Network Time Protocol
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

M. Langer
R. Bermbach
Ostfalia University

February 22, 2021

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

draft-langer-ntp-nts-for-ptp-00

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 August 26, 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 (IETF RFC 8915). Its NTS Key Establishment sub-protocol is secured by the Transport Layer Security (TLS 1.3, IETF RFC 8446) 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 [B58] for immediate security processing complementing the exemplary GDOI proposal in the IEEE Std 1588-2019. 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.

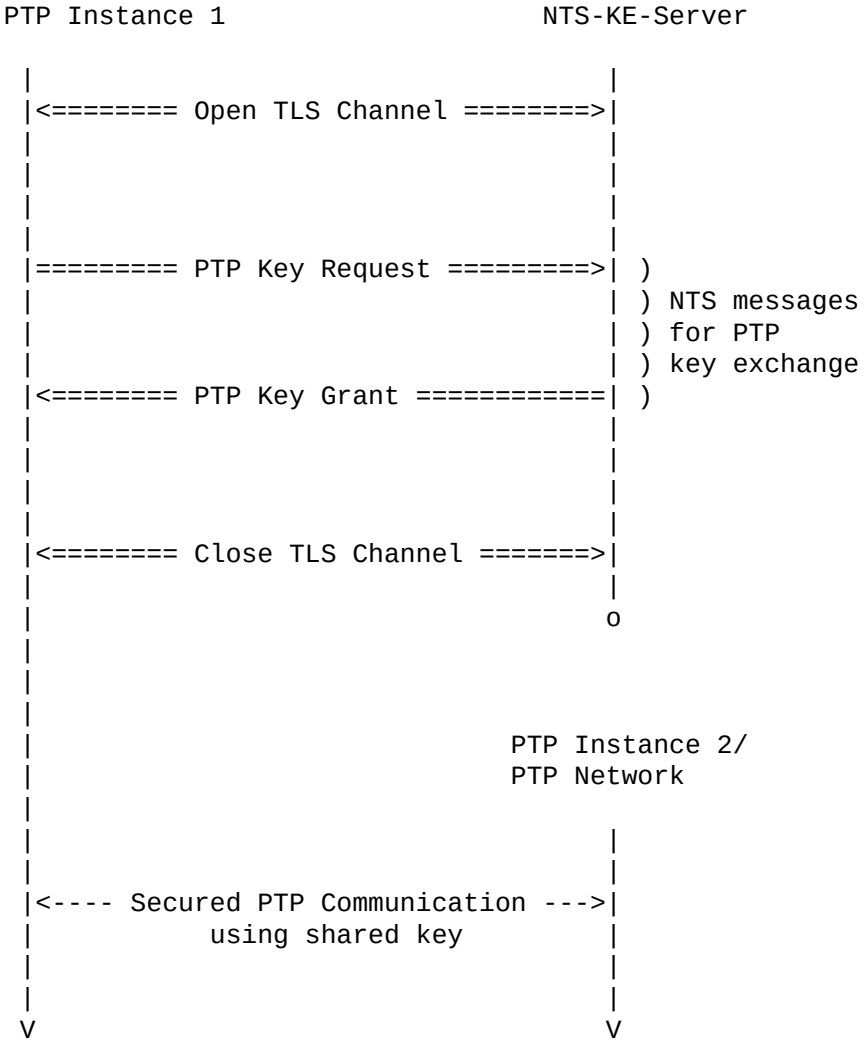


Figure 1: NTS Key distribution sequence

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.

Implementations must conform to the rules stated in chapter 3 "TLS Profile for Network Time Security" of IETF RFC 8915".

"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.

