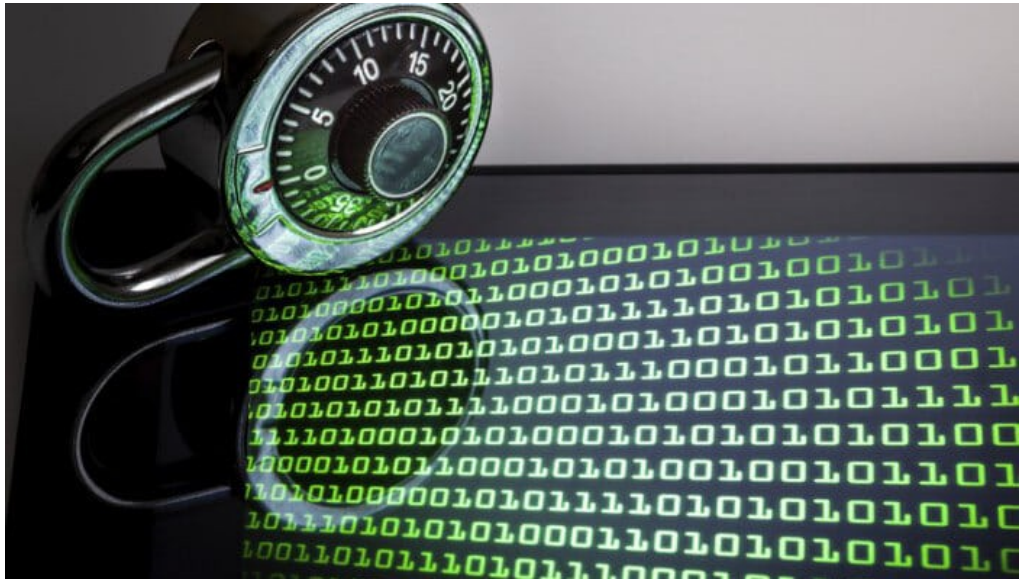


Kryptographie

Vorlesung 3: Sicherheit

Babeş-Bolyai Universität, Department für Informatik, Cluj-Napoca
csacarea@cs.ubbcluj.ro

Perfekte Sicherheit

Szenario:

- Angreifer besitzt unbeschränkte Berechnungskraft. Seien M, K, C versehen mit folgenden Wahrscheinlichkeits-Verteilungen.
  - Sei M eine Zufallsvariable für eine beliebige Wahrscheinlichkeits-Verteilung auf M, d.h. wir ziehen ein $m \in M$ mit $Ws[M = m]$.
  - Sei K eine Zufallsvariable induziert durch $K \leftarrow Gen(1^n)$.
  - Sei $C \leftarrow E_K(M)$ eine Zufallsvariable für die Wahrscheinlichkeits-Verteilung auf C.
  - K und M sind unabhängig, C hängt von K und M ab.
- Es gelte oBdA $Ws[M = m] > 0$ und $Ws[C = c] > 0$ für alle $m \in M, c \in C$. (Andernfalls entferne m aus M bzw. c aus C.)

Definition (Perfekte Sicherheit)

Ein Verschlüsselungsverfahren $\Pi = (Gen, Enc, Dec)$ *heißt perfekt sicher, falls für alle Wahrscheinlichkeits-Verteilungen auf* M, $m \in M, c \in C$ *gilt:*

$$Ws[M = m \mid C = c] = Ws[M = m].$$

Interpretation: c liefert dem Angreifer keine Informationen über m.

Verteilung auf Chiffretexten unabhängig vom Plaintext

Satz (Satz Chiffretext-Verteilung)

Ein Verschlüsselungsverfahren Π *ist perfekt sicher gdw* $Ws[C = c \mid M = m] = Ws[C = c]$ *für alle* $m \in M, c \in C$.

Satz (Satz von Bayes)

Für zwei Ereignisse A, B *mit* $Ws[B] > 0$ *gilt:*

$$Ws[A \mid B] = \frac{Ws[B \mid A] \cdot Ws[A]}{Ws[B]}.$$

Beweis

- $\Rightarrow$: Sei Π perfekt sicher. Nach dem Satz von Bayes gilt

$$Ws[C = c \mid M = m] = \frac{Ws[M = m \mid C = c] \cdot Ws[C = c]}{Ws[M = m]} = Ws[C = c].$$

- $\Leftarrow$: Aus $Ws[C = c \mid M = m] = Ws[C = c]$ folgt mit dem Satz von Bayes $Ws[M = m \mid C = c] = Ws[M = m]$.
- Damit ist Π perfekt sicher.

Ununterscheidbarkeit von Verschlüsselungen

Satz (Ununterscheidbarkeit von Verschlüsselungen)

Ein Verschlüsselungsverfahren Π *ist perfekt sicher gdw für alle* $m_0, m_1 \in M, c \in C$ *gilt* $Ws[C = c \mid M = m_0] = Ws[C = c \mid M = m_1]$.

Beweis:

- $\Rightarrow$: Mit dem Satz auf voriger Folie gilt für perfekt sichere Π

$$Ws[C = c \mid M = m_0] = Ws[C = c] = Ws[C = c \mid M = m_1].$$

Beweis (Fortsetzung)

- $\Leftarrow$: Sei $m' \in M$ beliebig. Es gilt

$$\begin{aligned} Ws[C = c] &= \sum_{m \in M} Ws[C = c \mid M = m] \cdot Ws[M = m] \\ &= Ws[C = c \mid M = m'] \cdot \sum_{m \in M} Ws[M = m] \\ &= Ws[C = c \mid M = m']. \end{aligned}$$

- Die perfekte Sicherheit von Π folgt mit dem Satz auf voriger Folie.

