Der Data Encryption Standard (DES)

Im Folgenden werden die Funktionsweise des DES beschreiben und darüber hinausgehende Aspekte während des Chiffrierprozesses erläutert.

Der Fokus liegt auf der Visualisierung aller Chiffrierrunden $(m_{i-1}, m_i)_{i=1,\dots,17}$ mit $m_i \in \{0,1\}^{32}$, aller Rundenschlüssel $(K_i)_{i=1,\dots,16}$ mit $K_i \in \{0,1\}^{48}$ sowie einiger statistischer Daten, die die Avalanche-Fähigkeiten des DES aufzeigen.

Außerdem wird – bzgl. eines gegebenen Schlüssels $k \in \{0,1\}^{64}$ – die Berechnung von Fixpunkten $p \in \{0,1\}^{64}$ (definiert durch $DES(k,p) = p$) sowie von Anti-Fixpunkten (definiert durch $DES(k,p) = \overline{p}$, wobei $\overline{(x_i)} := (1 - x_i)$ bitweise Komplementierung ist) beschrieben. Innerhalb dieses Kontextes erläutern wir auch die Rolle der speziellen Schlüssel $k(5j)_{j=0,1,2,3}$ and $k(3n)_{n=1,2,3,4}$.

(1) Verschlüsselung und Entschlüsselung

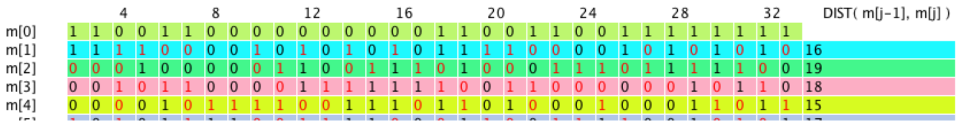
Der DES-Algorithmus führt eine bijektive Abbildung $DES : \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ durch, die einen Block von 64 bit Länge (genannt Klartext) in einen Block von 64 bit Länge (genannt Geheimtext) verschlüsselt.

Dreh- und Angelpunkt bei der Erzeugung sukzessiver Runden $(m_{i-1}, m_i)_{i=1,\dots,17}$ (bei der Chiffrierung als auch Dechiffrierung) ist die Gleichung $m_{i+1} := m_{i-1} \oplus f(m_i, K_i)$, die bei der Rundenprozessierung $(m_{i-1}, m_i) \xrightarrow{K_i} (m_i, m_{i-1} \oplus f(m_i, K_i))$ verwendet wird. Die Bijektivität des DES resultiert aus der Bijektivität von $\psi : \{0,1\}^{32} \times \{0,1\}^{32} \rightarrow \{0,1\}^{32} \times \{0,1\}^{32} : (a,b) \rightarrow (b, a \oplus f(a,K))$, wobei f irgendeine Abbildung, K irgendein Rundenschlüssel ist, und $IP : \{0,1\}^{64} \rightarrow \{0,1\}^{64}$. IP wird *initiale Permutation* genannt. Die Bijektivität sorgt auch dafür, dass die gesamte Sequenz $(m_{i-1}, m_i)_{i=1,\dots,17}$ bereits durch Kenntnis irgendeiner Runde $(m_{i_0-1}, m_{i_0}),\ 1 \leq i_0 \leq 17$ (und dem Schlüssel k) definiert ist.

Verschlüsselung eines Klartextes p

$$p \xrightarrow{IP} (m_0, m_1) \xrightarrow{K_1} (m_1, m_2) \xrightarrow{K_2} \dots \xrightarrow{K_8} (m_8, m_9) \rightarrow \dots \xrightarrow{K_{16}} (m_{16}, m_{17}) \xrightarrow{Flip} (m_{17}, m_{16}) \xrightarrow{IP^{-1}} c$$

Weil (in benachbarten Runden) die rechte Hälfte einer Vorgängerrunde identisch ist mit der linken Hälfte der aktuellen Runde (dadurch tritt Redundanz auf), genügt es, lediglich $(m_i)_{i=0,\dots,17}$ anstatt $(m_{i-1}, m_i)_{i=1,\dots,17}$ darzustellen (vgl. Abb. 1). Die Spalte "DIST" am rechten Rand zeigt die Hamming-Distanzen $dist(m_{i-1}, m_i)_{i=1,\dots,17}$. Man kann sehen, dass jedesmal $dist(m_{i-1}, m_i) \approx 16$ gilt.