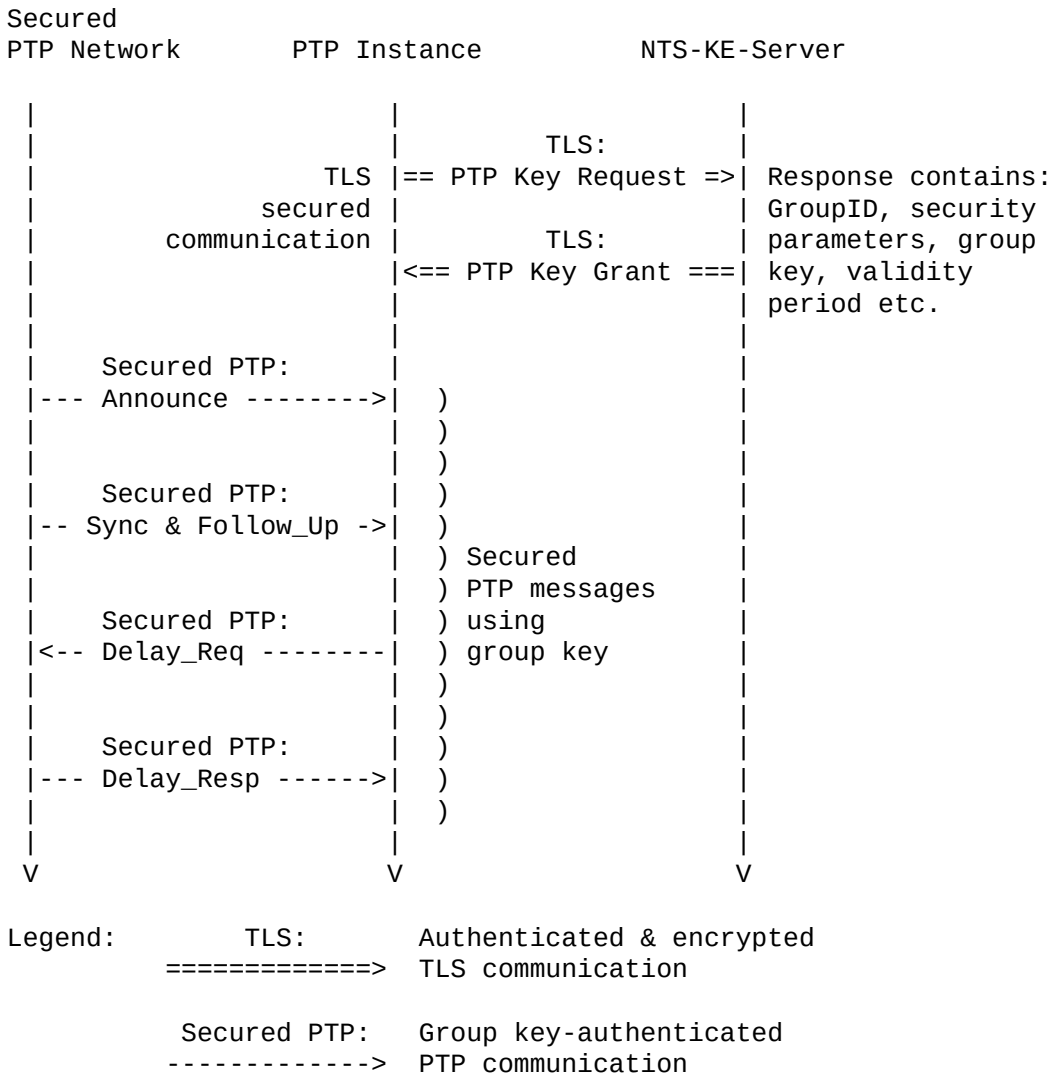


Figure 2: Message exchange for the group-based approach

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 (IEEE Std 1588-2019, 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.

