Glosse

Kai Denker

Mit Sicherheit genervt

Unlängst habe ich versucht, einen alten Freund aus Jugendtagen zu erreichen. Er war vor Jahren nach Berlin gezogen, und ich hatte schon länger nichts mehr von ihm gehört. Die alte Telefonnummer ist nun nicht mehr gültig. Per E-Mail erhalte ich eine Antwort – allerdings von einer bisher unbekannten Adresse. Es gebe viel zu berichten, ließ mich der Absender wissen. Es sei alles in Ordnung. Aber mehr könne man über eine unsichere Leitung nicht sagen: Es sei größte Vorsicht geboten. Der Absender der Nachricht schlug mir eine Smartphone-App vor, über die man ›sicher‹ kommunizieren – gar telefonieren – könne. Vorher indes, so betonte die E-Mail noch einmal, könne man wirklich nicht mehr sagen.

Eine seltsame Sache, befinde ich. Fast vage genug für eine Phishingmail, mit der unbedarfte Computernutzer*innen zur Preisgabe sensibler Daten verleitet werden sollen, aber ungewöhnlich genug, um vielleicht doch authentisch zu sein, zumal man ja an meinen Versuch der Kontaktaufnahme kommunikativ anschloss. Meine Neugier ist also geweckt. Kaum habe ich die mir bis dahin unbekannte App installiert, wofür ich auf die Sicherheitsmaßnahmen von wenigstens vier internationalen, mir praktisch nur aus Werbeeinblendungen und von gelegentlichen Rechnungen bekannten Unternehmen und vermutlich einer Heerschar namenloser weiterer Programmier*innen vertrauen muss (etwas, was für mich und alle anderen Computernutzer*innen mittlerweile zur Gewohnheit geworden ist), erhalte ich die erste Chat-Nachricht von einer mir unbekannten Telefonnummer aus dem außereuropäischen Ausland: »Is this thing on?« – »It is. Who are you?«

Man müsse nun, so erklärt mir mein konspirativer Chatpartner – ich halte fest, dass er immerhin den richtigen Vornamen meines alten Freundes nutzt – die kryptografischen Schlüssel miteinander abgleichen, die die frisch installierte App völlig automatisch und gänzlich transparent generiert. Das sei leicht, und dann könne man endlich verschlüsselt, also ›abhörsicher‹ miteinander telefonieren. Denn vorher verbiete dies die App – aus Sicherheitsgründen. Auch ohne Schlüsselabgleich lässt sich eine Verbindung zwar verschlüsseln, aber man hat keine Informationen darüber, mit wem die verschlüsselte Verbindung besteht. Es könnte also genauso gut eine dritte Person – vielleicht ein Agent namens Mallory irgendeines amerikanischen Nachrichtendienstes – eine verschlüsselte Verbindung mit mir unterhalten, alle Nachrichten abfangen und über einen weiteren verschlüsselten Kanal an meine ebenso arglose

Kommunikationspartner*in senden. In der IT-Sicherheit spricht man beinahe selbsterklärend von einem ›man in the middle‹-Angriff.

Nun habe ich zwar Informatik studiert, habe also eine grobe Vorstellung davon, was in der App passiert sein muss, bevor sie mir nun einen QR-Code und darunter eine sechzigstellige Ziffernfolge (immerhin mit Leerzeichen und Zeilenumbrüchen recht übersichtlich) anzeigt. Wir können mit diesen Codes eine Vertrauenskette aufbauen, eine ›trust chain‹. Alles, was wir dazu zu tun haben, ist, mit dem einen Smartphone den QR-Code des anderen zu fotografieren.

Aufgrund der (wie die Telefonnummer nahelegt: erheblichen) Entfernung fällt dies allerdings definitiv aus. Alternativ hätten wir die sechzig Ziffern über einen anderen Kanal, auf dem wir die Identität des Gegenübers prüfen könnten, miteinander vergleichen können. E-Mail, unser einziges anderes Kommunikationsmedium zu diesem Zeitpunkt, reicht dafür nicht aus: Bekanntlich lassen sich unsignierte E-Mails leicht manipulieren, und Mallory besäße sicher ein komfortables Programm genau hierfür! Man kann eine E-Mail zwar natürlich signieren, aber um wiederum die Echtheit der E-Mail-Signatur prüfen zu können, müssen wir zunächst die Zuordnung der verwendeten Schlüssel zum ›Gegenüber‹ überprüfen – ›authentifizieren‹, wie man in der IT-Sicherheit sagt.

