Dezentralisierte asymmetrische Verschlüsselung über Tor

Die Lösung für sicheres Messaging?

Hendrik Lind

Facharbeit

Windthorst-Gymnasium Meppen

Seminarfach Informatik

21. Januar 2024

Inhaltsverzeichnis

1 Einleitung

Russland, China, Iran. In all diesen totalitären Staaten herrscht eine starke Zensur [**AmnReport**]. Rund 1,7 Milliarden Menschen sind allein nur in diesen drei Staaten von der Einschränkung der Meinungsfreiheit betroffen [**UnPop**]. Wie können Bürger dieser Staaten ihre Meinung also verbreiten und andere Staaten auf staatskritische Probleme aufmerksam machen ohne sich selber in Gefahr zu bringen?

Bei herkömmlichen Messengern, wie Whatsapp, Signal und co., braucht die Außenwelt die Telefonnummern der im totalitären Staat wohnenden Bürgern und Reportern, um diese zu kontaktieren. Allerdings könnte ein totalitärer Staat, sich als Empfänger ausgeben, sodass Bürger/Reporter ihre private Nummer an den Staat überreichen und dieser somit jene Nummer rückverfolgen kann [**LocPolice**]. Und genau hier liegt das Problem: Bürger und Reporter können nicht durch alltägliche Messenger mit der Außenwelt kommunizieren, da der Staat deren Nummer zurückverfolgen kann und somit weiter die Meinungsfreiheit einschränkt und unterbindet.

Durch die zentrale Infrastruktur, welche die meisten Messenger, wie zum Beispiel WhatsApp und Signal verwenden, ist es für totalitäre Staaten, wie China, möglich, die IP-Adressen jener Server zu blockieren und somit für Bürger und Reporter unzugänglich zu machen [**ChinaFirewall**].

Ein dezentralisierter Messenger, welcher Ende-zu-Ende verschlüsselt ist und über das Tor-Netzwerk kommuniziert, könnte bei diesen Problemen eine Lösung sein. Die Frage, ob ein solcher Messenger die Lösung für Bürger eines totalitären Staates ist, soll in dieser Arbeit geklärt werden.

Um diese Frage beantworten zu können, beschäftigt sich diese Arbeit in dem zweiten Kapitel mit der asymmetrischen Verschlüsselung, welche benötigt wird um die Ende-zu-Ende-Verschlüsselung (E2EE) umzusetzen und die Definition der E2EE und asymmetrischen Verschlüsselung [**E2EE-Method**]. Das dritte Kapitel beinhaltet eine mögliche Lösung, um eine Anonymität über das Internet zu gewährleisten, wobei das Tor-Netzwerk eine wichtige Rolle spielt. Im vierte Kapitel befasst sich diese Arbeit mit einer Dezentralisierung der Infrastruktur, um eine weitere Sicherheitsebene zu schaffen. Zuletzt werden im fünften Kapitel die Vor- und Nachteile eines solchen Messengers betrachtet, im sechsten Kapitel wird eine mögliche Umsetzung

des Messengers beschrieben und im siebten Kapitel wird ein Fazit gezogen.

2 Asymmetrische Verschlüsselung

Um einen sicheren Nachrichtenaustausch zu gewährleisten, wird in dieser Arbeit die E2EE implementiert. Bei der E2EE wird von dem Sender die Nachricht, bevor sie an den Empfänger geschickt wird, verschlüsselt [**E2EE**]. Dazwischenliegende Akteure, wie zum Beispiel Server oder mögliche Angreifer, können demzufolge die Nachricht nicht lesen. **Nur** der Empfänger der Nachricht kann diese auch entschlüsseln. Als Ent- und Verschlüsselungsverfahren der Nachrichten wird die asymmetrische Verschlüsselung verwendet. Für diese Arbeit werde ich die gängigste asymmetrische Verschlüsselung, das RSA-Verfahren, verwenden.

2.1 Grundlagen

Grundsätzlich gibt es bei der asymmetrischen Verschlüsselung ein Schlüsselpaar (Keypair), welches aus einem privaten Schlüssel (private key) und einem öffentlichen Schlüssel (public key) besteht [**Rsa-Basics**]. Diese beiden Schlüssel hängen mathematisch zusammen, sodass der öffentliche Schlüssel Nachrichten **nur** verschlüsseln und nicht entschlüsseln kann. **Nur** der zum Schlüsselpaar dazugehörige private Schlüssel ist in der Lage, die verschlüsselte Nachricht wieder zu entschlüsseln (siehe Abb. 1).