Beispiel

Verschiebungschiffre

- $\mathcal{K} = \{0, \ldots, 25\}$ mit $Pr[K = k] = 1/26$ für alle $k \in \mathcal{K}$.
- Gegeben sei folgende Verteilung über $\mathcal{M}$:

$$Pr[M = a] = 0.7 \text{ und } Pr[M = z] = 0.3.$$

- Welche ist die Wahrscheinlichkeit, dass der Schlüsseltext B ist?
- Es gibt genau 2 Möglichkeiten dafür
  - $M = a$ und $K = 1$ oder
  - $M = z$ und $K = 2$.
- M und K sind unabhängig, daher gilt

$$Pr[M = a \wedge K = 1] = Pr[M = a] \cdot Pr[K = 1] = 0.7 \cdot \left(\frac{1}{26}\right).$$

Beispiel

Verschiebungschiffre

- Analog $Pr[M = z \wedge K = 2] = 0.3 \cdot \left(\frac{1}{26}\right)$. Es folgt

$$Pr[C = B] = Pr[M = a \wedge K = 1] + Pr[M = z \wedge K = 2]$$

$$= 0.7 \cdot \left(\frac{1}{26}\right) + 0.3 \cdot \left(\frac{1}{26}\right) = \frac{1}{26}.$$

Beispiel

Verschiebungschiffre

- Analog kann man auch bedingte Wahrscheinlichkeiten berechnen:
  - Welche ist die Wahrscheinlichkeit, dass a verschlüsselt worden ist, gegeben die abgehörte Nachricht B?
- Aus dem Satz von Bayes folgt

$$Pr[M = a \mid C = B] = \frac{Pr[C = B \mid M = a] \cdot Pr[M = a]}{Pr[C = B]}$$

$$= \frac{0.7 \cdot Pr[C = B \mid M = a]}{1/26}.$$

- $Pr[C = B \mid M = a] = 1/26$ weil für $M = a$, die einzige Möglichkeit für $C = B$ ist über $K = 1$ und dies geschieht mit der Wahrscheinlichkeit $1/26$.
- Es folgt $Pr[M = a \mid C = B] = 0.7$.

Das One-Time Pad (Vernam Verschlüsselung)

Definition (One-Time Pad (1918))

Sei $M = C = K = \{0,1\}^\ell$.

1. `Gen`: *Ausgabe* $k \in \{0,1\}^\ell$
2. `Enc`: *Für* $m \in \{0,1\}^\ell$ *berechne* $c = Enc_k(m) := m \oplus k$.
3. `Dec`: *Für* $c \in \{0,1\}^\ell$ *berechne* $m = Dec_k(c) := c \oplus k$.

Satz

Sicherheit des One-Time Pads Das One-Time Pad ist perfekt sicher gegen über COA Angriffen.

Beweis

- Wegen $C = Enc_K(M) = M \oplus K$ gilt für alle $m_0, m_1 \in M$ und $c \in C$

$$Ws[C = c \mid M = m_0] = Ws[M \oplus K = c \mid M = m_0] = Ws[K = m_0 \oplus c]$$
$$= \frac{1}{2^\ell} = Ws[C = c \mid M = m_1].$$

- Damit ist das One-Time Pad perfekt sicher.

Bemerkung (Nachteil:)

Schlüsselraum ist so groß wie der Nachrichtenraum.

Beschränkungen perfekter Sicherheit

Satz (Größe des Schlüsselraums)

Sei Π *perfekt sicher. Dann gilt* $|\mathcal{K}| \geq |\mathcal{M}|$.

Beweis: Angenommen $|\mathcal{K}| < |\mathcal{M}|$.

- Sei M die Gleichverteilung auf $\mathcal{M}$.
- Für $c \in C$ definiere $D(c) = \{m \mid m = Dec_k(c) \text{ für ein } k \in \mathcal{K}\}$.
- Es gilt $|D(c)| \leq |\mathcal{K}|$, da jeder Schlüssel k höchstens ein m liefert.
- Wegen $|\mathcal{K}| < |\mathcal{M}|$ folgt $|D(c)| < |\mathcal{M}|$. D.h. es gibt ein $m \in \mathcal{M} \setminus D(c)$ mit

$$0 = Ws[M = m \mid C = c] < Ws[M = m].$$

- Damit ist Π nicht perfekt sicher.

Satz von Shannon (1949)

Satz (Shannon)

Sei $\Pi = (Gen, Enc, Dec)$ *mit* $|\mathcal{M}| = |\mathcal{C}| = |\mathcal{K}|$. Π *ist perfekt sicher gdw*

1. `Gen` *wählt alle* $k \in \mathcal{K}$
2. *Für alle* $m \in \mathcal{M}, c \in \mathcal{C}$ *existiert genau ein* $k \in \mathcal{K}$: $c = Enc_k(m)$.

Beweisidee:

- $\Leftarrow$: Jedes $m \in \mathcal{M}$ korrespondiert zu genau einem $c \in \mathcal{C}$ via k.
- D.h. m wird zu c verschlüsselt, falls k verwendet wird.
- Damit gilt

$$Ws[C = c \mid M = m] = Ws[K = k] = \frac{1}{|\mathcal{K}|} \text{ für alle } m \in \mathcal{M}.$$

- Es folgt $Ws[C = c \mid M = m_0] = \frac{1}{|\mathcal{K}|} = Ws[C = c \mid M = m_1]$.
- Damit ist Π perfekt sicher.

Beweisidee (Fortsetzung):

- $\Rightarrow$: Sei Π perfekt sicher mit $|\mathcal{M}| = |C| = |\mathcal{K}|$.
- Definiere $S(m) = \{Enc_k(m) \mid k \in \mathcal{K}\}$.
- Für alle (m, c) existiert mindestens ein $k \in \mathcal{K}$ mit $c = Enc_k(m)$. (Sonst: $\exists(m, c)$ mit $c \neq Enc_k(m)$ für alle $k \in \mathcal{K}$. Dann gilt $Ws[M = m \mid C = c] = 0 < Ws[M = m]$. Widerspruch.)
- $\Rightarrow |C| \leq |S(m)| \leq |C|$ und deshalb $|S(m)| = |C| = |\mathcal{K}|$.
- Also: für jedes (m, c) gibt es genau einen Schüssel $k_{m,c}$ mit

$$c = Enc_{k_{m,c}}(m).$$

- Daraus folgt für alle m, m'

$$\begin{aligned} Ws[K = k_{m,c}] &= Ws[C = c \mid M = m] \\ &= Ws[C = c \mid M = m'] = Ws[K = k_{m',c}]. \end{aligned}$$

- D.h. es gilt $Ws[K = k] = \frac{1}{|\mathcal{K}|}$ für alle $k \in \mathcal{K}$.

