Differenzen ΔA folgende Verteilung:

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

Wir haben also hier, wie auch bei den anderen Differenzen und anderen S-Boxen eine ziemlich inhomogene Verteilung. (Die Fallanzahlen für jede S-Box und jede Ausgabedifferenz sind aufgelistet im Anhang von

E. BIHAM, A. SHAMIR: Differential Cryptanalysis of the Data Encryption Standard, *Springer,* 1993;

in diesem Buch ist die differentielle Kryptoanalyse des DES einschließlich der Verallgemeinerung auf andere Systeme vollständig beschrieben.)

Auf die Ausgabe der acht S-Boxen wird noch eine Permutation angewandt; die ist wieder mit Differenzbildung kompatibel. Falls wir also die Differenzen der Ausgaben der S-Boxen kennen, bereitet diese Permutation keine Schwierigkeiten und wir kennen die Eingabedifferenzen für die zweite Runde, mit der wir genauso weiter verfahren können.

Tatsächlich kennen wir die Ausgaben der acht S-Boxen der ersten Runde natürlich nicht; wir können nur für jede einzelne S-Box eine Wahrscheinlichkeitsverteilung der Ausgabedifferenzen angeben. Für einzelne Bits oder Bitgruppen kommen wir dabei durchaus auf recht ansehnliche Wahrscheinlichkeiten: Im obigen Beispiel für die Eingabedifferenz 100100 etwa ist mit jeweils einer Wahrscheinlichkeit von $14 : 64$, also in mehr als 20% aller Fälle, die Ausgabedifferenz gleich acht oder neun, binär geschrieben also 0100 oder 0101. Die Wahrscheinlichkeit dafür, daß die ersten drei Bits der Ausgabedifferenz gleich 010 sind, ist also $28 : 64 = 7 : 16$, und wenn wir uns auf das erste und dritte Bit beschränken, kommen wir auf eine Wahrscheinlichkeit von $40 : 64 = 5 : 8$ dafür, daß die Ausgabedifferenz die Form $0x0y$ hat. Das dritte Bit schließlich ist in 52 der 64 möglichen Fällen gleich null, so daß wir zumindest dieses eine Bit mit der recht hohen Wahrscheinlichkeit von $13 : 16$ kennen.

Der Sinn eines FEISTEL-Netzwerks ist es, uns solche Informationen, die wir für eine Runde haben, in den weiteren Runden durch Konfusion

und Diffusion weitestgehend zu zerstören. Man wird erwarten, daß dies nicht für alle Klartextdifferenzen gleich gut gelingt; die Idee hinter der differentiellen Kryptoanalyse ist, sich auf die zu konzentrieren, bei denen es möglichst schlecht gelingt. Wir müssen uns also genauer anschauen, wie eine Klartextdiffernz durch die Runden geht.

Definition: Eine r-*Rundencharakteristik* ist eine Folge

$$\Delta = (\delta_0, \delta_1, \dots, \delta_r)$$

von Elementen aus $\mathbb{F}_2^{64}$.
Ein Klartextpaar $(x_0, y_0) \in \mathbb{F}_2^{64} \times \mathbb{F}_2^{64}$ *gehört zur Charakteristik* Δ, wenn für die Paare (x_i, y_i) der Ausgaben der i-ten Runde gilt: $x_i \oplus y_i = \delta_i$ für $i = 0, \dots, r$.
Die *Wahrscheinlichkeit* einer Charakteristik ist die Wahrscheinlichkeit dafür, daß ein Klartextpaar (x_0, y_0) mit Differenz δ_0 zur Charakteristik Δ gehört.

Natürlich sind die Wahrscheinlichkeiten der meisten Charakteristiken sehr gering: In einem idealen Kryptosystem wären alle Ausgabewerte einer Runde gleich wahrscheinlich, die Wahrscheinlichkeit einer r-Rundencharakteristik sollte also im Mittel bei $1/2^{-64\times r}$ liegen, und schon die Wahrscheinlichkeit dafür, daß überhaupt ein Klartextpaar mit gegebener Differenz nach r-Runden eine vorgegebene andere Differenz hat, sollte im allgeinen bei nur etwa 2^{-64} liegen.

