Kommunikation ohne Tricks

IPv6 – das Internet-Protokoll der nächsten Generation (Teil 2)

Torsten Braun

Aufgrund der steigenden Popularität des Internet wächst die Zahl der Internet-Knoten seit Jahren stark an. Experten gehen davon aus, daß der derzeit verfügbare IPv4-Adreßraum spätestens im Jahr 2020 nicht mehr ausreicht. Eine Lösung des Problems besteht darin, einen größeren Adreßraum – und damit ein neues IP-Protokoll – zu schaffen. Doch die 1992 aus der Taufe gehobene IPv6 wird in puncto Verbreitung den Erwartungen bisher nicht gerecht. Der Grund: Die Zahl der benutzten IP-Adressen wächst nicht so schnell wie ursprünglich angenommen. Der anhaltende Einsatz von IPv4 führt allerdings dazu, daß wertvolle neue IPv6-Funktionen nicht verwendet werden. Und die Integration neuer Funktionalitäten in IPv4 erfordert immer größere Tricks.

Prof. Dr. Torsten Braun ist Leiter der Forschungsgruppe Rechnernetze und Verteilte Systeme am Institut für Informatik und Angewandte Mathematik der Universität Bern

Sicherheitsfunktionen

Die für IPv4 und IPv6 definierten IP-Sicherheitsfunktionen (IPSec) umfassen Erweiterungen zur Unterstützung von Authentifizierungen und vertraulicher Kommunikation. Während diese Sicherheitserweiterungen für IPv4 optional sind, müssen IPv6-Implementierungen eine minimale Menge von Funktionalitäten unterstützen.

Zur Durchführung von Sicherheitsfunktionen müssen sich Sender und Empfänger auf eine Menge von Parametern einigen, die vor der eigentlichen Datenkommunikation vereinbart werden müssen. Eine Sicherheits-Assoziation, identifiziert durch die Zieladresse und den Security Parameter Index (SPI), beschreibt unter anderem verschiedene ausgehandelte Parameter wie Schlüssel für Authentifizierung/ Verschlüsselung, Algorithmen zur Verschlüsselung/Authentifizierung, Lebenszeit der Schlüssel, Lebenszeit der Security-Assoziation, Quelladresse, Sicherheitsstufe der Kommunikation (vertraulich, unklassifiziert usw.). Die Verwaltung der Schlüssel wird dabei durch separate Schlüsselverwaltungsprotokolle wie z.B. das Internet Security Association and Key Management Protocol (ISAKMP) oder Internet Key Exchange (IKE) unterstützt

Authentifizierungen stellen sicher, daß die Daten vom angegebenen Sender wirklich gesendet wurden und daß sie unverfälscht beim Empfänger ankommen, während Verschlüsselungen der Pakete gewährleisten, daß nur der beabsichtigte Empfänger die Daten in eine lesbare Form verwandeln kann. Zur Überprüfung der Echtheit der Daten wird der Authentifizierungs-Header verwendet. Innerhalb des Authentifizierungs-Headers existieren Felder für Authentifizierungsdaten, deren Inhalt vom verwendeten Sicherheitsalgorithmus abhängt. Die Sequenznummer im Header soll sogenannte Replay-Attacken aufdecken, bei denen Angreifer Pakete kopieren und zu einem späteren Zeitpunkt wiederholen könnten. Bei Verwendung des MD5-Algorithmus (Message Digest) wird aus dem zu sendenden Paket und einem zuvor zwischen den Kommunikationspartnern vereinbarten Schlüssel eine 128-Bit-Kennung berechnet und in die Authentifizierungsdaten eingetragen. Der Empfänger, der im Besitz des korrekten Schlüssels sein muß, führt dieselben Operationen wie der Sender durch und wird nur dann dieselben Authentifizierungsdaten wie der Sender errechnen, wenn seine Operationen auf dem korrekten Schlüssel basieren und die Daten unverfälscht, d.h. ohne Bitfehler, Manipulationen Dritter usw., übertragen wurden.