Die meisten Computernutzer*innen kennen dies aus dem Online-Banking: Das kleine, meist grüne Schloss in der Browser-Adresszeile soll signalisieren, dass der eigene Computer verschlüsselt mit dem Server der Bank kommuniziert, damit niemand die Bankgeschäfte abhören kann. Wie aber jeder weiß, muss stets auch geprüft werden, ob der Server auf der Gegenseite wirklich der Server der Bank ist. ›Cyber-Kriminelle‹ versuchen, dies auszunutzen, indem sie Links auf täuschend echt aussehende Kopien der Bank-Webseiten per E-Mail verschicken, auf denen sie zur Preisgabe von Kontodaten animieren oder zur Installation angeblich erforderlicher Software verleiten wollen – Phishing eben. Um dergleichen zu verhindern, sollte man die jeweilige Adresse genau kontrollieren, haben wir gelernt: nicht also einfach Links in E-Mails anklicken, außerdem die Quellen der eigenen Software genau prüfen und im Zweifelsfall auch einer als ›sicher‹ angezeigten Verbindung nicht einfach vertrauen. Glücklicherweise kontrollieren Computernutzer*innen heutzutage auch ohne Anlass dies alles immer wieder und vertrauen einer Webseite oder einer Software jedenfalls nicht schon allein deshalb, weil sie vom Computer mit einem kleinen grünen Schloss geschmückt wird.

Eine ganz ähnliche Authentifizierung steht zur Lösung unseres Problems noch aus: die Überprüfung der Identität meines konspirativen Chatpartners für mich – und meine Identität für ihn. Trotz aller global verfügbaren Kommunikationstechnik fehlt uns noch ein geeignetes Mittel zum sicheren (und hier vor allem: authentischen) Kontakt (und die Möglichkeit, einen alten Freund zu sprechen und von einer unerwarteten Geschichte zu hören, rückt in weite Ferne).

Wir überlegen und erproben weitere Optionen: Fotos? Nein. Verraten nicht selten den Aufenthaltsort und sind auch nicht vor Manipulation geschützt. Für Mallory sicher ein Klacks! Eine Videokonferenz? Nein. Auch diese verraten allzu schnell den Aufenthaltsort, und das will mein Gegenüber doch unbedingt vermeiden (als ob die E-Mail nicht schon gereicht hätte!). Überdies: technisch funktionierte Skype gerade sowieso nicht. Kann eine dritte Person oder Institution als Vertrauensanker, als ›trust anchor‹, fungieren? Wir wissen niemanden. An einem gemeinsamen ›web of trust‹, einem recht unübersichtlichen Netz von Vertrauensketten, in dem unbekannte Personen füreinander bürgen, nehmen wir auch nicht teil. Gibt es irgendein privates Geheimnis von früher, das als Erkennungszeichen dienen kann? Freilich: Welches? Was will man schon einem potentiell Fremden verraten, zumal eine geschickte Angreifer*in wie Mallory auch hier noch nicht mit ihrem Latein am Ende gewesen wäre?

Kurz: Mein konspirativer Chatpartner und ich haben ein Henne-Ei-Problem. Wir besitzen Techniken, um Vertrauen von einem Kanal auf den anderen zu übertragen, aber dieses Vertrauen ist nirgends verankert. Unsere letztendliche Lösung, nachdem wir noch einige weitere Möglichkeiten zur Authentifizierung durchgespielt haben, wird Sie enttäuschen und würde Mallory gelangweilt haben: Wir haben schlicht und klassisch miteinander telefoniert, unsere Stimmen erkannt (wie erwartet: er war es tatsächlich – oder Mallory imitiert Stimmen absolut überzeugend), wir haben uns als die alten Freunde auf diesem Wege identifiziert und noch abwechselnd die Ziffernfolge vorgelesen. Sie passte. Damit sind die Schlüssel authentifiziert. Das alles verrät zwar etwas mehr über ihn und mich als beabsichtigt, aber das wird uns nach einer guten Viertelstunde des Hin-und-Hers egal: Er will schließlich nur verschlüsselt telefonieren und endlich erzählen. Nach der zwar unverschlüsselten, aber dank Stimme und Genervtsein uns überzeugenden Authentifizierung fühlt mein alter Freund sich endlich ›sicher‹ genug. Authentifiziert und verschlüsselt erfahre ich eine kleine Revolvergeschichte.

