Jürgen Schmidt

Verpfuschte Verschlüsselung

Web- und E-Mail-Server schützen Daten unzureichend

Mit einem kleinen Feature namens Perfect Forward Secrecy könnte man der NSA gewaltig in die Suppe spucken. Doch von den großen US-Konzernen setzt es nur ein einziger ein – und auch in Deutschland ist die Situation unbefriedigend.

Nach heutigem Kenntnisstand überwacht die NSA große Teile der Kommunikation im Internet und archiviert dabei alle verschlüsselten Datenströme. Sie häuft dabei Terabytes an verschlüsselten Daten an. Dies geschieht teilweise in der Hoffnung, sie vielleicht in zehn, zwanzig Jahren mit Quantencomputern und roher Rechengewalt dechiffrieren zu können. Noch wahrscheinlicher und viel einfacher ist jedoch ein anderes Szenario: Die NSA könnte irgendwann an den Schlüssel kommen, mit dem sie die verschlüsselten Daten einfach auspacken und anschließend lesen kann.

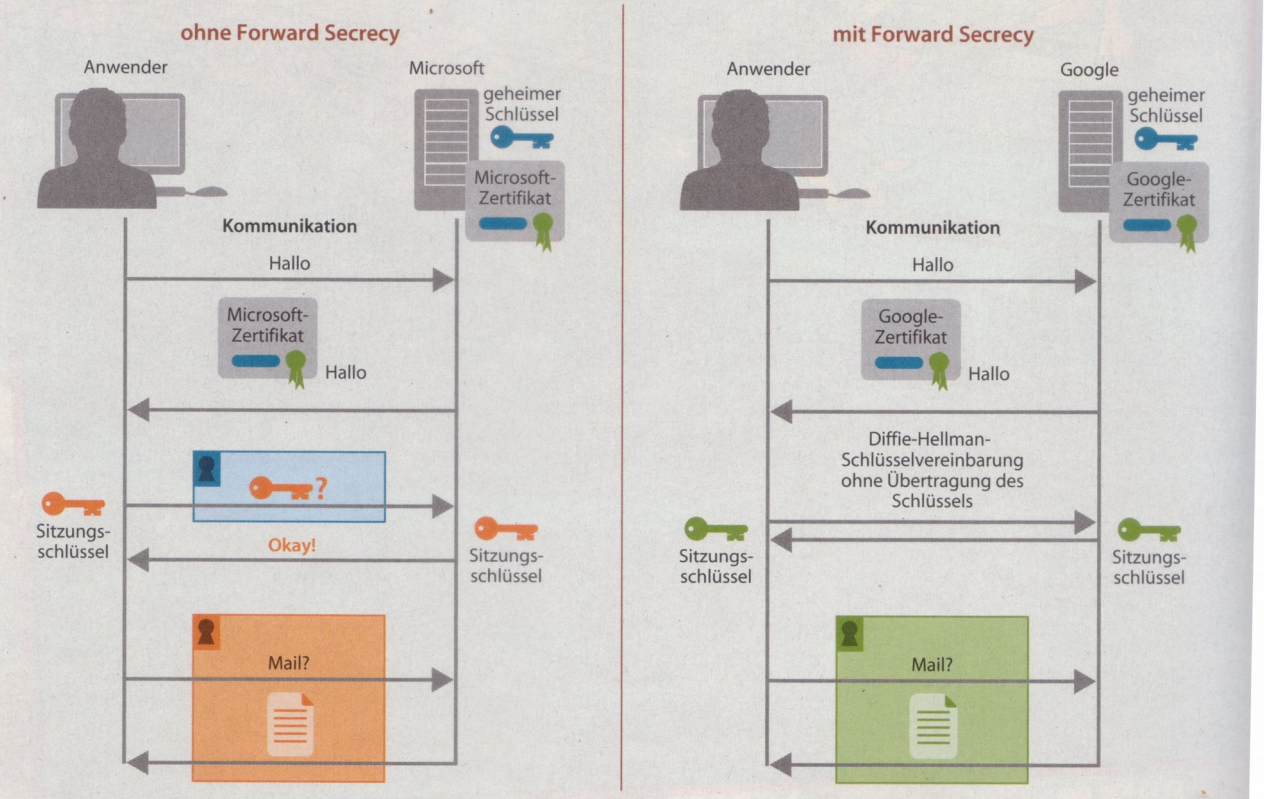
Dass so etwas tatsächlich passieren kann, liegt daran, dass Microsoft, Apple, Facebook, Twitter, Yahoo und eigentlich so gut wie alle US-Firmen bei der Verschlüsselung der Kommunikation schludern. Es gibt seit langem ausgereifte Konzepte, wie man sich vorbeugend gegen das „heute einsammeln, morgen knacken" schützen kann. Wer sich mit Kryptografie auskennt und es mit der Sicherheit der übertragenen Daten ernst nimmt, kennt diese Verfahren und setzt sie ein. Und immerhin ein US-Konzern hat auch bereits vor zwei Jahren ohne viel Aufsehen all seine Server darauf umgestellt. Aber die überwiegende Zahl aller Internet-Dienste verzichtet auf diesen vorbeugenden Schutz.