				4				8				12				16				20				24				28				32	DIST(m[j-1], m[j])
m[0]	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	
m[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	16
m[2]	0	0	0	1	0	0	0	0	0	1	1	0	0	1	1	1	0	1	0	0	0	1	1	1	0	1	1	1	1	1	0	0	19
m[3]	0	0	1	0	1	1	0	0	0	0	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0	18
m[4]	0	0	0	0	1	0	1	1	1	1	0	0	1	1	1	0	1	1	0	1	0	0	0	1	0	0	0	1	1	0	1	1	15

Abbildung 1: Die ersten 3 Chiffrierrunden $(m_0, m_1) \xrightarrow{K_1} \dots \xrightarrow{K_3} (m_3, m_4)$

Entschlüsselung eines Geheimtextes c

$$c \overset{IP}{\rightarrow} (m_{17}, m_{16}) \overset{K_{16}}{\rightarrow} (m_{16}, m_{15}) \overset{K_{15}}{\rightarrow} \ldots \overset{K_9}{\rightarrow} (m_9, m_8) \rightarrow \ldots \overset{K_1}{\rightarrow} (m_1, m_0) \overset{Flip}{\rightarrow} (m_0, m_1) \overset{IP^{-1}}{\rightarrow} p$$

Der Dechiffrier-Prozess verläuft logisch in gleicher Weise wie der Chiffrier-Prozess. Er startet mit Eingabe $c = IP^{-1}(m_{17}, m_{16})$, verwendet jedoch die Rundenschlüssel in umgekehrter Reihenfolge $(K_{16}, K_{15}, \ldots, K_2, K_1)$, wohingegen die Chiffrier-Reihenfolge $(K_1, K_2, \ldots, K_{15}, K_{16})$ ist.

(2) Erzeugung der Rundenschlüssel $(K_i)_{i=1,\ldots,16}$ aus einem Schlüssel k

Abbildung 2: Farbmuster der Rundenschlüssel $K_1, \ldots, K_4$, erzeugt aus $k(9)$.

Jeder Satz $(K_i)_{i=1,\ldots,16}$ von Rundenschlüsseln wird aus einem Schlüssel $k := (k_i)_{i=1,\ldots,64}$ erzeugt und generiert Rundenchiffrate mit $m_{i+1} = m_{i-1} \oplus f(m_i, K_i)$.

Wir beschreiben die Erzeugung der $(K_i)_{i=1,\ldots,16}$ unter Verwendung von $PC_1 : \{0,1\}^{64} \rightarrow \{0,1\}^{28} \times \{0,1\}^{28}$, $PC_2 : \{0,1\}^{28} \times \{0,1\}^{28} \rightarrow \{0,1\}^{48}$ und $(v_1, \ldots, v_{16}) \in \{1,2\}^{16}$ (komplette Definitionen finden sich in der Literatur).

Darüberhinaus erläutern wir die Rolle der $(C_i, D_i)_{i=0,\ldots,16}$, die bei der Erzeugung der Rundenschlüssel lediglich Zwischenprodukte darstellen:

(a) Für $k \in \{0,1\}^{64}$ sei $(C_0, D_0) := PC_1(k) := \overbrace{k_{57}k_{49}\ldots k_{36}}^{C_0}, \overbrace{k_{63}k_{55}\ldots k_4}^{D_0}$
(b) Für $i = 1, \ldots, 16$ sei $C_i :=$ zyklische Linksverschiebung von C_{i-1} um v_i Stellen. Analoges gilt für die Erzeugung der (D_i).

Weil $v_1 + \ldots + v_{16} = 28$, erzeugen 16 Linksverschiebungen wieder C_0 (bzw. D_0). Wegen $K_i := PC_2(C_i, D_i)$ selektiert PC_2 spezielle Positionen aus (C_i, D_i), um K_i zu erzeugen.

Auf Seite 3 sehen wir unter (3), dass jeder Anti-/Fixunkt einem von acht speziellen Schlüsseln zugeordnet werden kann, die als *schwache Schlüssel* $:= \{k(0), k(5), k(10), k(15)\}$ oder als *semi-schwache Schlüssel* $:= \{k(3), k(6), k(9), k(12)\}$ Eingang in die Literatur fanden. Vermöge dieser Zuordnung spielen *semi-/schwache Schlüssel* eine spezielle Rolle unter den Schlüsseln. Die aus ihnen erzeugten Rundenschlüssel sowie deren redundante Informationen werden deshalb im Panel "Rundenschlüssel K_i" über ein blau-gelbes Farbmuster visualisiert (vgl. Abb. 2). Für alle anderen Schlüssel spielen die Farbmuster keine Rolle, wohl aber deren Bit-Werte.

