Michael Stehmann

Kryptographie

nur mit

Freier Software!

ErsterTeil:

Kurze Einführung in Kryptographie

Bei der Kryptographie geht es um die Zukunft von Freiheit und Demokratie

Artur P. Schmidt, 1997

http://www.heise.de/tp/artikel/1/1357/1.html

Ein Beispiel:

die katze ist leider tot.
inj pfyej nxy qjnijw yty.

```
abcdefghijklmnopqrstuvwxyz
fghijklmnopqrstuvwxyzabcde
```

Nachteil:

Neben dem Schlüssel x (hier: 5) muss auch das Verfahren (Verschiebe um x) geheim gehalten werden.

Eine lange Geschichte:

Erste Hinweise bei den Ägyptern um 1900 v. Chr.: unübliche Hieroglyphen

Quelle:
Walter Unger 28.01.2014 Vortragsabend am Tag des Datenschutzes – RWTH Aachen

https://www.asta.rwth-aachen.de/media/medien/01_unger_folien-1_96660.pdf

Sieben Namen:

- Whitfield Diffie
- Martin Hellman
- Ralph Merkle

Schlüsselaustausch zur Erzeugung eines geheimen Schlüssels

- Ronald L. Rivest
- Adi Shamir
- Leonard Adleman

RSA-Verfahren (Primzahlen)

- Taher Elgamal

Kurzer Exkurs:

Was ist Modulo?

Modulo berechnet den Rest b der Division n geteilt durch m.

Beispiele: 17 mod 3 = 2 (17 = 5 * 3 + 2)
2 mod 3 = 2 (2 = 0 * 3 + 2)

Vorteil: Es gibt unendlich viele n, für die gilt: n mod m = b

Diffie-Hellman-Merkle

Schlüsselaustauschverfahren

Quelle:
http://de.wikipedia.org/wiki/Diffie-Hellman-Schlüsselaustausch

Diffie-Hellman-Merkle: Ein Beispiel:

- Alice und Bob einigen sich auf $p = 13$ und $g = 2$.
- Alice wählt die Zufallszahl $a = 5$. Bob wählt die Zufallszahl $b = 8$.
- Alice berechnet $A = g^a \bmod p = 6$ und sendet A an Bob.
- Bob berechnet $B = g^b \bmod p = 9$ und sendet B an Alice.
- Alice berechnet $K = B^a \bmod p = 3$.
- Bob berechnet $K = A^b \bmod p = 3$.
- Beide erhalten das gleiche Ergebnis $K = 3$.

Ein eventuell vorhandener Lauscher könnte zwar die Zahlen 13, 2, 6 und 9 mithören, das eigentliche gemeinsame Geheimnis von Alice und Bob K = 3 bleibt ihm aber verborgen. K = 3 kann als Schlüssel für die nachfolgende Kommunikation verwendet werden.

Symmetrisch und Asymmetrisch

- Symmetrisch: Alice und Bob haben einen gemeinsamen geheimen Schlüssel
- Asymmetrisch: Alice und Bob haben keinen gemeinsamen geheimen Schlüssel
- sondern:

  Alice erzeugt ein Schlüsselpaar: Öffentlichen und geheimen Schlüssel

  Alice sendet Bob öffentlichen Schlüssel, Bob verschlüsselt mit diesem Nachricht

  Alice entschlüsselt Nachricht mit ihrem geheimen Schlüssel
- Public-Key-Verschlüsselungsverfahren

Hybrid

- z. B. bei GnuPG
- Ein symmetrischer Schlüssel wird ad hoc erzeugt
- Der Mailcontent wird mit diesem Schlüssel verschlüsselt
- Der Schlüssel wird mit dem Public-Key des Empfängers verschlüsselt
- Verschlüsselter Content und asymmetrisch verschlüsselter symmetrischer Schlüssel werden an Empfänger versandt

Stichworte:

- Man-in-the-middle-Attacke
- Web of Trust
- End-to-End-Verschlüsselung

Elgamal-Kryptosystem

Public-Key-Verschlüsselungsverfahren

„Das Elgamal-Verschlüsselungsverfahren beruht, wie auch das Diffie-Hellman-Protokoll, auf Operationen in einer zyklischen Gruppe endlicher Ordnung. Das Elgamal-Verschlüsselungsverfahren ist beweisbar IND-CPA-sicher unter der Annahme, dass das Decisional-Diffie-Hellman-Problem in der zugrundeliegenden Gruppe schwierig ist.“

Quelle:
http://de.wikipedia.org/wiki/Elgamal-Verschl%C3%BCsselungsverfahren

Hilfe!

