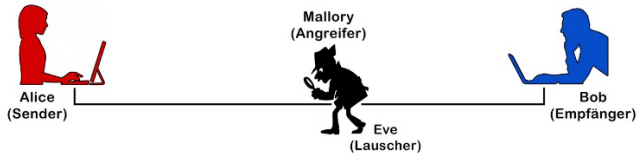
KRYPTOGRAFIE THEORIE

Wir werden uns mit den folgenden Teilbereichen der Kryptografie beschäftigen:

- Symmetrische Verschlüsselung *(Vorzugsweise Klassische Verfahren)*
- Asymmetrische Verschlüsselung
- Digitale Signatur *(Hashwertbildung)*
- Public Key Infrastruktur
- Internet und Zertifikate *(HTTPS, TLS)*
- PGP und OpenPGP *(gpg4win, Kleopatra)*
- Sichere E-Mails *(OpenPGP, S-MIME, Thunderbird)*

Mit **Kryptografie** war ursprünglich die **Wissenschaft der Verschlüsselung** gemeint. Heute umfasst sie allgemein die Informationssicherheit und die Widerstandsfähigkeit gegen Manipulation und unbefugtes Lesen.

- **Kryptologie**: Wissenschaft vom Entwurf, der Anwendung und der Analyse von kryptografischen Verfahren
- **Kryptografie**: Wie kann eine Nachricht ver- und wieder entschlüsselt werden?
- **Kryptoanalyse**: Wie sicher ist ein Verschlüsselungsverfahren?
- **Akteure** in der Literatur:



- **Symmetrische Verschlüsselung**: Symmetrische Verschlüsselungsverfahren zeichnen sich dadurch aus, dass der Absender die Botschaft mit demselben Schlüssel verschlüsselt, wie der Empfänger, der die Botschaft wieder entschlüsselt. Man unterscheidet zwischen historischen und heutzutage unsicheren Verfahren wie ROT etc. und aktuellen sicheren Verfahren wie AES.
- **Asymmetrische Verschlüsselung**: Asymmetrische Verschlüsselungsverfahren zeichnen sich dadurch aus, dass jeder Teilnehmer ein Schlüsselpaar Public/Privat-Key besitzt. Dies erleichtert den Schlüsseltausch und verringert die Schlüsselanzahl.
- **Digital signieren**: Hier geht es nicht darum, einen Inhalt vor fremden Personen zu verbergen, sondern Authentizität, Integrität und Verbindlichkeit einer Nachricht zu gewährleisten. Es kommt dabei dieselbe Technik zur Anwendung, wie bei der asymmetrischen Verschlüsselung.

Geheimhaltung bedeutet:
Nachrichten unter Verschluss

Verschluss bedeutet:
Abgeschlossen mit Schlüssel

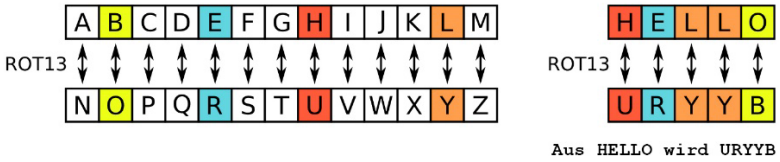
Schlüssel bedeutet:
Muss sicher verschickt werden

SYMMETRISCHE VERSCHLÜSSELUNGSVERFAHREN

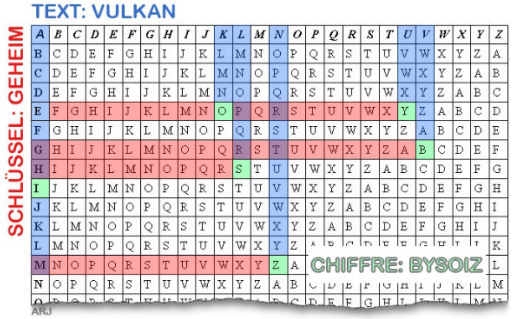
Symmetrische Verschlüsselungsverfahren zeichnen sich dadurch aus, dass der Absender die Botschaft mit **demselben Schlüssel verschlüsselt**, wie der Empfänger, der die Botschaft wieder entschlüsselt.
Wir unterscheiden bei der symmetrischen Verschlüsselung zwischen **klassischen, historischen Verfahren** wie ROT, Vigenere, XOR etc. und **modernen, aktuellen Verfahren** wie AES. Beginnen wir mit den historischen Verfahren:

Schon der römische Feldherr und spätere Kaiser Julius Cäsar kannte einen der folgenden Verschlüsselungstrick und nutzte ihn bei seinen geheimen Botschaften.

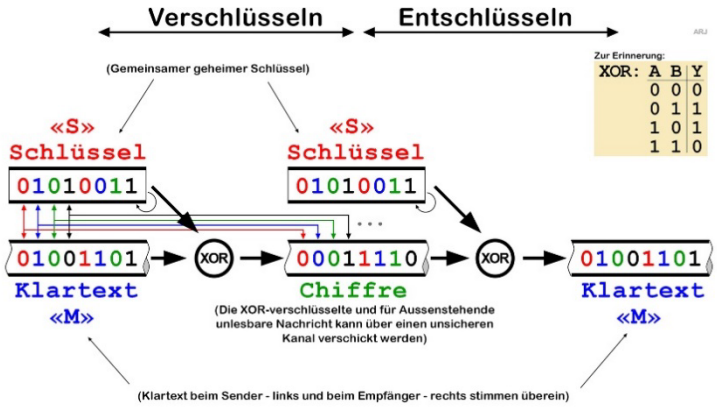
Die Rotationschiffre ROT: Dies ist ein **klassisches, historisches Verfahren**. Bei der Rotationschiffre (Verschiebechiffre) wird jeder Buchstabe des lateinischen Alphabets durch den im Alphabet um eine bestimmte Anzahl Stellen davor bzw. dahinter liegenden Buchstaben ersetzt. Bei der ROT-13 wird z.B. um 13 Buchstaben verschoben:



Die Vigenèreverschlüsselung: Dies ist ebenfalls ein **klassisches, historisches Verfahren**. Bei monoalphabetischen Chiffrierverfahren wie der Rotationschiffre wird ein Buchstabe immer durch denselben Buchstaben ersetzt. Darum sind derart verschlüsselte Geheimtexte schnell nicht mehr geheim, weil sie mit einer einfachen Häufigkeitsanalyse leicht entschlüsselbar sind. Der nun vorgestellte Algorithmus ist ein einfaches polyalphabetisches Chiffrierverfahren. Beim polyalphabetisches Chiffrierverfahren wird ein Buchstabe in der Regel mit verschiedenen Buchstaben chiffriert. Ein einfaches polyalphabetisches Chiffrierverfahren wurde von Blaise de Vigenère (1523-1596), basierend auf den Ideen eines Benediktinermönches, entwickelt. Das nach ihm benannte Vigenère-Verfahren galt lange Zeit als unknackbar:



Die XOR-Stromchiffre: Dies ist ebenfalls ein **klassisches Verfahren**, dass heute eher nicht mehr verwendet wird. Bei der Stromchiffre werden Klartext Bit für Bit bzw. Zeichen für Zeichen XOR-ver- bzw. entschlüsselt. Sender und Empfänger benutzen denselben Schlüssel.



