

E-Mail-Verschlüsselung

Prof. Dr. rer. nat. Peter Gerwinski

30. Oktober 2017

E-Mail-Verschlüsselung

<https://gitlab.cvh-server.de/pgerwinski/hp.git>

1 Einführung

1.1 Warum verschlüsseln?

1.2 Einfache Verschlüsselungsalgorithmen

2 Ihr Verschlüsselungs-„Führerschein“

2.1 Geheime und öffentliche Schlüssel

2.2 Digitale Signaturen

2.3 Zertifikate – Web of Trust

3 Verschlüsselung in der Praxis

4 Fazit

1 Einführung

1.1 Warum verschlüsseln?

„Ich habe nichts zu verbergen.“

- Vor wem? Wirklich vor jedem?
Wirklich alles über Sie?
- Auch nicht vor dem organisierten Verbrechen?
- Illegaler Zugriff auf staatlich gesammelte Daten ist möglich!

—→ Der einzige wirklich sichere Weg,
Datensammlungen gegen Mißbrauch zu schützen,
ist, die Daten gar nicht erst zu sammeln.

1 Einführung

1.2 Einfache Verschlüsselungsalgorithmen

- Cäsar-Chiffre: Alphabet rotieren
Beispiel: ROT13 (mit Schlüssel 13)
25 verschiedene Schlüssel \longrightarrow Schlüssellänge ≈ 4.64 Bit ($2^{4.64} \approx 25$)
leicht zu knacken

1 Einführung

1.2 Einfache Verschlüsselungsalgorithmen

- Cäsar-Chiffre: Alphabet rotieren
Beispiel: ROT13 (mit Schlüssel 13)
25 verschiedene Schlüssel \longrightarrow Schlüssellänge ≈ 4.64 Bit ($2^{4.64} \approx 25$)
leicht zu knacken
- Zufälliges Rotieren des gesamten Zeichensatzes:
Für jedes Zeichen wird neu „gewürfelt“.

1 Einführung

1.2 Einfache Verschlüsselungsalgorithmen

- Cäsar-Chiffre: Alphabet rotieren
Beispiel: ROT13 (mit Schlüssel 13)
25 verschiedene Schlüssel \rightarrow Schlüssellänge ≈ 4.64 Bit ($2^{4.64} \approx 25$)
leicht zu knacken
- Zufälliges Rotieren des gesamten Zeichensatzes:
Für jedes Zeichen wird neu „gewürfelt“.
 \rightarrow Schlüssellänge = Länge der Nachricht
unknackbar (mathematisch beweisbar)

1 Einführung

1.2 Einfache Verschlüsselungsalgorithmen

- Cäsar-Chiffre: Alphabet rotieren
Beispiel: ROT13 (mit Schlüssel 13)
25 verschiedene Schlüssel \rightarrow Schlüssellänge ≈ 4.64 Bit ($2^{4.64} \approx 25$)
leicht zu knacken
- Zufälliges Rotieren des gesamten Zeichensatzes:
Für jedes Zeichen wird neu „gewürfelt“.
 \rightarrow Schlüssellänge = Länge der Nachricht
unknackbar (mathematisch beweisbar)
praktische Probleme:
 - Schlüssel genauso lang wie Nachricht \rightarrow schwer zu übertragen
 - Schlüssel (echter Zufall!) schwer zu generieren

1 Einführung

1.2 Einfache Verschlüsselungsalgorithmen

- Cäsar-Chiffre: Alphabet rotieren
Beispiel: ROT13 (mit Schlüssel 13)
25 verschiedene Schlüssel \rightarrow Schlüssellänge ≈ 4.64 Bit ($2^{4.64} \approx 25$)
leicht zu knacken
- Zufälliges Rotieren des gesamten Zeichensatzes:
Für jedes Zeichen wird neu „gewürfelt“.
 \rightarrow Schlüssellänge = Länge der Nachricht
unknackbar (mathematisch beweisbar)
praktische Probleme:
 - Schlüssel genauso lang wie Nachricht \rightarrow schwer zu übertragen
 - Schlüssel (echter Zufall!) schwer zu generieren
- Dasselbe mit Pseudozufallsgenerator
Schlüssel = Startwert

1 Einführung

1.2 Einfache Verschlüsselungsalgorithmen

- Cäsar-Chiffre: Alphabet rotieren
Beispiel: ROT13 (mit Schlüssel 13)
25 verschiedene Schlüssel \rightarrow Schlüssellänge ≈ 4.64 Bit ($2^{4.64} \approx 25$)
leicht zu knacken
- Zufälliges Rotieren des gesamten Zeichensatzes:
Für jedes Zeichen wird neu „gewürfelt“.
 \rightarrow Schlüssellänge = Länge der Nachricht
unknackbar (mathematisch beweisbar)
praktische Probleme:
 - Schlüssel genauso lang wie Nachricht \rightarrow schwer zu übertragen
 - Schlüssel (echter Zufall!) schwer zu generieren
- Dasselbe mit Pseudozufallsgenerator
Schlüssel = Startwert
 \rightarrow Schlüssellänge = Bits des Startwerts
meistens knackbar, nur für spezielle Pseudozufallsgeneratoren sicher
 \rightarrow Verschlüsselungsverfahren

