Forschungsbericht WS 2017/2018

Network Time Security (NTS) for NTP (Update)

Prof. Dr.-Ing. Rainer Bermbach

Einleitung

Wie in vorausgegangenen Anträgen und Berichten schon ausgeführt, benötigen moderne vernetzte Systeme häufig eine genaue Uhrzeit, die meist per Network Time Protocol (NTP) [1] oder Precision Time Protocol (PTP) [2] verteilt wird. Da diese Protokolle aber nicht oder nicht ausreichend gegen Manipulationen und Angriffe geschützt sind, benötigen sicherheitskritische Anwendungen zusätzliche Mechanismen zur Absicherung. Das als IETF Draft verfügbare Protokoll Network Time Security (NTS) bietet eine solche Sicherung, derzeit primär für NTP.

Die Arbeiten [3, 4] analysierten die Protokollvorgaben, brachten Verbesserungsvorschläge ein und implementierten schließlich den letzten Stand des Drafts [5] vom September 2016 (die weltweit erste Realisierung von NTS [6]). Damit konnte gezeigt werden, dass die grundsätzlichen Überlegungen korrekt waren und das Protokoll in der Praxis seinen Zweck gut erfüllt.

Der Draft ging mittlerweile in den sog. Last Call vor dem Übergang in den RFC-Status. Hier gab es unvermutet etliche Kritikpunkte am bisherigen Konzept von NTS seitens der IETF Security Group (die man sich natürlich zu einem früheren Zeitpunkt gewünscht hätte). Diese Kritik führte zu einem überarbeiteten Protokollvorschlag [7], der mehr Standardprotokolle bei der Schlüsselaushandlung und der anschließenden Zeitpaketübertragung vorsieht. Die Grundüberlegung war, dass bewährte Standardverfahren bereits vielfach geprüft und realisiert seien und daher vermutlich weniger Fehler und Angriffspunkte böten, als eine komplette Neuentwicklung aller Teile eines Sicherungsprotokolls. Zudem wurde eine Reihe weiterer Argumente angeführt (z.B. dass die IP-Fragmentierung unsicher sei), die nicht immer nachvollziehbar waren und u.U. eher politisch oder wirtschaftlich motiviert schienen. Der neue Draft-Entwurf macht leider eine fast vollständige Neuimplementierung notwendig, will man die Brauchbarkeit des Konzeptes beweisen.

Analyse der Protokollvorgaben

Wie geplant begannen die Arbeiten im Projektzeitraum mit einer intensiven Auseinandersetzung und Analyse des geänderten Protokoll-Drafts von NTS. Dies sollte sich aber noch einige Male wiederholen, da sich bis zum Berichtszeitpunkt die Version von draft-ietf-ntp-using-nts-for-ntp-08 bis draft-ietf-ntp-using-nts-for-ntp-11 [8] weiterentwickelte. Die Kooperation mit der PTB und die teilweise Mitarbeit in der Working Group erleichterten das Verständnis der vielen Änderungen. Die derzeitige Version 11 spezifiziert nur noch die Modi 3 (Client-Mode) und 4 (Server-Mode) des NTP-Protokolls, da sich gezeigt hat, dass die anderen Modi abweichende und zum Teil gegenläufige Anforderungen besitzen und diese besser zu einem späteren Zeitpunkt in weiteren RFC(s) spezifiziert werden sollten.

Die Kommunikation zwischen Client und Zeitserver erfolgt nun in zwei getrennten Phasen. Zuerst erfolgt der TLS-basierte Key Exchange (NTS Key Establishment Protocol) und, wenn die Schlüssel für die Sicherung der nachfolgenden Kommunikation gewonnen wurden, beginnt der

geschützte Informationsaustausch zwischen Client und Server (s. Abb. 1).

