Inhaltsverzeichnis

1 Einleitung und historische Beispiele **2**
- 1.1 Skytale . . . 2
- 1.2 Caesar-Chiffre . . . 3
- 1.3 affine Chiffre . . . 3
- 1.4 Historische Beispiele . . . 4

2 Absolute Sicherheit und Informationstheorie **5**
- 2.1 One-Time-Pad von Vernam . . . 7
- 2.2 Entropie . . . 8

3 Stromchiffren und Schieberegister **10**
- 3.1 Golombsche Axiome . . . 11
- 3.2 Berlekamp-Massey-Algorithmus . . . 17

4 Symmetrische Kryptoverfahren **19**
- 4.1 Feistel-Chiffren . . . 19
- 4.2 Data Encryption Standard (DES) . . . 20
- 4.3 Advanced Encryption Standard (AES) . . . 21
- 4.4 Betriebsarten symmetrischer Verfahren . . . 21
  - 4.4.1 Electronic Code Book (ECB) Mode . . . 21
  - 4.4.2 Cipher Block Chaining (CBC) Mode . . . 22
  - 4.4.3 Cipher Feedback (CFB) Mode . . . 23

5 Asymmetrische Kryptoverfahren **25**
- 5.1 Zahlentheoretische Grundlagen . . . 26
  - 5.1.1 kleiner Fermat . . . 26
  - 5.1.2 Erweiterter Euklidischer Algorithmus (EEA) . . . 27
- 5.2 Das RSA-Verfahren . . . 27
- 5.3 Sicherheit von RSA . . . 28
- 5.4 Weitere Anwendungen von RSA . . . 29

6 Der diskrete Logarithmus **30**
- 6.1 Schlüsselaustausch nach Diffie / Hellman . . . 30
- 6.2 Verschlüsselung nach ElGamal . . . 31
- 6.3 Berechnung diskreter Logarithmen . . . 32

7 Primzahltests **34**
- 7.1 Sieb des Eratosthenes . . . 34
- 7.2 Fermat-Test . . . 35
- 7.3 $(p-1)$-Test . . . 36
- 7.4 Miller-Rabin-Test . . . 36

8 Faktorisierung **37**
- 8.1 Probedivision . . . 37
- 8.2 $(p-1)$-Methode von Pollard . . . 37
- 8.3 Quadratisches Sieb (Pomerance 1982) . . . 38
- 8.4 Auswahl geeigneter Kongruenzen . . . 39

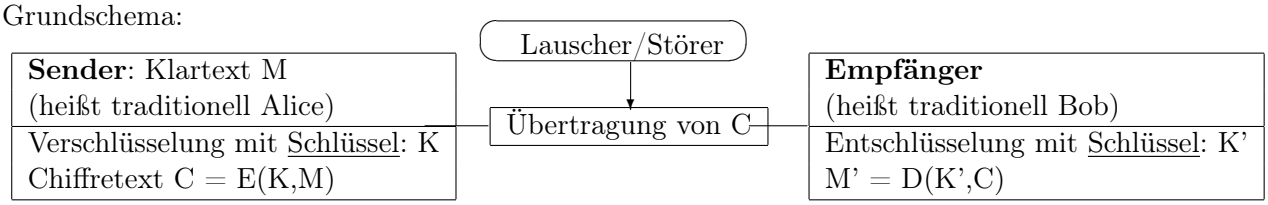
01. Vorlesung (02. April 2008)

1 Einleitung und historische Beispiele

Unterscheidung:

- KODIERUNGSTHEORIE: Sicherung der Nachrichtenübertragung gegen Informationsverlust durch Störungen (Rauschen, Datenverlust, technische Fehler, ...)
  Bsp.: ISBN, CDs $\rightsquigarrow$ kodieren[1] von Nachrichten
- KRYPTOGRAPHIE: Sicherung der Vertraulichkeit, Authentizität und Integrität der Nachrichtenübertragung $\rightsquigarrow$ Verschlüsseln der Nachricht

Grundschema:



Anmerkung: D = Decrypt und E = Encrypt.
Sinnvolles Verfahren: $M' = M$. Wenn K, K' eng zusammenhängen, spricht man von *symmetrischen* Verfahren, ansonsten von einem *asymmetrischen* Verfahren.

KRYPTOANALYSE: Angriffe auf ein Kryptosystem mit dem Ziel, es zu brechen
(Lauscher heißt "Eve")
Verschiedene Szenarien sind möglich:

- "cypher text only attack": man kennt nur den Chiffretext
- "Known plain text attack": man kennt gewisse Klartexte und die zugehörigen Chiffretexte
- "chosen plain text attack": man kann zu beliebigen Klartexten die Chiffretexte erzeugen

Ein gutes System sollte all diesen Typen von Angriffen wiederstehen.

Prinzip von Kerkhoffs: Sicherheit des Verfahrens beruht ausschließlich auf Geheimhaltung des Schlüssels (Verfahren selbst allgemein bekannt)

1.1 Skytale

Zylinder (Holzstab) mit festgelegtem Durchmesser (der Schlüssel K) umwickelt mit einem Streifen Papyrus. (vor ca. 2400 Jahren)

Schreibe in Richtung der Achse und wickle ab $\rightsquigarrow$ Verschlüsselung. Zum Entschlüsseln braucht man den Durchmesser des Stabs ($K' = K$)

[1] kodieren im rein technischen Sinn

Abbildung 1: Scytale

moderne Variante:

$$\begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}$$

Schreibe Klartext zeilenweise, lese Chriffretext aber Spaltenweise aus:

Bsp.: KAHERPTNYHUFPINRTESEOMAUGALDRCLE (4x8 Matrix)

$$\begin{bmatrix} K & R & Y & P & T & O & G & R \\ A & P & H & I & E & M & A & C \\ H & T & U & N & S & A & L & L \\ E & N & F & R & E & U & D & E \end{bmatrix}$$

"mathematisches Prinzip": Alphabet bleibt gleich, Stellen werden permutiert.

1.2 Caesar-Chiffre

Ersetze jeden Buchstaben durch den, der b-Positionen dahinter im Alphabet steht (Caesar: $b = 3$) (vor ca. 2000 Jahren).

$$\begin{bmatrix} A & B & C & D & E & \cdots \\ X & Y & Z & A & B & \cdots \end{bmatrix}$$

$\Rightarrow$ Krypto $\rightsquigarrow$ HOVMQL

70:45

allgemeine mathematische Formulierung:
Sei $\mathcal{A}$ ein Alphabet[2]; $\mathcal{A}$ zugleich ein Ring[3]; z. B.: $A = \mathbb{Z}$ (mit Sonderzeichen $\mathcal{A} = \mathbb{Z}_m$ mit $m > 26$)
Zerlege Klartext in *Blöcke* der Länge $n \leq \mathbb{N}(M_1, M_2, M_3, \ldots) \in \mathcal{A}^n \times \mathcal{A}^n \times \ldots$. Wähle als Schlüssel eine Permutation $\sigma : A^n \to A^n$ bijektiv[4] (man spricht von einer *Substitution*) 76:05
Verschlüsselung $(\sigma(M_1), \sigma(M_2), \cdots)$:

- n = 1: monoalphabetische Substitution (bijektive Abbilidung von $\mathcal{P}$ auf $\mathcal{C}$-Text)
- n > 1: polyalphabetische Substitution

79:26

1.3 affine Chiffre

Matrix[5] $S \in Gl(n, A)$, Vektor $b \in \mathcal{A}^n$ setze $\sigma(M) = S \cdot M + b = C$ (also $\sigma^{-1}(c) = S^{-1}(c - b)$)

[2] Menge von Symbolen
[3] d. h. wir können Elemente im Alphabet addieren & multiplizieren
[4] d.h. es gibt inverse
[5] reguläre $n \times n$-Matrix mit Einträgen aus $\mathcal{A}$

WIKI: Bei diesem Verschlüsselungsgerfahren wird der Klartext Buchstabe für Buchstabe nach einer bestimmten mathematischen Formel verschlüsselt. Die affine Chiffre lässt sich zwar ohne größeren Aufwand berechnen. Dafür ist sie allerdings nicht besonders sicher. Einerseits gibt es nur eine begrenzte Anzahl geheimer Schlüssel, sodass diese alle durchprobiert werden können. Andererseits kann der Geheimtext entschlüsselt werden, sobald die Verschlüsselung von nur zwei Zeichen bekannt ist.

1.4 Historische Beispiele

(n = Blöcke der Länge n mit Klartext)

- Caesar Chiffre: $n = 1, S = 1, b = 3$
- Verschiebung: $n = 1, S = 1, b =$ beliebig aber fest
- Vigenère-Chiffre: $n > 1, S = E_n, b =$ beliebig aber fest
- Hill-Chiffre: $n = 2, S \neq 1, b = 0$

02. Vorlesung (09. April 2008)

WIKI: *Ring* (d. h. es existieren Addition und Mulitplikation)
Ein Ring ist eine algebraische Struktur, in der, ähnlich wie in den ganzen Zahlen $\mathbb{Z}$, Addition und Multiplikation definiert und miteinander bezüglich Klammersetzung verträglich sind. (Die Namensgebung Ring bezieht sich nicht auf etwas anschaulich Ringförmiges, sondern auf einen Zusammenschluss von Elementen zu einem Ganzen.)
-> kein multiplikativ Inverses

Körper
Ein Körper ist im mathematischen Teilgebiet der Algebra eine ausgezeichnete algebraische Struktur, in der die Addition, Subtraktion, Multiplikation und Division wie bei den "normalen" (reellen) Zahlen durchgeführt werden können.

$\mathbb{F}$ (Körper im englischen Field. Wir haben endliche Körper (Finite Field))

03:11

Wieviele Schlüssel gibt es bei affinen Chiffren?
Annahme: $\mathcal{A} = \mathbb{Z}_p$ mit $p \in \mathbb{P}$ Primzahl (d. h. $\mathcal{A}$ ist ein Körper)

- Matrix $S \in Gl(n, \mathcal{A}) \Rightarrow$ Spalten müssen linear unabhängig sein
  1. Spalte: $p^n - 1$ Möglichkeiten (Nullvektor ausschließen)
- 2. Spalte: $p^n - p$ Möglichkeiten (kein Vielfaches der ersten Spalte)
  $\vdots$
  k. Spalte: $p^n - p^{k-1}$ Möglichkeiten (kein LK der vorherigen Spalte)
  $\Rightarrow |Gl(n, \mathcal{A})| = (p^n - 1) \cdot (p^n - p) \cdot \ldots \cdot (p^n - p^{k-1})$
- $b \in \mathcal{A}^n \Rightarrow p^n$ Möglichkeiten, also gibt es $p^k(p^n - 1) \cdot \ldots \cdot (p^k - p^{n-1})$ verschiedene Schlüssel
  betrachte Konkret p = 29
  n = 1 $\rightsquigarrow 29^1 \cdot (29 - 1) = 812$
  n = 4 $\rightsquigarrow 29^4 \cdot (29^4 - 1) \cdot (29^4 - 29) \cdot (29^4 - 29^2) \cdot (29^4 - 29^3) = 1,7 \cdot 10^{29}$

17:55

Falls nicht unbedingt affine Chiffren $\rightsquigarrow |S(A^n)| = (|A|^n)!$
Sei wieder $b|\mathcal{A}| = 29$
n = 1 $\rightsquigarrow 29! \approx 8,8 \cdot 10^{30}$
n = 4 $\rightsquigarrow (29^4)! \approx 2,3 \cdot 10^{383014}$

ANNAHME: pro Sekunde lassen sich 10^6-Schlüssel austesten.
1 Jahr hat ca. 10^8-Sekunden $\rightsquigarrow$ teile durch 10^{14} $\rightsquigarrow$ Rechenzeit in Jahren
z.B. $n = 4$ Vigénère: $29^4 = 7 \cdot 10^5 \rightsquigarrow 1$ sec

27:12

Besserer Angriff, wenn n bekannt ist
Analysiere Häufigkeit von Buchstaben, beispielsweise Buchstabenkombinationen in der Sprache des Klartexts, z.B. im Deutschen (Ohne Sonderzeichen)

- e = 18.1 %, n = 10.42%, r = 8.08%, i = 7.52%, s = 6.35%, t = 5.57%, $\cdots$, x = 0.01%,
- Bigramme: EN, ER, CH, ND, ES
- Trigramme: EIN, ICH, NDE, DIE, UND

33:25

DEF. 1: Ein *Verschlüsselungssystem* oder *Kryptosystem* ist ein
Fünftupel $(\mathcal{P}, \mathcal{C}, \mathcal{K}, E(K,\cdot), D(K,\cdot))$ mit:

- einer Menge $\mathcal{P}$ (von Plain), der *Klartextraum*,
- einer Menge $\mathcal{C}$ (von Chiffre), der *Chiffretextraum*,
- einer Menge $\mathcal{K}$ (von Key), der *Schlüsselraum*,
- für jeden Schlüssel $K \in \mathcal{K}$ eine *Verschlüsselungsfunktion* $E(K,\cdot) : \mathcal{P} \to \mathcal{C}$,
- für jeden Schlüssel $K \in \mathcal{K}$ eine *Entschlüsselungsfunktion* $D(K,\cdot) : \mathcal{C} \to \mathcal{P}$,

so dass es zu jedem $K \in \mathcal{K}$ ein $K' \in \mathcal{K}$ gibt mit $D(K', E(k,p)) = p$ für alle $p \in \mathcal{P}$

41:29

BSP. 1: (i) *Verschiebungschiffre*
$\mathcal{P} = \mathcal{C} = \mathcal{K} = \{A, B, \ldots, Z\}$“ = “$\{0, 1, 2, \ldots, 25\}$

$$E(K,p) = (p+k) \mod 26$$
$$D(K,c) = (c-k) \mod 26$$

(ii) *Blockchiffre*
$\mathcal{P} = \mathcal{C} = \{A, B, \ldots, Z\}^n$“ = “$\{0, 1, 2, \ldots, 25\}$
n = Blocklänge
$\mathcal{K} = S(P)$ symmetrische Gruppe,
alle Permutationen $E(K,p) = K(p), D(K,c) = K^{-1}(c)$ [6]

48:46

2 Absolute Sicherheit und Informationstheorie

DEF. 1: Sei $\Omega \neq \varnothing$ eine endliche Menge, der *Ereignisraum* und $P : \mathcal{P}(\Omega) \to [0,1]$ eine Abbildung mit (i) $P(\Omega) = 1$ und (ii) $P(A \cup B) = P(A) + P(B)$ für alle $A, B \subseteq \Omega$ mit $A \cap B = \varnothing$, die *Wahrscheinlichkeitsverteilung.*
Dann heißt (Ω, P) *diskreter Wahrscheinlichkeitsraum.*
Eine Teilmenge $A \subseteq \Omega$ heißt *Ereignis*,
falls $|A| = 1$, spricht man von einem *Elementarereignis.*
Ω ist das sichere, ϕ das unmögliche Ereignis.

55:52

BSP. 2: Würfel $\Omega = \{1, 2, \ldots, 6\}$, $P(A) = \frac{|A|}{|\Omega|}$
Elementarereignisse: $P(i) = 1/|\Omega| = \frac{1}{6} \rightsquigarrow$ *Gleichverteilung* für alle $i \in \Omega$
$A = \{2, 4, 6\} \rightsquigarrow$ Ereignis “gerade Zahl” gewürfelt
$(ii) \Rightarrow P(A) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}$

60:30

[6] inverse Abbildung

DEF. 3: Sei (Ω, P) ein Wahrscheinlichkeits-Raum, $A, B \leq \Omega$ mit P ist P(B) > 0. Dann heißt $P_B(A) = \frac{P(A \cap B)}{P(B)}$[7] die *bedingte Wahrscheinlichkeit* von A bzgl. B. (dies ist die Wahrscheinlichkeit, mit der A eintritt, wenn wir bereits wissen, dass B eingetreten ist). Zwei Ereignisse A, B sind *unabhängig*, wenn $P(A \cap B) = P(A) \cdot P(B)$ (offensichtlich
65:16 gilt dann: $P_B(A) = P(A)$ und $P_A(B) = P(B)$).