Vorsicht: Sind einem Angreifer Klartext und Chiffre bekannt, kann er den Schlüssel ohne Probleme rekonstruieren.

AES (Advanced Encryption Standard): Im Gegensatz zu den vorangegangenen klassischen Verfahren handelt es sich bei AES um ein **aktuelles, modernes, symmetrisches Verschlüsselungsverfahren**, dass z.B. bei PGP zusammen mit dem asynchron verschlüsselten Schlüsseltauch RSA Verwendung findet. Eine ausführliche Beschreibung findet man z.B. auf Wikipedia oder im Crypttool.

Schlussfolgerung zur symmetrischen Verschlüsselung:

Problematik der symmetrischen Verschlüsselung

ASYMMETRISCHE VERSCHLÜSSELUNGSVERFAHREN

Die Nachteile bei der symmetrischen Verschlüsselung ist der sichere Schlüsseltausch und die Schlüsselanzahl die quadratisch zur Anzahl Teilnehmer wächst. Diese Nachteile sollen mit einem neuen Ansatz behoben werden: Mit der asymmetrischen Verschlüsselungstechnik!

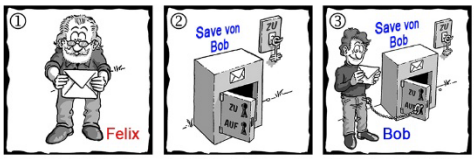
Das Schlüsselkonzept bei der asymmetrischen Verschlüsselung

Der andere Ansatz: Zweiteiliger Schlüssel Man nennt dies asymmetrische Verschlüsselung

Das heisst: Der öffentliche Schlüssel einer Person (Public-Key) kann von allen eingesehen werden

Wobei der private Schlüssel dieser Person (Private-Key) von ihr geheim gehalten wird

Das Verfahren bei der asymmetrischen Verschlüsselung



Eine Botschaft zu übermitteln funktioniert also so: Jederman kann mir eine Botschaft hinterlegen

Damit diese Botschaft auch geheim bleibt, schliesst der Absender den Safe mit meinem öffentlichen Schlüssel zu

Nur ich, mit meinem privaten (geheimen) Schlüssel, kann den Safe öffnen und die Botschaft lesen

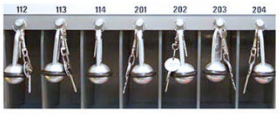
Die Verteilung des öffentlichen Schlüssels bei der asymmetrischen Verschlüsselung

Wo erhalte ich den Public-Key einer Person, die nicht grad um die Ecke wohnt?

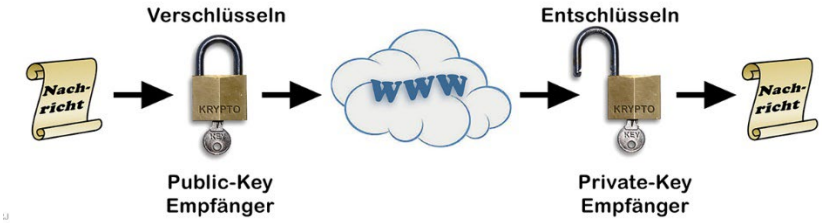
Nicht am Kiosk gegenüber sondern von global vernetzten Keyservern

Wer garantiert mir die Echtheit von auf Key-Servern hinterlegten Public-Keys?
- X.509-CA's
- Web-of-Trust
- PGP-CA's

Schlüsselpaar erzeugen



Public-Key veröffentlichen
Private-Key geheim halten



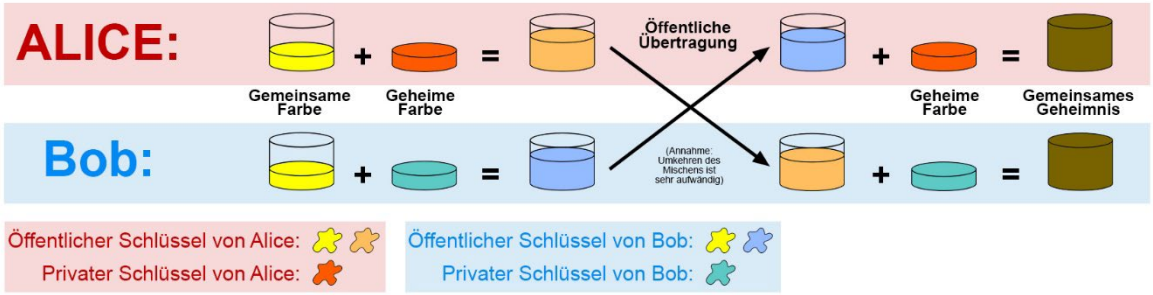
Public-Key Empfänger

Private-Key Empfänger

Wie sicher ist das Verfahren wie z.B. Diffie-Hellman grundsätzlich?

Der Einstieg in die asymmetrische Verschlüsselung erfolgte um das Jahr 1976 durch die Entwicklung eines Protokolls für eine sichere Schlüsselvereinbarung durch die beiden Kryptologen Whitfield Diffie und Martin Hellman. Zwei Kommunikationspartner konnten nun über eine öffentliche, abhörbare Leitung einen gemeinsamen, geheimen und von Drittpersonen nicht berechenbaren Schlüssel in Form einer Zahl vereinbaren. Dieser gemeinsame Schlüssel konnte anschliessend für eine symmetrische Datenverschlüsselung verwendet werden (z.B. DES Data Encryption Standard oder AES Advanced Encryption Standard).

