Direkte Ende-zu-Mitte Authentifizierung in kooperativen Netzen

Tobias Heer
COMSYS, RWTH Aachen University
heer@cs.rwth-aachen.de

Abstract: Kooperative Netze beruhen auf dem Prinzip der Zusammenarbeit von Benutzern auf Netzwerkebene. Sie ermöglichen dabei Kommunikation, wo andere Netzformen an wirtschaftliche oder technische Grenzen stoßen. Beispiele für kooperative Netzwerke sind dezentrale drahtlose Mesh-Netzwerke, Wi-Fi-Communities oder hybride Formen dieser Netzwerk-Typen. In kooperativen Netzen übernehmen Benutzergeräte Kernfunktionen des Netzes, wie z.B. die Weiterleitung von Paketen. Diese Kooperation bedingt eine opportunistische Offenheit, welche zu neuen Sicherheitsrisiken führt. Besonders die fehlende Möglichkeit zur Authentifizierung von Datenverkehr durch die weiterleitenden Geräte macht sie anfällig für Angriffe. Diese Arbeit schafft die Grundlagen für eine hocheffiziente Authentifizierung von Netzwerkverkehr durch die Knoten im Netz. Dies erlaubt es, die Identität eines Senders und die Integrität seiner Nachrichten effizient zu überprüfen, bevor die Nachrichten weitergeleitet werden. So können unauthentifizierte Datenströme und Angriffe effizient unterbunden werden.

1 Einleitung: Nutzen und Gefahren der Kooperation

Die Möglichkeit der spontanen drahtlosen Vernetzung von mobilen und stationären Geräten erlaubt es neue Netzwerkkonzepte zu etablieren, welche die Grenzen zwischen Netzwerkanbieter und Netzwerkbenutzer aufheben. Solche kooperativen Netze beruhen auf dem Prinzip der Zusammenarbeit von Benutzern auf Netzwerkebene, um Dienste, wie z.B. das Weiterleiten von Paketen oder den gemeinsamen Zugriff auf andere Netzwerkressourcen, wie Speicherplatz und Internetzugang, gemeinschaftlich zu erbringen. Beispiele für kooperative Netzwerke sind Ad-Hoc-Netze, dezentrale drahtlose Mesh-Netzwerke, Micro-Operator-Netzwerke, WLAN-Communities oder hybride Formen. Kooperative Netzwerke können dabei Lösungen schaffen wo andere Netzformen an technischen oder wirtschaftlichen Problemen scheitern. So gelingt es zum Beispiel Bewegungen wie Freifunk oder Funkfeuer mit einem minimalen Budget ganze Stadtteile mit Wi-Fi drahtlos zu vernetzen. Dabei kooperieren alle Benutzer, um sowohl gemeinsame Ziele (der Ausbau des Netzes) als auch egoistische Ziele (z.B. der Zugang zum Internet) zu erreichen.

Jedoch schafft das technische Konzept eines gemeinschaftlich organisierten Netzes auch neue Angriffsmöglichkeiten für egoistische und bösartige Benutzer. Zum Beispiel sind drahtlose Multi-Hop-Netzwerke (siehe Abb. 1) besonders anfällig gegenüber Angriffen, die auf dem Fluten des Netzwerks mit Schadpaketen oder der Manipulation und Fälschung von Paketen beruhen. Dabei lassen sich viele der möglichen Angriffe gegen kooperati-

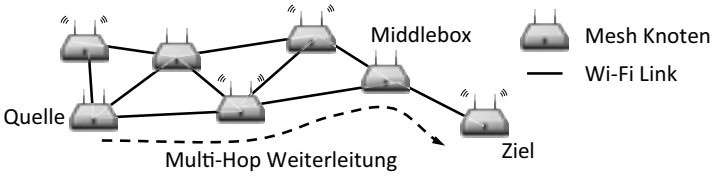


Abbildung 1: In kooperativen drahtlosen Mesh-Netzen kooperieren Knoten, um weite Distanzen drahtlos zu überbrücken. Dabei verschwimmt die Grenze zwischen Betreiber und Benutzer, da die Wi-Fi Router im Netz in der Regel ebenfalls Benutzern gehören.

ve Netze auf deren Offenheit gegenüber der Mitwirkung Unbekannter und dem Fehlen adäquater Authentifizierungsmechanismen auf verschiedenen technischen Ebenen zurückführen. Ein wichtiger Schritt zur Vermeidung von Missbrauch ist es daher, weiterleitenden Geräten, sogenannten *Middleboxen*, zu ermöglichen, sowohl die Identität der kommunizierenden Geräte als auch die Herkunft und Authentizität des Netzwerkverkehrs zu überprüfen. Protokolle, die dieses Ziel erreichen, fallen dabei in die Klasse der Ende-zu-Mitte Authentifizierungsprotokolle.

