the Lifetime is counted down completely - to avoid peak load on the KE server. Figure 4 presents an example of the key update mechanism. A PTP instance sending a PTP Key Request to the KE server during the update period will receive the current security parameters (Current Parameters) as well as the security parameters of the following period (Next Parameters). As with the lifetime, requests during the currently running lifetime will receive respectively adapted count values for the current TuU.

Lifetime and Time until Update allow a cyclic rotation of security parameters during the running operation. This approach guarantees continuous secured PTP communication without interruption by key rotation.

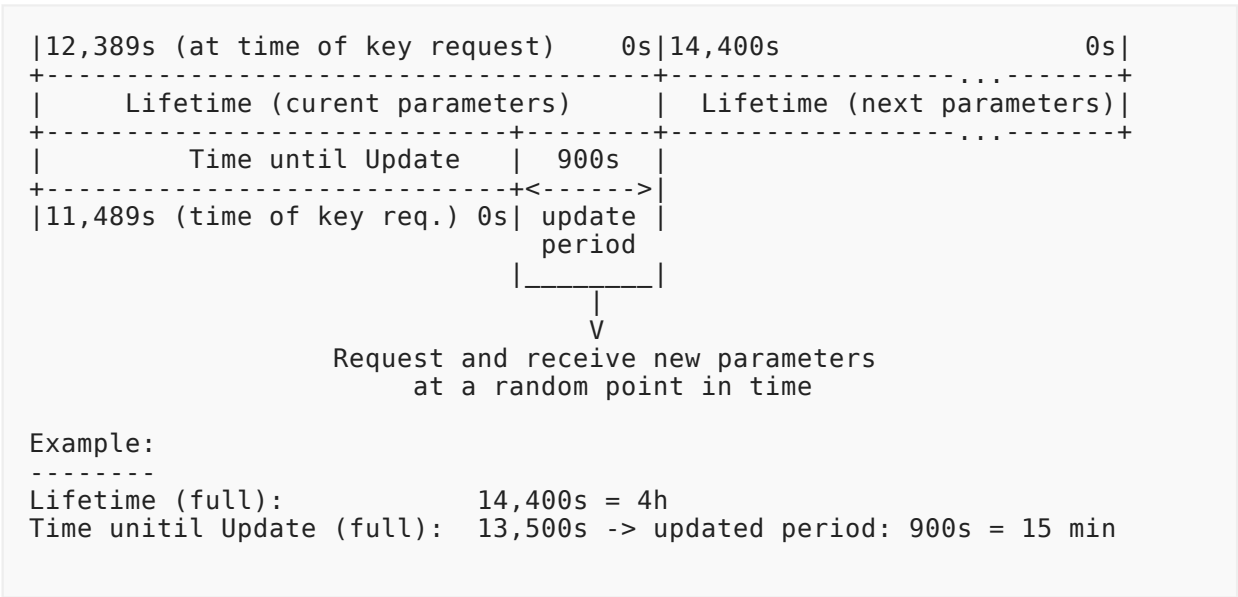

Figure 4: Example of the parameter rotation using Lifetime and Time until Update in group-based mode

The key rotation mechanism described also applies for the ticket-based approach. As there are two keys, the ticket key and the unicast key, some details need to be explained (see Figure 5). When the grantor registers with the KE server it receives the ticket key with the PTP Registration Success message together with the Lifetime and the respective Time until Update records. The lifetime parameters also apply to the ticket a requester would receive.

A requester wanting to communicate in unicast sends a PTP Key Request message with the particular parameters to the KE server. In the response it receives a specific unicast key with Lifetime and TuU as well as the encrypted ticket containing all the necessary security information for the grantor. The lifetime of the unicast key will end at the same point in time as the ticket key. Requests during the currently running lifetime of the ticket key will receive respectively adapted count values. The lifetime can be at most the remaining lifetime of the respective ticket key of the grantor.

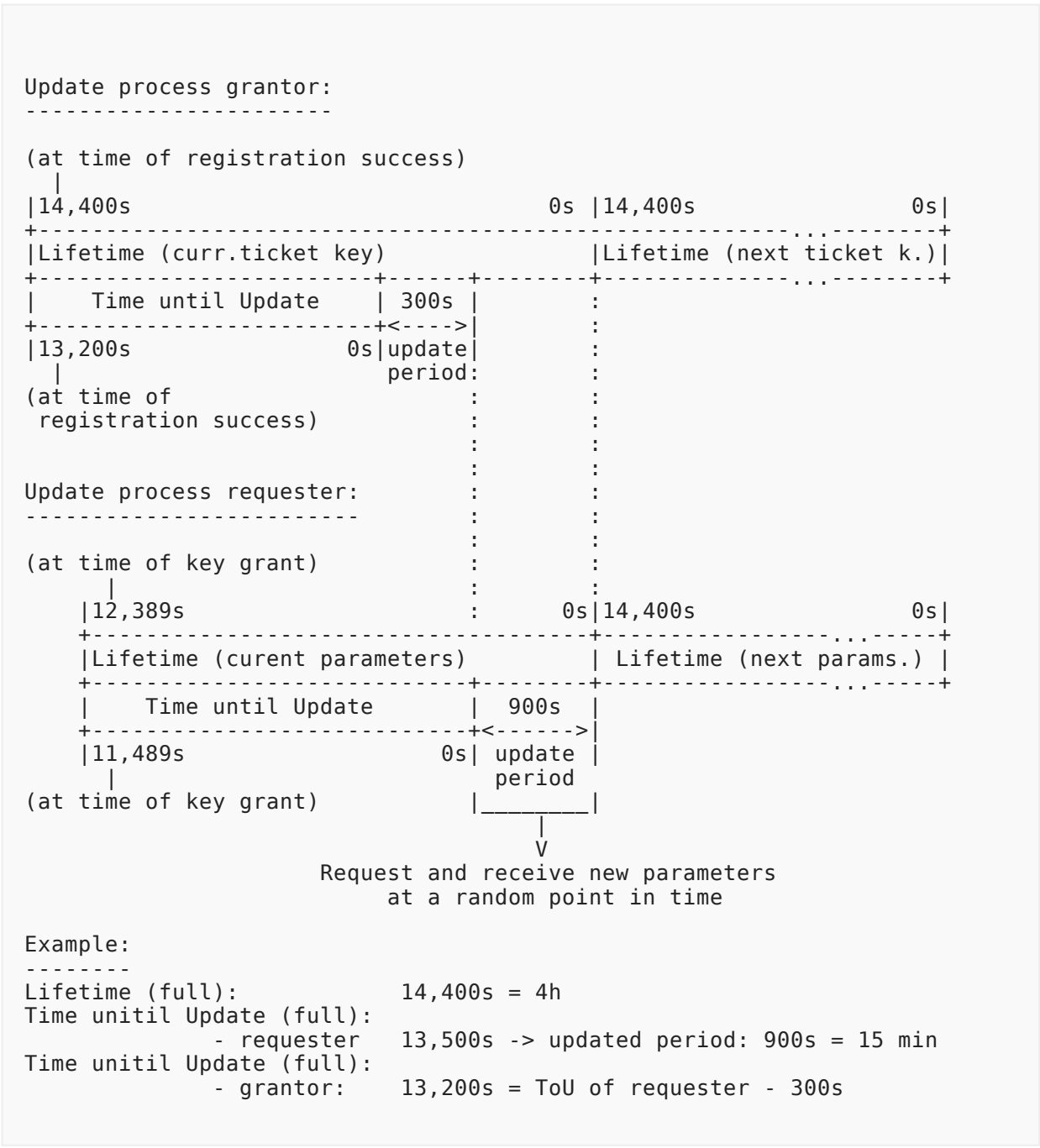

Figure 5: Example of the parameter rotation using Lifetime and Time until Update in ticket-based mode

