                 Security Requirements of Time Protocols
                      in Packet Switched Networks
              draft-ietf-tictoc-security-requirements-11.txt

Abstract

   As time and frequency distribution protocols are becoming
   increasingly common and widely deployed, concern about their exposure
   to various security threats is increasing. This document defines a
   set of security requirements for time protocols, focusing on the
   Precision Time Protocol (PTP) and the Network Time Protocol (NTP).
   This document also discusses the security impacts of time protocol
   practices, the performance implications of external security
   practices on time protocols and the dependencies between other
   security services and time synchronization.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

As time protocols are becoming increasingly common and widely
deployed, concern about the resulting exposure to various security
threats is increasing. If a time protocol is compromised, the
applications it serves are prone to a range of possible attacks
including Denial-of-Service (DoS) or incorrect behavior.

This document discusses the security aspects of time distribution
protocols in packet networks, and focuses on the two most common
protocols, the Network Time Protocol [NTPv4] and the Precision Time
Protocol (PTP) [IEEE1588]. Note, that although PTP was not defined by
the IETF, it is one of the two most common time protocols and hence
it is included in the discussion.

The Network Time Protocol was defined with an inherent security
protocol; [NTPv4] defines a security protocol that is based on a
symmetric key authentication scheme, and [AutoKey] presents an
alternative security protocol, based on a public key authentication
scheme. [IEEE1588] includes an experimental security protocol,
defined in Annex K of the standard, but this Annex was never
formalized into a fully defined security protocol.

While NTP includes an inherent security protocol, the absence of a
standard security solution for PTP undoubtedly contributed to the
wide deployment of unsecured time synchronization solutions. However,
in some cases security mechanisms may not be strictly necessary,
e.g., due to other security practices in place, or due to the
architecture of the network. A time synchronization security
solution, much like any security solution, is comprised of various
building blocks, and must be carefully tailored for the specific
system it is deployed in. Based on a system-specific threat
assessment, the benefits of a security solution must be weighed
against the potential risks, and based on this tradeoff an optimal
security solution can be selected.

The target audience of this document includes:

o Timing and networking equipment vendors - can benefit from this
   document by deriving the security features that should be
   supported in the time/networking equipment.

o Standard development organizations - can use the requirements
   defined in this document when specifying security mechanisms for a
   time protocol.

o Network operators - can use this document as a reference when
   designing the network and its security architecture. As stated
   above, the requirements in this document may be deployed
   selectively based on a careful per-system threat analysis.

This document attempts to add clarity to the time protocol security
requirements discussion by addressing a series of questions:

(1) What are the threats that need to be addressed for the time
protocol, and thus what security services need to be provided? (e.g.
a malicious NTP server or PTP master)

(2) What external security practices impact the security and
performance of time keeping, and what can be done to mitigate these
impacts? (e.g. an IPsec tunnel in the time protocol traffic path)

(3) What are the security impacts of time protocol practices? (e.g.
on-the-fly modification of timestamps)

(4) What are the dependencies between other security services and
time protocols? (e.g. which comes first - the certificate or the
timestamp?)

In light of the questions above, this document defines a set of
requirements for security solutions for time protocols, focusing on
PTP and NTP.

2. Conventions Used in this Document

2.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [KEYWORDS].

This document describes security requirements, and thus requirements
are phrased in the document in the form "the security mechanism
MUST/SHOULD/...". Note, that the phrasing does not imply that this
document defines a specific security mechanism, but defines the
requirements with which every security mechanism should comply.

2.2. Abbreviations

BC   Boundary Clock [IEEE1588]

DoS   Denial of Service

MITM   Man In The Middle

NTP   Network Time Protocol [NTPv4]

OC   Ordinary Clock [IEEE1588]

P2P TC   Peer-to-Peer Transparent Clock [IEEE1588]

   PTP   Precision Time Protocol [IEEE1588]

   TC   Transparent Clock [IEEE1588]

2.3. Common Terminology for PTP and NTP

   This document refers to both PTP and NTP. For the sake of
   consistency, throughout the document the term "master" applies to
   both a PTP master and an NTP server. Similarly, the term "slave"
   applies to both PTP slaves and NTP clients. The term "protocol
   packets" refers generically to PTP and NTP messages.

2.4. Terms used in this Document

   o Clock - A node participating in the protocol (either PTP or NTP).
     A clock can be a master, a slave, or an intermediate clock (see
     corresponding definitions below).

   o Control packets - Packets used by the protocol to exchange
     information between clocks that is not strictly related to the
     time. NTP uses NTP Control Messages. PTP uses Announce, Signaling
     and Management messages.

   o End-to-end security - A security approach where secured packets
     sent from a source to a destination is not modified by
     intermediate nodes.

   o Grandmaster - A master that receives time information from a
     locally attached clock device, and not through the network. A
     grandmaster distributes its time to other clocks in the network.

   o Hop-by-hop security - A security approach where secured packets
     sent from a source to a destination may be modified by
     intermediate nodes. In this approach intermediate nodes share the
     encryption key with the source and destination, allowing them to
     re-encrypt or re-authenticate modified packets before relaying
     them to the destination.

   o Intermediate clock - A clock that receives timing information from
     a master, and sends timing information to other clocks.
     In NTP this term refers to an NTP server that is not a Stratum 1
     server. In PTP this term refers to a BC or a TC.

   o Master - A clock that generates timing information to other clocks
     in the network.
     In NTP 'master' refers to an NTP server. In PTP 'master' refers to
     a master OC (aka grandmaster) or to a port of a BC that is in the
     master state.

   o Protocol packets - Packets used by the time protocol. The
     terminology used in this document distinguishes between time
     packets and control packets.

   o Secured clock - A clock that supports a security mechanism that
     complies to the requirements in this document.

   o Slave - A clock that receives timing information from a master. In
     NTP 'slave' refers to an NTP client. In PTP 'slave' refers to a
     slave OC, or to a port of a BC that is in the slave state.

   o Time packets - Protocol packets carrying time information.

   o Unsecured clock - A clock that does not support a security
     mechanism according to the requirements in this document.