BEW. 4: gegeben sind zwei Wahrscheinlichkeits-Räume $(\Omega_1, P_1), (\Omega_2, P_2) \rightsquigarrow$ neuer Wahrscheinlichkeits-Raum (ω, P) mit

$$\Omega = \Omega_1 \times \Omega_2 = \{(\omega_1 \times \omega_2) | \omega_1 \in \Omega_1, \omega_2 \in \Omega_2\}$$

und

$$P((\omega_1, \omega_2)) = P_1(\omega_1) \cdot P_2(\omega_2)$$

betrachte zu $\omega_1 \in \Omega_1$ das Ereignis

$$\hat{\omega}_1 = \{(\omega_1, \omega_2') | \omega_2 \in \Omega_2\}$$

offensichtlich gilt $P_1(\hat{\omega_1}) = P_1(\omega_1)$
Beweis:

$$P(\hat{\omega}_1) = \sum_{\omega_2' \in \Omega_2} P((\omega_1, \omega_2')) = \sum_{\omega_2' \in \Omega_2} P(\omega_1) \cdot P(\omega_2') = P_1(\omega_1)$$

Analog: $\omega_2 \in \Omega_2 \rightsquigarrow \hat{\omega}_2 = \{(\omega_1, \omega_2') | \omega_1 \in \Omega_1\} \leq \Omega$
$P(\hat{\omega_1}) = P_2(\omega_2)$ Behauptung: $\hat{\omega}_1, \hat{\omega}_2$ unabhängig
Beweis: $P(\hat{\omega}_1, \hat{\omega}_2) = P((\omega_1, \omega_2)) = P_1(\omega_1) \cdot P_2(\omega_2) = P(\hat{\omega}_1), P(\hat{\omega}_2) \square$

80:30
Betrachte Kryptosystem: $(\mathcal{P}, \mathcal{C}, \mathcal{K}, E(K, \cdot), D(K, \cdot))$
Wahrscheinlichkeits-Verteilung $P_\mathcal{P}$ der möglichen Klartexte, Wahrscheinlichkeits-Verteilung $P_\mathcal{K}$ der verwendeten Schlüssel
Zwei Wahrscheinlichkeits-Räume: $(\mathcal{P}, P_\mathcal{P}), (P_\mathcal{K}, P_\mathcal{K})$ gemäß Beweis 4 bilden wir $(\Omega = P \times \mathcal{K}, P)$ zu jedem $p \in \mathcal{P}$ bzw. bilde $\hat{p}, \hat{\mathcal{K}} \in \Omega$

ANNAHME: Angreifer kennt $P_\mathcal{P}$ und hat Chiffretext $c \in \mathcal{C}$ abgefangen,
d. h. das Ereignis $\hat{C} = \{(P, K) | E(p, k) = c\}$ ist eingetreten.

03. Vorlesung (16. April 2008)

Klartext $p \in \mathcal{P}$ soll mit Wahrscheintlichkeit $P_\mathcal{P}(p)$ auftreten. Bei jeder Übertragung wird zufällig ein Schlüssel $K \in \mathcal{K}$ ausgewählt $\rightsquigarrow$ Wahrscheinlichkeitsraum $(\mathcal{K}, P_\mathcal{K})$

BEM. 4: $\rightsquigarrow (\Omega = \mathcal{P} \times \mathcal{K}, P)$ mit $P((p, k)) = P_\mathcal{P}(p) P_\mathcal{K}(K)$ betrachte zu $p \in \mathcal{P}, k \in \mathcal{K}$ die
10:59 Ereignisse $\hat{p} = \{(p, k') | k' \in \mathcal{K}\}, \hat{k} = \{(p', k) | p' \in \mathcal{P}\}$

ANNAHME: Angreifer kennt P_p und hat Chiffretext $c \in \mathcal{C}$ abgefangen
$\rightsquigarrow$ Ereignis $\hat{c} = \{(p, k) | E(k, p) = c\}$ eingetreten, betrachte WS $P_{\hat{c}}(\hat{p})$ und vergleiche mit $P_\mathcal{P}(p)$.
17:01

[7] $P_B(A)$ alternative Schreibweise: $P(A|B)$

DEF. 5: Das Kryptosystem heißt *absolut sicher* (oder *perfekt geheim*), wenn für alle $p \in P, c \in \mathcal{C}$ gilt $P_{\hat{c}}(\hat{p}) = P_{\mathcal{P}}(p)$.

BSP. 6:

$$\mathcal{P} = \{0,1\} \quad \text{mit} \quad P_{\mathcal{P}}(0) = \frac{1}{4}, P_{\mathcal{P}}(1) = \frac{3}{4}$$
$$\mathcal{K} = \{A,B\} \quad \text{mit} \quad P_{\mathcal{K}}(A) = \frac{1}{4}, P_{\mathcal{K}}(B) = \frac{3}{4}$$
$$\mathcal{C} = \{a,b\}$$
mit
$$E(A,0) = a, E(B,0) = b$$
$$E(A,1) = b, E(B,1) = a$$

$\Rightarrow$

$$P(\hat{0}) = P_{\mathcal{P}}(0) = \frac{1}{4}$$
$$P_{\hat{a}}(\hat{0}) = \frac{P(\hat{a} \cap \hat{0})}{P(\hat{a})} = \frac{P((0,A))}{P(\{(0,A),(1,B)\})} = \frac{P_{\mathcal{P}}(0) \cdot P_{\mathcal{K}}(A)}{P_{\mathcal{P}}(0) \cdot P_{\mathcal{K}}(A) + P_{\mathcal{P}}(1) \cdot P_{\mathcal{K}}(B)} = \frac{1/16}{1/16 + 9/16} = \frac{1}{10} \neq \frac{1}{4}$$

26:55

Umgekehrt $P_{\hat{a}}(\hat{1}) = \frac{9}{10}$ wenn Chiffretext a abgefangen wird, war der Klartext mit Wahrscheinlichkeit von $90\% = 1 \Rightarrow$ das System ist nicht absolut sicher.

2.1 One-Time-Pad von Vernam

WIKI: Das One-Time-Pad gehört zu den polyalphabetischen Substitutionsverfahren, bei denen die einzelnen Buchstaben in jeweils andere Buchstaben umgewandelt (verschlüsselt) werden. Kennzeichnendes Merkmal der Einmalverschlüsselung ist die einmalige Verwendung eines zufälligen Schlüssels, der (mindestens) die gleiche Länge wie die zu verschlüsselnde Nachricht aufweist. Es ist die einzige kryptographische Methode, welche informationstheoretisch sicher ist und nachweislich nicht gebrochen werden kann - vorausgesetzt, sie wird bestimmungsgemäß verwendet.

BSP. 7: *One-Time-Pad von Vernam*
$\mathcal{P} = \mathcal{C} = \mathcal{K} = \{0,1\}^n = (\mathbb{Z}_2)^n$ E(k, p) = p + k D(k, c) = c + k
wähle für $P_{\mathcal{K}}$ die Gleichverteilung: $P_{\mathcal{K}}(k) = 2^{-n}$, $P_{\mathcal{P}}$ darf beliebig sein.

$$P_{\hat{c}}(\hat{p}) = \frac{P(\hat{c} \cap \hat{p})}{P(\hat{c})} = \frac{P(p,k)}{\sum\limits_{\tilde{p} \in \mathcal{P}} P((\tilde{p}, c - \tilde{p}))} = \frac{P_{\mathcal{P}}(p) \cdot 2^{-n}}{\sum\limits_{\tilde{p} \in \mathcal{P}} P_{\mathcal{P}}(\tilde{p}) \cdot 2^{-n}} = P_{\mathcal{P}}(p)$$

$\Rightarrow$ One-Time Pad absolut sicher (Nachteil: höherer Aufwand für Schlüsselaustausch)

40:59

SATZ 8: Sei $(\mathcal{P}, \mathcal{C}, \mathcal{K}, E(K,\cdot), D(K,\cdot))$ ein Kryptosystem mit $|\mathcal{C}| = |\mathcal{K}|$ und $P_{\mathcal{P}}(p) > 0$ für alle $p \in \mathcal{P}$. Dieses System ist genau dann absolut sicher, wenn $\mathcal{K}$ gleichverteilt ist und es zu jedem $p \in \mathcal{P}$ und $c \in \mathcal{C}$ genau ein $k \in \mathcal{K}$ mit $E(k,p) = c$.

Beweis: "⇐". Sei $K_{p,c} \in \mathcal{K}$ der eindeutige Schlüssel mit $E(K_{p,c}, p) = c$.

$$\text{“} \Rightarrow \text{”} P_{\hat{c}}(\hat{p}) = \frac{P((p, K_{p,c}))}{\sum\limits_{\tilde{p}\in\mathcal{P}} P((\tilde{p}, K_{\tilde{p},c}))} = \frac{P_{\mathcal{P}}(p)\cdot|\mathcal{K}|^{-1}}{\sum\limits_{\tilde{p}\in\mathcal{P}} P_{\mathcal{P}}(\tilde{p})\cdot|\mathcal{K}|^{-1}} = P_{\mathcal{P}}(p)$$

50:01 System absolut sicher. "⇒": Das System sei absolut sicher.

Ann.: zu vorgegebenen $p \in \mathcal{P}, c \in \mathcal{C}$ gäbe es kein $k \in \mathcal{K}$ mit $E(k,p) = c$.

$$\Rightarrow P_{\hat{c}}(\hat{p}) = 0, \quad \text{da} \quad \hat{p} \cap \hat{c} = 0$$
$$\Rightarrow P_{\hat{c}}(\hat{p}) \neq P_{\mathcal{P}} > 0 \lightning$$

56:20 $|\mathcal{K}| = |\mathcal{C}| \Rightarrow$ es gibt genau einen Schlüssel $K_{p,c} \in \mathcal{K}$ mit $E(K_{p,c}) = c$

$$P_{\hat{c}}(\hat{p}) = \frac{P(\hat{c}\cap\hat{p})}{P(\hat{c})} \wedge P_{\hat{p}}(\hat{c}) = \frac{P(\hat{c}\cap\hat{p})}{P(\hat{p})} \Rightarrow P_{\hat{c}}(\hat{p}) = P_{\hat{c}}(\hat{p}) \cdot \frac{P(\hat{p})}{P(\hat{p})} \quad \text{(Formel von Bayes)}$$

System absolut sicher $\Rightarrow P_{\mathcal{P}}(p) = P_{\hat{c}}(\hat{p})$

$$P_{\mathcal{P}}(p) = P_{\hat{p}}(\hat{c}) \cdot \frac{P_{\mathcal{P}}(p)}{P(\hat{c})} = P_{\hat{p}}(\hat{c}) = P(\hat{c})$$

andererseits: $\Rightarrow P_{\hat{p}}(\hat{c}) = \frac{P(\hat{p}\cap\hat{c})}{P(\hat{p})} = \frac{P(\mathcal{P}, K_{p,c})}{P_{\mathcal{P}}(p)} = P_{\mathcal{K}}(K_{p,c})$

$\Rightarrow P_{\mathcal{K}}(K_{p,c}) = P(\hat{c})$ unabhängig von p! Zu jedem $K \in \mathcal{K}$ und $C \in \mathcal{C}$ existiert genau ein $p_{k,c} \in P$ mit $E(k, p_{k,c}) = c$
$\Rightarrow$ Gleichverteilung $P_{\mathcal{K}}(k) = |\mathcal{K}|^{-1}$

2.2 Entropie

67:47 DEF. 9: Sei (Ω, P) ein diskreter Wahrscheinlichkeits-Raum. Die *Entropie* (Unsicherheit, Unordnung) eines Wahrscheinlichkeits-Raums (Ω, P) ist
$H(\Omega, P)'' = H(\Omega)'' = -\sum\limits_{\substack{\omega_p\in\Omega \\ P(\omega)>0}} P(\omega)\log_2 P(\omega)$

77:07 (wegen $\lim\limits_{t\to 0} t\log_2 t = 0$) kann die Einschränkung $P(\omega) > 0$ weggelassen werden.

SATZ 10: Sei (Ω, P) ein diskreter Wahrscheinlichkeits-Raum:

(i) $H(\Omega) > 0$
(ii) $H(\Omega) = 0 \Leftrightarrow$ es gibt ein $\omega \in \Omega$ mit $P(\omega) = 1$
81:09 (iii) $H(\Omega) \leq \log_2 |\Omega|$ und Gleichheit gilt genau dann, wenn Ω gleichverteilt ist

Beweis: —

Abbildung 2: Irgendein Graph

04. Vorlesung (23. April 2008)

maximiere rechte Seite in Abhängigkeit von q .
Ableitung nach $q \rightsquigarrow -\log_2(n-1) - \log_2 q + \log_2(1-q) = \log_2\left(\frac{1-q}{q(n-1)}\right)$
Ableitung wird 0 für $1-q = q(n-1) \Leftrightarrow q = \frac{1}{n}$
also $p_n = q = \frac{1}{n} \Rightarrow P_i = \frac{1-\frac{1}{n}}{n-1} = \frac{1}{n}$ für $1 \leq i < n \Rightarrow$ Maximum für Gleichverteilung
Die Entropie eines Kryptosystems $(\mathcal{P}, \mathcal{C}, \mathcal{K}, E, D)$ wird definiert als $H(P, P_p)$ ("Informationsgehalt pro Klartext").
Die *bedingte Entropie* ist definiert als: $H_e(\mathcal{P}) = -\sum_{c \in e} P_{\hat{c}}(\hat{p}) \cdot \underbrace{\sum_{c \in e} P_{\hat{c}}(\hat{p}) \log_2 P_{\hat{c}}(\hat{p})}_{*}$

* = benötigte Informationen, wenn Klartext aus Chiffretext c zugewiesen.

Satz 11: (i) $H(\mathcal{P}) \geq H_e(\mathcal{P})$
(ii) $H(\mathcal{P}) = H_e(\mathcal{P}) \Leftrightarrow$ Kryptosystem absolut sicher

Sei A ein Alphabet, betrachte Klartextraum der Länge $n : \mathcal{P}_n = A^n$ mit zugehöriger Wahrscheinlichkeits-Verteilung P_n. Im Allgemeinen $P_n \neq P_1^n$

Def. 12: Die Entropie der 'Sprache' $L = (A^n)_{n \in \mathbb{N}}$ ist $H_L = \lim_{n \to \infty} \frac{H(\mathcal{P}_n, P_n)}{n}$

Falls $P_n = P_1^n \cdot H(\mathcal{P}_n, P_n) = n \cdot H(\mathcal{P}_1, P_1) \Rightarrow H_L = H(\mathcal{P}_1, P_1) \leq \log_2 |A|$. Konkret für die englische Sprache $H(\mathcal{P}_1, P_1) \approx 4,19 < \log_2 26 = 4,76$.
Experimentell: $1 \leq H_L \leq 1,5$

Def. 13: Die *Redundanz* einer "Sprache" L ist $R_L = 1 - \frac{H_L}{\log_2 |A|}$. Offensichtlich $R_L = 0$, wenn $P_n = P_1^n$ und $P_1(0) = \frac{1}{|A|}$ (Gleichverteilung). Für die englische Sprache ergibt sich $R_L \approx 0,75$.

Frage: Wieviel Text braucht man, um mit Statistik entschlüsseln zu können? Betrachte den zugehörigen Chiffreraum $e_n = e^n$. Schlüsselraum $\mathcal{K} \rightsquigarrow$ monoalphabetische Substitution mit gleichen Schlüssel für jeden Buchstaben. 40:00

Zu Chiffretext $c \in e_n$ setze $\mathcal{K}(c) = \{K \in \mathcal{K} | \exists p \in \mathcal{P}_n : P_n(p) > 0 \wedge E(K,p) = c\}$
$\rightsquigarrow$ die Menge der möglichen Schlüssel. Nur ein Schlüssel ist richtig $\Rightarrow$ erwartete Anzahl falscher Schlüssel $\overline{s}_n = \sum_{c \in e_n} P(\hat{c}\,(|\mathcal{K}(c)| - 1))$. Man nennt $n_0 = \min\{n | \overline{s}_n \approx 0\}$ den Eindeutigkeitsabstand des Systems.