The TuU of the ticket key will end earlier than the TuU of associated unicast keys. The grantor should re-register in its update period beginning after the Time until Update of the ticket key was decremented to zero and ending when an associated unicast key TuU is counted down. As the grantor does not know how long its update period lasts it should re-register immediately after its

TuU has ended. (A profile or a general configuration may fix the length of a grantors' update period. Then the grantor could re-register at a random point in time during its update period. Because masters register asynchronously, their re-registration will also be asynchronous. So typically, no peak load for the KE server will be generated.) Its update period is a mere timing buffer for cases where re-registration will not work instantly. The re-registration should be completed before any requester can start a PTP Key Request for ticket-based unicast mode. This guarantees the availability of a new ticket. When re-registering in its update period the grantor will receive together with the ticket key, etc., Lifetime and Time until Update of the current period as well as the parameters of the following period - similar to multicast keys. (A registration during the TuU period will supply only current data, not parameters of the following period. A late re-registration after the end of the current Lifetime will start a new period with respective full lifetime und update parameters.)

A requester needs to ask for a new unicast key and ticket at the KE server during the update period for uninterrupted unicast communication possibility or else at any later point in time. During the update period it will receive the Current Parameters as well as the Next Parameters. Embedded in the respective data, it will receive the ticket for the grantor including the encrypted ticket. Each ticket carries the same security information as the respective Current Parameters or Next Parameters data structure.

If a grantor does not have re-registered (in time or at all) when corresponding requesters try to get unicast keys, they will receive a PTP Refusal message.

If a grantor has revoked his registration with a PTP Registration Revoke message, requesters will receive a PTP Refusal message when trying to update for a new unicast key. No immediate key revoke mechanism exists. The grantor should not grant respective unicast requests until the revoked key expires.

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 3.3.2) respectively the AEAD algorithm utilized.

2.2.3. Time Information of the KE Server

As the KE server embeds time duration information in the respective messages, its local time should be sufficiently precise to a maximum 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], chapter 8.5 "Initial Verification of Server certificates" which applies to client 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) 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 KE server would be necessary: which domain, sdoId and/or subgroup attributes belong to which certificate.

As TLS is used to secure the NTS Key Establishment 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 KE server. This includes the Security Policies, i.e. which PTP messages are to be secured. 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 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 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 profile.

2.2.5.2. Key Lifetimes

Supplementary to the above mentioned SAs the desired key rotation periods, i.e. the lifetimes of keys resp. 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 Time until Update parameters need to be defined which (together with the lifetime) specify the relevant update period. Any particular Lifetime and Time until Update are configured as time spans counted in seconds and start at the same point in time.

2.2.5.3. Certificates

The network administrator has to supply each PTP instance and the 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 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 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 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 lies 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 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 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 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], chapter 8.5 "Initial Verification of Server certificates" which applies to client 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 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 Fig. 3 starts) of chapter 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]`."*

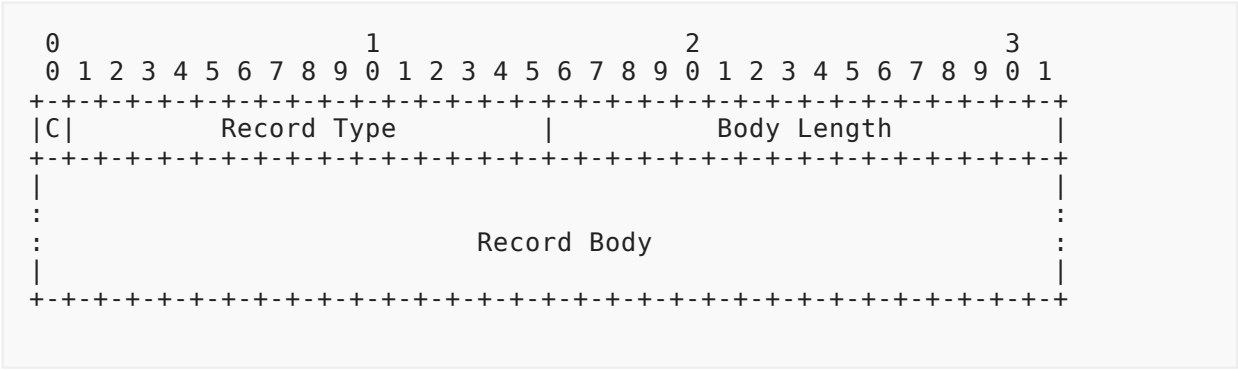

Figure 6: NTS-KE Record format

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 Parameter container.

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 Grant message (see Section 2.3.2)
- PTP Refusal message (see Section 2.3.3)
- PTP Registration Request message (see Section 2.3.4)
- PTP Registration Grant message (see Section 2.3.5)
- PTP Registration Revoke 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

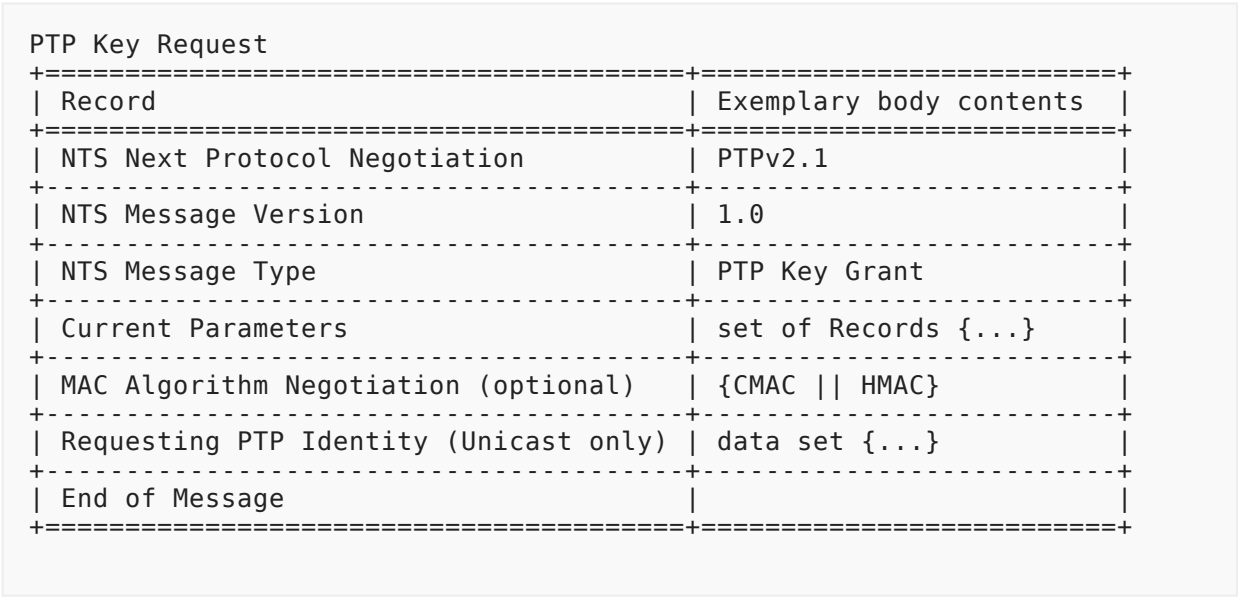

Figure 7: Structure of a PTP Key Request message

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. Currently, the following NTS Message Version record always contains 1.0. The next record characterizes the message type, in this case PTP Key Request. The 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.