Noch mehr Mathematik?

Noch mehr Mathematik?

Nein!

Praktische Sicherheit

- Das Elgamal-Verfahren ist theoretisch sicher.
- Dies gilt aber nur für den allgemeinen Fall, das heißt ein beliebiges Elgamal-Problem.
- Durch schlechte Wahl der Parameter oder Fehler in der Implementierung können Spezialfälle erzeugt werden, die dennoch unsicher sind.
- Unsicherheit beruht bei Verschlüsselungssoftware in der Regel auf einer fehlerhaften Implementation, nicht auf der Verwendung unsicherer Algoritmen.

Zweiter Teil: Freie Software

- Was ist Freie Software?
- Warum Freie Software?

Definition

Software wird frei genannt, wenn sie unter einer Lizenz verbreitet wird, die bestimmten Anforderung genügt.

Daraus folgt:

* Die Lizenz entscheidet, ob ein Programm Freie Software ist.

* Freie Software ist kein Produkt (keine Produktgruppe), sondern beschreibt eine rechtliche Eigenschaft.

4 Freiheiten

„use, study, share, improve“

„verwenden, verstehen, verbreiten, verbessern“

Die Freiheit, das Programm für jeden Zweck zu benutzen (Freiheit 0).

Die Freiheit, zu verstehen, wie das Programm funktioniert und wie man es für seine Ansprüche anpassen kann (Freiheit 1). Der Zugang zum Quellcode ist dafür Voraussetzung.

Die Freiheit, Kopien weiterzuverbreiten, so dass man seinem Nächsten weiterhelfen kann (Freiheit 2).

Die Freiheit, das Programm zu verbessern und die Verbesserungen der Öffentlichkeit zur Verfügung zu stellen, damit die ganze Gemeinschaft davon profitieren kann (Freiheit 3). Der Zugang zum Quellcode ist dafür Voraussetzung.

Quelle: Die Definition Freier Software - http://www.gnu.org/philosophy/free-sw.de.html

Was ist Freie Software?

„**Ein Programm ist Freie Software, wenn die Benutzer alle diese Freiheiten haben.“**

Quelle: Die Definition Freier Software

http://www.gnu.org/philosophy/free-sw.de.html

Warum Freie Software?

Die Funktionsweise und Sicherheit Freier Software. kann durch Studium des Quellcodes jederzeit nachvollzogen und beurteilt und, wenn nötig, auch verbessert werden.

Da viele kritische Blicke auf den Quellcode geworfen werden können, werden Sicherheitslücken in Freier Software oft schnell erkannt und in der Regel kurzfristig beseitigt.

Auch versteckte Funktionen können erkannt werden.

Kerckhoffs’ Prinzip

Benannt nach: Jean Guillaume Auguste Victor François Hubert Kerckhoffs von Nieuwenhof (* 1835 † 1903)

1883: La Cryptographie militaire

Die Sicherheit eines Verschlüsselungsverfahrens muss auf der Geheimhaltung des Schlüssels beruhen und nicht auf der Geheimhaltung des Verschlüsselungsalgorithmus.

"Security by obscurity" führt zu einem Verlust von Sicherheit, da Sicherheitsmethoden nicht von unabhängigen Dritten auf ihre Wirksamkeit überprüft und unwirksame Methoden nicht rechtzeitig verworfen werden können.

Warum Freie Software?

Ohne die Freiheit, ein Programm zu ändern oder ändern zu lassen, blieben Anwender vom Wohlwollen des Anbieters abhängig.

Man kann also, ohne dass jemanden fragen oder gar um Erlaubnis bitten zu müssen, Freie Software verbessern.

Freie Software

Ganz ohne Vertrauen geht es doch nicht!

Denn vollständige Kontrolle ist nicht möglich.

Zur Lektüre:

Erik Albers:
Schutz vor Überwachung durch Verschlüsselung mit Freier Software

https://blogs.fsfe.org/eal/2014/06/20/verschluesselung-mit-freier-software/

und

E-Mail-Selbstverteidigung

https://emailselfdefense.fsf.org/de/

Zum Schluss

Vielen Dank für Ihre Aufmerksamkeit.

Noch Fragen?

Diese Präsentation wurde erstellt mit
Apache OpenOffice – Impress

ApacheOpenOffice – die Freie Officesuite

Kontakt:

Dr. Michael Stehmann:
info@rechtsanwalt-stehmann.de