

# Entschlüsselte Geheimnisse

Martin Kreuzer

Fakultät für Informatik und Mathematik

Universität Passau

`martin.kreuzer@uni-passau.de`

Regionale Lehrerfortbildung

Oberviechtach, 19.1.2010

## Inhaltsübersicht

1. Was ist Kryptographie?
2. Häufigkeitsanalyse
3. Kryptoanalyse des Vigenere-Systems
4. Die RSA-Verschlüsselung
5. Kann man RSA knacken?
6. Protokolle und ihre Tücken
7. Algebraische Kryptoanalyse

## 1 – Was ist Kryptographie?

Ich kann Dir das Geheimnis des Universums nicht verraten,  
denn dann wäre es ja kein Geheimnis mehr.  
(Mullah Nasruddin)

<i>κρυπτος</i>	<b>cryptos</b>	das Verborgene
<i>γραφειν</i>	<b>graphein</b>	schreiben
<i>λογος</i>	<b>logos</b>	die Rede, die Lehre
<i>ανάλυσις</i>	<b>analysis</b>	die Auflösung
<i>στηγανος</i>	<b>steganos</b>	bedeckt, versteckt

## 2 – Häufigkeitsanalyse

**Klappt die Kryptoanalyse?  
Nicht immer, aber immer öfter.**

Wenn bei einem Verschlüsselungssystem jeder Klartextbuchstabe stets durch das gleiche Geheimtextzeichen ersetzt wird, spricht man von einer **monoalphabetischen** Verschlüsselung.

**Beispiel: Caesar-Verschlüsselung** (Jeder Buchstabe wird durch den im Alphabet drei Stellen späteren Buchstaben ersetzt.)

Ein solches System kann man mit der **Häufigkeitsanalyse** brechen. Diese Methode wurde ca. 850 in Bagdad veröffentlicht von

**Abu Yusuf Yaqub ibn Is-haq ibn as Sabbah ibn 'Omran ibn  
Ismail Al-Kindi (805 – 873 n.Chr.)**

Erschrieb ein Buch mit dem Titel

**Risalah fi Istikhraj al-Mu'amma**

Manuskript für die Entschlüsselung verschlüsselter Nachrichten

Al-Kindi war hauptberuflich Philosoph und Mathematiker am

**Haus der Weisheit**

der Kaliphen von Bagdad.

## Also schrieb Al-Kindi ...

Eine Methode eine verschlüsselte Nachricht zu knacken, **wenn wir ihre Sprache kennen**, besteht darin einen **Klartext** in derselben Sprache zu finden, der mindestens eine Seite lang ist. Darin zählen wir die Häufigkeit jedes einzelnen Buchstaben.

Den am häufigsten vorkommenden Buchstaben nennen wir den “ersten”, den nächsthäufigsten den “zweiten”, den folgenden den “dritten”, und so weiter, bis wir alle im Klartextbeispiel vorkommenden Buchstaben abgearbeitet haben.

Jetzt betrachten wir den verschlüsselten Text und klassifizieren seine Symbole ebenso. Das am häufigsten auftretende Symbol wandeln wir in den “ersten” Buchstaben um, das nächsthäufigste Symbol in den “zweiten” Buchstaben, und so weiter, bis wir alle Symbole in der verschlüsselten Nachricht abgearbeitet haben.

## Legrand versus Captain Kidd

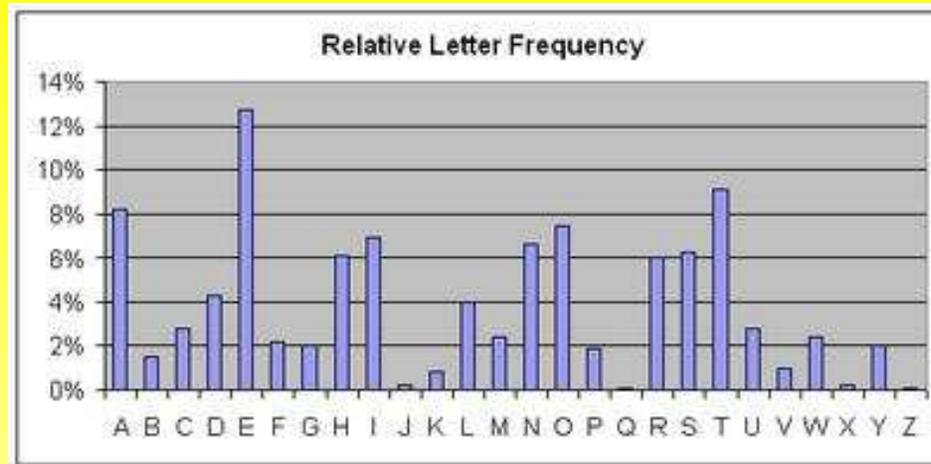
### Beispiel einer Häufigkeitsanalyse *a la* Al-Kindi