1.1 Ansätze zur Ende-zu-Mitte Authentifizierung

Effiziente Standardlösungen, um eine paketbezogene Authentifizierung zu erreichen, setzen typischerweise geteilte symmetrische Schlüssel zwischen den signierenden und verifizierenden Geräten voraus oder beruhen auf einer stets verfügbaren Verbindung zu einem Authentifizierungsserver. Diese Einschränkungen führen jedoch in kooperativen Netzen zu deutlichen Nachteilen bezüglich der Flexibilität, Effizienz, und Zuverlässigkeit. Daher ist oft eine *direkte* Authentifizierung zwischen den Endgeräten und Middleboxen durch alternative Authentifikationsmechanismen sinnvoll.

Eine weit verbreitete Methode zur Feststellung der Identität von Kommunikationspartnern bzw. der Authentizität und Integrität ihrer Nachrichten sind Public-Key Systeme. In der zugrundeliegenden Dissertation und den zugehörigen Veröffentlichungen wird ausführlich auf die Eigenschaften von Public-Key Verfahren zur Sicherung von *sporadischen* Authentfizierungsereignissen, wie z.B. der Authentifizierung eines Geräts beim Verbindungsaufbau eingegangen [HHK$^+$09]. Dieser Artikel setzt den Fokus jedoch auf die Authentifizierung von *hochfrequenten* Ereignissen, z.B. der paketweisen Verifikation von breitbandigem Nutzlastverkehr. Hier stellen die beachtlichen Anforderung der Public-Key Kryptographie an die Rechenleistung der Endgeräte und Middleboxen eine bedeutende Einschränkung dar. So benötigt die Verifikation eines Datenstroms mit einem Durchsatz von 30 Mbit/s bereits etwa 3.000 unabhängige Verifikationen, um jedes Nutzlastpaket zu authentifizieren. Gleichzeitig liegt die Leistung von typischen drahtlosen Wi-Fi Routern (z.B. mit AMD Geode 500 MHz CPU) für weit verbreitete asymmetrische Signaturen ohne spezielle Hardwarebeschleunigung, um zwei Größenordnungen *unter* diesem Wert (DSA 1024 Bits: 55 Verifikationen, ECDSA 160 Bits: 55 Verifikationen). Basierend auf diesen Werten könnte dieser Router daher nur schmalbandige Datenströme mit einem Volumen von weniger als 660 Kbit/s paketweise verifizieren.

Um eine ausreichend effiziente Ende-zu-Mitte Verifikation von Datenpaketen zu erreichen, müssen daher alternative Authentifizierungsmethoden geschaffen werden, welche die beschränkten Hardwareressourcen in kooperativen Netzen berücksichtigen. Im Gegensatz zu asymmetrischen Public-Key Verfahren zeichnen sich symmetrische Authentifizierungsverfahren durch eine hohe Effizienz aus. Jedoch lassen sich diese Verfahren in der Regel nicht ohne paarweise Schlüssel betreiben. Eine Ausnahme stellen hierbei Verfahren dar, die auf kryptographischen Einwegfunktionen, also Hash-Funktionen, und deren Verkettung zu Hash-Ketten beruhen. Basierend auf Hash-Ketten lassen sich Authentifizierungsprotokolle entwickeln, die keine gemeinsamen *geheimen* Schlüssel benötigen, sondern für die ein gesicherter *öffentlicher* Wert ausreichend ist.

Auf Hash-Ketten basierende Authentifizierungsprotokolle wurden bislang erfolgreich zur Authentifizierung auf Ende-zu-Ende-Basis und für die effiziente Sicherung von Multicast eingesetzt. Diese Arbeit erweitert und ergänzt bestehende Ansätze, um sie zur Authentifizierung in Ende-zu-Mitte Szenarien anwendbar zu machen. Dazu stellt dieser Artikel zwei geeignete Verfahren vor: Das “Adaptive and Lightweight Protocol for Hop-by-Hop Authentication” (ALPHA) verwendet effiziente Hash-Funktionen und Hash-Ketten, um eine schnelle Überprüfung der *Quelle* und *Integrität* eines Netzwerkpakets zu erreichen. Die “Stream-based Per-packet One-time Tokens for Cryptographic Source Authentication” (SPOTS) verzichten auf Ende-zu-Mitte Integritätsschutz, um die kryptographische Komplexität der Authentifizierung weiter zu senken. Dies bedeutet, dass SPOTS nur eine Überprüfung der *Paketquelle* durchführt, was SPOTS ermöglicht, noch effizientere Ansätze zu verfolgen. Beide Mechanismen lassen sich zusätzlich flexibel parametrisieren, um effiziente Ende-zu-Mitte Authentifizierung für ein breites Spektrum von Szenarien, innerhalb und außerhalb von kooperativen Netzwerken, zu ermöglichen.