Schlüsselaustausch

Um das Konzept zu verstehen, muss man ein klein bisschen tiefer in die Funktionsweise der SSL-Verschlüsselung einsteigen, die etwa für sichere Web-Seiten und auch für die Übertragung von E-Mail als Transport-Sicherung genutzt wird. Sie arbeitet grundsätzlich in zwei Stufen (blau und orange/grün in der Grafik). Zunächst kommt ein asymmetrisches Verschlüsselungsverfahren wie RSA zum Einsatz (blau), mit dessen Hilfe sich die Kommunikationspartner ausweisen und dann insbesondere ein gemeinsames Geheimnis – den sogenannten Sitzungsschlüssel – für den zweiten Teil, die eigentliche Datenverschlüsselung, aushandeln. Dort verwendet man dann aus Performance-

SSL-Verschlüsselung

Bei der Schlüsselvereinbarung mit Diffie-Hellman (rechts) geht der geheime Sitzungsschlüssel nicht über die Leitung.



Gründen eine deutlich schnellere symmetrische Verschlüsselung etwa mit AES (orange beziehungsweise grün).

Das lässt sich sehr schön an einer sicheren, verschlüsselten Verbindung zum Web-Mail-Frontend eines Mail-Providers wie Microsofts Hotmail demonstrieren, das neuerdings unter Outlook.com firmiert. Beim Aufbau der Verbindung passiert grob vereinfacht Folgendes:

1. Browser kontaktiert https://mail.live.com.
2. Server präsentiert einen öffentlichen Schlüssel, dem eine vertrauenswürdige Zertifizierungsstelle attestiert hat, dass er tatsächlich Microsoft gehört.
3. Browser überprüft die Unterschrift der Zertifizierungsstelle und ist danach überzeugt, dass er tatsächlich mit Microsoft spricht. Er verschlüsselt seine Nachrichten jetzt mit dem soeben erhaltenen öffentlichen Schlüssel.
4. Server kann die Nachrichten mit dem zugehörigen geheimen Schlüssel entschlüsseln.
5. Browser schlägt eine Zufallszahl wie 19243 als geheimen Sitzungsschlüssel vor.
6. Server bestätigt den geheimen Sitzungsschlüssel.
7. Beide können jetzt Daten ver- und entschlüsseln.

Das Problem dabei ist, dass dabei der geheime Sitzungsschlüssel über die Leitung gegangen ist und wahrscheinlich von der NSA aufgezeichnet wurde. Das geschah zwar verschlüsselt, aber wenn die NSA eines Tages den geheimen Microsoft-Schlüssel in die Hände bekommt, kann sie damit die 19243 aus dem ersten Teil der aufgezeichneten, verschlüsselten Sitzung fischen und dann auch die übertragenen Mails im zweiten Teil lesen. Auf dieselbe Weise kann sie alle Mails entschlüsseln, die der Microsoft-Server in den letzten Jahren verschickt oder empfangen hat.