3. Security Threats

   This section discusses the possible attacker types and analyzes
   various attacks against time protocols.

   The literature is rich with security threats of time protocols, e.g.,
   [Traps], [AutoKey], [TM], [SecPTP], and [SecSen]. The threat analysis
   in this document is mostly based on [TM].

3.1. Threat Model

   A time protocol can be attacked by various types of attackers.

   The analysis in this document classifies attackers according to 2
   criteria, as described in Section 3.1.1. and Section 3.1.2.

3.1.1. Internal vs. External Attackers

   In the context of internal and external attackers, the underlying
   assumption is that the time protocol is secured either by an
   encryption or an authentication mechanism, or both.

   Internal attackers either have access to a trusted segment of the
   network, or possess the encryption or authentication keys. An
   internal attack can also be performed by exploiting vulnerabilities

in devices; for example, by installing malware, or obtaining
credentials to reconfigure the device. Thus, an internal attacker can
maliciously tamper with legitimate traffic in the network, as well as
generate its own traffic and make it appear legitimate to its
attacked nodes.

Note that internal attacks are a special case of Byzantine failures,
where a node in the system may fail in arbitrary ways; by crashing,
by omitting messages, or by malicious behavior. This document focuses
on nodes that demonstrate malicious behavior.

External attackers, on the other hand, do not have the keys, and have
access only to the encrypted or authenticated traffic.

Obviously, in the absence of a security mechanism there is no
distinction between internal and external attackers, since all
attackers are internal in practice.

3.1.2. Man in the Middle (MITM) vs. Packet Injector

MITM attackers are located in a position that allows interception and
modification of in-flight protocol packets. It is assumed that an
MITM attacker has physical access to a segment of the network, or has
gained control of one of the nodes in the network.

A traffic injector is not located in an MITM position, but can attack
by generating protocol packets. An injector can reside either within
the attacked network, or on an external network that is connected to
the attacked network. An injector can also potentially eavesdrop on
protocol packets sent as multicast, record them and replay them
later.

3.2. Threat Analysis

3.2.1. Packet Manipulation

A packet manipulation attack results when an MITM attacker receives
timing protocol packets, alters them and relays them to their
destination, allowing the attacker to maliciously tamper with the
protocol. This can result in a situation where the time protocol is
apparently operational but providing intentionally inaccurate
information.

3.2.2. Spoofing

In spoofing, an injector masquerades as a legitimate node in the
network by generating and transmitting protocol packets or control
packets. Two typical examples of spoofing attacks:

o An attacker can impersonate the master, allowing malicious
   distribution of false timing information.

o An attacker can impersonate a legitimate clock, a slave or an
   intermediate clock, by sending malicious messages to the master,
   causing the master to respond to the legitimate clock with
   protocol packets that are based on the spoofed messages.
   Consequently, the delay computations of the legitimate clock are
   based on false information.

As with packet manipulation, this attack can result in a situation
where the time protocol is apparently operational but providing
intentionally inaccurate information.

3.2.3. Replay Attack

In a replay attack, an attacker records protocol packets and replays
them at a later time without any modification. This can also result
in a situation where the time protocol is apparently operational but
providing intentionally inaccurate information.

3.2.4. Rogue Master Attack

In a rogue master attack, an attacker causes other nodes in the
network to believe it is a legitimate master. As opposed to the
spoofing attack, in the Rogue Master attack the attacker does not
fake its identity, but rather manipulates the master election process
using malicious control packets. For example, in PTP, an attacker can
manipulate the Best Master Clock Algorithm (BMCA), and cause other
nodes in the network to believe it is the most eligible candidate to
be a grandmaster.

In PTP, a possible variant of this attack is the rogue TC/BC attack.
Similar to the rogue master attack, an attacker can cause victims to
believe it is a legitimate TC or BC, allowing the attacker to
manipulate the time information forwarded to the victims.

3.2.5. Packet Interception and Removal

A packet interception and removal attack results when an MITM attacker intercepts and drops protocol packets, preventing the destination node from receiving some or all of the protocol packets.

3.2.6. Packet Delay Manipulation

In a packet delay manipulation scenario, an MITM attacker receives protocol packets, and relays them to their destination after adding a maliciously computed delay. The attacker can use various delay attack strategies; the added delay can be constant, jittered, or slowly wandering. Each of these strategies has a different impact, but they all effectively manipulate the attacked clock.

Note that the victim still receives one copy of each packet, contrary to the replay attack, where some or all of the packets may be received by the victim more than once.

3.2.7. L2/L3 DoS Attacks

There are many possible Layer 2 and Layer 3 DoS attacks, e.g., IP spoofing, ARP spoofing [Hack], MAC flooding [Anatomy], and many others. As the target's availability is compromised, the timing protocol is affected accordingly.

3.2.8. Cryptographic Performance Attacks

In cryptographic performance attacks, an attacker transmits fake protocol packets, causing high utilization of the cryptographic engine at the receiver, which attempts to verify the integrity of these fake packets.

This DoS attack is applicable to all encryption and authentication protocols. However, when the time protocol uses a dedicated security mechanism implemented in a dedicated cryptographic engine, this attack can be applied to cause DoS specifically to the time protocol.

3.2.9. DoS Attacks against the Time Protocol

An attacker can attack a clock by sending an excessive number of time protocol packets, thus degrading the victim's performance. This attack can be implemented, for example, using the attacks described in Section 3.2.2. and Section 3.2.4.

[3.2.10](#). Grandmaster Time Source Attack (e.g., GPS fraud)

   Grandmasters receive their time from an external accurate time
   source, such as an atomic clock or a GPS clock, and then distribute
   this time to the slaves using the time protocol.

   Time source attack are aimed at the accurate time source of the
   grandmaster. For example, if the grandmaster uses a GPS based clock
   as its reference source, an attacker can jam the reception of the GPS
   signal, or transmit a signal similar to one from a GPS satellite,
   causing the grandmaster to use a false reference time.

   Note that this attack is outside the scope of the time protocol.
   While various security measures can be taken to mitigate this attack,
   these measures are outside the scope of the security requirements
   defined in this document.

