Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik

Alpenhorn - ein Protokoll zur Kommunikationsinitialisierung ohne Metadaten preiszugeben

Chris Becker

Leipzig, 19. Juni 2017

Inhaltsverzeichnis

1 Einleitung

> "We kill people based on metadata"
> - General Michael Hayden,
> former director of the NSA and the CIA (vgl. [1])

Spätestens mit den Enthüllungen von Edward Snowden [2], durch die das Abhören von weit verbreiteten Kommunikationsmitteln weltweit bekannt wurde, ist die Problematik einer möglichst privaten und sicheren Kommunikation in den Aufmerksamkeitsradius einer breiteren Öffentlichkeit geraten.
Eine große Bedeutung wird dabei nicht ausschließlich den Inhalten der Kommunikation beigemessen, sondern auch den Metadaten, also bspw. Zeitpunkt und Häufigkeit des Austausches, beteiligte Kommunikationspartner oder ob eine Kommunikation überhaupt stattgefunden hat.
Nicht zuletzt deswegen wird auf dem Gebiet der Kryptographie an neuen Protokollen geforscht, die das Kommunizieren vertraulich und abhörsicher ermöglichen sollen. Die Mehrzahl der Protokolle beschäftigt sich dabei mit dem eigentlichen Nachrichtenaustausch und dessen Absicherung. Eine Forschergruppe des Massachusetts Institute of Technology (MIT) hat im letzten Jahr einen neuen Ansatz für die (kryptographisch) sichere Initialisierung von der Kommunikation vorgestellt, bevor es zu einem Austausch von Botschaften kommt: Das Alpenhorn-Protokoll, welches in diesem Artikel kurz vorgestellt wird.

2 Protokoll

Kern des Veröffentlichung [3] der Forschergruppe am "MIT Computer Science and Artificial Intelligence Laboratory", also dem Labor für Informatik und künstliche Intelligenz ist das Protokoll selbst, welches auch auf der OSDI (Operating Systems Design and Implementation) 2016 [4] vorgestellt wurde.
Ziel von Alpenhorn ist es, dass zwei Kommunikationspartner, bspw. Alice und Bob genannt, eine sichere Kommunikation zwischen sich etablieren können und das beide sich der Identität des jeweils anderen sicher sind (s. [3], S.1). Dazu müssen verschiedene Vorkehrungen getroffen werden, die im Folgenden kurz vorgestellt werden.

Ein Angreifer kann z. B. bereits Informationen gewinnen, wenn Alice Bobs Informationen auf einem öffentlichen Server abfragt, damit sie weiß, wie sie mit ihm in Verbindung treten kann. Alpenhorn baut auf drei zentralen Ideen auf [3], S.2:

- Die Verwendung eines Adressbuches anstelle von langlebigen öffentlichen Schlüsseln: Ein Eintrag mit einem Freund enthält ein gemeinsames Geheimnis, dass sich über die Zeit verändert, wodurch es keinen angreifbaren, lange gültigen öffentlichen Schlüssel gibt.

- Alpenhorn verwendet "identity-based-encryption" (IBE), also identitätsbasierte Verschlüsselung [5], bei dem öffentlicher Schlüssel eines Nutzers aus dessen Mailadresse und dem öffentlichen Schlüssel eines Servers besteht. Das hat den Vorteil, dass man den öffentlichen Schlüssel eines Freundes berechnen kann, ohne seine Identität gegenüber dem Server preisgeben zu müssen.
- Alpenhorn verwendet ein "Schlüsselrad", um die geteilten Geheimnisse eines Nutzers mit anderen im Verlauf einer Konversation immer wieder zu verändern und so sicherzustellen, dass nicht auf vergangene Nachrichten oder Metadaten zugegriffen werden kann.

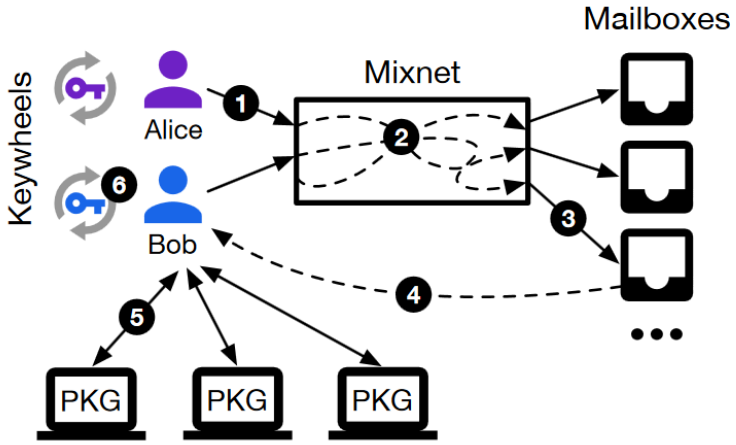

Abbildung 1: Aufbau des Alpenhorn-Protokolls (entnommen aus [3], S. 4).