In Edgar Allan Poes Erzählung “Der Goldkäfer” erhält der Held Legrand folgende verschlüsselte Nachricht:

53 ≠ ≠ † 305)) 6\*;48 26)4 ≠ .)4 ≠) ;806\* ;48 † 8  
]/60) )85;1 ≠ (;:≠ \*8 † 83 (88)5 \*†;46 (;88\*  
96\*?; 8)\* ≠ ( ;485) ;5 \* †2 : \* ≠ (; 4956\* 2(5 \* –  
4)8]/ 8\*;40 69285 );)6† 8)4 ≠ ≠ ;1(≠ 9 ;4808  
1;8 : 8 /1;48 †85;4 )485† 52880 6 \* 81( /9;48  
;(88; 4(≠?3 4;48) 4 ≠;16 1;: 18 8;≠?;

Wegen einer Markierung weiß er, dass sie von dem berühmten Seeräuber Captain Kidd stammt und in Englisch verfasst wurde.

Die Häufigkeitstabelle der englischen Sprache ist:



Wir folgen al-Kindi und erstellen eine Häufigkeitstafel der Symbole:

1. 8	2. ;	3. 4	4. )	5. ≠	6. *	7. 5	8. 6	9. (
16.5%	13%	9.5%	8.5%	7%	6.5%	6%	5.5%	5%

Auf jeden Fall sollte also  $8 \mapsto E$  und  $;\mapsto T$  gelten.

Damit erhalten wir folgende Teilentschlüsselung:

$53 \neq \neq \dagger$      $305))$      $6 * T4E$      $26)4 \neq$      $.)4 \neq)$      $TE06*$      $T4E \dagger E$   
 $] / 60)$      $)E5T1$      $\neq (T : \neq$      $*E \dagger E3$      $(EE)5$      $* \dagger T46$      $(TEE*$   
 $96*?T$      $E)* \neq ($      $T4E5)$      $T5 * \dagger 2$      $: * \neq (T$      $4956*$      $2(5 * -$   
 $4)E] /$      $E * T40$      $692E5$      $)T)6\dagger$      $E)4 \neq \neq$      $T1(\neq 9$      $T4E0E$   
 $1TE : E$      $/1T4E$      $\dagger E5T4$      $)4E5\dagger$      $52EE0$      $6 * E1($      $/9T4E$   
 $T(EET$      $4(\neq?3$      $4T4E)$      $4 \neq T16$      $1T : 1E$      $ET \neq?T$

Wegen des siebenmaligen Auftretens der Kombination  $T4E$  liegt es nahe,  $4 \mapsto H$  zu vermuten, was mit der Häufigkeitstabelle verträglich ist. Wir erhalten eine neue Teilentschlüsselung:

53 ≠ ≠ † 305)) 6\**THE* 26)*H* ≠ .)*H* ≠) *TE*06\* *THE*†*E*  
 ]/60) )*E*5*T*1 ≠ (*T* :≠ \**E* † *E*3 (*EE*)5 \* † *TH*6 (*TEE*\*  
 96\*?*T* *E*)\* ≠ ( *THE*5) *T*5 \* †2 : \* ≠ (*T* *H*956\* 2(5 \* –  
*H*)*E*] / *E*\**TH*0 692*E*5 )*T*)6† *E*)*H* ≠ ≠ *T*1(≠ 9 *THE*0*E*  
 1*TE*:*E* /1*THE* †*E*5*TH* )*HE*5† 52*EE*0 6 \* *E*1( /9*THE*  
*T*(*EET* *H*(≠?3 *HTHE*) *H*≠*T*16 1*T* : 1*E* *ET* ≠?*T*

Jetzt zeigt der Anfang der letzten Zeile ( $\mapsto R$  (bei 5% vs. 6.5%) und dann liefert das Ende der zweiten Zeile das Wort “thirteen”, also  $6 \mapsto I$  (bei 5.5% vs. 6.5%) sowie  $*$   $\mapsto N$  (bei 6% vs. 7%).