[3.3](#). Threat Analysis Summary

   The two key factors to a threat analysis are the impact and the
   likelihood of each of the analyzed attacks.

   Table 1 summarizes the security attacks presented in [Section 3.2](#).
   For each attack, the table specifies its impact, and its
   applicability to each of the attacker types presented in [Section 3.1](#).

   Table 1 clearly shows the distinction between external and internal
   attackers, and motivates the usage of authentication and integrity
   protection, significantly reducing the impact of external attackers.

   The Impact column provides an intuitive measure of the severity of
   each attack, and the relevant Attacker Type columns provide an
   intuition about how difficult each attack is to implement, and hence
   about the likelihood of each attack.

   The impact column in Table 1 can have one of 3 values:

   o DoS - the attack causes denial of service to the attacked node,
     the impact of which is not restricted to the time protocol.

   o Accuracy degradation - the attack yields a degradation in the
     slave accuracy, but does not completely compromise the slaves'
     time and frequency.

o False time - slaves align to a false time or frequency value due
   to the attack. Note that if the time protocol aligns to a false
   time, it may cause DoS to other applications that rely on accurate
   time. However, for the purpose of the analysis in this section we
   distinguish this implication from 'DoS', which refers to a DoS
   attack that is not necessarily aimed at the time protocol.
   All attacks that have a '+' for 'False Time' implicitly have a '+'
   for 'Accuracy Degradation'.
   Note, that 'False Time' necessarily implies 'Accuracy
   Degradation'. However, two different terms are used, indicating
   two levels of severity.

The Attacker Type columns refer to the 4 possible combinations of the
attacker types defined in Section 3.1.

Attack	Impact		Attacker Type			
	False Time	Accuracy Degrad.	DoS	Internal MITM	Internal Inj.	External MITM	External Inj.
Manipulation	+			+			
Spoofing	+			+	+		
Replay attack	+			+	+		
Rogue master attack	+			+	+		
Interception and removal		+	+	+		+	
Packet delay manipulation	+			+		+	
L2/L3 DoS attacks			+	+	+	+	+
Crypt. performance attacks			+	+	+	+	+
Time protocol DoS attacks			+	+	+		
Master time source attack (e.g., GPS spoofing)	+			+	+	+	+

Table 1 Threat Analysis - Summary

The threats discussed in this section provide the background for the
security requirements presented in Section 5.

4. Requirement Levels

The security requirements are presented in Section 5. Each
requirement is defined with a requirement level, in accordance with
the requirement levels defined in Section 2.1.

The requirement levels in this document are affected by the following
factors:

o Impact:
   The possible impact of not implementing the requirement, as
   illustrated in the 'impact' column of Table 1.
   For example, a requirement that addresses a threat that can be
   implemented by an external injector is typically a 'MUST', since
   the threat can be implemented by all the attacker types analyzed
   in Section 3.1.

o Difficulty of the corresponding attack:
   The level of difficulty of the possible attacks that become
   possible by not implementing the requirement. The level of
   difficulty is reflected in the 'Attacker Type' column of Table 1.
   For example, a requirement that addresses a threat that only
   compromises the availability of the protocol is typically no more
   than a 'SHOULD'.

o Practical considerations:
   Various practical factors that may affect the requirement.
   For example, if a requirement is very difficult to implement, or
   is applicable to very specific scenarios, these factors may reduce
   the requirement level.

Section 5. lists the requirements. For each requirement there is a
short explanation detailing the reason for its requirement level.

5. Security Requirements

This section defines a set of security requirements. These
requirements are phrased in the form "the security mechanism
MUST/SHOULD/MAY...". However, this document does not specify how
these requirements can be met. While these requirements can be
satisfied by defining explicit security mechanisms for time
protocols, at least a subset of the requirements can be met by
applying common security practices to the network or by using
existing security protocols, such as [IPsec] or [MACsec]. Thus,

security solutions that address these requirements are outside the
scope of this document.

5.1. Clock Identity Authentication and Authorization

Requirement

The security mechanism MUST support authentication.

Requirement

The security mechanism MUST support authorization.

Requirement Level

The requirements in this subsection address the spoofing attack
(Section 3.2.2.), and the rogue master attack (Section 3.2.4.).

The requirement level of these requirements is 'MUST' since in the
absence of these requirements the protocol is exposed to attacks that
are easy to implement and have a high impact.

Discussion

Authentication refers to verifying the identity of the peer clock.
Authorization, on the other hand, refers to verifying that the peer
clock is permitted to play the role that it plays in the protocol.
For example, some nodes may be permitted to be masters, while other
nodes are only permitted to be slaves or TCs.

Authentication is typically implemented by means of a cryptographic
signature, allowing to verify the identity of the sender.
Authorization requires clocks to maintain a list of authorized
clocks, or a "black list" of clocks that should be denied service or
revoked.

It is noted that while the security mechanism is required to provide
an authorization mechanism, the deployment of such a mechanism
depends on the nature of the network. For example, a network that
deploys PTP may consist of a set of identical OCs, where all clocks
are equally permitted to be a master. In such a network an
authorization mechanism may not be necessary.

The following subsections describe 5 distinct cases of clock
authentication.

5.1.1. **Authentication and Authorization of Masters**

Requirement

   The security mechanism MUST support an authentication mechanism,
   allowing slaves to authenticate the identity of masters.

Requirement

   The authentication mechanism MUST allow slaves to verify that the
   authenticated master is authorized to be a master.

Requirement Level

   The requirements in this subsection address the spoofing attack
   (Section 3.2.2.), and the rogue master attack (Section 3.2.4.).

   The requirement level of these requirements is 'MUST' since in the
   absence of these requirements the protocol is exposed to attacks that
   are easy to implement and have a high impact.

Discussion

   Clocks authenticate masters in order to ensure the authenticity of
   the time source. It is important for a slave to verify the identity
   of the master, as well as to verify that the master is indeed
   authorized to be a master.

5.1.2. **Recursive Authentication and Authorization of Masters (Chain of
   Trust)**

Requirement

   The security mechanism MUST support recursive authentication and
   authorization of the master, to be used in cases where time
   information is conveyed through intermediate clocks.