Ununterscheidbarkeit von Chiffretexte

- Äquivalente Definition zur perfekten Sicherheit.
- Experiment: ein Gegner hört passiv Schlüsseltexte ab und versucht zu raten welche von zwei verschiedene Nachrichten verschlüsselt worden ist.
- Sei $\mathcal{A}$ der Gegner und $m_0, m_1 \in \mathcal{M}$.
- Einer dieser Nachrichten wird gleichverteilt gewählt und verschlüsselt mit einem zufälligen Schlüssel.
- $\mathcal{A}$ bekommt das Ergebnis der Verschlüsselung und ratet nun welche Nachricht verschlüsselt worden ist.
- Ein Verschlüsselungsschema ist perfekt ununterscheidbar falls kein Gegner $\mathcal{A}$ das Ergebnis mit einer Wahrscheinlichkeit größer als 1/2 raten kann.

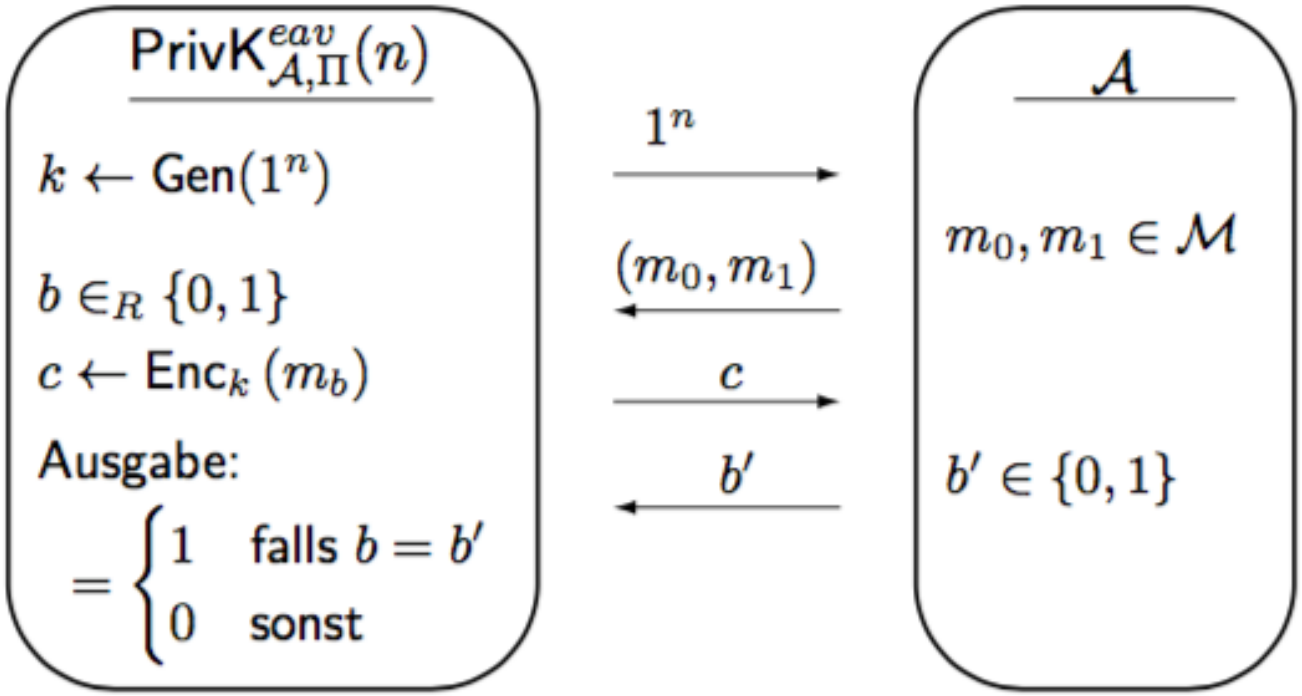
Ununterscheidbarkeit von Chiffretexte

Spiel: Ununterscheidbarkeit von Chiffretexten $PrivK^{eav}_{\mathcal{A},\Pi}(n)$

Sei Π ein Verschlüsselungsverfahren und $\mathcal{A}$ ein Angreifer

1. $(m_0, m_1) \leftarrow \mathcal{A}$.
2. $k \leftarrow Gen$
3. Wähle $b \in \{0,1\}.b' \leftarrow \mathcal{A}(Enc_k(m_b))$.
4. $$PrivK^{eav}_{\mathcal{A},\Pi}(n) = \begin{cases} 1 \text{ für } b = b' \\ 0 \text{ sonst} \end{cases}$$

Spiel


$\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n)$
$k \leftarrow \mathsf{Gen}(1^n)$
$b \in_R \{0,1\}$
$c \leftarrow \mathsf{Enc}_k(m_b)$
Ausgabe:
$= \begin{cases} 1 & \text{falls } b = b' \\ 0 & \text{sonst} \end{cases}$
1^n
(m_0, m_1)
c
b'
$\mathcal{A}$
$m_0, m_1 \in \mathcal{M}$
$b' \in \{0,1\}$