2 Grundlagen: Hash-Ketten basierte Authentifizierung

Hash-Ketten nach Lamport [Lam81] sind flexible Konstrukte, die sich zur Verifikation der Quelle und Integrität eines Pakets eignen. Um eine Hash-Kette zu erzeugen wird ein Zufallswert s gewählt, auf den eine kryptographische Hash-Funktion H iterativ angewandt wird. Das Ergebnis $h_1 = H(s)$ bildet das erste Glied der Kette. Weitere Glieder werden durch wiederholte Anwendung von H auf das vorige Glied erzeugt: $h_i = H(h_{i-1}) = H^i(s)$. Das letzte Glied der Hash-Kette, h_n, wird dabei als Anker bezeichnet. Nach dem gegenseitigen gesicherten Austausch eines Ankers können sich zwei Kommunikationspartner zu einem beliebigen Zeitpunkt über die Preisgabe des nächst niedrigen Elements in der Hash-Kette h_{n-1} authentifizieren bzw. wiedererkennen.

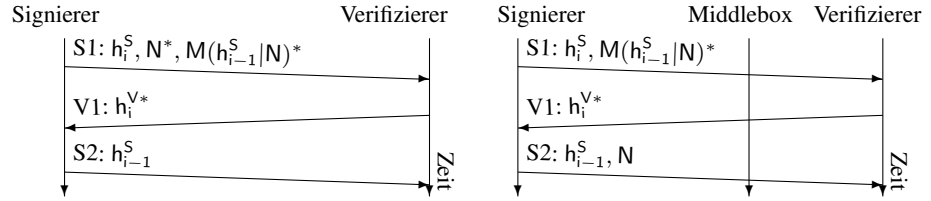
Um die Integrität von Nachrichteninhalten zu schützen, können noch nicht preisgegebene Elemente einer Hash-Kette als Schlüssel für ein symmetrisches Signaturverfahren (z.B. HMAC) verwendet werden. Diese Technik nennt sich ”Delayed Secret Disclosure”, also die zeitverzögerte Preisgabe eines vormals geheimen Schlüssels. Abbildung 2a veranschaulicht den Vorgang in einer leicht vereinfachten Variante. Ein Signierer übermittelt dabei eine geschützte Nachricht N an einen Verifizierer. Beide erzeugen zuvor eine Hash-Kette mit Elementen h_i^S und h_i^V und tauschen deren Ankerelemente aus. Die Superskripte

S und V deuten dabei die Herkunft an (Signierer, Verifizierer). Um die Nachricht N zu übermitteln, erzeugt der Signierer eine HMAC Signatur $M(h_{i-1}^S | N)$ der Nachricht, wobei er das nächste unveröffentlichte Hash-Ketten Element h_{i-1}^S als Schlüssel verwendet. Der Signierer übermittelt N und die Signatur in einem ersten Paket (S1) an den Verifizierer. Dieser speichert beide Werte und bestätigt den Empfang mit einem Element seiner Hash-Kette h_i^V in seiner Antwort (V1). Nach Erhalt der Bestätigung veröffentlicht der Sender das Hash-Ketten Element h_{i-1}^S in einem weiteren Paket (S2). Nach Erhalt des S2 Pakets kann der Empfänger die Authentizität von N prüfen. Zum Senden einer weiteren Nachricht beginnt der beschriebene Ablauf erneut. Durch die zeitliche Trennung zwischen Signaturerstellung bzw. Übermittlung und Preisgabe des Signaturschlüssels wird sichergestellt, dass nur der legitime Signierer zum Erstellungszeitpunkt der Signatur alle notwendigen Informationen besitzt. Aufgrund des interaktiven Charakters des Signaturablaufs werden diese Signaturen auch interaktive Hash-Ketten oder Interactive Hash Chain (IHC) Signaturen genannt.

IHC Signaturen zeichnen sich durch einen sehr geringen Berechnungsaufwand aus, da die Erstellung bzw. Verifikation nur wenige Anwendungen einer Hash-Funktion bedingt. Im Vergleich zu den zuvor genannten Werten für DSA und ECDSA kann der oben genannte Router ca. 100.000 Hash-Funktionen pro Sekunde berechnen. Wäre ausschließlich dieser Berechnungsaufwand ausschlaggebend, wäre die Rechenleistung des oben beschriebenen Routers mehr als ausreichend, um breitbandigen Verkehr mit hunderten Mbit/s zu verifizieren.

3 ALPHA: Effiziente Ende-zu-Mitte Authentifizierung

Die hohe Berechnungseffizienz von IHC Signaturen verspricht zwar eine effiziente Verifikation von Paketen im Netzwerk, jedoch besitzen diese Signaturen einige praktische Nachteile für den Einsatz in der Ende-zu-Mitte Authentifizierung. Dies lässt sich einfach an zwei Beispielen zeigen. Zum Einen soll die Authentifizierung das Netzwerk vor Angriffen wie dem unerlaubten Fluten mit Schadpaketen sichern. Zwar leistet die beschriebene Form der IHC Signatur eine Verifikation der Nachricht N, jedoch muss diese Nachricht zu-



(a) IHC Signatur ohne Commitment

(b) IHC Signatur mit Commitment.