Requirement Level

   The requirement in this subsection addresses the spoofing attack
   (Section 3.2.2.), and the rogue master attack (Section 3.2.4.).

   The requirement level of this requirement is 'MUST' since in the
   absence of this requirement the protocol is exposed to attacks that
   are easy to implement and have a high impact.

Discussion

In some cases a slave is connected to an intermediate clock, that is
not the primary time source. For example, in PTP a slave can be
connected to a Boundary Clock (BC) or a Transparent Clock (TC), which
in turn is connected to a grandmaster. A similar example in NTP is
when a client is connected to a stratum 2 server, which is connected
to a stratum 1 server. In both the PTP and the NTP cases, the slave
authenticates the intermediate clock, and the intermediate clock
authenticates the grandmaster. This recursive authentication process
is referred to in [AutoKey] as proventication.

Specifically in PTP, this requirement implies that if a slave
receives time information through a TC, it must authenticate the TC
it is attached to, as well as authenticate the master it receives the
time information from, as per Section 5.1.1. Similarly, if a TC
receives time information through an attached TC, it must
authenticate the attached TC.

5.1.3. Authentication and Authorization of Slaves

Requirement

   The security mechanism MAY provide a means for a master to
   authenticate its slaves.

Requirement

   The security mechanism MAY provide a means for a master to verify
   that the sender of a protocol packet is authorized to send a packet
   of this type.

Requirement Level

   The requirement in this subsection prevents DoS attacks against the
   master (Section 3.2.9.).

   The requirement level of this requirement is 'MAY' since:

   o Its low impact, i.e., in the absence of this requirement the
     protocol is only exposed to DoS.

   o Practical considerations: requiring an NTP server to authenticate
     its clients may significantly impose on the server's performance.

   Note that while the requirement level of this requirement is 'MAY',
   the requirement in Section 5.1.1. is 'MUST'; the security mechanism
   must provide a means for authentication and authorization, with an

emphasis on the master. Authentication and authorization of slaves is
specified in this subsection as 'MAY'.

Discussion

   Slaves and intermediate clocks are authenticated by masters in order
   to verify that they are authorized to receive timing services from
   the master.

   Authentication of slaves prevents unauthorized clocks from receiving
   time services. Preventing the master from serving unauthorized clocks
   can help in mitigating DoS attacks against the master. Note that the
   authentication of slaves might put a higher load on the master than
   serving the unauthorized clock, and hence this requirement is a MAY.

5.1.4. **PTP: Authentication and Authorization of P2P TCs by the Master**

Requirement

   The security mechanism for PTP MAY provide a means for a master to
   authenticate the identity of the P2P TCs directly connected to it.

Requirement

   The security mechanism for PTP MAY provide a means for a master to
   verify that P2P TCs directly connected to it are authorized to be
   TCs.

Requirement Level

   The requirement in this subsection prevents DoS attacks against the
   master (Section 3.2.9.).

   The requirement level of this requirement is 'MAY' for the same
   reasons specified in Section 5.1.3.

Discussion

   P2P TCs that are one hop from the master use the PDelay_Req and
   PDelay_Resp handshake to compute the link delay between the master
   and TC. These TCs are authenticated by the master.

   Authentication of TCs, much like authentication of slaves, reduces
   unnecessary load on the master and peer TCs, by preventing the master
   from serving unauthorized clocks.

5.1.5. PTP: Authentication and Authorization of Control Messages

Requirement

   The security mechanism for PTP MUST support authentication of
   Announce messages. The authentication mechanism MUST also verify that
   the sender is authorized to be a master.

Requirement

   The security mechanism for PTP MUST support authentication and
   authorization of Management messages.

Requirement

   The security mechanism MAY support authentication and authorization
   of Signaling messages.

Requirement Level

   The requirements in this subsection address the spoofing attack
   (Section 3.2.2.), and the rogue master attack (Section 3.2.4.).

   The requirement level of the first two requirements is 'MUST' since
   in the absence of these requirements the protocol is exposed to
   attacks that are easy to implement and have a high impact.

   The requirement level of the third requirement is 'MAY' since its
   impact greatly depends on the application for which the Signaling
   messages are used for.

Discussion

   Master election is performed in PTP using the Best Master Clock
   Algorithm (BMCA). Each Ordinary Clock (OC) announces its clock
   attributes using Announce messages, and the best master is elected
   based on the information gathered from all the candidates. Announce
   messages must be authenticated in order to prevent rogue master
   attacks (Section 3.2.4.). Note, that this subsection specifies a
   requirement that is not necessarily included in Section 5.1.1. or in
   Section 5.1.3. , since the BMCA is initiated before clocks have been
   defined as masters or slaves.

   Management messages are used to monitor or configure PTP clocks.
   Malicious usage of Management messages enables various attacks, such
   as the rogue master attack, or DoS attack.

Signaling messages are used by PTP clocks to exchange information
that is not strictly related to time information or to master
selection, such as unicast negotiation. Authentication and
authorization of Signaling message may be required in some systems,
depending on the application these messages are used for.

5.2. Protocol Packet Integrity

Requirement

The security mechanism MUST protect the integrity of protocol
packets.

Requirement Level

The requirement in this subsection addresses the packet manipulation
attack (Section 3.2.1.).

The requirement level of this requirement is 'MUST' since in the
absence of this requirement the protocol is exposed to attacks that
are easy to implement and have high impact.

Discussion

While Section 5.1. refers to ensuring the identity an authorization
of the source of a protocol packet, this subsection refers to
ensuring that the packet arrived intact. The integrity protection
mechanism ensures the authenticity and completeness of data from the
data originator.

Integrity protection is typically implemented by means of an
Integrity Check Value (ICV) that is included in protocol packets and
is verified by the receiver.

5.2.1. PTP: Hop-by-hop vs. End-to-end Integrity Protection

Specifically in PTP, when protocol packets are subject to
modification by TCs, the integrity protection can be enforced in one
of two approaches, end-to-end or hop-by-hop.