If there is an option to choose from additional MAC algorithms, then an optional record follows presenting the supported algorithms from which the KE server may choose. In ticket-based unicast mode, the Requesting PTP Identity 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 Grant Message

Figure 8 shows the record structure of a PTP Key Grant 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. Currently, the following NTS Message Version record always contains 1.0. The next record characterizes the message type, in this case PTP Key Grant.

```
PTP Key Grant
+=======================================+============================+
| Record                                | Exemplary body contents    |
+=======================================+============================+
| NTS Next Protocol Negotiation         | PTPv2.1                    |
+---------------------------------------+----------------------------+
| NTS Message Version                   | 1.0                        |
+---------------------------------------+----------------------------+
| NTS Message Type                      | PTP Key Grant              |
+---------------------------------------+----------------------------+
| Current Parameters                    | set of Records {...}       |
+---------------------------------------+----------------------------+
| Next Parameters                       | set of Records {...}       |
+---------------------------------------+----------------------------+
| End of Message                        |                            |
+=======================================+============================+
```

Figure 8: Structure of a PTP Key Grant 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 see Section 3.2.3.

```
Current Parameters Container record (PTP Key Grant)
+================================+=====================================+
| Record                         | Exemplary body contents             |
+================================+========+============================+
| Security Policies              |{(PTPmsg1||SPP:1)||(PTPmsg2||SPP:)}|
+--------------------------------+-----------------------------------+
| Security Association           | data set for SPP:1 {...}          |
+--------------------------------+-----------------------------------+
| [Security Association]         | data set for SPP:2 {...}          |
+--------------------------------+-----------------------------------+
| Lifetime                       | 1560s (=0h 26min)                 |
+--------------------------------+-----------------------------------+
| Time until pdate               | 0s                                |
+--------------------------------+-----------------------------------+
| Grace Period (optional)        | 10 seconds                        |
+--------------------------------+-----------------------------------+
| Ticket Key ID (Unicast only)   | 156                               |
+--------------------------------+-----------------------------------+
| Ticket (Unicast only)          | data set {...}                    |
+================================+===================================+
```

Figure 9: Exemplary contents of a Current Parameters Container record of a PTP Key Grant message

If the request lies inside the update interval (i.e. TuU = 0, compare Figure 9), a Next Parameters Container record is appended giving all the security data needed in the upcoming validity period. Its structure follows the same composition as the Current Parameters record (in the ticked-based approach also including the Ticket Key ID record and the Ticket record). The messages always end with an End of Message record.

2.3.3. PTP Refusal Message

The message starts with the NTS Next Protocol Negotiation record which in this application always holds PTPv2.1. Currently, the following NTS Message Version record always contains 1.0. The next record characterizes the message type, in this case PTP Refusal, see Figure 10. The Error record contains information about the reason of refusal. The messages always end with an End of Message record.

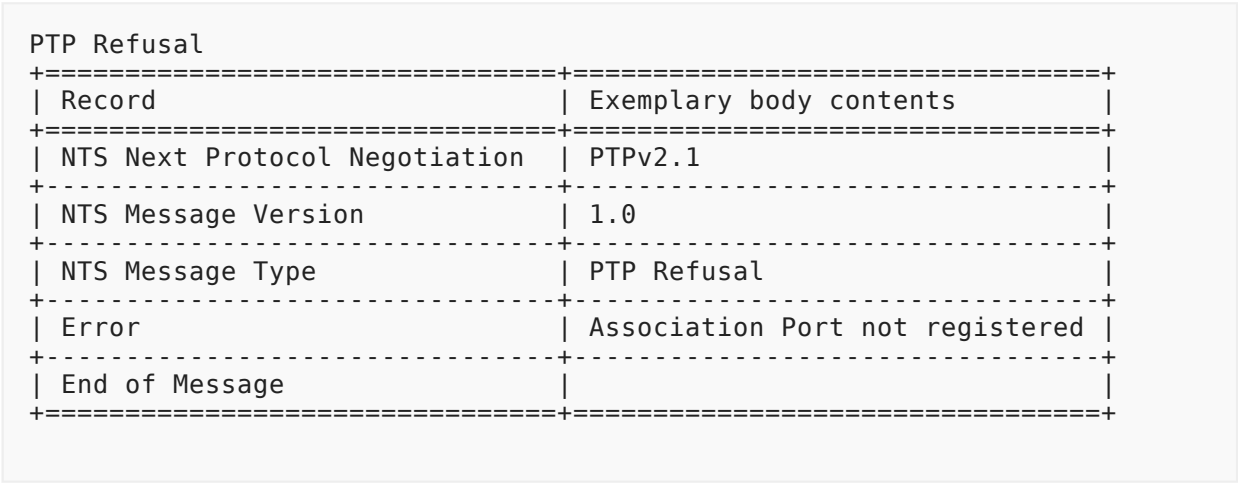

Figure 10: Structure of a PTP Refusal message

2.3.4. PTP Registration Request Message

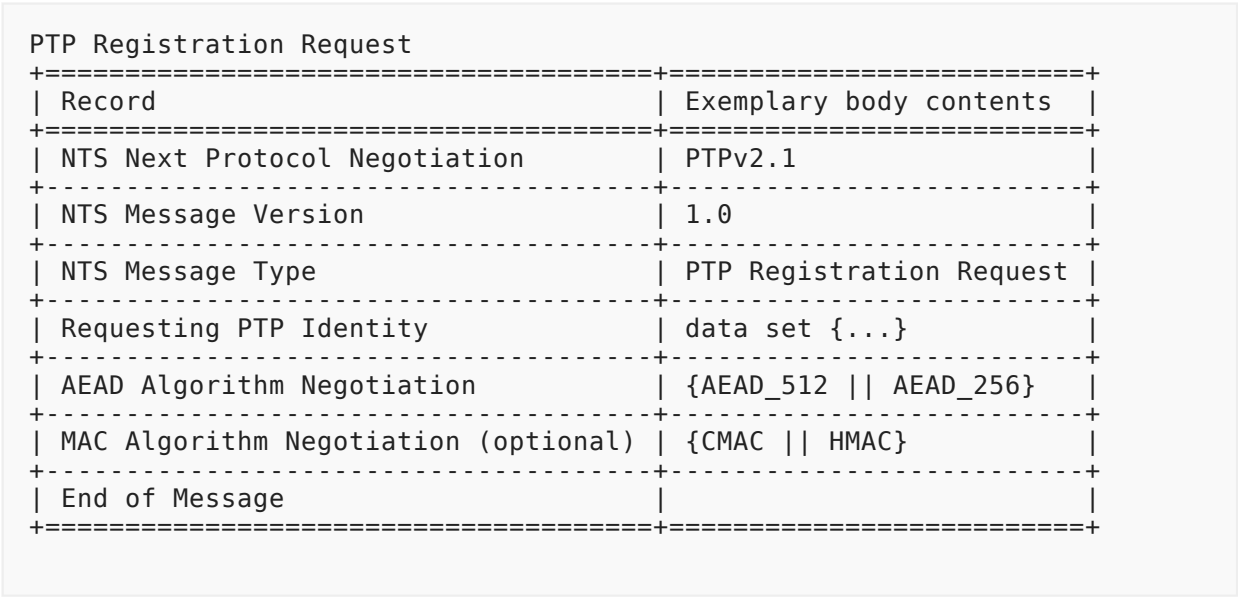

Figure 11: Structure of a PTP Registration Request message

The message starts with the NTS Next Protocol Negotiation record which in this application always holds PTPv2.1. Currently, the following NTS Message Version record always contains 1.0. The next record characterizes the message type, in this case PTP Registration Request, see Figure 11.