Abbildung 2: IHC Signaturen ohne Commitment (a) bedingen eine Zwischenspeicherung von Nachricht und Authentisierungscode und eignen sich daher nicht für den Einsatz in Ende-zu-Mitte Szenarien. Schemata mit Commitment (b) reduzieren die zu speichernden Daten stark. Der Stern (*) markiert Werte, die von Endsystemen und Middleboxen bis zum Empfang von S2 gespeichert werden müssen.

erst an den Verifizierer übermittelt werden, bevor deren Integrität überprüft werden kann. Dies bedeutet, dass Schadpakete nicht im Netzwerk aufgehalten werden können. Zum Anderen müssen alle Netzwerkteilnehmer, die an der Verifikation der Nachricht interessiert sind, die Nachricht bis zur Preisgabe des Schlüssels h_{i-1}^S im S2 Paket speichern. Im Ende-zu-Mitte Fall trifft dies auch auf die Middleboxen zu, was bedeutet, dass diese Netzwerkelemente große Datenmengen zwischenspeichern müssten und dadurch angreifbar für Denial of Service (DoS) Angriffe würden. Somit würden sich die beschriebenen IHC Signaturen zwar zur authentifizierten Übertragung von Informationen vom End-Gerät an die Middlebox eignen, jedoch könnten sie nicht größere Mengen an Netzwerkverkehr authentifizieren.

Eine leichte Abwandlung des Schemas kann beide Probleme beheben. Abbildung 2b zeigt eine IHC Variante, die zuerst nur ein "Commitment" im S1 Paket verschickt und später die Nachricht N während der Veröffentlichung des Schlüssels im S2 Paket versendet. Diese Variante der IHC Signaturen hat drei vorteilhafte Eigenschaften im Ende-zu-Mitte Fall:

Bandbreiteneffizienz: Das S1 Paket wird sehr klein, da die HMAC Signatur nur ca. 20 Byte (SHA-1) in Anspruch nimmt. Dies bedeutet, dass nur ein kleines Paket ohne Überprüfung an den Verifizierer ausgeliefert wird. Das große S2 Paket mit der Nachricht N kann vor der Weiterleitung durch die Middleboxen verifiziert werden.

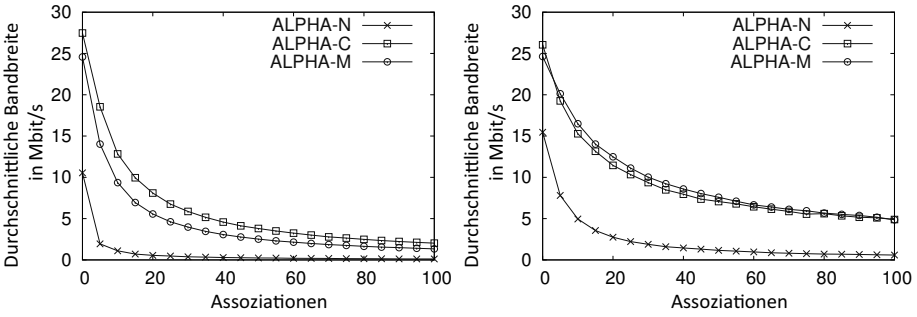
Speichereffizienz: Middleboxen müssen nur die kleinen Commitments anstatt der gesamten Nachricht N zwischenspeichern, was die Speicheranforderungen drastisch senkt und die Angreifbarkeit dieser Geräte deutlich verringert.

Senderausschluss: Ohne eine Bestätigung des Verifizierers durch ein V1 Paket kann der Sender keine großen S2 Pakete senden. Der Verifizierer kann so den Signierer daran hindern Datenpakete zu senden, wobei die Middleboxen dies, durch Weiterleitung oder Verwerfen des S2 Pakets, implizit umsetzen.

Aufgrund dieser Eigenschaften verwenden wir für das "Adaptive and Lightweight Protocol for Hop-by-Hop Authentication", ALPHA, diese zweite Variante der IHC Signaturen [HGGMW08]. Wir implementierten diese Variante und evaluierten sie für die oben genannten Router. Dabei konnte der Router bereits Durchsätze von 10 Mbit/s verifizieren, was den theoretischen Wert der genannten asymmetrischen Signaturen bereits um das 15-fache übersteigt. Jedoch zeigt sich bei genauerer Betrachtung des Routers, dass weder das Netzwerk, noch die CPU des Geräts der limitierende Faktor ist. Die offensichtliche Wurzel des Problems liegt dabei auf der Hand: Zwar senkt ALPHA den *Berechnungsaufwand* der Signaturen, jedoch verdreifacht es durch die IHC Signatur den *Kommunikationsaufwand*. Zur weiteren Erhöhung des möglichen Durchsatzes ist es daher notwendig, den Kommunikationsaufwand wieder zu senken.