5.2.1.1. Hop-by-Hop Integrity Protection

Each hop that needs to modify a protocol packet:

o Verifies its integrity.

   o Modifies the packet, i.e., modifies the correctionField.
     Note: Transparent Clocks (TCs) improve the end-to-end accuracy by
     updating a "correctionField" (clause 6.5 in [IEEE1588]) in the PTP
     packet by adding the latency caused by the current TC.

   o Re-generates the integrity protection, e.g., re-computes a Message
     Authentication Code.

   In the hop-by-hop approach, the integrity of protocol packets is
   protected by induction on the path from the originator to the
   receiver.

   This approach is simple, but allows rogue TCs to modify protocol
   packets.

5.2.1.2. End-to-End Integrity Protection

   In this approach, the integrity protection is maintained on the path
   from the originator of a protocol packet to the receiver. This allows
   the receiver to directly validate the protocol packet without the
   ability of intermediate TCs to manipulate the packet.

   Since TCs need to modify the correctionField, a separate integrity
   protection mechanism is used specifically for the correctionField.

   The end-to-end approach limits the TC's impact to the correctionField
   alone, while the rest of the protocol packet is protected on an end-
   to-end basis. It should be noted that this approach is more difficult
   to implement than the hop-by-hop approach, as it requires the
   correctionField to be protected separately from the other fields of
   the packet, possibly using different cryptographic mechanisms and
   keys.

5.3. Spoofing Prevention

Requirement

   The security mechanism MUST provide a means to prevent master
   spoofing.

Requirement

   The security mechanism MUST provide a means to prevent slave
   spoofing.

Requirement

   PTP: The security mechanism MUST provide a means to prevent P2P TC
   spoofing.

Requirement Level

   The requirements in this subsection address spoofing attacks (Section
   3.2.2.). As described in Section 3.2.2. , when these requirements
   are not met, the attack may have a high impact, causing slaves to
   rely on false time information. Thus, the requirement level is
   'MUST'.

Discussion

   Spoofing attacks may take various different forms, and can
   potentially cause significant impact. In a master spoofing attack,
   the attacker causes slaves to receive false information about the
   current time by masquerading as the master.

   By spoofing a slave or an intermediate node (the second example of
   Section 3.2.2.), an attacker can tamper with the slaves' delay
   computations. These attacks can be mitigated by an authentication
   mechanism (Section 5.1.3. and 5.1.4.), or by other means, for
   example, a PTP Delay_Req can include a Message Authentication Code
   (MAC) that is included in the corresponding Delay_Resp message,
   allowing the slave to verify that the Delay_Resp was not sent in
   response to a spoofed message.

5.4. Availability

Requirement

   The security mechanism SHOULD include measures to mitigate DoS
   attacks against the time protocol.

Requirement Level

   The requirement in this subsection prevents DoS attacks against the
   protocol (Section 3.2.9.).

   The requirement level of this requirement is 'SHOULD' due to its low
   impact, i.e., in the absence of this requirement the protocol is only
   exposed to DoS.

Discussion

   The protocol availability can be compromised by several different
   attacks. An attacker can inject protocol packets to implement the

spoofing attack (Section 3.2.2.) or the rogue master attack (Section 3.2.4.), causing DoS to the victim (Section 3.2.9.).

An authentication mechanism (Section 5.1.) limits these attacks strictly to internal attackers, and thus prevents external attackers from performing them. Hence, the requirements of Section 5.1. can be used to mitigate this attack. Note, that Section 5.1. addresses a wider range of threats, whereas the current section is focused on availability.

The DoS attacks described in Section 3.2.7. are performed at lower layers than the time protocol layer, and are thus outside the scope of the security requirements defined in this document.

5.5. Replay Protection

Requirement

The security mechanism MUST include a replay prevention mechanism.

Requirement Level

The requirement in this subsection prevents replay attacks (Section 3.2.3.).

The requirement level of this requirement is 'MUST' since in the absence of this requirement the protocol is exposed to attacks that are easy to implement and have a high impact.

Discussion

The replay attack (Section 3.2.3.) can compromise both the integrity and availability of the protocol. Common encryption and authentication mechanisms include replay prevention mechanisms that typically use a monotonously increasing packet sequence number.

5.6. Cryptographic Keys and Security Associations

5.6.1. Key Freshness

Requirement

The cryptographic keys MUST be refreshed frequently.

Requirement Level

   The requirement level of this requirement is 'MUST' since key
   freshness is an essential property for cryptographic algorithms, as
   discussed below.

Discussion

   Key freshness guarantees that both sides share a common updated
   secret key. It also helps in preventing replay attacks. Thus, it is
   important for keys to be refreshed frequently. Note that the term
   'frequently' is used without a quantitative requirement, as the
   precise frequency requirement should be considered on a per-system
   basis, based on the threats and system requirements.

5.6.2. Security Association

Requirement

   The security protocol SHOULD support a security association protocol
   where:

   o Two or more clocks authenticate each other.

   o The clocks generate and agree on a cryptographic session key.

Requirement

   Each instance of the association protocol SHOULD produce a different
   session key.

Requirement Level

   The requirement level of this requirement is 'SHOULD' since it may be
   expensive in terms of performance, especially in low-cost clocks.

Discussion

   The security requirements in Section 5.1. and Section 5.2. require
   usage of cryptographic mechanisms, deploying cryptographic keys. A
   security association (e.g., [IPsec]) is an important building block
   in these mechanisms.

   It should be noted that in some cases different security association
   mechanisms may be used at different levels of clock hierarchies. For
   example, the association between a Stratum 2 clock and a Stratum 3
   clock in NTP may have different characteristics than an association
   between two clocks at the same stratum level. On a related note, in

some cases a hybrid solution may be used, where a subset of the
network is not secured at all (see Section 5.10.2.).

5.6.3. Unicast and Multicast Associations

Requirement

   The security mechanism SHOULD support security association protocols
   for unicast and for multicast associations.

Requirement Level

   The requirement level of this requirement is 'SHOULD' since it may be
   expensive in terms of performance, especially for low-cost clocks.

Discussion

   A unicast protocol requires an association protocol between two
   clocks, whereas a multicast protocol requires an association protocol
   among two or more clocks, where one of the clocks is a master.