The Requesting PTP Identity record gives the addresses of the grantor requesting registration whereas the following AEAD Algorithm Negotiation record indicates which algorithms for encryption of the ticket the requester supports.

If there is an option to choose from additional MAC algorithms, then an optional record follows presenting all the grantor's supported algorithms from which the KE server may choose. The messages always end with an End of Message record.

2.3.5. PTP Registration Success Message

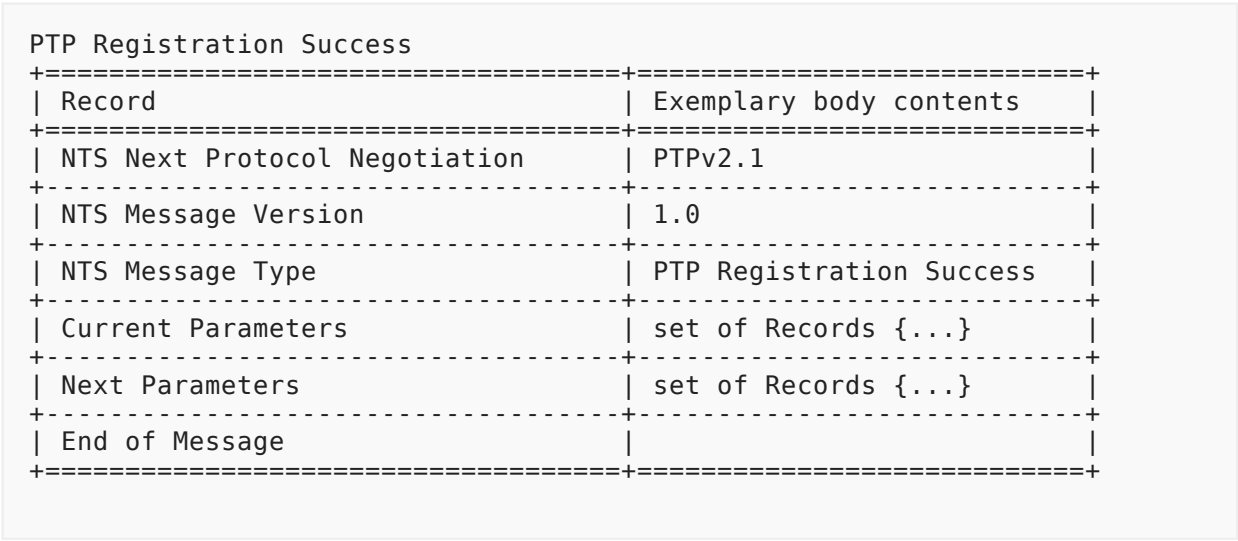

Figure 12: Structure of a PTP Registration Success message

The message starts with the NTS Next Protocol Negotiation record which in this application always holds PTPv2.1. Currently, the following NTS Message Version record always contains 1.0. The next record characterizes the message type, in this case PTP Registration Success, see Figure 12.

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 13 gives an example of data contained in that container as a response to PTP Registration Request. For more details on the records contained in the Current Parameters container see Section 3.2.3.

```
Current Parameters Container record (PTP Registration Success)
+==================================+========+=====================+
| Record                           | Exemplary body contents      |
+==================================+========+=====================+
| AEAD Algorithm Negotiation       | AEAD_CMAC_512                |
+----------------------------------+------------------------------+
| Lifetime                         | 2,460s (=0h 41min)           |
+----------------------------------+------------------------------+
| Time until pdate                 | 0s                           |
+----------------------------------+------------------------------+
| Ticket Key                       | {binary data}                |
+----------------------------------+------------------------------+
| Ticket Key ID                    | 278                          |
+----------------------------------+------------------------------+
| Grace Period (optional)          | 10 seconds                   |
+==================================+========+=====================+
```

Figure 13: Exemplary contents of a Current Parameters Container record of a PTP Registration Success message

If the registration request lies inside the update interval a Next Parameters Container record is appended giving all the security data needed in the upcoming validity period. Its structure follows the same composition as the Current Parameters record. The messages always end with an End of Message record.

2.3.6. PTP Registration Revoke Message

```
PTP Registration Revoke
+==================================+==============================+
| Record                           | Exemplary body contents      |
+==================================+==============================+
| NTS Next Protocol Negotiation    | PTPv2.1                      |
+----------------------------------+------------------------------+
| NTS Message Version              | 1.0                          |
+----------------------------------+------------------------------+
| NTS Message Type                 | PTP Registration Revoke      |
+----------------------------------+------------------------------+
| End of Message                   |                              |
+==================================+==============================+
```

Figure 14: Structure of a PTP Registration Revoke message

The message starts with the NTS Next Protocol Negotiation record which in this application always holds PTPv2.1. Currently, the following NTS Message Version record always contains 1.0. The next record characterizes the message type, in this case PTP Registration Revoke, see Figure 14. The messages always end with an End of Message record.

3. NTS Messages for PTP

This chapter 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], 7.6. Network Time Security Key Establishment Record Types Registry).

NTS Record Types	Description	Reference
0	End of Message	[RFC8915], section 4.1.1, this document, Section 3.2.4
1	NTS Next Protocol Negotiation	[RFC8915], section 4.1.2, this document, Section 3.2.12
2	Error	[RFC8915], section 4.1.3, this document, Section 3.2.5
3	Warning	[RFC8915], section 4.1.4
4	AEAD Algorithm Negotiation	[RFC8915], section 4.1.5, this document, Section 3.2.1
5	New Cookie for NTPv4 (not needed for PTP)	[RFC8915], section 4.1.6
6	NTPv4 Server Negotiation (not needed for PTP)	[RFC8915], section 4.1.7
7	NTPv4 Port Negotiation (not needed for PTP)	[RFC8915], section 4.1.8
8 - 1023	Reserved for NTP	
1024	Association Mode	This document, Section 3.2.2
1025	Current Parameters Container	This document, Section 3.2.3
1026	Grace Period	This document, Section 3.2.6
1027	Lifetime	This document, Section 3.2.7
1028	MAC Algorithm Negotiation	This document, Section 3.2.8

NTS Record Types	Description	Reference
1029	Next Parameters Container	This document, Section 3.2.9
1030	NTS Message Type	This document, Section 3.2.10
1031	NTS Message Version	This document, Section 3.2.11
1032	Requesting PTP Identity	This document, Section 3.2.13
1033	Security Association	This document, Section 3.2.14
1034	Security Policies	This document, Section 3.2.15
1035	Ticket	This document, Section 3.2.16
1036	Ticket Container	This document, Section 3.2.17
1037	Ticket Key	This document, Section 3.2.18
1038	Ticket Key ID	This document, Section 3.2.19
1039	Time until Update	This document, Section 3.2.20
1040 - 16383	Unassigned	
16384 - 32767	Reserved for Private or Experimental Use	[RFC8915]

Table 1: NTS Key Establishment 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 is mandatory the final record.