Hier setzt die Perfect Forward Secrecy (PFS) an, die genau das verhindern soll – dass nämlich eine in der Vergangenheit geführte, bereits abgeschlossene, aber verschlüsselt aufgezeichnete Kommunikation durch nachträgliches Bekanntwerden des geheimen Schlüssels kompromittiert wird. Dazu einigen sich die beiden Kommunikationspartner auf einen nur ihnen bekannten, geheimen Sitzungsschlüssel, ohne dass dieser zwischen ihnen übertragen wird. Das klingt zwar unmöglich, lässt sich aber mit Hilfe eines cleveren Schlüsselaushandlungsverfahrens namens Diffie-Hellman tatsächlich realisieren.

Nach dem Ende der Sitzung zerstören die beiden ihre Kopie dieses Schlüssels, der damit nicht mehr existiert – auch nicht in irgendwelchen verschlüsselten Aufzeichnungen. Ein passiver Lauscher kann also die Sitzungsdaten im Nachhinein auch mit Kenntnis des geheimen, asymmetrischen Schlüssels nicht entschlüsseln. Nur als aktiver Man-in-the-Middle, der die Kommunikation manipuliert und etwa beiden Endpunkten seinen eigenen Sitzungsschlüssel aufzwingt, kann er nach wie vor mitlauschen. Aber zumindest abgeschlossene Sitzungen sind damit perdu.

Handshake Simulation (Experimental)

Chrome 27	TLS 1.0	TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) Forward Secrecy	256
Firefox 21	TLS 1.0	TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) Forward Secrecy	256
Internet Explorer 10	TLS 1.0	TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
Safari iOS 6.0.1	TLS 1.0	TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128
Safari 5.1.9	TLS 1.0	TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	128

Der Test der SSL-Labs zeigt an, mit welchen Browsern ein Server Forward Secrecy aushandeln kann.

Schlüsselaustausch mit Diffie-Hellman und Perfect Forward Secrecy sind keine schwarze Magie, sondern längst gute Praxis in Bereichen, in denen viel Wert auf die Sicherheit der Daten gelegt wird. Secure Shell aka SSH erfordert seit der 1998 vorgestellten Protokollversion 2 einen Schlüsselaustausch mit Diffie-Hellman und Perfect Forward Secrecy (RFC 4253); bei IPSec, mit dem man etwa Firmen mit ihren Filialnetzen verbindet, ist PFS ebenfalls gute Praxis und etwa bei StrongSwan in der Voreinstellung aktiv.

Auch die SSL/TLS-Spezifikation bietet mehrere Schlüsselaustauschverfahren, die auf Diffie-Hellman beruhen und somit PFS bieten: DHE_* und das äquivalente, auf elliptischen Kurven beruhende ECDHE_*. Das abschließende E steht dabei übrigens jeweils für „ephemeral", also flüchtige, vergängliche Schlüssel.

Sowohl Server als auch Browser unterstützen Diffie-Hellman und damit auch Forward Secrecy. Trotzdem kommt es in der Praxis äußerst selten zum Einsatz. Das mag daran liegen, dass die erforderlichen Berechnungen Zeit kosten und den ohnehin langsamen SSL-Handshake noch langsamer machen. Je nach Verfahren dauert der dann im besten Fall 15 bis 30 Prozent (ECDHE) oder sogar 200 bis 300 Prozent (DHE) länger.

Browser-Differenzen

Letztlich entscheidet immer der Server, welches Verschlüsselungsverfahren zum Einsatz kommt. Der Browser kann zwar Präferenzen äußern, aber insbesondere größere Server ignorieren die in der Regel und nehmen stattdessen das aus der Liste des Browsers, was sie für angemessen halten. Theoretisch könnte der Client zwar seine Auswahl an Cipher-Suiten auf die mit Schlüsselaustausch via (EC)DHE beschränken. Aber mal ganz abgesehen davon, dass die Browser das nicht vorsehen, würde das bedeuten, dass mit vielen Servern gar keine Verschlüsselung ausgehandelt werden könnte, was natürlich kontraproduktiv ist.

