TICTOC Working Group
Internet Draft
Intended status: Informational
Expires: December 2012
Tal Mizrahi
Marvell
Karen O'Donoghue
ISOC
June 17, 2012

TICTOC Security Requirements
draft-ietf-tictoc-security-requirements-02.txt

Abstract

As time synchronization protocols are becoming increasingly common and widely deployed, concern about their exposure to various security threats is increasing. This document defines a set of requirements for security solutions for time synchronization protocols, focusing on the IEEE 1588 and NTP. This document also discusses the security impacts of time synchronization protocol practices, the time synchronization performance implications of external security practices, the dependencies between other security services and time synchronization.

Copyright Notice

Table of Contents

1. Introduction 3
2. Conventions Used in this Document 4
   2.1. Terminology 4
   2.2. Terms & Abbreviations 5
3. Security Threats 5
   3.1. Threat Model 5
      3.1.1. Internal vs. External Attackers 5
      3.1.2. Man in the Middle (MITM) vs. Packet Injector 6
   3.2. Threat Analysis.. 6
      3.2.1. Packet Interception and Manipulation 6
      3.2.2. Spoofing 6
      3.2.3. Replay Attack 6
      3.2.4. Rogue Master Attack 6
      3.2.5. Packet Interception and Removal 7
      3.2.6. Packet Delay Manipulation 7
      3.2.7. Cryptographic Performance Attacks 7
      3.2.8. L2/L3 DoS Attacks 7
      3.2.9. Master Time Source Spoofing (e.g. GPS fraud) 7
   3.3. Threat Analysis Summary 7
4. Security Requirements 9
   4.1. Clock Identity Authentication 9
      4.1.1. Authentication of Masters 9
      4.1.2. Proventication of Masters 9
      4.1.3. Authentication of Slaves 10
      4.1.4. PTP: Authentication of Transparent Clocks.......... 10
      4.1.5. PTP: Authentication of Announce Messages 11
   4.2. Data integrity 11
      4.2.1. PTP: Hop-by-hop vs. End-to-end Integrity Protection 11
         4.2.1.1. Hop by Hop Integrity Protection 12
         4.2.1.2. End to End Integrity Protection 12

4.3. Availability 12
4.4. Replay Protection 13
4.5. Cryptographic Keys & Security Associations 13
4.5.1. Security Association 13
4.5.2. Unicast and Multicast 13
4.5.3. Key Freshness 14
4.6. Performance 14
4.7. Confidentiality... 14
4.8. Protection against packet delay attacks 15
4.9. Combining Secured with Unsecured Nodes 15
4.9.1. Secure Mode 15
4.9.2. Hybrid Mode 16
5. Summary of Requirements 16
6. Additional security implications 18
6.1. Security and on-the-fly Timestamping 18
6.2. Security and Two-Step Timestamping 18
6.3. Intermediate Clocks 19
6.4. The Effect of External Security Protocols on Time
Synchronization 19
6.5. External Security Services Requiring Time Synchronization20
7. Issues for Further Discussion 20
8. Security Considerations 20
9. IANA Considerations 20
10. Acknowledgments 20
11. References 21
11.1. Normative References 21
11.2. Informative References 21

1. Introduction

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

This document focuses on the security aspects of the Precision Time Protocol ([IEEE 1588]) and the Network Time Protocol ([NTPv4]). The Network Time Protocol was defined with an inherent security protocol, defined in [NTPv4] and in [AutoKey]. The IEEE 1588 includes an experimental security protocol, defined in Annex K of the standard, but this Annex was never formalized into a fully defined security protocol.

This document attempts to add clarity to the time synchronization protocol security requirements discussion by addressing a series of questions. It is expected that this document will evolve into

possibly two documents including one on requirements and one providing clarity around the additional questions raised below. Until the discussion has matured sufficiently, it will be captured in this document. The four primary questions addressed by this draft include:

(1) What are the threats that need to be addressed for the time synchronization 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 synchronization traffic path)

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

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

It is expected that the final version of this document will define a set of requirements for security solutions for time synchronization protocols, focusing on the IEEE 1588 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 that every security mechanism should comply to.

