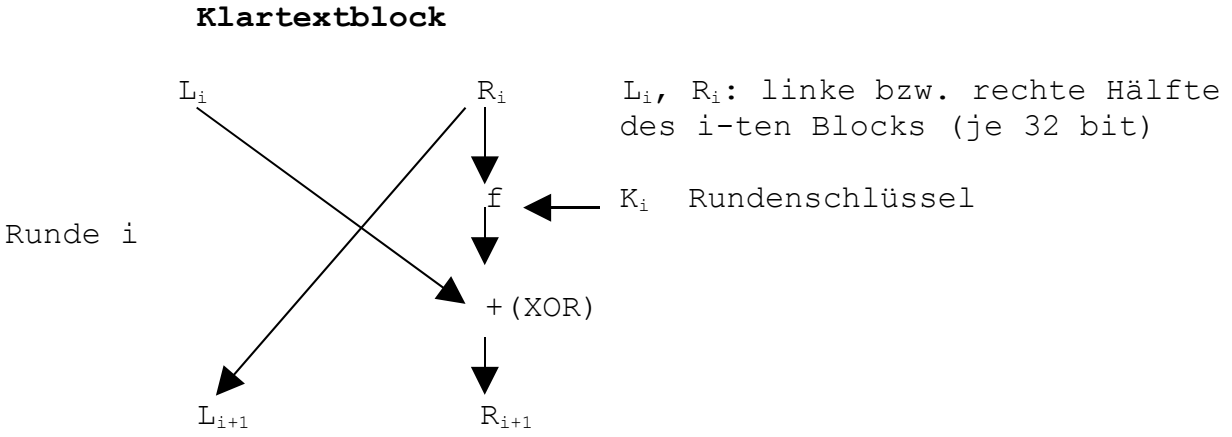
Teil 4 Feistel-Chiffren, lineare Kryptoanalyse beim DES

Feistel-Chiffren

DES (Data Encryption Standard)war Jahrzehnte im Einsatz, er ist der am besten untersuchte Chiffrieralgorithmus überhaupt, und er hat allen Versuchen, ihn zu brechen, erstaunlich gut widerstanden. Inzwischen ist die Zeit von DES aber abgelaufen, denn auf Grund seiner geringen Schlüssellänge (nur 56 bit) kann er heutzutage mit Gewalt (*Brute-Force*) gebrochen werden. Falls man ein zusammengehörendes Klartext-Geheimtext-Paar kennt (*Known-Plaintext-Angriff*), so probiert man einfach für alle 2^{56} mögliche Schlüssel, ob DES angewandt auf den Klartext den Geheimtext liefert. – Vor einiger Zeit war ein derartiges Vorgehen Illusion oder dauerte Ewigkeiten an Rechenzeit, inzwischen dauert es nur noch Stunden.

Bei DES handelt es sich um eine *Feistel-Chiffre* mit 16 Runden. Bei einer Feistel-Chiffre hat eine einzelne Runde diese Struktur:



Algebraisch:

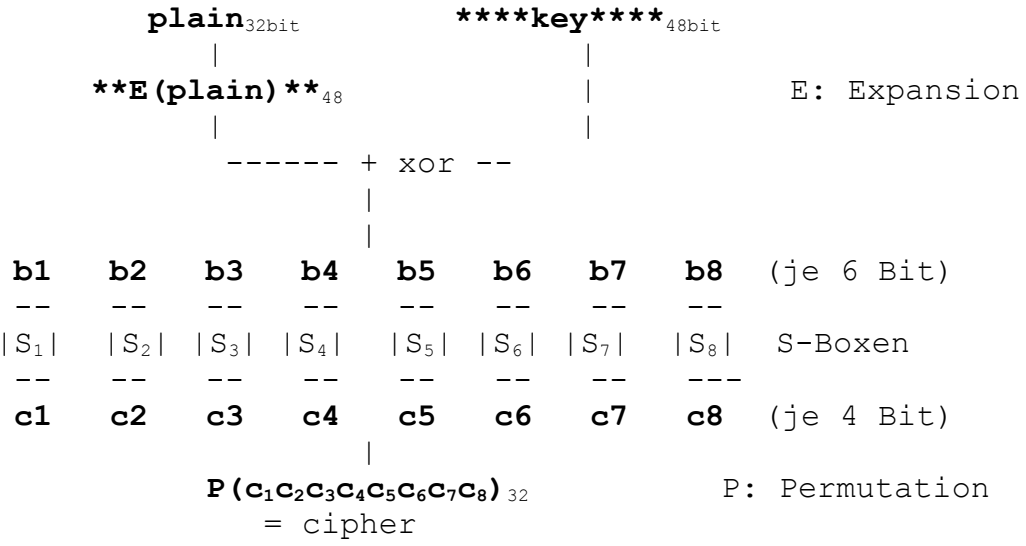
$$L_{i+1} = R_i$$
$$R_{i+1} = L_i + f(R_i, K_i)$$