Authentifizierungs-Mechanismen verändern die Daten des Pakets nicht und können daher keinen vertraulichen, abhörsicheren Datenaustausch ermöglichen. Für diese Zwecke wird in IPv6 der DES-CBC-Verschlüsselungsalgorithmus (Data Encryption Standard, Cipher Block Chaining) vorgeschlagen. Der IPv6-Verschlüsselungs-Hea-

Das Thema in Kürze

Der zweiteilige Beitrag gibt anhand eines Vergleichs mit der Version 4 einen Überblick über Funktionen, die für IP Version 6 entwickelt wurden. In *Teil 1* standen die Adressierung sowie Änderungen beim Daten-/Paketformat, die Angabe von Dienstgüten und die einfachere Konfigurierbarkeit von IP-Knoten im Mittelpunkt. *Teil 2* befaßt sich vor allem mit Übergangsstrategien von IPv4 auf IPv6. Darüber hinaus werden Vor- und Nachteile des NAT-Konzeptes diskutiert.

der enthält in diesem Fall neben dem generell in einem Verschlüsselungs-Kopf vorhandenen SPI-Feld und der Sequenznummer einen Initialisierungsvektor. Der Nutzlast, d.h. den verschlüsselten Daten, folgen Padding-Bytes, da die üblichen Verschlüsselungsalgorithmen auf 8-Byte-Grenzen arbeiten, sowie die Anzahl der zusätzlich erforderlichen Padding-Bytes. Daten können mit dem Tunnel-Modus und dem Transport-Modus verschlüsselt übertragen werden. Beim Tunnel-Modus wird das gesamte IP-Paket verschlüsselt und ein neuer unverschlüsselter IP-Header erzeugt, während beim Transport-Modus nur die Nutzlast und die Ende-zu-Ende-Erweiterungs-Header verschlüsselt werden *(Bild 4)*.

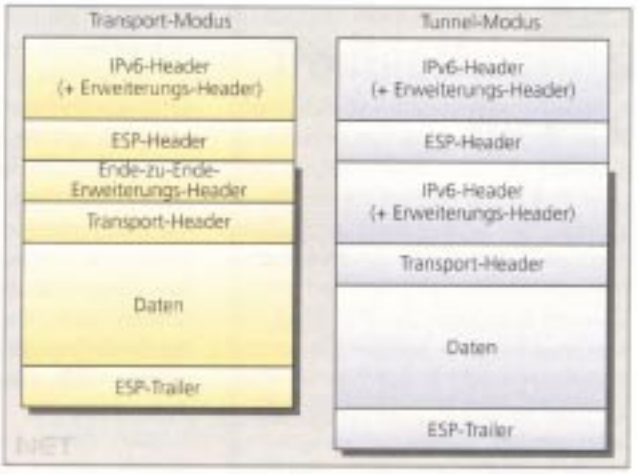

Bild 4: Verschlüsselung beim Transport-Modus und beim Tunnel-Modus

Auswirkungen auf Routing-Protokolle und Anwendungen

Das modifizierte IPv6-Adreßformat erfordert zunächst die Anpassung der im IPv6-Umfeld verwendeten Routing-Protokolle wie zum Beispiel RIPng und OSPFv6. Neben den Routing-Protokollen muß jedes Transportprotokoll oder andere höhere Protokolle, welche IP-Adressen zur Berechnung ihrer Prüfsumme verwenden, aufgrund des veränderten IPv6-Adreßformats modifiziert werden. Außerdem erfordern die neuen Adreßformate Änderungen an der Socket-Schnittstelle, die sich wiederum in notwendigen Anpassungen der darauf aufsetzenden Anwendungen auswirken.

Domain Name System

Das Domain Name System (DNS) wird sowohl in IPv4 als auch in IPv6 benutzt, um Rechnernamen auf IP-Adressen abzubilden. Für IPv6 wurde ein neuer Eintrag, ein sogenannter Resource Record (RR) mit der Kennung AAAA im Gegensatz zu A für IPv4 definiert. Bei DNS-Anfragen mit der Bezeichnung AAAA wird dann die IPv6-Adresse vom DNS-Server zurückgeliefert, während A-Anfragen die IPv4-Adresse liefern. Auflösungsprozeduren (sogenannte Resolver-Routinen) in den Endsystemimplementierungen müssen so geändert werden, daß sie auch AAAA-Anfragen generieren und die Antworten entsprechend verarbeiten können. Da IPv6-Systeme üblicherweise einen doppelten Protokollstack, d.h. eine IPv4- und eine IPv6-Adresse besitzen, ist davon auszugehen, daß die meisten DNS-Antworten sowohl eine IPv4- als auch eine IPv6-Adresse beinhalten. Die lokale Auflösungsprozedur kann dann nur die IPv4-Adresse, nur die IPv6-Adresse oder eine Liste mit beiden Adressen, d.h. der IPv4- und der IPv6-Adresse, zurückgeben. Im letzten Fall kann die Anwendung eine der in der Liste enthaltenen Adressen selektieren.

