

NTP Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2014

D. Sibold
PTB
S. Roettger
TU-BS
K. Teichel
PTB
October 18, 2013

Network Time Security
draft-ietf-ntp-network-time-security-01.txt

Abstract

This document describes the Network Time Security (NTS) protocol that enables secure authentication of time servers using Network Time Protocol (NTP) or Precision Time Protocol (PTP). Its design considers the special requirements of precise timekeeping, which are described in Security Requirements of Time Protocols in Packet Switched Network [I-D.ietf-tictoc-security-requirements].

Requirements Language

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 RFC 2119 [RFC2119].

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 <http://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 April 21, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Security Threats	3
3. Objectives	3
4. Terms and Abbreviations	4
5. NTS Overview	4
5.1. Symmetric and Client/Server Mode	4
5.2. Broadcast Mode	5
6. Protocol Sequence	5
6.1. Association Message	5
6.2. Certificate Message	5
6.3. Cookie Message	6
6.4. Broadcast Parameter Message	6
6.5. Time Request Message	7
6.6. Broadcast Message	7
6.7. Server Seed Refresh	7
7. Hash Algorithms and MAC Generation	8
7.1. Hash Algorithms	8
7.2. MAC Calculation	8
8. Server Seed Considerations	8
8.1. Server Seed Algorithm	9
8.2. Server Seed Live Time	9
9. IANA Considerations	9
10. Security Considerations	9
10.1. Initial Verification of the Server Certificates	9
10.2. Revocation of Server Certificates	9
10.3. Usage of NTP Pools	10
10.4. Denial-of-Service in Broadcast Mode	10
11. Acknowledgements	10
12. References	10
12.1. Normative References	10
12.2. Informative References	11
Appendix A. TICTOC Security Requirements	11
Appendix B. Broadcast Mode	12
Authors' Addresses	12

1. Introduction

Time synchronization protocols are more and more utilized to synchronize clocks in networked infrastructures. The reliable performance of such infrastructures can be degraded seriously by successful attacks against the time synchronization protocol. Therefore, time synchronization protocols applied in critical infrastructures have to provide security measures to defeat possible adversaries. Consequently, the widespread Network Time Protocol (NTP) [RFC5905] was supplemented by the autokey protocol [RFC5906] which shall ensure authenticity of the NTP server and integrity of the protocol packets. Unfortunately, the autokey protocol exhibits various severe security vulnerabilities as revealed in a thorough analysis of the protocol [Roettger]. For the Precision Time Protocol (PTP) Annex K of the standard document IEEE 1588 [IEEE1588] defines an informative security protocol that is still in experimental state.

Because of autokey's security vulnerabilities and the absence of a standardized security protocol for PTP these protocols cannot be applied in environments in which compliance requirements demand authenticity and integrity protection. This document specifies a security protocol which ensures authenticity of the time server via a Public Key Infrastructure and integrity of the time synchronization protocol packets and which therefore enables the usage of NTP and PTP in such environments.

The protocol is specified with the prerequisite in mind that precise timekeeping can only be accomplished with stateless time synchronization communication, which excludes standard security protocols like IPsec or TLS. This prerequisite corresponds with the requirement that a security mechanism for timekeeping must be designed in such a way that it does not degrade the quality of the time transfer [I-D.ietf-tictoc-security-requirements].

Note:

It is intended to formulate the protocol to be applicable to NTP as well as PTP. In the current state the draft focuses on the application to NTP.

2. Security Threats

A profound analysis of security threats and requirements for NTP and PTP can be found in the I-D [I-D.ietf-tictoc-security-requirements].

3. Objectives

The objectives of the NTS specifications are as follows:

- o Authenticity: NTS enables the client to authenticate its time server.
- o Integrity: NTS protects the integrity of time synchronization protocol packets via a message authentication code (MAC).
- o Confidentiality: NTS does not provide confidentiality protection of the time synchronization packets.
- o Modes of operation: All operational modes of NTP are supported.
- o Operational modes of PTP should be supported as far as possible.
- o Hybrid mode: Both secure and insecure communication modes are possible for NTP servers and clients, respectively.
- o Compatibility:
 - * Unsecured NTP associations shall not be affected.
 - * An NTP server that does not support NTS shall not be affected by NTS authentication requests.