3.1 Amortisierung des Kommunikationsaufwands

Die Tatsache, dass die Commitments (die HMAC Signaturen) in den S1 Paketen sehr klein sind, ermöglicht eine erste Optimierung des Verfahrens. Anstatt ein einziges Commitment für eine einzige Nachricht (ein S2 Datenpaket) im ersten S1 Paket zu senden, kann der Si-



(a) Eine Assoziation (Bidirektional)

(b) Fünf Assozationen (Bidirektional)

Abbildung 3: ALPHA Durchsatz bei 30 ms Netzwerklatenz.

gnierer mehrere Commitments für mehrere Nachrichten N_i gesammelt (kumuliert) versenden. Dieser Betriebsmodus von ALPHA nennt sich daher ALPHA-C (Cumulative Mode). Nach der Bestätigung durch den Verifizierer kann der Signierer dann alle Nachrichten N_i in S2 Paketen parallel versenden. ALPHA-C ermöglicht es, die Kommunikationskosten für den S1/V1 Austausch auf viele S2 Pakete zu verteilen. Um dies zu erreichen, müssen beim Senden des S1 Pakets bereits alle Signaturen der Datenpakete vorliegen. Da Transportprotokolle wie TCP stets mehrere Datenpakete gleichzeitig senden, um einen hohen Durchsatz zu erreichen, stellt dies in der Praxis keine Einschränkung dar. Die Anzahl der S2 Pakete pro S1 Paket ist bei dieser Optimierung jedoch durch zwei Faktoren begrenzt. Zum Einen müssen alle Signaturen Platz im S1 Paket finden. Je nach Protokollentwurf begrenzt dies die Anzahl der S2 Pakete in der Praxis auf ca. 70. Zweitens müssen Middleboxen alle Signaturen im S1 Paket bis zum Erhalt der S2 Pakete zwischenspeichern. Sollten die Middleboxen stark platzbeschränkt sein, kann dies eine Einschränkung darstellen und die Maximalzahl der S2 Pakete weiter senken.

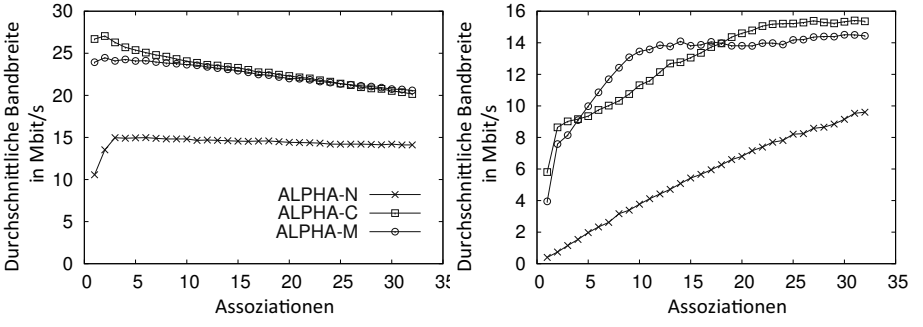
Um diese Einschränkungen zu umgehen bietet ALPHA einen weiteren Modus: Binäre Hash-Bäume, so genannte Merkle-Bäume [Mer90], erlauben es, eine große Zahl von Eingabewerten sicher auf einen einzigen Ausgabewert (die Wurzel) mit fester Größe abzubilden, sodass der Ausgabewert von allen Eingabewerten abhängt. Durch die Verwendung eines Merkle-Baums ist es möglich, eine beliebige Zahl von Nachrichten N_i durch einen kleinen Wert r zu repräsentieren. Der Sender kann somit r im S1 Paket authentisiert übermitteln und später nach Erhalt der Bestätigung alle N_i parallel versenden. Um ein S2 Paket zu verifizieren muss eine Middlebox nachvollziehen, ob die Nachricht N_i Teil der Eingabemenge des Baums war. Hierzu benötigt die Middlebox a) die Nachricht N_i, b) die Wurzel r und c) die Nachbarknoten des Pfads durch den Baum von N_i zu r. Da Merkle-Bäume balancierte Binärbäume sind, handelt es sich dabei um $log_2(n)$ Nachbarknoten für n Nachrichten. Dieser Betriebsmodus nennt sich aufgrund der Verwendung von Merkle-Bäumen ALPHA-M. ALPHA-M stellt konstante Speicherplatzanforderungen an Middleboxen zur vorübergehenden Speicherung von r. Der Platzbedarf in jedem S2 Paket wächst jedoch mit der Anzahl der parallel verschickten Nachrichten logarithmisch. Im Gegensatz dazu wächst bei ALPHA-C der Speicheraufwand der Middleboxen linear zur Anzahl der parallel versendeten S2 Paketen, wobei der Verifikationsaufwand konstant ist.