Satz 14: $n_0 \approx \frac{\log_2 |\mathcal{K}|}{R_L \log_2 |A|}$ (für die englische Sprache gilt $n_0 \approx 25$)

51:00

3 Stromchiffren und Schieberegister

WIKI: *Stromverschlüsselung* (engl. stream cipher) ist ein kryptographischer Algorithmus, bei dem Bits des Klartextes mit den Bits eines Schlüsselstroms einzeln (zumeist XOR-) verknüpft werden. Der Schlüsselstrom ist eine pseudozufällige Bitfolge. Die meisten Stromchiffrierungen benutzen einen symmetrischen Schlüssel. Der Schlüssel bestimmt den Initialzustand des Systems.

Anwendung
Eine Stromverschlüsselung ist im Gegensatz zum Blockchiffre nicht darauf angewiesen, dass sich erst genug zu verschlüsselnde Daten angesammelt haben, bis sie die Größe für einen Eingabeblock eines Blockchiffres erreicht haben, sondern kann jedes Klartextbit sofort in ein chiffriertes Ausgabebit übersetzen.

Dieses Bit kann dann sofort über den unsicheren Kanal (unsicher im Sinn von abhörbar) zum Empfänger übertragen werden. Daher sind Stromchiffren besonders für Echtzeitübertragungen geeignet (zum Beispiel Mobilfunk)

Arbeitsweise
Eine synchrone Stromchiffrierung generiert den Schlüsselstrom unabhängig vom Klar- oder Schlüsseltext. Das Output Feedback Mode (OFB) von Blockchiffren ist ein Beispiel für eine synchrone Stromchiffrierung. Im Gegensatz dazu hängt bei einem selbstsynchronisierenden Stromchiffre der Schlüsselstrom von vorhergehenden verschlüsselten Bits ab. Ein Beispiel für ein selbstsychronisierendes Stromchiffreverfahren ist das Cipher Feedback Mode (CFB) von Blockchiffren.

Synchrone Stromchiffren werden oft als lineare Schieberegister mit Rückkopplung (Linear Feedback Shift Register, LFSR) konstruiert. LFSR können einfach in Hardware implementiert werden, sind schnell und produzieren Pseudozufallsfolgen mit guten statistischen Eigenschaften.

Ein LFSR ist im Prinzip ein Schieberegister der Länge n. Jedoch wird im normalen Betrieb in jedem Takt das freiwerdende Bit am Eingang des Schieberegisters mit dem Ergebnis der XOR-Verknüpfung zweier oder mehrerer anderer Stellen des Schieberegisters in Form einer Rückkopplung (engl. Feedback) gefüllt. Bei geeigneter Wahl der Abzweigungen ist eine maximale Periodenlänge von bis zu $2n-1$ möglich, wobei n die Anzahl der Bits des Schieberegisters ist. Zur Initialisierung kann das Schieberegister mit beliebigen Werten gefüllt werden, nicht jedoch nur mit Nullen.

Schieberegister mit Rückkopplung
Ein Schieberegister mit Rückkopplung besteht aus einem Schieberegister und einer Rückkopplungsfunktion. Das Schieberegister speichert eine Folge von n Bits. Die Rückkopplungsfunktion ist eine Abbildung von $\{0,1\}^n$ auf $\{0,1\}$. Wenn ein Bit ausgegeben werden soll, werden alle Bits im Schieberegister um einen Speicherplatz (sagen wir nach rechts) verschoben; das neue Bit am linken Ende des Schieberegisters wird abhängig von den anderen Bits berechnet. Dieser Vorgang zählt als ein Takt. Das einfachste Schieberegister mit Rückkopplung ist das lineare Schieberegister mit Rückkopplung. Die Rückkopplungsfunktion ist die XOR-Verknüpfung bestimmter Bits des Registers. Diese Bits werden durch das Rückkopplungspolynom festgelegt. Schieberegister mit nichtlinearer Rückkopplung verwenden nichtlineare boolesche Funktionen.

Als Schlüsselstromgenerator auf der Grundlage von LFSRs nimmt man eine oder mehrere LFSR, die meist unterschiedlich lang sind und unterschiedliche Rückkopplungspolynome haben. Damit kombiniert man LFSRs zu nichtlinearen Generatoren. Diese generieren einen Schlüsselstrom.

Zerlege Klartext $p \in \mathcal{P}$ in n-Teile, $p = (p_1, \ldots, p_n)$
und verschlüssle mit $K = (K_1, \ldots, K_n)$: $c = (c_1, \ldots, c_n)$ mit $c_i = E(K_i, P_i)$

(i) $K_i = K \forall i \rightsquigarrow$ Blockchiffren (Verschlüsselungsoperation für jeden Block diesselbe)

(ii) $K_i = K_i(K_1, \ldots, K_{i-1}, p_1, \ldots, p_{i-1}) \rightsquigarrow$ Stromchiffren (die Folge der Klartextzeichen wird nacheinander mit einer in jedem Schritt variierenden Funktion verschlüsselt).

Wir betrachten hier nur $K_i = K_i(K_1, \ldots, K_{i-1})$ mit (K_i) eine *Pseudozufallsfolge*. Dies ist eine deterministisch erzeugte Folge $(s_i)_{i \in \mathbb{N}_0}$ mit $s_i \in \mathbb{Z}_2$ die zufällig aussieht. Solche Folgen werden von endlichen Automaten erzeugt und sind daher periodisch. Nach einer Vorperiode l gibt es eine *Periode* $n : s_{i+N} = s_i \forall i \geq l \rightsquigarrow$ *Zyklus*$(s_i, s_{i+1}, \ldots, s_{i+N-1})$ für $i \geq l$.

3.1 Golombsche Axiome

Man stellt drei Forderungen an "gute" Pseudozufallsfolgen:

(i) In einem Zyklus unterscheidet sich die Anzahl der "1" von der der "0" höchsten um "1"

(ii) Ein "Run" ist eine Folge gleicher Werte. In einem Zyklus gibt es für $1 \leq i \leq \log_2 N$ ungefähr $N/2^i$ Runs der Länge i für jeden Wert

(iii) Betrachte für $\hat{c} = 0, 1, \ldots, N-1$
die Zyklen $(s_e, s_{e+1}, \ldots, s_{e+N-1})$ und $s^{(\tau)} = (s_{e+\tau}, s_{e+\tau+1}, \ldots, s_{\tau+e+N-1})$
setze $D(s) = d(s, s^{(\tau)}) = \#$ verschiedene Einträge von s und s^τ
$\rightsquigarrow$ Autokorrelationskoeffizienten $c(\tau) = \frac{N-2D(\tau)}{N} \Rightarrow c(0) = 1$ und $-1 \leq c(\tau) \leq 1$ es soll gelten $c(1) = c(2) = \ldots = c(n-1)$
Die Autokorrelationsfunktion $C(\tau)$ der Folge $s^{(\tau)}$ konstant für Werte von τ zwischen 1 und $n-1$

Wiki: *Autokorrelation* Vergleicht man eine geordnete Folge von Zufallsvariablen mit sich selbst, so spricht man von Autokorrelation. Da jede unverschobene Folge mit sich selbst am ähnlichsten ist, hat die Autokorrelation für die unverschobenen Folgen den höchsten Wert. Wenn zwischen den Gliedern der Folge eine Beziehung besteht, die mehr als zufällig ist, hat auch die Korrelation der ursprünglichen Folge mit der verschobenen Folge in der Regel einen Wert, der signifikant von Null abweicht. Man sagt dann, die Glieder der Folge sind autokorreliert.

Mit Hilfe der Autokorrelation ist es möglich, Zusammenhänge zwischen den beobachteten Ergebnissen zu verschiedenen Beobachtungszeitpunkten einer Messreihe festzustellen.

Bsp. 1: $S_{i+4} = S_i + S_{i+3} \rightsquigarrow$ Periode $N = 2^4 - 1 = 15 \Rightarrow$ Zyklus (1111 0101 1001 000)

Beweis

$i=1$	$8 \cdot 1$	$7 \cdot 0$	$(15/2^1 = 7,5)$
$i=2$	$4 \cdot 11$	$3 \cdot 00$	$(15/2^2 = 3,75)$
$i=3$	$2 \cdot 111$	$1 \cdot 000$	$(15/2^3 = 1,875)$
$i=4$	$1 \cdot 1111$	$0 \cdot 0000$	$(15/2^4 = 0,375)$

$\Rightarrow$ gibt $c(0) = 1, c(1) = c(2) = \ldots = c(14) = -1/15$ z. B.

$$s^{(8)} = (1100\,1000\,1111\,010) \rightsquigarrow D(8) = 8$$
$$s^{(14)} = (0111\,1010\,1100\,100) \rightsquigarrow D(14) = 8$$

05. Vorlesung (30. April 2008)

02:22

WIKI: *Rekurrenzrelation*
In mathematics, a recurrence relation is an equation that defines a sequence recursively: each term of the sequence is defined as a function of the preceding terms.

DEF. 2: Ein *lineares Schieberegister* der Länge n ist eine Rekurrenzrelation

$$S_{k+n} = C_0 S_k + C_1 S_{k+1} + \ldots + C_{n-1} S_{k+n-1} \forall k \in \mathbb{N}_0 \quad (*)$$

mit Systemkonstanten $C_0, \ldots, C_{n-1} \in \mathbb{Z}_2$ und Startwerte $S_0 \ldots, S_{n-1} \in \mathbb{Z}_2$

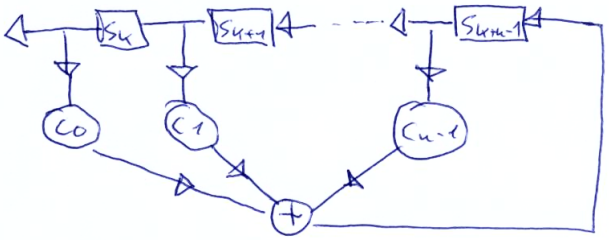


Abbildung 3: lineares Schieberegister

Ordne einer Folge $(S_i)_{i \in \mathbb{N}_0}$ die formale Potenzreihe
$s(x) = \sum_{i=0}^{\infty} s_i x^i \in \mathbb{Z}_2[[x]]$ zu $\rightsquigarrow$ *erzeugende Funktion* der Folge (s_i)

Erinnerung:

$$\left(\sum_{i=0}^{\infty} a_i x^i\right) + \left(\sum_{i=0}^{\infty} b_i x^i\right) = \sum_{i=0}^{\infty} (a_i + b_i)\, x^i$$

$$\left(\sum_{i=0}^{\infty} a_i x^i\right) \cdot \left(\sum_{i=0}^{\infty} b_i x^i\right) = \sum_{i=0}^{\infty} \left(\sum_{j=0}^{i} a_j + b_{i-j}\right) x^i$$

ANN.: Die Folge (s_i) habe Periode N und Vorperiode $l = 0$

$$\Leftrightarrow s(x) - x^N s(x) = s_0 + s_1 x + \ldots + s_{N-1} x^{N-1} \qquad |s(x) \cdot (1 - x^N)$$
$$\Leftrightarrow s(x) = \frac{s_0 + s_1 x + \ldots + s_{N-1} x^{N-1}}{1 - x^N} \in \mathbb{Z}_2(x)$$

DEF. 3: Das *Rückkopplungspolynom* des Schieberegisters ist

$$f = x^n - c_{n-1} x^{n-1} - c_{n-2} x^{n-2} - \ldots - c_1 x - c_0$$

Das zu f *reverse Polynom* (= charakteristische Polynom) ist

$$f^*(x) = f(1/x)x^n = 1 - c_{n-1} x - c_{n-2} x^2 - \ldots - c_1 x^{n-1} - c_0 x^n$$

BSP.: (bereits normiertes) Rückkopplungspolynom:

$$f(x) = 1x^3 + 1x^2[= c_2] + 0x^1[= c_1] + 1[= c_0]$$

reverses Polynom:

$$\begin{aligned} f*(x) &= 1 - c_2 x - c_1 x^2 - c_0 x^3 \\ &= 1 - 1x - 0x^2 - 1x^3 \\ &= 1 - x - x^3 \end{aligned}$$

SATZ 4: Es gibt ein Polynom $g \in \mathbb{Z}_2[x]$ mit $\deg g < \deg f$ oder $g = 0$, so dass $s = \frac{g}{f*}$

Beweis: —

WIKI: *Irreduzibilität*:

1. in der Algebra für Polynome, die sich nicht als Produkt zweier nichtkonstanter Polynome schreiben lassen (irreduzibles Polynom)
2. in der Algebra allgemeiner für Elemente von Ringen, die sich nicht als Produkt zweier Nichteinheiten schreiben lassen (irreduzibles Element)

SATZ 5: Es sei $f \in \mathbb{Z}_2[x]$ irreduzibel und $s(x) \neq 0$. Die Folge (s_i) ist genau dann periodisch mit Periode N, wenn N die kleinste Zahl ist, so dass $f | x^N - 1$.

Beweis: —

WIKI: *prime Restklassengruppe*
Die Menge aller invertierbaren Elemente in der multiplikativen Gruppe der Restklasse mod n ist die prime Restklassengruppe. Sie wird mit $(\mathbb{Z}/n\mathbb{Z})^\times$ oder $\mathbb{Z}_n^*$ symbolisiert.

Eine Primitivwurzel mod n ist dadurch charakterisiert, dass sie die prime Restklassengruppe mod n erzeugt.

Berechnung der inversen Elemente
Zu jeder primen Restklasse $a + n\mathbb{Z}$ existiert eine prime Restklasse $b + n\mathbb{Z}$, sodass gilt: $ab \equiv 1 \pmod{n}$

Die prime Restklassengruppe $b+n\mathbb{Z}$ ist also das inverse Element zu $a+n\mathbb{Z}$ bezüglich der Multiplikation in der primen Restklassengruppe $(\mathbb{Z}/n\mathbb{Z})^\times$. Ein Repräsentant von $b+n\mathbb{Z}$ lässt sich mit Hilfe des Erweiterten Euklidischen Algorithmus bestimmen. Indem man die obige Kongruenz umschreibt zu $ab+kn=1$ sieht man, dass der Erweiterte Euklidische Algorithmus auf a und n angewandt mit b einen Repräsentanten von $b+n\mathbb{Z}$ berechnet.

Primitivwurzel -> erzeugendes Element einer Gruppe
Die Primitivwurzel ist ein Begriff aus dem mathematischen Teilgebiet Zahlentheorie. Es handelt sich dabei um ein ausgezeichnetes Element einer primen Restklassengruppe. Die besondere Eigenschaft einer Primitivwurzel ist, dass jedes Element der Restklassengruppe als Potenz der Primitivwurzel dargestellt werden kann. Beispielsweise ist die 3 eine Primitivwurzel modulo 7, da gilt:

$$\begin{aligned}
3^1 &\equiv 3 \pmod 7\\
3^2 &\equiv 2 \pmod 7\\
3^3 &\equiv 6 \pmod 7\\
3^4 &\equiv 4 \pmod 7\\
3^5 &\equiv 5 \pmod 7\\
3^6 &\equiv 1 \pmod 7
\end{aligned}$$

Es lassen sich also alle Elemente $1, 2, \ldots, 6$ der primen Restklassengruppe modulo 7 als Potenzen von 3 darstellen.

primitv
Nullstellen besitzen maximale Ordnung

Ordnung
heißt wir suchen das Kleinste mit dem es funktioniert. Bsp.: $\alpha = 2$, dann K finden, das möglichst klein ist im Körper $\mathbb{F}_2$. $2^1 = 2 \Rightarrow$ also 1 die kleinste Ordnung von 2 (Anmerkung: 2^1 gibt $\mathbb{F}_2 = 1$)

$\mathbb{Z}/p\mathbb{Z}_p$
$\mathbb{Z}/p\mathbb{Z}_p$ ist ein Körper wenn $p \in \mathbb{P}$ Primzahl. Körper heißt es existiert ein mulitplikatives Invereses. Wir haben p-Elemente und rechnen $\mod p$, damit wir den Körper nicht verlassen.