Wenn wir eine gute r-Rundencharakteristik gefunden haben, deren Wahrscheinlichkeit deutlich besser ist als 2^{-64}, können wir daher ziemlich sicher sein, daß ein zufällig gewähltes Klartextpaar (x_0, y_0) mit Anfangsdifferenz δ_0 und Enddifferenz δ_r in den r Runden so verschlüsselt wurde, wie es der Charakteristik entspricht. Ist etwa $p = 2^{-44}$, sollte die Wahrscheinlichkeit für alles andere bei nur etwa 2^{-20} oder etwa eins zu einer Million liegen. Konkret heißt das: Wir kennen die Differenzen $x_i \oplus y_i$ mit hoher Wahrscheinlichkeit und können die Verschlüsselung durch die r Runden verfolgen.

Zunächst brauchen wir aber Charakteristiken. Für nur eine Runde ist das sehr einfach: Für jeden Halbblock $d_0 \in \mathbb{F}_2^{32}$ führt der mit 32 Nullen auf 64 Bit aufgefüllte Block δ_0 auf die Charakteristik (δ_0, δ_0) mit Wahrscheinlichkeit eins: Sind nämlich (m_0, m_1) und (m_0', m_1) zwei Klartexte mit (nach der hier stets ignorierten Anfangspermutation) gleicher rechter Hälfte, so wird die linke Hälfte ersetzt durch

$$m_2 = m_0 \oplus f(s_1, m_1) \quad \textit{bzw.} \quad m_2' = m_0' \oplus f(s_1, m_1)\,,$$

und die Differenz ist

$$m_2 \oplus m_2' = \bigl(m_0 \oplus f(s_1, m_1)\bigr) \oplus \bigl(m_0' \oplus f(s_1, m_1)\bigr) = m_0 \oplus m_0' = d_0\,,$$

wie wir das schon oben gesehen haben. Nach Durchgang durch die erste Runde haben wir also die beiden Paare (m_2, m_1) und (m_2', m_1), deren Differenz wieder δ_0 ist.

Leider können wir diese Charakteristik nicht iterieren, denn nach der ersten Runde werden ja die beiden Hälften vertauscht, so daß wir dann (m_1, m_2') und (m_1, m_2') haben, worüber wir nicht soviel sagen können, da nun die FEISTEL-Funktionen verschiedene Werte liefern.

Um trotzdem zu einer Zweirundencharakteristik zu kommen, benutzen wir die Nichtinjektivität der FEISTEL-Funktion f: Wir suchen zwei Halbblöcke m_0 und m_0' derart, daß $f(s, m_0) = f(s, m_0')$ für möglichst viele Schlüssel s; die Differenz $m_0 \oplus m_0'$ bezeichnen wir mit d_0.

Zwei beliebige Klartexte der Form (m_0, m_1) und (m_0', m_1) mit $m_0 \oplus m_0' = d_0$ gehen in der ersten Runde nach (m_2, m_1) und (m_2', m_1) mit

$$m_2 = m_0 \oplus f(s_1, m_1) \qquad \text{und} \qquad m_2' = m_0' \oplus f(s_1, m_1)\,,$$

danach werden die linke und die rechte Hälfte vertauscht, so daß die Eingaben zur zweiten Runde gleich (m_1, m_2) und (m_1, m_2') sind, wobei m_2 und m_2' die Differenz d_0 haben. In der zweiten Runde gehen die beiden Klartexte dann nach

$$\bigl(m_1 \oplus f(s_2, m_2), m_2\bigr) \qquad \text{und} \qquad \bigl(m_1 \oplus f(s_2, m_2'), m_2'\bigr)\,.$$