Durch Verwendung von ALPHA-C und ALPHA-M lässt sich der Durchsatz von ALPHA bereits auf 26 Mbit/s steigern, jedoch fällt bei genauerer Betrachtung auf, dass für größere Netzwerklatenzen der Durchsatz überproportional stark abfällt. Abbildung 3a stellt diesen Abfall für alle ALPHA Modi dar. ALPHA-N repräsentiert dabei den einfachen ALPHA Modus ohne parallele Übermittlung mehrerer S2 Pakete. Der Grund für diesen steilen Abfall ist, dass während des Austauschs der S1 und V1 Pakete keine Datenpakete gesendet werden können. Daher entstehen bei größerer Latenz lange Zeiten der Inaktivität. Um dieses Problem zu beheben, können mehrere unabhängige Signaturprozesse, also unabhängige ALPHA Assoziationen, ineinander verzahnt werden. Ein oder mehrere Assoziationen können so weiter Daten übertragen, während andere Assoziationen auf eine Bestätigung des S1 Pakets durch den Verifizierer warten. Abbildung 3a zeigt die Durchsatzwerte für fünf parallele Assoziationen. Insgesamt lässt sich der scharfe Abfall der Bandbreite durch die Verwendung mehrerer Assoziationen deutlich mindern. Abbildung 4 zeigt den kombinierten Effekt der vorgestellten Techniken. Abhängig vom Modus, von der Verbindungsverzögerung und dem Grad der Parallelität erreicht ALPHA dabei Werte von bis zu 25 Mbit/s auf dem drahtlosen Router und liegt damit bereits jenseits der Leistungsfähigkeit vieler drahtloser 802.11 Multi-Hop Verbindungen [WHBW11].

4 SPOTS: Token-basierte Quellauthentifizierung

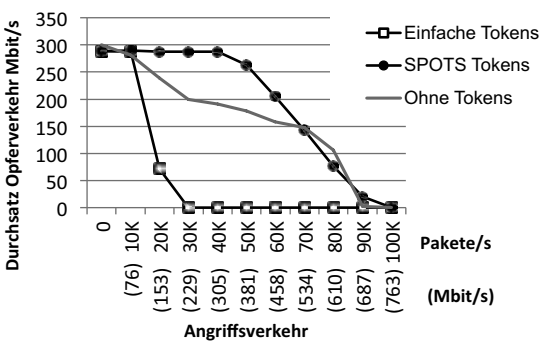
In einigen Anwendungsfällen ist für Middleboxen nur die Herkunft eines Pakets von Bedeutung. Zum Beispiel kann das Fluten des Netzwerks von Middleboxen bereits durch eine sichere Bestimmung der Quelle verhindert werden und bedingt keinen Integritätsschutz des Paketinhalts. ALPHA leistet daher durch seine Quell- *und* Integritätsprüfung für diese Szenarien zuviel und macht durch das IHC Verfahren die Kommunikation unnötig aufwändig. Eine einfachere Art der Quellauthentifizierung ohne eine Überprüfung der Paketinhalte ist daher notwendig, um weniger anspruchsvolle Szenarien effizient zu bedienen.

Eine auf Tokens basierende Quellauthentifizierung eignet sich für die einfache und sichere Bestimmung der Herkunft eines Pakets. Dabei wird jedem Paket ein *einmal* gültiges Token

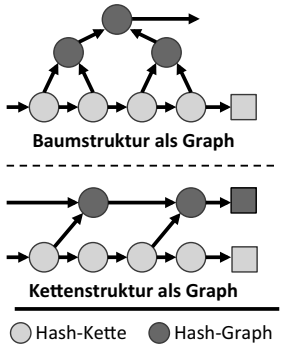


(a) 0,5 ms Netzwerklatenz (b) 20 ms Netzwerklatenz

Abbildung 4: Durchsatz der verschiedenen ALPHA-Modi.



(a) Angriff auf Tokens



(b) SPOTS Tokens verwenden Hash-Graphen

Abbildung 5: Die einfachen Token-Schemata stellen ein zusätzliches Risiko während DoS Angriffen dar. Die SPOTS Tokens ermöglichen effizienten Schutz gegen DoS Angriffe durch die Verwendung von Hash-Graphen. (PC Middlebox, 2.5GHz AMD Athlon 4800+).

hinzugefügt, welches der Empfänger effizient verifizieren kann. Hash-Ketten können als Basis für solche Tokens dienen, indem jedes Element der Kette als Token interpretiert wird. Die sukzessive Preisgabe der Hash-Ketten Elemente von h_n hin zu h_0 beschreibt dabei einen Strom von Tokens, in dem jedes Token durch Hashen einem zuvor empfangenen Token zugeordnet werden kann: $H(h_i) = h_{i+1}$. Da Hash-Ketten hocheffizient erzeugt und verifiziert werden können, verspricht dieser Ansatz einen hohen Durchsatz.