Ununterscheidbarkeit von Chiffretexte

Definition

Das Verschlüsselungsverfahren Π *ist perfekt ununterscheidbar, falls für alle* $\mathcal{A}$ *gilt*

$$Pr[PrivK^{eav}_{\mathcal{A},\Pi}(n)] = \frac{1}{2}.$$

Lemma

Π *ist genau dann perfekt sicher, wenn es perfekt ununterscheidbar ist.*

Beispiel

- Vigenere ist nicht perfekt ununterscheidbar.
- Sei Π die Vigenere Chiffre, $\mathcal{M}$ besteht aus Wörter über einem binären Alphabet und der Schlüssel ist gleichverteilt in $\{1, 2\}$.
- Wir basteln einen Gegner $\mathcal{A}$ mit $Pr[PrivK^{eav}_{\mathcal{A},\Pi}(n)] > \frac{1}{2}$.
- $\mathcal{A}$ wählt $m_0 = aa$ und $m_1 = ab$.
- Beim Erhalten des Schlüsseltexts $c = c_1 c_2$:
  - Falls $c_1 = c_2$ rate 0
  - Sonst 1

Beispiel

- Wir berechnen nun $Pr[PrivK^{eav}_{\mathcal{A},\Pi}(n)]$

$$
\begin{aligned}
&Pr[PrivK^{eav}_{\mathcal{A},\Pi}(n)] \\
&= \frac{1}{2} \cdot Pr[PrivK^{eav}_{\mathcal{A},\Pi}(n) = 1 \mid b = 0] + \frac{1}{2} \cdot Pr[PrivK^{eav}_{\mathcal{A},\Pi}(n) \mid b = 1] \\
&= \frac{1}{2} \cdot Pr[\mathcal{A} \text{ ratet } 0 \mid b = 0] + \frac{1}{2} \cdot Pr[\mathcal{A} \text{ ratet } 1 \mid b = 1],
\end{aligned}
$$

wobei b das gleichverteilte Bit ist, welches die Messages selektiert.

BEISPIEL

- $\mathcal{A}$ ratet 0 gdw. $c_1 = c_2, c = c_1 c_2$.
- Falls $b = 0$ wir $m_0 = aa$ verschlüsselt dann ist $c_1 = c_2$ falls
  1. Die Schlüssellänge =1 ist
  2. Die Schlüssellänge = 2 und beide Zeichen im Schlüssel sind gleich.
- Der erste Fall kommt mit Wahrscheinlichkeit $\frac{1}{2}$ vor und ds zweite mit Wahrscheinlichkeit $\frac{1}{2} \cdot \frac{1}{26}$.
- Es gilt

$$Pr[\mathcal{A} \text{ ratet } 0 \mid b = 0] = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{26} \cong 0.52.$$

Beispiel

- Für $b = 1$ dann ist $c_1 = c_2$ genau dann, wenn ein Schlüssel der Länge 2 gewählt wird und das erste Zeichen im Schlüssel hat einen genau mit 1 größerem Wert als das zweite. Die geschieht mit Wahrscheinlichkeit $\frac{1}{2} \cdot \frac{1}{26}$.
- Es gilt

$$Pr[\mathcal{A} \text{ ratet } 1 \mid b = 1] = 1 - Pr[\mathcal{A} \text{ ratet } 0 \mid b = 1] = 1 - \frac{1}{26} \cdot \frac{1}{26} \cong 0.98.$$

- Es gilt

$$Pr[PrivK^{eav}_{\mathcal{A},\Pi}(n)] = \frac{1}{2} \cdot \left(\frac{1}{2} + \frac{1}{2} \cdot \frac{1}{26} + 1 - \frac{1}{2} \cdot \frac{1}{26} \right) = 0.75 > \frac{1}{2},$$

- Vigenere ist nicht perfekt ununterscheidbar.

Data Encryption Standard

DES

- Viele Jahre lang war das DES Verfahren der Verschlüsselungsstandard in den USA
- Das einfache DES gilt nicht mehr als sicher
- Triple-DES gilt weiterhin als sicher
- DES wichtiges Vorbild für neue symmetrische Verfahren

Feistel-Chiffre

- Blockchiffre mit Alphabet $\{0, 1\}$
- t Blocklänge
- Verschlüsselungsfunktion zum Schlüssel K ist f_K

Feistel-Chiffre

- Schlüsselraum $\mathcal{K}$
- Rundenzahl $r \geq 1$
- Wähle eine Methode, die aus einem Schlüssel $k \in \mathcal{K}$ eine Folge $K_1, \ldots, K_r$ von Rundenschlüsseln konstruiert. Die Rundenschlüssel gehören zum Schlüsselraum der zugrundeliegenden Blockchiffre.

Verschlüsselungsfunktion E_k zum Schlüssel $k \in \mathcal{K}$

- Sei p ein Klartext der Länge $2t$.
- Teile p in zwei Hälften der Länge t auf.
- $p = (L_0, R_0)$. Dabei ist L_0 die linke Hälfte des Klartextes, und R_0 ist seine rechte Haälfte.
- Konstruiere eine Folge $((L_i, R_i))_{1 \leq i \leq r}$ nach folgender Vorschrift:

$$(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus f_{K_i}(R_{i-1})), 1 \leq i \leq r.$$

Verschlüsselungsfunktion E_k zum Schlüssel $k \in \mathcal{K}$

- Setze
$$E_k(L_0, R_0) = (R_r, L_r).$$
- Die Sicherheit der Feistelchiffre hängt natürlich zentral von der Sicherheit der internen Blockchiffre ab.
- Deren Sicherheit wird aber durch iterierte Verwendung noch gesteigert.

Entschlüsselung

$$(R_{i-1}, L_{i-1}) = (L_i, R_i \oplus f_{K_i}(L_i)), 1 \leq i \leq r.$$

- Daher kann man unter Verwendung der Schlüsselfolge $(K_r, K_{r-1}, \ldots, K_1)$ in r Runden das Paar (R_0, L_0) aus dem Schlüsseltext (R_r, L_r) zurückgewinnen.
- Die Feistel-Chiffre wird also entschlüsselt, indem man sie mit umgekehrter Schlüsselfolge auf den Schlüsseltext anwendet.