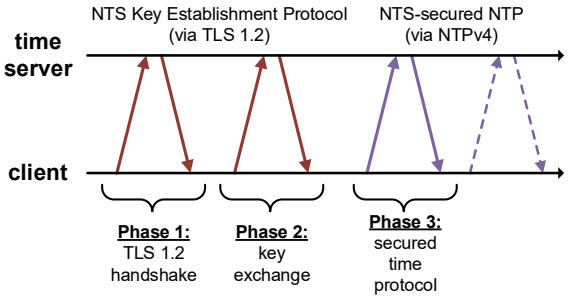

Abb. 1 Phasen des Informationsaustauschs im NTS-Protokoll

Das NTS-Key-Establishment-Protokoll

In einer ersten Phase geschieht der Aufbau einer TLS-Verbindung (nur TLS 1.2 und größer erlaubt) über TCP, so dass eine verschlüsselte Verbindung zur Verfügung steht. Die TLS-Verbindung ermöglicht die gegenseitige Identifikation der beiden Kommunikationspartner. In der zweiten Phase ereignet sich der sog. Key Exchange, bei dem die Schlüssel C2S (Client to Server) und S2C (Server to Client) mit Standardverfahren [9] gewonnen werden. Dabei tauschen Client und Server auch die verfügbaren AEAD-Verfahren (Authenticated Encryption with Associated Data) aus, woraus der Server eines auswählt. Mindestens unterstützt werden muss von beiden der Algorithmus AEAD_AES_SIV_-CMAC_256 [10]. Ebenfalls erhält der Client einen Vorrat an Cookies, mit denen er anschließend die Zeitanfragen authentisieren kann. Außerdem enthalten die Cookies jeweils den ausgehandelten AEAD-Algorithmus, die Keys C2S und S2C sowie eine zufällige Nonce. Mit einem vom Server gewählten Key und einem ebenfalls gewählten AEAD-Algorithmus (auch verschieden vom ausgehandelten) verschlüsselt der Server die Cookie-Daten. Gibt der Client nun ein solches Cookie seiner Zeitanfrage mit, kann der Server daraus alle relevanten Kommunikationsdaten ableiten und daher zustandslos betrieben werden. Dies ist eine wesentliche Voraussetzung, um einen ressourcenschonenden Serverbetrieb auch mit einer großen Anzahl von Clients durchführen zu können. Für die Key-Exchange-Phase (wie auch für den Zeitinformationsaustausch) wurde ein detailliertes Sequenzdiagramm erstellt, welches in stark vereinfachter Form Eingang in den Draft gefunden hat (s. Abb. 2). Nach Austausch dieser Informationen wird die TLS-Verbindung beendet. Solange keine Fehler auftreten, ist keine neue TLS-Verbindung notwendig.

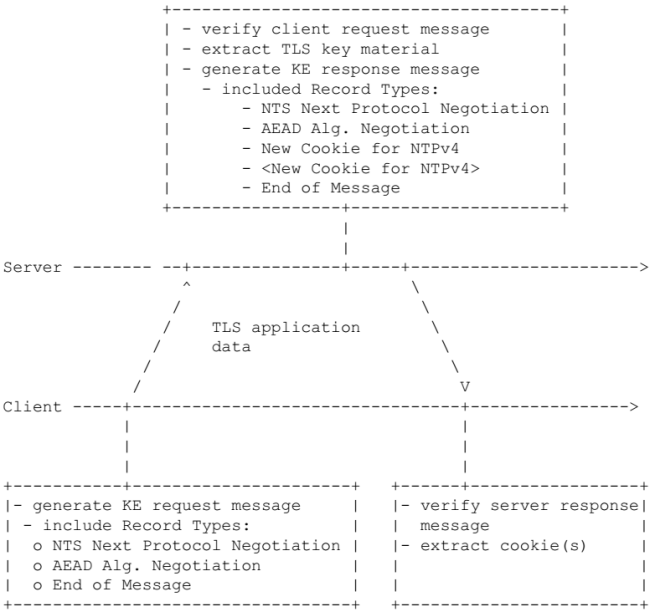

Abb. 2 Informationsaustausch in der Key-Exchange-Phase (KE) [8]