Die Grundidee dieses Diffie-Hellman-Schlüsseltauschs soll nun anhand einer Farb-Analogie bzw. mittels Farbmischung erklärt werden. Das Farbenmischen wird hier als eine Einwegfunktion aufgefasst. Damit meint man, dass es einfach ist, zwei oder mehrere verschiedene Farben zusammenzuschütten. Wesentlich schwieriger ist es allerdings, die erhaltene Farbmischung wieder in ihre ursprünglichen Komponenten aufzuteilen. (Umkehrung)



- Alice und Bob einigen sich öffentlich auf eine gemeinsame Farbe (Gelb).
- Jeder wählt für sich eine eigene, geheime Farbe (Alice: Orange, Bob; Türkis).
- Bob und Alice mischen nun jeweils die gemeinsame Farbe (Gelb) mit ihrer geheimen Farbe (Orange/Türkis). Alice erhält die Farbe Beige und Bob Graublau.
- Diese Farbmischungen tauschen Alice und Bob nun aus. Da darf jeder zuschauen. Diese beiden Farbmischungen sind nämlich nicht geheim. Für einen Aussenstehenden ist es nicht effizient möglich, aus den «öffentlichen» Farben (Gelb, Beige, Graublau) auf die geheimen Farben von Alice und Bob zu schliessen.
- Nun mischen Alice und Bob die Farbmischung ihres Gegenübers mit ihrer eigenen geheimen Farbe. Daraus entsteht wiederum eine neue Farbe (Ockerbraun), die für beide Kommunikationspartner gleich ist (Gelb + Orange + Türkis = Gelb + Türkis + Orange = Ockerbraun). Somit haben Alice und Bob eine gemeinsame geheime Farbe. Einer Drittperson ist es nicht möglich, die geheimen Farben von Alice und Bob herauszufinden, da diese Alices und Bobs geheime Farbzutaten nicht kennt.

Obwohl das mathematische Verfahren sicher ist und eine Brute-Force-Attacke an der Verarbeitungsgeschwindigkeit heutiger Prozessoren scheitert, bleiben zwei Gefahren:

- Unsichere, leicht erratbare Passwörter
- Der unbewusste Einsatz von gefälschten Public-Keys. Damit ist ein Public-Key gemeint, der vorgibt, der Person Bob zu gehören, tatsächlich aber von Mallory kreiert wurde. Die vermeintlich an Bob verschlüsselt verschickte Datei kann nun von Mallory problemlos geöffnet werden. Abhilfe schafft hier eine möglichst zentrale und vertrauenswürdige Public-Key-Verwaltung (PKI) oder ein Vertrauensnetzwerk.

Hybride Verfahren

Symmetrische Verschlüsselungsverfahren haben das bereits besprochene Problem des Schlüsseltauschs und Schlüsselmanagements, das bei den asymmetrischen Verfahren so nicht besteht. Allerdings benötigen symmetrische Verfahren weniger Rechenzeit zur Erstellung des Chiffretexts als rein asymmetrische Verfahren. Darum liegt es auf der Hand, dass in einem hybriden Verfahren die Vorteile der beiden Verfahren genutzt werden:

- **Asymmetrisches** oder Public-Key-Verfahren für **Schlüsselmanagement**, z.B. RSA
- **Symmetrisches** Verfahren zum Versenden der eigentlichen **Nachricht**, z.B. RC4, DES, AES

Digitale Signatur

Nicht immer ist es das Ziel, eine Nachricht zu verschlüsseln. Oftmals besteht auch das Bedürfnis, die Authentizität, Integrität, Verbindlichkeit etc. einer Nachricht zu erfahren:

- Wer hat in meinem E-Shop bestellt?
- wer hat mir diese oder jene E-Mail zugestellt?
- Hat wirklich der Absender das geschrieben hat, was ich da nun lese?
- Woher stammt das Applet, das ich gerade auf meinen PC lade?
- Handelt es sich bei dem Update wirklich um das richtige und unmanipulierte Original?
- Wohin wird meine Kreditkartennummer übermittelt?
- Wer hat bei einer Wahlveranstaltung gerade seine Stimme abgegeben?
- Stammt der Inhalt von www.admin.ch wirklich von unserer Regierung?

Ausgehend von seinem Grundprinzip ist das Internet nicht sicher. Die Gefahren:

- Mitlesen von Daten (Sniffing)
- Vortäuschen falscher Identitäten (Spoofing)
- Angriffe auf die Verfügbarkeit (Denial-of-Service)
- Übertragen von Programmen mit Schadfunktion (Viren, Würmer...)
- Menschliches Fehlverhalten (Preisgabe von geheimen Daten)

Das möchte man erreichen:

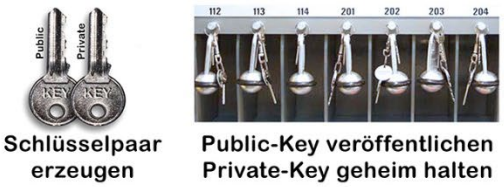
- **Authentisierung** - Sicherstellung der Identität eines Kommunikationspartners
- **Vertraulichkeit** - Zugänglichkeit der Nachricht nur für bestimmten Empfängerkreis
- **Integrität** - Schutz vor Verfälschung von Nachrichten bei der Übermittlung
- **Autorisierung** - Prüfung der Zugriffsberechtigung auf Ressourcen
- **Verfügbarkeit** - Schutz vor Datenverlust, Sicherstellung des laufenden Betriebs
- **Verbindlichkeit** - Sicherer Nachweis der Absendung bzw. des Empfangs

Und da kommt die digitale Signatur ins Spiel, weil diese folgendes bieten kann:

- Sichere **Identifizierung** des Absenders eines Dokumentes
- Sicherheit vor nachträglichen **Manipulationen** des Dokumentes
- Elektronisch signierte Dokumente sind mit unterschriebenen Papierdokumenten gleichgesetzt
- Ist ein Prüfwert einer Information
- Die Digitale Signatur hat die Aufgabe einer Unterschrift und eines Siegels

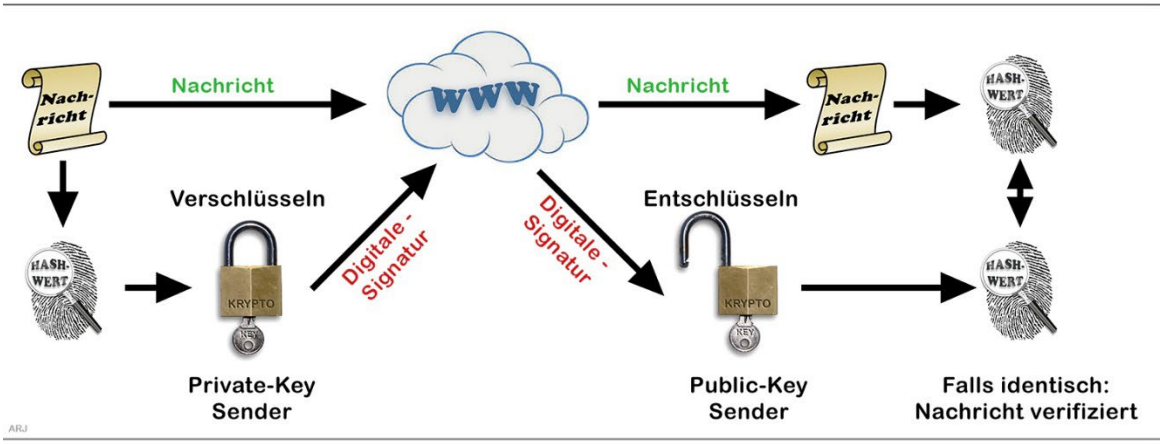
Technisch kommt **dasselbe Verfahren** zur Anwendung, wie bei der **asymmetrischen Verschlüsselung**. Bis auf einen kleinen, aber nicht unwesentlichen Unterschied, nämlich der Verwendung der Schlüssel:

- Der Inhalt der Nachricht wird auf eine eindeutige Kenngrösse abgebildet. Dazu bedient man sich eines **Hash-Algorithmus**
- Der **Hash-Wert** wird mit dem **privaten Schlüssel** verschlüsselt und zur Nachricht hinzugefügt
- Der Empfänger kann mit Hilfe des **öffentlichen Schlüssels** des Senders **prüfen**, ob die Information wirklich vom Absender stammt und nicht verändert wurde



Schlüsselpaar erzeugen

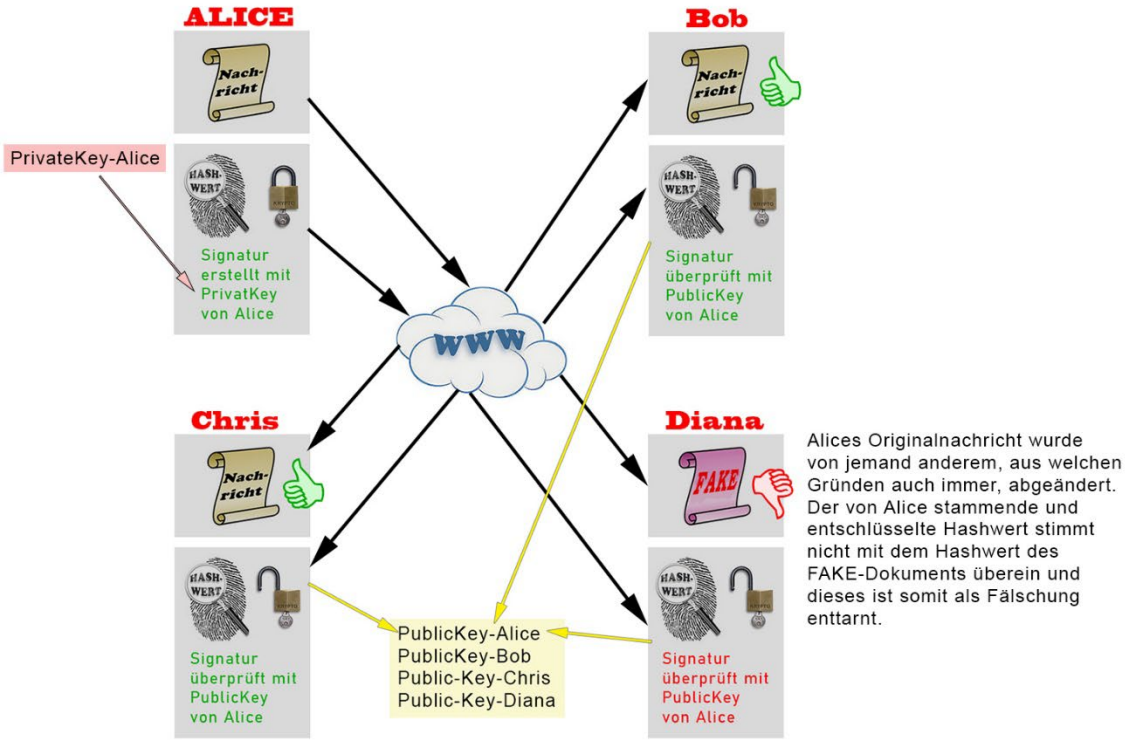
Public-Key veröffentlichen
Private-Key geheim halten



Private-Key Sender

Public-Key Sender

Falls identisch: Nachricht verifiziert



Alices Originalnachricht wurde von jemand anderem, aus welchen Gründen auch immer, abgeändert. Der von Alice stammende und entschlüsselte Hashwert stimmt nicht mit dem Hashwert des FAKE-Dokuments überein und dieses ist somit als Fälschung enttarnt.

Wie wird der Hash-Wert gebildet? (Hash-Algorithmus)

- Der Hashwert ist eine Art **Fingerabdruck** (Fingerprint) eines Dokuments.
- Der Hashwert bildet einen beliebig langen Nachrichtentext auf einen Wert vorgegebener, kurzer Länge an. (Hashwert, Prüfsumme)
- Aus dem Hashwert kann die ursprüngliche Nachricht **nicht errechnet** werden. (Irreversibilität)
- Die Konstruktion von Nachrichten mit identischem Hashwert muss praktisch unmöglich sein.
- Die zufällige Übereinstimmung von Hashwerten beliebiger Nachrichten ist sehr unwahrscheinlich. (Kollisionsresistenz, Integrität prüfbar)

DIE SCHLÜSSELVERWALTUNG

PKI - Public-Key-Infrastruktur

So sicher asymmetrische Verschlüsselungsverfahren auch sein mögen, wirklich geschützt ist man nur, wenn man dem ausgewählten Public-Key auch vertrauen kann. Das heisst, dass z.B. der Public-Key von Bob auch wirklich Bob gehört und nicht ein mit Bob beschrifteter Public-Key von Mallory ist. Um dies sicherzustellen, muss entweder eine Art Vertrauensnetz aufgebaut werden, oder die Public/Private-Keys werden von offiziellen Stellen, ähnlich den Passbüros, ausgestellt. Diese Zertifizierungsstellen klären vor der Schlüsselausgabe die Identität des Antragstellers (mehr oder weniger) seriös ab und garantieren danach (mehr oder weniger) für die Echtheit dieser Schlüssel, wenn jemand diese überprüft haben möchte. Mit Public-Key-Infrastruktur (PKI) bezeichnet man in der Kryptologie ein System, das digitale Zertifikate ausstellen, verteilen und prüfen kann. Die innerhalb einer PKI ausgestellten Zertifikate werden zur Absicherung rechnergestützter Kommunikation verwendet.