Der DES-Algorithmus: Chronologie

- 15. Mai 1973 Das NBS veröffentlicht eine erste Ausschreibung für ein standardisiertes Verschlüsselungsverfahren
- 27. August 1974 Das NBS veröffentlicht eine zweite Ausschreibung für ein standardisiertes Verschlüsselungsverfahren
- 17. März 1975 DES wird im `Federal Register` veröffentlicht
- August 1976 Erster Workshop zu DES
- September 1976 Zweiter Workshop, welcher die mathematischen Grundlagen von DES behandelt

Der DES-Algorithmus: Chronologie

- November 1976 DES wird als Standard zugelassen
- 15. Januar 1977 DES wird als FIPS-Standard `FIPS PUB 46` veröffentlicht
- 1983 DES wird das erste Mal neu bestätigt
- 1986 Videocipher II, ein auf DES basierendes Verschlüsselungssystem für Fernsehsatelliten wird von der HBO verwendet
- 22. Januar 1988 DES wird als `FIPS 46-1` revalidiert, welches FIPS PUB 46 ersetzt
- 1992 Biham und Shamir publizieren den ersten theoretischen Angriff mit gegenüber der Brute-Force-Methode verminderter Komplexität: die differentielle Kryptanalyse. Dieser Angriff erfordert jedoch unrealistische 2^{47} frei gewählte Klartexte.

Der DES-Algorithmus: Chronologie

- 30. Dezember 1993 DES wird ein drittes Mal bestätigt, diesmal als `FIPS 46-2`
- 1994 Die erste experimentelle Kryptoanalyse von DES wird mittels linearer Kryptoanalyse durchgeführt (Matsui, 1994)
- Juni 1997 Das DESCHALL-Projekt bricht erstmals öffentlich eine mit DES verschlüsselte Nachricht
- Juli 1998 Der DES-Knacker `Deep Crack` der Electronic Frontier Foundation bricht einen DES-Schlüssel binnen 56 Stunden
- Januar 1999 Deep Crack und distributed.net brechen in einer Kooperation einen DES-Schlüssel in 22 Stunden und 15 Minuten
- 25. Oktober 1999 DES wird ein viertes Mal in Gestalt des `FIPS 46-3` bestätigt. Dieser gibt als bevorzugte Anwendung 3DES an und erlaubt DES selbst nur für den Einsatz in veralteten Systemen

Der DES-Algorithmus: Chronologie

- 26. November 2001 Der Advanced Encryption Standard (AES) wird als `FIPS 197` publiziert
- 26. Mai 2002 Der AES tritt in Kraft
- 26. Juli 2004 Im *Federal Register* wird die Absetzung des FIPS 46-3 und verwandter Standards empfohlen
- 19. Mai 2005 NIST setzt den FIPS 46-3 außer Kraft
- März 2006 Der FPGA-basierte Parallelrechner COPACOBANA kostet weniger als 10.000 Dollar (Materialkosten) und bricht DES in weniger als 9 Tagen
- Nov. 2008 Die Weiterentwicklung des FPGA-basierten Parallelrechners COPACOBANA, die RIVYERA, bricht DES erstmals in weniger als einem Tag

Der DES-Algorithmus

- Alphabet $\{0,1\}$ und Blocklänge 64
- Klartext- und Schlüsseltextraum des DES ist $\mathcal{P} = \mathcal{C} = \{0,1\}^{64}$.
- Die DES-Schlüssel sind Bitstrings der Länge 64, die folgende Eigenschaft haben:
- Teilt man einen String der Länge 64 in acht Bytes auf, so ist jeweils das letzte Bit eines jeden Bytes so gesetzt, dass die Quersumme aller Bits im betreffenden Byte ungerade ist. Es ist also

$$\mathcal{K} = \{(b_1, \ldots, b_{64}) \in \{0,1\}^{64} \mid \sum_{i=1}^{8} b_{8k+i} \equiv 1 \mod 2, 0 \leq k \leq 7\}.$$

Der DES-Algorithmus

- Die ersten sieben Bits eines Bytes in einem DES-Schlüssel legen das achte Bit fest.
- Dies ermöglicht Korrektur von Speicher- und Übertragungsfehlern.
- In einem DES-Schlüssel sind also nur 56 Bits frei wählbar.
- Insgesamt gibt es $2^{56} \sim 7.2 \cdot 10^{16}$ viele DES-Schlüssel.
- Der DES-Schlüssel für Ver- und Entschlüsselung ist derselbe → symmetrisches Verfahren

Beispiel

- Ein gültiger DES Schlüssel ist hexadezimal geschrieben 133457799*BBCDFF*1.
- Binär:

0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

Der DES-Algorithmus

- Feistel-Chiffre
- Sei p ein Eingabetext
- Im ersten Schritt wird eine initiale Permutation IP angewandt
- Dies ist eine für das Verfahren fest gewählte, vom Schlüssel unabhängige, Bitpermutation auf Bitvektoren der Länge 64.

Die initiale Permutation

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Abbildung 1: Die initiale Permutation *IP*

Die initiale Permutation

- Ist $p \in \{0,1\}^{64}, p = p_1 p_2 p_3 \ldots p_{64}$, dann ist $IP(p) = p_{58} p_{50} p_{42} \ldots p_7$.
- Auf das Ergebnis dieser Permutation wird eine 16-Runden Feistel-Chiffre angewendet.
- Zuletzt wird die Ausgabe als