Vielleicht fragen Sie sich nun, ob wir mit dem klassischen Telefonat nicht alle Sicherheitsmaßnahmen über Bord geworfen haben. Haben wir nicht. Warum? Damit die Verschlüsselung der App funktioniert, müssen sich beide Smartphones auf ein gemeinsames Geheimnis (in Form einer Zahl mit bestimmten algebraischen Eigenschaften)[1] einigen. Das geschieht mittels des Diffie-Hellmann-Verfahrens, das es erlaubt, gemeinsame Geheimnisse auch über abhörbare Leitungen miteinander auszu-

1 Diese Eigenschaften zu beschreiben ist nicht möglich, ohne das Format einer Glosse mit zu viel Fachwissenschaftlichem aus Zahlentheorie und Kryptographie zu überfordern, wofür ich selbst kein Experte bin. Um dies zu verdeutlichen, mag folgender Hinweis genügen: Wenn die beiden Smartphones a bzw. b als Zufallszahlen wählen, dann rechnen sie jeweils a und b als Exponent zu einer [öffentlich bekannten] Basis g, die in einer zyklischen Gruppe $\mathbb{Z}_p$ mit $g<p$, p ist Primzahl, ›Erzeuger‹ ist, einsetzen: $A=g^a \bmod p$ bzw. $B=g^b \bmod p$. A und B werden dann über die abhörbare Leitung übertragen. Da das Problem des diskreten Logarithmus (=das Logarithmieren

handeln. Ein so erzeugtes gemeinsames Geheimnis kann als Schlüssel dienen. Die vorgelesene Nummer ist aber kein solches Geheimnis, sondern nur dessen ›Fingerabdruck‹. Solche Fingerabdrücke werden mit speziellen Einwegfunktionen, sogenannten kryptographischen Hash-Funktionen, erzeugt, die es nahezu unmöglich machen, aus einem Fingerabdruck wieder das ursprüngliche Geheimnis, den Schlüssel also, zu rekonstruieren. Da sie für einen Schlüssel aber immer dasselbe Ergebnis liefern, kann man sie gut verwenden, um zu überprüfen, ob die Gegenstelle dasselbe Geheimnis kennt wie man selbst. Derartigen Algorithmen können wir glücklicherweise vertrauen, so dass unsere einzige Schwierigkeit nur noch darin bestand, sicherzugehen, dass jemand, den ich zuvor als meinen alten Freund identifizieren konnte (und umgekehrt), die gleiche Nummer angezeigt bekäme wie ich. Das ist in Nullkommanichts (also in etwa nach einer guten Viertelstunde und gerade mal zwei Dutzend Chatnachrichten) erledigt. Darüber kann ich mich also nicht beschweren. Die erforderlichen mathematischen Verfahren sind recht kompliziert und in ihren Details durchaus verzwickt, diesbezüglich denke ich mit Grausen an einschlägige Klausuren zurück. Es fängt damit an, dass solche Verfahren nur in der richtigen Kombination sicher und stets auf Zufallszahlen hoher Qualität angewiesen sind, was Computer vor einige Schwierigkeiten stellen kann, und es endet noch nicht bei der Tatsache, dass nicht jeder Schlüssel gleich sicher ist.[2] Für Programmier*innen gibt es so viele Möglichkeiten, fatale Fehler zu machen, dass die meisten sicherheitshalber auf eine der wenigen Programmbibliotheken für Verschlüsselungen zurückgreifen. Diese sind in den letzten Jahren glücklicherweise nicht allzu oft durch dramatische Sicherheitslücken aufgefallen, das meiste konnte schnell durch neue Versionen behoben werden. Diese müssen nur immer rechtzeitig in alle betroffenen Apps eingespielt werden. Auch das nimmt uns die App erfreulicherweise ab. Eine Mathematik, die jedes Jahr unzählige Informatik-Studierende zur Verzweiflung treibt, verschwindet ebenso unter dem eleganten Design einer schlichten Benutzeroberfläche