Die Bestandteile einer PKI:

- Digitale Zertifikate
- Zertifizierungsstelle (Certificate Authority, CA)
- Registrierungsstelle (Registration Authority, RA)
- Zertifikatsperrliste (Certificate Revocation List, CRL)
- Verzeichnisdienst (Directory Service)
- Validierungsdienst (Validation Authority, VA)
- Dokumentationen: CP (Certificate Policy), CPS (Certification Practice Statement),
- DS (Policy Disclosure Statement)
- Subscriber: Inhaber von Zertifikaten
- Participant: Nutzer von Zertifikaten

Der ITU-T-Standard X.509

Die Alternative zur dezentralen Schlüsselverwaltung (Web-of-Trust), wie wir sie bei OpenPGP vorfinden, ist eine zentrale, hierarchische Verwaltung. X.509 ist dabei ein ITU-T-Standard für eine Public-Key-Infrastruktur zum Erstellen digitaler Zertifikate.

Das X.509-Zertifikat beweist, dass dessen Besitzer seine Identität bzw. Glaubwürdigkeit von einer CA (=Certification Authority) bescheinigen hat lassen. Das heisst: Personenangaben werden mit dem Schlüsselpaar gekoppelt (Elektronischer Ausweis), ausgestellt von einer vertrauenswürdigen Instanz TC (=Trust Center) oder CA. Diese Schlüssel unterscheiden sich von denen bei OpenPGP!

Die Funktionen von TC und CA (X.509):

- **Zertifikate ausstellen** - Korrekte Identifikation des Teilnehmers, Zertifizieren von User, Server, Sub CA's. Erstellung des Zertifikates durch digitale Signatur über Identifikationsdaten und öffentlichen Schlüssel eines Teilnehmers. Bei Ausgabe eines Zertifikates an einen Teilnehmer ist dieser über Funktionalität und Gefahren von zertifizierten Signaturschlüsseln zu unterrichten.
- **Veröffentlichen der Public-Keys** der User und Publizieren eigenes Zertifikat.
- **Verwalten bzw. Archivierung** angelaufener Zertifikate:
  Zertifikate zurückziehen oder für ungültig erklären
  Festlegung der Gültigkeitsdauer von Zertifikaten
  Zeitstempeldienst - Absicherung im Falle des Diebstahls des geheimen Schlüssels. Nachweis einer zeitpunktsbezogenen Willensbekundung. Nachweis und Sicherheit, dass die digitale Signatur nicht nach Ablauf der Gültigkeit des Zertifikates erstellt wurde.

GPG4WIN unterstützt neben persönlichen OpenPGP-Schlüsselpaaren (Dezentral, Web-of-Trust) auch X.509-Schlüsselpaare (Zentrale Zertifizierungsinstanz, CA). Die beiden Schlüsselvarianten sind untereinander aber nicht kompatibel!

SICHERES INTERNET UND ZERTIFIKATE

Sichere Webseiten mit HTTPS → TLS + http

Wie sicher ist eine Webseite? Vor allem dann, wenn sensitive Daten darüber verschickt werden, wie z.B. bei Webshops und Onlinebanking, wo Webformulareingaben wie Benutzernamen und Passwörter vor fremden Augen versteckt werden sollten. Gefürchtet ist auch die "Man-in-the-middle-Attacke, wo eine Person zwischen Webbrowser und Webserver Daten anzapft und modifiziert weiterleitet, bzw. die Antwort abfängt und in seinem Sinne "bereinigt" an den Kunden zurückgibt. Im Weiteren ist es ein grosses Bedürfnis, sicher zu sein, dass herunterladbare Programme/Patches auch echt sind bzw. tatsächlich vom vermeintlichen Absender stammen. Abhilfe schafft da SSL/TLS, ein Verschlüsselungsprotokoll, dass zwischen Transport-Layer und Applikationslayer (siehe ISO-OSI-Schichtenmodell) für Sicherheit sorgt. Möchte man seine Webseite damit aufrüsten, stellt sich sofort wieder die Frage der Authentizität öffentlicher Schlüssel: d.h. es führt kein Weg an einem offiziellen SSL/TLS Zertifikat vorbei. Ausserdem muss der Webhoster SSL/TLS auch unterstützen.

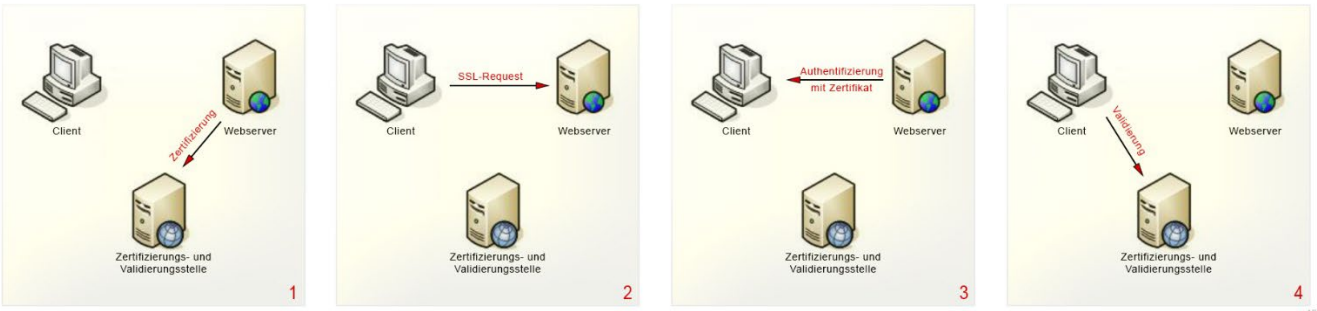
Wie damalige Beiträge im Internet zeigen, werden ab dem Jahre 2018 verschlüsselte Webseiten fast zur Pflicht:

- Ein SSL/TLS-Zertifikat macht deine Seite sicherer und ist gut für Google.
- Schon vor gut zwei Jahren hatte Google angekündigt verschlüsselte Seiten bei den Suchergebnissen bevorzugt zu behandeln.
- Der Google-Browser Chrome brandmarkt unverschlüsselte Seiten mit einem "Nicht sicher" in der Adresszeile.
- Verschlüsselte Kommunikation sollte ein Grundrecht im Internet sein.
- Eine verschlüsselte Webseite ist eben nicht per se vertrauenswürdig.
- Allein mit dem Stichwort Paypal hätten seit Januar 2016 fast 15.000 eindeutig als Phishing-Seiten identifizierte Domains Zertifikate von Let's Encrypt erhalten.
- Etwas über zwei Jahre nach dem Start der gemeinnützigen Zertifizierungsstelle Let's Encrypt (2016) stellt diese nun über die Hälfte aller SSL-Zertifikate im Netz.