Bestehende Hash-Ketten-basierte Systeme [TO03, DHM05] sind jedoch entweder nur für geringe Bandbreiten geeignet oder lassen Netzwerkeffekte, wie Paketverlust und mögliche DoS Angriffe unberücksichtigt. Daher eignen sie sich nicht für den Schutz von breitbandigen Datenströmen in realen Szenarien. Das Grundproblem dieser Ansätze besteht dabei in der Annahme, dass die Tokens fast vollständig bei den verifizierenden Stellen (End-Systeme und Middleboxen) eintreffen. Bei nur geringem Verlust von wenigen Tokens kann dann durch mehrmalige Anwendung der Hash-Funktion die Lücke zwischen einem zuvor verifizierten Token h_i und einem neu eingetroffenen Token h_{i-n} geschlossen werden: $H^n(h_{i-n}) = h_i$. Jedoch steigt der Verifikationsaufwand eines Tokens dabei linear zur Anzahl der zuvor verlorenen Tokens, sodass die Verifikation nach längeren sequentiellen Verlusten deutlich aufwändiger wird. Dieses Verhalten macht den Einsatz solcher einfacher Token-Schemas unsicher, da ein Angreifer durch Fälschen von Paketzählern sehr einfach ungültige Tokens erzeugen kann, die einen hohen vorangegangenen Verlust andeuten. Um diese Tokens zu falsifizieren muss die Middlebox zahlreiche Iterationen der Hash-Funktion ausführen. Dies bedeutet, dass jedes Angriffspaket nicht nur Bandbreite, sondern in hohem Maße auch Rechenkapazität der Middlebox vergeudet. So stellt der scheinbare Schutz der Tokens in der Praxis eine Bedrohung dar, da ein Angreifer durch Ausnutzung der Tokens CPU-beschränkte Geräte deutlich einfacher überlasten kann. Abbildung 5a zeigt die Auswirkungen eines solchen Angriffs. Der durch einfache Tokens "geschützte" Opferstrom wird anfällig gegenüber DoS Angriffen. Dadurch ist die Leistungsfähigkeit der Middlebox bereits bei deutlich geringeren Angriffsraten erreicht und überschritten, sodass der Opferverkehr stark abnimmt.

4.1 Hash-Graphen für robuste Token-Verifikation

Um Hash-basierte Tokens in der Praxis einsetzbar zu machen ist es notwendig, Möglichkeiten zur effizienten Verifikation von Tokens nach sequentiellem Verlust voriger Pakete zu schaffen. Die "Stream-based Per-packet One-time Tokens for Cryptographic Source Authentication" (SPOTS) Schemata sind eine Alternative zur Verwendung von linearen Hash-Ketten. SPOTS verwendet dabei Hash-Graphen anstelle von Hash-Ketten. Ähnlich zu einem Merkle-Baum können beliebige azyklische gerichtete Graphen aus Verkettungen von Hash-Funktionen gestaltet werden. Jeder Knoten im Graph ist dabei das Ergebnis der Anwendung einer Hash-Funktion, während jede Kante die Anwendung einer Hash-Funktion darstellt. Die Herausforderung bei der Gestaltung von geeigneten Hash-Graphen besteht darin, im Normal- und im Maximalfall möglichst wenige Kanten für die Verbindung eines neuen Wertes (eines frischen Tokens) hin zu einem bereits verifizierten Wert zu durchschreiten. Gleichzeitig muss die Menge von Zusatzinformationen, die zur Durchschreitung der Kanten benötigt wird, gering gehalten werden. SPOTS verwendet Ketten- und Baumartige Graphen, um eine schnelle Verifikation zu ermöglichen. Wie in Abbildung 5b gezeigt, wird dazu eine reguläre Hash-Kette um einen Hash-Graphen erweitert. Falls kein Paketverlust angenommen wird (d.h. es liegt auch kein Angriff vor), kann entlang der originären Hash-Kette effizient verifiziert werden. Falls ein Paketverlust angezeigt wird, kann der Hash-Graph dazu verwendet werden, um ein Element der Hash-Kette mit wenigen Hash-Berechnungen zu verifizieren oder zu falsifizieren. Wie Abbildung 5a zeigt, lässt sich dadurch die Schwäche der einfachen Tokens nicht nur ausgleichen, sondern es kann zusätzlich ein effizienter Schutz vor DoS Angriffen erreicht werden. Gleichzeitig erzielen die mit SPOTS geschützten Datenströme in unserer Evaluation einen hohen Durchsatz von bis zu 69 Mbit/s für die oben genannten drahtlosen Router und bis zu 288 Mbit/s für eine Middlebox mit Athlon 4800+ Prozessor. Neuere Messungen mit aktueller PC Hardware zeigen sogar einen Durchsatz von bis zu 748 Mbit/s auf einem Intel i7-870 System. Dabei wird die Quelle jedes einzelnen Nutzlastpakets kryptographisch verifiziert, sodass gefälschte Pakete bereits früh im Netzwerk verworfen werden können.