Trotzdem sind die Präferenzen der Browser, die ein Dienst der Uni Hannover [1] verrät, durchaus aufschlussreich: Chrome, Firefox, Opera und Safari haben Schlüsselaustausch mit (EC)DH ganz oben auf der Liste; Microsofts Internet Explorer 10 kann das zwar auch, zieht jedoch als einziger einfaches RSA vor.

Ziemlich düster sieht es derzeit auf der Server-Seite aus. Bei den in den USA beheimateten großen Internet-Diensten wie Facebook, Twitter, Yahoo, eBay, Paypal und so weiter ist PFS ohnehin Fehlanzeige. Ein bisschen besser schneiden im Vergleich die deutschen Firmen ab. Bei den getesteten Web-Mail-Frontends bieten zumindest GMX

Schlüsselerzeugung nach Diffie-Hellman

Alice und Bob wollen sich auf einen geheimen Schlüssel einigen, ohne dass dieser zwischen ihnen übertragen wird. Als Zutaten brauchen sie dafür eine große Primzahl p sowie eine feste Zahl g, die im Voraus festgelegt werden und öffentlich bekannt sein dürfen. Alice erzeugt eine Zufallszahl x, berechnet $X = g^x \bmod p$ und schickt das Ergebnis an Bob. Bob seinerseits würfelt eine Zufallszahl y, berechnet $Y = g^y \bmod p$ und schickt diese Zahl an Alice. Alice berechnet nun $Y^x \bmod p$, Bob berechnet $X^y \bmod p$. Beide erhalten dabei dasselbe Ergebnis, nämlich $g^{xy} \bmod p$ und haben damit ein gemeinsames Geheimnis.

Ein eventueller Lauscher kennt nur g^x und g^y, kann daraus g^{xy} aber nicht errechnen. Dazu bräuchte er x oder y.

Das Verfahren beruht darauf, dass man wenig Rechenleistung braucht, um eine Potenz $g^x \bmod p$ zu errechnen, das umgekehrte Problem, von g^x auf x zurückzuschließen, aber sehr schwierig ist (diskreter Logarithmus). Mathematiker nennen die Zahlen modulo p zusammen mit der Multiplikation als Rechenoperation eine Gruppe.

Es gibt noch viele andere Arten von Gruppen, und das Diffie-Hellman-Verfahren lässt sich 1:1 auf diese übertragen, solange sie nur die gleiche Grundeigenschaft haben: Potenzieren ist leicht, Logarithmus schwer. Ein prominentes Beispiel für eine geeignete Art von Gruppen sind sogenannte elliptische Kurven, die ebenfalls für die Schlüsselerzeugung nach Diffie-Hellman zum Einsatz kommen. (bo)

und Web.de zukunftssichere Verschlüsselung, alle andern müssen ebenfalls passen.

Wer seinem eigenen Lieblingsdienst auf den Zahn fühlen will, hat zwei Optionen. Als einziger Browser zeigt Chrome in den Eigenschaften einer aktiven, verschlüsselten Verbindung an, wie der Schlüsselaustausch stattgefunden hat. Die Information ist allerdings nur Eingeweihten verständlich. Steht da etwas mit „DHE_" oder „ECDHE_", ist Perfect Forward Secrecy gewährleistet. „RSA" ist ein typisches Beispiel für SSL ohne PFS.