SSL/TLS

- **SSL** = Secure Sockets Layer (Veraltet! Wurde durch TLS abgelöst)
- **TLS** = Transport Layer Security (Transportschichtsicherheit)

TLS liegt zwischen der Transportschicht (z.B. TCP via Port 443) und der Applikationsschicht (z.B. HTTP). Man nennt dieses Gespann dann HTTPS. HTTPS ist also kein eigenes Protokoll, sondern die Kombination von TLS mit HTTP. HTTPS ist im Webbrowser bereits integriert, muss also nicht extra nachinstalliert werden. Als Software zum Betrieb eines HTTPS-fähigen Webservers wird eine SSL-Bibliothek wie OpenSSL benötigt. Das digitale Zertifikat für SSL, das die Authentifizierung ermöglicht (Server und die Domain sind eindeutig), ist vom Server bereitzustellen und kann z.B. bei einer Zertifizierungsstelle erworben werden. Der Client-Browser ruft eine sichere Webseite mit https, gefolgt von der URL auf. Ist eine HTTPS-Verbindung etabliert, wird in der Statusleiste des Webbrowsers ein Schloss- oder Schlüssel-Symbol eingeblendet oder die Adresszeile in ihrer Farbe geändert. So erfolgt ein Standard-SSL-Handshake, wenn ein RSA-Algorithmus zum Schlüsseltausch verwendet wird. (Ablauf siehe nächste Seite)



a) Client Hello: Informationen, die der Server benötigt, um mit dem Client mithilfe von SSL zu kommunizieren. Dies beinhaltet die SSL-Versionsnummer, Verschlüsselungseinstellungen und sitzungsspezifische Daten.
b) Server Hello: Informationen, die der Server benötigt, um mit dem Client mithilfe von SSL zu kommunizieren. Dies beinhaltet die SSL-Versionsnummer, Verschlüsselungseinstellungen und sitzungsspezifische Daten.
c) Authentifizierung und Pre-Master Secret: Der Client authentifiziert das Serverzertifikat. (z.B. Common Name/Datum/Aussteller). Der Client (abhängig von der Verschlüsselungssoftware) erstellt den Pre-Master-Secret-Wert für die Sitzung, verschlüsselt ihn mit dem öffentlichen Schlüssel des Servers und sendet den verschlüsselten Pre-Master-Secret-Wert an den Server.
d) Entschlüsselung und Master Secret: Der Server entschlüsselt den Pre-Master-Secret-Wert mithilfe seines privaten Schlüssels. Sowohl Server als auch Client führen Schritte aus, um den Master-Secret-Wert mithilfe des vereinbarten Verschlüsselungsverfahrens zu generieren.
e) Verschlüsselung mit Sitzungsschlüssel: Client und Server tauschen Meldungen aus, um mitzuteilen, dass zukünftige Nachrichten verschlüsselt werden.

Die sichere Verbindung zum Mailserver

Was Sie beim Einrichten eines E-Mail-Clients beachten sollten: Beim Abfragen und Verschicken von E-Mail's sollte man auch an die Verbindungssicherheit denken: Unverschlüsselt können Sie auf dem Transportweg (Mail-Client zu Mail-Server) leicht abgehört werden. Darum ist es von Vorteil, eine Verschlüsselungsmethode wie SSL/TLS einzusetzen, sofern dies von Ihrem Mail-Provider auch angeboten wird. Allerdings muss man wissen, dass es sich dabei "nur" um eine Transportverschlüsselung handelt. Das bedeutet, dass Ihre EMails auf dem Weg zum Mailserver nicht eingesehen werden können. Auf dem Mailserver liegen Ihre EMails allerdings unverschlüsselt vor, ausser Sie bedienen sich einer Mailverschlüsselung. OpenPGP und S-MIME sind dabei die beiden wichtigsten Standards für eine E-Mail-Verschlüsselung.

SSH-Login mit OpenSSH

Secure Shell (SSH) ist eine Möglichkeit, sich sicher bei einem Remote-Computer anzumelden, da alle Daten zwischen dem lokalen Computer und dem Remote-Computer in beide Richtungen verschlüsselt werden. Dazu verwendet man asymmetrische SSH-Schlüssel: Auf dem lokalen Computer wird ein SSH-Schlüsselpaar generiert, der private Schlüssel verbleibt auf dem lokalen PC und der öffentliche Schlüssel wird auf den entfernten Computer hochgeladen.
OpenSSH ist eine Implementierung der ssh-Protokoll-Suite. Der Quellcode ist offen und wird ständig vom OpenSSH-Projekt-Team weiterentwickelt. OpenSSH ist für sämtliche Betriebssysteme compiliert worden und im Lieferumfang der meisten Linux- oder BSD-Distributionen enthalten.

Die OpenSSH-Distribution enthält folgende Programme:

- **ssh**: Ist der ssh-Client. Er baut die Verbindung zu einem ssh-Server auf. Er wird in den folgenden Kapiteln detailliert beschrieben.
- **sshd**: Ist der ssh-Server. Er nimmt die Verbindung von ssh-Clients auf.
- **ssh-keygen**: Erstellt und konvertiert Schlüssel.
- **ssh-agent, ssh-add**: Hält den entschlüsselten Privatekey im Arbeitsspeicher, wo er von ssh-clients angesprochen werden kann.
- **ssh-keyscan**: Zeigt den Hostkey eines ssh-Servers an.

Die Arbeitsschritte (Lokaler LIN-PC=Linux-PC und Remote-WIN-PC):

- Auf dem LIN-PC existiert z.B. der Benutzer HansMuster
  (Falls ihr LIN-PC kein CH-Layout aufweist, dies entweder ändern oder bedenken, dass y und z vertauscht sind bzw. der Bindestrich sich mit der ?-Taste aufrufen lässt und der Slash / sich auf der Taste _ befindet)

- IP-Adresse des LIN-PCs prüfen und gegebenenfalls anpassen:
  `sudo apt install net-tools`
  (NetTools installieren, sofern noch nicht geschehen.)
  `ifconfig` (Aktuelle IP-Adresse überprüfen)
  `sudo ifconfig eth0 192.168.1.111 netmask 255.255.255.0`
  (IP gem. Vorgabe anpassen)
  `ifconfig` (Einstellung überprüfen)

- Verbindung lokaler LIN-PC und Remote-WIN-PC mit ping überprüfen.