Austausch von NTS-gesicherten NTP-Zeitinformationspaketen

In der dritten Phase erfolgt der Austausch von Zeitsynchronisationsinformationen (vgl. Abb. 1). Hier kommunizieren die Partner über UDP mittels Network Time Protocol. Die Sicherung der Zeitinformation geschieht in den sog. Erweiterungsfeldern von NTP (NTS Extension Fields for NTPv4). Die in den Extension Fields übertragenen Informationen zu den Zeitpaketen ermöglichen die signierte und teilweise verschlüsselte Übertragung sämtlicher, benötigter Informationen. Dabei achtet man wie in den vorangegangenen Drafts darauf, dass der Server zustands-

los arbeiten soll und daher aus den Daten alle wesentlichen Informationen extrahieren kann, um das Antwortpaket aus diesen und seinen generellen Daten erzeugen zu können. Neu ist auch, dass der Client jetzt nicht mehr „trackbar“ ist, der Draft also erhöhten Privacy-Anforderungen genügt. Der Client schickt ein NTP-Paket (Time Request Message, s. Abb. 3) an den Server, das in den Erweiterungsfeldern ein Cookie, einen sog. Unique Identifier, der zur Identifikation der Antwort dient, und Cookie-Platzhalter enthält. Die Anzahl der Platzhalter entspricht dabei den verbrauchten Cookies. Der Platzhalter stellt sicher, dass die Anfragenachricht des Client die gleiche Größe wie die Antwort des Servers besitzt und damit ungleiche Übertragungszeiten vermeidet, was sonst zu Ungenauigkeiten der Zeitinformation führen würde.

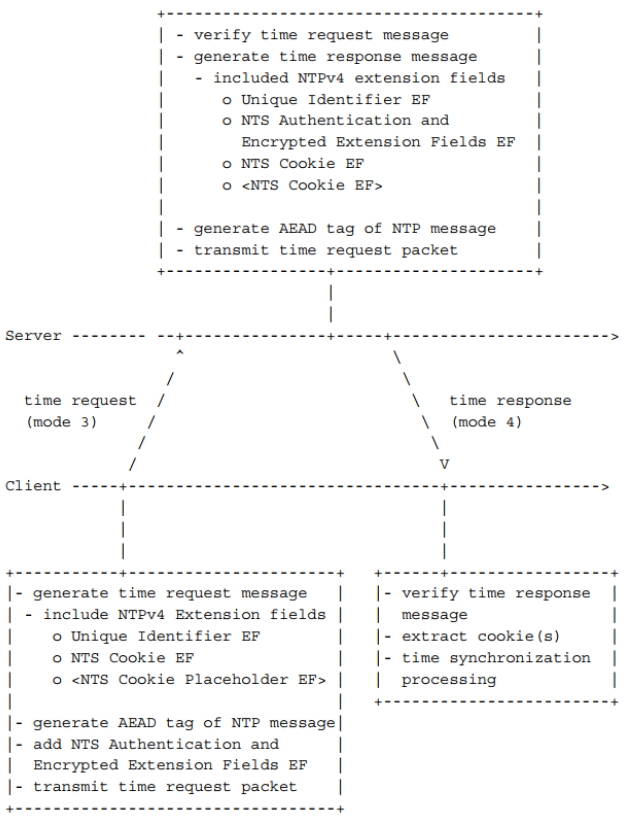

Abb. 3 Informationsaustausch mit Time Request und Time Response Messages (nach [8])

Der Server verifiziert die Anfrage, extrahiert aus dem Cookie alle nötigen Informationen dieser Kommunikationsverbindung und erzeugt die Time-Response-Nachricht damit. Dazu gehören der Unique Identifier sowie ein oder mehrere neue Cookies entsprechend der Platzhalter. Nach Generierung eines AEAD Tags verpackt er die Daten in den Erweiterungsfeldern der NTP-Nachricht und schickt sie an den Client zurück. Der verifiziert die Nachricht, extrahiert das bzw. die Cookies und teilt im positiven Fall NTP mit, dass die Zeitinformation gültig ist.