4. Terms and Abbreviations

- o TESLA: Time efficient stream loss-tolerant authentication

5. NTS Overview

5.1. Symmetric and Client/Server Mode

Authenticity of the time server is verified once by a Public Key Infrastructure. Authenticity and integrity of the NTP packets are then ensured by a Message Authentication Code (MAC), which is attached to the NTP packet. The calculation of the MAC includes the whole NTP packet and the cookie which is shared between client and server. It is calculated according to:

$$\text{cookie} = \text{MSB}_{128} (\text{H}(\text{server seed} || \text{H}(\text{public key of client}))),$$

where `||` indicates concatenation and in which H is a hash algorithm. The function `MSB128` cuts off the 128 most significant bits of the result of the hash function. The server seed is a 128 bit random value of the server, which has to be kept secret. The cookie thus never changes as long as the server seed stays the same. The server seed has to be refreshed periodically in order to provide key freshness as required in [I-D.ietf-tictoc-security-requirements]. The server does not keep a state of the client. Therefore it has to

recalculate the cookie each time it receives a request from the client. To this end, the client has to attach the hash value of its public key to each request (see Section 6.5).

5.2. Broadcast Mode

Just as in the case of the client server mode and symmetric mode, authenticity and integrity of the NTP packets are ensured by a MAC, which is attached to the NTP packet by the sender. The verification of the authenticity is based on the TESLA protocol [RFC4082]. TESLA is based on a one-way chain of keys, where each key is the output of a one-way function applied on the previous key in the chain. The last element of the chain is shared securely with all clients. The server splits time into intervals of uniform duration and assigns each key to an interval in reverse order, starting with the penultimate. At each time interval, the server sends an NTP broadcast packet appended by a MAC, calculated using the corresponding key, and the key of the previous interval. The client verifies the MAC by buffering the packet until the disclosure of the key in the next interval. In order to be able to verify the validity of the key, the client has to be loosely time synchronized to the server. This has to be accomplished during the initial client server exchange between broadcast client and server. For a more detailed description of the TESLA protocol see Appendix B.

6. Protocol Sequence

6.1. Association Message

The protocol sequence starts with the association message, in which the client sends an NTP packet with an extension field of type association. It contains the hostname of the client and a status word which contains the algorithms used for the signatures and the status of the connection. The response contains the hostname of the server and the algorithms for the signatures. The server notifies the cryptographic hash algorithms which it supports.

6.2. Certificate Message

In this step, the client receives the certification chain up to a trusted authority (TA). To this end, the client requests the certificate for the subject name (hostname) of the NTP server. The response contains the certificate with the issuer name. If the issuer name is different from the subject name, the client requests the certificate for the issuer. This continues until it receives a certificate in which the subject name and the issuer name are identical, which indicates that it is issued by a TA. The client then checks that the issuer is indeed on its list of issuers which

are accepted as TAs. The client has to check that each issuer in the certificate chain is authorized to issue new certificates. To this end, the certificates have to include the X.509v3 extension field "CA:TRUE". With the established certification chain the client is able to verify the server signatures and, hence, the authenticity of the server messages with extension fields is ensured.

Discussion:

Note that in this step the client validates the authenticity of its NTP server only. It does not recursively validate the authenticity of each NTP server on the time synchronization chain. But each NTP server on the time synchronization chain validates the NTP server to which it is synchronized. This conforms to the recursive authentication requirement in the TICTOC security requirements [I-D.ietf-tictoc-security-requirements].

6.3. Cookie Message

The client requests a cookie from the server. It selects a hash algorithm from the list of algorithms supported by the server. The request includes its public key and the selected hash algorithm. The hash of the public key is used by the server to calculate the cookie (see Section 5.1). The response of the server contains the cookie and a signature of the cookie signed with the server's private key, both encrypted with the client's public key.

6.4. Broadcast Parameter Message

In the broadcast mode the client requests the following information from the server:

- o the last key of the one-way key chain,
- o the disclosure schedule of the following keys. This contains:
 - * time interval duration, time at which the next time interval will start and its associated index,
 - * key disclosure delay (number of time intervals for which a key is valid).

The server will sign all transmitted properties so that the client is able to verify their authenticity. For this packet exchange a new extension field "broadcast parameters" is used. The client synchronizes its time with the server in the client server mode and saves an upper bound of its time offset with respect to the time of the server. See Appendix B for more details.