Abb. 2 zeigt die farbige Matrix M mit Rundenschlüssel K_i in Reihe i.

Die Bedeutung der Farbmuster: Innerhalb M spezifizieren wir die 12 Blockmatrizen $[(1, 4k+1), (16, 4k+4)]_{k=0,1,\ldots,11}$. Für jeden dieser Blöcke gilt (bei Zugrundelegung von *semi-/schwachen Schlüsseln*), dass Vierertupel mit derselben

Farbe identisch sind.

Beispiel: Chiffriere in "Key/Plaintext" die Eingabe $(k(9), p)$ mit beliebigem Klartext p, und beobachte das Panel "Rundenschlüssel K_i". Die dortige 1. Blockmatrix $[(1,1),(16,4)]$ zeigt $(0,1,1,0)$ in blau und $(1,0,0,1)$ in gelb. Dagegen zeigt die 3. Blockmatrix $[(1,9),(16,13)]$ das Tupel $(1,0,1,0)$ in blau und $(0,1,1,0)$ in gelb.

(3) Berechnung von Fixpunkten und Anti-Fixpunkten

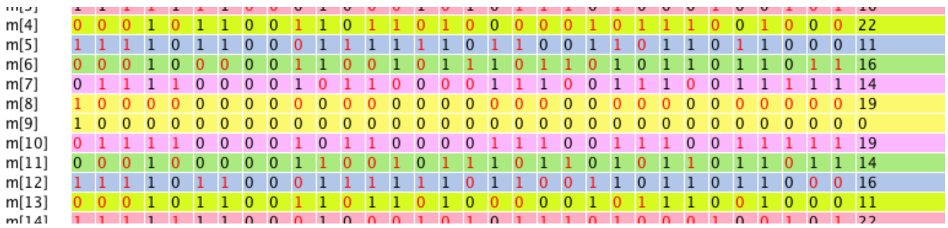
Die Behandlung der Anti-/Fixpunkte komplettiert die Diskussion des DES. Das Applet berechnet alle bis heute (Dez. 2011) bekannten Anti-/Fixpunkte. Die *schwachen* (bzw. *semi-schwachen*) *Schlüssel* sind zudem die Lösungen der Probleme $DES(k,.) = [DES(k,.)]^{-1}$ (bzw. $DES(k,.) = [DES(k',.)]^{-1}$).

Fixpunkte

Ein Fixpunkt p bzgl. einem Schlüssel k ist definiert durch $DES(k,p) = p$. Bis heute weiß man nicht genau, wie viele Fixpunkte es gibt (bzgl. einem Schlüssel k). Allerdings weiß man, dass es 2^{32} Fixpunkte bzgl. jedem der vier *schwachen Schlüssel* $k(5j)_{j=0,\ldots,3}$ gibt. Um die Gründe hierfür zu verstehen, werfen wir einen Blick auf den Chiffrierprozess mit seinen Rundenchiffraten.

$$p \overset{IP}{\to} (m_0, m_1) \overset{K_1}{\to} (m_1, m_2) \overset{K_2}{\to} \ldots \overset{K_8}{\to} (m_8, m_9) \to \ldots \overset{K_{16}}{\to} (m_{16}, m_{17}) \overset{Flip}{\to} (m_{17}, m_{16}) \overset{IP^{-1}}{\to} c$$

Wendet man die sequenzerzeugende Gleichungen $m_{i+1} = m_{i-1} \oplus f(m_i, K_i), i = 1, \ldots, 16$ auf einen Fixpunkt (p,k) an, so muss u.a. gelten: $p = c$ und $(m_0, m_1) = (m_{17}, m_{16})$ und $m_{15} = m_{17} \oplus f(m_{16}, K_{16}) = m_0 \oplus f(m_1, K_{16})$. Falls also $K_{16} = K_1$, so folgt $m_{15} = m_0 \oplus f(m_1, K_1) = m_2$. Durchführung des nächsten Schrittes liefert $m_{14} = m_{16} \oplus f(m_{15}, K_{15}) = m_1 \oplus f(m_2, K_{15})$. Annahme der ähnlichen Relation $K_{15} = K_2$ induziert $m_{14} = m_1 \oplus f(m_2, K_2) = m_3$. Wiederholte Anwendung der Annahme $K_{17-i} = K_i$ for $i = 1, \ldots, 16$ liefert schließlich die Symmetrie $m_{17-k} = m_k$ for $k = 0, 1, \ldots, 17$. Speziell gilt damit $m_8 = m_9$.