Alternativ kann man die Testergebnisse der SSL Labs von Qualys abrufen [2]. Die zeigen unter „Configuration" die Ergebnisse eines simulierten Verbindungsaufbaus durch verbreiteten Browser und markieren dabei eine erfolgreiche DH-Schlüsselvereinbarung mit „Forward Secrecy". Dabei fällt auf, dass einige Server diese zwar mit Chrome und Firefox aushandeln; nicht aber mit dem Internet Explorer. Viele dieser Server laufen auf Debian Stable mit Apache 2.2 und gerade die finden nicht richtig zusammen. Apache 2.2 kann noch kein ECDHE und Microsofts Browser mag reguläres DHE ausschließlich mit dem bei Apache verpönten DSS. Erst mit Apache 2.4 können sich die beiden auf eine Cipher-Suite mit ECDHE einigen.

E-Mail

Doch SSL bedeutet mehr als nur sichere Web-Seiten. Fast noch wichtiger ist, dass E-Mail sowohl beim Versand (SMTP) als auch beim Empfang (IMAP, selten noch POP) via SSL/TLS gesichert wird und das natürlich auch zukunftssicher geschehen sollte. Wir haben für diesen Artikel die in Deutschland verbreiteten E-Mail-Provider genauer analysiert.

Das Ergebnis war durchwachsen: Komplett durchgefallen sind dabei Microsofts Hotmail/Outlook.com, 1&1 und T-Online. Doch auch bei Arcor, Strato und Web.de erscheint der vereinzelte Einsatz von PFS eher zufällig denn als konsequente Firmenpolitik zugunsten der Privatsphäre der Kunden. Gut schnitt der kleine Mail-Provider Posteo ab, der lediglich beim Web-Mail-Frontend patzt. Und GMX kann man die fehlende Forward Secrecy beim kaum noch verwendeten Abruf der Mails via POP durchaus nachsehen. Der einzige jedoch, der konsequent alle verschlüsselten Verbindungen zukunftssicher mit PFS abwickelt, ist ausgerechnet der von Datenschützern viel gescholtene US-Konzern Google.

Selbst testen

Wer die Sicherheit seiner eigenen Server testen möchte, kann das etwa mit dem unter Linux standardmäßig installierten Kommandozeilen-Tool openssl tun. Der Befehl

```
openssl s_client -connect imap.1und1.de:993
```

liefert unter anderem etwas wie

```
SSL-Session:
    Protocol  : TLSv1
    Cipher    : AES256-SHA
```

was bedeutet, dass keine PFS gewährleistet ist. Nur wenn die Cipher-Suite mit ECDHE oder DHE beginnt, findet Schlüsselaustausch via Diffie-Hellman statt. Bei Diensten, bei denen der Client die Verschlüsselung via SSL mit dem Befehl starttls einschalten muss, liefert

```
openssl s_client -starttls smtp -connect ↗
                                    smtp.gmx.net:587
```

die gewünschte Information (hier ECDHE-RSA-AES256-SHA mit PFS). Interessant ist, dass viele Server PFS durchaus beherrschen, wenn man sie darauf festnagelt:

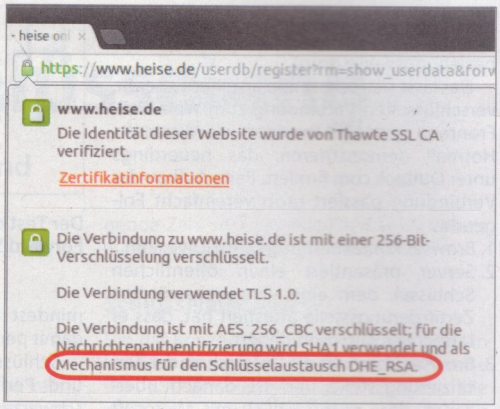
```
openssl s_client -cipher 'ECDH:DH' -connect ↗
                                    login.live.com:443
```

ergibt einen TLS-Handshake mit ECDHE-RSA-AES256-SHA. Leider kann man dieses Festnageln mit dem Browser oder E-Mail-Client praktisch nicht umsetzen.