- Auf dem LIN-PC ssh installieren:
  `sudo apt install openssh-server`
- Auf dem LIN-PC ssh-Installation überprüfen:
  `sudo systemctl status ssh` (Sollte active – running - als Antwort enthalten)
  `sudo systemctl enable ssh` (Falls ssh noch nicht aktiv ist)
  `sudo systemctl start ssh` (Falls ssh noch nicht aktiv ist)

- Auf dem LIN-PC das .ssh-Verzeichnis im eigenen Homeverzeichnis auf bereits existierende SSH-Keys untersuchen:
  `ls -la ~/.ssh/`

- Auf dem LIN-PC neues Schlüsselpaar erstellen:
  `ssh-keygen -C mein_SSH-Keyname`
  (mein_SSH-Keyname durch meinen bevorzugten Schlüsselnamen ersetzen)
  (Hinweis: Mit der Option -t kann man den zu erstellenden Schlüsseltyp angeben. Z.B. **`ssh-keygen -t ed25519 -C mein_SSH-Keyname`**. Mögliche Schlüsseltypen sind **`dsa, ecdsa, ecdsa-sk, ed25519, ed25519-sk, rsa`**. Weitere Optionen entnehmen man den LIN-Manual-Pages.)

- Jetzt sollte auf dem LIN-PC mindesten ein Schlüsselpaar existieren:
  `ls -la ~/.ssh/`
  ergibt mein_SSH-Keyname als PrivateKey und mein_SSH-Keyname.pub als PublicKey.

- Den soeben erstellten öffentlichen Schlüssel auf den WIN-PC kopieren.

  Auf dem WIN-PC ein Terminal öffnen und Remote-Verbindung etablieren:
  `ssh HansMuster@192.168.1.111`
  (HansMuster ist ein LIN-PC-User. Die IP-Adresse entsprechend anpassen)
  `ssh HansMuster@192.168.1.111 -p Port-Nummer`
  (Falls nicht Standardport 22 verwendet wird)
  Anstatt der IP-Adresse kann auch ein Domänenname verwendet werden, sofern dieser über z.B. DNS aufgelöst wird.

- Nun erscheint das Linux-Shell-Prompt des LIN-PCs.

TLS-Zertifikate

Zertifikate ermöglichen es zusammen mit der Public Key Infrastruktur (PKI), Informationen im Internet sicher und verschlüsselt zu übertragen.
Ein digitales Zertifikat bestätigt bestimmte Eigenschaften von Personen oder Objekten. Dessen Authentizität und Integrität kann durch kryptografische Verfahren geprüft werden.

Folgende TLS Zertifikatsarten werden unterschieden:

- Domain Validated
- Organization Validated
- Extended Validation

Eine Auswahl an Zertifizierungsstellen:

- **Letsencrypt**: Bietet seit 2015 kostenlose TLS-Zertifikate an.
  So lange ein Domain validiertes Zertifikat ausreicht ist Let's Encrypt auf jeden Fall zu empfehlen. Soll aber ein organisationsvalidiertes oder erweitert validiertes Zertifikat genutzt werden, um zum Beispiel die Existenz der jeweiligen Firma ersichtlich zu machen, muss auf einen anderen Anbieter zurückgegriffen werden.
  Link zum Anbieter: https://letsencrypt.org/de/
- **CaCert**: Ist eine der älteren gemeinschaftsbetriebenen, nichtkommerziellen Zertifizierungsstellen. CAcert stellt für jedermann kostenfrei X.509-Zertifikate für verschiedene Einsatzzwecke aus und soll eine Alternative zu den kommerziellen Zertifizierungsstellen sein, die zum Teil recht hohe Gebühren für ihre Zertifikate erheben.
  Der Webbrowsersupport ist limitiert, was einer Verbreitung von CaCert bisher eher hinderlich war.
  Link zum Anbieter: http://www.cacert.org/
- **Verysign**: Ist ein kommerzieller Anbieter aus den USA.
  Link zum Anbieter: https://www.verisign.com/
- **Globalsign**: Ist ein kommerzieller Anbieter aus Japan.
  Link zum Anbieter: https://www.globalsign.com/en
- **DigiCert/Quovadisglobal**: Ist ein kommerzieller Anbieter aus der Schweiz.
  Link zum Anbieter: https://www.quovadisglobal.com/ch/
- **Swisssign**: Ist ein kommerzieller Anbieter aus der Schweiz.
  Link zum Anbieter: https://www.swisssign.com/

PGP und OpenPGP

Pretty Good Privacy: PGP und OpenPGP

PGP ist ein von Phil Zimmermann entwickeltes Programm zur Verschlüsselung und Signieren von Daten und arbeitet auf hybrider Basis. Es benutzt ein PrivateKey/PublicKey-Verfahren wie bei RSA. PGP basiert dabei auf dem sogenannten Web-of-Trust, bei dem es keine zentrale Zertifizierungsstelle gibt, sondern ein Vertrauensnetzwerk, das von den Benutzern selbst verwaltet wird. Mit PGP kann man wahlweise eine Nachricht signieren, verschlüsseln oder signieren und verschlüsseln.

Um von IT-Multis unabhängig zu bleiben, wurde 1998 der OpenPGP-Standard entwickelt. Das unter der GNU-GPL stehende Programm GnuPG bzw. GPG4WIN ist wiederum eine Implementierung von OpenPGP. GPG4WIN unterstützt neben persönlichen OpenPGP-Schlüsselpaaren (Dezentral, Web-of-Trust) auch X.509-Schlüsselpaare (Zentrale Zertifizierungsinstanz, CA). Die beiden Schlüsselvarianten sind untereinander aber nicht kompatibel!

OpenPGP: Sowohl Private/PublicKey als auch verschlüsselte Nachricht sind **ASCII-Dateien** (.asc)

- Beispiel eines **öffentlichen** OpenPGP-Schlüssels (Auszugsweise):

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2
mHjzBFYjc.
...
-----END PGP PUBLIC KEY BLOCK-----
```

- Beispiel eines **privaten** OpenPGP-Schlüssels (Auszugsweise):