m[4]	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	1	1	1	0	0	1	0	0	0	22
m[5]	1	1	1	1	0	1	1	0	0	0	1	1	1	1	1	1	0	1	1	0	0	1	1	0	1	1	0	1	1	0	0	0	11
m[6]	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1	0	1	1	16
m[7]	0	1	1	1	1	0	0	0	0	1	0	1	1	0	0	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	1	1	14
m[8]	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19
m[9]	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m[10]	0	1	1	1	1	0	0	0	0	1	0	1	1	0	0	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	1	1	19
m[11]	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1	0	1	1	14
m[12]	1	1	1	1	0	1	1	0	0	0	1	1	1	1	1	1	0	1	1	0	0	1	1	0	1	1	0	1	1	0	0	0	16
m[13]	0	0	0	1	0	1	1	0	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	1	1	1	0	0	1	0	0	0	11

Abbildung 3: Symmetrie: $m_{17-i} = m_i \iff Color(row[17-i]) = Color(row[i])$

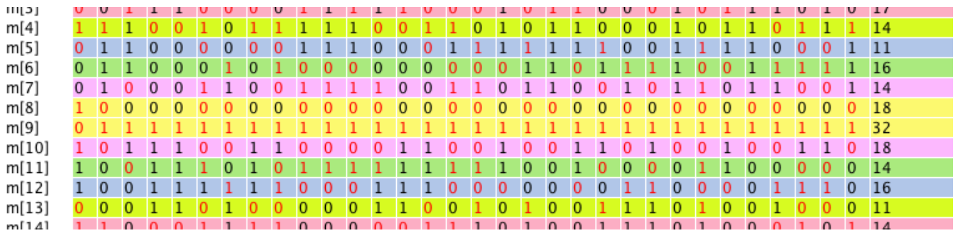
Wir erinnern uns, dass jede einzelne Runde (m_{i_0-1}, m_{i_0}) alle (m_i) generiert. Also tut dies speziell (m_8, m_9), wobei $m_8 = m_9$ sein muss im Falle eines Fixpunktes. Wenn wir also ein $m_8 \in \{0,1\}^{32}$ selektieren und die restlichen (m_i) über $m_9 := m_8$ sowie $m_{i+1} = m_{i-1} \oplus f(m_i, K_i)$ erzeugen, erhalten wir schließlich in c einen Fixpunkt, der die Symmetrie $m_{17-i} = m_i$ aufrechterhält.
Offensichtlich könnten wir auch mit einer anderen Runde (m_{i-1}, m_i) starten, so-

lange diese (m_8, m_9) erzeugt mit $m_8 = m_9$. Um die Überprüfung "$m_8 \stackrel{?}{=} m_9$" zu vermeiden, starten wir der Einfachheit halber mit m_8 und fahren wie oben beschrieben fort.
Im Programm gibt ein Benutzer irgendein $m_8 \in \{0,1\}^{32}$ ein, und der interne Algorithmus komplettiert die restlichen Runden über $m_9 := m_8$. Darüberhinaus erlaubt die Symmetrie $m_{17-i} = m_i$ eine reduzierte Visualisierung nur mit m_8, m_9, m_{10}, ..., m_{17}. Falls ein User jedoch alle Runden sehen möchte, kopiert er (p, k) ins Panel "Key/Plaintext" und verschlüsselt diese Eingaben. Das Programm zeigt dann jedes $(m_i)_{i=0,\ldots,17}$ (vgl. Abb. 3).