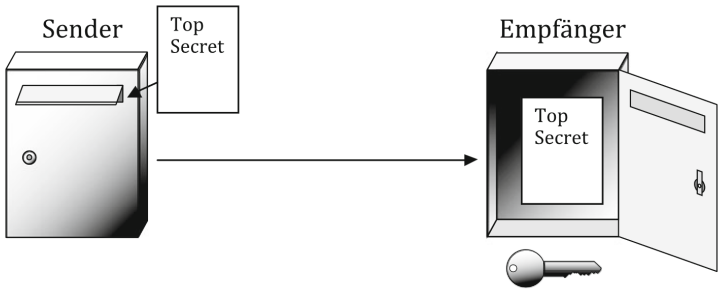

Abbildung 1: Jeder Sender kann mit dem öffentlichen Schlüssel die Nachricht „verschlüsseln“ (also in den Briefkasten eine Nachricht werfen), aber nur der Empfänger kann den Briefkasten mit seinem privaten Schlüssel öffnen [**fig:Rsa-Cryptography**]

2.2 Mathematische Betrachtung

Alle Variablen der folgenden Berechnungen liegen im Bereich $\mathbb{N}$.

Für die Generierung des Schlüsselpaares benötigen wir zuerst zwei große zufällige Primzahlen, P und Q. Daraus ergibt sich $n = P * Q$, wobei $P \neq Q$, sodass P bzw. Q nicht durch $\sqrt{n}$ ermittelt werden kann [**RsaGenCond**]. Der private Schlüssel besteht aus den Komponenten $\{n, d\}$ währenddessen der öffentliche Schlüssel aus $\{n, e\}$ besteht [**RsaVariables**].

2.2.1 Eulersche Phi-Funktion

Die Eulersche Phi-Funktion spielt eine wichtige Rolle in dem RSA-Verfahren [**TotientFuncMultiplicative**]. Grundsätzlich gibt $\phi(x)$ an, wie viele positive teilerfremde Zahlen bis x existieren (wo also der größte gemeinsamer Teiler (gcd) 1 ist) [**EulersTotientFunction**]. Somit ergibt $\phi(6) = 2$ oder bei einer Primzahl $\phi(7) = 7 - 1 = 6$ somit $\phi(x) = x - 1$, wenn x eine Primzahl ist, da jede Zahl kleiner als x teilerfremd sein muss.

$$\begin{aligned}
\phi(n) &= \phi(P \cdot Q) \\
\phi(n) &= \phi(P) \cdot \phi(Q) \\
\phi(P) &= P - 1 \qquad\qquad \phi(Q) = Q - 1 \\
\phi(n) &= (P - 1) \cdot (Q - 1)
\end{aligned}$$

[**TotientFuncMultiplicative**]

2.2.2 Generierung des Schlüsselpaares

Sowohl der private als auch der öffentliche Schlüssel besteht unter anderem aus folgender Komponente: $n = P \cdot Q$. Für den öffentlichen Schlüssel benötigen wir die Komponente e, welche zur Verschlüsselung einer Nachricht benötigt wird. e ist hierbei eine zufällige Zahl, bei welcher folgende Bedingungen gelten [**RsaMaths1**]:

$$e = \begin{cases} 1 < e < \phi(n) \\ \text{gcd}(e, \phi(n)) = 1 \\ e \text{ kein Teiler von } \phi(n) \end{cases}$$

Mit der errechneten Komponente e, welche Nachrichten verschlüsselt, kann der öffentliche Schlüssel nun an den Sender übermittelt werden.

Um den privaten Schlüssel zu berechnen benötigen wir die Komponente d, welche zur Entschlüsselung verwendet wird [**RsaEncryptionDecryption**].

$$\phi(n) = (P-1)(Q-1)$$
$$e * d = 1 \mod \phi(n)$$

2.3 Sicherheit

Das RSA-Verfahren macht sich die Trapdoor-Einwegfunktion zu Nutze. Das bedeutet, dass eine Funktion mit modernen Computern unmöglich ist, zu invertieren, wenn eine Komponente fehlt (in diesem Fall d). Wenn diese jedoch gegeben ist, ist die Umkehroperation leicht. Zu sehen ist dies bei der Ver-/Entschlüsselung von Nachrichten mit dem öffentlichen und privaten Schlüssel.

$c = m^e \mod n$ Verschlüsselung zu c mit m als Nachricht

$m = c^d \mod n$ Umkehroperation (Entschlüsslung) von c zu m

Um die verschlüsselte Nachricht c zu entschlüsseln, bräuchte ein möglicher Angreifer die Komponente des privaten Schlüssels d. Diese ist allerdings schwer zu berechnen, da, wie schon vorher bereits gezeigt, dafür $\phi(n)$ benötigt wird. Auch $\phi(n)$ ist rechenaufwendig, da dafür eine Primfaktorzerlegung von n benötigt wird. Wichtig hierbei ist aber, dass die Länge von n (Schlüssellänge) mindestens 3000 Bit betragen sollte, da sonst die Primfaktorzerlegung von n mit modernen Computern möglich sein könnte [**RsaKeyLength**].