Bei Feistel-Chiffren kann man in besonders einfacher Weise entschlüsseln und zwar ohne die Details der *Rundenfunktion* f kennen zu müssen. Umstellen der letzten beiden Gleichungen liefert:

$$R_i = L_{i+1}$$
$$L_i = R_{i+1} + f(R_i, K_i)$$

Man erkennt: Beim Entschlüsseln kommt man rückwärts von der Stufe i+1 zur Stufe i ganz genauso, wie man beim Verschlüsseln vorwärts von Stufe i zur Stufe i+1 kam.
Bei DES läuft es darauf hinaus, daß man beim Entschlüsseln lediglich die Reihenfolge der Rundenschlüssel umzukehren hat.

Später werden wir die genaue Form der DES-Rundenfunktion f benötigen, die wir gleich angeben:



Die Expansion E ersetzt die Bits gemäß folgender Tabelle:

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

Die Permutation P substituiert nach dieser Tabelle:

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 Sicherheit einer Feistel-Chiffre steckt in dieser Rundenfunktion f . Beim DES enthält sie Boolesche Addition, Permutationen, Bitverdopplungen und als harten Kern nichtlineare Transformationen in der tabellarischen Form sogenannter **S-Boxen.**

Man beachte: DES *ohne* die S-Boxen wäre mit einem Known-Plaintext-Angriff leicht angreifbar. Wie bei den linearen Schieberegistern

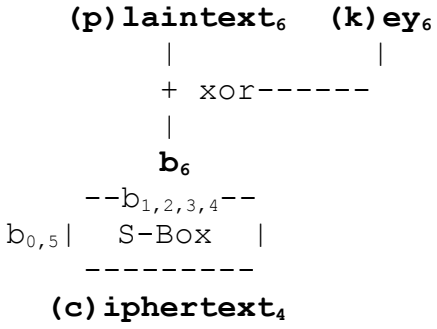
von Teil 3 könnte man Gleichungen für die gesuchten Schlüsselbits aufstellen und lösen.

Jede S-Box transformiert 6 Bits gemäß einer festen und öffentlich bekannten Tabelle (mathematisch: einer nichtlinearen Abbildung) zu 4 Bits. Der im folgenden beschriebene lineare Kryptoangriff wird am deutlichsten bei der S-Box Nr.5, die wir deswegen hier angeben:

S-Box Nr.5:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	4 Zeilen
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	16 Spalten
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	

Die Transformation 6 Bits->4 Bits macht die S-Box folgendermaßen:



Also: 6 Bits des Klartextes

p_0,p_1,p_2,p_3,p_4,p_5 , p: plain

werden mit 6 (gesuchten) Schlüsselbits

k_0,k_1,k_2,k_3,k_4,k_5 , k: key

verodert: $b = p+k$

Von diesem Zwischenresultat $b=(b_0b_1b_2b_3b_4b_5)$ bestimmen die beiden *äußersten* Bits b_0 und b_5 eindeutig eine Zeile, die *übrigen Bits* b_1,b_2,b_3,b_4 bestimmen eine Spalte der S-Box.
Der in dieser Weise anvisierte Eintrag der S-Box ist eine Zahl zwischen 0 und 15, wodurch also 4 Bits definiert werden. Diese Bits sind der resultierende Geheimtext:

c_0,c_1,c_2,c_3 , c: cipher

Damit ist das Funktionieren der S-Boxen beim DES beschrieben.