Schließlich bleibt noch die offene Frage "Welche Schlüssel $k \in \{0,1\}^{64}$ generieren Rundenschlüssel (K_i) so dass $K_{17-i} = K_i$?". Man kann beweisen, dass diese Schlüssel k gerade $k(0), k(5), k(10), k(15)$ sind, wobei $k(abcd) := \left[a^3b^4(P_1)c^3d^4(P_2)\right]^2$ $\left[a^4b^3(P_3)c^4d^3(P_4)\right]^2$ (Exponenten repräsentieren Bit-Wiederholungen) mit a, b, $c, d \in \{0,1\}$ und ungerade Parität produzierende Bits (P_k). Darüberhinaus erfüllen diese Rundenschlüssel sogar die strengeren Relationen $K_i \stackrel{alle\ i,j}{=} K_j$ (siehe Panel "Rundenschlüssel K_j").

Beispiel: Um $k(5) = k(5_{10})$ zu repräsentieren, gehen wir wie folgt vor:
$k(5) = k(0101) = \left[0^31^4(1)0^31^4(0)\right]^2 \left[0^41^3(0)0^41^3(0)\right]^2 =$ `1F1F 1F1F 0E0E 0E0E`.

Anti-Fixpunkte
Ein Anti-Fixpunkt p bzgl. einem Schlüssel k ist definiert durch $DES(k, p) = \overline{p}$. Wie bei Fixpunkten, weiß man auch bei Anti-Fixpunkten nicht genau wie viele hiervon existieren. Allerdings weiß man, dass es 2^{32} Anti-Fixpunkte bzgl. jedem der vier *semi-schwachen Schlüssel* $k(3n)_{j=1,\ldots,4}$ gibt. Eine Begründung hierfür ergibt sich durch Betrachtung des Chiffrierprozesses mit seinen Rundenchiffraten.



m[4]	1	1	1	0	0	1	0	1	1	1	1	1	0	0	1	1	0	1	0	1	1	0	0	0	1	0	1	1	0	1	1	1	14
m[5]	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	11
m[6]	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0	0	1	1	1	1	1	16
m[7]	0	1	0	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	1	1	0	0	1	0	1	1	0	1	1	0	0	1	14
m[8]	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18
m[9]	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	32
m[10]	1	0	1	1	1	0	0	1	1	0	0	0	0	1	1	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	1	0	18
m[11]	1	0	0	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0	14
m[12]	1	0	0	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	16
m[13]	0	0	0	1	1	0	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	1	1	0	1	0	0	1	0	0	0	11

Abbildung 4: Symmetrie: $m_{17-i} = \overline{m_i} \iff Color(row[17-i]) = Color(row[i])$

Die Generierung ist ähnlich der von Fixpunkten, jedoch etwas komplexer. Wir benötigen hierzu 3 Eigenschaften: $m_{i+1} = m_{i-1} \oplus f(m_i, K_i)$, $f(\overline{m}, \overline{K}) = f(m, K)$ und $\overline{x} \oplus y = \overline{x \oplus y}$. Um die gesuchten Anti-Fixpunkte zu finden, werfen wir wieder einen genaueren Blick auf den Chiffrierprozess mit seinen Rundenchiffraten

$$p \stackrel{IP}{\rightarrow} (m_0, m_1) \stackrel{K_1}{\rightarrow} (m_1, m_2) \stackrel{K_2}{\rightarrow} \ldots \stackrel{K_8}{\rightarrow} (m_8, m_9) \rightarrow \ldots \stackrel{K_{16}}{\rightarrow} (m_{16}, m_{17}) \stackrel{Flip}{\rightarrow} (m_{17}, m_{16}) \stackrel{IP^{-1}}{\rightarrow} c = \overline{p}$$

Daraus folgt $(m_{17}, m_{16}) = IP(\overline{p}) = \overline{IP(p)} = \overline{(m_0, m_1)} = (\overline{m_0}, \overline{m_1})$, also $m_{17} = \overline{m_0}$ und $m_{16} = \overline{m_1}$. Sowie $m_{15} = m_{17} \oplus f(m_{16}, K_{16}) = \overline{m_0} \oplus f(\overline{m_1}, K_{16})$. Falls wir also

