Klassische Kryptographie

Prof. Dr. Rüdiger Weis

TFH Berlin

Sommersemester 2008

Geschichte

Seit der Antike: Verbreiteter, aber unsystematischer Einsatz kryptographischer Methoden (z.B. durch Caesar).

Ende 19. Jhdt.: Systematisierung und Formalisierung.

2. Weltkrieg: Polen, Briten und Amerikaner knacken sehr starke deutsche Chiffren (u.a. "Enigma"). Erstmalig Einsatz von Rechenmaschinen zum "Code-Knacken".

70er Jahre: Data Encryption Standard (DES).
Public-Key Kryptographie.

80er Jahre: Zero-Knowledge Protokolle.

Seitdem: Massenhafte Verbreitung der Kryptographie (Geldautomaten, Internet, Mobilfunk, Pay-TV, Signaturgesetz ...).

Klassische versus Moderne Kryptographie

- Die klassische Kryptographie diente der Geheimhaltung von Nachrichten und wurde hauptsächlich von Miltiärs, Geheimdienstlern und Diplomaten genutzt.
- Die moderne Kryptographie (etwa seit 1975) beschäftigt sich mit erheblich weitergehenden Kommunikations- und Sicherheitsproblemen:
  - *"Cryptography is about communication in the presence of adversaries."* (Ron Rivest)
  - *"Die Kryptographie beschäftigt sich mit Kommunikationsproblemen in der Anwesenheit von Gegnern."*

Ziele beim Einsatz von Kryptographie:

- Die **Geheimhaltung** von Daten
  ($\longrightarrow$ klassische Kryptographie)
  (*Nur wir können diesen Text lesen.*)
- Die **Authentizität und Integrität** von Daten
  (*Du hast diesen Brief geschrieben, und niemand hat am Text etwas geändert.*)
- Die **Authentizität** von Kommunikationspartnern
  (*Ach, Du bist es!*)
- **Anonyme** Kommunikation
  (elektronisches Geld, ...).

Klassische Kryptographie

- Eine **Chiffre** wird def. durch drei Mengen
  1. P: Klartexte (Nachrichten),
  2. C: Chiffretexte (Kryptogramme),
  3. K: Schlüssel
- und zwei (bzw. drei) effiziente Algorithmen
  1. $E : K \times P \longrightarrow C$ (Verschlüsseln),
  2. $D : K \times C \longrightarrow P$ (Entschlüsseln),
  3. $G \ldots \longrightarrow K$ (Schlüssel erzeugen)).
- Für jeden Klartext $x \in P$ und jeden Schlüssel $k \in K$ gilt:

$$D(k, E(k, x)) = x.$$

Das Shannon-Modell und Kerkhoffs Prinzip

Shannon-Modell

Shannon (1949): „Der Feind kennt das benutzte System“
Kerkhoffs (1883):

- Unterscheidung zwischen Kryptosystem und Schlüssel.
- Forderung: Ein System soll auch dann sicher sein, wenn der Gegner das System kennt, bis auf den verwendeten Schlüssel.

Einige typische Angriffe

- known ciphertext
- known plaintext
- chosen plaintext
- chosen ciphertext
- chosen plaintext / chosen ciphertext („zweiseitiger Angriff“)

Die Caesar-Chiffre

Julius Caesar verschlüsselte seine Nachrichten, indem er Klartext-Buchstaben „a“, . . . , „z“ auf die folgende Weise auf Chiffretext-Buchstaben „A“, . . . , „Z“ abbildete:

a	$\longrightarrow$	D		v	$\longrightarrow$	Y
b	$\longrightarrow$	E		w	$\longrightarrow$	Z
c	$\longrightarrow$	F	$\cdots$	x	$\longrightarrow$	A
d	$\longrightarrow$	G		y	$\longrightarrow$	B
e	$\longrightarrow$	H		z	$\longrightarrow$	C

Beispiel: „`caesar`“ $\longrightarrow$ „FDHVDU“

Die Caesar-Chiffre (2)

Ist das nun eine Chiffre?

- Klartextmenge $P = \{\text{„a“}, \ldots, \text{„z“}\}$.
- Chiffretextmenge $C = \{\text{„A“}, \ldots, \text{„Z“}\}$ (Großbuchstaben wg. Übersicht).
- Ver- und Entschlüsselungsalgorithmus E und D: Klar!
- Schlüsselmenge???

Die Caesar-Chiffre (3)

Man ordne den Buchstaben eine Zahl aus der Menge $\mathbb{Z}_{26} = \{0, \ldots, 25\}$ zu:

a	bzw.	A	$\leftrightarrow$	0
b	bzw.	B	$\leftrightarrow$	1
c	bzw.	C	$\leftrightarrow$	2
y	bzw.	Y	$\leftrightarrow$	24
z	bzw	Z	$\leftrightarrow$	25

Mengen $P = C = K = \mathbb{Z}_{26}$.

Die Caesar-Chiffre (4)

- Verschlüsseln: $E(k, x) = x + k \bmod 26$.
- Entschlüsseln: $D(k, y) = y - k \bmod 26$.
- Schlüsselerzeugung G: Trivial.

Die Algorithmen sich offenbar effizient. Es gilt

$$D(k, E(k, x)) = x + k - k = x.$$

Wir haben eine Chiffre!!!

Die Caesar-Chiffre (5)

Beispiel:

Wir haben den Chiffretext

“PHH WPH DWW KHX VXD OSO DFH DWH LJK WRF ORF N”

aufgefangen. Wer kann diese Nachricht dechiffrieren?

