Network Time Security

draft-ietf-ntp-network-time-security-11
draft-ietf-ntp-using-nts-for-ntp-02
draft-ietf-ntp-cms-for-nts-message-04

Dr. Dieter Sibold   Kristof Teichel   Stephen Röttger

IETF 94 (Yokohama, Japan) 1–6, 2015

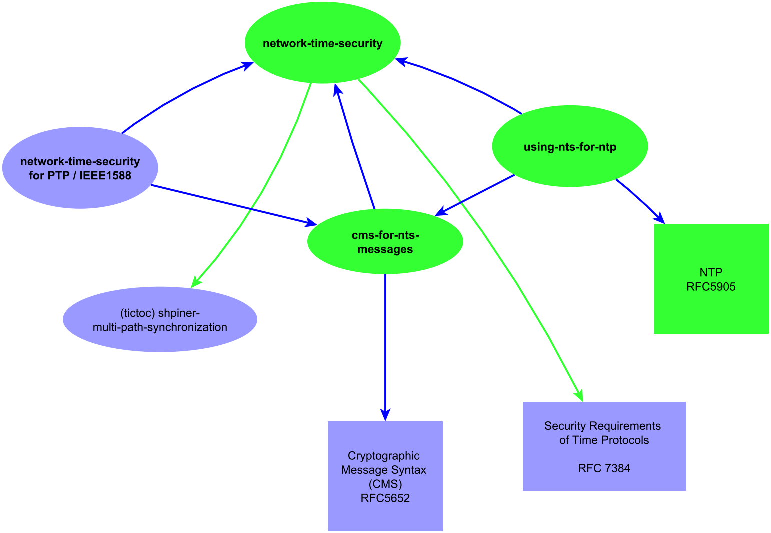
Outline

- History
- Document's Dependency Graph
- Scope
- Progress/Major Changes
    - Implementation
    - Implementation Status
    - Major Changes
- Open Issues
- Next Steps

History

- **IETF 83:** Presentation of security issues of RFC 5906 (autokey)
- **IETF 84:** Presentation of plan for a new autokey standard
- **IETF 85–86:** I-D "draft-sibold-autokey-*nn*"
- **IETF 87–90:** I-D "draft-ietf-ntp-network-time-security-*nn*"
- **Since IETF 92:**
  - draft-ietf-ntp-network-time-security-*NN*
  - draft-ietf-ntp-cms-for-nts-message-*NN*
  - draft-ietf-ntp-using-nts-for-ntp-*NN*

New Structure: Overview


network-time-security
network-time-security for PTP / IEEE1588
using-nts-for-ntp
cms-for-nts-messages
(tictoc) shpiner-multi-path-synchronization
NTP RFC5905
Cryptographic Message Syntax (CMS) RFC5652
Security Requirements of Time Protocols RFC 7384

Scope

Network Time Security provides:

- Authenticity of time servers
- Ability to authenticate time clients to the server
- Ability to perform authorization checks for time clients and servers
- Integrity of synchronization data packets
- Conformity with TICTOC's Security Requirements (RFC 7384)
- Support for NTP
- Ability for other time synchronization protocols, e. g. PTP

Implementation

Two independent implementations from:

- Network Time Foundation
- University of Applied Science Wolfenbüttel, Germany

Currently both implementations focus on the realization of NTS for NTP

- Implementation of the authentication frame work and the secure cookie exchange
- Securing the time request and time response messages of the unicast associations

Implementation Status

Network Time Foundation

- Authentication framework (association, cookie exchange)
  - coded
  - testing in progress
- Unicast time message exchange
  - coding in progress
- Allocation of OID values
  - testing using *unofficial* values
  - NTF has applied for a Private Enterprise Number to host OID assignments