Lineare Kryptoanalyse beim DES
Bei dem im folgenden beschriebene linearen Angriff gegen die S-Box Nr.5 handelt es sich um einen **Chosen-Plaintext Angriff**, also die schwierigste Form des Angriffs gegen eine Chiffre: Die Angreiferin Eve muß die Verschlüsselerin Alice dazu veranlassen, von ihr (Eve) vorgegebenen Klartext zu verschlüsseln und ihr auch den Geheimtext zur Verfügung zu stellen.
Wir wollen gleich betonen: Im Falle des DES hatte diese Angriffsform erstaunlich wenig Erfolg. Selbst damit hat man nur unwesentlich weniger Mühe hat als mit Brute-Force Angriff. Dies ist eine Tatsache, die deutlich für die Qualität des DES-Design spricht!

Der lineare Angriff gegen die S-Box Nr.5 nutzt eine Unsymmetrie zwischen linker und rechter Hälfte des S-Box. Wir ersetzen interessehalber jede Zahl der Box durch die boolesche Summe

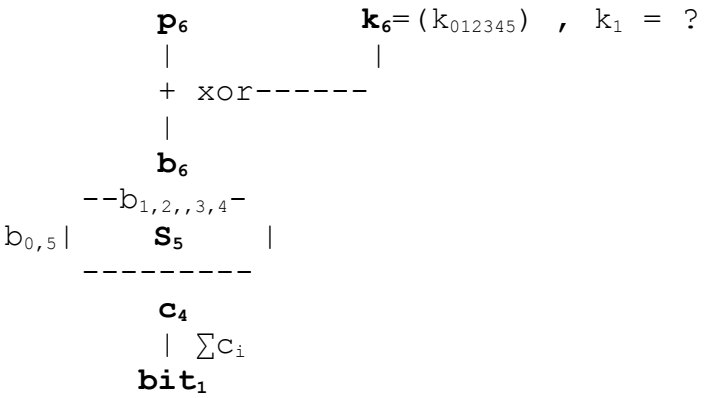
$$c_0+c_1+c_2+c_3$$

ihrer Bits:

S-Box Nr.5 , $\sum c_i$

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	1	0	1	1	1	0	1	0	1	0	0	0	1	0	1	0
1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	1	0
2	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	1
3	1	1	0	1	1	1	1	1	0	0	0	0	0	1	0	0

Man konstatiert eine deutliche Konzentration der Einsen links von der Mitte, und dies nutzt die Angreiferin aus:



Klartext **p** und der Geheimtext **c** sind der Angreiferin bekannt, den Klartext (unterstellen wir) kann sie sogar vorwählen.
Herausbekommen möchte die Angreiferin den Schlüssel **k**=(k_{012345})
Mit der folgenden Technik bekommt sie (nur!) das erste Bit $\mathbf{k_1}$ von **k** heraus.
Sie wählt zufällige 6 Bit Klartexte p aber *immer mit p_1=0* und unterscheidet zwei Fälle:

Fall 1: k_1=0 , dann ist b_1=0, d.h. der Spaltenindex ist klein, und **bit** ist *wahrscheinlich* 1.

Fall 2: $k_1=1$, dann ist $b_1=1$, d.h. der Spaltenindex ist groß, und **bit** ist *wahrscheinlich* 0.

Das Vorgehen für die Angreiferin Eve ist nun klar. Sie läßt ihr Opfer Alice solange Klartext (immer mit $p_1=0$) verschlüsseln und beobachtet **bit**, bis sie hinreichend Klarheit hat: $\mathbf{k_1}$ ist *wahrscheinlich* das, was **bit** meistens nicht ist.

Die hier beschriebene Angriffstechnik kann in vielfacher Weise variiert und auch auf mehrere Runden ausgedehnt werden:
- alle 8 S-Boxen können angegriffen werden
- man addiert nur einige der c_i statt alle
- die S-Box kann in vielfältiger anderer Weise eingeteilt werden (Schachbrett, Streifen...) statt nur links-rechts

Alle diese Möglichkeiten können kombiniert und natürlich automatisiert werden. Tatsache ist: Viele Chiffren sind solchen linearen Angriffstechniken (bzw. alternativen *differentiellen* Angriffen) zum Opfer gefallen, nicht aber er DES!