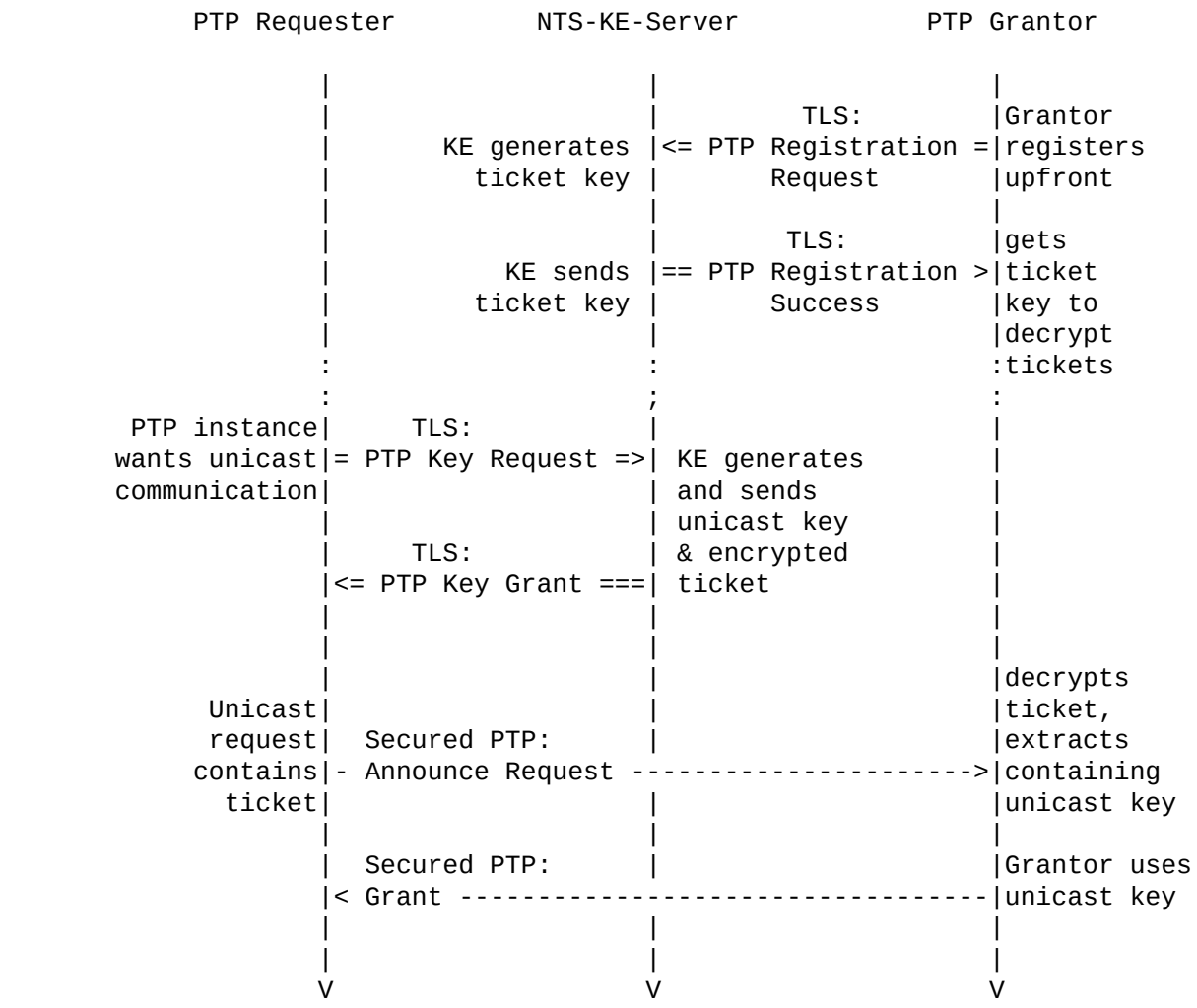


Figure 3: Message exchange for ticket-based unicast mode

(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.

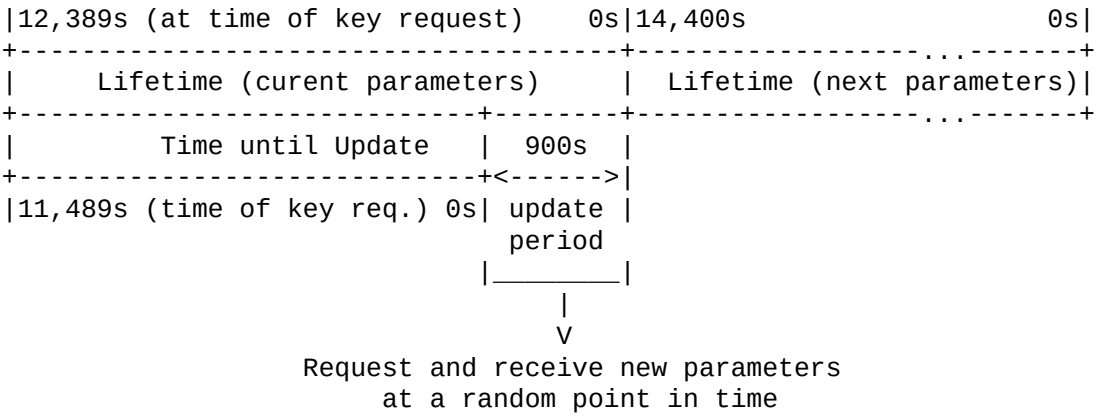
2.2.1. Key Update Process

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.