Bem. 6: Einiges Material zu endlichen Körpern:

(i) $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}_p$ ist für jede Primzahl p ein Körper mit p' Elementen
$\rightsquigarrow \mathbb{F}_p = \mathbb{Z}_p$ ($\mathbb{F}$ = finite Field)
Bis auf Isomorphie sind die einzigen weiteren endlichen Körper von der Form $\mathbb{F}_q$ mit $q = p^n$ für eine Primzahl p und eine ganze Zahl $n \in \mathbb{N}$.
Dabei gilt $\mathbb{F}_q \supseteq \mathbb{F}_p$ und $\mathbb{F}_q$ hat q Elemente, nämlich die Nullstellen von $x^q - x \in \mathbb{F}_p[x]$. Alternativ kann ein Körper mit $q = p^n$ Elementen als Quotient $\mathbb{F}_p[x]/ < f >$ mit $f \in \mathbb{F}_p[x]$ irreduzibel und $\deg f = n$ erzeugt werden

(ii) Für $f, g \in \mathbb{F}_q$ gilt: $(f+g)^p = f^p + g^p$

BEWEIS:

$$(f+g)^p = \sum_{k=0}^{p} \binom{p}{k} f^k g^{p-k}$$

$$\binom{p}{k} = \frac{p!}{k!(p-k)!} \Rightarrow \quad \text{für} \quad k \neq 0, p \quad \text{gilt} \quad p \Big| \binom{p}{k}$$

(iii) Sei $f \in \mathbb{F}_p[x], \alpha \in \mathbb{F}_q$ Nullstelle von f
$\Rightarrow \forall j \geq 0 : \alpha^{p^j}$ ebenfalls Nullstelle von f

BEWEIS: Sei $f = \sum_{i=0}^{n} e_i x^i$ mit $c_i \in \mathbb{F}_p$ (d. h. $c_i^p = c_i$)

$$\text{also} \quad 0 = \sum_{i=0}^{n} c_i \alpha^i = \left(\sum_{i=0}^{n} c_i \alpha^i \right)^p = \sum_{i=0}^{n} c_i^p \alpha^{ip}$$
$$= \sum_{i=0}^{n} c_i^p (\alpha^p)^i \Rightarrow \alpha^p \quad \text{Nullstelle von} \quad f$$

$\rightsquigarrow$ Iteration $(\alpha^p)^p = \alpha^{p^2}$ Nullstellen von f usw.

06. Vorlesung (07. Mai 2008)

BEM. 6: Fortsetzung:

(iv) Sei $\alpha \in \mathbb{F}_q / \{0\}$. Dann heißt $ord(\alpha) = \min\{K | \alpha^K = 1\}$ die *Ordnung* von α.

BEH.: Sei $f \in \mathbb{F}_q[x]$ irreduzibel von Grad n. Dann haben alle Nullstellen von f in $\mathbb{F}_{p^n}$ die gleiche Ordnung.

BEW.: Sei α Nullstelle mit $ord(\alpha) = e$
$\Rightarrow 1 = \alpha^e = (\alpha)^e = (\alpha^p)^e$
$\Rightarrow ord(\alpha^e) \leq e = ord(\alpha)$
$\Rightarrow e = ord(\alpha) \geq ord(\alpha^p) \geq ord(\alpha^{p^2}) \geq \ldots \geq ord(\alpha^{p^n}) = ord(\alpha) = e$
$\Rightarrow \forall_j : ord(\alpha^{pj}) = e$

(v) Für alle $d \geq 1$ ist $x^{q^d} - x \in \mathbb{F}_q[x]$ das Produkt aller *normierten*[8] irreduziblen Polynome in $\mathbb{F}_q[x]$, deren Grad d teilt (ohne Beweis).
Also:
$f \in \mathbb{F}_q[x]$ irreduzibel von Grad n $\Rightarrow f | x^{p^n} - x \Rightarrow f | x^{p^n - 1} - 1$
Sei $\alpha \in \mathbb{F}_{\mathbb{P}}^n$ Nullstelle von f mit $ord(\alpha) = e$
$\Leftrightarrow$ alle Nullstellen von f haben Ordnung e
$\Leftrightarrow$ jede Nullstelle von f auch Nullstelle von $x^e - 1$
$\Leftrightarrow$ ($\mathbb{F}_{p^n}$ Zerfällungskörper von f) $f | x^e - 1$. f heißt *primitiv*, falls $e = p^n - 1$ (d. h. maximal mögliche Ordnung)

SATZ 7: Das Schieberegister $(*)$ habe ein irreduzibles Rückkopplungspolynom $f \in \mathbb{F}_i[x]$ vom Grad n. Dann ist die Folge $(s_i)_{i \in \mathbb{N}^0}$ periodisch mit einer Periode $N \leq 2^n - 1$. Die Periode ist genau dann $N = 2^n - 1$, wenn f primitiv ist.

[8] normiert = Leitkoeffizient gleich 1

BEWEIS: Bemerkung 6.5 $\Rightarrow f | x^{2^n-1} - 1 \overset{Satz5}{=} (s_i)$ periodisch
mit Periode $N \leq 2^n - 1$
Bem. 6(v) & Satz 5. $\Rightarrow$ Periode von (s_i) gegeben durch Ordnung der Nullstellen von f

BSP. 8: (i) $f = x^3 + x^2 + 1 \in \mathbb{F}_2[x]$ irreduzibel (also keine Faktoren über $\mathbb{F}_2$)
(da $f(0) = f(1) = 1 \Rightarrow$ kein Linearfaktor über $\mathbb{F}_2$ möglich)
f auch primitiv, da $2^3 - 1 = 7$ Primzahl
zugehörige Schieberegister: $S_{k+3} = S_{k+2} + S_k$
wähle Anfangswert: $s_0 = s_1 = s_2 = 1$

$\rightsquigarrow$	1	1	1	("Speicherzustand")
	$\leftarrow$	$\leftarrow$	$\leftarrow$	
einmal schieben	1	1	0	
	1	0	1	
	0	1	0	
	1	0	0	
	0	0	1	
7mal schieben	0	1	1	
	(1	1	1)	

also Zyklus der Länge 7 (1110 100)

(ii) $f = x^4 + x^3 + x^2 + 1 \in \mathbb{F}_2[x]$ irreduzibel
f nicht primitv, da $x^5 - 1 = (x - 1)(x^4 + x^3 + x^2 + 1)$
(Prinzip der geometrischen Summe)
Schieberegister: $S_{k+4} = S_{k+3} + S_{k+2} + S_{k+1} + S_k$
mit Anfangswert: $s_0 = s_1 = s_2 = s_3 = 1$

$\rightsquigarrow$	1	1	1	1
	1	1	1	0
	1	1	0	1
	1	0	1	1
	0	1	1	1
	(1	1	1	1)

also Periode $N = 5$, Zyklus (11110)

SATZ 9: Wenn die Folge $(s_i)_{i \in \mathbb{N}_0}$ durch ein lineares Schieberegister mit primitiven Rückkopplungspolynom erzeugt wird, dann genügt sie den Golombschen Forderungen.

BEWEIS: (Idee)
identifiziere Speicherzustand mit den Elementen von $\mathbb{F}_{2^n}$ f primitiv
$\Rightarrow$ jedes Element aus $\mathbb{F}_{2^n}^x$ kommt genau einmal in einer Periode vor.
$\Rightarrow$ wir haben 2^{n-1} mal "1" und $2^{n-1} - 1$ mal "0" $\Rightarrow$ Golomb 1 erfüllt
aufwändiges Zählen beweist die anderen Golomb-Forderungen.

Diser Satz besagt, dass Folgen, die mittels primitiver Polynome erzeugt wurden (sogenannte *maximale Folgen* oder *M-Folgen*) statistisch gesehen gut sind. Kryptographisch gesehen sind sie aber schlecht, da bei einer Periodenlänge von $N = 2^n - 1$

bereits $2n \approx 2 \cdot \log_2 N$-Werte ausreichen, um das Bildungsgesetz zu rekonstruieren.
Elemtarer Ansatz: *lineare Algebra* (Gauß-Verfahren Komplexität $O(n^3)$). Fasse (*) als Matrixmulitplikation auf.

$$s_{n+1} = \begin{pmatrix} s_k & s_{k+1} & \dots & s_{k+n+1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix}$$

Annahme: Wir kennen $s_0, s_1, \dots, s_{2n-1} \Rightarrow$ LGS für $c_0, \dots, c_{n-1}$

$$\begin{pmatrix} s_0 & s_1 & \dots & s_{n-1} \\ s_1 & s_2 & \dots & s_n \\ \vdots & \vdots & & \vdots \\ s_{n-1} & s_n & \dots & s_{2n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n+1} \end{pmatrix} = \begin{pmatrix} s_n \\ s_{n+1} \\ \vdots \\ s_{2n-1} \end{pmatrix}$$

Dieses LGS lässt sich in $O(n^3)$ Schritten lösen.
Beachte: wir setzen die Kenntnis von n voraus.
Annahme: n sei nicht bekannt; wahre Länge des Registers sei $\tilde{n}$
Falls $\tilde{n} < n \rightsquigarrow$ die ersten $\tilde{n} + 1$ Spalten sind linear abhängig
Falls $\tilde{n} > n \rightsquigarrow$ LGS unlösbar $\Rightarrow$ versuche größeren Wert

07. Vorlesung (14. Mai 2008)

3.2 Berlekamp-Massey-Algorithmus

WIKI: Der Berlekamp-Massey-Algorithmus dient dazu, das kürzeste, lineare rückgekoppelte Schieberegister zu finden, das eine gegebene Folge von Symbolen ausgibt.

Effizienteres Verfahren: *Berlekamp-Massey-Algorithmus* (Komplexität: $O(n^2)$)

Gegeben: Folge $(s_i)_{i \in \mathbb{N}_0}$ mit $(s_i) \in \mathbb{Z}$
Idee: beschreibe endliche, immer länger werdende Segmente $(s_0, s_1, \dots, s_r)$ durch Polynome.
Wir berechnen für jedes $r \in \mathbb{N}$ ein normiertes Rückkopplungspolynom f_r mit minimalem Grad L_r:
$f_r = x^{L_r} + \sum_{j=0}^{L_r - 1} c_j^{(r)} x^j$
dass $(s_0, \dots, s_r)$ in dem Sinne erzeugt, dass gilt:
$c_0^{(r)} s_k + c_1^{(r)} s_{k+1} + \dots + c_{L_r-1}^{(r)} s_{k+L_r-1} + s_{k+L_r} = 0$ für alle $0 \leq k \leq r - 1 - L_r$. $L_r =$ *lineare Komplexität* von $(s_0, s_1, \dots, s_{r-1})$ im Allgemeinen wächst L_r mit r bei Folgen, die mit einem linearen Schieberegister erzeugt wurden, wird L_r (und auch f_r) irgendwann konstant.

WIKI: *Ziel* des Berlekamp-Massey Algorithmus ist es also, zu einer gegebenen Folge $(s_0, s_1, \dots, s_{n-1})$ von Symbolen das Rückkopplungspolynom $C(x)$ und die Länge L des erzeugenden Schieberegisters zu finden.

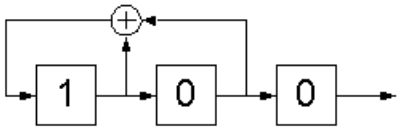


Abbildung 4: Berlekamp-Massey Algorithmus

Beispiel
Der Berlekamp-Massey Algorithmus ermittelt zur Symbolsequenz (0 0 1 1 0 1 1) das kürzeste, linear rückgekoppelte Schieberegister, welches diese Sequenz ausgibt:

Das Rückkopplungspolynom lautet für dieses Beispiel $C(x) = 1 + x + x^2$ und die Länge des Schieberegisters ist $L = 3$.

SATZ 10: Sei $(s_i)_{i \in \mathbb{N}_0}$ eine Folge über $\mathbb{Z}_2$. Wir setzen $f_0 = 1, L_0 = 0$. Dann existiert für jedes $r \in \mathbb{N}$ ein Polynom $f_r = x^{L_r} + \sum_{j=0}^{L_r - 1} c_j^{(r)} x^j$, das explizit induktiv erzeugt werden kann.
Wenn wir einfügen:

$$\delta_r = s_{r-1} + \sum_{j=0}^{L_{r-1}-1} c_j^{(r-1)} s_{r-1-L_{r-1}+j}$$

dann gilt $L_r = L_{r-1}$ falls $s_r = 0$ und $L_r = \max\{L_{r-1}, r - L_{r-1}\}$ sonst.

Beweisskizze Konstruiere f_r induktiv: $f_0 = 1, L_0 = 0$

$$\begin{aligned}
r = 1: \quad & \delta_1 = 0 \rightsquigarrow f_1 = f_0, L_1 = L_0 \\
& \delta_1 \neq 0 \rightsquigarrow f_1 = x, L_1 = 1 \\
r \to r+1: \quad & \delta_{r+1} = 0 \rightsquigarrow f_{r+1} = f_r, L_{r+1} = L_r \\
& \delta_{r+1} \neq 0 \rightsquigarrow \text{technisch} \quad f_{r+1} = f_r \cdot x^{r+1-m+L_{m-1}-L_r} + f_{m-1}{}^*
\end{aligned}$$

* = für ein gewisses m (technisch!)
(beachte: $\delta_{r+1} = 0$ heißt, dass das nächste Folgeglied von f_r richtig vorhergesagt wurde)

Falls (s_i) durch ein lineares Schieberegister der Länge n erzeugt wurde, dann liefert f_{2n} das zugehörige Rückkopplungspolynom.

Zur Abwehr solcher Angriffe versucht man, die lineare Komplexität zu erhöhen, ohne aber die Komplexität des Verfahrens stark zu erhöhen. Eine Möglichkeit besteht in der nicht-linearen Kombination von Schieberegistern. Allerdings wird dann die Analyse schwierig.

BSP. 11: *Generator von Geffe*
Nehme drei lineare Schieberegister $(s_i^{(0)}), (s_i^{(1)}), (s_i^{(2)})$

bilde $s_i = s_i^{(0)} \cdot s_i^{(1)} + s_i^{(1)} \cdot s_i^{(2)} + s_i^{(2)}$
Für Maximalfolgen mit linearer Komplexität $n_0 = 19, n_1 = 20, n_2 = 21$ kann man dann eine Komplexität von $19 \cdot 20 + 20 \cdot 21 + 21 = 821$ erreichen statt maximal 60 mit linearen Kombinationen. -> danach ist Schieberegister nicht mehr linear und kann nicht mehr so leicht mit Gauß gelöst werden

4 Symmetrische Kryptoverfahren

ERINNERUNG: Symmetrisch heißt Ver- und Entschlüsselung hängen eng zusammen und benutzen im wesentlichen denselben Schlüssel.

In der Praxis benutzt man als Schlüsselraum (einen Teilraum) von $\mathbb{Z}_2^k$ für eine hinreichend große *Schlüssellänge* k.

Verschlüsselung: $E : \mathbb{Z}_2^K \times \mathbb{Z}_2^t \to \mathbb{Z}_2^t$ mit *Blocklänge* t.

Bei einem vernünftigen Verfahren muss:
(i) Schlüssellänge k so groß sein, dass systematisches Durchprobieren für den zu erwartenden Gegner nicht realistisch ist.
(ii) Es darf keine nennenswert schnellere Attacke geben.

Nach Shannon sollten dazu (zusätzlich) zwei Techniken eingesetzt werden.
(i) *Diffusion*: Redundanz des Klartexts soll möglichst weiträumig über den Chiffretext verteilt werden. (z. B. durch Permutation oder Aufblähungen)
Ziel: jedes Klartextbit soll möglichst viele Chiffretextbits beeinflussen. Regelmäßigeiten verteilt, schwerer Aufzufinden.
(ii) *Konfusion*: Zusammenhang zwischen Schlüssel und Chiffretext möglichst undurchsichtig (im Idealfall perfekte Sicherheit). Verschleierung des Zusammenhangs zwischen Klartext und Geheimtext mit einfachen Regeln.

4.1 Feistel-Chiffren

Bsp. 1: *Feistel-Chiffren* (IBM, 1960er Jahre)
- operieren in mehreren Runden
- in jeder Runde wird nur ein Teil des Blocks und ein Teil des Schlüssels benutzt.
- sei $t = 2n, l$ Länge des Rundenschlüssels $\rightsquigarrow$ Feistel-Funktion $f : \mathbb{Z}_2^l \times \mathbb{Z}_2^n \to \mathbb{Z}_2^n$
- sei $p_i = (L_i, R_i) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^n$ der Chiffretext in Runde i (d. h. Klartext in Runde 1)
  $\rightsquigarrow P_{i+1} = (R_i, f(s_i, R_i) + L_i)$ mit s_i der i-te Rundenschlüssel (f erzeugt Konfusion, das Vertauschen von L, R plus die Addition führt zur Diffusion).