Implementierung und weitere Aktivitäten

Nach den grundsätzlichen Untersuchungen konnten die Anforderungen seitens der Kryptographie und sonstiger Randbedingungen geklärt werden. Es zeigte sich, dass OpenSSL zwar alle wesentlichen Anforderungen unterstützt, allerdings teilweise recht umständlich. Weitere Untersuchungen ergaben, dass die Bibliothek Boost.Asio wesentlich das Handling mit OpenSSL unterstützen kann. Aufgrund der vorangegangen Analysen und den daraus entstandenen detaillierten Sequenzdiagrammen konnte mit überschaubarem Aufbau ein Softwarekonzept für eine neue Implementierung entworfen werden. Abbildung 4 zeigt die Komponenten der Software. Es erwies sich, dass die neue Version mit ihrer Verwendung vieler Standardverfahren tatsächlich besser strukturierbar und leichter implementierbar ist.

Nach den entsprechenden Vorarbeiten konnte im Rahmen des Projekts eine (weltweit) erste vorläufige Implementierung des aktuellen Draft-Standes (preliminary Proof of Concept) realisiert werden. Damit stehen eine Client- und eine Server-Version des Drafts [8] zur Verfügung, die mit der vorhandenen NTP-Implementierung zusammen laufen. Ausführliche Tests und passendes Überarbeiten der Implementierung müssen allerdings zu einem späteren Zeitpunkt in Angriff genommen werden.

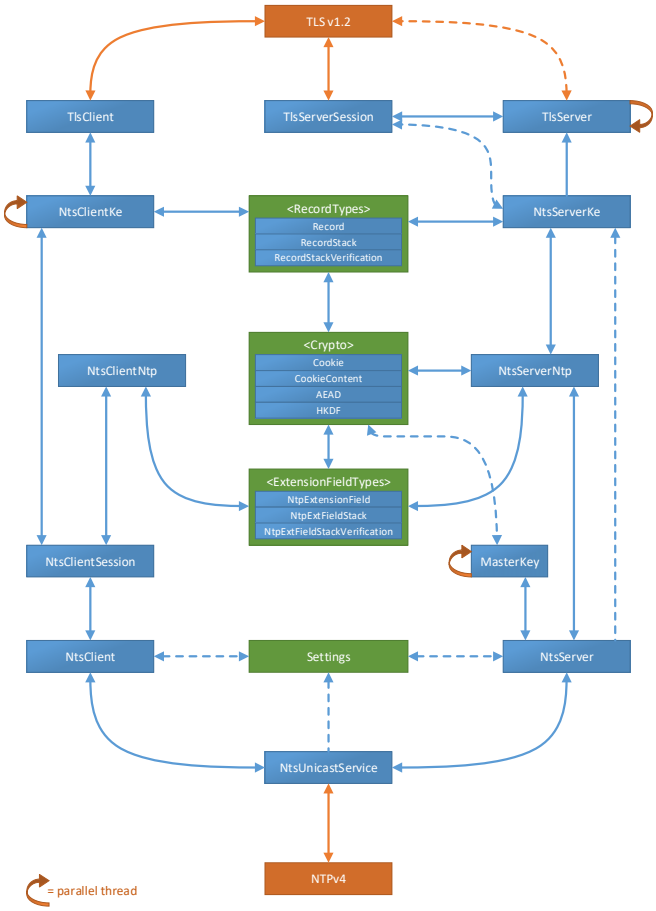

Abb. 4 Komponenten des Softwarekonzepts für eine NTS-Implementierung

Im März 2018 konnte auf dem IETF-Meeting 101 in London der aktuelle Stand präsentiert und diskutiert werden. Erfahrungen bei der Implementierung flossen in die weitere Arbeit in der Working Group ein. Auf dem Hackathon des Meetings konnte die Implementierung gegen eine „brandneue“ Python-Client-Realisierung seitens eines Autors des Drafts, Daniel Fox Franke, getestet werden. Zwar gab es zu Anfang ein kleines Problem im Client sowie eine unterschiedliche Auslegung einer Vorgabe des Drafts, beides konnte aber schnell behoben werden. Danach lief die NTS-gesicherte NTP-Kommunikation in diesem Testmodus problemlos.