---

in diskreten mathematischen Strukturen) nicht gelöst ist, kann man aus A bzw. B nicht durch simples Logarithmieren in $\mathbb{Z}_p$ wieder a bzw. b rekonstruieren, zumindest nicht, ohne einfach alle möglichen a bzw. b durchzuprobieren, was sehr lang dauern kann. Nun rechnet die eine Seite $K_1=B^a \bmod p$ und die andere Seite $K_2=A^b \bmod p$. Da sich die Exponenten exponenzierter Potenzen multiplizieren, lies: $\forall x,y,z\in\mathbb{R}.(x^y)^z=x^{(y*z)}$, folgt: $K_1=g^{ba} \bmod p$ und für die andere Seite $K_2=g^{ab} \bmod p$. Da auch in $\mathbb{Z}_p$ die Multiplikation von Exponenten kommutativ ist, also $g^{ba} \bmod p=g^{ab} \bmod p$ ist, folgt $K_1=K_2$. Ergo ist $K=K_1=K_2$ das gemeinsame Geheimnis K. Tada!

2 Diesen letzten Punkt können Sie sich vielleicht so vorstellen: Wenn Sie ein Passwort erfinden sollen und dazu Zeichen zufällig auswählen, dann könnte Sie der Zufall auf das unsichere Passwort ›123456‹ führen. Passwortmanager nutzen daher oft einige Tricks, um solche unsicheren Passwörter zu vermeiden. Nur ausreichend lange Passwörter mit genügend ›Entropie‹ sind sicher, aber leider meist auch schlecht zu merken. Das ist gerade dann ärgerlich, wenn man sie regelmäßig ändern soll. Erlauben Sie mir einen ernstgemeinten Hinweis am Rande: Lassen Sie sich Passwörter von einem Passwortmanager erzeugen und speichern. Selbstausgedachte Passwörter sind praktisch immer weniger sicher. Und verwenden Sie unbedingt für jede Webseite und jeden Account ein eigenes Passwort. Mit einem guten Passwortmanager können Sie sich viel Ärger ersparen und die Kontrolle über Ihre Passwörter behalten.

wie die lästige Aufgabe, Bibliotheken immer auf dem neusten Stand zu halten. Um all das kümmern sich ein paar internationale Firmen und Heerscharen namenloser Programmierer*innen – hoffentlich…

Es ist befreiend, dass endlich auch für Verschlüsselungstechniken – das Herzstück der IT-Sicherheit – immer einfachere Oberflächen programmiert werden, die die Herstellung von Vertrauen wie Vertraulichkeit auf ein Foto oder den Vergleich zweier Zahlen beschränken. Zugegeben: Sechzig Ziffern sind schon etwas lang. Aber pro Kontakt ist dieses Maß an Aufwand nur einmal erforderlich, zumindest solange niemand sich ein neues Smartphone anschafft, seine Telefonnummer ändert oder solange es nicht zu irgendeiner anderen Störung kommt.

Im Falle von Komplikationen trifft man sich freilich wieder oder man gleicht die Ziffern auf einem anderen, vertrauenswürdigen Kanal ab. Damit ist Verschlüsselung so ›seamless‹ in Anwendungen integriert, dass Computernutzer*innen nicht einmal mehr merken, ob sie nun verschlüsselt und sicher kommunizieren oder nicht. Die Zeiten dieser nervigen Sicherheit sind also endlich vorbei.