einen Schlüssel k haben, der $K_{16} = \overline{K_1}$ induziert, dann gilt $\overline{m_0} \oplus f(\overline{m_1}, \overline{K_1}) = \overline{m_0} \oplus f(m_1, K_1) = \overline{m_0 \oplus f(m_1, K_1)} = \overline{m_2}$ und damit $m_{15} = \overline{m_2}$.
Man erkennt, dass die benötigten Relationen zwischen Rundenschlüsseln die Form $K_{17-i} = \overline{K_i}$ $(i = 1, \ldots, 16)$ haben. Diese erzeugen Rundenchiffrate (m_i) mit der Eigenschaft $m_{17-k} = \overline{m_k}$ $(k = 1, \ldots, 17)$, speziell gilt also $m_8 = \overline{m_9}$.
Schließlich benötigen wir noch die Antwort zur Frage "Welche Schlüssel $k \in \{0,1\}^{64}$ erzeugen Rundenschlüssel (K_i) s.d. $K_{17-i} = \overline{K_i}$?". Analog zu Fixpunkten kann man beweisen, dass diese Schlüssel $k(3), k(6), k(9), k(12)$ sind.

Die beiden sich anschließenden Behauptungen beenden die Diskussion über Fixpunkte (FP) und Anti-Fixpunkte (AFP). Beide ergeben sich aus der Beziehung $DES(\overline{k}, \overline{p}) = \overline{DES(k,p)}$ und dienen der Erzeugung neuer Anti-/Fixpunkte aus bereits bekannten.

$$(1) \quad FP(k, m_8) = \eta \iff FP(\overline{k}, \overline{m_8}) = \overline{\eta}$$
$$(2) \quad AFP(k, m_8) = \eta \iff AFP(\overline{k}, \overline{m_8}) = \overline{\eta}$$

Zusammenfassung: Die Erzeugung von Fixpunkten und Anti-Fixpunkten verläuft ähnlich, allerdings findet bei Anti-Fixpunkten zusätzlich die Operation "Komplementbildung" (bitweise Komplementierung) Verwendung. Es bestehen die Symmetrien $m_{17-i} = m_i$ für Fixpunkte sowie $m_{17-i} = \overline{m_i}$ für Anti-Fixpunkte. Bis heute kennt man Anti-/Fixpunkte nur bzgl. den *semi-/schwachen Schlüsseln*. Jeder dieser acht Schlüssel sollte wegen der von ihnen hervorgerufenen Beziehung zwischen Klartext und Geheimtext vermieden werden.

(4) Der interne Avalanche Effekt bzgl. Eingabedifferenzen Δp

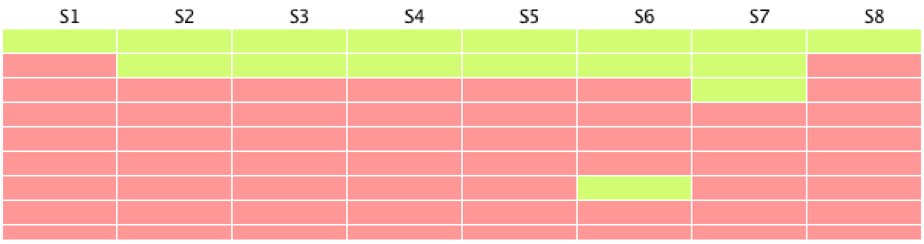


Abbildung 5: Zelle (i,j) = GELB $\iff$ In Runde i erhält S-Box j die gleichen Eingaben

Wir verschlüsseln zwei Klartexte $p, p + \Delta p$ mit einem Schlüssel k und erhalten $DES(k,p)$ und $DES(k, p + \Delta p)$. Da S-Boxen (aufgrund ihrer Nicht-Linearität) das Herz des DES darstellen, ist eine naheliegende Frage (speziell für $\Delta p = e_i :=$ i-ter *Einheitsvektor* oder $\Delta p = e_i + e_j$): "Welche S-Boxen erhalten verschiedene Eingaben, wenn man parallel (k,p) und $(k, p + \Delta p)$ verschlüsselt ?". Die Antwort hierzu wird pro Runde und pro S-Box durch eine gelb-rot gefärbte Matrix (Abb. 5) gegeben. Deren Farben bedeuten:

- Zelle (i,j) ist ROT $\iff$ In Runde i, S-Box j erhält verschiedene Eingaben
- Zelle (i,j) ist GELB $\iff$ In Runde i, S-Box j erhält gleiche Eingaben

Da die Eingaben, welche die Farbgebung definieren, nicht unmittelbar offensichtlich sind, erläutern wir deren Bestimmung unter Verwendung von Abb. 6, 7, die dadurch letztendlich auf die Zeilen im Panel "$m_0 - m_{17}$" verweisen.