$$c = IP^{-1}(R_{16} L_{16})$$

erzeugt

Die interne Blockchiffre

- Alphabet $\{0, 1\}$
- Blocklänge 32
- Schlüsselraum $\{0, 1\}^{48}$

VERSCHLÜSSELUNG

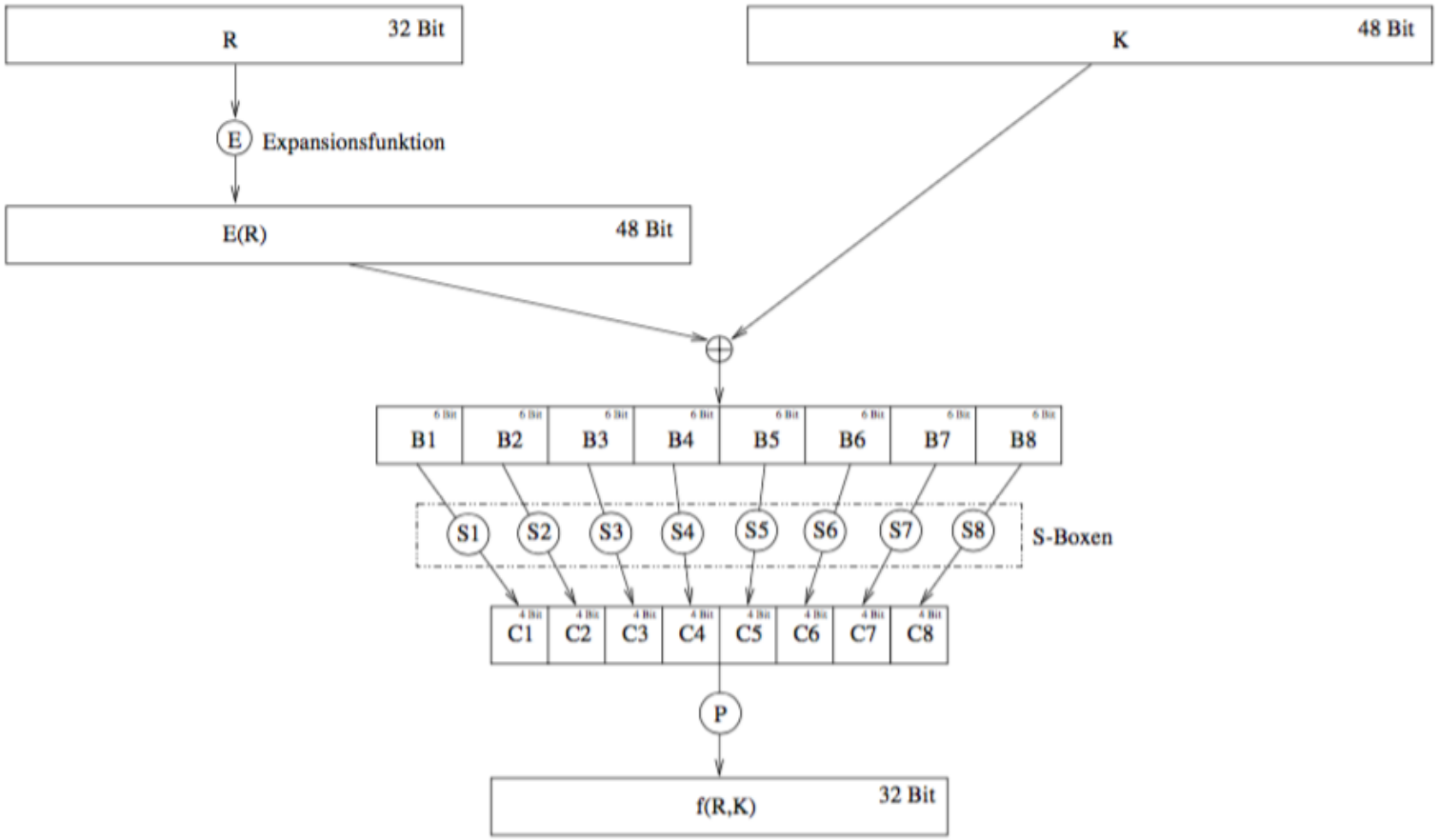


Abbildung 2: Schema der f-Funktion im DES

VERSCHLÜSSELUNG

- Das Argument $R \in \{0,1\}^{32}$ wird mittels einer Expansionsfunktion $E\colon \{0,1\}^{32} \to \{0,1\}^{48}$ verlängert.

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

- Ist $R = R_1R_2 \ldots R_{32}$, dann ist $E(R) = R_{32}R_1R_2 \ldots R_{32}R_1$.

VERSCHLÜSSELUNG

- Anschließend wird der String $E(R) \oplus K$ gebildet und in 8 Blöcke $B_i, 1 \leq i \leq 8$, der Länge 6 aufgeteilt.
- Es wird also

$$E(R) \oplus K = B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8$$

gebildet mit $B_i \in \{0,1\}^6, 1 \leq i \leq 8$.

VERSCHLÜSSELUNG

- Im nächsten Schritt werden Funktionen

$$S_i : \{0,1\}^6 \to \{0,1\}^4, 1 \leq i \leq 8$$

verwendet (die sogenannten S-Boxen), die unten noch genauer beschrieben sind.

- Mit diesen Funktionen wird der String

$$C = C_1C_2C_3C_4C_5C_6C_7C_8$$

berechnet, wobei $C_i = S_i(B_i), 1 \leq i \leq 8$, ist.

- Er hat die Länge 32.

VERSCHLÜSSELUNG

- Dieser Bitstring wird gemäß der Permutation P permutiert.
- Das Ergebnis ist $f_K(R)$.

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Die S-Boxen

- Jede S-Box wird durch eine Tabelle mit vier Zeilen und 16 Spalten beschrieben.
- Für einen String $B = b_1b_2b_3b_4b_5b_6$ wird der Funktionswert $S_i(B)$ folgendermaßen berechnet.
- Man interpretiert die natürliche Zahl mit Binärentwicklung b_1b_6 als Zeilenindex und die natürliche Zahl mit Binärentwicklung $b_2b_3b_4b_5$ als Spaltenindex.
- Den Eintrag in dieser Zeile und Spalte der S-Box stellt man binär dar und füllt diese Binärentwicklung vorne so mit Nullen auf, dass ihre Länge 4 wird.
- Das Ergebnis ist $S_i(B)$.