5 Zusammenfassung

Kooperative Netze erlauben einen flexiblen Einsatz in Szenarien, in denen klassische geplante Netzwerkarchitekturen an technische oder wirtschaftliche Grenzen stoßen. Jedoch birgt der Aspekt der Kooperation neue Risiken, denen mit adäquaten Mitteln begegnet werden muss. Authentifizierung von Benutzern und Datenströmen auf Netzwerkebene kann dabei zu einer deutlichen Erhöhung der Widerstandsfähigkeit des Netzes gegen Angriffe führen, jedoch muss diese effizient erreichbar sein, um nicht die Leistungsfähigkeit des Netzes zu begrenzen und neue Schwachstellen zu schaffen. Dieser Artikel stellt zwei komplementäre Authentifizierungssysteme vor, die es Netzwerkkomponenten erlauben, sehr effizient die Authentizität von Paketen zu überprüfen. Die Systeme können als komplementär betrachtet werden, da sie maßgeschneiderte Lösungen für Szenarien bereithalten, die Quellauthentifizierung und/oder Integritätsschutz benötigen.

Literatur

[DHM05] Jing Deng, Richard Han und Shivakant Mishra. Defending Against Path-based DoS Attacks in Wireless Sensor Networks. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks, SASN*, Seiten 89–96. ACM Press, 2005.

[HGGMW08] T. Heer, S. Götz, O. Garcia Morchon und K. Wehrle. ALPHA: An Adaptive and Lightweight Protocol for Hop-by-hop Authentication. In *ACM CoNEXT: Proceedings of the 2008 ACM CoNEXT Conference, Madrid, Spain*. ACM, 2008.

[HHK+09] T. Heer, R. Hummen, M. Komu, S. Götz und K. Wehrle. End-host Authentication and Authorization for Middleboxes based on a Cryptographic Namespace. In *Proceedings of the IEEE International Conference on Communications 2009 (ICC 2009), Dresden, Germany*. IEEE, 2009.

[Lam81] L. Lamport. Password Authentication with Insecure Communication. *Communications of the ACM*, 24(11):770–772, November 1981.

[Mer90] RC. Merkle. A Certified Digital Signature. In *CRYPTO '89: Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*, Seiten 218–238, London, UK, 1990. Springer.

[TO03] H. Tewari und D. O'Mahony. Multiparty Micropayments for Ad Hoc Networks. In *Wireless Communications and Networking, WCNC 2003*, Jgg. 3, Seiten 2033–2040. IEEE, 2003.

[WHBW11] H. Wirtz, T. Heer, R. Backhaus und K. Wehrle. Establishing Mobile Ad-Hoc Networks in 802.11 Infrastructure Mode. In *ACM MobiCom 2011 Workshop on Challenged Networks (CHANTS'11)*, September 2011.

Tobias Heer studierte, nach einem zweijährigen Exkurs in die Erziehungswissenschaften, Informatik an der Universität Tübingen. Das Studium schloss er 2006 mit Auszeichnung ab. Seine Diplomarbeit verfasste er während eines (kalten, dunklen) Gastaufenthalts am Helsinki Institute for Information Technology in Finnland. Dabei befasste er sich mit leichtgewichtigen Sicherheitslösungen für mobile und ressourcenbeschränkte Geräte. Die Arbeit wurde 2008 mit dem KuVS Preis für die beste Diplomarbeit im Bereich Kommunikation und Verteilte Systeme ausgezeichnet. Er kehrte seither mehrmals, im Rahmen von Forschungsaufenthalten, nach Helsinki zurück. Seine Faszination für Sicherheit und Kommunikationsprotokolle konnte Tobias am DFG Graduiertenkolleg *Software für mobile Kommunikationssysteme* an der RWTH Aachen weiter ausleben. Seit 2009 ist er Projektleiter für die Netzwerkaspekte des *Mobile ACcess* Kooperationsprojekts und arbeitet am Entwurf sicherer kooperativer Wi-Fi Netzwerke. Neben seiner wissenschaftlichen Arbeit hat er sich, im Rahmen der Internet Engineering Task Force (IETF), an der Standardisierung von Mobilitäts- und Sicherheitsprotokollen beteiligt. Seine Promotion schloss Tobias im Dezember 2011 mit Auszeichnung ab. Er ist stolzer Vater von zwei Kindern: Jana und Jannik.