Der Aufbau von Alpenhorn lässt sich an Abbildung 1 erkennen. Es gibt zwei Arten von Servern: Die "Private Key Generator" (PKG) und Mixnet-Server, s. [3], S. 4. Erstere dienen zum Ermitteln der öffentlichen Schlüssel eines Nutzers, indem sie IBE benutzen. Mixnet Server dienen dazu zu verschleiern, welcher Client (in der Abbildung Alice oder Bob) welche Anfrage gesendet hat. Dieses Hinzufügen von Rauschen baut auf vorhandenen Erkenntnissen des "vuvuzela"-Protokolls auf, [6].
Sowohl die Mixnet-Server als auch die PKG-Instanzen operieren in einem anytrust-Modell, das bedeutet, solange mindestens ein Server nicht kompromittiert ist, kann man sicher kommunizieren (vgl. [3], S.2). Des weiteren hatte der Testaufbau auch noch einen Eingangsserver, der allerdings nicht vertrauenswürdig sein muss, [3], S. 11.

Jeder Client hat ein Schlüsselrad, in dem die sich fortwährend ändernden geteilten Geheimnisse für die Kommunikation mit anderen Nutzern gespeichert sind. Die Funktionsweise des Schlüsselrades ist in Abbildung 2 dargestellt. K_r bezeichnet einen geteilten, geheimen Schlüssel im Schlüsselrad zur Runde r. H_i bezeichnet eine Familie von Werten einer kryptografischen Hashfunktion (einen Keyed-Hash Message Authentication Code (HMAC), beispielsweise HMAC-SHA256), wobei die Zahl für den jeweiligen Schlüssel steht, vgl. [3] S. 8.

Für alle Freunde eines Nutzers wird jede Runde im Schlüsselrad das neue geteilte Geheimnis berechnet. Jeder Client sendet auch Anfragen an den ersten Mixnet-Server in jeder Runde, selbst wenn der Nutzer keine Kommunikationsabsicht hat. Diese "falschen" Anfragen dienen als Rauschen, dass die Mixnetserver benutzen können, um das eigentliche Kommunikationsaufkommen zu verschleiern.

Freundschaftsanfragen werden in Mailboxen gespeichert, die die Clients herunterladen, um darin enthaltene verschlüsselten Anfragen zu dechiffrieren (vgl. [3], S.6). Lässt sich eine Anfrage erfolgreich entschlüsseln, so war sie auch für den Client bestimmt (da er den korrekten privaten Schlüssel besitzt). Falls die Entschlüsselung fehlschlägt, bedeutet das lediglich, dass diese Nachricht aus der Mailbox nicht für den Client bestimmt war (auf die Gesamtzahl der Nachrichten in einer Mailbox gesehen, können die meisten Nachrichten nicht entschlüsselt werden, wenn man viele Teilnehmer mit eher geringem individuellen Kommunikationsgesuchen voraussetzt).

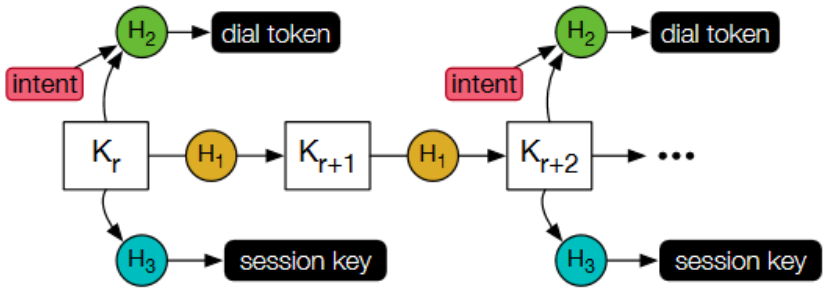

Abbildung 2: Funktionsweise des Schlüsselrads (entnommen aus [3], S. 8).

Alpenhorn unterscheidet zwischen einem "dialing"- und einem "add friend"-Protokoll. Das Protokoll für das Hinzufügen von Freunden verwendet ein asymmetrisches Kryptosystem und ist durch das Herunterladen der Mailboxen mit einer hohen Latenz verbunden, [3] S. 4. Das Anrufprotokoll wird dagegen zum Herstellen der Kommunikation an sich verwendet und bedient sich eines symmetrischen Kryptosystems, um mit niedriger Latenz Schlüssel auszutauschen, [3] S. 4.
Die Sicherheitsziele von Alpenhorn sind s. [3], S.5 :

- Authentifizierter Schlüsselaustausch: Wenn ein Angreifer einen Server übernommen hat und den Datenverkehr manipulieren kann, so darf er dennoch nicht in der Lage sein, die Sitzungsschlüssel der Nutzer von Alpenhorn zu erlangen oder sich als ein Nutzer auszugeben, auf dessen Mailadresse er keinen Zugriff hat.

- Geheimhaltung der Metadaten: Ein Angreifer soll keine Informationen über die Freundeslisten von einem Nutzer erlangen können, oder darüber, wer mit wem zu welchem Zeitpunkt eine Unterhaltung geführt wurde.

