Tor als Prüfstein der Data Retention Richtlinie

Lukas Feiler, Mai 2005

Inhalt

1. Einleitung

Tor ist ein Projekt der Electronic Frontier Foundation (EFF)[1] mit dem Ziel die Privatsphäre im Internet zu erhöhen. Tor (The Onion Router) verwendet ein als „Onion Routing“[2] bekanntes Konzept um die Anonymität von Kommunikationspartnern im Internet zu gewährleisten. Das Konzept des Onion Routing versucht die Identifikation von Kommunikationsteilnehmern vom Routing der Kommunikationsdaten zu trennen. Identifikation und Routing scheinen deshalb untrennbar, da jeder Router im Internet die IP-Adresse des Empfängers benötigt, um Datenpakte an diese weiterleiten zu können. Dies entspricht dem selbstverständlichen Erfordernis der Angabe eines Empfängers auf einem traditionellen Brief. Denn erst durch diese Angabe ist es einem Postamt möglich den Brief dem Empfänger zuzustellen.

2. Eine Darstellung einer beispielhaften Kommunikation mittels Tor

Tor ist derzeit die am weitesten entwickelte Implementierung eines Onion Routing. Im Folgenden soll die beispielhafte Kommunikation mittels Tor zwischen einem Client und dem Server example.com[3] erläutert werden. Dies soll dazu dienen das Konzept des Onion Routing zu veranschaulichen.[4] Anzumerken ist jedoch, dass Tor ausschließlich für den Transport des Protokolls TCP – nicht jedoch für ICMP oder UDP – geeignet ist. Dies ist weitgehend unproblematisch, da die meisten der verwendeten Anwendungsprotokolle ohnedies auf TCP aufbauen.[5]

1) Zunächst wählt der Client nach zufälligen Kriterien eine Entry Node, eine Middle Node und eine Exit Node aus einer Liste verfügbarer Nodes aus. Jede dieser Nodes (zu Deutsch Knotenpunkte) ist ein Server des Tor-Netzwerkes. Der Client verbindet sich mit allen drei Nodes und fordert von jeder ihren Public Key an.

2) Im zweiten Schritt verschlüsselt der Client das an example.com zu übertragende Paket dreifach – zuerst mit dem Public Key der Exit Node, nachfolgend mit dem Public Key der Middle Node und abschließend mit dem Public Key der Entry Node. Das daraus resultierende Paket wird nun an die Entry Node übertragen.

---

[1] http://www.eff.org
[2] vgl. http://www.onion-router.net
[3] vgl. RFC 2606; ftp://ftp.rfc-editor.org/in-notes/rfc2606.txt
[4] vgl. Roger Dingledine, Berkman Center for Internet & Society at Harvard Law School, The Filter, April 2006, http://cyber.law.harvard.edu/audio/uploads/45/41/dingledine_2006-03-14.mp3; TheOnionRouter/TorFAQ, http://wiki.noreply.org/noreply/TheOnionRouter/TorFAQ
[5] Da DNS grundsätzlich UDP verwendet, implementiert Tor zusätzliche Mechanismen um ein “DNS leakage“ zu verhindern; vgl. http://wiki.noreply.org/noreply/TheOnionRouter/TorFAQ#SOCKSAndDNS.

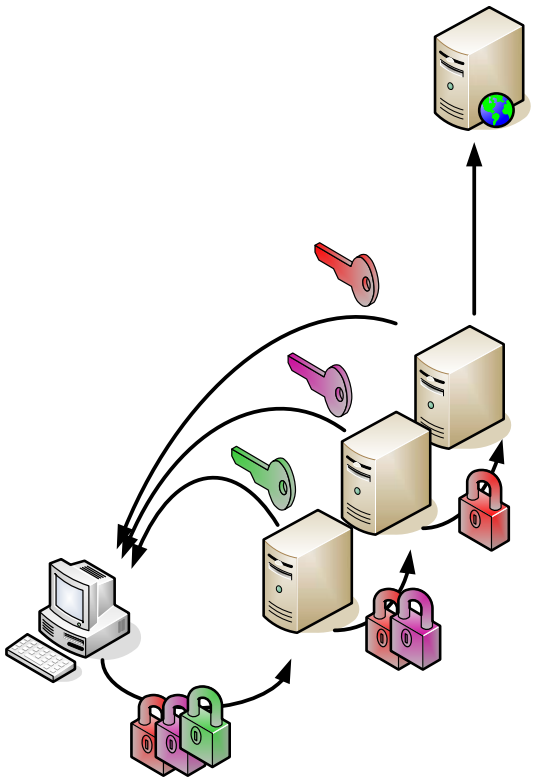
3) Die zuletzt vom Client durchgeführt Verschlüsselung (gleichsam die äußerste Schale der Zwiebel) kann nun von der Entry Node aufgehoben werden, da diese (als einzige) über den passenden Private Key verfügt. Das Ergebnis der Entschlüsselung ist ein noch immer zweifach verschlüsseltes Datenpaket und die Information an welche Middle Node dieses weiter zu leiten ist.

4) Die Middle Node kann nun das von der Entry Node enthaltene Paket mit ihrem Private Key entschlüsseln, wodurch gleichsam die zweit-äußerste Schale der Zwiebel entfernt wird. Das Ergebnis der Entschlüsselung ist ein nunmehr nur noch mit dem Private Key der Exit Node verschlüsseltes Paket und die Information an welche Exit Node das Paket zu übertragen ist.