1 Einführung

1.2 Einfache Verschlüsselungsalgorithmen

| Verfahren | Schlüssellänge | sicher |
|------------------------------|-----------------------|--------|
| Cäsar-Chiffre | 4.64 Bit | nein |
| One-Time Pad (OTP) | Länge der Nachricht | ja |
| Standard-Pseudozufallszahlen | typischerweise 64 Bit | nein |
| DSA | 56 Bit | nein |
| IDEA | 128 Bit | ja |
| Blowfish | 128 Bit | ja |
| Twofish | 128 Bit | ja |
| CAST | 128 Bit | ja |
| AES / Rijndael | 128 Bit | ja |

2 Ihr Verschlüsselungs-„Führerschein“

2.1 Geheime und öffentliche Schlüssel

Verschlüsselte Kommunikation:

- Versenden der Nachricht über öffentlichen Kanal (z. B. E-Mail)
- Versenden des Schlüssels über geheimen Kanal (z. B. persönlich)

Probleme:

- Kommunikation mit Unbekannten
- Kommunikation in Gruppen

Lösung:

- Asymmetrische Verschlüsselungsverfahren
- Geheime und öffentliche Schlüssel

2 Ihr Verschlüsselungs-„Führerschein“

2.1 Geheime und öffentliche Schlüssel

Asymmetrische Verschlüsselungsverfahren:

- Unterschiedliche Schlüssel zum Ver- und Entschlüsseln
- Verschlüsseln: *öffentlicher Schlüssel*
- Entschlüsseln: *geheimer Schlüssel*

Schlüsselaustausch:

- Abhören: kein Problem mehr
- Identifikation: persönlicher Kontakt weiterhin erforderlich

Lösung:

- Digitale Signaturen
- Zertifikate – Web of Trust

| Verfahren | Schlüssellänge | sicher |
|-----------|----------------|--------|
| RSA | ab 2048 Bit | ja |
| ElGamal | ab 2048 Bit | ja |
| ECDSA | 160 Bit | ja |

2 Ihr Verschlüsselungs-„Führerschein“

2.2 Digitale Signaturen

Asymmetrische Verschlüsselungsverfahren:

- Unterschiedliche Schlüssel zum Ver- und Entschlüsseln
- Verschlüsseln: geheimer Schlüssel
- Entschlüsseln: öffentlicher Schlüssel
- Prüfwert der Nachricht verschlüsseln

—> Wer das verschlüsselt hat,
muß im Besitz des geheimen Schlüssels sein.

—> *digitale Signatur*

| Verfahren | Schlüssellänge | sicher |
|-----------|----------------|--------|
| RSA | ab 2048 Bit | ja |
| ElGamal | ab 2048 Bit | ja |
| ECDSA | 160 Bit | ja |

2 Ihr Verschlüsselungs-„Führerschein“

2.3 Zertifikate – Web of Trust

- Alice will Bob eine verschlüsselte Nachricht schicken.
- Alice lädt den öffentlichen Schlüssel von Bob herunter.

→ Ist der Schlüssel wirklich der von Bob?

2 Ihr Verschlüsselungs-„Führerschein“

2.3 Zertifikate – Web of Trust

- Alice will Bob eine verschlüsselte Nachricht schicken.
- Alice lädt den öffentlichen Schlüssel von Bob herunter.

—> Ist der Schlüssel wirklich der von Bob?

- Trent hat den öffentlichen Schlüssel von Bob unterschrieben:
„Ich bezeuge, daß dieser Schlüssel wirklich Bob gehört.“

—> Trent agiert als *Zertifizierungsstelle*.

- Alice hat bereits den öffentlichen Schlüssel von Trent und kann daher die Signatur prüfen.
- Alice vertraut Trent. —> *Vertrauen in Person*

—> Der öffentliche Schlüssel von Bob ist echt. —> *Vertrauen in Schlüssel*

3 Verschlüsselung in der Praxis

- Software: GNU Privacy Guard (GnuPG)
herunterladen und installieren
 - Plug-In für Mozilla Thunderbird: Enigmail
 - GnuPG Basics Pack
 - GPG4Win
 - ...
- Schlüsselerzeugung
- Öffentliche Schlüssel austauschen und prüfen

3 Verschlüsselung in der Praxis

- Software: GNU Privacy Guard (GnuPG)
herunterladen und installieren
 - Plug-In für Mozilla Thunderbird: Enigmail
 - GnuPG Basics Pack
 - GPG4Win
 - ...
- Schlüsselerzeugung
- Öffentliche Schlüssel austauschen und prüfen
- Schlüssel-Server
- Ende-zu-Ende-Verschlüsselung
- Auf verschlüsselte Nachrichten immer verschlüsselt antworten!
- Betreff-Zeilen werden nicht mit verschlüsselt.
- Die Verbindungsdaten werden nicht mit verschlüsselt.

4 Fazit

- E-Mail-Verschlüsselung ist sinnvoll . . .
- . . . und machbar.
- Der richtige Umgang ist entscheidend.

—> Sicherheitsbewußtsein ist entscheidend.