WIKI: Feistel-Chiffren sind eine allgemeine Struktur mit der Blockchiffren realisiert werden können. Die Feistelchiffre war später dann die Grundlage für den DES-Algorithmus.

Viele moderne symmetrische Verschlüsselungsalgorithmen basieren auf Feistelchiffren. Dies rührt daher, dass Blockverschlüsselungen, welche auf Feistelchiffren basieren, garantiert umkehrbar sind. Damit ist die notwendige Grundbedingung für Blockchiffren

erfüllt, dass es bei der Abbildung von Chiffreblöcken auf Klartextblöcke bei der Entschlüsselung zu keinen Mehrdeutigkeiten kommen darf. Weiterhin wurde diese Struktur von sehr vielen Kryptografen analysiert und für gut befunden.

Arbeitsweise
Wie es der Name "Blockchiffre" schon nahelegt, wird der Klartext zuerst in einzelne Blöcke zerlegt. Die Größe dieser Blöcke kann frei gewählt werden, üblich sind oftmals Vielfache von 64 Bit. Jeder dieser Blöcke wird danach in zwei gleichgroße Hälften (L_0 und R_0) geteilt und in n Runden mit verschiedenen Schlüsseln verschlüsselt. Nach den Runden werden die Hälften wieder zusammengesetzt.

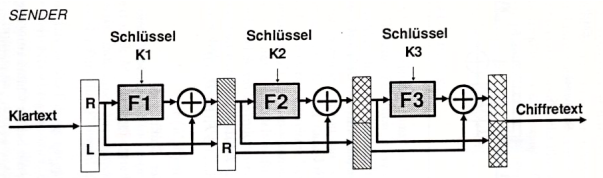


Abbildung 5: Feistel-Chiffren mit 3 Verschlüsselungsschritten

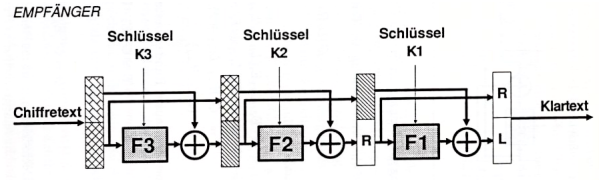


Abbildung 6: Feistel-Chiffren mit 3 Entschlüsselungsschritten

4.2 Data Encryption Standard (DES)

- 1977 in den USA vom National Bureau of Standards eingeführt mit geplanter Lebensdauer von 10 Jahren
- Design-Kriterien lange geheimgehalten
- bis heute keine Attacke bekannt, die wesentlich schneller als systematisches Durchprobieren ist. (heute aber praktisch möglich)
- Schlüssellänge $k = 56$ (mit Prüfziffern $k = 64$), Blocklänge $t = 64$
- Feistel Chiffre mit 16 Runden
  Feistel-Funktion benutzt Rundenschlüssel der Länge 48
  - benutzt Funktion $F : \mathbb{Z}_2^{48} \to \mathbb{Z}_2^{32}$, die aus 8 verschiedenen Funktionen ("S-Basen" $\mathbb{F}_2^{6^2} \to \mathbb{F}_2^4$) zusammengesetzt wird; für $f : \mathbb{Z}_2^{48} \to \mathbb{Z}_2^{32}$ wird Text R_i zunächst auf Länge 48 aufgebläht und zu Schlüssel addiert, Ergebnis wird in F einsetzt.
- Triple DES: verschlüssele mit Schlüssel 1,
  entschlüssele mit Schlüssel 2,
  verschlüssele nochmal mit Schlüssel 1.

08. Vorlesung (21. Mai 2008)

4.3 Advanced Encryption Standard (AES)

- ein symmetrisches Kryptosystem
- seit 2001: *Advanced Encryption Standard*
- nach öffentlicher Ausschreibung 1997 und Diskussion auf 3 Konferenzen
- entwickelt von 2 Postdocs der Uni Leuven
  $\rightsquigarrow$ *Rijndael-Chiffre* (gesprochen wie dt. "Reyndahl")
- Block- und Schlüssellänge können jeden durch 32 teilbaren Wert zwischen 128 und 256 annehmen (AES: $t = 128, K = 128, 192, 256$); benutzt Arithmetik in $\mathbb{F}_{256}$; wieder rundenbasiert (aber keine Feistel-Chiffre) 14 Runden mit Bytesubsitutionen, Zeilenshifts, Spaltenmischen in Matrizen und Additionen von Rundenschlüssel; kann sehr effizient implementiert werden; erhoffte Lebensdauer 20-30 Jahre

WIKI: Rijndael ist, wie bereits erwähnt, ein Blockchiffre. Bei Rijndael können Blocklänge und Schlüssellänge unabhängig voneinander die Werte 128, 160, 192, 224 oder 256 Bits erhalten, während bei AES die Einschränkung der festgelegten Blockgröße von 128 Bit und der Schlüsselgröße von 128, 192 oder 256 Bit gilt. Jeder Block wird zunächst in eine zweidimensionale Tabelle mit vier Zeilen geschrieben, deren Zellen ein Byte groß sind. Die Anzahl der Spalten variiert somit je nach Blockgröße von 4 (128 Bits) bis 8 (256 Bits). Jeder Block wird nun nacheinander bestimmten Transformationen unterzogen. Aber anstatt jeden Block einmal mit dem Schlüssel zu verschlüsseln, wendet Rijndael verschiedene Teile des erweiterten Originalschlüssels nacheinander auf den Klartext-Block an. Die Anzahl r dieser Runden variiert und ist von der Schlüssellänge k und Blockgröße b abhängig (beim AES also nur von der Schlüssellänge).

4.4 Betriebsarten symmetrischer Verfahren

DES oder AES beschreibt die Verschlüsselung *eines* Blocks. Wie verschlüsselt man längere Texte?

ANN.: symmetrisches Verfahren $C = E(K, P), P = D(K, C)$
Klartext zerlegt in Blöcke: $(P_1, P_2, \ldots, P_r)$ mit $P_i \in \mathbb{Z}_2^t$
gesucht: Chiffretext $(C_1, C_2, \ldots, C_r)$ mit $C_i \in \mathbb{Z}_2^t$

4.4.1 Electronic Code Book (ECB) Mode

$C_i = E(K, P_i)$

WIKI: ist eine unsichere Betriebsart für Blockverschlüsselungen wie es z. B. der Advanced Encryption Standard (AES) darstellt.

ECB ist der einfachste und zugleich unsicherste Modus, denn dabei werden die Klartextblöcke nacheinander und unabhängig voneinander in den Geheimtextblock überführt. Dies birgt große Gefahren, denn dadurch werden Klartextmuster nicht verwischt. Gleiche Klartextblöcke ergeben bei gleichen Schlüssel auch immer den gleichen Geheimtextblock, wodurch man bei hinreichend vielen Geheimtextblöcken und partiellen Annahmen über den Klartext Rückschlüsse auf den geheimen Schlüssel ziehen kann. Der Name des Modus rührt daher, dass Codebücher über die Zuordnung von Chiffretexten und Klartexten erstellt werden können.

Man kann Geheimtextblöcke austauschen, wodurch sich zum Beispiel die Summe oder der Empfänger einer Überweisung ändern könnte. Deswegen sollte man lieber den CBC-Modus benutzen, denn dieser ist viel sicherer als der ECB-Modus.

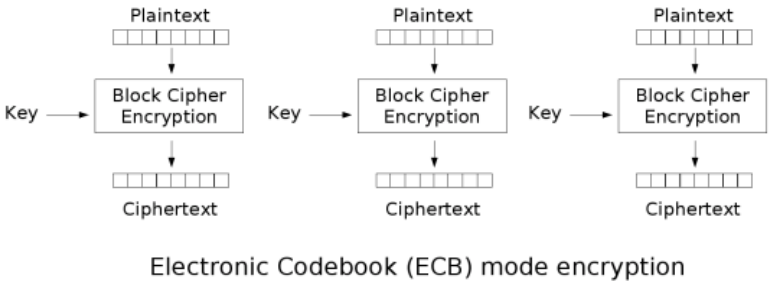


Abbildung 7: ECB Encryption

Schwächen:
- gleiche Klartextblöcke führen stehts zu gleichen Chiffretextblöcken (Regelmäßigkeit und Strukturen bleiben erhalten).
- Angreifer kann unbemerkt Chiffretextblöcke einschleusen oder deren Reihenfolge verändern.

4.4.2 Cipher Block Chaining (CBC) Mode

Wiki: Cipher Block Chaining Mode (CBC) ist eine Betriebsart, in der Blockchiffre betrieben werden können. Vor dem Verschlüsseln eines Klartextblocks wird dieser erst mit dem im letzten Schritt erzeugten Geheimtextblock per XOR (exklusives Oder) verknüpft.

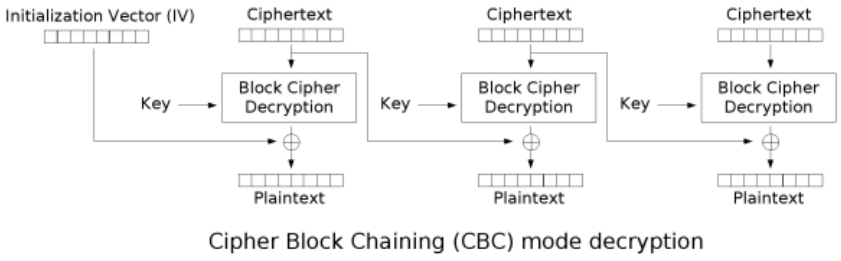


Abbildung 8: CBC Decryption

Der CBC-Mode hat einige wichtige Vorteile:

1. Klartextmuster werden zerstört.
2. Jeder Geheimtextblock hängt von allen vorherigen Klartextblöcken ab.
3. Identische Klartextblöcke ergeben unterschiedliche Geheimtexte.
4. Verschiedene Angriffe (Time-Memory-Tradeoff und Klartextangriffe) werden erschwert.

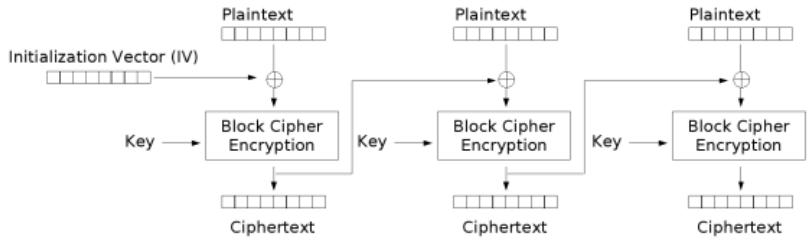


Abbildung 9: CBC Encryption

Da ein Geheimtextblock nur von dem vorherigen Block abhängt, verursacht ein beschädigter Geheimtextblock, wie beispielsweise ein Bitfehler bei der Datenübertragung, beim Entschlüsseln keinen allzugroßen Schaden, denn es werden nur der betroffene Klartextblock zu 50% und im darauffolgende Klartextblock ein Bit falsch im Klartext dechiffriert. Dies ist unmittelbar aus der Definition der Entschlüsselung und obiger Abbildung ersichtlich, da ein beschädigter Geheimtextblock C_i nur die Klartextblöcke P_i und $P_i + 1$ beeinflusst und sich nicht unbeschränkt weiter verbreitet. Trotzdem kann diese beschränkte Vervielfachung nur eines einzigen Bitfehlers im Chiffrat bei CBC eine Vorwärtsfehlerkorrektur des Klartextes erschweren bzw. unmöglichen machen. Genauso verursacht ein beschädigter Initialisierungsvektor beim Entschlüsseln keinen allzugroßen Schaden, da dadurch nur der Klartextblock P_1 beschädigt wird.

Der CBC-Modus ist wesentlich sicherer als der ECB-Modus, vor allem wenn man keine zufälligen Texte hat. Unsere Sprache und andere Dateien, wie z. B. Video-Dateien, sind keinesfalls zufällig, weswegen der ECB-Mode im Gegensatz zum CBC-Mode Gefahren birgt. Generell sollte ein Blockchiffre immer im CBC-Modus betrieben werden - Ausnahmen sollten gut begründet sein.

setze $C_i = E(K, P_i + C_{i-1})$,
d. h. C_i hängt von C_{i-1} ab $\rightsquigarrow$ Regelmäßigkeiten im Klartext werden zerstört.
Zur Berechnung von C_1 wird (zufällig) ein Block $C_o \in \mathbb{Z}_2^t$ gewählt und mit ECB übertragen.
Entschlüsselung: $P_i = D(K, C_i) + C_{i-1}$

Beachte: Übertragungsfehler im Block C_i beeinflusst nur P_i und P_{i+1};
bereits $P_{i+2} = D(K, C_{i+2}) + C_{i+1}$ ist unabhängig von C_i

4.4.3 Cipher Feedback (CFB) Mode

Wiki: Cipher Feedback Mode (CFB) ist eine Betriebsart (Modus), in der Blockchiffren betrieben werden, damit Klartexte verschlüsselt werden können, die länger als die Blocklänge des Chiffrierverfahrens sind. Beispiele für Blockchiffre sind der Data Encryption Standard (64 Bit) oder der Advanced Encryption Standard (128 Bit, 192 Bit, 256 Bit). Der CFB kann auch als Stromchiffre verwendet werden. In diesem Modus wird, wie in der Abbildung dargestellt, die Ausgabe der Blockchiffre mit dem Klartext bitweise XOR (exklusives ODER) verknüpft um daraus den Geheimtext zu bilden. Diese Betriebsart

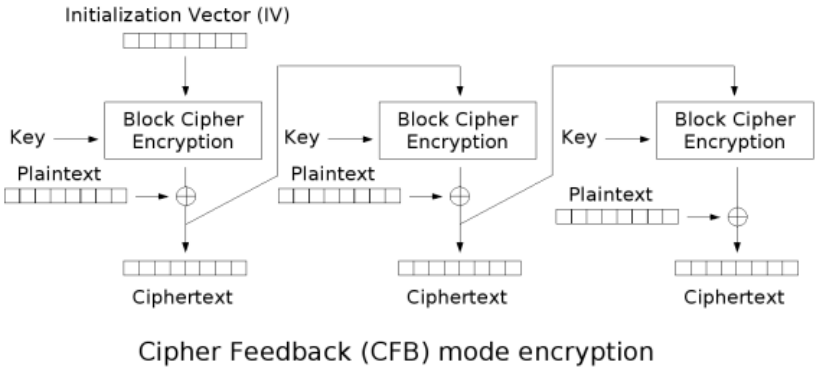


Abbildung 10: CFB Encryption

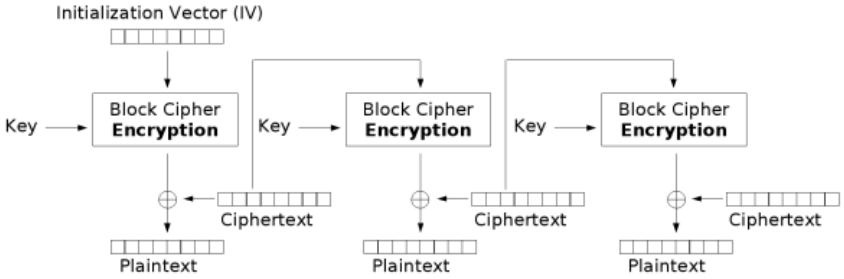


Abbildung 11: CFB Decryption

bzw. dieser Modus ergibt damit eine Stromchiffre. Die ausgegebenen Geheimtextdaten fließen als Eingabe in den nächsten Block zur Verschlüsselung.

Damit ergibt sich als wesentlicher Unterschied zu dem Output Feedback Mode (OFB) eine Selbstsynchronisation. Dies bedeutet, dass der Empfänger bei der Entschlüsselung nicht den genauen Inhalt (inneren Zustand) der Blockchiffre kennen muss bzw. durch geeignete, zusätzliche Übertragungsverfahren im Rahmen der Übertragungsprotokolle mitgeteilt bekommen muss.

Der Initialisierungsvektor (IV) dient ähnlich wie bei dem Cipher Block Chaining (CBC) als Startwert.