Da m_2 und m_2' Differenz d_0 haben, ist dabei mit einer gewissen Wahrscheinlichkeit

$$f(s_2, m_2) = f(s_2, m_2')\,.$$

Falls dem so sein sollte, haben wir $(0, d_0)$ als Differenz zwischen den Ausgabewerten. Es gibt daher eine Zweirundencharakteristik der Form

$$\bigl((d_0, 0), (d_0, 0), (0, d_0)\bigr)\,,$$

die mit einer gewissen Wahrscheinlichkeit p auftritt. Diese Charakteristik kann offensichtlich beliebig oft iteriert werden, denn bevor ihre Ausgabewerte in die nächste Runde gehen, werden die linke und die rechte Hälfte vertauscht, so daß wir wieder die Ausgangsdifferenz haben. Damit haben wir für jedes n eine n-Rundencharakteristik gefunden; ihre Wahrscheinlichkeit ist $p^{[n/2]}$, wobei die GAUSS-Klammer $[x]$ die größte ganze Zahl $\leq x$ bezeichnet.

Diese Charakteristik ist umso nützlicher, je größer p ist. Wie sich zeigt, kann p nur dann größer als null sein, wenn mindestens drei benachbarte S-Boxen aktiv sind; die größten Wahrscheinlichkeiten sind also zu erwarten bei *genau* drei aktiven S-Boxen. Systematisches Probieren zeigt, daß der beste erreichbare Wert dann

$$p = \frac{14}{64} \cdot \frac{8}{64} \cdot \frac{10}{64} = \frac{35}{8192} \approx 0{,}004272461 \approx \frac{1}{234} \approx 2^{-7{,}870716983}$$

ist. Er wird erreicht für

$$d_0 = (19\,60\,00\,00)_{\text{hex}} \qquad \text{und} \qquad d_0 = (1B\,60\,00\,00)_{\text{hex}}\,.$$

Damit haben wir also für beliebiges n eine n-Rundencharakteristik gefunden; leider ist sie aber nicht für jedes n brauchbar: Für 16 Runden ist ihre Wahrscheinlichkeit nur etwa

$$2^{-7{,}870716983 \times 8} \approx 2^{-62{,}96573586}\,,$$

wir bräuchten also mindestens 2^{63} Klartextpaare bekannter Differenz, um eines zu dieser Charakteristik zu finden, während wir mit nur 2^{56} Versuchen alle Schlüssel durchprobieren könnten. Tatsächlich reichten sogar bereits 2^{55} Versuche, denn nimmt man das Einserkomplement von Schlüssel und Klartext, so entsteht das Einserkomplement des Schlüsseltexts. Deshalb sind auch die 14- und die 15-Rundencharakteristik, deren Wahrscheinlichkeiten bei

$$2^{-7{,}870716983 \times 7} \approx 2^{-55{,}09501888}$$

liegen, nicht sonderlich interessant; erst die 13-Rundencharakteristik liefert mit

$$p \approx 2^{-7{,}870716983 \times 6} \approx 2^{-47{,}22430190}$$

eine halbwegs interessante Wahrscheinlichkeit.

Wählen wir ein Halbwort m_0 derart, daß $m_0 \oplus f(s, m_1)$ für möglichst viele Schlüssel s gleich ist! Für einen solchen Schlüssel sind dann

$$m_2 = m_0 \oplus f(s, m_1) \qquad \text{und} \qquad m_2' = m_0' \oplus f(s, m_1)$$

Außerdem interessieren wir uns nicht für die Wahrscheinlichkeitsverteilung der Chiffretexte – Chiffretext ist schließlich das, was wir immer haben – sondern für Klartext oder besser noch den Schlüssel bei gegebenem Chiffretext.