PTP Key Request

NTS Record Name	Comm. Type*	Use	Reference
NTS Next Protocol Negotiation	Multicast / Unicast	mand.	This document, Section 3.2.12
NTS Message Version	Multicast / Unicast	mand.	This document, Section 3.2.11
NTS Message Type	Multicast / Unicast	mand.	This document, Section 3.2.10
Association Mode	Multicast / Unicast	mand.	This document, Section 3.2.2
MAC Algorithm Negotiation	Unicast	opt.	This document, Section 3.2.8
Requesting PTP Identity	Unicast	mand.	This document, Section 3.2.13
End of Message	Multicast / Unicast	mand.	This document, Section 3.2.4

Table 2: Record structure of the PTP Key Request message

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

PTP Key Grant

NTS Record Name	Comm. Type	Use	Reference
NTS Next Protocol Negotiation	Multicast / Unicast	mand.	This document, Section 3.2.12
NTS Message Version	Multicast / Unicast	mand.	This document, Section 3.2.11
NTS Message Type	Multicast / Unicast	mand.	This document, Section 3.2.10
Current Parameters Container	Multicast / Unicast	mand.	This document, Section 3.2.3
Next Parameters Container	Multicast / Unicast	opt. (conditional)	This document, Section 3.2.9

NTS Record Name	Comm. Type	Use	Reference
End of Message	Multicast / Unicast	mand.	This document, Section 3.2.4

Table 3: Record structure of the PTP Key Grant message

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

NTS Record Name	Comm. Type	Use	Reference
Security Policies	Multicast / Unicast	mand.	This document, Section 3.2.15
Security Association (one or more)	Multicast / Unicast	mand.	This document, Section 3.2.14
Lifetime	Multicast / Unicast	mand.	This document, Section 3.2.7
Time until Update	Multicast / Unicast	mand.	This document, Section 3.2.20
Grace Period	Multicast / Unicast	opt.	This document, Section 3.2.6
Ticket Key ID	Unicast	mand.	This document, Section 3.2.19
Ticket	Unicast	mand.	This document, Section 3.2.16

Table 4: Record structure of the container records

The encrypted Ticket Container within the Ticket record also includes a set of records listed below:

NTS Record Name	Comm. Type	Use	Reference
Requesting PTP Identity	Unicast	mand.	This document, Section 3.2.13
Security Policies	Multicast / Unicast	mand.	This document, Section 3.2.15
Security Association (one or more)	Multicast / Unicast	mand.	This document, Section 3.2.14

NTS Record Name	Comm. Type	Use	Reference
Lifetime	Multicast / Unicast	mand.	This document, Section 3.2.7
Time until Update	Multicast / Unicast	mand.	This document, Section 3.2.20
Grace Period	Multicast / Unicast	opt.	This document, Section 3.2.6

Table 5: Record structure of the enctypted Ticket container record

PTP Refusal

NTS Record Name	Comm. Type	Use	Reference
NTS Next Protocol Negotiation	Multicast / Unicast	mand.	This document, Section 3.2.12
NTS Message Version	Multicast / Unicast	mand.	This document, Section 3.2.11
NTS Message Type	Multicast / Unicast	mand.	This document, Section 3.2.10
Error	Multicast / Unicast	mand.	This document, Section 3.2.5
End of Message	Multicast / Unicast	mand.	This document, Section 3.2.4

Table 6: Record structure of the PTP Refusal message

PTP Registration Request

NTS Record Name	Comm. Type	Use	Reference
NTS Next Protocol Negotiation	Multicast / Unicast	mand.	This document, Section 3.2.12
NTS Message Version	Multicast / Unicast	mand.	This document, Section 3.2.11
NTS Message Type	Multicast / Unicast	mand.	This document, Section 3.2.10

NTS Record Name	Comm. Type	Use	Reference
Requesting PTP Identity	Unicast	mand.	This document, Section 3.2.13
AEAD Algorithm Negotiation	Unicast	mand.	This document, Section 3.2.1
MAC Algorithm Negotiation	Unicast	opt.	This document, Section 3.2.8
End of Message	Multicast / Unicast	mand.	This document, Section 3.2.4

Table 7: Record structure of the PTP Registration Request message

PTP Registration Success

NTS Record Name	Comm. Type	Use	Reference
NTS Next Protocol Negotiation	Multicast / Unicast	mand.	This document, Section 3.2.12
NTS Message Version	Multicast / Unicast	mand.	This document, Section 3.2.11
NTS Message Type	Multicast / Unicast	mand.	This document, Section 3.2.10
Current Parameters Container	Multicast / Unicast	mand.	This document, Section 3.2.3
Next Parameters Container	Multicast / Unicast	mand. (conditional)	This document, Section 3.2.9
End of Message	Multicast / Unicast	mand.	This document, Section 3.2.4

Table 8: Record structure of the PTP Registration Success message

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

NTS Record Name	Comm. Type	Use	Reference
AEAD Algorithm Negotiation	Unicast	mand.	This document, Section 3.2.1
Lifetime	Multicast / Unicast	mand.	This document, Section 3.2.7
Time until Update	Multicast / Unicast	mand.	This document, Section 3.2.20

NTS Record Name	Comm. Type	Use	Reference
Grace Period	Multicast / Unicast	opt.	This document, Section 3.2.6
Ticket Key ID	Unicast	mand.	This document, Section 3.2.19
Ticket	Unicast	mand.	This document, Section 3.2.16

Table 9: Record structure of the container records in th PTP Regsitration Success message

PTP Registration Revoke

NTS Record Name	Comm. Type	Use	Reference
NTS Next Protocol Negotiation	Multicast / Unicast	mand.	This document, Section 3.2.12
NTS Message Version	Multicast / Unicast	mand.	This document, Section 3.2.11
NTS Message Type	Multicast / Unicast	mand.	This document, Section 3.2.10
End of Message	Multicast / Unicast	mand.	This document, Section 3.2.4

Table 10: Record structure of the PTP Registration Revoke message

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.

Global rules:

- The NTS Next Protocol Negotiation record MUST offer (at least) Protocol ID 1 for "PTPv2.1" (see Section 3.2.12).
- The NTS Message Version record MUST be v1.0.
- Note: Records must be used only in the mentioned messages. Not elsewhere.
- The notational conventions of Section 1 MUST be followed.

3.2.1. AEAD Algorithm Negotiation

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], 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.
- 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_385	Opt.	24	[RFC5297]
18	AEAD_AES_SIV_CMAC_512	Opt.	32	[RFC5297]
32 - 32767	Unassigned			
32768 - 65535	Reserved for Private or Experimental Use			[RFC5116]

Table 11: AEAD algorithms

- In a PTP Registration Request message, this record MUST be contained exactly once.
- In this 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 Success 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 MUST be present only during the update period.

- The KE server SHOULD choose the highest priority AEAD algorithm from the request message that grantor and KE server support.
- The KE server MAY ignore the priority and choose a different algorithm that grantor and KE server support.
- In a PTP Registration Success message, this record MUST contain exactly one AEAD algorithm.
- The selected algorithm MAY differ in the Current Parameters Container and Next Parameters Container records.