Die Entschlüsselung beim Empfänger, wie in obiger Abbildung dargestellt, funktioniert wie Verschlüsselung, erzeugt also bei gleichem Initialisierungsvektor und gleichem Schlüssel die gleiche binäre Datenfolge mit der die XOR-Operation des Sender rückgängig gemacht werden kann. Die Grafik zeigt auch den wesentlichen Nachteil dieser Stromchiffre: Durch nur einen einzigen Bitfehler der bei der Übertragung auftreten kann, wird im aktuellen Klartextdatenblock genau ein Bitfehler erzeugt und zusätzlich im nachfolgenden Datenblock im Mittel 50% der Datenbits zerstört. Diese Fehlerfortpflanzung ist ähnlich wie bei der Betriebsart Cipher Block Chaining (CBC)

und erschwert die Entschlüsselung des Klartextes.

Trotz des Vorteils der Selbstsynchronisation wird der CFB in der Praxis nur selten eingesetzt: Spielt die Fehlerfortpflanzung auf den nächsten Block in einer bestimmten Anwendung keine Rolle bzw. wird durch geeignete zusätzliche Verfahren kompensiert, kommt meist der CBC zur Anwendung. Wird eine Stromchiffre ohne Fehlerfortpflanzung in einer Anwendung benötigt, kommt meist der Modus OFB zu Anwendung.

ANN.: Daten fallen in kleineren Blöcken als t an $\rightsquigarrow$ "wahre" Blocklänge $r \leq t$(oft $r = 8$), d. h. Klartext ist Folge $(P_1, \ldots, P_n)$ mit $P_i \in \mathbb{Z}_2^r$

Wähle Initialisierungsvektor $V_1 \in \mathbb{Z}_2^t$

- *Verschlüsselung:*
  $O_i = E(K, V_i)$
  $T_i =$ erste r Bits von O_i (d. h. $T_i \in \mathbb{Z}_2^r$)
  $C_i = P_i + T_i$
  $V_{i+1} = V_i \cdot 2^r + C_i \mod 2^t$ ($\rightsquigarrow$ in V_i die ersten r Bits löschen und C_i hinten angehängen)
  $\rightsquigarrow$ Schlüsseltext $(C_1, \ldots, C_n)$ mit $C_i \in \mathbb{Z}_2^r$
- *Entschlüsselung:* (benutze dasselbe V_1)
  $O_i = E(K, V_i)$ (nicht D!)
  $T_i =$ erste r Bits von O_i
  $P_i = C_i + T_i$
  $V_{i+1} = V_i 2^r + C_i \mod 2^t$
  $\rightsquigarrow$ Klartext $(P_1, P_2, \ldots, P_n)$

5 Asymmetrische Kryptoverfahren

Grundidee: (Diffie/Hellman 1976)

- völlige Trennung von Entschlüsselung und Verschlüsselung: Sender einer verschlüsselten Nachricht muss diese nicht selbst entschlüsseln können
- Empfänger publiziert *öffentlichen Schlüssel*, mit dem Nachrichten an ihn verschlüsselt werden können. Entschlüsselung erfordert *geheimen Schlüssel*.
- Verschlüsselung erfordert "*Einwegfunktion*": bijektive Abbildung, die leicht berechenbar ist, deren Umkehrabbildung aber nicht mit vertretbarem Aufwand gefunden werden kann.
- Entschlüsselung benötigt "*Falltür*": mit geheimen Schlüssel kann Umkehrung leicht berechnet werden.

WIKI: Der Vorteil asymmetrischer Verfahren besteht in der Vereinfachung der Schlüsselverteilung. Bei symmetrischen Verfahren ist der Austausch eines geheimen Schlüssels nötig, den Sender und Empfänger gemeinsam benutzen. Der Austausch muss dabei sicher erfolgen. Dies bedeutet, der Austausch muss abhörsicher sein. Es ist jedoch auch sicherzustellen, dass der Schlüssel tatsächlich von der Person stammt, mit der geheime Nachrichten ausgetauscht werden sollen.

Durch Public-Key-Systeme entfällt die Notwendigkeit, den Schlüsselaustausch gegen Abhören zu härten, da nur der öffentliche Schlüssel ausgetauscht werden muss. Es bleibt aber ein wesentliches Problem, da weiterhin sicherzustellen ist, dass der öffentliche

Schlüssel tatsächlich von der Person stammt, mit der geheime Botschaften ausgetauscht werden sollen. In der Praxis ist dies oft nur dann möglich, wenn zugleich auch die Geheimhaltung gewährleistet werden kann.

Falls die Kommunikation zwar nicht abhörsicher ist, aber kein Zweifel über die Identität des Partners besteht und die Nachricht nicht von einem Dritten verändert werden kann, ist der Schlüsselaustausch beim Public-Key-Verfahren problemlos. Allerdings ist dies auch mit anderen Verfahren möglich.

5.1 Zahlentheoretische Grundlagen

RSA basiert auf der Funktion: $\phi : \mathbb{Z}_n \to \mathbb{Z}_n, x \mapsto x^e \mod n$

Repeated Squaring:
Schreibe e als Binärzahl: $e = \sum_{i=0}^{k} e_i 2^i$ mit $e_i \in \{0,1\} \rightsquigarrow x^e = x^{\sum e_i 2^i} = \prod_{i=0}^{k} (x^{2^i})^{e_i} = \prod_{i=0, e_i=1}^{k} x^{2^i}$
berechne x^{2^i} als $(x^{2^{i-1}})^2$.

Bsp. 1: berechne $6^{73} \mod 100 \rightsquigarrow 73 = 1 + 2^3 + 2^6$ (Binärentwicklung des Exponenten)
wiederholtes Quadrieren:
$6^1 = 6$,
$6^2 = 36$,
$6^{2^2} = 36^2 \equiv -4 \mod 100$,
$6^{2^3} = (-4)^2 \equiv 16 \mod 100$,
$6^{2^4} = 16^2 \equiv 56 \mod 100$,
$6^{2^5} = 56^2 \equiv 36 \mod 100$,
$6^{2^6} = 36^2 \equiv -4 \mod 100$,
$\Rightarrow 6^{73} \equiv 6 \cdot 16 \cdot (-4) \equiv 16 \mod 100$.

Lemma 2: Sei e eine Zahl mit k-Bit. Dann benötigt die Berechnung von $x^e \mod n$ höchsten n Quadrierungen und Multiplikationen *modulo* n.

09. Vorlesung (28. Mai 2008)

5.1.1 kleiner Fermat

Satz 3: ("*kleiner Fermat*").
Sei $a \in \mathbb{Z}$ und $p \in \mathbb{P}$ eine Primzahl. Dann gilt: $a^p \equiv a \mod p$
Wenn a nicht durch p teilbar ist, gilt $a^{p-1} \equiv 1 \mod p$.

Wiki: "der kleine Fermat" macht eine Aussage über die Eigenschaften von Primzahlen. Der Satz beschreibt die allgemein gültige Kongruenz: $a^p \equiv a \mod p$, wobei a eine ganze Zahl und p eine Primzahl ist. Falls a kein Vielfaches von p ist, kann man das Resultat in die häufig benutzte Form $a^{p-1} \equiv 1 \mod p$ bringen.

Beweis: —

Hinweis: brauchen wir für Primzahltests und für RSA

5.1.2 Erweiterter Euklidischer Algorithmus (EEA)

SATZ 4: ("*Bézaut*" (Erweiterter Euklidischer Algorithmus (EEA))).
Sei $m, n \in \mathbb{N}$ und $d = ggT(m, n)$. Dann gilt: $s, t \in \mathbb{Z}$ mit $d = sm + tn$ (s, t heißen Bézaut-Koeffizienten und sind nicht eindeutig).

ALGORITHMUS: ("*Erweiterter Euklidischer Algorithmus (EEA)*").
$r_0 \leftarrow m; r_1 \leftarrow n; s_0 \leftarrow 1; s_1 \leftarrow 0; t_0 \leftarrow 0; t_1 \leftarrow 1; i \leftarrow 1$
repeat
$r_{i-1} = q_i r_i + r_{i+1}$ (Division mit Rest)
$s_{i+1} \leftarrow s_{i-1} - q_i s_i$
$t_{i+1} \leftarrow t_{i-1} - q_i t_i$
$i \leftarrow i + 1$
until $r_i = 0$
return $(r_{i-1}, s_{i-1}, t_{i-1})$
in jedem Schritt gilt:
$r_i = s_i m + t_i n$
$r_i = 0 \Rightarrow r_{i-1} = ggT(m, n)$

5.2 Das RSA-Verfahren

WIKI: RSA ist ein asymmetrisches Kryptosystem, das sowohl zur Verschlüsselung als auch zur digitalen Signatur verwendet werden kann. Es verwendet ein Schlüsselpaar bestehend aus einem privaten Schlüssel, der zum Entschlüsseln oder Signieren von Daten verwendet wird, und einem öffentlichen Schlüssel, mit dem man verschlüsselt oder Signaturen prüft. Der private Schlüssel wird geheim gehalten und kann nicht oder nur mit extrem hohem Aufwand aus dem öffentlichen Schlüssel berechnet werden. RSA ist nach seinen Erfindern Rivest, Shamir und Adleman benannt.

- wähle zufällig zwei Primzahlen $p \neq q$ und bestimme $n = pq$ (*RSA-Modul*)
- wähle eine Zahl $e \in \mathbb{N}$ mit $1 < e < (p-1)(q-1)$ und $ggT(e, (p-1)(q-1)) = 1$ (*Verschlüsselungsexponent*)
- berechne $d \in \mathbb{N}$ mit $1 < d < (p-1)(q-1)$ und $de \equiv 1 \mod (p-1)(q-1)$ (*Entschlüsselungsexponent*) (d existiert nach Satz 4 und lässt sich mit EEA berechnen)
- öffentlicher Schlüssel: (n, e)
  geheimer Schlüssel: d
- (p, q) müssen ebenfalls geheimgehalten werden, am besten werden sie nach der Berechnung von n, e, d weggeworfen)
- Klartext $0 \leq m < n$ wird verschlüsselt als $c = m^e \mod n$
- Chiffretext $0 \leq c < n$ wird verschlüsselt als $m = c^d \mod n$

SATZ 5: Sei (n, e) ein öffentlicher RSA-Schlüssel und d der zugehörige geheime Schlüssel.
Dann gilt: $(m^e)^d \mod n = m$

Beweis: —

BSP. 6.: Sei $p = 11, q = 23 \Rightarrow n = 253; (p-1)(q-1) = 220$
wähle $e = 3 \Rightarrow d = 147$
Klartext $m = 165$

Chiffretext $c = 165^3 \mod 253 = 110$
Entschlüsselung: $110^{147} \mod 253 = 165$

5.3 Sicherheit von RSA

Wir zeigen nun, dass die Kenntnis von e und d erlaubt, n zu Faktorisieren. Da Faktorisieren als schwierig gilt, gilt damit RSA als sicher.

Sei $x \in \mathbb{N}$ mit $ggT(x,n) = 1 \rightsquigarrow ord_n(x) = \min\{t \in \mathbb{N} | x^t \equiv 1 \mod n\}$
Setze $s = \max\{t \in \mathbb{N} | 2^t | ed-1\}$; $k = (ed-1)/2^s$ (ungerade)

LEMMA 7: Sei $ggT(x,n) = 1$. Dann gilt $ord_n(x^k) \in \{2^i | 0 \leq i \leq s\}$

BEWEIS: Satz 5 $\Rightarrow x^{ed-1} \equiv 1 \mod n \Rightarrow x^{k2^s} \equiv 1 \mod n \Rightarrow ord_n(x^k) | 2^s$
Nach Beweis von Satz 5 gilt dieselbe Aussage auch wenn n durch p oder q ersetzt wird.

SATZ 6: Sei $ggT(x,n) = 1$
Falls $ord_p(x^k) \neq ord_q(x^k)$,
so gilt $1 < ggT(x^{2^t k} - 1, n) < n$ für ein $t \in \{0, 1, 2, \ldots, s-1\}$.

BEWEIS: Lemma 7 $\Rightarrow \exists 0 \leq s_1, s_2 \leq s : ord_p(x^k) = 2^{s_1} \wedge ord_q(x^k) = 2^{s_2}$
o.B.d.A. sei $s_1 > s_2$ (d.h. $s_2 < s_1$). $\Rightarrow x^{2^{s_2} k} \equiv 1 \mod q \wedge x^{2^{s_2} k} \not\equiv 1 \mod p$
$n = pq \Rightarrow ggT(x^{2^{s_2} k} - 1, n) = q$
$ord_p(x) = \min\{t \in \mathbb{N} | x^t \equiv \mod p\}$ für Primzahl $p \in \mathcal{P} \rightsquigarrow ord_p(x)$ die Ordnung von $[x]$ in $(\mathbb{Z}_p^x, \cdot)$

$$ggT(x, \underset{n=pq}{n}) = 1 \Rightarrow p \nmid x \wedge q \nmid x$$
$$\Rightarrow [x]_p \neq [0]_p \wedge [x]_q \neq [0]_q$$

10. Vorlesung (04. Juli 2008)

SATZ 9: Die Anzahl der Elemente $x \in \{1, 2, \ldots, n-1\}$ mit $ord_p(x^k) \neq ord_q(x^k)$ ist mindestens $\frac{1}{2}(p-1)(q-1)$

BEWEIS:
- *Vorbemerkung*: für jede Primzahl p ist $\mathbb{Z}_p^x = \mathbb{Z}_p\{0\}$ eine zyklische Gruppe, d. h. es existiert eine *Primitivwurzel* $g \mod p$ mit $\mathbb{Z}_p^x = g, g^2, \ldots, g^{p-1} \equiv 1 \mod p$; wenn p, q zwei verschiedene Primzahlen sind, gibt es auch ein g, dass sowohl $\mod p$ als auch $\mod q$ eine Primzahl ist.
- *Lemma 7* $\Rightarrow ord_p(g^k) = 2^i \wedge ord_p(g^k) = 2^j$ für $i, j \in \{0, 1, \ldots, s\}$
  g simultane Primwurzel $\mod g$ bzw. $\mod q \Rightarrow x \equiv g^a \mod p \wedge x \equiv g^b \mod q$ für geeignete $0 \leq a \leq -2, 0 \leq b \leq q \cdot 2$
  also: $x^k \equiv (g^k)^a \mod p, x^k \equiv (g^k)^b \mod q$
  Lemma 7 $\Rightarrow ord_p(g^k) = 2^{i'} \wedge ord_p(g^k) = 2^{j'}$ für $i', j' \in \{0, 1, \ldots, s\}$

  a ungerade $\Rightarrow i' = i \quad (ggT(a, 2^i) = 1)$
  a gerade $\Rightarrow i' < i \quad (2 | ggT(a, 2^i))$

analog für b

- *Fallunterscheidung*
  - I > J: wähle x so, dass a ungerade, b beliebig $\Rightarrow ord_p(x^k) = 2^i > 2^j \geq ord_q(g^k)$
    mindestens $\frac{1}{2}(p-1)(q-1)$ Möglichkeiten
  - I < J: analog (a beliebig, b ungerade)
  - I = J: wähle a gerade, b ungerade oder a ungerade, b gerade $\Rightarrow ord_p(x^k) \neq ord_q(x^k)$
    mindestens $\frac{1}{2}(p-1)(q-1)$ Möglichkeiten

5.4 Weitere Anwendungen von RSA

(i) Identitätsnachweis

ZIEL: A möchte B beweisen, dass er wirklich A ist.

VORGEHENSWEISE:
- B besorgt sich öffentlichen Schlüssel (n, e) von A
- B schickt A eine Zufallszahl $1 < x < n$ an A
- A berechnet mit seinem geheimen Schlüssel d die Zahl $y = x^d \mod n$ und schickt y an B
- B überprüft, ob $y^e \equiv m \mod n$

BEM.: Falls A, B auch über RSA kommunizieren wollen, benötigen sie dafür ein weiteres Schlüsselpaar. Sonst kann Angreifer A einen Chiffreblock als angebliche Zufallszahl X schicken und erhält als Klartext y zurück.

(ii) Elektronische Unterschrift

ZIEL: B möchte C beweisen, dass er die Nachricht a von A erhalten hat.