Im Projektzeitraum erfolgten auch weitere Arbeiten im Kontext von NTS. So konnte eine erste Version der NTS-kompatiblen NTP-Version für Windows implementiert werden. Auch wenn Optimierungen und weitere Tests noch ausstehen, so steht doch damit ein weiterer nützlicher Baustein zur Verfügung. Auch mit der älteren NTS-Softwareversion wurde weiter gearbeitet. So wurden umfangreiche Tests und Untersuchungen vorgenommen, die u.a. bisherige Nachteile des Konzepts verdeutlichten, die aber im neuen Konzept vermieden werden können. Der Einsatz von NTS-geschütztem NTP im Feld des IoT wurde auf der Fachtagung „Internet of Things - vom Sensor bis zur Cloud 2017“ in München mit dem Beitrag „Gesicherte Zeitübertragung im Internet of Things mit dem NTS-Protokoll“ präsentiert [11]. Die Ergebnisse wesentlich detaillierterer Untersuchungen werden in Kürze auf dem „32nd European Frequency and Time Forum 2018 (EFTF)“ in Turin vorgestellt (Observed Time Synchronization Performance Using the Network Time Security Protocol) [12]. Die Veröffentlichungen sind dem Bericht als Anlage beigefügt.

Literatur

[1] Mills, D., Martin, J., Ed., Burbank, J., Kasch, W.: *Network Time Protocol Version 4: Protocol and Algorithms Specification*. RFC 5905, DOI 10.17487/RFC5905, 2010.

[2] IEEE: *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE Std. 1588–2008. New York, 2008.

[3] Bermbach, R.: *Network Time Security (NTS) – Unicast-Modus über NTP (Teil 1)*. Forschungsbericht SS 2016. Wolfenbüttel: Ostfalia Hochschule für angewandte Wissenschaften, 2016.

[4] Bermbach, R.: *Network Time Security (NTS) – Unicast-Modus über NTP (Teil 2)*. Forschungsbericht WS 2016/2017. Wolfenbüttel: Ostfalia Hochschule für angewandte Wissenschaften, 2017.

[5] Sibold, D., Röttger, S., Teichel, K.: *Network Time Security*. draft-ietf-ntp-network-time-security-15, IETF, September 2016.

[6] Langer, M.: *Implementierung des Network-Time-Security-Protokolls für den Unicast-Betrieb*. Masterarbeit. Wolfenbüttel: Ostfalia Hochschule für angewandte Wissenschaften, 2016.

[7] Franke, D., Sibold, D., Teichel, K.: *Network Time Security for the Network Time Protocol*. draft-ietf-ntp-using-nts-for-ntp-08, IETF, März 2017.

[8] Franke, D., Sibold, D., Teichel, K.: *Network Time Security for the Network Time Protocol*. draft-ietf-ntp-using-nts-for-ntp-11, IETF, März 2018.

[9] Rescorla, E.: *Keying Material Exporters for Transport Layer Security (TLS)*. RFC 5705, DOI 10.17487/RFC5705, März 2010.

[10] Harkins, D.: *Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES)*. RFC 5297, DOI 10.17487/RFC5297, Oktober 2008.

[11] Langer, M., Bermbach, R.: *Gesicherte Zeitübertragung im Internet of Things mit dem NTS-Protokoll.* Internet of Things – vom Sensor bis zur Cloud 2017, München 2017.

[12] Langer, M., Teichel, K., Sibold, D., Bermbach, R.: *Observed Time Synchronization Performance Using the Network Time Security Protocol.* 32nd European Frequency and Time Forum 2018 (EFTF), Turin, 2018.

Kontaktdaten

Ostfalia Hochschule für angewandte Wissenschaften
Fakultät Elektrotechnik
Prof. Dr.-Ing. Rainer Bermbach
Salzdahlumer Straße 46/48
38302 Wolfenbüttel
Telefon: +49 (0)5331 939 42620
E-Mail: r.bermbach@ostfalia.de
Internet: www.ostfalia.de/pws/bermbach