```
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: : GnuPG v2
wqdsf
...
-----END PGP PRIVATE KEY BLOCK-----
```

Sichere E-Mails

Sichere E-Mails mit S/MIME

S/MIME = **S**ecure **M**ultipurpose **I**nternet **M**ail **E**xtensions
Dies ist ein Standard für die Verschlüsselung und das Signieren von MIME-gekapselter E-Mail durch ein hybrides Kryptosystem (Asymmetrisch/Symmetrisch). S/MIME wird von den meisten modernen Mailclients unterstützt. Es erfordert X.509-basierte Zertifikate für den Betrieb.

OpenPGP ist eine Schlüsselvariante und Alternative zu S/MIME.

S/MIME-Zertifikate:

- **X.509** ist ein **Standard** für eine **Public-Key-Infrastruktur** (PKI) zum Erstellen digitaler Zertifikate.
- Mit **Public-Key-Infrastruktur** (PKI) bezeichnet man in der Kryptologie ein System, das digitale Zertifikate ausstellen, verteilen und prüfen kann. Die innerhalb einer PKI ausgestellten Zertifikate werden zur Absicherung rechnergestützter Kommunikation verwendet.
- Die Anbieter von kostenpflichtigen S/MIME-Zertifikaten für sichere E-Mail-Kommunikation klassifizieren diese meist in drei Klassen. Dabei sichert bei Klasse 1 die Zertifizierungsstelle (CA) die Echtheit der E-Mail-Adresse zu und nur diese ist Teil des Zertifikats. Bei Klasse 2 wird zusätzlich der zur E-Mail-Adresse gehörende Name in das Zertifikat mit aufgenommen, sowie die Organisation/Firma. Diese Daten werden mithilfe von Drittdatenbanken und Ausweiskopien verifiziert. Bei Zertifikaten der Klasse 3 muss der Antragsteller sich persönlich ausweisen.
- Manche Unternehmen und nichtkommerzielle Organisationen bieten kostenlose S/MIME-Zertifikate an. Es können dabei nach einer Registrierung mehrere Zertifikate erstellt werden, die aber erst nach einer gewissen Anzahl von Identitätsnachweisen den Namen beinhalten. Diese können durch Mitglieder in einem Web-of-Trust oder anderen vertrauenswürdigen Stellen wie Rechtsanwälten oder Wirtschaftstreuhändern erfolgen. Grundsätzlich zerstört eine Ausstellung von Zertifikaten durch Anbieter ohne Identitätsüberprüfung des Antragstellers den Grundgedanken des sicheren Informationsaustauschs zwischen zwei Computern im Netz. So ist es bei einigen Zertifikatsausstellern tatsächlich möglich, mit erfundenen Betreiberangaben ein Zertifikat für eine völlig fremde Website zu erhalten. Der Benutzer würde über ein Umleiten der Informationsübertragung beispielsweise durch den Browser nicht mehr informiert, wenn die Zertifizierungsstelle durch den Browserhersteller von vornherein als vertrauenswürdig eingestuft wurde.

OpenPGP oder S/MIME?

Für die Verschlüsselung von Mails und deren Anhängen stehen grundsätzlich zwei verschiende Verschlüsselungs-Varianten zur Auswahl:

- OpenPGP-Standard (z.B. GnuPG)
- S/MIME

Obwohl beide Verfahren grundsätzlich auf Basis der gleichen Verschlüsselungsverfahren arbeiten, sind die Verfahren in der Umsetzung untereinander inkompatibel. Eine S/MIME-verschlüsselte Mail kann also nicht mit GnuPG entschlüsselt werden und umgekehrt. Gleiches gilt für die digitalen Unterschriften die mit beiden Verfahren erstellt werden. Damit stehen Anwender vor der Qual der Wahl:

1. Verwendung von OpenPGP-kompatibler Software - meist GnuPG
2. Verwendung von S/MIME
3. Parallele Verwendung beider Varianten - je nach Absender oder Empfänger wird das eine oder das andere Verfahren verwendet
4. Nachfolgend einige Argumente zu den beiden Verfahren.

GnuPGP

GnuPGP ist kein Standard in Mailprogrammen – muss deshalb fast immer über Add-Ons / Plugins nachgerüstet werden. Diese Add-Ons / Plugins sind nur für eine eng überschaubare Anzahl an Plattformen bzw. Programmen verfügbar. Gerade für Smartphones oder Tablets somit nicht oder nur sehr schwer verwendbar.

- Z.B. mit Mailvelope ist auch für Browser, und damit für diverse Webmail-Oberflächen, die Anwendung von OpenPGP kompatibler Verschlüsselung möglich.
- Benötigt im Allgemeinen zusätzlich GnuPG-Software für die eigentliche Verschlüsselung.
- Insbesondere bei der Verschlüsselung von Dateianhängen eher anfällig für Konfigurations- und Bedienungsfehler.

Vertrauensbasis der verwendeten Schlüssel:

- Gegenseitige Absprachen / Beglaubigungen der Anwender untereinander, das sogenannte "Web of Trust"
- Überprüfung von Beglaubigungen im Vergleich zu S/MIME aufwändig
- Ausnahme sind Beglaubigungen durch das de facto GnuPG-Trustcenter CAcert

S/MIME

Standard in fast allen Mailprogrammen. Somit meist schon "out-of-the-box" vorhanden - gerade auch bei den meisten Mail-Apps für Smartphones und Tablets. Somit ist keine Installation weiterer Verschlüsselungs-Software nötig. S/MIME-Unterstützung ebenfalls in diverse Webmail-Oberflächen integriert

Vertrauensbasis der verwendeten Schlüssel:

- Beglaubigung / Zertifikat durch „digitales Notariat“ (Fachsprache: Certification Authority – CA bzw. Trust-Center)
- Beglaubigt / zertifiziert wird im Allgemeinen der Besitz der Mailadresse
- Nur wenige CAs stellen die Beglaubigung kostenlos aus
- Zertifikate maximal 3 Jahre gültig - kostenfreie Zertifikate im Allgemeinen nur 1 Jahr gültig.
- Überprüfung der Beglaubigung vollautomatisch mit im Mailprogramm hinterlegten Notariats-Schlüssel
- Schlüsselaustausch zwischen Kommunikationspartnern im Vergleich zu GnuPG erheblich einfacher

Fazit GnuPG vs. S/MIME

S/MIME wird auf mehr Plattformen und durch mehr Mailprogramme direkt, ohne weitere Zusatzsoftware, unterstützt. Auf mobilen Geräten wie Smartphones und Tablets ist damit eine Mailverschlüsselung leichter realisierbar.

- Fehlkonfigurationen und/oder Bedienungsfehler führen bei GnuPG auch bei Profis leichter zu unverschlüsselten Mailanhängen - mit ggf. fatalen Folgen!
- Der Schlüsselaustausch ist bei S/MIME erheblich einfacher.
- Die kurze Gültigkeitsdauer von kostenlosen S/MIME-Zertifikaten (1 Jahr) bedingt allerdings die zyklische Neuerstellung dieser Zertifikate. Längere Gültigkeitsintervalle der Zertifikate - z.B. drei Jahre - sind nur über Bezahlung verfügbar. Auch beim OpenPGP-Standard sollten, obwohl möglich, keine Schlüsselpaare ohne Ablaufdatum erstellt werden.
- In der praktischen Anwendung mit verschiedenen Benutzergruppen hat sich S/MIME als einfacher erwiesen.

Quelle: https://lehrerfortbildung-bw.de/st_digital/medienwerkstatt/dossiers/sicherheit/email/openpgp-vs-smime/