In IPv4-Umgebungen waren DNS-Einträge bislang statischer Natur, d.h., sie konnten nur durch die Änderungen der entsprechenden Tabellen bzw. Dateien modifiziert werden. Diese Änderungen waren üblicherweise sehr selten, z.B. dann wenn ein neuer Knoten installiert wurde oder ein Rechner manuell umkonfiguriert wurde, z.B., der Rechnername oder die IPv4-Adresse geändert wurde. In IPv6-Umgebungen sind Adreßänderungen wegen der Möglichkeiten zur automatischen Adreßkonfiguration jedoch viel häufiger zu erwarten. Daher besteht die Notwendigkeit, auch den entsprechenden DNS-Eintrag zu modifizieren. Es wurde daher eine neue DNS-Update-Nachricht spezifiziert, um solche dynamischen DNS-Updates zu erlauben.

Mobilität

Ein immer wichtigeres Thema ist auch im Internet die Existenz mobiler Endgeräte. Die IETF hat hierzu das Mobile-IP-Protokoll sowohl für IPv4 als auch für IPv6 standardisiert. Mobile IP basiert darauf, daß ein mobiles Endgerät zwei Adressen benutzt: die (konstante) Heimatadresse sowie die lokationsabhängige Care-of-Adresse. Bei Mobile IPv4 müssen spezielle Server, sogenannte Foreign-Agenten installiert werden, von denen die mobilen IP-Knoten die Care-of-Adresse lernen können, während in Mobile IPv6 die Adressen durch die zustandslose automatische Adreßkonfiguration gelernt werden, was Foreign-Agenten überflüssig macht. Pakete, die an die Heimatadresse gerichtet sind, werden vom Heimatagenten erkannt und bei Mobile IPv4 per Tunnel (d.h. Erzeugen eines neuen IPv4-Pakets mit der Care-of-Adresse des mobilen Knotens) an den mobilen Knoten weitergeleitet. Mobile IPv6 kann die Tunnel-Technik durch den Einsatz des Routing-Headers vermeiden, indem in den Routing-Header die Care-of-Adresse des mobilen Knotens eingetragen wird.

Übergangsstrategien

Die Strategien zum Übergang von IPv4 auf IPv6 sollen eine inkrementelle und einfache Migration von IPv4 nach IPv6 ermöglichen. IPv4- und IPv6-Knoten sollen in einem gemeinsamen Netz existieren können. In einer Übergangsphase werden daher alle IPv6-Knoten auch einen IPv4-basierten Protokollstack besitzen, um mit der großen Anzahl von reinen IPv4-basierten Knoten weiterhin kommunizieren zu können.

Zur Kommunikation der IPv6-Systeme mit den reinen IPv4-Systemen wird

zunächst das Konzept der doppelten Protokollstacks eingesetzt. Doppelte Protokollstacks benutzen die sogenannten IPv4-mapped IPv6-Adressen. Diese zeigen an, daß das adressierte System IPv6 nicht unterstützt, so daß über IPv4 kommuniziert werden muß. Wenn eine Anwendung eine IPv4-mapped Adresse an die IPv6-Implementierung übergibt, wird entweder in einem Endsystem der IPv4-Stack zur Kommunikation ausgewählt oder das IPv6-Paket muß z.B. durch einen Router (Übersetzer) ins IPv4-Format übersetzt werden.

Für IPv6 soll die bestehende IPv4-Routing-Infrastruktur zur Kommunikation zwischen IPv6-Knoten verwendet werden. Dazu wird die Tunnel-Technik angewendet. IPv6-Pakete werden hierzu in IPv4-Pakete eingepackt, d.h. mit einem zusätzlichen IPv4-Kopf versehen, und über einen oder mehrere IPv4-Router an den nächsten IPv6-fähigen Knoten übertragen. Letzterer streift den IPv4-Header wieder ab und verarbeitet das IPv6-Paket entsprechend. Dieser Knoten wird auch als Tunnel-Ende bezeichnet. Bei der Tunnel-Technik müssen beim Einpacken eines IPv6-Pakets die IPv4-Zieladressen je nach Situation unterschiedlich bestimmt werden. Ist das ursprüngliche Ziel auch gleichzeitig das Tunnel-Ende, so kann, sofern es sich bei der Zieladresse des IPv6-Zielknotens um eine IPv4-kompatible IPv6-Adresse handelt, aus dieser die IPv4-Adresse des Zielknotens extrahiert und diese als Zieladresse im IPv4-Header eingesetzt werden. Die IPv4-kompatiblen Adressen werden also in Situationen verwendet, in denen zwei IPv6-Systeme miteinander kommunizieren wollen, die dazwischen liegenden Router aber nur IPv4-fähig sind. Man spricht wegen der automatischen Abbildung der IPv6-Adresse auf die IPv4-Adresse auch von automatischem Tunneln. Ist das Tunnel-Ende jedoch ein Router, muß die IPv4-Zieladresse aus der Routing-Information beim Tunnel-Anfang ermittelt werden, was konfiguriertes Tunneln genannt wird. Das heutige IPv6-Backbone-Netz (6Bone) wird ähnlich wie das Multicast Backbone (Mbone) inkrementell ausgebaut, indem die zwischen IPv6-Routern konfigurierten Tunnels nach und nach deaktiviert werden.