6.5. Time Request Message

The client request includes a new extension field "time request" which contains the hash of its public key, a 128-bit nonce, and the chosen hash algorithm. The server needs the hash of the public key and the notified hash algorithm to recalculate the cookie for the client. The response is a NTP packet with a new extension field "time response" which contains the nonce and a MAC generated over the time synchronization data, the cookie and the nonce.

6.6. Broadcast Message

In broadcast mode the NTP packet includes a new extension field "broadcast message" which contains the disclosed key of the previous disclosure interval (current time interval minus disclosure delay). The NTP packet is appended by a MAC, calculated with the key for the current time interval. When a client receives a broadcast message it has to perform the following tests:

- o Proof that the MAC is based on a key that is not yet disclosed. This is achieved via a disclosure schedule, so this is where loose time synchronization is required. If verified the packet will be buffered for later authentication otherwise it has to be discarded. Note that the time information included in the packet will not be used for synchronization until their authenticity could be verified.
- o The client checks whether it already knows the disclosed key. If so, the packet is discarded to avoid a buffer overrun. If not, the client verifies that the disclosed key belongs to the one-way key chain by applying the one-way function until equality with a previous disclosed key is verified. If falsified the packet has to be discarded.
- o If the disclosed key is legitimate the client verifies the authenticity of any packet that it received during the corresponding time interval. If authenticity of a packet is verified it is released from the buffer and the packet's time information can be utilized. If the verification fails authenticity is no longer given. In this case the client MUST request authentic time from the server by means of a unicast time request message.

See RFC 4082[RFC4082] for a detailed description of the packet verification process.

6.7. Server Seed Refresh

According to the requirements in [I-D.ietf-tictoc-security-requirements] the server has to refresh its server seed periodically. As a consequence the cookie used in the time request messages becomes invalid. In this case the client cannot verify the attached MAC and has to respond accordingly by re-initiating the protocol with a cookie request (Section 6.3). This is true for the unicast and broadcast mode, respectively.

Additionally, in broadcast mode the client has to restart the broadcast sequence with a time request message if the one-way key chain expires.

During certificate message exchange the client reads the expiration date of the period of validity of the server certificate. The client MAY restart the protocol sequence with the association message before the server certificate expires.

7. Hash Algorithms and MAC Generation

7.1. Hash Algorithms

Hash algorithms are used at different points: calculation of the cookie and the MAC, and hashing of the public key. The client selects the hash algorithm from the list of hash algorithms which are supported by the server. This list is notified during the association message exchange (Section 6.1). The selected algorithm is used for all hashing processes in the protocol.

In the broadcast mode hash algorithm are used as pseudo random functions to construct the one-way key chain.

The list of the hash algorithms supported by the server has to fulfill the following requirements:

- o it MUST NOT include MD5 or weaker algorithms,
- o it MUST include SHA-256 or stronger algorithms.

7.2. MAC Calculation

For the calculation of the MAC client and server are using a Keyed-Hash Message Authentication Code (HMAC) approach [RFC2104]. The HMAC is generated with the hash algorithm specified by the client (see Section 7.1).

8. Server Seed Considerations

The server has to calculate a random seed which has to be kept secret and which has to be changed periodically. The server has to generate a seed for each supported hash algorithm.

8.1. Server Seed Algorithm

8.2. Server Seed Live Time

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

10. Security Considerations

10.1. Initial Verification of the Server Certificates

The client has to verify the validity of the certificates during the certification message exchange (Section 6.2). Since it generally has no reliable time during this initial communication phase, it is impossible to verify the period of validity of the certificates. Therefore, the client **MUST** use one of the following approaches:

- o The validity of the certificates is preconditioned. Usually this will be the case in corporate networks.
- o The client ensures that the certificates are not revoked. To this end, the client uses the Online Certificate Status Protocol (OCSP) defined in [RFC6277].
- o The client requests a different service to get an initial time stamp in order to be able to verify the certificates' periods of validity. To this end, it can, e.g., use a secure shell connection to a reliable host. Another alternative is to request a time stamp from a Time Stamping Authority (TSA) by means of the Time-Stamp Protocol (TSP) defined in [RFC3161].

