Nachrichten-Verschlüsselung

Mit S/MIME

Höma, watt is S/MIME?

- *S/MIME* ist eine Methode zum signieren und verschlüsseln von Nachrichten, ähnlich wie das in der Öffentlichkeit vielleicht bekanntere *PGP* oder seine quelloffene Implementation *GPG*
- Im Gegensatz zu GPG bietet S/MIME einige Komfortvorteile

S/MIME Vorteile

- S/MIME wird standardmäßig von allen (gängigen) email-Programmen unterstützt
- Schlüssel werden automatisch durch das Versenden signierter Nachrichten verteilt
- Man braucht (kann aber) nicht jedem Schlüssel sagen, dass man ihm vertraut. Vertraut man dem Stammzertifikat, vertraut man allen Schlüsseln, die dagegen gezeichnet wurden

S/MIME Nachteile

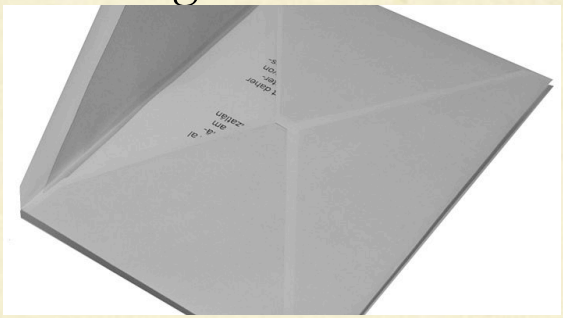
- Das System besteht auf Vertrauen gegenüber den Zertifikatsstellen (Certification Authorities = CAs), bzw. deren Stammzertifikaten

Digitale Signatur

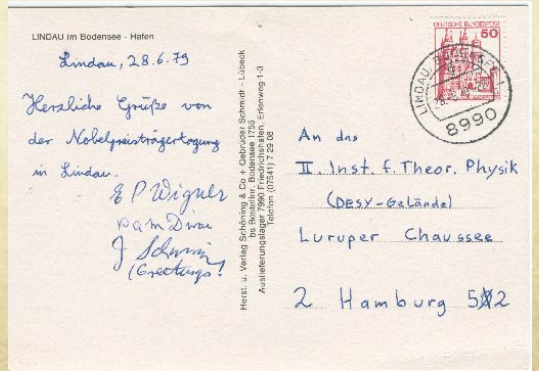
- Eine Signatur stellt sicher, dass der Absender der ist, der er vorgibt zu sein und dass die Nachricht unterwegs nicht verändert wurde
- Sie entspricht also in etwa einer Unterschrift / einem Siegel auf einem konventionellen Brief

Verschlüsselung

- Verschlüsselung stellt sicher, dass nur designierte Empfänger die Nachricht lesen kann. Sie entspricht also grob dem Umschlag eines konventionellen Briefes



- Eine unverschlüsselte Nachricht entspricht also in etwa einer Postkarte: Jeder unterwegs kann sie lesen und auch drin rum kritzeln



Zertifikate

- Das Unterschreiben und Verschlüsseln erfolgt durch sog. Zertifikate
- Dabei gibt es verschiedene Typen Man kann nämlich nicht nur Strompost verschlüsseln, sondern auch Quelltext unterschreiben und der Gleichen
- Zertifikate bekommt man von Zertifizierungsstellen ausgestellt (CAcert, Verisign, TrustCenter, Telekom, DFN...)

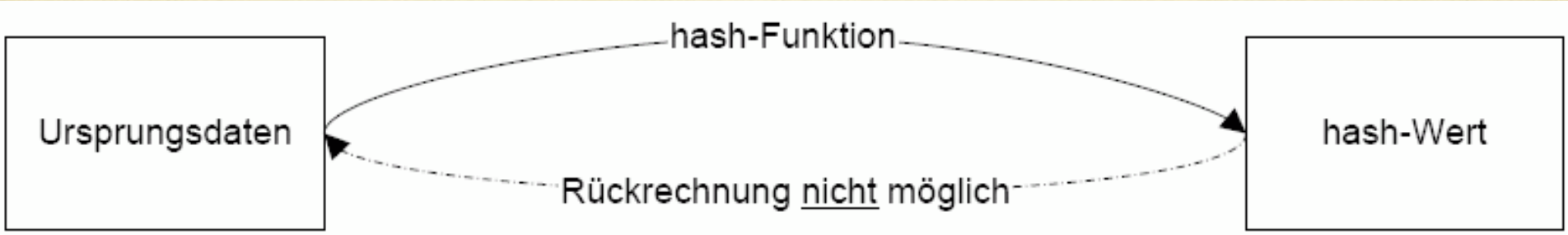
DEMO

(Zertifikate in Firefox)