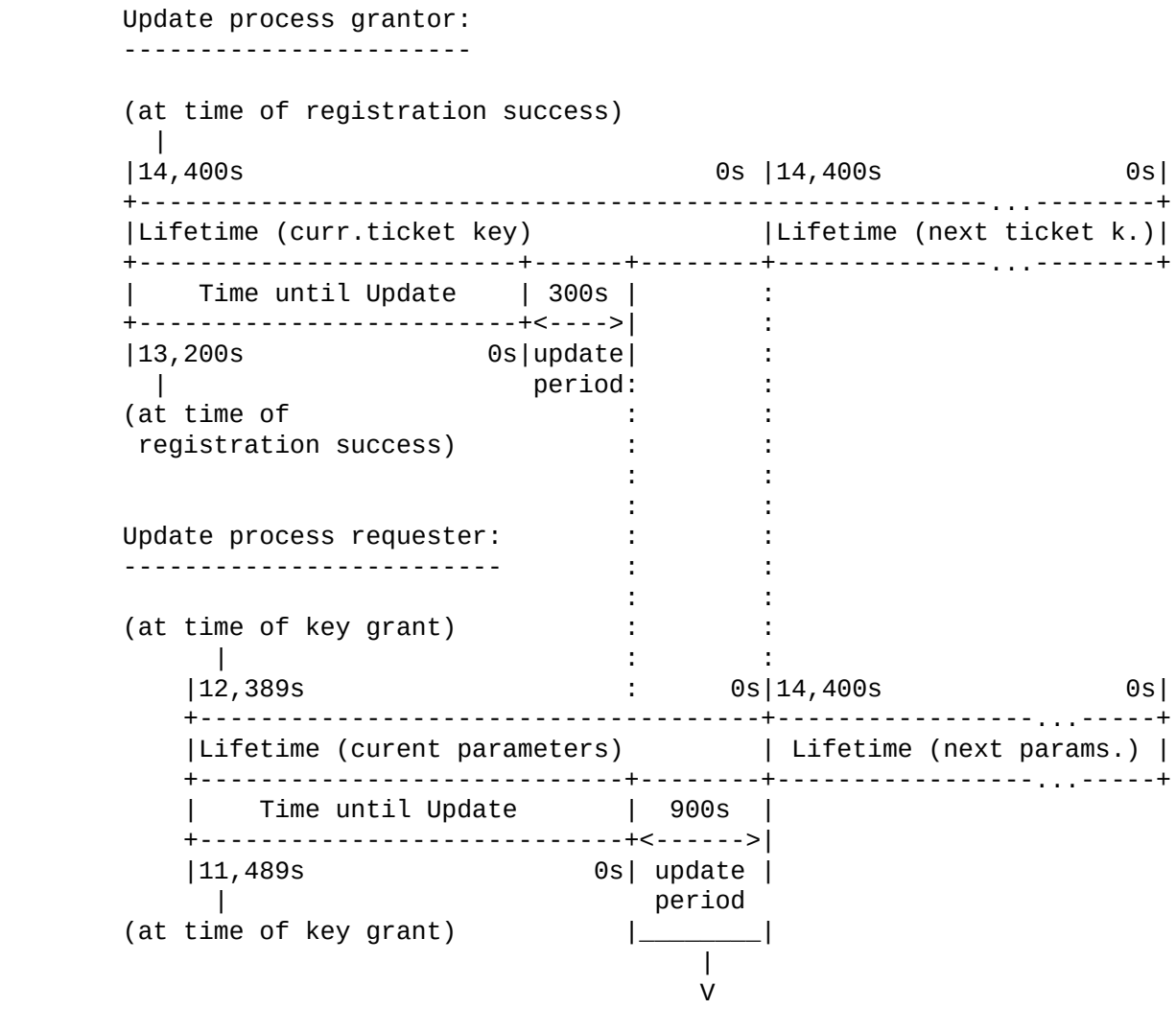
Example:

Lifetime (full): 14,400s = 4h
Time unitil Update (full): 13,500s -> updated period: 900s = 15 min

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.



Example:

Lifetime (full): 14,400s = 4h
Time unitil Update (full):
- requester 13,500s -> updated period: 900s = 15 min
Time unitil Update (full):
- grantor: 13,200s = ToU of requester - 300s

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, 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, 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:

"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[RFC 7301]), 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.

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]‘."