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 general term "clock" refers to masters, slaves and PTP Transparent Clocks (TC). The term "protocol packets" is refers generically to PTP and NTP messages.

2.2. Terms & Abbreviations

BC	Boundary Clock
MITM	Man In The Middle
NTP	Network Time Protocol
OC	Ordinary Clock
PTP	Precision Time Protocol
Secured clock	A clock that supports a security mechanism that complies to the requirements in this document
TC	Transparent Clock
Unsecured clock	A clock that does not support a security mechanism according to the requirments in this document

3. Security Threats

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

The literature is rich with security threats of time synchronization 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 synchronization protocol can be attacked by various types of attackers.

The analysis in this documents classifies attackers according to 2 criteria, as described in 3.1.1. and 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 synchronization protocol is secured either by an encryption or an authentication mechanism.

Internal attackers either have access to a trusted segment of the network, or possess the encryption or authentication keys. External attackers, on the other hand, do not have the keys, and are exposed only to the encrypted or authenticated traffic. 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.

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, while a traffic injector cannot intercept legitimate packets, but can record them, replay old messages, and generate its own traffic.

3.2. Threat Analysis

3.2.1. Packet Interception and Manipulation

A packet interception and manipulation attack results when a Man-In-The-Middle (MITM) attacker intercepts 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 attacker masquerades as a legitimate node in the network. For example, an attacker can impersonate the master, allowing malicious distribution of false timing information. As with packet interception and manipulation, this 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. 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 Rouge Master attack the attacker does not

fake its identity, but rather manipulates the master election process. 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.

3.2.5. Packet Interception and Removal

A packet interception and removal attack results when a Man-In-The-Middle attacker intercepts and drops protocol packets, preventing the destination node from receiving the timing information.

3.2.6. Packet Delay Manipulation

In a packet delay manipulation scenario, a Man-In-The-Middle attacker intercepts protocol packets, and relays them to their destination after adding a maliciously computed delay.

3.2.7. Cryptographic Performance Attacks

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

3.2.8. L2/L3 DoS Attacks

There are many possible Layer 2 and Layer 3 Denial of Service attacks. As the target's availability is compromised, the timing protocol is affected accordingly.

3.2.9. Master Time Source Spoofing (e.g. GPS fraud)

In time source spoofing, an attacker spoofs the accurate time source of the master. For example, if the master uses a GPS based clock as its reference source, an attacker can spoof the GPS satellites, causing the master to use a false reference time.

3.3. Threat Analysis Summary

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

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

The Impact column provides an intuition to the severity of each attack, and the relevant Attacker Type columns provide an intuition about the 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 False time - slaves align to a false time or frequency value due to the attack.

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

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

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

Attack	Impact			Attacker Type			
				Internal		Extenal	
	False Time	Accuracy Degrad.	DoS	MITM	Inj.	MITM	Inj.
Interception and manipulation	+			+			
Spoofing	+			+	+		
Replay attack	+			+	+		
Rogue master attack	+			+	+		
Interception and Removal		+		+		+	
Packet delay manipulation	+			+		+	
Crypt. performance attacks			+	+	+	+	+
DoS attacks			+	+	+	+	+

```
+------------------------------+-----+--------+----++----+----+----+----+
|Master Time source spoofing   |  +  |        |    || +  | +  | +  | +  |
+------------------------------+-----+--------+----++----+----+----+----+
```

Table 1 Threat Analysis - Summary

4. Security Requirements

4.1. Clock Identity Authentication

Requirement

The security mechanism MUST provide a means for each clock to authenticate the sender of a protocol packet.

Discussion

In the context of this document, authentication refers to:

- o Identification: verifying the identity of the peer clock.
- o Authorization: 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.

  The following subsections describe 4 distinct cases of clock authentication.

4.1.1. Authentication of Masters

Requirement

The security mechanism MUST support an authentication mechanism, allowing slave clocks to authenticate the identity of master clocks.

4.1.2. Proventication of Masters

Requirement

The security mechanism MUST support a proventication mechanism, to be used in cases where end-to-end authentication is not possible.