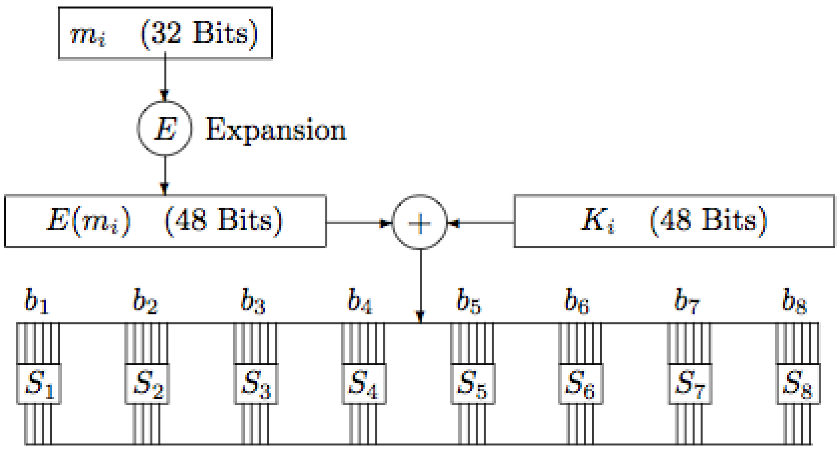


Abbildung 6: Eingabe $E(m_i) \oplus K_i$ für alle acht S-Boxen

Abb. 6: Wir sehen, dass (in Runde i) die acht S-Boxen als Eingaben $E(m_i) \oplus K_i$ und $E(m_i') \oplus K_i$ erhalten. Da hierbei K redundant ist, liefert es keinen Beitrag zur Eingabe-Differenz und kann vernachlässigt werden. Deshalb arbeiten wir nur mit $E(m_i), E(m_i')$. Und da die Panels im Applet lediglich die m_i, aber nicht die benötigten $E(m_i)$ zeigen, ist unser Problem: "Wie können wir $E(m_i)$ aus m_i bestimmen ?". Die Antwort hierzu kann in Abb. 7 abgelesen werden, die die Eigenschaften der Expansionsfunktion E visualisiert.

Abb. 7: Offensichtlich erhält S-Box j von $E(m_j)$ Eingabebits von den Bit-Positionen $6(j-1)+1, 6(j-1)+1, \ldots, 6j$ (in Abb. 7 die eingerahmten 6 Zahlen der niedrigsten Zeile). Diese stammen von m_i's Bit-Positionen $4(j-1)$ $(mod\ 32)$, $4(j-1)+1$ $(mod\ 32), \ldots, 4j$ $(mod\ 32)$, $4j+1$ $(mod\ 32)$ (mit $\mathbb{Z}/32\mathbb{Z} := \{\overline{1}, \overline{2}, \overline{3}, \ldots, \overline{32}\}$, dadurch werden z.B. die Bit-Positionen $0, 33$ durch die Zahlen $32, 1$ repräsentiert).

Beispiele:

- In Runde i, S-Box 1 erhält Eingaben von m_i's Bit-Positionen $32, 1, 2, 3, 4, 5$.
- In Runde i, S-Box 2 erhält Eingaben von m_i's Bit-Positionen $4, 5, 6, 7, 8, 9$.
- In Runde i, S-Box 8 erhält Eingaben von m_i's Bit-Positionen $28, 29, 30, 31, 32, 1$.

Um also in Abb. 5 die Farbe für die Zelle (i, j) zu bestimmen, müssen wir nacheinander (k, p) and $(k, p+\Delta p)$ ins Panel "Key/Plaintext" kopieren und dann dort verschlüsseln. Danach vergleichen wir im Panel '$m_1 - m_{17}$" die jeweiligen Strings auf m_i's Bit-Positionen $4(j-1)$ $(mod\ 32)$, $4(j-1)+1$ $(mod\ 32), \ldots, 4j$ $(mod\ 32)$, $4j+1$ $(mod\ 32)$.