Dazu nutzen wir aus, daß Eingabewert der S-Boxen nicht der ersten Runde nicht der Klartext ist, sondern der mit gewissen Schlüsselbits geXORte Klartext. Wenn wir nun für ein Klartextpaar mit gegebener Differenz die Ausgabedifferenz kennen, haben wir die möglichen Eingabepaare der S-Boxen, deren Differenzen ja genau dieselben sind wie für das Klartextpaar, von 64 auf eine erheblich kleinere Zahl reduziert. Für jedes dieses möglichen Paare können wir die entsprechenden Schlüsselbits durch XOR mit dem tatsächlichen Klartext berechnen und haben somit eine relativ kleine Anzahl potentieller Schlüsselteile. Wenn wir das ganze für hinreichend viele Klartextpaare wiederholen, sollte der für die jeweilige S-Box zuständige Schlüsselanteil relativ bald eindeutig feststehen.

DES mit nur einer Runde ist auf diese Weise also relativ einfach zu entschlüsseln, falls wir genügend viele Paare von Klartext mit fester Differenz haben. Diese können wir uns nur verschaffen durch eine Attacke mit wählbarem Klartext, also der schwierigsten Form der Attacke.

Auch diese Attacke liefert aber nicht die Ausgabedifferenzen der ersten Runde, sondern nur die der *letzten.* Der Ansatz der differentiellen Kryptoanalyse des DES ist daher folgender:

1. Man wähle eine geeignete Differenz zwischen Klartexten.
2. Dazu erzeuge man hinreichend viele Paare von Klartextblöcken mit dieser Differnz, verschlüssele sie und behalte nur die so berechneten Chiffretextpaare.
3. Durch Analyse der Klartextdifferenzen und des Verhaltens der S-Boxen in den verschiedenen Runden bestimme man die zu erwartende Wahrscheinlichkeitsverteilung der Eingabedifferenzen der letzten Runde.

Differentielle Kryptanalyse war der erste Ansatz, DES mit geringerem Aufwand als der vollständigen Durchsuchung des Schlüsselraums zu zu brechen. Da aber, wie bereits erwähnt, die Designer des DES die differentielle Kryptanalyse schon kannten lange bevor sie in der offenen Literatur auftauchte und den Algorithmus so gut wie möglich dagegen immun machten, ist diese Attacke nicht sehr praktikabel: Selbst bei Angriffen mit frei wählbarem Klartext braucht man über 2^{40} Paare aus Klartext und Chiffretext, um den Schlüssel zu finden.

d) Lineare Kryptanalyse

Eine leichte Verbesserung bietet die kurz später entdeckte *lineare Kryptanalyse:* Zwar sind die Ausgabebits der S-Boxen nach Design-Kriterium ($S2$) auch nicht näherungsweise lineare Funktionen der Eingabebits, aber es kann dennoch vorkommen, daß man eine Linearkombination von Ausgabebits mit einer Wahrscheinlichkeit, die deutlich über 50% liegt durch eine Linearkombination der Eingabebits vorhersagen kann. Mit hinreichend vielen Paaren aus Klartext und Chiffretext Läßt sich dadurch ein niedrigdimensionaler affiner Unterraum des Schlüsselraums $\mathbb{F}_2^{56}$ finden, in dem der Schlüssel mit hoher Wahrscheinlichkeit liegen muß. Die vollständige Durchsuchung dieses Unterraums ist unproblematisch, so daß der Schlüssel mit hoher Wahrscheinlichkeit gefunden werden kann.

Auch für diese Attacke sind allerdings unrealistisch viele Paare aus Klartext und Chiffretext erforderlich (in einer Variante genügen noch mehr reine Chiffretexte), so daß der Gesamtaufwand nicht wirklich geringer sein dürfte als die vollständige Durchsuchung des Schlüsselraums.