Unsere Zwischenresultat lautet jetzt:

53 ≠ ≠ † 305)) INTHE 2I)H ≠ .)H ≠) TE0IN THE†E  
 ]/I0) )E5T1 ≠ RT:≠ NE†E3 REE)5 N†THI RTEEN  
 9IN?T E)N ≠ R THE5) T5N † 2 :N ≠ RT H95IN 2R5N–  
 H)E]/ ENTH0 I92E5 )T)I† E)H ≠ ≠ T1R ≠ 9 THE0E  
 1TE:E /1THE †E5TH )HE5† 52EE0 INE1R /9THE  
 TREET HR ≠ ?3 HTHE) H≠T1I 1T : 1E ET ≠ ?T

Die letzte Zeile “tree thr...h the” macht nur für “through” Sinn, also  
 $\neq \mapsto O$  (bei 7% vs. 8%) und  $? \mapsto U$  (bei 1.5% vs. 3%) sowie  $3 \mapsto G$   
 (bei 2% vs. 1.5%). Dann muss die Fortsetzung “through the .hot”  
 mit “shot” ergänzt werden, da nur “a” und “s” häufig genug sind.  
 Dies liefert  $) \mapsto S$  (bei 8.5% vs. 6%). Schließlich bleibt für das “a”  
 nur  $5 \mapsto A$  übrig (bei 6% vs. 8%). Jetzt lautet der Zwischenstand:

AGOO† G0ASS INTHE 2ISHO .SHOS TE0IN THE†E  
 ]/I0S SEAT1 ORT:O NE†EG REESA N†THI RTEEN  
 9INUT ESNOR THEAS TAN†2 :NORT H9AIN 2RAN—  
 HSE]/ ENTH0 I92EA STSI† ESHOO T1RO9 THE0E  
 1TE:E /1THE †EATH SHEA† A2EE0 INE1R /9THE  
 TREET HROUG HTHES HOT1I 1T :1E ETOUT

Von hier aus ist es ein Leichtes, den Text zu vervollständigen:

A good glass in the bishop's hostel in the devil's seat  
 forty one degrees and thirteen minutes northeast and by north  
 main branch seventh limb east side  
 shoot from the left eye of the death's head a bee line  
 from the tree trough the shot fifty feet out.

## 3 – Kryptoanalyse des Vigenere-Systems

Das Geheimnis der Kreativität ist das Wissen darum,  
wie man seine Quellen verbirgt.

(Albert Einstein)

Die erste **polyalphabetische Verschlüsselung** wurde 1466  
beschrieben von **Leon Battista Alberti** in seinem Buch

### **De Componendis Cifris**

(Anleitung zum Verschlüsseln von Texten)

und später u.a. von **Blaise de Vigenere** und anderen  
weiterentwickelt. Eine einfache Version dieses Verfahrens funktioniert  
wir folgt:

(1) Wähle ein **Schlüsselwort**, z.B. **KATZE**. Es entspricht Verschiebungen des Alphabets um (10, 0, 19, 25, 4) Buchstaben.

(2) Schreibe den Klartext und die Verschiebungen untereinander:

<i>S</i>	<i>E</i>	<i>N</i>	<i>D</i>	<i>E</i>	<i>T</i>	<i>B</i>	<i>I</i>	<i>T</i>	<i>T</i>	<i>E</i>	<i>H</i>	<i>I</i>	<i>L</i>	<i>F</i>	<i>E</i>
10	0	19	25	4	10	0	19	25	4	10	0	19	25	4	10

(3) Wende die Verschiebungen an und erhalte den Geheimtext

*C E G C I D B B S X O H B K J O*

Ein fester Buchstabe, z.B. **E** wird immer wieder anders verschlüsselt, d.h. das System ist **polyalphabetisch**. Wie knackt man eine solche Verschlüsselung?

## Der Kasiski-Test

dient dazu, die Schlüsselwortlänge zu finden. **Friedrich Wilhelm Kasiski** (1805 – 1881) war ein pensionierter Offizier des 33. Preußischen Infanterieregiments.