Die genannten Strategien zum Übergang von IPv4 nach IPv6 sind solange geeignet, bis die IPv4-Adressen auslaufen. Danach können neuen IPv6-Knoten allerdings keine IPv4-Adresse mehr zugewiesen werden, so daß der Ansatz der doppelten Protokollstacks zur Kommunikation zwischen IPv4-Knoten und IPv6-Knoten nicht in der beschriebenen Form angewendet werden kann. Für den Zeitraum nach dem Auslaufen der IPv4-Adressen müssen daher neue Verfahren entwickelt werden, da bestimmt nicht alle IPv4-Knoten auf IPv6 umgestellt werden und es zumindest in absehbarer Zeit immer reine IPv4-Knoten wie z.B. Drucker geben wird. In der IETF-Arbeitsgruppe Next Generation Transition (ngtrans) werden derzeit verschiedene Ansätze diskutiert.

Der erste Ansatz No Network Address Translation (NNAT) basiert auf der Strategie der doppelten Protokollstacks und der Annahme, daß jeder IPv6-Protokollstack auch eine IPv4-Protokollstack-Implementierung besitzt. Allerdings wird einem solchen Knoten nach dem Auslaufen der IPv4-Adressen keine permanente IPv4-Adresse zugewiesen. Vielmehr gibt es eine Art Pool freier IPv4-Adressen, wobei jedem IPv6-Knoten temporär eine IPv4-Adresse aus dem Pool zugewiesen wird, um die Kommunikation mit einem reinen IPv4-Knoten durchzuführen. Die Kommunikation erfolgt dann wie bei einem Szenario mit doppelten Protokollstacks. Der NNAT-Ansatz ist mit dem unten beschriebenen NAT-Ansatz vergleichbar.

Bei der Header-Translation-Strategie werden zwischen IPv4-Netzen und IPv6-Netzen Übersetzer (Header Translator) eingesetzt, die IPv4- und ICMPv4-Paket in IPv6- bzw. ICMPv6-Pakete übersetzen. Im Gegensatz zu Application Level Gateways, d.h. Anwendungs-Gateways, die oberhalb der TCP- bzw. UDP-Schicht Pakete umsetzen (z.B. Proxy-Server oder Firewalls), werden bei einem Header Translator die IP-Pakete direkt auf der IP-Schicht übersetzt. Die Übersetzung von IP-Paketen ist eine relativ schwierige Aufgabe, insbesondere wenn bestimmte IP-Optionen nur in einer der beiden IP-Versionen vorkommen und eine funktional äquivalente Option in der anderen Protokoll-Version fehlt. Aus diesem Grund wird auf die Übersetzung der IPv6-Erweiterungs-Header Routing-Header, Hop-by-Hop-Header und Destination-Options-Header verzichtet. Authentifizierungs-Header und Verschlüsselungs-Header sind für einen Header-Translator transparent und können auch von einem IPSec-kompatiblen IPv4-Endsystem verarbeitet werden. Der einzige Erweiterungs-Header, der daher von einem Übersetzer transformiert werden muß, ist der Fragment-Header.

Die wesentliche Aufgabe eines Übersetzers ist die Übersetzung der IPv4- und IPv6-Adressen. Hierzu muß ein IPv6-Endsystem eine IPv4-Adresse z.B. per DHCP temporär allokieren und die resultierende IPv4-kompatible IPv6-Adresse als IPv6-Quelladresse verwenden. Als Zieladresse wird die sich aus der IPv4-Adresse des IPv4-Endsystems ergebende IPv4-mapped IPv6-Adresse verwendet. Ein bei einem Übersetzer eintreffendes Paket mit einer IPv4-mapped IPv6-Adresse muß dann in ein IPv4-Paket umgewandelt werden. In der Gegenrichtung muß der Übersetzer wissen, welche IPv4-Adressen temporär zur Bildung einer IPv4-mapped IPv6-Adresse allokiert wurden. Eintreffende Pakete mit einer Zieladresse aus dieser Adreßmenge müssen dann in ein IPv6-Paket mit reiner IPv6-Adresse umgewandelt werden.