Die Substitutionschiffre

Bei einer Substitutionschiffre ordnet man jedem Klartext-Buchstaben eindeutig einen Chiffretext-Buchstaben zu, z.B.:

a	$\longrightarrow$	D		v	$\longrightarrow$	I
b	$\longrightarrow$	R		w	$\longrightarrow$	K
c	$\longrightarrow$	L	$\cdots$	x	$\longrightarrow$	F
d	$\longrightarrow$	M		y	$\longrightarrow$	X
e	$\longrightarrow$	H		z	$\longrightarrow$	Z

Die Substitutionschiffre (2)

Die Caesar-Chiffre ist eine Spezialfall der Substitutionschiffre. Während die Caesar-Chiffre bei einem Alphabet der Größe 26 nur 26 verschiedene Schlüssel erlaubt, sind es bei der Substitutionschiffre

$$26! = 26 * 25 * 24 * \cdots * 2 * 1 \approx 4 * 10^{26} \approx 2^{88.4}$$

„Blindes“ Ausprobieren („brute-force“) führt bei der Caesar-Chiffre schnell zum Erfolg, ist bei der Substitutionschiffre aber aussichtslos.

Die Substitutionschiffre (3)

Doch: Auch die Substitutionschiffre ändert die Spachstatistik nicht. Im Deutschen ist

- etwa ein Sechstel aller Buchstaben ein „e“,
- etwa ein Drittel aller Buchstaben einer der drei häufigsten Buchstaben „e“, „n“, oder „i“, und
- etwa zwei Drittel aller Buchstaben einer der acht häufigsten Buchstaben.

Die Häufigkeit von Buchstabenpaaren, -tripeln, . . . , ist auch hilfreich.

Die Substitutionschiffre (4)

Kann man genug (z.B., die häufigsten acht) Buchstaben richtig zuordnen, kann man den Text fast schon flüssig lesen.

$\Longrightarrow$ Ist der Klartext deutsch (englisch, französisch, ...), und wird buchstabenweise verschlüsselt, so ist eine Substitutionschiffre hochgradig verwundbar gegen known ciphertext Angriffe.

Die Vignére-Chiffre

Ziel: Verschleiern der Sprachstatistik!

Idee: Wähle m unabhängige Schlüssel $k_0, \ldots, k_{m-1}$. Benutze zur Verschlüsselung des i-ten Buchstabens den Teilschlüssel k_i.

Detailproblem: Man braucht arg viel Speicherplatz (Gedächtnis), um den Schlüssel zu speichern (sich zu merken), da es $26!^m$ viele Schlüssel gibt.

Lösung: Caesar-Chiffren als Teil-Chiffren.
Dann: 26^m Schlüssel; für $m = 19$: $26^{19} \approx 2^{89.3}$.
Verschleierung der Sprachstatistik, Erweiterung der Caesar-Chiffre mit annehmbarer Schlüssellänge.

Die Kryptoanalyse der Vignére-Chiffre

Ist m bekannt, dann ist die Kryptoanalyse mittels Buchstabenhäufigkeiten ebenso einfach, wie bei der Caesar-Chiffre.

Und wenn der Gegner m nicht kennt?

Dann kann er es entweder mit **Ausprobieren** versuchen, oder mit **statistischen Methoden**.

Die Vignére-Chiffre (mehr)

Ausprobieren:

Sei Chiffretext y gegeben.

Setze $m' = 1$ und versuche, y zu dechiffrieren.

Scheitert dies, probiere $m' = 2, 3, \ldots$

Ist $m' = m$, dann kann man erwarten, daß es dem Gegner gelingt, y zu dechiffrieren
– nachdem dies für alle $m' < m$ scheiterte.

Die Vignére-Chiffre (noch mehr)

Braucht der Gegner Zeit T, um bei bekanntem m die Chiffre zu „knacken“, kann der für diese Methode erforderliche Zeitaufwand mit mT abgeschätzt werden.

Ein Anwachsen um den Faktor m ist gut zu verkraften – zumal der (Speicher-) Aufwand für den legalen Nutzer um den Faktor m zunimmt.

Die Vignére-Chiffre (noch mehr)

Statistische Methoden erlauben es dem Gegner, anhand der prinzipiell bekannten Sprachstatistik der Klartextmenge und der einfach zu messenden Sprachstatistik des gegebenen Chiffretextes y den Wert m zu schätzen, um y noch schneller zu dechiffrieren.

Bei der Kryptoanalyse von Vignére-Chiffren verwendet man insbesondere den **Friedmann-Test** („phi-Test“) und den **Kasiski-Test** („kappa-Test“).

Diese Tests werden im Rahmen dieser Vorlesung nicht weiter behandelt.

Mehrschleifige Vignére-Chiffren

Verwende zwei (oder mehr) Teilschlüssel

$$\begin{aligned} k &= (k_0, \ldots, k_{R-1}) \\ l &= (l_0, \ldots, l_{S-1}) \end{aligned}$$

der Längen R und L (möglichst teilerfremd). Die Chiffretextbuchstaben y_i errechnen sich aus den Klartextbuchstaben x_i durch:

$$y_i = x_i + k_{i \bmod R} + l_{i \bmod S}.$$

$\Longrightarrow$ gute Verschleierung der Sprachstatistik
$\Longrightarrow$ Ausprobieren kann aufwendig sein
$\Longrightarrow$ Trotzdem known ciphertext Angriffe mit statistischen Methoden

©opyleft

©opyleft

- Erstellt mit Freier Software
- © Rüdiger Weis, Berlin 2008
- nach Folien von Stefan Lucks
- unter der GNU Free Documentation License.