3.2.2. Association Mode

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 12: 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

Description	Assoc. Type Number	Association Mode	Association Value Content	Assoc. Value Octets
IPv6	2	Unicast	IPv6 address of the target port	16
802.3	3	Unicast	MAC address of the target port	6
PortIdentity	4	Unicast	PortIdentity of the target PTP entity	10

Table 13: 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, 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

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

Table 14: 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, 5.3.5, 7.5

3.2.3. Current Parameters Container

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 Grant 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 Policies	Multicast / Unicast	Mand.	This document, Section 3.2.15
Security Associations (one or more)	Multicast / Unicast	Mand.	This document, Section 3.2.14
Lifetime	Multicast / Unicast	Mand.	This document, Section 3.2.7
Time until Update	Multicast / Unicast	Mand.	This document, Section 3.2.20
Grace Period	Multicast / Unicast	Opt.	This document, Section 3.2.6
Ticket Key ID	Unicast	Mand.	This document, Section 3.2.19
Ticket	Unicast	Mand.	This document, Section 3.2.16

Table 15: Current Parameters Container for PTP Key Grant message

- The records Security Policies, Lifetime and Time until Update MUST be contained exactly once.
- The number of the Security Association records depends on the content of the Security Policies record (see Section 3.2.15).

- At least one Security Association record MUST be included.
- The Grace Period record is optional and MAY be absent.
- If it is present, it MUST be included exactly once.
- In order to establish a unicast connection with the PTP Key Grant message, the records Ticket Key ID and Ticket MUST be contained exactly once.
- If the requester wants to join a multicast group, the records Ticket Key ID and Ticket MUST NOT be included.

- In a PTP Registration Success 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	Mand.	This document, Section 3.2.1
Lifetime	Mand.	This document, Section 3.2.7
Time until Update	Mand.	This document, Section 3.2.20
Grace Period	Opt.	This document, Section 3.2.6
Ticket Key ID	Mand.	This document, Section 3.2.19
Ticket	Mand.	This document, Section 3.2.16

Table 16: Current Parameters Container for PTP Registration Success Message

- The records AEAD Algorithm Negotiation, Lifetime, Time until Update, Ticket Key ID and Ticket Key MUST be contained exactly once.
- The Grace Period record is optional and MAY be absent.
- If it is present, it MUST be included exactly once.

3.2.4. End of Message

The End of Message record is defined in IETF RFC8915 [RFC8915], 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 message.

- This record SHOULD NOT be included in the container records and MUST be ignored if present.
- See also: IETF RFC8915, 4.1.1

3.2.5. Error

The Error record is defined in IETF RFC8915 [RFC8915], 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	Requester not Authorized
4	Grantor not Registered
5 - 32767	Unassigned
32768 - 65535	Reserved for Private or Experimental Use

Table 17: Error Codes

Content and conditions:

- The record has a Record Type number of 2 and body length of two octets consisting of an unsigned 16-bit integer in network byte order, denoting an error code.
- The Critical Bit MUST be set.
- The Error code 3 "Requester not Authorized" is sent by the KE server if the requester is not authorized to join the desired multicast group.
- This Error code MUST NOT be included as a response to PTP Registration Request message.
- The Error code 4 "Grantor not Registered" is sent by the KE server when the requester wants to establish a unicast connection to a grantor that is not registered with the KE server.
- This Error code MUST NOT be included as a response to a PTP Key Request message.

3.2.6. Grace Period

The Grace Period determines the time period in which expired security parameters may still be accepted. It allows the verification of PTP messages, which have been secured with the previous key at the rotation time of the security parameters.

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.

- This value contains the transition time in seconds in which an expired key MAY still be accepted.
- A time of zero seconds is valid.
- If this optional record is absent, a default time of zero seconds is used unless a PTP profile defines something else.
- The Grace Period record MAY only appear as part of a PTP Key Grant or PTP Registration Success message.
- In a PTP Key Grant message, the Grace Period MAY be in the Current Parameters Container and Next Parameters Container records, as well as a part of the encrypted Ticket Container (if present).
- The Grace Period record MUST NOT appear more than once in each container or ticket.
- In a PTP Registration Success message, the Grace Period record MAY be present in the Current Parameters Container record as well as in the Next Parameters Container record.
- The Grace Period MUST NOT be included more than once in each of those container records.
- The Next Parameters Container MUST be present only during the update period.

3.2.7. Lifetime

This record specifies the lifetime of a defined set of parameters. The value contained in this record is counted down by the receiver of the NTS message every second. When the value reaches zero, the parameters associated with this record are considered to have expired.

Content and conditions:

- The record has a Record Type number of 1027 and the Critical Bit MAY be set.
- The record body consists of a 32-bit unsigned integer in network byte order, denoting the expiration time of specific parameters in seconds.
- The maximum value is set by the NTS-KE administrator or the PTP profile.
- In conjunction with a PTP unicast establishment, the Lifetime of the unicast key, the ticket key and registration lifetime of a grantor with the KE server MUST be identical.
- The Lifetime record MAY only appear as part of a PTP Key Grant or PTP Registration Success message.
- In both messages, the Next Parameters Container MUST be present only during the update period.
- In a PTP Key Grant message, the Lifetime record MUST be included exactly once in the Current Parameters Container and Next Parameters Container records, as well as in the encrypted Ticket Container (only present in a unicast PTP Key Grant message).
- In a PTP Registration Success message, the Lifetime MUST be included exactly once in both records, Current Parameters Container and Next Parameters Container.

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.

3.2.8. MAC Algorithm Negotiation

This optional 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 only used in unicast mode.

Content and conditions:

- The record has a Record Type number of 1028 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		

MAC Algorithm Types	MAC Algorithm	ICV Length (octets)	Reference
32768 - 65535	Reserved for Private or Experimental Use		

Table 18: MAC Algorithms

In PTP multicast 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.
- Unless this is specified by a PTP profile, the HMAC-SHA256-128 algorithm SHALL be used by default.

In PTP unicast mode:

- In a PTP Key Request message, this record MAY be contained if the requester wants a unicast connection 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.
- The default MAC algorithm (HMAC-SHA256-128) MAY be omitted in the record.
- In a PTP Registration Request message, this record MUST be present and the grantor MUST include all supported MAC algorithms in any order.
- The KE server selects the algorithm after receiving a PTP Key Request message in unicast mode.
- The KE server SHOULD choose the highest priority MAC algorithm from the request message that grantor and requester support.
- The 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 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, 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.

3.2.9. Next Parameters Container

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 1029 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 Grant and PTP Registration Success message, this record MUST be contained exactly once during the update period.
- Outside the update period, this record MUST NOT be included.
- In multicast mode, this record MAY also be missing if the requester is to be explicitly excluded from a multicast group after the security parameter rotation process by the KE server.
- The update period starts with the expiration of the Time until Update timer, which is stored in the Current Parameter Container record.
- In the PTP Key Grant and PTP Registration Success message, the expiration of the Lifetime marks the end of the update period.
- More details are described in Section 2.2.1.

3.2.10. NTS Message Type

This record enables the distinction between different NTS message types for PTP.