Network Address Translator (NATs)

Die Installation eines Network Address Translators (NAT) ist eine Möglichkeit, der Umstellung auf IPv6 zunächst aus dem Wege zu gehen. NATs arbeiten auf der IP-Ebene, d.h. sie modifizieren die IP-Adressen in den IP-Paketen, die weitergeleitet werden. Üblicherweise arbeiten NATs unidirektional, d.h. eine Sitzung (alle zu einer Anwendung gehörenden Pakete) muß vom System hinter dem NAT, d.h. im privaten Netz, initiiert werden *(Bild 5)*. Sobald der NAT bemerkt hat, daß eine neue Sitzung initiiert wurde, reserviert er aus seinem Pool von öffentlichen IP-

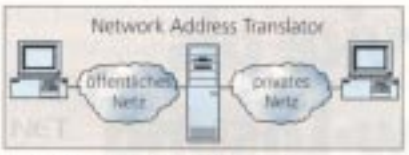

Bild 5: Network Address Translator

Adressen (oft nur eine einzige IP-Adresse) und Port-Nummern eine Adreßkombination und bildet die private Adresse auf die selektierte Adreßkombination ab. Die Abbildung ist dann für beide Richtungen, d.h. ausgehende Pakete und auch einkommende Pakete, gültig. Problematisch ist die Freigabe der öffentlichen Adresse, die dann erfolgen sollte, wenn die Sitzung beendet wurde. Oft ist dies nur mit Timeout-Mechanismen möglich.

Ebenfalls problematisch stellt sich der Fall dar, wenn ein System hinter einem NAT, d.h. im privaten Netz kontaktiert werden soll, wie dies bei Servern oder in IP-Telefoniearwendungen der Fall ist. Dies ist nur mit bidirektionalen NATs unterstützbar, wobei die bidirektionalen NATs einen DNS-Abbildungsmechanismus beinhalten müssen. Noch kompliziertere Situationen entstehen in Protokollen wie z.B. ftp, wo IP-Adressen in den Nutzdaten vorkommen können. In diesem Fall müssen NATs die Nutzdaten nach zu übersetzenden IP-Adressen abprüfen und diese ggf. modifizieren. Des weiteren können IP-Adressen in Kontrollprotokollen wie z.B. RSVP, H.323 usw. vorkommen. NATs müssen in diesen Fällen auch dies erkennen können und die Übersetzung durchführen. Speziell in den immer populärer werdenden Anwendungen wie Internet-Telefonie oder Audio/Video-Streaming stellt dies ein signifikantes Problem dar. Oft muß für jede neue Anwendung ein spezielles Application Level Gateway im NAT installiert werden, um diese über ein NAT hinweg betreiben zu können.

Auch Firewalls benötigen wie NATs nur eine öffentliche IP-Adresse. Firewalls sind meist als Application Level Gateways realisiert, welche typischerweise am Übergang zwischen Firmennetz und öffentlichem Internet installiert werden und TCP-Verbindungen terminieren. Ein Firewall-System benötigt dann wie ein NAT nur eine einzige IPv6-Adresse. Allerdings haben Firewalls auch signifikante Nachteile, die viele Benutzer – speziell im Forschungsumfeld, aber auch viele private Benutzer, die an Internet-Provider angeschlossen sind – nicht gerne hinnehmen. Hierzu gehören beispielsweise Leistungseinbußen durch die Firewall-Verarbeitung, die mangelnde Unterstützung von Multicast-Kommunikation durch Firewalls und die Tatsache, daß viele Firewalls UDP-Verkehr blockieren.

Ausblick

Insgesamt bleibt festzuhalten, daß NATs und Application Level Gateways in Firewalls heutzutage sicherlich ein geeignetes Mittel sind, um die Einführung von IPv6 hinauszuschieben. Der dahinter stehende technische Aufwand und die beträchtlichen funktionalen Einschränkungen lassen jedoch starke Zweifel aufkommen, ob dies der richtige Weg ist, um das Adreßproblem im Internet mittel- oder längerfristig zu beheben. Die Entwicklung und Benutzung von neuen Internet-Anwendungen wird mit Sicherheit behindert, wertvolle neue IPv6-Eigenschaften und Funktionen wie die automatische Adreßkonfiguration werden nicht verwendet, und die Integration neuer Funktionalitäten in IPv4 erfordert immer größere Tricks, so daß aus technischer Sicht fast alles für die beschleunigte Einführung von IPv6 spricht.

Universität Bern
Institut für Informatik und angewandte Mathematik
Tel.: (00 41) 3 16 31-49 94
Fax: (00 41) 3 16 31-39 65
braun@iam.unibe.ch
www.iam.unibe.ch/~braun