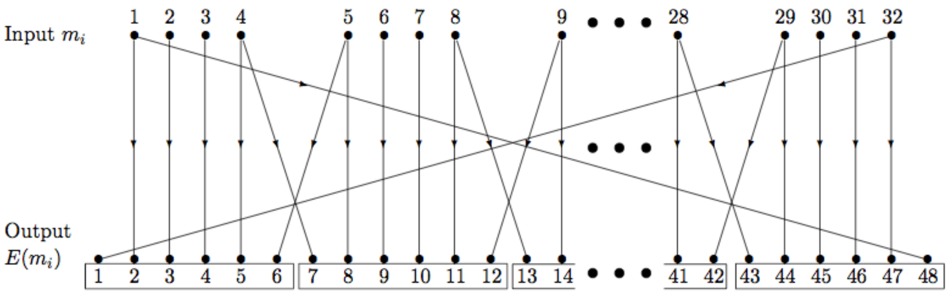


Abbildung 7: Die Wirkung der Expansionsfunktion E

Tests zeigten, dass die Farbmuster "fast unabhängig" von (k, p) und also mehr oder weniger nur abhängig von Δp waren. Genauer gesagt bedeutet dies, dass die meisten Farben invariant unter Eingaben (k, p) waren solange nur $\Delta p \equiv const.$. Nur wenige Zellen waren abhängig von (k, p). Mehrfaches Verschlüsseln mit randomisierten Eingaben (k, p) und konstantem Δp zeigt, wie sich Farbmuster minimal bei jedem Versuch ändern. Da das Gesamt-Farbmuster darlegt wie sich Eingabedifferenzen über den gesamten Chiffrierprozess ausbreiten, kann es als Maß für die Avalanche-Fähigkeiten des DES verwendet werden.

Grob kann man also sagen, dass sich – aufgrund der (vier) Überkreuz-Ausbreitungen in der Expansionsfunktion E (vgl. Abb. 7) – mit Beginn der vierten Runde selbst geringe Eingabedifferenzen (i.e. $dist(\Delta p, 0) \leq 2$) über alle S-Boxen ausbreiten und vom DES dann völlig unterschiedliche Chiffrate erzeugt werden.

(5) Literatur für ein vertiefendes Verständnis der DES-Eigenschaften

[Ba09] W. Baltes: *Charakteristika des DES und deren programmiertechnische Visualisierung*, Masterarbeit in Computer Science bei Prof. Dr. Jörg Keller, Fern-Universität in Hagen (GER), Februar 2009

[Bi88] E. Biham and A. Shamir: *Differential Cryptanalysis of DES-like Cryptosystems*, (Extended Abstract), 1988, Springer, pp. 2-21, Advances in Cryptology: Proceedings of CRYPTO '88, Springer, 1990, pp. 450-468

[Bi91] E. Biham and A. Shamir: *Differential Cryptanalysis of DES-like Cryptosystems*, CTYPTO '90 & Journal of Cryptology, Vol. 4, No. 1, pp. 3-72, 1991

[Bi92] E. Biham and A. Shamir: *Differential Cryptanalysis of the Full 16-Round DES*, Advances in Cryptology: Proceedings of CRYPTO '92, Springer, 1993, pp. 487-496

[Bu04] Johannes Buchmann, *Einführung in die Kryptographie*, Springer, 2004, ISBN 3-540-40508-9

[Br86] E.F. Brickell, J.H. Moore, and M.R. Purtill, *Structure in the S-Boxes of the DES*, Advances in Cryptology: Proceedings of CRYPTO '86, Springer, 1987, pp. 3-8

[Co94] D. Coppersmith, *The Data Encryption Standard (DES) and its strength against attacks*, IBM J. Res. Develop., Vol. 38, No. 3, May 1994, pp. 243-250

[Da82] Donald W. Davies, *Some Regular Properties of the DES Algorithm*, Advances in Cryptology: Proceedings of Crypto '82, Plenum Press, 1983, pp. 89-96

[Ko82] Matthias König, Elmar Meyer zu Bexten, *DES Data Encryption Standard*, 14. August 1996, *http://www.uni-paderborn.de/fachbereich/AG/agmadh/.../bexten.ps.gz*

[Ln00] Susan Landau, *Standing the Test of Time: The Data Encryption Standard*, Notices of the AMS, March 2000, pp. 341-349

[Mo86] J.H. Moore and G.J. Simmons, *Cycle Structure of the DES for Keys Having Palindromic (or Antipalindromic) Sequences of Round Keys*, Proceedings of Eurocrypt '86, Linköping, Sweden, May 20-22, 1986

[Mo87] J.H. Moore and G.J. Simmons, *Cycle Structure of the DES with Weak and Semi-Weak Keys*, Advances in Cryptology: Proceedings of CRYPTO '86 s, Springer, 1987, pp. 9-32