1863 schrieb er das Buch

### **Die Geheimschriften und die Dechiffrier-Kunst**

Darin erklärte er, wie man Vigenère Kryptosysteme mit wiederholendem Schlüsselwort knacken kann, aber keiner interessierte sich dafür.

Kasiski wandte sich daraufhin der Amateur-Anthropologie und der Amateur-Archäologie zu. Er starb ohne jemals zu erfahren, dass er die Kunst des Codeknackens revolutioniert hatte.

**Idee:** Kommt zweimal derselbe Buchstabe oder dieselbe Buchstabenfolge im Klartext vor (z.B. das Wort **der**) und ist der Abstand durch die Schlüsselwortlänge teilbar, so wird jeweils gleich verschlüsselt.

(1) Suche also im Geheimtext Folgen von drei oder mehr Buchstaben, die mehrfach vorkommen.

(2) Bestimme die Abstände dieser Buchstabengruppen.

(3) Der **ggT** möglichst vieler dieser Abstände ist höchstwahrscheinlich die Schlüsselwortlänge  $\ell$ .

(4) Die Buchstaben  $1, \ell + 1, 2\ell + 1, \text{etc.}$  werden nun alle mit der gleichen Verschiebung verschlüsselt. Führe eine Häufigkeitsanalyse durch und finde diese Verschiebung!

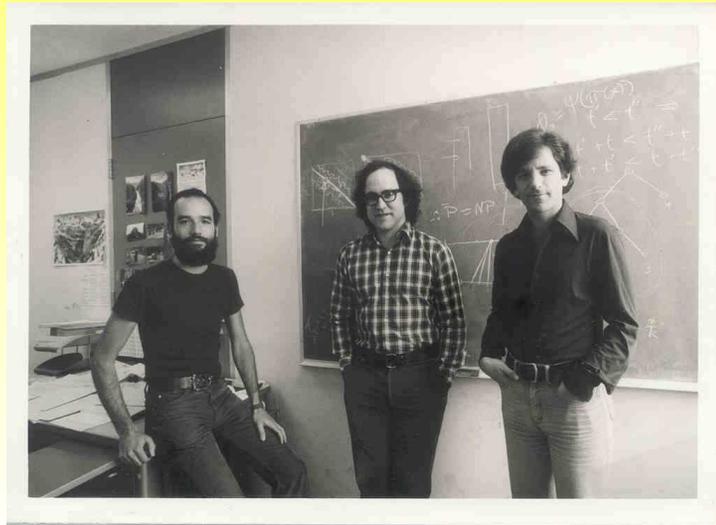
## Der Friedmann-Test

beruht ebenfalls auf sehr einfachen kombinatorischen Überlegungen, die man mit den Mitteln der Schulmathematik nachvollziehen kann. Er liefert eine Formel für die wahrscheinlichste Schlüsselwortlänge beim Vigenere-System.

## 4 – Die RSA-Verschlüsselung

Manche Menschen würden eher sterben  
als selbständig zu denken,  
und das tun sie dann auch.  
(Bertrand Russell)

Rivest – Shamir – Adleman



**1976:** Ronald Rivest, Adi Shamir (zwei Informatiker) und Leonard Adleman (ein Mathematiker) suchen nach einer Funktion für die **Public Key Kryptographie**.

Die beiden Informatiker machen ein Jahr lang zahlreiche Vorschläge, der Mathematiker widerlegt sie.

**April 1977:** Rivest findet eine geeignete **Einweg-Funktion** und schreibt eine Forschungsarbeit.

Adleman will anfangs nicht als Koautor beteiligt sein. Dann gibt er nach unter der Bedingung, dass er als letzter genannt wird.