Addendum

Im Anschluss an den letzten Satz hat sich noch ein weiterführender Austausch zwischen dem Autor und der Redaktion entsponnen, der hier in Auszügen (und der Lesbarkeit halber leicht bearbeitet) dokumentiert werden soll:

Redaktion 15.5.2018, 9:59 Uhr: Ein schöner Beitrag! Allerdings fragt man sich als unbedarfter Leser vielleicht, der nun auch gerne so eine App hätte, woran man eine gute bzw. sichere App eigentlich erkennt. Ich hatte auch schon mehrere probiert, die spielend leicht irgendwelche Schlüssel erzeugten, um dann mit anderen Leuten sicher kommunizieren zu können, ohne dass so ein aufwendiger Tausch stattfinden musste, den Du in dem Beitrag beschreibst. Ich habe die Apps daher für unseriös gehalten und wieder gelöscht. Allerdings nur aufs Geratewohl und auch nach Deinem Beitrag wüsste ich nicht, nach welchen Kriterien ich mich entscheiden sollte. Die schönen neuen Oberflächen machen zwar alles viel einfacher, aber eben auch schwerer zu durchschauen, jedenfalls für Leute, die keine Programmierer oder entsprechend gebildete Nerds sind. Es könnte ja auch sein, dass man auf betrügerische Apps hereinfällt, die nur vortäuschen, sicher zu sein und eigentlich sind es Spionageprogramme, denen es reicht, nur einmal kurz installiert worden zu sein, um einen kräftigen Schluck Daten zu nehmen, bevor sie wieder gelöscht werden? Wie kann oder soll man mit dem Problem umgehen? Dies wäre meine Anschlussfrage an Deinen Beitrag.

Autor 15.5.2018, 15:42 Uhr: Die Frage am Ende des Textes, mag ich gerne beantworten, indem ich sie auf ›Wie soll man mit dem Problem umgehen?‹ kondensiere und darauf zwei Zitate anschließe, die für sich selbst sprechen:

> *Montag, 14.5.2018:* »Signal-desktop is the standalone desktop version of the secure Signal messenger. This software is vulnerable to remote code execution from a malicious contact, by sending a specially crafted message containing HTML code that is injected

into the chat windows (Cross-site scripting). [...] For safer communications on desktop systems, please consider the use of a safer end-point client like PGP or GnuPG instead.«[3]

Montag, 13.5.2018: »Our advice, which mirrors that of the researchers, is to immediately disable and/or uninstall tools that automatically decrypt PGP-encrypted email. Until the flaws described in the paper are more widely understood and fixed, users should arrange for the use of alternative end-to-end secure channels, such as Signal, and temporarily stop sending and especially reading PGP-encrypted email.«[4]

Die Antwort lautet also: »Gar nicht!« Das ist ja der Punkt: Es gibt keine Chance, nicht einmal für einen Informatiker, alle relevanten Fragen zu klären. Es gibt keine sinnvollen Unterscheidungskriterien, da es keine praktischen Unterscheidungskriterien gibt. Sicherheit nervt, weil sie nicht einzulösen ist. Sie wächst überall in jedem Moment in groteske Komplexität hinein. Daher ist der letzte Satz doppelbödig: Ist die Zeit der Sicherheit vorbei, die genervt hat, so dass nun die nicht-nervende Sicherheit übrigbleibt? Oder ist die Zeit der Sicherheit vorbei, die sowieso nur genervt hat? – Da die meisten Testsubjekte die erste Lesart bevorzugen (da sie ja auch den allgemeinen Parolen der Nerds entspricht) und auf die zweite Lesart trotz aller Ironie anscheinend nicht (allzu schnell) kommen, habe ich mir erlaubt, den letzten Satz durch ein Demonstrativpronomen noch etwas stärker irritierend zu machen – Funktioniert das? Ansonsten könnte man natürlich in der Bevorzugung von Lesart I eine Technikgläubigkeit vermuten.