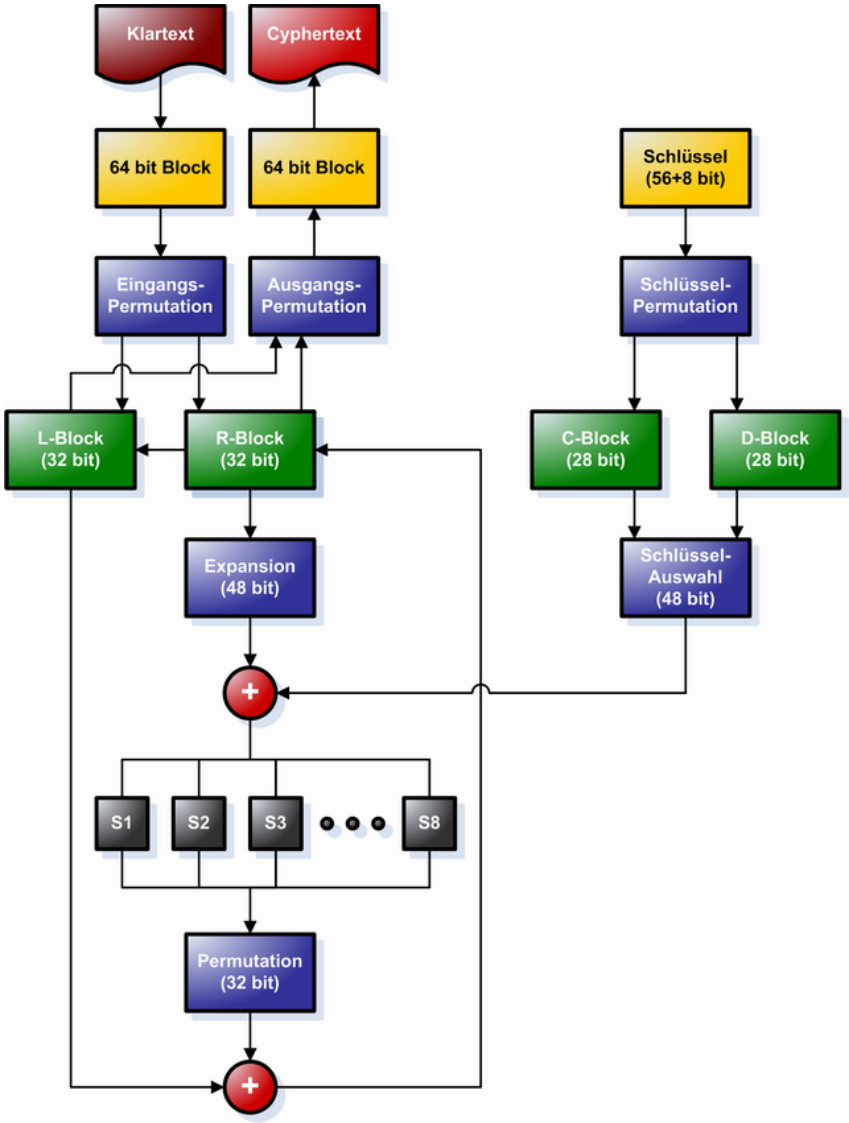
Die S-Boxen

Zeile	Spalte															
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
S_1																
[0]	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
[1]	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
[2]	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
[3]	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2																
[0]	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
[1]	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
[2]	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
[3]	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3																
[0]	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
[1]	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
[2]	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
[3]	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4																
[0]	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
[1]	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
[2]	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
[3]	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Die S-Boxen

S_5																
[0]	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
[1]	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
[2]	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
[3]	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6																
[0]	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
[1]	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
[2]	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
[3]	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7																
[0]	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
[1]	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
[2]	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
[3]	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8																
[0]	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
[1]	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
[2]	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
[3]	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

DES Schema


Klartext
Cyphertext
64 bit Block
64 bit Block
Schlüssel (56+8 bit)
Eingangs-Permutation
Ausgangs-Permutation
Schlüssel-Permutation
L-Block (32 bit)
R-Block (32 bit)
C-Block (28 bit)
D-Block (28 bit)
Expansion (48 bit)
Schlüssel-Auswahl (48 bit)
+
S1
S2
S3
S8
Permutation (32 bit)
+

Beispiel

- Wir berechnen $S_1(001011)$.
- Das erste Bit des Argumentes ist 0 und und das letzte Bit ist 1.
- Also ist der Zeilenindex die ganze Zahl mit Binärentwicklung 01, also 1.
- Die vier mittleren Bits des Argumentes sind 0101.
- Dies ist die Binärentwicklung von 5.
- Also ist der Spaltenindex 5.
- In der ersten S-Box steht in Zeile 1 und Spalte 5 die Zahl 2.
- Die Binärentwicklung von 2 ist 10.
- Also ist $S_1(001011) = 0010$.

Die Rundenschlüssel

- Sei ein DES-Schlüssel $k \in \{0,1\}^{64}$ gegeben.
- Daraus werden Rundenschlüssel $K_i, 1 \leq i \leq 16$, der Länge 48 generiert.
- Dazu definiert man $v_i, 1 \leq i \leq 16$, folgendermaßen:

$$v_i = \begin{cases} 1 & \text{für } i \in \{1,2,9,16\} \\ 2 & \text{andernfalls.} \end{cases}$$

Die Rundenschlüssel

- Zwei Funktionen

$$PC1\colon \{0,1\}^{64} \to \{0,1\}^{28}\times\{0,1\}^{28}, PC2\colon \{0,1\}^{28}\times\{0,1\}^{28} \to \{0,1\}^{48}$$

PC1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Die Rundenschlüssel

- Setze $(C_0, D_0) = PC1(k)$.
- Für $1 \leq i \leq 16$ berechne K_i folgendermaßen.
- Setze C_i auf den String, den man durch einen zirkulären Linksshift um v_i Stellen aus C_{i-1} gewinnt und D_i auf den String, den man durch einen zirkulären Linksshift um v_i Stellen aus D_{i-1} gewinnt.
- Berechne dann $K_i = PC2(C_i, D_i)$.