VORGEHENSWEISE:
- A berechnet mit geheimen Schlüssel d die Zahl $u = a^d \mod n$
- A schickt an B das Paar (a, u)
- C überprüft ob $a \equiv u^e \mod n$ ist

(iii) Elektronisches Bargeld

ZIEL: sicheres und anonymes Bezahlen im Internet

VORGEHENSWEISE:
- Ausgehende Bank erzeugt für jede Stückelung einen öffentlichen Schlüssel (n, e)
- Kunde erzeugt unauffällig eine Seriennummer $1 < m < n$.
  $n > 10^{150}$ festlegen, dass m eine 150-stellige Zahl ist, die spiegelsymmetrisch ist
- Kunde erzeugt weitere Zufallszahl r (teilerfremd zu n) und schickt $m \cdot r^e \mod n$ an Bank
- Bank schickt Unterschift u für diese Zahl zurück (und belastet Konto des Kunden),
  d. h. $v = (m \cdot r^e)^d \mod n = m^d \cdot r \mod n$
- Kunde berechnet durch Division $r \mod n$ die Zahl $v = m^d \mod n$
- Kunde bezahlt durch Angabe von v
- Zahlungsempfänger berechnet $v^e \mod n$; er akzeptiert die Zahlung falls Ergebnis eine gültige Seriennummer ist.

- Zahlungsempfänger meldet Seriennummer an Bank
- Falls Seriennummer nicht bereits registriert, schreibt Bank Geldbetrag gut und registriert Seriennummer

11. Vorlesung (11. Juli 2008)

6 Der diskrete Logarithmus

WIKI: In der Gruppentheorie ist der diskrete Logarithmus das Analogon zum gewöhnlichen Logarithmus aus der Analysis; diskret kann in diesem Zusammenhang etwa wie ganzzahlig verstanden werden. Die diskrete Exponentiation in einer zyklischen Gruppe bildet eine Umkehrfunktion des diskreten Logarithmus. Als Vergleich: die stetige Exponentialfunktion ist eine Umkehrfunktion des gewöhnlichen Logarithmus.

Als ein Beispiel diene das Rechnen modulo n. Der diskrete Logarithmus ist hier die kleinste Lösung x der Gleichung $a^x \equiv m \mod p$ bei gegebenen natürlichen Zahlen m, a und der Primzahl p.

Da sich die diskrete Exponentiation leicht (im Sinne der Komplexitätstheorie) berechnen lässt, während für die Umkehrfunktion, den diskreten Logarithmus, meist nur schwere (im Sinne der Komplexitätstheorie) Algorithmen bekannt sind, eignet sich die diskrete Exponentiation als Einwegfunktion in der Kryptografie.

Anwendungsbeispiele sind u. a.

- Diffie-Hellman-Schlüsselaustausch
- Elgamal-Kryptosystem

Sei G eine zyklische Gruppe der Ordnung n,
d. h. $\exists g \in G : G = \{g, g^2, g^3, \ldots, g^n\} \Rightarrow G \cong \mathbb{Z}_n, g^m \mapsto [m]$

EINFACH: gegeben m, berechne g^m

SCHWIERIG: gegeben $h \in G$, finde m mit $h = g^m \rightsquigarrow m$ *diskrete Logarithmus* von h.

6.1 Schlüsselaustausch nach Diffie / Hellman

WIKI: Der Diffie-Hellman-Schlüsselaustausch ist ein Protokoll aus dem Bereich der Kryptografie. Mit ihm erzeugen zwei Kommunikationspartner einen geheimen Schlüssel, den nur diese beiden kennen. Dieser Schlüssel wird üblicherweise verwendet, um verschlüsselte Nachrichten mittels eines symmetrischen Kryptosystems zu übertragen.

Beim Diffie-Hellman-Schlüsselaustausch senden sich beide Kommunikationspartner über einen unsicheren Kanal jeweils eine Nachricht zu. Das Problem, aus diesen beiden Nachrichten den geheimen Schlüssel zu berechnen, wird als Diffie-Hellman-Problem bezeichnet. Von diesem nimmt man an, dass es praktisch nicht lösbar ist. Deshalb kann jemand, der beide Nachrichten mithört, daraus im Allgemeinen nicht den geheimen Schlüssel berechnen. Der Diffie-Hellman-Schlüsselaustausch ist jedoch nicht mehr

sicher, wenn sich ein Angreifer zwischen die beiden Kommunikationspartner schalten und Nachrichten verändern kann. Diese Lücke schließen Protokolle wie das Station-to-Station-Protokoll, indem sie zusätzlich digitale Signaturen und Message Authentication Codes verwenden.

ZIEL: A und B wollen sich über eine offene Leitung auf eine gemeinsame Zahl K zu einigen.

VORGEHENSWEISE:
- A, B einigen sich offen auf eine Primzahl $p \in \mathbb{P}$ und eine natürliche Zahl $a \in \mathbb{N}$.
- A wählt eine Zufallszahl $k < p$ und schickt B die Zahl $u = a^k \mod p$
- B wählt eine Zufallszahl $e < p$ und schickt A die Zahl $v = a^e \mod p$
- A berechnet $K = v^k \mod p = a^{ke} \mod p$
- B berechnet $K = u^e \mod p = a^{ke} \mod p$

BEM.: Durch abhören der Leitung kann ein Gegner die Zahlen a, p, u, v in Erfahrung bringen.
$\rightsquigarrow$ er kennt alle Zahlen $a^{\alpha k+\beta e} = u^\alpha v^\beta$. Ohne diskrete Logarithmen kann er aber nicht a^{ke} bestimmen.
Man in the middle Attack

ANN.: C hat die Kontrolle über einen Knoten, über diesen die Kommunikation von A, B läuft.

- C hört Vereinbarung von a, p ab
- C wählt eigene Zufallszahlen $\overline{k}, \overline{e} < p$
- C fängt $u = a^k \mod p$ ab und schickt B $\overline{u} = a^{\overline{k}} \mod p$
- C hat nun gemeinsamen Schlüssel $K_A = a^{k\overline{e}} \mod p$ mit A bzw. $K_B = a^{k\overline{e}} \mod p$ mit B
- C fängt Kommunikation von A ab, entschlüsselt sie mit K_A, schickt sie anschließend mit K_B verschlüsselt an B weiter (und umgekehrt)

6.2 Verschlüsselung nach ElGamal

WIKI: Das Elgamal-Kryptosystem (auch al-Dschamal-Kryptosystem) ist ein Schema zur Verschlüsselung, das auf dem mathematischen Problem des diskreten Logarithmus beruht. Elgamal ist ein asymmetrischer Verschlüsselungsalgorithmus aufbauend auf der Idee des Diffie-Hellman-Algorithmus, der mit diesen diskreten Logarithmen arbeitet.

VORGEHENSWEISE:
- a, p sind entweder allgemein bekannte Systemparameter oder Teil des öffentlichen Schlüssels jedes Teilnehmers
- A wählt eine Zufallszahl k und veröffentlich $u = a^k \mod p$
- B möchte Nachricht $(m_1, \ldots, m_r)$ an A schicken
- B erzeugt für jeden Block m_i eine Zufallszahl e_i und berechnet $v_i = a^{e_i} \mod p$ und $c_i = u^{e_i} m_i \mod p$
- B schickt Folge $((v_1, c_1), \ldots, (v_r, c_r))$ an A
- A berechnet $v_i^k \equiv a^{e_i k} \equiv u^{e_i} \mod p$ und dann $m_i \equiv c_i u^{-e_i} \mod p$

BEM.:
- wie bei RSA Sicherheit unbewiesen

- wie bei RSA gibt es Fallen bei der Parameterwahl
- wesentlicher Nachteil: Verdopplung des Datenvolumens
- wie RSA kann auch ElGamal für weitere Anwendungen wie digitale Unterschrift eingesetzt werden

6.3 Berechnung diskreter Logarithmen

Es sind bisher keine schnellen Algorithmen zur Berechnung des diskreten Logarithmus bekannt. Deren Laufzeit verhielte sich polynomial zur Länge der Eingabe. Es gibt aber Algorithmen, die die Lösung gezielter finden als bloßes Ausprobieren. Aufgrund des angesprochenen Laufzeitverhaltens und den in der Kryptografie üblichen Größenordnungen (mehrere Hundert Dezimalstellen in Numerus und Basis) spielen sie praktisch aber keine Rolle. Zu den bekanntesten Algorithmen zählen:

(i) *Probieren*
- berechne zu $a \in G$ nacheinander $g, g^2, g^3, \ldots$ bis $g^m = a$ gilt
- im Schnitt benötigt man n/2 Schritte ($n = |G|$) <= Ordnung von G

(ii) *Baby / Giant Steps*
Auf Kosten eines erhöhten Speicherbedarfs, wird die Rechenzeit auf $O(\sqrt{n})$ reduziert.
- wir wählen zunächst eine Zahl $m \in \mathbb{N}$ mit $m \approx \sqrt{n} \log_2 n$ (genaue Kenntnis von n nicht so wichtig, Größenordnung sollte ungefähr stimmen)
- Vorberechnung: berechne $g, g^2, g^3, \ldots, g^m$ und speichere $\rightsquigarrow m$ "baby steps"
- zu jedem $a \in G$ für das der diskrete Logarithmus bestimmt werden soll (also $l \in \mathbb{N}$ mit $a = g^l$), werden nacheinander Werte $a/(g^m)^j$ für $j = 0, 1, 2, \ldots$ berechnet und dann mit den vorberechneten Werten verglichen ("giant steps"). Der Vergleich kann z. B. als binäre Suche oder über eine Hash-Tabelle realisiert werden mit einem Aufwand $O(\log_2 n)$
- Wenn der Quotient mit g^i übereinstimmt, dann gilt $l = j \cdot m + i$
(j = giant steps, i = baby steps, l = gesuchter Exponent)
- Im schlimmsten Fall benötigen wir $n/m + \sqrt{n}$ giant steps (im Mittel wird die Hälfte benötigt) also hat Verfahren Aufwand $O(\sqrt{n})$

(iii) *Indexkalkül*
- wähle eine Schranke B und setze $F(B) = \{q \in \mathbb{P} | q \leq B\}$ (*Faktorbasis*)
$m \in \mathbb{N}$ heißt *B-glatt*, wenn alle Primfaktoren von m in $F(B)$ liegen
- Vorberechung: berechne den diskreten Logarithmus aller Elemente von $F(B)$, d. h. gesucht sind Werte $x(q) \in \mathbb{N} : g^{x(q)} \equiv q \mod p$
  - wähle zufällig Elemente $1 \leq z < p$
  - berechne $g^z \mod p$ und prüfe, ob Ergebnis B-glatt
  - wenn ja, berechne $g^z \mod p = \sum_{q \in F(B)} q^{t(q,z)}$ "Relation"
    $\Rightarrow g^z \equiv \sum_{q \in F(B)} q^{t(q,z)} \equiv \sum_{q \in F(B)} g^{x(q) f(q,z)}$
    $\Rightarrow g^z \equiv \sum_{q \in F(B)} x(q) f(q,z) \mod p - 1$ (wegen kleinen Fermat)
  - fahre fort, bis $n = |F(B)|$ Relationen gefunden werden
  - (12. Vorlesung 25. Juni 2008)
  - gegeben a, berechne diskreten Logarithmus $dlog(a) = x$, wähle zufällig Exponenten $1 \leq y < p$ bis $ag^y \mod p$ B-glatt.

$$\Rightarrow ag^y \equiv \prod_{q\in F(B)} q^{e(q)} \equiv \prod_{q\in F(B)} g^{x(q)e(q)} \mod p$$

$$\Rightarrow x = \left(-y + \sum_{q\in F(B)} x(q)e(q)\right) \mod (p-1)$$

12. Vorlesung (18. Juni 2008)

08:33

BSP.: Sei $p = 2027, g = 2, B = 11 \rightsquigarrow F(B) = \{2, 3, 5, 7, 11\}$

- wir wählen zufällig $z = 293, 983, 1318, 1593, 1918$

$$\begin{aligned}
\rightsquigarrow 2^{293} &\equiv 63 = 3^2 \cdot 7 \mod p \\
2^{983} &\equiv 385 = 5 \cdot 7 \cdot 11 \mod p \\
2^{1318} &\equiv 1408 = 2^7 \cdot 11 \mod p \\
2^{1593} &\equiv 33 = 3 \cdot 11 \mod p \\
2^{1918} &\equiv 1600 = 2^6 \cdot 5^2 \mod p
\end{aligned}$$

WIKI: *Kongruenz* Man nennt zwei Zahlen kongruent bezüglich eines Moduls (eine weitere Zahl), wenn sie bei Division durch den Modul denselben Rest haben. (Stimmen die Reste nicht überein, so nennt man die Zahlen inkongruent bzgl. des Moduls.)

Relationen liefern Kongruenzsysteme

$$\begin{aligned}
2 \cdot x(3) + x(7) &\equiv 293 \mod p-1 \\
x(5) + x(7) + x(11) &\equiv 983 \mod p-1 \\
7 \cdot x(2) + x(11) &\equiv 1318 \mod p-1 \\
x(3) + x(11) &\equiv 1593 \mod p-1 \\
6 \cdot x(2) + 2 \cdot x(5) &\equiv 1918 \mod p-1
\end{aligned}$$

Es gilt $p - 1 = 2026 = 2 \cdot 1013$ 20:04
$\rightsquigarrow$ betrachte das System einmal mod 2 und einmal mod 1013
$g = 2 \in F(B) \Rightarrow x(2) = 1$

$$\begin{array}{lll}
\underline{\text{mod 2}} & x(7) & \equiv 1 \mod 2 \\
& x(5) + x(7) + x(11) & \equiv 1 \mod 2 \\
& x(2) + x(11) & \equiv 0 \mod 2 \\
& x(3) + x(11) & \equiv 1 \mod 2
\end{array}$$

Inspektion $\rightsquigarrow x(5) \equiv x(7) \equiv x(11) \equiv 1 \mod 2, x(3) \equiv 0 \mod 2$

$$\begin{array}{lll}
\underline{\text{mod 1013}} & 2x(3) + x(7) & \equiv 293 \mod 1013 \\
& x(5) + x(7) + x(11) & \equiv 983 \mod 1013 \\
& 7x(2) + x(11) & \equiv 305 \mod 1013 \\
& x(3) + x(11) & \equiv 580 \mod 1013 \\
& 6x(2) + 2x(5) & \equiv 905 \mod 1013
\end{array}$$

Inspektion $\rightsquigarrow x(3) \equiv 282 \mod 1013, x(5) \equiv 956 \mod 1013$,
$x(7) \equiv 742 \mod 1013, x(3) \equiv 298 \mod 1013$

Chinesischer Restesatz:
$x(2) = 1, x(3) = 282, x(5) = 1969, x(7) = 1755, x(11) = 1311$
Berechnung von $dlog(13) = x$, wähle zufällig $y = 1397$
$\rightsquigarrow 13 \cdot 2^{1397} \equiv 110 = 2 \cdot 5 \cdot 11 \mod p$
$\Rightarrow x = (-1397 + 1 \cdot 1 + 1969 \cdot 1 + 1311 \cdot 1) \mod p - 1 = 1884$
also gilt $2^{1884} \equiv 13 \mod p$

Bem.: Komplexitätsanalyse ergibt eine *subexponentielle* Laufzeit
(schlechter als polynomial, besser als exponentiell)
Genauer: Laufzeit in $O(e^{c(\log p)^{1/2}(\log\log p)^{1/2}})$
(polynomial entspräche Exponenten 0,1;
exponentiell entspräche Exponenten 1,0)

46:39

7 Primzahltests

Bem. 1: Sei $\Pi(x) = \#\{p \in \mathbb{P} | p \leq x\}$
Sylvester 1892: $0,95695\frac{x}{\ln x} < \Pi(x) < 1,04423\frac{x}{\ln x}$
Primzahlsatz: $\lim\limits_{x\to\infty} \frac{\Pi(x)}{x/\ln x} = 1$
Interpretation: zwischen zwei Primzahlen der Größenordnung N liegen *im Mittel* $\ln N$ zusammengesetzte Zahlen (für $N = 2^{1027} \rightarrow \ln N \approx 710$). Die tatsächliche Verteilung schwankt stark: in jeder betrachteten Größenordnung hat man bisher Primzahlzwillinge gefunden (d. h. $p, p+2 \in \mathbb{P}$); aber auch längere Primzahl-Intervalle als $\ln N$.
Satz von Betrand: zwischen $N > 1$ und $2N$ liegt mindestens eine Primzahl.