5.7. Performance

Requirement

   The security mechanism MUST be designed in such a way that it does
   not significantly degrade the quality of the time transfer.

Requirement

   The mechanism SHOULD minimize computational load.

Requirement

   The mechanism SHOULD minimize storage requirements of client state in
   the master.

Requirement

   The mechanism SHOULD minimize the bandwidth overhead required by the
   security protocol.

Requirement Level

   While the quality of the time transfer is clearly a 'MUST', the other
   3 performance requirements are 'SHOULD', since some systems may be

   more sensitive to resource consumption than others, and hence these
   requirements should be considered on a per-system basis.

Discussion

   Performance efficiency is important since client restrictions often
   dictate a low processing and memory footprint, and because the server
   may have extensive fan-out.

   Note that the performance requirements refer to a time-protocol-
   specific security mechanism. In systems where a security protocol is
   used for other types of traffic as well, this document does not place
   any performance requirements on the security protocol performance.
   For example, if IPsec encryption is used for securing all information
   between the master and slave node, including information that is not
   part of the time protocol, the requirements in this subsection are
   not necessarily applicable.

5.8. Confidentiality

Requirement

   The security mechanism MAY provide confidentiality protection of the
   protocol packets.

Requirement Level

   The requirement level of this requirement is 'MAY' since it does not
   prevent severe threats, as discussed below.

Discussion

   In the context of time protocols, confidentiality is typically of low
   importance, since timing information is typically not considered
   secret information.

   Confidentiality can play an important role when service providers
   charge their customers for time synchronization services, and thus an
   encryption mechanism can prevent eavesdroppers from obtaining the
   service without payment. Note that these cases are, for now, rather
   esoteric.

   Confidentiality can also prevent an MITM attacker from identifying
   protocol packets. Thus, confidentiality can assist in protecting the
   timing protocol against MITM attacks such as packet delay (Section
   3.2.6.), manipulation and interception and removal attacks. Note,
   that time protocols have predictable behavior even after encryption,

such as packet transmission rates and packet lengths. Additional
measures can be taken to mitigate encrypted traffic analysis by
random padding of encrypted packets and by adding random dummy
packets. Nevertheless, encryption does not prevent such MITM attacks,
but rather makes these attacks more difficult to implement.

5.9. Protection against Packet Delay and Interception Attacks

Requirement

   The security mechanism MUST include means to protect the protocol
   from MITM attacks that degrade the clock accuracy.

Requirement Level

   The requirements in this subsection address MITM attacks such as the
   packet delay attack (Section 3.2.6.) and packet interception attacks
   (Section 3.2.5. and Section 3.2.1.).

   The requirement level of this requirement is 'MUST'. In the absence
   of this requirement the protocol is exposed to attacks that are easy
   to implement and have a high impact. Note that in the absence of this
   requirement, the impact is similar to packet manipulation attacks
   (Section 3.2.1.), and thus this requirement has the same requirement
   level as integrity protection (Section 5.2.).

   It is noted that the implementation of this requirement depends on
   the topology and properties of the system.

Discussion

   While this document does not define specific security solutions, we
   note that common practices for protection against MITM attacks use
   redundant masters (e.g. [NTPv4]), or redundant paths between the
   master and slave (e.g. [DelayAtt]). If one of the time sources
   indicates a time value that is significantly different than the other
   sources, it is assumed to be erroneous or under attack, and is
   therefore ignored.

   Thus, MITM attack prevention derives a requirement from the security
   mechanism, and a requirement from the network topology. While the
   security mechanism should support the ability to detect delay
   attacks, it is noted that in some networks it is not possible to
   provide the redundancy needed for such a detection mechanism.

[5.10](). Combining Secured with Unsecured Nodes

   Integrating a security mechanism into a time synchronized system is a
   complex and expensive process, and hence in some cases may require
   incremental deployment, where new equipment supports the security
   mechanism, and is required to interoperate with legacy equipment
   without the security features.

[5.10.1](). Secure Mode

Requirement

   The security mechanism MUST support a secure mode, where only secured
   clocks are permitted to take part in the time protocol. In this mode
   every protocol packet received from an unsecured clock MUST be
   discarded.

Requirement Level

   The requirement level of this requirement is 'MUST' since the full
   capacity of the security requirements defined in this document can
   only be achieved in secure mode.

Discussion

   While the requirement in this subsection is similar to the one in
   5.1. , it refers to the secure mode, as opposed to the hybrid mode
   presented in the next subsection.

[5.10.2](). Hybrid Mode

Requirement

   The security protocol SHOULD support a hybrid mode, where both
   secured and unsecured clocks are permitted to take part in the
   protocol.

Requirement Level

   The requirement level of this requirement is a 'SHOULD'; on one hand
   hybrid mode enables a gradual transition from unsecured to secured
   mode, which is especially important in large-scaled deployments. On
   the other hand, hybrid mode is not required in all systems; this
   document recommends to deploy the 'Secure Mode' described in [Section
   5.10.1](). where possible.

Discussion

   The hybrid mode allows both secured and unsecured clocks to take part
   in the time protocol. NTP, for example, allows a mixture of secured
   and unsecured nodes.

Requirement

   A master in the hybrid mode SHOULD be a secured clock.

   A secured slave in the hybrid mode SHOULD discard all protocol
   packets received from unsecured clocks.

Requirement Level

   The requirement level of this requirement is a 'SHOULD', since it may
   not be applicable to all deployments. For example, a hybrid network
   may require the usage of unsecured masters or TCs.

Discussion

   This requirement ensures that the existence of unsecured clocks does
   not compromise the security provided to secured clocks. Hence,
   secured slaves only "trust" protocol packets received from a secured
   clock.

   An unsecured slave can receive protocol packets either from unsecured
   clocks, or from secured clocks. Note that the latter does not apply
   when encryption is used. When integrity protection is used, the
   unsecured slave can receive secured packets ignoring the integrity
   protection.