Fazit

Auch mit dem Einsatz von Perfect Forward Secrecy ist man nicht gegen alle möglichen Spionage-Aktivitäten gefeit. Soll eine konkrete Person ausgespäht werden, sind nach wie vor Angriffe als Man-in-the-Middle möglich. Doch die erfordern einigen Aufwand und skalieren nicht beliebig. Für die routinemäßige Überwachung der E-Mails von Milliarden Menschen ist das keine Option.

Natürlich ist diese Perfect Forward Secrecy eine bislang weitgehend unbekannte Spezialeigenschaft bestimmter kryptografischer Verfahren. Aber in den Sicherheitsabteilungen von Konzernen wie Microsoft und der Telekom gibt es genug Spezialisten, die wissen, was es damit auf sich hat und warum man das haben möchte. Deshalb müssen sich diese Firmen auch jetzt die Frage gefallen lassen, warum sie PFS nicht standardmäßig einsetzen, wenn ihnen Datenschutz und die Privatsphäre ihrer Kunden tatsächlich so am Herzen liegen, wie sie nicht müde werden zu beteuern. Auf ein Gesetz, das sie zum Einsatz minderwertiger Verschlüsselung verpflichtet, können sie sich hier nicht rausreden. So was gibt es nicht einmal in den USA. Und Google macht vor, dass es geht.



Als einziger Browser zeigt Chrome das aktuell verwendete Verfahren zum Schlüsselaustausch an. DHE steht für das erwünschte Diffie-Hellman.

Als Anwender hat man leider keinen Einfluss darauf, ob diese wünschenswerte Eigenschaft zum Einsatz kommt – das entscheidet allein der Administrator des jeweiligen Servers. Doch das ist andererseits auch eine große Chance für alle Server-Betreiber, uns Kunden jetzt zu zeigen, dass ihnen unsere Privatsphäre ein paar zusätzliche CPU-Zyklen wert ist.

Mit einer Einführung von PFS können Microsoft, die Telekom, Strato, 1&1 und alle anderen Anbieter von Internet-Diensten durch ganz konkrete Aktionen beweisen, dass ihre Bekenntnisse zu Datenschutz und Privatsphäre keine hohlen Phrasen sind. Und auf der anderen Seite haben wir als Kunden endlich ein hartes Kriterium an der Hand, an dessen Umsetzung wir die gleich klingenden Sonntagsreden der Service-Provider messen können.

Facebook hat das – nach Google natürlich – als einer der ersten US-Konzerne begriffen und bereits erklärt, man wolle diese Funktion ab Herbst ebenfalls unterstützen. Man darf gespannt sein, ob und wann Microsoft, die Telekom und all die anderen in Bewegung kommen. (ju)

E-Mail-Verschlüsselung der Provider

	Arcor	Google	GMX	Hotmail	Web.de	1&1	Strato	T-Online	Posteo
SMTP	⊖/⊕	⊖/⊕	⊖/⊕	⊕/–	⊖/–	⊕/⊕	⊖/⊕	⊕/⊕	⊖/⊕
POP	⊕/⊕	–/⊕	⊕/⊕	–/⊕	⊕/⊕	⊕/⊕	⊕/⊕	–/⊕[1]	⊖/⊕
IMAP	⊕/⊕	–/⊕	⊖/⊕	–/–	⊕/⊕	⊕/⊕	⊕/⊕	–/⊕	⊖/⊕
Web	⊕	⊖	⊖	⊕	⊕	⊖	⊕	⊕	⊕
Aktivierung jeweils durch starttls/SSL-Port				⊖ = PFS	⊕ = kein PFS	– = kein SSL	[1] nicht empfohlen		
Wertung	⊖	⊕⊕	⊕⊕	⊖⊖	○	⊖⊖	⊖	⊖⊖	⊕

⊕⊕ sehr gut ⊕ gut ○ zufriedenstellend ⊖ schlecht ⊖⊖ sehr schlecht – nicht vorhanden

Literatur

[1] SSL Cipher Suite Details of Your Browser: https://cc.dcsec.uni-hannover.de/
[2] SSL-Test für Server: https://www.ssllabs.com/ssltest/

c't