Discussion

Slaves and transparent clocks authenticate masters in order to ensure the authenticity of the time source.

In some cases a slave is connected to an intermediate master, that is not the primary time source. For example, in PTP a slave can be connected to a Boundary Clock (BC), 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 master, and the intermediate master authenticates the primary master. This inductive authentication process is referred to in [AutoKey] as proventication.

4.1.3. Authentication of Slaves

Requirement

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

Discussion

Slaves are authenticated by masters in order to verify that the slave is authorized to receive timing services from the master.

Authentication of slaves prevents unauthorized clocks from receiving time services, and also reduces unnecessary load on the master clock, by preventing the master from serving unauthorized clocks. It could be argued that the authentication of slaves could put a higher load on the master then serving the unauthorized clock. This tradeoff will need to be discussed further.

4.1.4. PTP: Authentication of Transparent Clocks

Requirement

The security mechanism for PTP SHOULD provide a means for a master to authenticate the TCs.

Discussion

Transparent clocks are authenticated by peer masters, slaves and TCs.

Authentication of TCs, much like authentication of slaves, reduces unnecessary load on the master clock and peer TCs, by preventing the master from serving unauthorized clocks. It also prevents malicious TCs from attacking the protocol by manipulating the correctionField. It could also be argued that the authentication could result in a higher load then merely serving the unauthorized devices. This tradeoff will need to be discussed further.

4.1.5. PTP: Authentication of Announce Messages

Requirement

The security mechanism for PTP MUST support authentication of Announce messages.

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 malicious master attacks.

Note, that this subsection specifies a requirement that is not necessarily included in 4.1.1. or in 4.1.3. , since the BMCA is initiated before clocks have been defined as masters or slaves.

4.2. Data integrity

Requirement

The security mechanism MUST protect the integrity of protocol packets.

Discussion

While subsection 4.1. refers to ensuring WHO sent the 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.

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

Requirement

A security mechanism for PTP MUST support hop-by-hop integrity protection.

Requirement

A security mechanism for PTP SHOULD support end-to-end integrity protection.

Discussion

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.

4.2.1.1. 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.
- 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 malicious TCs to modify protocol packets.

4.2.1.2. 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 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.

4.3. Availability

Requirement

The security mechanism MUST protect the time synchronization protocol from DoS attacks by external attackers.

Discussion

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

spoofing attack (3.2.2.) or the rogue master attack (3.2.4.), causing denial of service to the attackee. An authentication mechanism (4.1.) limits these attacks strictly to internal attackers, and thus prevents external attackers from performing them.

DoS attacks at lower layers of the protocol stack (3.2.8.) can still be implemented by external attackers even in the presence of an authentication mechanism.

4.4. Replay Protection

Requirement

Protocol messages MUST be resistant to replay attacks.

4.5. Cryptographic Keys & Security Associations

4.5.1. Security Association

Requirement

The security protocol MUST support an association protocol where:

o Two or more clocks authenticate each other.

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

Discussion

The security requirements in 4.1. and 4.2. require usage of cryptographich mechanisms, deploying cryptographic keys. A security association is an essential building block in these mechanisms.

4.5.2. Unicast and Multicast

Requirement

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

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.

4.5.3. Key Freshness

Requirement

The cryptographic keys MUST be refreshed periodically.

Requirement

The association protocol MUST be invoked periodically, where each instance of the association protocol MUST produce a different session key.

4.6. Performance

Requirement

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

Requirement

The mechanism SHOULD be relatively lightweight, as client restrictions often dictate a low processing and memory footprint, and because the server may have extensive fan-out.

Requirement

The mechanism also SHOULD not require excessive storage of client state in the master, nor significantly increase bandwidth consumption.

Discussion

Note that the performance requirements refer to a time-synchronization-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.

4.7. Confidentiality

Requirement

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

Discussion

In the context of time synchronization, 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 payment for time synchronization services, but these cases are rather esoteric.

Confidentiality can also prevent an MITM attacker from identifying protocol packets. Thus, confidentiality can assist in protecting the timing protocol against packet delay attacks, where the attacker selectively adds delay to time protocol packets.