10.2. Revocation of Server Certificates

According to Section 6.7 it is the client's responsibility to initiate a new association with the server after the server's certificate expires. To this end the client reads the expiration date of the certificate during the certificate message exchange (Section 6.2). Besides, certificates may also be revoked prior to the normal expiration date. To increase security the client **MAY** verify the state of the server's certificate via OCSP periodically.

10.3. Usage of NTP Pools

The certification based authentication scheme described in Section 6 is not applicable to the concept of NTP pools. Therefore, NTS is not able to provide secure usage of NTP pools.

10.4. Denial-of-Service in Broadcast Mode

TESLA authentication buffers packets for delayed authentication. This makes the protocol vulnerable to flooding attacks, causing the client to buffer excessive numbers of packets. To add stronger DoS protection to the protocol client and server SHALL use the "Not Re-using Keys" scheme of TESLA as pointed out in section 3.7.2 of RFC 4082 [RFC4082]. In this scheme the server never uses a key for the MAC generation more than once. Therefore the client can discard any packet that contains a disclosed key it knows already, thus preventing memory flooding attacks.

Note, an alternative approach to enhance TESLA's resistance against DoS attacks involves the addition of a group MAC to each packet. This requires the exchange of an additional shared key common to the whole group. This adds additional complexity to the protocol and hence is currently not considered in this document.

11. Acknowledgements

12. References

12.1. Normative References

- [IEEE1588] IEEE Instrumentation and Measurement Society. TC-9 Sensor Technology, "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems", 2008.
- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3161] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001.

- [RFC4082] Perrig, A., Song, D., Canetti, R., Tygar, J.D., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, June 2005.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [RFC5906] Haberman, B. and D. Mills, "Network Time Protocol Version 4: Autokey Specification", RFC 5906, June 2010.
- [RFC6277] Santesson, S. and P. Hallam-Baker, "Online Certificate Status Protocol Algorithm Agility", RFC 6277, June 2011.

12.2. Informative References

- [I-D.ietf-tictoc-security-requirements]
Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", draft-ietf-tictoc-security-requirements-05 (work in progress), April 2013.
- [Roettger]
Roettger, S., "Analysis of the NTP Autokey Procedures", February 2012.

Appendix A. TICTOC Security Requirements

The following table compares the NTS specifications against the TICTOC security requirements [I-D.ietf-tictoc-security-requirements].

Section	Requirement from I-D tictoc security-requirements-05	Requirement level	NTS
5.1.1	Authentication of Servers	MUST	OK
5.1.1	Authorization of Servers	MUST	-
5.1.2	Recursive Authentication of Servers (Stratum 1)	MUST	NO
5.1.2	Recursive Authorization of Servers (Stratum 1)	MUST	-
5.1.3	Authentication and Authorization of Slaves	MAY	-
5.2	Integrity protection.	MUST	OK
5.3	Protection against DoS attacks	SHOULD	OK
5.4	Replay protection	MUST	OK
5.5.1	Key freshness.	MUST	OK
5.5.2	Security association.	SHOULD	OK

5.5.3	Unicast and multicast associations.	SHOULD	OK
5.6	Performance: no degradation in quality of time transfer.	MUST	OK
	Performance: lightweight computation	SHOULD	OK
	Performance: storage, bandwidth	SHOULD	OK
5.7	Confidentiality protection	MAY	NO
5.8	Protection against Packet Delay and Interception Attacks	SHOULD	NA*)
5.9.1	Secure mode	MUST	-
5.9.2	Hybrid mode	MAY	-

*) Ensured by NTP via multi-source configuration.

Comparison of NTS specification against TICTOC security requirements.

Appendix B. Broadcast Mode

Authors' Addresses

Dieter Sibold
 Physikalisch-Technische Bundesanstalt
 Bundesallee 100
 Braunschweig D-38116
 Germany

Phone: +49-(0)531-592-8420
 Fax: +49-531-592-698420
 Email: dieter.sibold@ptb.de

Stephen Roettger
 Technische Universitaet Braunschweig
 Email: stephen.roettger@googlemail.com

Kristof Teichel
 Physikalisch-Technische Bundesanstalt
 Bundesallee 100
 Braunschweig D-38116
 Germany

Email: kristof.teichel@ptb.de