   Note that the security scheme in [NTPv4] with [AutoKey] does not
   satisfy this requirement, since nodes prefer the server with the most
   accurate clock, which is not necessarily the server that supports
   authentication. For example, a stratum 2 server is connected to two
   stratum 1 servers, Server A, supporting authentication, and server B,
   without authentication. If server B has a more accurate clock than A,
   the stratum 2 server chooses server B, in spite of the fact it does
   not support authentication.

6. Summary of Requirements

```
   +-----------+----------------------------------------------+--------+
   | Section   | Requirement                                  | Type   |
   +-----------+----------------------------------------------+--------+
   | 5.1.      | Authentication & authorization of sender.    | MUST   |
   |           +----------------------------------------------+--------+
```

| | Authentication & authorization of master. | MUST |
| +---+--------+
| | Recursive authentication & authorization. | MUST |
| +---+--------+
| | Authentication & authorization of slaves. | MAY |
| +---+--------+
| | PTP: Authentication & authorization of | MAY |
| | P2P TCs by master. | |
| +---+--------+
| | PTP: Authentication & authorization of | MUST |
| | Announce messages. | |
| +---+--------+
| | PTP: Authentication & authorization of | MUST |
| | Management messages. | |
| +---+--------+
| | PTP: Authentication & authorization of | MAY |
| | Signaling messages. | |
+-----------+---+--------+
| 5.2. | Integrity protection. | MUST |
+-----------+---+--------+
| 5.3. | Spoofing prevention. | MUST |
+-----------+---+--------+
| 5.4. | Protection from DoS attacks against the | SHOULD |
| | time protocol. | |
+-----------+---+--------+
| 5.5. | Replay protection. | MUST |
+-----------+---+--------+
| 5.6. | Key freshness. | MUST |
| +---+--------+
| | Security association. | SHOULD |
| +---+--------+
| | Unicast and multicast associations. | SHOULD |
+-----------+---+--------+
| 5.7. | Performance: no degradation in quality of | MUST |
| | time transfer. | |
| +---+--------+
| | Performance: computation load. | SHOULD |
| +---+--------+
| | Performance: storage. | SHOULD |
| +---+--------+
| | Performance: bandwidth. | SHOULD |

```
+-----------+-------------------------------------------+--------+
| 5.8.      | Confidentiality protection.               | MAY    |
+-----------+-------------------------------------------+--------+
| 5.9.      | Protection against delay and interception | MUST   |
|           | attacks.                                  |        |
+-----------+-------------------------------------------+--------+
| 5.10.     | Secure mode.                              | MUST   |
|           +-------------------------------------------+--------+
|           | Hybrid mode.                              | SHOULD |
+-----------+-------------------------------------------+--------+
```
                Table 2 Summary of Security Requirements

7. Additional security implications

   This section discusses additional implications of the interaction
   between time protocols and security mechanisms.

   This section refers to time protocol security mechanisms, as well as
   to "external" security mechanisms, i.e., security mechanisms that are
   not strictly related to the time protocol.

7.1. Security and on-the-fly Timestamping

   Time protocols often require that protocol packets be modified during
   transmission. Both NTP and PTP in one-step mode require clocks to
   modify protocol packets based on the time of transmission and/or
   reception.

   In the presence of a security mechanism, whether encryption or
   integrity protection:

   o During transmission the encryption and/or integrity protection
      MUST be applied after integrating the timestamp into the packet.

   To allow high accuracy, timestamping is typically performed as close
   to the transmission or reception time as possible. However, since the
   security engine must be placed between the timestamping function and
   the physical interface, it may introduce non-deterministic latency
   that causes accuracy degradation. These performance aspects have been
   analyzed in literature, e.g., [1588IPsec] and [Tunnel].

7.2. PTP: Security and Two-Step Timestamping

   PTP supports a two-step mode of operation, where the time of
   transmission of protocol packets is communicated without modifying

the packets. As opposed to one-step mode, two-step timestamping can
be performed without the requirement to encrypt after timestamping.

Note that if an encryption mechanism such as IPsec is used, it
presents a challenge to the timestamping mechanism, since time
protocol packets are encrypted when traversing the physical
interface, and are thus impossible to identify. A possible solution
to this problem [IPsecSync] is to include an indication in the
encryption header that identifies time protocol packets.

7.3. Intermediate Clocks

A time protocol allows slaves to receive time information from an
accurate time source. Time information is sent over a path that often
traverses one or more intermediate clocks.

o In NTP, time information originated from a stratum 1 server can be
   distributed to stratum 2 servers, and in turn distributed from the
   stratum 2 servers to NTP clients. In this case, the stratum 2
   servers are a layer of intermediate clocks. These intermediate
   clocks are referred to as "secondary servers" in [NTPv4].

o In PTP, BCs and TCs are intermediate nodes used to improve the
   accuracy of time information conveyed between the grandmaster and
   the slaves.

A common rule of thumb in network security is that end-to-end
security is the best policy, as it secures the entire path between
the data originator and its receiver. The usage of intermediate nodes
implies that if a security mechanism is deployed in the network, a
hop-by-hop security scheme must be used, since intermediate nodes
must be able to send time information to the slaves, or to modify
time information sent through them.

This inherent property of using intermediate clocks increases the
system's exposure to internal threats, as there is a large number of
nodes that possess the security keys.

   Thus, there is a tradeoff between the achievable clock accuracy of a
   system, and the robustness of its security solution. On one hand high
   clock accuracy calls for hop-by-hop involvement in the protocol, also
   known as on-path support. On the other hand, a robust security
   solution calls for end-to-end data protection.

7.4. External Security Protocols and Time Protocols

   Time protocols are often deployed in systems that use security
   mechanisms and protocols.

   A typical example is the 3GPP Femtocell network [3GPP], where IPsec
   is used for securing traffic between a Femtocell and the Femto
   Gateway. In some cases, all traffic between these two nodes may be
   secured by IPsec, including the time protocol traffic. This use-case
   is thoroughly discussed in [IPsecSync].

   Another typical example is the usage of MACsec encryption ([MACsec])
   in L2 networks that deploy time synchronization [AvbAssum].