Zu Deiner Teilfrage zu Apps, die keinen aufwändigen Schlüsselabgleich verlangen, z.B. Whatsapp, Facebook-Messenger: Die versprechen Dir ohne Schlüsselabgleich eine End2End-Verschlüsselung (und Du musst das Versprechen halt glauben), aber sie garantieren Dir nicht, wer am anderen Ende sitzt. Angenommen Du wolltest mit unserer Kollegin P sicher™ kommunizieren, um schlimm über mich zu lästern und da der Weg von einem Raum in den anderen zu weit ist, nutzt Ihr die besagte App aus der Glosse. Nun gehört mir das WLAN und der Handyfunkturm, da ich (einen fiktiven) Systemadministrator und (einen fiktiven) Telekommunikationsprovider bestochen habe. Die App auf Deinem Smartphone will nun (wie in Fußnote 1 beschrieben) ein gemeinsames Geheimnis mit dem Smartphone von P erzeugen. Da ich gewitzt bin, rechne ich damit und fange die Aufforderung zum Schlüsselaustausch ab und gebe mich mit meinem Hacker-Smartphone als P aus. Du und ich haben dann das Geheimnis K, aber Du glaubst, ich wäre P. Da ich ja weiß, dass Du mit P reden wolltest, lasse ich mein Hacker-Smartphone sofort mit Ps Smartphone ein Geheimnis K' aushandeln. Wenn Du nun an P eine Nachricht N schickst, verschlüsselt Dein Smartphone die Nachricht N mit dem Geheimnis K und erhält den Chiffretext C, den Dein Smartphone auf das von mir gehackte Netz schickt. Ich fange C ab. Ich entschlüssele C mit K zu N, verschlüssele N mit K' zu C' und schicke das an P, deren Smartphone C' nun mit K' zu N entschlüsselt und anzeigt, als käme N von Dir, denn mein Hacker-Smartphone hat auch in ihre Richtung einfach gelogen und behauptet, ich sei Du. Während P nun kichert über die fiese Nachricht N, die Du lästerlich über

3 Ivan A. Barrera Oro: ›Signal-desktop HTML tag injection‹, in: *ivan.barreraoro.com.ar*, 14.5.2018, https://ivan.barreraoro.com.ar/signal-desktop-html-tag-injection/advisory/ (aufgerufen 28.8.2018).

4 Danny O'Brien und Gennie Gebhart: ›Attention PGP Users: New Vulnerabilities Require You To Take Action Now‹, in: *eff.org* 13.5.2018, https://www.eff.org/de/deeplinks/2018/05/attention-pgp-users-new-vulnerabilities-require-you-take-action-now, hier ist PGP=GnuPG (aufgerufen 28.8.2018).

mich geschrieben hast, sitze ich auf meinem Funktürmchen und reiße mir angesichts des Zorns die Haare aus, weil ich N gar nicht lustig finde.;-) In der anderen Richtung genauso: P will mit Nachricht N_2 antworten. Ihr Smartphone verschlüsselt N_2 mit K' zu C_2' und sendet C_2', ich fange C_2' ab, entschlüssele zu N_2, verschlüssele sofort N_2 mit K zu C_2 und schicke C_2 zu Dir, bevor ich N_2 lese, in der P mich in Schutz nimmt (hoffentlich).

Diesen Angriff nennt man ›man in the middle‹-Angriff. Man kann ihn verhindern, indem Du…

(a) …verhinderst, dass ich die Leitung hacke, was in der IT-Sicherheit quasi per definitionem nicht bzw. nur durch Verschlüsselung zu verhindern ist. (Will man Verschlüsselung aber gerade erst aufsetzen, hat man ein Henne-Ei-Problem.)

(b) …Sicherheit herstellst, indem Du (wie in der Glosse) Identität (Der Autor dieses Textes ist nicht P) durch einen schwer fälschbaren Kanal in Vertrauen übersetzt, dass Du K auch wirklich mit P und nicht mit mir ausgehandelt hast.

(c) …Vertrauen in eine dritte Stelle in Vertrauen in die Identität der Gegenstelle und dieses in Vertrauen übersetzt, dass Du K auch wirklich mit P und nicht mit mir ausgehandelt hast. (Das ist das, was die ›certificate authorities‹ machen.)

(d) …einen zweiten Kanal zum Abgleich benutzt, der schwer zu manipulieren ist (dann bist Du Dir zwar nicht sicher, aber Du kannst vermuten, dass es mir auch nicht so wichtig ist, dass ich zusätzlich noch Deine Post abfange etc.) (Das ist das mTAN/iTAN-Verfahren beim Online-Banking.)

In der fraglichen App geht das auch: Chat aufrufen → 3-Punkte-Menü → Unterhaltungseinstellungen → Sicherheitsnummer anzeigen → Nummer angezeigt bekommen → abgleichen → wenn es passt, dann das Verifikationshäkchen setzen. Fertig. – Allerdings: Was hat die App getan? Sie hätte Dir das alles genauso gut vorspielen können.