Content and conditions:

- The record has a Record Type number of 1030 and the Critical Bit MUST be set.
- The record body is a 16-bit unsigned integer in network byte order, denoting the type of the current NTS message for PTP.
- The message types are defined in the following table.
- More details about the messages are described in Section 2.3

NTS Message Type Number	NTS Message Name
0	PTP Key Request
1	PTP Key Grant

NTS Message Type Number	NTS Message Name
2	PTP Refusal
3	PTP Registration Request
4	PTP Registration Success
5	PTP Registration Revoke
6 - 32767	Unassigned
32768 - 65535	Reserved for Private or Experimental Use

Table 19: NTS message type numbers

3.2.11. NTS Message Version

This record enables the distinction between different NTS message versions for PTP. It provides the possibility to update or extend the NTS messages in future specifications.

Content and conditions:

- The record has a Record Type number of 1031 and the Critical Bit MUST be set.
- The record body consists of a tuple of two 8-bit unsigned integers in network byte order.
- The first octet represents the major version and the second octet the minor version.
  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 record body MUST match 0100h.

3.2.12. NTS Next Protocol Negotiation

The Next Protocol Negotiation record is defined in IETF RFC8915 [RFC8915], 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 requests messages for PTPv2.1, 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], 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 20: 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, NTS4NTP 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.

3.2.13. Requesting PTP Identity

This record allows the KE server to associate an NTS unicast request of a requester with a registered grantor based on their address or identifier (e.g.: IP address or PortIdentity). Furthermore, this record allows the grantor to verify the origin of a secured PTP message that is currently transmitting a ticket.

Content and conditions:

- The record has a Record Type number of 1032 and the Critical Bit MAY be set.
- The record body consists of a set of Association Types together with their respective Association Values.

Field	Octets	Offset
Association Type 1	2	0

Field	Octets	Offset
Association Value 1	A1	2
Association Type 2	2	A1+2
Association Value 2	A2	A1+4
Association Type n	A2	A1+A2+4
Association Value n	An	A1+A2+6

Table 21: Requesting PTP identity list

- Structure and values are based on the contents defined in Section 3.2.2 of this document.
  - Therefore, the Association Type is a 16-bit unsigned integer.
  - The length and content of Association Value depends on the value of Association Type.
  - All bytes are stored in network byte order and the rules in Section 3.2.2 MUST be followed.
- A Requesting PTP Identity record MUST contain at least one association tuple (type + value).
- This record can contain several association tuples in any order.
- It MUST NOT contain more than one association tuple of the same type.
- In a PTP Key Request message, this record MUST be contained exactly once in the unicast mode, which depends on the content of the Association Mode record of this message.
- In this case the Requesting PTP Identity record MUST contain exactly one association tuple.
- This association tuple MUST contain one identification feature of the PTP requestor (IPv4, IPv6, 802.3 or PortIdentity).
- The association tuple MUST NOT contain the Group association type 0.
- In a PTP Key Grant message, this record MUST be contained exactly once in the encrypted Ticket Container.
- This record MUST contain exactly one association tuple.
- The record body MUST be identical to the Requesting PTP Identity record of the related PTP Key Request message.
- Therefore, the association tuple MUST NOT contain the Group association type 0.
- In a PTP Registration Request message, this record MUST be included exactly once.
- The grantor SHOULD add the following association tuples as far as they are available: IPv4, IPv6, 802.3 and PortIdentity.
- The grantor MUST NOT include the Group association type 0.
- This allows a requester to be assigned to a grantor, regardless of whether the requester specifies IPv4, IPv6, 802.3 or the PortIdentity of the grantor in its PTP Key Request message.

3.2.14. Security Association

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, 16.14.3). Static values and flags, such as the secParamIndicator, are described in more detail in Section 5.

Content and conditions:

- The record has a Record Type number of 1033 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
Key	K	9

Table 22: Security Association record

- In a PTP Key Grant message, the Security Association record MUST be included at least once in the Current Parameters Container record and the Next Parameters Container record.
- In unicast mode, the Security Association record MUST be included at least once in the encrypted Ticket Container as well.
- The Next Parameters Container record MUST be present only during the update period.
- The Ticket record MUST be present in unicast mode and MUST NOT be present in multicast mode.
- The number of Security Association records in the respective container or Ticket Container depends on the content of the associated Security Policies (see also Section 3.2.15).

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 KE server (see Section 3.3.2).

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 3.2.8.

- 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 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 3.3).

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 (multicast mode) or a unicast key (unicast mode).

3.2.15. Security Policies

This record contains the information "which" PTP message types must be secured.

Content and conditions:

- The record has a Record Type number of 1034 and the Critical Bit MAY be set.
- The record body contains a sequence of tuples in network byte order:
  Record body = **{Security Policies** = **{tuple 1** || **tuple 2** || **tuple 3** || **tuple n}}**
- Each tuple has a length of 2 octets and consists of a sequence of a PTP Message Type and a Security Parameter Pointer.

Field	Octets	Offset
PTP Message Type	1	0
Security Parameter pointers	1	1

Table 23: Security Policy tuple

- The PTP Message Type is an 8-bit unsigned integer.
- The most significant 4 bits are zero-padded and the least significant 4 bits are the PTP message type:

Structure of PTP Message Type (see also [IEEE1588-2019], 13.3.2.3, table 36):

Bits 7 - 4	Bits 3 - 0
Zero Padding	PTP Message type

Table 24: PTP Message Type

- The Security Parameter Pointer (SPP) is an 8-bit unsigned integer in the closed range 0 to 255.
- The record body MUST contain at least one tuple.
- A tuple associates a PTP message type with an SPP.
- Every PTP message type that is mentioned in the Security Policies record MUST be secured.
- Thus, a PTP message type that is not included in this record MUST NOT contain an AUTHENTICATION TLV and will not be secured.
- Multiple tuples with the same PTP message type MUST NOT be included.
- Multiple tuples MAY use the same SPP to use a shared security association or an individual one.
- For the number of contained and different SPPs in the Security Policies record, the same number of security associations MUST be created.
- The number of security associations determines the number of Security Associations records in the respective container record (e.g. Current Parameters Container).
- In a PTP Key Grant message, this record MUST be included exactly once each in the Current Parameters Container record, the Next Parameters Container record as well as the encrypted Ticket Container record.
- The Next Parameters Container record MUST be present only during the update period.
- The Ticket record MUST be present in unicast mode and MUST NOT be present in multicast mode.

3.2.16. Ticket

This record contains the parameters of the selected AEAD algorithm, as well as an encrypted Ticket Container record. The encrypted record contains all the necessary security parameters that the grantor needs for a secured PTP unicast connection to the requester. The ticket container 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 ticket container.

Content and conditions:

- The record has a Record Type number of 1035 and the Critical Bit MAY be set.
- The record body consists of several data fields and MUST be formatted as follows.

Field	Octets	Offset
Nonce Length	2	0
Nonce	N	2

Field	Octets	Offset
Encrypted Ticket Container Length	2	N+2
Encrypted Ticket Container	C	N+4

Table 25: Structure of a Ticket record

- In a PTP Key Grant message, this record MUST be included exactly once each in the Current Parameters Container record and the Next Parameters Container record if the requester wants a unicast communication to a specific grantor.
- The Next Parameters Container record MUST be present only during the update period.

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 3.3.1.

Encrypted Ticket Container Length

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

Encrypted Ticket Container

- This field contains the output of the AEAD operation ("Ciphertext") after the encryption process of the respective Ticket Container record.
- The plaintext of this field is described in Section 3.2.17.
- More details about the AEAD process and the required input data are described in Section 3.3.1.