64:12

7.1 Sieb des Eratosthenes

Wiki: Das Sieb des Eratosthenes ist eigentlich ein Algorithmus um eine Liste von Primzahlen zu erzeugen. Da diese Liste bis zu einer frei wählbaren Grenze alle Primzahlen enthält, kann sie für einen Primzahltest verwendet werden. Man überprüft dazu lediglich, ob die übergegebene Zahl in der Liste ist.

- schreibe alle Zahlen von 2 bis N in eine Liste
- für $i = 2, 3, \ldots, \sqrt{N}$ streiche, wenn i noch in der Liste steht alle Vielfachen von i
- verbleibende Zahlen sind alle Primzahlen $\leq N$
  Für große N ist dieses Verfahren praktisch untauglich.
  Anwendung: betrachte Suchintervalle $[N, N + e]$, sei $p \in \mathbb{P}$ eine (kleine) Primzahl $< e$
  $\rightsquigarrow N + p - (N \mod p)$ erste durch p teilbare Zahl im Suchintervall; streiche alle weiteren Vielfachen von p.
  $\rightsquigarrow$ relativ billige Elimination vieler zusammengesetzter Zahlen im Suchbereich.

7.2 Fermat-Test

WIKI: Mit dem fermatschen Primzahltest kann man Primzahlen von zusammengesetzten Zahlen unterscheiden. Der Test erhält eine Zahl n und eine Basis a als Eingabe. n muss eine ungerade Zahl > 3 sein. Außerdem muss a die Bedingung $1 < a < n - 1$ erfüllen. Der Test liefert eines von zwei möglichen Ergebnissen: entweder

- "n ist (bezüglich a) Primzahlkandidat (engl: probable prime)" oder
- "n ist keine Primzahl".

Im letzteren Fall ist n keine Primzahl. Falls der Test n zum Primzahlkandidaten erklärt, kann man in den meisten Fällen davon ausgehen, dass n eine Primzahl ist. n ist also "wahrscheinlich" eine Primzahl. Daher wird dieses Verfahren auch PRP-Test (= probable prime test) genannt.

KOROLLAR 2: Falls für eine natürliche Zahl $1 \leq a < p$ gilt $a^{p-1} \not\equiv 1 \mod p$, so ist p keine Primzahl

BEWEIS: kleiner Fermat

BEM. 3: Die Umkehrung gilt nicht! Selbst wenn für alle $1 \leq a < p$ gilt $a^{p-1} \equiv 1 \mod p$, muss p keine Primzahl sein $\rightsquigarrow$ *Carmichael-Zahlen* (es gibt natürlich unendlich viele solcher Zahlen)

WIKI: *Carmichael-Zahl*
Eine Carmichael-Zahl n ist pseudoprim zu allen Basen, die keine gemeinsamen Primfaktoren mit n haben. Jede Carmichael-Zahl ist das Produkt aus mindestens 3 Primzahlen.

Eine zusammengesetzte natürliche Zahl q heißt Carmichael-Zahl, falls für alle zu q teilerfremden Zahlen a gilt: $a^{q-1} \equiv 1 \mod q, \quad \forall a \in \mathbb{Z}_q^*$

Beispiel
$561 = 3 * 11 * 17$ ist die kleinste Carmichael-Zahl. Für alle Basen a, die keinen Primfaktor mit 561 gemeinsam haben, gilt: $a^{560} \equiv 1 \mod 561$

561 ist durch 3, 11, 17, 33, 51 und 187 teilbar.
Für diese Teiler gilt $a^{q-1} \equiv 1 \mod q$ nicht!
$3^{560} \equiv 375 \mod 561$
$11^{560} \equiv 154 \mod 561$
$17^{560} \equiv 34 \mod 561$

Für große p ist es aber sehr unwahrscheinlich, dass sie auch nur für ein a den Fermat-Test bestehen, ohne tatsächlich eine Primzahl zu sein.
Kim & Pomerance geben folgende Oberschranken für die Fehlerwahrscheinlichkeit an:

p	10^{60}	10^{80}	10^{100}	10^{200}	10^{500}
$\sum$	$8 \cdot 10^{-2}$	$9 \cdot 10^{-5}$	$3 \cdot 10^{-8}$	$4 \cdot 10^{-25}$	$3 \cdot 10^{-55}$

7.3 $(p-1)$-Test

SATZ 4: Sei $N \in \mathbb{N}$ ungerade mit $N \geq 3$. Es gelte die Faktorisierung $N-1 = \prod_i P_i^k$.
N ist genau dann eine Primzahl, wenn es ein $a \in \mathbb{N}$ gibt mit $a^{N-1} \equiv 1 \mod N$ und $a^{(N-1)/p_i} \not\equiv 1 \mod N$ für alle i.

BEWEIS: (Skizze)
"$\Rightarrow$" N Primzahl $\Rightarrow$ wähle für a eine Primitivwurzel $\mod N$
"$\Leftarrow$" $ord(a) = N-1 \rightsquigarrow$ nur möglich wenn $N \in \mathbb{P}$

BEM 5.: Für $N \in \mathbb{P}$ gibt es $(N-1)\prod_i(1-1/p_i)$ Primitivwurzeln $\mod N$
$\rightsquigarrow$ Abschätzung für Fehlerwahrscheinlichkeit
Nachteil: Faktorisierung von $N-1$ nötig.

13. Vorlesung (25. Juni 2008)

7.4 Miller-Rabin-Test

WIKI: Der Miller-Rabin-Test ist ein Monte-Carlo-Algorithmus, der durch die Randomisierung eine akzeptable Laufzeit erreicht, sowie schon nach wenigen Durchführungen mit hoher Wahrscheinlichkeit das korrekte Ergebnis gefunden hat.

Er ist ein probabilistischer Primzahltest. Erhält er eine natürliche Zahl n als Eingabe, so gibt er aus, ob diese Zahl eine Primzahl ist oder nicht. Gibt er dabei "n ist keine Primzahl" aus, so ist das richtig. Gibt er allerdings "n ist wahrscheinlich eine Primzahl" aus, so ist dies mit geringer Wahrscheinlichkeit falsch. Die Zahlen, die der Miller-Rabin-Test nicht als zusammengesetzt erkennt, nennt man starke Pseudoprimzahlen. Da der Algorithmus als ein wesentliches Element eine Zufallszahl benutzt, zählt er zur Klasse der Monte-Carlo-Algorithmen.

Sei $N \in \mathbb{N}$ ungerade mit $N-1 = 2^s n$ für n ungerade

SATZ 6: Wenn N eine Primzahl ist, dann gilt für $a \in \mathbb{N}$ mit $ggT(a,N) = 1$
entweder $a^n \equiv 1 \mod N$ oder es gibt ein $0 \leq r < s$ mit $a^{2^r n} \equiv -1 \mod N$

BEWEIS: (Skizze)
(Bem.: Sei G eine Gruppe und $g \in G$) mit $ord(g) = k$. Dann gilt $ord(g^l) = \frac{k}{ggT(k,l)}$
N Primzahl $\Rightarrow |\mathbb{Z}_N^x| = N-1$
$ggT(a,N) = 1 \Rightarrow [a] \in \mathbb{Z}N^x \wedge ord([a]^n) = 2^l$ mit $0 \leq l \leq s$
(Anmerkung: das r ist jetzt $l-1$)

$l = 0 \Rightarrow a^n \equiv 1 \mod N$

$l > 0 \Rightarrow ord([a]^{2^{l-1}n}) = 2$

$\Rightarrow a^{2^{l-1}n} \equiv -1 \mod N$ da [-1] das einzige Element in $\mathbb{Z}_N^x$der Ordnung 2 ist.

21:05

Ein $a \in N$ mit $ggT(a,N) = 1$, das keine der beiden Bedingungen aus Satz 6 erfüllt, heißt *Zeuge* gegen die Primaltät von N.

SATZ 7: Sei $N \geq 3$ eine ungerade zusammengesetzte Zahl (also keine Primzahl). Dann enthält die Menge $\{1, 2, \ldots, N-1\}$ höchstens $(N-1)/4$ Zahlen, die zu N teilerfremd und kein Zeuge gegen die Primalität von N sind.

BEWEIS: siehe Buchmann

Praktische Anwendung

- wähle zufällig und gleichverteilt Zahlen $a \in \{2, 3, \ldots, N-1\}$
- prüfe Bedingung ob Teilerfremd: $ggT(a, N) > 1 \Rightarrow N$ keine Primzahl
- berechne der Reihe nach $a^n, a^{2n}, a^{4n}, \ldots, a^{2^{s-1}n}$ falls a Zeuge gegen die Primalität $\Rightarrow N$ keine Primzahl
- wiederhole Test so oft, bis Fehlerwahrscheinlichkeit hinreichend klein
- Satz 7 $\Rightarrow$ nach t Wiederholungen ist Fehlerwahrscheinlichkeit $\leq (1/4)^t$

BSP. 8: Sei $N = 561$ (dies ist eine Carmichael-Zahl, für die der Fermat-Test versagt). Es gilt $s = 4, n = 35$. Für $a = 2$ berechnet man

$$\begin{aligned} 2^{35} &\equiv 263 \mod 561, \\ 2^{2 \cdot 35} &\equiv 166 \mod 561, \\ 2^{4 \cdot 35} &\equiv 67 \mod 561, \\ 2^{8 \cdot 35} &\equiv 1 \mod 561. \end{aligned}$$

Jetzt sehen wir, dass keine der Bedingungen erfüllt ist $\Rightarrow a$ ist ein Zeuge gegen die Primalität von N.

BEM. 9: Wie erzeugt man zufällge Primzahlen der Bitfolge k?
Sei n die gesuchte Primzahl.

- setze erstes und letztes Bit von n auf 1 ($\rightsquigarrow n$ ungerade; Bitlänge k)
- wähle verbleibende Bits zufällig und gleichverteilt
- teile n durch alle Primzahlen kleiner einer Schranke B (typischerweise $B = 10^6$, Primzahlen $\leq B$ tabelliert)
- wende Miller-Rabin t-mal an

53:30

8 Faktorisierung

8.1 Probedivision

Auf den meisten Systemen existiert eine Tabelle aller Primzahlen bis zu einer Schranke B (berechnet z. B. mit dem Sieb des Eratosthenes). Gegeben eine Zahl $n \in \mathbb{N}$, teilt man der Reihe nach durch diese Primzahlen, um kleinere Faktoren zu entdecken.

8.2 $(p-1)$-Methode von Pollard

WIKI: Die Pollard-$(p-1)$-Methode ist ein Verfahren zur Faktorisierung von zusammengesetzten Zahlen.

Sei p ein (unbekannter) Primteiler von n. Angenommen es gibt ein $k \in \mathbb{N}$ mit $x^k \not\equiv 1 \mod n$ aber $x^k \equiv 1 \mod p$ für ein $x \in \mathbb{N} \Rightarrow ggT(x^k - 1, n)$ ist ein echter Teiler von n, da $x^k - 1$ durch p teilbar ist. 64:12

- wähle ein $n \in \mathbb{N}$, z. B. $x = 2$
- wähle Schranke B und bestimme für alle Primzahlen $q \leq B$ den maximalen Exponenten e_q, so dass $q^{e_q} \leq B$.
  $\rightsquigarrow$ setze $k = \prod_q q^{e_q}$
- wenn $ggT(x^k - 1, n) = 1$, erhöhe Schranke B

  Bem.: $x^k \equiv 1 \mod p \Rightarrow k$ Vielfaches von $ord_p(x)$. $ord_p(x) = p - 1$ falls x eine Primitivwurzel und ein Teiler von $p-1$ sonst $\Rightarrow$ das Verfahren ist besonders effizient (d. h. wir können B relativ klein wählen), wenn $p-1$ nur kleine Primteiler hat.

  Bsp. 1: Sei $n = 3^{21} + 1 = 10460353204$
  - Probedivision durch alle Primzahlen unter 50
    $\rightsquigarrow$ Faktoren $2^2, 7^2, 43$
    $\rightsquigarrow n = 2^2 \cdot 7^2 \cdot 43 \cdot m$ mit $m = 1241143$
    Fermat-Test: $2^{m-1} \equiv 793958 \mod n$
    $\Rightarrow m$ ist immer noch zusammengesetzt
  - $(p-1)$-Test mit $B = 13$
    $\rightsquigarrow k = 2^3 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 = 360360$
    $\rightsquigarrow ggT(2^k - 1, m) = ggT(2^{360360} - 1, 1241143) = 547$
    $m = 547 \cdot 2269$ es gilt $547, 2269 \in \mathbb{P}$
    $\Rightarrow 2^2 \cdot 7^2 \cdot 43 \cdot 547 \cdot 2269$

14. Vorlesung (02. Juli 2008)

8.3 Quadratisches Sieb (Pomerance 1982)

Zu faktorisieren sei $n \in \mathbb{N}$

Grundprinzip: angenommen wir kennen $x, y \in \mathbb{Z}$ mit $x^2 \equiv y^2 \mod n$ aber $x \not\equiv \pm y \mod n$
$\Rightarrow n | x^2 - y^2 = (x+y)(x-y)$ aber $n \nmid x + y \wedge n \nmid x - y$
$\Rightarrow ggT(n, x-y)$ ist ein echter Teiler von n

Lemma 2: Sei $n \in \mathbb{N}$ ungerade, zusammengesetzt und keine Primzahl. Dann gibt es Zahlen $x, y \in \mathbb{N}$ mit $x^2 \equiv y^2 \mod n$ und $x \not\equiv \pm y \mod n$.

Beweis: n zusammengesetzt, keine Primzahlpotenz $\Rightarrow \exists u, v \in \mathbb{N} : n = uv \wedge ggT(u, v) = 1$ wähle $y \in \mathbb{Z}_n^x = \{[n] \in \mathbb{Z}_n \ \{[0]\} | ggT(z, n) = 1\}$
chinesischer Restesatz $\Rightarrow \exists x \in \mathbb{Z}_0^x$: $x \equiv y \mod u \wedge x \equiv -y \mod v$
$\Rightarrow n = uv | (x+y)(x-y) = x^2 - y^2 \Rightarrow x^2 \equiv y^2 \mod n$

Ann.: $x \equiv -y \mod n$ (Fall $x \equiv y \mod n$ analog)
$\Rightarrow n | x+y \Rightarrow u | x+y$ (da n aus u zusammengesetzt ist), d. h. $x \equiv -y \mod n$
nach Konstruktion $x \equiv y \mod n$
$\Rightarrow 2x \equiv 0 \mod n$ $\quad n$ ist ungerade $\Rightarrow u$ ist ungerade $\Rightarrow x \equiv 0 \mod n$
$\Rightarrow ggT(x, u) \neq 1 \Rightarrow ggT(x, n) \neq 1 \lightning x \in \mathbb{Z}_n^x$

Suchprinzip: (wie finden wir x, y in der Praxis?)
Sei $m = \lfloor \sqrt{n} \rfloor$, setze $f(s) = (s+m)^2 - n \Rightarrow (s+n)^2 \equiv f(s) \mod n$

Bsp. 3: Sei $n = 7429$
$m = 86 \Rightarrow f(s) = (s + 86)^2 - 7429$

$$f(-3) = -540 = -1 \cdot 2^2 \cdot 3^3 \cdot 5 \quad (1)$$
$$f(1) = 140 = 2^2 \cdot 5 \cdot 7 \quad (2)$$
$$f(2) = 315 = 3^2 \cdot 5 \cdot 7 \quad (3)$$

aus $((2), (3))$ folgt $\Rightarrow (87 \cdot 88)^2 \equiv (2 \cdot 3 \cdot 5 \cdot 7)^2 \mod 7429$
$\rightsquigarrow$ setze $x = 87 \cdot 88 \mod 7429 = 227$ und $y = 2 \cdot 3 \cdot 5 \cdot 7 \mod 7429 = 210$
Es gilt $x^2 - y^2 = 7429$ (also $x^2 \equiv y^2 \mod n$) $x - y = 17$, $x + y = 437$
$\rightsquigarrow ggT(17, 7429) = 17 \Rightarrow 17|7429$ (es gilt $7429 = 17 \cdot 19 \cdot 23$)

31:14

8.4 Auswahl geeigneter Kongruenzen

...