**“Ich dachte, dies wäre die uninteressanteste Arbeit bei der ich jemals Koautor wäre.”**

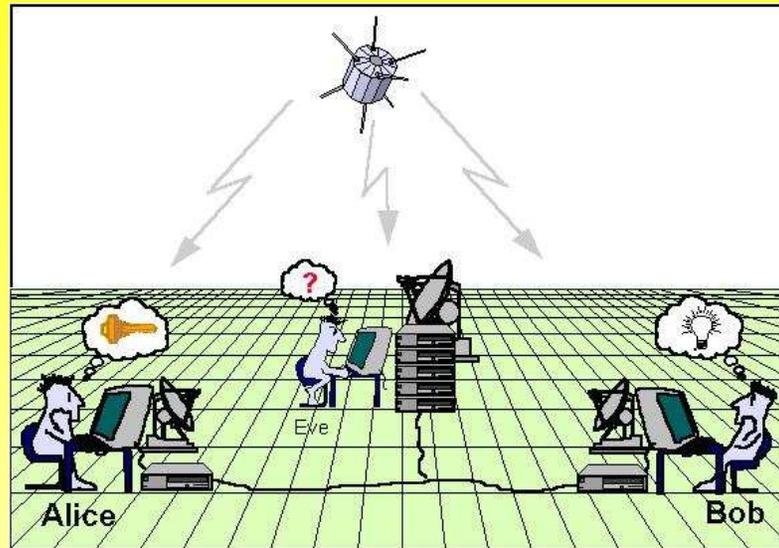
**Feb. 1978:** Der Artikel “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems” erscheint in *Communications of the ACM*, Band 21, Seiten 120-126.

Das MIT erhält 5000 Anfragen nach Sonderdrucken der Arbeit.

**Sep. 1983:** US-Patent # 4,405,829 “Cryptographic Communications System and Method” für das RSA Kryptosystem

**ab 1983:** Die USA verhängen strenge Exportbeschränkungen für RSA. Der Einsatz ist strikt reglementiert und bleibt im wesentlichen auf Banken beschränkt.

## Wie funktioniert RSA ?



**Bob** verschlüsselt seine Nachricht mit dem **Public Key** von Alice.

**Eve** hört mit, kann die Verschlüsselung aber nicht knacken.

Nur **Alice** kennt ihren **geheimen Schlüssel** und kann die Nachricht entschlüsseln.

## Wie funktioniert RSA nun wirklich?

**Geheim:** Zwei Primzahlen  $p$  und  $q$  und eine Zahl  $e$  mit  $10 < e < n = p \cdot q$ .

**Öffentlich:** Das Produkt  $n$  und eine Zahl  $d$  mit  $de \equiv 1 \pmod{(p-1)(q-1)}$ .

**Verschlüsselung:** Verwandle die Nachricht in Zahlen  $m$  mit  $m < n$ . Berechne die Zahl  $c = m^e \pmod{n}$  und sende sie.

**Entschlüsselung:** Berechne die Zahl  $c^d \pmod{n}$  und erhalte  $m$ .

**Korrektheit:** Nach dem **Satz von Euler** gilt  $c^d = (m^e)^d = m^{de} \equiv m^1 \pmod{n}$ .

**Sicherheit:** Um  $p, q$  oder  $d$  zu berechnen muss man  $n$  in seine Primfaktoren zerlegen. Dies gilt als sehr schwierig.

## Der kleine Satz von Fermat

**Satz.** Sei  $p$  eine Primzahl und  $a$  eine nicht durch  $p$  teilbare Zahl.  
Dann gilt

$$a^{p-1} \equiv 1 \pmod{p}.$$

**Beweis:** Wenn wir von  $a$  ein Vielfaches von  $p$  subtrahieren, bleibt  $a^{p-1} \pmod{p}$  gleich. Also können wir  $0 < a < p$  annehmen..

Betrachte nun die Reste  $a \pmod{p}$ ,  $2a \pmod{p}$ , u.s.w. bis  $(p-1)a \pmod{p}$ . Dies sind  $p-1$  verschiedene Zahlen im Bereich  $\{1, \dots, p-1\}$ , denn wären zwei gleich, z.B.

$$i a \equiv j a \pmod{p}$$

so würde  $p$  das Produkt  $(i-j)a$  teilen. Folglich würde  $p$  dann  $a$  oder  $(i-j)$  teilen, was nicht geht, da beide Zahlen  $> 0$  und  $< p$  sind.

Somit sind diese Reste eine Permutation der Zahlen  $\{1, \dots, p-1\}$ .

Also ist ihr Produkt  $1 \cdot 2 \cdots (p-1) = (p-1)!$ , d.h. es gilt

$$a \cdot (2a) \cdots (p-1)a = (p-1)! a^{p-1} \equiv (p-1)! \pmod{p}$$

Folglich besitzt  $a^{p-1}$  den Rest 1 modulo  $p$ .