5) Die Exit Node erhält das nur noch mit ihrem Public Key verschlüsselte Datenpaket und kann dieses daher mit ihrem Private Key gänzlich entschlüsseln. Das Ergebnis dieser Entschlüsselung ist das Datenpaket, wie es an den eigentlichen Empfänger example.com zu senden ist.

6) Der Server example.com erhält das unverschlüsselte Datenpaket von der Exit Node übermittelt. Aus seiner Sicht scheint sein Kommunikationspartner daher die Exit Node zu sein.

Diese sechs Schritte werden in folgender Abbildung dargestellt:



Entscheidend ist hierbei, dass keine der Nodes sowohl die Identität des Clients als auch jene des Servers kennt. Die Entry Node kennt nur die Identität des Clients und die der Middle Node, nicht jedoch jene der Exit Node oder jene des Servers. Die Middle Node kennt nur die Identität der Entry Node und jene der Exit Node, jedoch weder die Identität des Clients noch

die des Servers. Die Exit Node kennt zwar die Identität des Servers und die der Middle Node aber nicht jene der Entry Node oder des Clients.
Dies bedeutet im Ergebnis, dass die Kompromittierung einer einzelnen Node es keinesfalls ermöglicht beide Kommunikationspartner festzustellen. Ähnliches gilt für die beteiligten Access Provider.
Der Access Provider des Clients kann lediglich feststellen, dass dieser zu drei verschiedenen anderen Systemen eine verschlüsselte Verbindung aufbaut – dass über diese Schlüssel angefordert werden, ist jedoch nicht erkennbar. Weiters kann der Access Provider des Clients feststellen, dass eine zweite verschlüsselte Verbindung mit der Entry Node hergestellt wird.
Der Access Provider der Entry Node kann nur eine verschlüsselte Verbindung vom Client zur Entry Node und von dieser zur Middle Node registrieren.
Der Access Provider der Middle Node „sieht" nur verschlüsselte Verbindungen von der Entry Node zur Middle Node und von dieser zur Exit Node.
Der Access Provider der Exit Node registriert nur eine verschlüsselte Verbindung von der Middle Node zur Exit Node und von dieser zum Server (example.com).
Der Access Provider des Servers „sieht" nur eine (unverschlüsselte) Verbindung von der Exit Node zum Server.

Die folgende Tabelle verdeutlicht welche Verbindungen für welchen Access Provider (AP) sichtbar, dh wahrnehmbar sind:

	Client – Entry Node	Entry Node – Middle Node	Middle Node – Exit Node	Exit Node - Server
AP Client	sichtbar			
AP Entry Node	sichtbar	sichtbar		
AP Middle Node		sichtbar	sichtbar	
AP Exit Node			sichtbar	sichtbar
AP Server				sichtbar

Dies zeigt, dass es keinem Access Provider alleine möglich ist, festzustellen, ob eine Kommunikation zwischen einem bestimmten Tor-Client und einem bestimmten Server stattfindet.

Selbst in jenem Fall in dem Client und Server denselben Access Provider verwenden, wäre es nur unter aufwändigen Verfahren uU möglich eine Relation zwischen den Datenpaketen festzustellen. Da die vom Client versendeten Pakete verschlüsselt sind, ist es ausschließlich möglich anhand zeitlicher Nähe zweier Pakete Relationen herzustellen. Angesichts der äußerst großen Anzahl der von einem Access Provider pro Sekunde weitergeleiteten Pakete scheint dies jedoch mehr als unwahrscheinlich.

3. Der Verbreitungsgrad von Tor
Bereits im November 2005 zählte Tor ca 120.000 User. Schätzungen der derzeitigen Benutzeranzahl belaufen sich auf 200.000 bis 300.000.[6] Dies führt vor Augen, dass es sich bei Tor keinesfalls um eine Randerscheinung handelt.
Einer der schwerwiegendsten Gründe Tor noch nicht einzusetzen, ist derzeit die schlechte Performance. Neben der notwendigen Umleitung über drei Nodes ist insbesondere die geringe Anzahl von zur Verfügung stehender Exit Nodes[7] hierfür ausschlaggebend.

[6] vgl. Roger Dingledine, Berkman Center for Internet & Society at Harvard Law School, The Filter, April 2006, http://cyber.law.harvard.edu/audio/uploads/45/41/dingledine_2006-03-14.mp3
[7] derzeit ca 500; vgl. http://wiki.noreply.org/noreply/TheOnionRouter/TorFAQ#head-ab553254a25232ea559bd65532da8a63e7b5f411

Im Zuge der Umsetzung der Data Retention RL ist zu erwarten, dass sich auch in Europa ein verstärktes Bedürfnis nach anonymer Internetkommunikation entwickelt. Ein verbreiteter Einsatz von Tor im europäischen Raum erscheint daher als durchaus wahrscheinlich.

4. Zusammenfassung

Tor führt sehr deutlich vor Augen, dass eine umfassende Vorratsspeicherung von Verbindungsdaten durch den Einsatz entsprechender technischer Mittel leicht umgangen werden kann. Da Tor nicht nur rechtstreuen Bürgern, sondern in gleicher Weise Terroristen oder anderen Verbrechen zur Verfügung steht, ist jedenfalls zu bezweifeln, ob sich die Data Rentention RL als effektiv im Kampf gegen schwere Straftaten und den Terrorismus erweisen wird.