Die Rundenschlüssel

- Die Funktion PC1 bildet einen Bitstring k der Länge 64 auf zwei Bitstrings C und D der Länge 28 ab. Dies geschieht gemäß der Tabelle PC1.
- Die obere Hälfte der Tabelle beschreibt, welche Bits aus K in C verwendet werden.
- Ist $k = k_1 k_2 \ldots k_{64}$, dann ist $C = k_{57} k_{49} \ldots k_{36}$.
- Die untere Hälfte dient der Konstruktion von D, also $D = k_{63} k_{55} \ldots k_4$.
- Die Funktion PC2 bildet umgekehrt ein Paar (C, D) von Bitstrings der Länge 28 (also einen Bitstring der Länge 56) auf einen Bitstring der Länge 48 ab.
- Die Funktion wird in Tabelle PC2 dargestellt.
- Der Wert $PC2(b_1 \ldots b_{56})$ ist $b_{14} b_{17} \ldots b_{32}$.

Beispiel

- $p = 0123456789ABCDEF$. Dessen Binärentwicklung ist

0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

Beispiel

- Die Anwendung von IP ergibt

1	1	0	0	1	1	0	0
0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0
1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0

BEISPIEL

- In der ersten Zeile von $IP(p)$ steht die umgekehrte zweite Spalte von p, in der zweiten Zeile von $IP(p)$ steht die umgekehrte vierte Spalte von p usw.
- Damit ist

$$L_0 = 11001100000000001100110011111111,$$

$$R_0 = 11110000101010101111000010101010.$$

- Sei $k = 133457799BBCDFF1$ der DES Schlüssel.

0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

BEISPIEL

- Daraus berechnen wir den ersten Rundenschlüssel.
- Es ist

$$C_0 = 11110000110011001010101111,$$

$$D_0 = 01010101011001100111100011111$$

$$C_1 = 11100001100110010101011111,$$

$$D_1 = 10101010110011001111100011110$$

und daher

$$K_1 = 000110110000001011101111111111000111000001110010.$$

Beispiel

- $E(R0) \oplus K_1 =$
  $011000010001011110111010100001100110010100100111,$

  $$f_{K_1}(R_0) = 00100011010010101010100110111011$$

  und schließlich

  $$R_1 = 11101111010010100110010101000100.$$

- Die anderen Runden werden analog berechnet.

Weitere Links

DES online

Plaintext Attacks

Die Chosen Plaintext Attack bei DES beruht darauf, dass ein komplementärer Schlüssel den komplementären Chiffretext bzw. der komplementäre Klartext denselben Chiffretext ergibt, was allerdings keine Besonderheit darstellt, da die Subkeys xor-verknüpft werden. Das einzige, was sich daraus ergibt, ist, dass ein Kryptoanalytiker nur die Hälfte der möglichen Schlüssel zu testen hat, also 2^{55} statt 2^{56}.
Eli Biham und Adi Shamir haben gezeigt, dass es eine Known Plaintext Attack derselben Komplexität gibt, die zumindesxt 2^{33} Klartexte benötigt. Diese *Sicherheitslücke* ist eigentlich keine solche, weil es sehr unwahrscheinlich ist, dass in einem Klartext das entsprechende Komplement enthalten ist, bzw. kann man Benutzer davor warnen, komplementäre Schlüssel zu verwenden.

Sicherheit des DES

- Es ist dazu wichtig, festzustellen, dass die DES-Verschlüsselungsfunktionen nicht abgeschlossen unter Hintereinanderausführung sind.
- Sie bilden also keine Untergruppe der Permutationsgruppe $S_{64!}$.
- Würden die DES-Verschlüsselungsfunktionen eine Gruppe bilden, dann könnte man für zwei DES-Schlüssel k_1, k_2 einen dritten DES-Schlüssel k_3 finden, für den $DES_{k_1} \circ DES_{k_2} = DES_{k_3}$ gelten würde.
- Mehrfachverschlüsselung würde also keinen Sicherheitsvorteil bieten.
- Es ist bekannt, dass die 2^{56} DES-Verschlüsselungsfunktionen eine Gruppe erzeugen, die wenigstens die Ordnung 10^{2499} hat.

Differentielle Kryptanalyse

Eli Biham und Adi Shamir haben 1990 die differentielle Kryptoanalyse eingeführt, die sich speziell mit Paaren von verschlüsselten Texten, deren Klartexte wesentliche Unterschiede aufweisen, befasst. Diese Methode beobachtet die Entwicklung dieser Differenzen, die sich durch jede DES-Runde verändert, wenn beide Klartexte mit dem selben Schlüssel verschlüsselt worden sind.
Man muss nur zwei Klartexte mit einer fixen Differenz wählen (Differenz = XOR) und dann entsprechend der Differenz der Chiffretexts den Schlüsseln verschiedene Wahrscheinlichkeiten zuordnen. Bei genügend vielen Analysen wird ein Schlüssel sich als der wahrscheinlichste herausstellen, und dieser Schlüssel ist der richtige!

Lineare Kryptanalyse

Lineare Kryptoanalyse ist ein anderes Verfahren, dass von Mitsuru Matsui erfunden wurde. Diese Attacke verwendet lineare Approximation, um die Aktionen eines Blockchiffrieralgorithmus zu beschreiben. Das heißt nichts anderes, als dass einige Klartexte xor-verknüpft und einige verschlüsselte Texte xor-verknüpft und die Ergebnisse wiederum xor-verknüpft werden, und daraus erhält man ein Bit, das der xor-Verknüpfung einiger Schlüsselbits entspricht. Das ist eine lineare Annäherung und enthält eine gewisse Wahrscheinlichkeit p. Wenn $p \neq 1/2$ kann diese Eigenschaft ausgenutzt werden. Wie kann man nun DES damit knacken?

Lineare Kryptanalyse

1. Finde gute Rundenapproximationen und verknüpfe diese.
2. Analysiere die S-Boxes (6 Bit Input, 4 Bit Output)

Die Inputbits können 63 verschiedene, sinnvolle Kombinationen haben ($2^6 - 1$), die Outputbits haben hingegen 15 verschiedene, sinnvolle Kombinationen. Für jede S-Box kann nun eine Wahrscheinlichkeit berechnet werden, sodass für einen zufälligen Input eine Input - xor - Kombination gleich einer Output - xor - Kombination ist. Falls es eine solche Eigenschaft gibt, wird die lineare Kryptoanalyse funktionieren.