Implementation Status

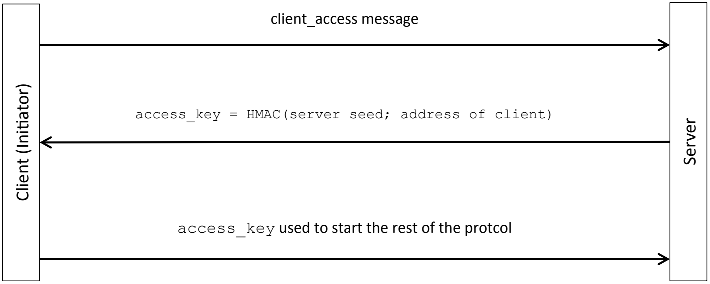
University of Applied Science Wolfenbüttel

- Currently: trying out the necessary OpenSSL core functions
- Next item: encoding of ASN.1 and CMS structures
- After that: usage for NTS message exchanges
- Deadline: by April 2016

Major Changes in the drafts

Network Time Security draft

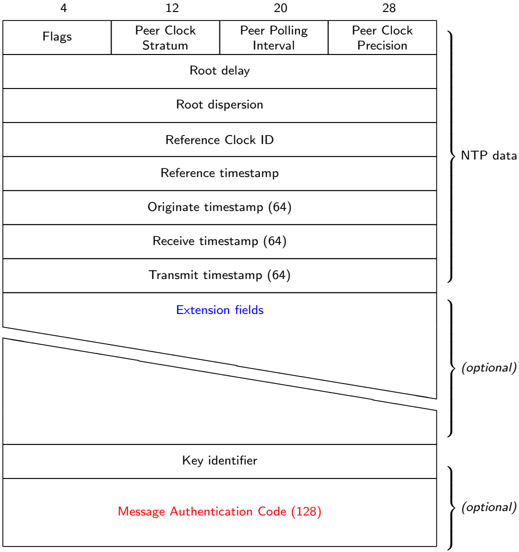
The authentication scheme described in Appendix B is enhanced by a message exchange similar to a Photuris cookie exchange, for protection against *amplification DoS* attacks (Appendix B.2)


client_access message
access_key = HMAC(server seed; address of client)
access_key used to start the rest of the protcol
Client (Initiator)
Server

Major Changes in the drafts

NTS for NTP draft

- An extension field instead the *classical* MAC field contains the MAC
- The extension fields' *type* flags now signal the included content as being NTS-related (with NTS version number)



4	12	20	28
Flags	Peer Clock Stratum	Peer Polling Interval	Peer Clock Precision

Root delay

Root dispersion

Reference Clock ID

Reference timestamp

Originate timestamp (64)

Receive timestamp (64)

Transmit timestamp (64)

NTP data

Extension fields

(optional)

Key identifier

Message Authentication Code (128)

(optional)

Open Issues

NTP's *Kiss-O'-Death-Packet*

KoD problematic revealed in a security analysis of NTP by Boston University (`http://www.cs.bu.edu/~goldbe/papers/NTPattack.pdf`)

- An off-path adversary can persuade a server to send a KoD packet to a client which delays its next time query for day or even years
- NTS does not currently protect against this attack
- NTS will protect against this attack if the *time request* message is authenticated and an NTP server only sends KoD packets in case of NTS secured associations
- Authentication for NTS' *time request* message is feasible (analogous to the *time response* message). This will impact
  - NTS' main draft
  - NTS for NTP draft

Open Issues

Data Structure issues

- Discussion on usage of CMS *SignedData* type for transporting payload and certificate, but without an actual signature.
- Discussion on where to place OIDs for the NTS objects in the extension fields (additional ASN.1 layer?).

These issues are most likely addressed in the draft *CMS for NTS messages*

Next Steps

- Implementation
  - Finalization and testing of the unicast associations
  - Considerations regarding Broadcast/Multicast mode
- KoD problematic
  - Introduction of authenticated *time request* message (NTS draft)
  - Description of NTP's server state machine (NTS for NTP draft)
- Last call for the NTS draft