4.8. Protection against packet delay attacks

Requirement

The security mechanism MAY include a means to detect packet delay attacks.

Requirement

The security mechanism MAY include a protection switching mechanism that allows a node that detects a delay attack to switch over to a secondary master.

4.9. Combining Secured with Unsecured Nodes

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

4.9.1. Secure Mode

Requirement

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

Discussion

While the requirement in this subsection is a bit similar to the one in 4.1. , it explicitly defines the secure mode, as opposed to the hybrid mode presented in the next subsection.

4.9.2. Hybrid Mode

Requirement

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

Discussion

The hybrid mode allows both secured and unsecured clocks to take part in the synchronization protocol.

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.

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 clock can receive protocol packets from either secured clocks, or unsecured clocks.

Note that the security scheme in [NTPv4] with [AutoKey] does not satisfy this requirement, since nodes prefer the server with the best clock, and 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.

5. Summary of Requirements

Section	Requirement	Type
4.1.	Authentication of sender.	MUST

	Authentication of master.	MUST
	Proventication.	MUST
	Authentication of slaves.	SHOULD
	PTP: Authentication of TCs.	SHOULD
	PTP: Authentication of Announce messages.	SHOULD
4.2.	Integrity protection.	MUST
	PTP: hop-by-hop integrity protection.	MUST
	PTP: end-to-end integrity protection.	SHOULD
4.3.	Protection against DoS attacks.	MUST
4.4.	Replay protection.	MUST
4.5.	Security association.	MUST
	Unicast and multicast associations.	MUST
	Key freshness.	MUST
4.6.	Performance: no degradation in quality of time transfer.	MUST
	Performance: lightweight.	SHOULD
	Performance: storage, bandwidth.	MUST
4.7.	Confidentiality protection.	MAY
4.8.	Protection against delay attacks.	MAY
4.9.	Secure mode.	MUST

```
|           | Hybrid mode.                        | MAY            |
+-----------+-------------------------------------+----------------+
```

Table 2 Summary of Security Requirements

6. Additional security implications

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

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

6.1. Security and on-the-fly Timestamping

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

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

- o During transmission the security protocol must be applied after integrating the timestamp into the packet.

- o During reception, the encryption or integrity check must be performed before modifying the packet with the time of reception.

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, in some cases it may introduce non-deterministic latency that causes accuracy degradation. These performance aspects have been analyzed in the literature, e.g., in [1588IPsec] and [Tunnel].

6.2. Security and Two-Step Timestamping

PTP supports a two-step mode of operation, where the time of transmission and the time of reception of protocol packets are measured without modifying the packets. As opposed to one-step mode, two step timestamping can be performed at the physical interface even in the presence of a security mechanism.

Note that if an encryption mechanism 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 synchronization packets.

6.3. Intermediate Clocks

A time synchronization 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.

- 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, all intermediate nodes must be exposed to the security key since they must be able to send time information to the slaves, or to modify time information sent through them.

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

6.4. The Effect of External Security Protocols on Time Synchronization

Time synchronization 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. All traffic between these two nodes is secured by IPsec, including the time synchronization protocol traffic. This use-case is thoroughly discussed in [IPsecSync].

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

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

- o Timestamping accuracy can be affected, as described in 6.1.

- 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 sent over the underlying network without modification, and thus cannot enjoy the improved accuracy provided by intermediate clock nodes.

6.5. External Security Services Requiring Time Synchronization

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

An even stronger interdependence between a time synchronization 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.

7. Issues for Further Discussion

- o The key distribution is outside the scope of this document. Although this is a cardinal element in any security system, it is not a security requirement, and is thus not described here.

8. Security Considerations

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

9. IANA Considerations

There are no new IANA considerations implied by this document.

10. Acknowledgments

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

11. References

11.1. Normative References

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

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

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

11.2. Informative References

[IEEE 1588] IEEE TC 9 Test and Measurement Society 2000, "1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Standard, 2008.

[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.

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

[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.

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

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

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

[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.

[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.

Authors' Addresses

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692 Israel

Email: talmi@marvell.com

Karen O'Donoghue
7167 Goby Lane
King George, VA 22485

Email: odonoghue@isoc.org