3.2.17. Ticket Container

This record is a simple container that can carry an arbitrary number of NTS records. It contains all relevant security parameters that a grantor needs for a secured unicast connection. 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. The Ticket Container record serves as input parameter for the AEAD operation (see Section 3.2.1) and is transmitted encrypted within the Ticket record (see Section 3.2.16).

Content and conditions:

- The record has a Record Type number of 1036 and the Critical Bit MAY be set.
- The record body is defined as a set of records and MAY contain the following records.

NTS Record Name	Use	Reference
Requesting PTP Identity	mand.	This document, Section 3.2.13
Security Policies	mand.	This document, Section 3.2.15
Security Association (one or more)	mand.	This document, Section 3.2.14
Lifetime	mand.	This document, Section 3.2.7
Time until Update	mand.	This document, Section 3.2.20
Grace Period	opt. (conditional)	This document, Section 3.2.6

Table 26: Structure of a Ticket Container

- The records Requesting PTP Identity, Security Policies, Lifetime and Time until Update MUST be contained exactly once.
- The number of the Security Association records depends on the content of the Security Policies record (see Section 3.2.15).
- All records within this Ticket Container (except Requesting PTP Identity) MUST be identical to the records of the respective Current Parameter Container.
- All records within this Ticket Container (except Requesting PTP Identity) MUST be identical to the records of the respective Next Parameter Container.
- The presence of the Grace Period record also depends on the respective Current/Next Parameter container.
- If a Grace Period record is present in the Current/Next Parameter container, it MUST also be present in the respective Ticket Container.
- If it is not present, it MUST NOT be included in the Ticket Container.

3.2.18. Ticket Key

This record contains the ticket key, which together with an AEAD algorithm is used to encrypt and decrypt the ticket.

Content and conditions:

- The record has a Record Type number of 1037 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 requirement of the associated AEAD algorithm.

- In a PTP Registration Success 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.

3.2.19. Ticket Key ID

The Ticket Key ID record is a unique identifier that allows a grantor to identify the associated ticket key.

Content and conditions:

- The record has a Record Type number of 1038 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 Grant 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 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 Success 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 unicast mode and MUST NOT be present in multicast mode.

3.2.20. Time until Update

The Time until Update (TuU) record specifies the point in time at which new security parameters are available. The value contained in this record is counted down by the receiver of the NTS message every second. When the value reaches zero, the update period begins and NTS response messages typically contain the Next Parameter Container record for a certain period of time (see also Section 2.2.1).

Content and conditions:

- The record has a Record Type number of 1039 and the Critical Bit MAY be set.
- The record body consists of a 32-bit unsigned integer in network byte order, denoting the begin of the update period in seconds.
- The value in the TuU MUST be less than the value in the associated Lifetime record (in the same container or ticket).

- If the value in the TuU is greater than zero in the Current Parameter Container, the corresponding message MUST NOT contain a Next Parameters Container.
- If the value in the TuU is zero in the Current Parameters Container, the corresponding NTS message MAY contain the Next Parameters Container record.
- The Time until Update record MAY only appear as part of a PTP Key Grant or PTP Registration Success message.
- In both messages, the Next Parameters Container MUST be present only during the update period.
- In a PTP Key Grant message, the Time until Update record MUST be included exactly once each in the Current Parameters Container and Next Parameters Container records, as well as in the encrypted Ticket Container (only present in a unicast PTP Key Grant message).
- In a PTP Registration Success message, the Time until Update MUST be included exactly once each in the Current Parameters Container and Next Parameters Container records.
- In both messages, the Next Parameters Container record MUST be present only during the update period.

Notes:

- Requests during the currently running lifetime will receive respectively adapted count values for Time until Update.
- During the update period the value for TuU in the Current Parameters Container will be zero.

3.3. Additional Mechanisms

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

3.3.1. 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 Ticket Container record, this parameter MUST remain empty.
- Nonce:
  - Corresponds to the value from the Nonce field in the Ticket (Section 3.2.16).
  - 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 Ticket Container 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 Ticket Container record with all records inside.
- This is also the output of the AEAD operation after the decryption process.

- Ciphertext:
  - Corresponds to the value from the Encrypted Ticket Container field in the Ticket (Section 3.2.16).
  - The Ciphertext is the output of the AEAD operation after the encryption process.
  - This is also the input parameter for the AEAD decryption operation.

3.3.2. 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 B.
- 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 Parameter). 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 Pseudorandom 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 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 Parameter).

4. 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 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 registered port. With the reception of the key grant the requester obtains the unicast key and the Ticket record containing the encrypted ticket container (see Section 2.1.2 and Section 3.2.16). The ticket container (see Section 3.2.17) includes the identification of the requester, the SAs along with the unicast key as well as the Lifetime/Time until Update data.

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 container 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 container 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 Requesting PTP Identity record in the Ticket Container and/or the ICV in the AUTHENTICATION TLV is not identical to the generated ICV by the grantor, then the unicast request message shall be denied.

The TICKET TLV structure is given in Table 27 below.

Field	Octets	Offset
tlvType	2	0
lengthField	2	2

Field	Octets	Offset
Ticket record	T	4

Table 27: Structure of the TICKET TLV

To comply with the TLV structure of IEEE Std 1588-2019 ([IEEE1588-2019], 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 3.2.16) containing the encrypted Ticket container. Eventually it may be necessary to define the Ticket TLV externally to IEEE 1588 SA. Then the structure should follow IEEE Std 1588-2019 ([IEEE1588-2019], 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

To transport the TICKET TLV with the Ticket container embedded via the PTP unicast negotiation message two possible solutions exist:

a. The TICKET TLV can be added to the PTP message preceding the AUTHENTICATION TLV as shown in Figure 48 of IEEE Std 1588-2019 ([IEEE1588-2019], 16.14.1.1). For this solution, a completely new TICKET TLV for IEEE Std 1588-2019 needs to be defined.
b. In an alternative solution the TICKET TLV is send embedded in the RES field of the AUTHENTICATION TLV as shown in Figure 49 of IEEE Std 1588-2019 ([IEEE1588-2019], 16.14.3). In this case the RP flag in the secParamIndicator must be set. As at the moment the use of the RES field is not permitted and the structure of the RES field is limited to UInteger (see [IEEE1588-2019], 16.14.3.8) the new usage needs to be defined:
"16.14.3.8 RES (UInteger R): This field is optional. If present, it shall have a data type of UInteger with a length of R octets. For this edition, the value of RP in the secParamIndicator field shall be FALSE and the value of RP shall be 0."

Which solution is chosen is a political question, not a technical one and needs to be discussed in the IEEE 1588 SA. The same applies to the format of the TICKET TLV (standard TLV or organization extension TLV).

5. 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	mand.	TLV Type
lengthField	mand.	TLV Length Information
SPP	mand.	Security Parameter Pointer
secParamIndicator	mand.	Security Parameter Indicator
keyID	mand.	Key Identifier or Current Key Disclosure Interval, depending on verification scheme
disclosedKey	opt.	Disclosed key from previous interval
sequenceNo	opt.	Sequence number
RES	opt.	Reserved
ICV	mand.	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 KE server in the PTP Key Grant Message

The optional disclosedKey, sequenceNo, and RES (see discussion in chapter 3) fields are omitted. So all of the flags in the SecParamIndicator are FALSE.

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

6. IANA Considerations

Considerations should be made ...

...

7. Security Considerations

...

8. Acknowledgements

The authors would like to thank ...

9. References

9.1. 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>.

9.2. 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>.

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