Krypto-Grundlagen

Hash

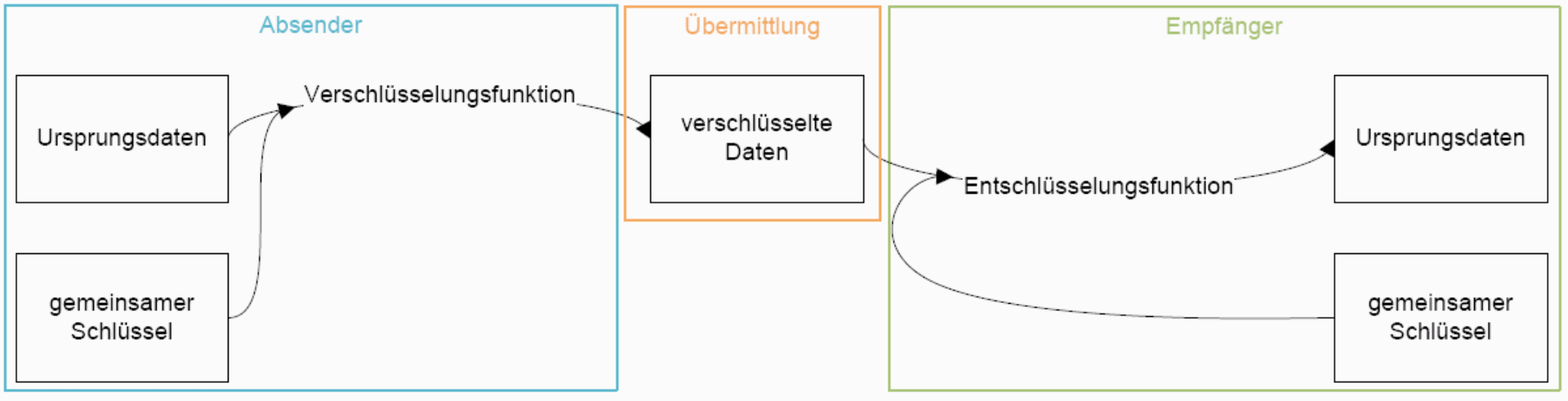
- Ein hash ist eine Funktion ähnlich einer Prüfsumme, welche unabhängig der Eingabe einen Wert mit konstanter Zeichenzahl liefert und zusätzlich die Eigenschaft besitzt nicht umkehrbar zu sein
- Es ist nicht möglich aus dem Ergebnis der hash-Funktion den Eingabewert zurückzurechnen
- Es ist extrem unwahrscheinlich, dass zwei unterschiedliche Eingaben zum gleichen hash führen
- Außerdem, führt schon das Ändern eines einzelnen Zeichens der Quelle zu einem drastisch anderem hash



DEMO

(Beispiel hash-Funktionen: sha, md5)

Symmetrische Verschlüsselung



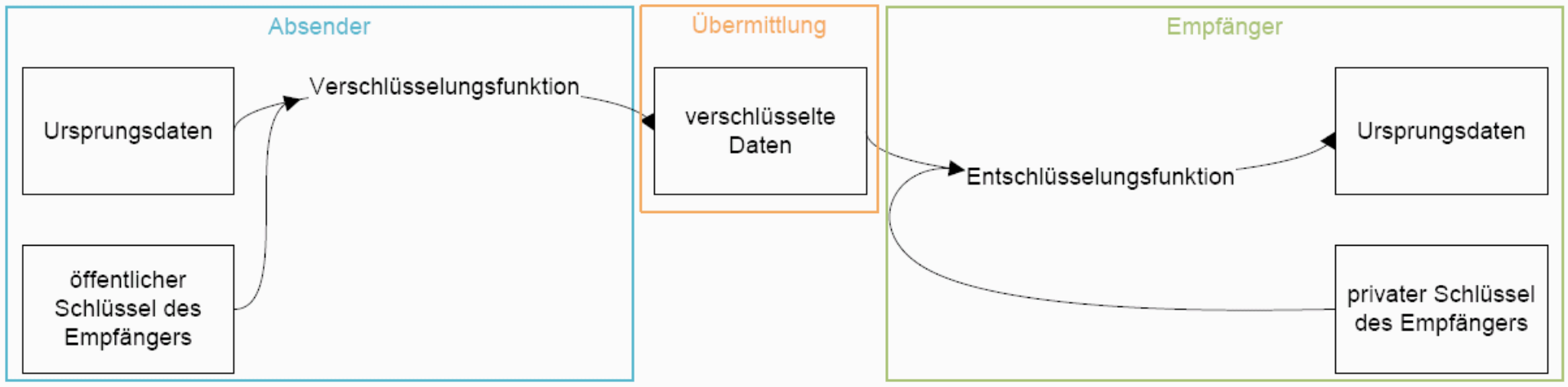
Beispiel Verfahren:

- 3DES (gilt als geknackt)
- AES
- blowfish

Beispiel Anwendungen:

- Kennwort geschützte Dateien
- Sitzungschlüssel

Asym. Verschlüsselung



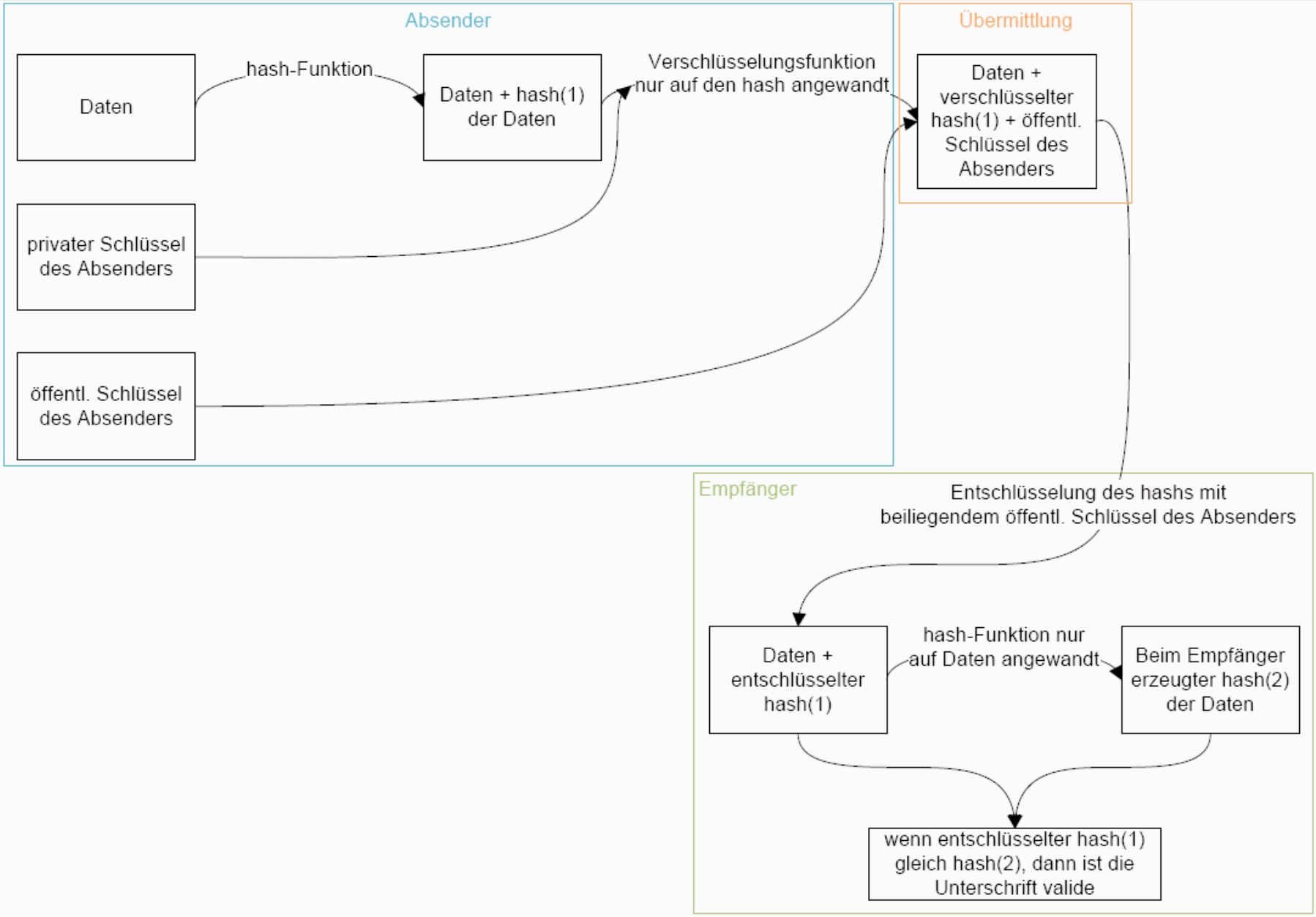
Beispiele für asymmetrische Verschlüsselungsverfahren:

- DSA (gilt als geknackt)
- RSA

Beispiel Anwendungen:

- Strompost (in Kombination)
- SSH (in Kombination)

Digitale Unterschrift



Nachrichten-Verschlüsselung

- Hier wird symmetrische & asymmetrischer Verschlüsselung kombiniert
- Der Inhalt der Nachricht wird aus Geschwindigkeitsgründen mit einem für diese Nachricht speziell erzeugten Schlüssel (Sitzungsschlüssel) symmetrisch Verschlüsselt
- Anschließend wird dieser Schlüssel (welcher idR deutlich kürzer als die eigentlich Nachricht/Nutzlast sein wird) asymmetrisch verschlüsselt und mit der symmetrisch verschlüsselten Nachricht versandt

Nachrichten-Verschlüsselung

- Man kann den Sitzungsschlüssel auch gegen mehrere öffentliche Schlüssel verschlüsseln, anhängen und dann mit den zugehörigen privaten Schlüsseln wieder entschlüsseln
- Dies erlaubt dann beispielsweise, dass man Nachrichten, die man verschickt hat auch selbst noch einmal wieder lesen kann (denn man hat ja nur seinen eigenen privaten Schlüssel, nicht aber den des Empfängers), oder dass man für mehrere Empfänger verschlüsseln kann

DEMO

(Nachrichten signieren & verschlüsseln)

Übung

(persönliches Zertifikat erzeugen)