- Vorwärts gerichtete Geheimhaltung der Metadaten: Wenn ein Server oder Klient

kompromittiert ist, darf ein Angreifer keine Informationen über die Metadaten oder den Inhalt der Konversationen der Vergangenheit erlangen

Um Alpenhorn nutzen zu können, muss sich ein Nutzer mit seiner E-Mailadresse registrieren. Es werden verschiedene Maßnahmen getroffen, um bei einem Verlust (ob an einen Angreifer oder nicht) der Mailadresse Alpenhorn weiter nutzen zu können, s. [3], S.8.

3 Anwendung

Zum ersten Mal praktisch eingesetzt wurde Alpenhorn in Verbindung mit dem auch vom MIT entwickelten Kommunikationsprotokoll "vuvuzela" [7]. Einige Prinzipien von Alpenhorn bauen auf Entwicklungen für vuvuzela auf, so zum Beispiel das Leiten der Nachrichten über mehrere Server, von denen jeder einzelne Rauschen hinzufügt.

Als Programmiersprache wurde Go verwendet, und es wurde praktisch gezeigt, dass bei Verwendung von drei Servern als Infrastruktur 10 Millionen Nutzer kommunizieren können bei einem Overhead von 3,7 KB/s und einer Durchschnittslatenz von 150 Sekunden, bis ein Kommunikationswunsch an den zweiten Teilnehmer weitergeleitet ist (vgl. [3], S. 1).
Als Server wurden Rechner aus Amazons EC2 (Elastic Compute Cloud) verwendet, auf denen jeweils eine virtuelle Maschine zum Einsatz kam, so dass im Testaufbau, drei PKG-Instanzen und drei Instanzen für das Mischen und Hinzufügen von Rauschen (Mixnet-Server) zur Verfügung standen (s. [3], S. 11). Ein weiterer Vorteil des Mietens von Servern auf diese Weise ist, dass es einfach möglich ist, global verteilte Clients und Server bereitzustellen. Für die Experimente wurden Rechner in den Regionen Virginia (USA), Irland und Frankfurt (Deutschland) eingesetzt (s. [3], S. 11).

Da für die kryptografische Sicherheit der IBE der Anwendung die BN-256-Kurve verwendet wird, gegen die eine neue Art von Angriff entdeckt wurde [8], räumen die Forscher ein, dass das Protokoll in Zukunft vielleicht eine andere Kurve verwenden sollte, um die Sicherheitsgarantien zu gewährleisten. Auf die Anwendung sollte das lineare oder sublineare Auswirkungen haben (vgl.[3], S.13).

Die genaueren Auswertungen (inklusive Diagrammen zu Bandbreitenauslastung und Latenzen) lassen sich dem Artikel ([3], S. 11ff.) entnehmen.
Der Quellcode der Implementierung ist auf github einsehbar [9] und umfasst etwa 10.000 Zeilen.
Weitere Details zu den verwendeten Algorithmen, ein Beweis zur Sicherheit der Anytrust-IBE-Server und einen Einblick in die API befinden sich ebenfalls in [3].

Quellen

[1] MICHAEL HAYDEN: *'We Kill People Based on Metadata'*. URL: http://www.nybooks.com/daily/2014/05/10/we-kill-people-based-metadata/ (besucht am 12. 06. 2017).

[2] THE GUARDIAN: *the NSA files*. URL: https://www.theguardian.com/us-news/the-nsa-files (besucht am 12. 06. 2017).

[3] DAVID LAZAR und NICKOLAI ZELDOVICH: „Alpenhorn: Bootstrapping Secure Communication without Leaking Metadata". In: *Proceedings of the 12th Symposium on Op- erating Systems Design and Implementation (OSDI)* (Nov. 2016).

[4] USENIX ASSOCIATION: *12th USENIX Symposium on Operating Systems Design and Implementation*. URL: https://www.usenix.org/conference/osdi16 (besucht am 12. 06. 2017).

[5] A. SHAMIR: „Identity-based cryptosystems and signature schemes". In: *Proceedings of the 4th Annual International Cryptology Conference (CRYPTO)* (1984).

[6] J. van den HOOFF u. a.: „Vuvuzela: Scalable private messaging resistant to traffic analysis". In: *Proceedings of the 25th ACM Symposium on Operating Systems Principles (SOSP)* (Okt. 2015).

[7] MIT COMPUTER SCIENCE und ARTIFICIAL INTELLIGENCE LABORATORY (CSAIL): *Vuvuzela.io*. URL: https://vuvuzela.io/ (besucht am 12. 06. 2017).

[8] T. KIM und R. BARBULESCU: „Extended tower number field sieve: A new complexity for the medium prime case". In: *Proceedings of the 36th Annual International Cryptology Conference (CRYPTO)* (Aug. 2016).

[9] DAVID LAZAR: *vuvuzela / alpenhorn: Bootstrapping Secure Communication without Leaking Metadata*. URL: https://github.com/vuvuzela/alpenhorn (besucht am 12. 06. 2017).

Abbildungen