   The usage of external security mechanisms may affect time protocols
   as follows:

   o Timestamping accuracy can be affected, as described in 7.1.

   o If traffic is secured between two nodes in the network, no
     intermediate clocks can be used between these two nodes. In the
     [3GPP] example, if traffic between the Femtocell and the Femto
     Gateway is encrypted, then time protocol packets are necessarily
     transported over the underlying network without modification, and
     thus cannot enjoy the improved accuracy provided by intermediate
     clock nodes.

7.5. External Security Services Requiring Time

   Cryptographic protocols often use time as an important factor in the
   cryptographic algorithm. If a time protocol is compromised, it may
   consequently expose the security protocols that rely on it to various
   attacks. Two examples are presented in this section.

7.5.1. Timestamped Certificates

   Certificate validation requires the sender and receiver to be roughly
   time synchronized. Thus, synchronization is required for establishing
   security protocols such as IKEv2 and TLS.

An even stronger interdependence between a time protocol and a security mechanism is defined in [AutoKey], which defines mutual dependence between the acquired time information, and the authentication protocol that secures it. This bootstrapping behavior results from the fact that trusting the received time information requires a valid certificate, and validating a certificate requires knowledge of the time.

7.5.2. Time Changes and Replay Attacks

A successful attack on a time protocol may cause the attacked clocks to go back in time. The erroneous time may expose cryptographic algorithms that rely on time, as a node may use a key that was already used in the past and has expired.

8. Issues for Further Discussion

The Key distribution is outside the scope of this document. Although this is an essential element of any security system, it is outside the scope of this document.

9. Security Considerations

The security considerations of network timing protocols are presented throughout this document.

10. IANA Considerations

There are no new IANA considerations implied by this document.

11. Acknowledgments

The authors gratefully acknowledge Stefano Ruffini, Doug Arnold, Kevin Gross, Dieter Sibold, Dan Grossman, Laurent Montini, Russell Smiley, and Shawn Emery for their thorough review and helpful comments. The authors would also like to thank members of the TICTOC WG for providing feedback on the TICTOC mailing list.

This document was prepared using 2-Word-v2.0.template.dot.

12. References

12.1. Normative References

[KEYWORDS]   Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

12.2. Informative References

    [1588IPsec]   A. Treytl, B. Hirschler, "Securing IEEE 1588 by IPsec
                   tunnels - An analysis", in Proceedings of 2010
                   International Symposium for Precision Clock
                   Synchronization for Measurement, Control and
                   Communication, ISPCS 2010, pp. 83-90, 2010.

    [3GPP]   3GPP, "Security of Home Node B (HNB) / Home evolved
                   Node B (HeNB)", 3GPP TS 33.320 10.4.0 (work in
                   progress), 2011.

    [Anatomy]   C. Nachreiner, "Anatomy of an ARP Poisoning Attack",
                   2003.

    [AutoKey]   Haberman, B., Mills, D., "Network Time Protocol
                   Version 4: Autokey Specification", RFC 5906, June
                   2010.

    [AvbAssum]   D. Pannell, "Audio Video Bridging Gen 2 Assumptions",
                   IEEE 802.1 AVB Plenary, work in progress, May 2012.

    [DelayAtt]   T. Mizrahi, "A Game Theoretic Analysis of Delay
                   Attacks against Time Synchronization Protocols",
                   accepted, to appear in Proceedings of the
                   International IEEE Symposium on Precision Clock
                   Synchronization for Measurement, Control and
                   Communication, ISPCS, 2012.

    [Hack]   S. McClure, J. Scambray, G. Kurtz, Kurtz, "Hacking
                   exposed: network security secrets and solutions",
                   McGraw-Hill, 2009.

    [IEEE1588]   IEEE TC 9 Instrumentation and Measurement Society,
                   "1588 IEEE Standard for a Precision Clock
                   Synchronization Protocol for Networked Measurement and
                   Control Systems Version 2", IEEE Standard, 2008.

    [IPsec]   S. Kent, K. Seo, "Security Architecture for the
                   Internet Protocol", IETF, RFC 4301, 2005.

    [IPsecSync]   Y. Xu, "IPsec security for packet based
                   synchronization", IETF, draft-xu-tictoc-ipsec-
                   security-for-synchronization (work in progress), 2011.

   [MACsec]   IEEE 802.1AE-2006, "IEEE Standard for Local and
                  Metropolitan Area Networks - Media Access Control
                  (MAC) Security", 2006.

   [NTPv4]   Mills, D., Martin, J., Burbank, J., Kasch, W.,
                  "Network Time Protocol Version 4: Protocol and
                  Algorithms Specification", RFC 5905, June 2010.

   [SecPTP]   J. Tsang, K. Beznosov, "A security analysis of the
                  precise time protocol (short paper)," 8th
                  International Conference on Information and
                  Communication Security (ICICS 2006), pp. 50-59, 2006.

   [SecSen]   S. Ganeriwal, C. Popper, S. Capkun, M. B. Srivastava,
                  "Secure Time Synchronization in Sensor Networks", ACM
                  Trans. Info. and Sys. Sec., Volume 11, Issue 4, July
                  2008.

   [TM]   T. Mizrahi, "Time synchronization security using IPsec
                  and MACsec", ISPCS 2011, pp. 38-43, 2011.

   [Traps]   Treytl, A., Gaderer, G., Hirschler, B., Cohen, R.,
                  "Traps and pitfalls in secure clock synchronization"
                  in Proceedings of 2007 International Symposium for
                  Precision Clock Synchronization for Measurement,
                  Control and Communication, ISPCS 2007, pp. 18-24,
                  2007.

   [Tunnel]   A. Treytl, B. Hirschler, and T. Sauter, "Secure
                  tunneling of high precision clock synchronisation
                  protocols and other timestamped data", in Proceedings
                  of the 8th IEEE International Workshop on Factory
                  Communication Systems (WFCS), vol. ISBN 978-1-4244-
                  5461-7, pp. 303-313, 2010.

13. Contributing Authors

   Karen O'Donoghue
   ISOC

   Email: odonoghue@isoc.org

Authors' Addresses

    Tal Mizrahi
    Marvell
    6 Hamada St.
    Yokneam, 20692 Israel

    Email: talmi@marvell.com