

Themen

Ende-zu-Ende-Verschlüsselung

1

RSA-Verschlüsselung

2

Nachrichten signieren

3

Hypertext Transfer Protocol Secure (HTTPS)

4

Beispielseiten zur Ansicht – Eigentum von EMS Kraus

Ende-zu-Ende-Verschlüsselung

Mehr als zwei Stunden täglich verbringen Jugendliche und junge Erwachsene mit der Nutzung von Chat- oder Messengerdiensten. Das ergaben Studien zur Mediennutzung von Jugendlichen in Deutschland, die in den „Grunddaten Jugend und Medien 2020“ zusammengefasst sind.¹⁾

Selbstverständlich möchte niemand, dass die mit Freundinnen und Freunden ausgetauschten Nachrichten von Dritten mitgelesen werden. Deshalb haben die Anbieter von E-Mail-, Chat- und Messengerdiensten Verfahren zu Verschlüsselung aller übertragenen Daten etabliert.

Problem der Schlüsselverteilung

Bis in die 1970er Jahre hinein kannte man nur symmetrische Verschlüsselungsverfahren, bei denen Sender und Empfänger einer Nachricht über denselben Schlüssel verfügen müssen.



Das führt im Smartphone-Zeitalter zu der Schwierigkeit, dass für viele geheim kommunizierende Personen auch viele unterschiedliche Schlüssel benötigt werden. Die Anzahl lässt sich aus der Anzahl der Messengerdienst-Nutzer errechnen:

$$\text{Anzahl Schlüssel} = \frac{\text{Nutzer} \times (\text{Nutzer} - 1)}{2}$$

Wollten alle zwei Milliarden WhatsApp-Nutzer miteinander verschlüsselte Nachrichten austauschen, wären dafür zwei Trillionen Schlüssel nötig.





Asymmetrische Verschlüsselung

Das so genannte Schlüsselverteilungsproblem wurde erst 1977 gelöst, als Ronald L. Rivest, Adi Shamir und Leonard M. Adleman am Massachusetts Institute of Technology das nach ihnen benannte RSA-Verfahren entwickelten.

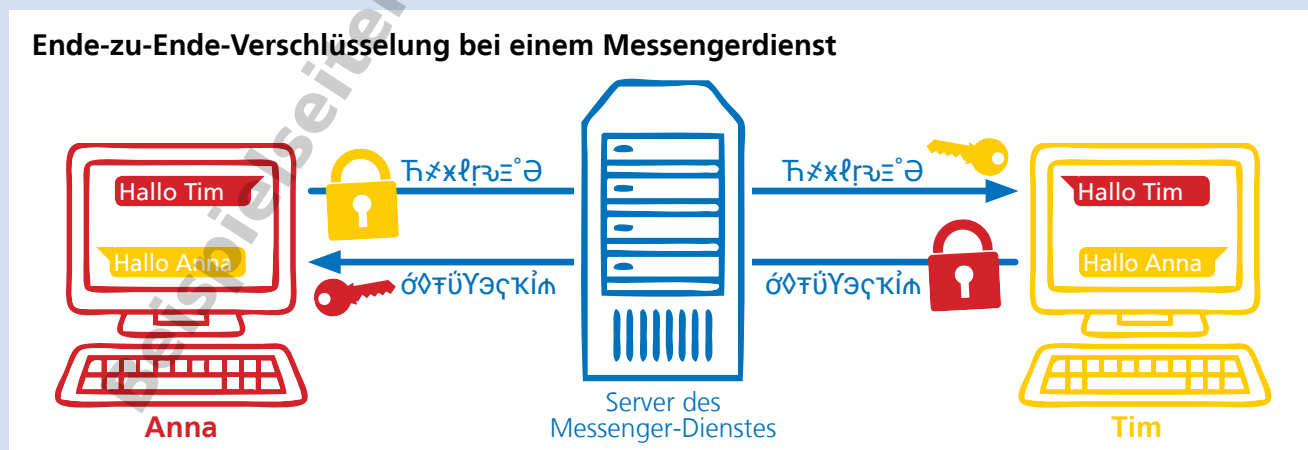
Bei diesem ersten asymmetrischen Verschlüsselungsverfahren verfügt jeder Nutzer über ein eigenes Schlüsselpaar: einen öffentlichen  und einen privaten  Schlüssel.

Asymmetrische Verschlüsselungsverfahren benötigen mehr Rechenleistung für das Ver- und Entschlüsseln der Daten, aber sie sind sicherer als symmetrische Verschlüsselungsverfahren. Sie ermöglichen eine sichere Kommunikation im Internet, ohne immense Kosten und Komplexität zu verursachen.

Mit den öffentlichen Schlüsseln können Daten nur verschlüsselt werden. Die öffentlichen Schlüssel aller Nutzer und Nutzerinnen eines Messengerdienstes sind in einem Verzeichnis abgelegt, auf das alle Nutzer und Nutzerinnen des Messengerdienstes zugreifen können.

Möchte Anna eine Nachricht an Tim senden, verschlüsselt sie die Nachricht mit dem öffentlichen Schlüssel von Tim . Die Nachricht ist auf dem Weg zu Tim von niemandem zu entziffern. Nur der zu Tims öffentlichem Schlüssel passende private Schlüssel  kann die Nachricht entschlüsseln. Die Antwort an Anna verschlüsselt Tim mit Annas öffentlichem Schlüssel . Und nur sie kann mit ihrem privaten Schlüssel  Tims Nachricht entschlüsseln.

Da das Ver- und Entschlüsseln nur beim Sender und Empfänger einer Nachricht möglich ist, spricht man von einer Ende-zu-Ende-Verschlüsselung. Beim Austauschen von Nachrichten bemerkt man davon nichts, denn Messengerdienste wie WhatsApp führen das Ver- und Entschlüsseln vollständig ohne unser Zutun durch.



¹⁾ https://www.br-online.de/jugend/izi/deutsch/Grunddaten_Jugend_Medien.pdf (Stand Juni 2021)

Ende-zu-Ende-Verschlüsselung

Aufgabe 1

Stell dir vor, ihr würdet in eurer Klasse über einen Messengerdienst symmetrisch verschlüsselte Nachrichten austauschen.

Wie viele unterschiedliche symmetrische Schlüssel wären notwendig, damit jede Schülerin und jeder Schüler mit allen anderen Schülerinnen und Schülern deiner Klasse Nachrichten austauschen kann?

Aufgabe 2

Beschreibe den Ablauf der Ende-zu-Ende-Verschlüsselung bei einem Messengerdienst.

Aufgabe 3

Warum kann der Betreiber des Messengerdienstes die auf seinem Server zwischengespeicherten Nachrichten nicht lesen?

Aufgabe 4

Worin besteht der wichtigste Unterschied zwischen symmetrischen und asymmetrischen Verschlüsselungen?

Beispielseiten zur Ansicht – Eigentum von EMS Kraus