**q.e.d.**

### **Satz. (Chinesischer Restsatz)**

Sind  $p, q$  teilerfremde Zahlen, so gibt es zu je zwei Zahlen  $a, b \in \mathbb{Z}$  genau ein  $c \in \{0, \dots, pq-1\}$  mit

$$c \equiv a \pmod{p} \quad \text{und} \quad c \equiv b \pmod{q}$$

## 5 – Kann man RSA knacken?

Es macht irgendwie Spaß das Unmögliche zu tun.

(Walt Disney)

Selbst mit modernsten Computern und den besten bekannten Algorithmen kann man Zahlen  $n$  nicht in ihre Faktoren  $p$  und  $q$  zerlegen, wenn  $p$  und  $q$  mehr als 100 Stellen haben.

Es wurde aber trotz intensiver Bemühungen noch nicht **bewiesen**, dass die Aufgabe nicht in vernünftiger Zeit lösbar ist.

Stattdessen haben die Mathematiker bewiesen, dass man große Zahlen mit einem Quantencomputer **ruck-zuck** faktorisieren kann.

**Es ist also nur noch eine Frage der Zeit, bis die Codeknacker wieder einmal triumphieren!**

## 6 – Protokolle und ihre Tücken

Frage nicht was Dein Land für Dich tun kann.

Frage was Dein Land für die großen Banken tut.

Ein (**kryptographisches**) **Protokoll** ist eine Folge von Anweisungen mit denen zwei (oder mehr) Parteien eine gewisse Aufgabe erledigen wollen, z.B. Schlüsselvereinbarung, Authentifikation, elektronische Signatur, u.s.w.

Das erste wichtige Protokoll war die **Diffie-Hellman Schlüsselvereinbarung**. Dabei wollen zwei Parteien **A** und **B** über einen öffentlichen Kanal einen gemeinsamen Geheimschlüssel vereinbaren.

## Diffie-Hellman Schlüsselvereinbarung

- (1) **A** und **B** einigen sich auf eine große Primzahl  $p$  und eine Zahl  $g$  deren Potenzen alle Reste modulo  $p$  (außer  $\bar{0}$ ) durchlaufen.
- (2) **A** wählt eine Zufallszahl  $a$  und **B** wählt eine Zufallszahl  $b$ .
- (3) **A** berechnet  $x = g^a \pmod{p}$  und **B** berechnet  $y = g^b \pmod{p}$ .
- (4) **A** und **B** tauschen  $x$  und  $y$  aus.
- (5) **A** berechnet  $k = y^a \pmod{p}$  und **B** berechnet  $k = x^b \pmod{p}$ .  
(In beiden Fällen gilt  $k = g^{ab} \pmod{p}$ .)

Kann ein Angreifer **diskrete Logarithmen** berechnen, d.h. kann er aus  $g^a \pmod{p}$  auf  $a$  schließen, so kann er den Geheimschlüssel  $k$  finden.

## Angriffe auf Protokolle

**Man-in-the Middle Angriff:** Der Angreifer **E** fängt die Nachrichten von **A** ab und leitet eigene Nachrichten an **B** weiter. Dann fängt er die Antworten von **B** ab und sendet eigene Antworten an **A** weiter.

**Replay Angriff:** Der Angreifer liest alle Nachrichten mit und sendet später noch einmal dieselbe Kommunikation an **B**.  
(Beispielsweise bei der Authentifikation gegenüber einem Computer mit dem Passwort-Verfahren.)

**Denial-of-Service (DOS) Angriff:** Der Angreifer sendet so viele Kommunikationsanfragen an **B**, dass der Rechner von **B** überlastet ist und zusammenbricht oder Fehler produziert.

## 7 – Algebraische Kryptoanalyse

Man kann nicht jemandem etwas beibringen  
wenn sein Gehalt davon abhängt,  
dass er es nicht kapiert.  
(Upton Sinclair)

Man kann jede Verschlüsselungsfunktion betrachten als eine Abbildung, die einem 0-1-Tupel (dem **Klartext**) ein anderes (den **Geheimtext**) zuordnet.

Ist  $\mathbb{F}_2 = \{\bar{0}, \bar{1}\}$  der Körper mit zwei Elementen, so ist dies also eine Abbildung  $\epsilon : (\mathbb{F}_2)^s \longrightarrow (\mathbb{F}_2)^n$ .

Jede solche Abbildung ist durch **Polynome** gegeben, d.h. es gibt  $f_1, \dots, f_s \in \mathbb{F}_2[x_1, \dots, x_n]$  mit  $\epsilon(v) = (f_1(v), \dots, f_s(v))$  für alle  $v$ .