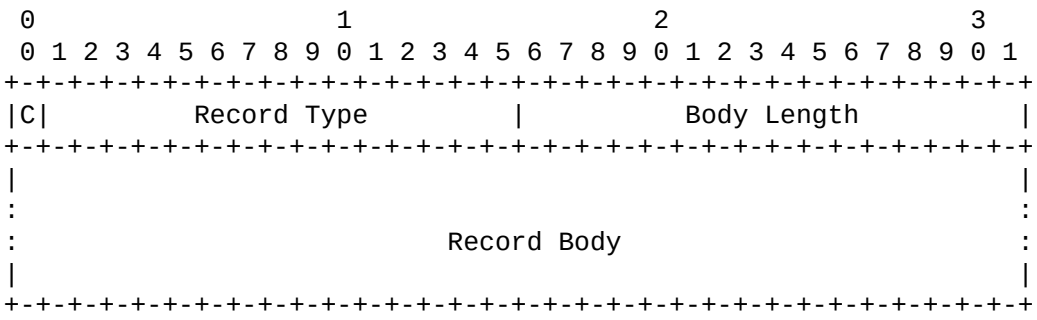


Figure 6: NTS-KE Record format

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

PTP Key Request

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 {...}
MAC Algorithm Negotiation (optional)	{CMAC || HMAC}
Requesting PTP Identity (Unicast only)	data set {...}
End of Message	

Figure 7: Structure of a PTP Key Request message

Figure 7 shows the record structure of a PTP Key Request message. In the right column typical values are shown as examples. Detailed information on types, sizes etc. is given in Section 3.2. The message starts with the NTS Next Protocol Negotiation record which in

this application always holds PTPv2.1. 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.

PTP Refusal

Record	Exemplary body contents
NTS Next Protocol Negotiation	PTPv2.1
NTS Message Version	1.0
NTS Message Type	PTP Refusal
Error	Association Port not registered
End of Message	

Figure 10: Structure of a PTP Refusal message

2.3.4. PTP Registration Request Message

PTP Registration Request

Record	Exemplary body contents
NTS Next Protocol Negotiation	PTPv2.1
NTS Message Version	1.0
NTS Message Type	PTP Registration Request
Requesting PTP Identity	data set {...}
AEAD Algorithm Negotiation	{AEAD_512 || AEAD_256}
MAC Algorithm Negotiation (optional)	{CMAC || HMAC}
End of 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

PTP Registration Success

Record	Exemplary body contents
NTS Next Protocol Negotiation	PTPv2.1
NTS Message Version	1.0
NTS Message Type	PTP Registration Success
Current Parameters	set of Records {...}
Next Parameters	set of Records {...}
End of 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 2 and begin with Record Type 1024 (compare IETF RFC 8915, 7.6 Table: NTS Next Protocol Negotiation IDs).

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.

NOTE: Currently the NTS messages are not repeated here. For the structure of the six NTS messages see Figures 7, 8, 10-12, 14.

3.2. NTS Records

NOTE: The detailed content of the NTS records defined as well as all necessary conditions of their usage are already specified. Due to time constraints, they could not yet be transferred into the XML format. So, currently they are missing in the descriptions below.

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 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, 4.1.5.

Content and conditions:

...

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:

...

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:

...

3.2.4. End of Message

The End of Message record is defined in IETF 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:

...

3.2.5. Error

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

Content and conditions:

...

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:

...

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:

...

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:

...

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:

...

3.2.10. NTS Message Type

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

Content and conditions:

...

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:

...

3.2.12. NTS Next Protocol Negotiation

The Next Protocol Negotiation record is defined in IETF 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:

...

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:

...

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:

...

3.2.15. Security Policies

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

Content and conditions:

...

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:

...

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:

...

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:

...

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:

...

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:

...

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:

...

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:

...

4. New TICKET TLV for PTP Messages

...

5. AUTHENTICATION TLV Parameters

...

6. IANA Considerations

Considerations should be made ...

...

7. Security Considerations

...

8. Acknowledgements

The authors would like to thank ...

9. References

9.1. Normative References

[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>.

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

[RFC7808] Douglass, M. and C. Daboo, "Time Zone Data Distribution Service", RFC 7808, DOI 10.17487/RFC7808, March 2016, <https://www.rfc-editor.org/info/rfc7808>.

[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>.

9.2. Informative References

[IEEE-1588-2008]
"IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", n.d..

[ntppool] "pool.ntp.org: the internet cluster of ntp servers", n.d., <https://www.ntppool.org>.

Authors' Addresses

Martin Langer
Ostfalia University of Applied Sciences

Email: mart.langer@ostfalia.de

Rainer Bermbach
Ostfalia University of Applied Sciences

Email: r.bermbach@ostfalia.de