## Wir knacken zwei Runden DES

**DES** = **D**ata **E**ncryption **S**tandard (1977 – 2001)

**AES** = **A**dvanced **E**ncryption **S**tandard (seit 2001)

Betrachte die Verschlüsselungsfunktion von DES als Abbildung

$$\epsilon : (\mathbb{F}_2)^{64} \longrightarrow (\mathbb{F}_2)^{64}.$$

**Satz 7.1** *Zu jeder Abbildung  $\epsilon : (\mathbb{F}_2)^{64} \longrightarrow (\mathbb{F}_2)^{64}$  gibt es*

**Polynome**  $f_1, \dots, f_{64}$  *in 64 Unbestimmten mit*

$$\epsilon(x_1, \dots, x_{64}) = (f_1(x_1, \dots, x_{64}), \dots, f_{64}(x_1, \dots, x_{64}))$$

Wir führen eine **Known-Plaintext-Attacke** durch. Dazu nehmen wir an, dass wir ein Paar (Klartext, Geheimtext) kennen, z.B. weil das erste Byte einer pdf-Datei verschlüsselt wurde.

Dann schreiben wir die Verschlüsselungsfunktion in Abhängigkeit der 56 bits des Geheimschlüssels (Unbestimmte  $k_0, \dots, k_{55}$ ) und erhalten ein **algebraisches Gleichungssystem**

$$\begin{aligned} g_1(k_0, \dots, k_{55}) &= 0 \\ &\vdots \\ g_{1120}(k_0, \dots, k_{55}) &= 0 \end{aligned}$$

Zum Beispiel seien folgende Daten gegeben:

**Klartext:** [0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0,  
0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1,  
1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0];

**Geheimtext:** [1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1,  
0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0,  
0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0];

Gleichungssystem (1120 Gleichungen, 288 Unbestimmte):

$$\begin{aligned} & x[0, 1]x[0, 3]x[0, 4] + x[0, 2]x[0, 3]x[0, 4] + x[0, 1]x[0, 4]x[0, 5] \\ + & x[0, 3]x[0, 4]x[0, 5] + x[0, 1]x[0, 2]y[0, 0] + x[0, 1]x[0, 3]y[0, 0] \\ + & x[0, 2]x[0, 3]y[0, 0] + x[0, 1]x[0, 4]y[0, 0] + x[0, 3]x[0, 4]y[0, 0] \\ + & x[0, 2]x[0, 5]y[0, 0] + x[0, 3]x[0, 5]y[0, 0] + x[0, 1]x[0, 2]y[0, 1] \\ + & x[0, 1]x[0, 3]y[0, 1] + x[0, 2]x[0, 3]y[0, 1] + x[0, 1]x[0, 4]y[0, 1] \\ + & x[0, 3]x[0, 4]y[0, 1] + x[0, 1]x[0, 5]y[0, 1] + x[0, 4]x[0, 5]y[0, 1] \\ + & x[0, 1]y[0, 0]y[0, 1] + x[0, 4]y[0, 0]y[0, 1] + x[0, 1]x[0, 3]y[0, 3] \\ + & x[0, 2]x[0, 3]y[0, 3] + x[0, 1]x[0, 4]y[0, 3] + x[0, 2]x[0, 4]y[0, 3] \\ + & x[0, 2]x[0, 5]y[0, 3] + x[0, 3]x[0, 5]y[0, 3] + x[0, 2]y[0, 0]y[0, 3] \end{aligned}$$

$$\begin{aligned} &+ x[0, 3]y[0, 0]y[0, 3] + x[0, 4]y[0, 0]y[0, 3] + x[0, 5]y[0, 0]y[0, 3] \\ &+ x[0, 1]y[0, 1]y[0, 3] + x[0, 4]y[0, 1]y[0, 3] + y[0, 0]y[0, 1]y[0, 3] \\ &+ x[0, 3]y[0, 2], \end{aligned}$$

und weitere **1119** Gleichungen ....

Es ergibt sich  $(k_0, \dots, k_{55}) = (0, 0, 1, 1, 1, 0, \dots)$ .

**DES ist nicht mehr sicher!**

**Und an AES arbeiten wir!**

**Vielen Dank für Ihre Aufmerksamkeit!**