Nach allem, was in der offenen Literatur bekannt ist, gibt es also zum Knacken des DES keine wesentlich bessere Alternative zur vollständigen Durchsuchung des Schlüsselraums.

e) DES-Cracker

Der erste in der offenen Literatur dokumentierte realistische Angriff auf DES war denn auch die vollständige Durchsuchung des Schlüsselraums. Eine amerikanische Bürgerrechtsorganisation, die *Electronic Frontier*

Foundation (EFF), konstruierte eine Maschine mit Spezialhardware zum Knacken von DES mit Chiffretext allein.

Die *Electronic Frontier Foundation* wurde 1990 nach dem großen *Hacker Crackdown* in den USA gegründet; Initiatoren waren unter anderem JOHN PERRY BARLOW, bekannt vor allem durch die Lieder, die er für *The Grateful Dead* schrieb, JOHN GILMORE, einer der Pioniere sowohl von *Sun Microsystems* als auch der *Free Software Foundation,* MITCHELL KAPOR, der Gründer von *Lotus,* sowie STEVE WOZNIAK, einer der beiden Gründer von *Apple.*

Ihr Ansatz ist im wesentlichen der unseres guten alten Feinds, des BAYESschen Gegners: Kodiert man einen englisch- oder deutschsprachigen Klartext im ASCII-Code sollten dem BAYESschen Gegner ein bis zwei Blöcke Chiffretext ausreichen, um den Schlüssel zu finden.

Natürlich verfügen selbst die vier obengenannten Gründer der *Electronic Frontier Foundation* nicht über die unbegrenzten Mittel, die der BAYESsche Gegner einsetzen kann; verglichen mit vielen anderen Gegnern verfügt aber doch jeder von ihnen über beträchtliche Mittel. Trotzdem war das 1997 begonnene und 1998 beendete DES-Cracker-Projekt kein Angriff ohne Rücksicht auf die Kosten: Die *Electronic Frontier Foundation* wollte gerade zeigen, daß DES auch mit begrenzten Mitteln geknackt werden kann. Aus diesem Grund wurde der Ansatz des BAYESschen Gegners an mehreren Stellen optimiert:

Zunächst ist es nicht notwendig, wirklich für *jeden* Schlüssel die bedingte Wahrscheinlichkeit auf Grund des Chiffretexts zu bestimmen: In vielen Fällen werden bei der Entschlüsselung nicht druckbare Zeichen entstehen, so daß schon nach wenigen Byte klar ist, daß die Wahrscheinlichkeit des Schlüssels null ist.

DES Cracker beginnt daher mit der Aussonderung unmöglicher Schlüssel durch massiv parallele Hardware: Die Maschine arbeitet mit zwei Blöcken Chiffretext; sie hat als Kern von EFF entwickelte ASICs *application specific integrated circuits),* die einen 64-Bit-Block mit einem vorgegebenen Schlüssel dechiffrieren können und die Bytes des Ergebnisses auf vom Benutzer einstellbare Bitmuster überprüfen – beispielsweise darauf, ob es sich um ASCII-Codes druckbarer Zeichen handelt.

Nur wenn alle Bytes den gewählten Kriterien genügen, wird auch der zweite Block entsprechend untersucht, und wenn auch hier kein im Klartext unmögliches Byte auftaucht, wird der Schlüssel zur weiteren Untersuchung an einen die Maschine steuerenden PC weitergegeben, der eine genauere Untersuchung gemäß dem Ansatz des BAYESschen Gegners durchführt.

Von den 256 möglichen ASCI-Werten sind etwa ein Viertel druckbare Zeichen; da DES eine Ausgabe liefert, die sich nur wenig von einer Zufallsfolge unterscheidet, wird ein Block nur mit einer Wahrscheinlichkeit von etwa $1 : 4^8 = 1 : 65536$ den Test bestehen; für zwei Blöcke liegt die Wahrscheinlichkeit entsprechend bei $1 : 4^{16} = 1 : 2^{32}$. Von den 2^{56} zu untersuchenden Blöcken werden also nur etwa

$$2^{56-32} = 2^{24} = 16\,777\,216$$

an den PC weitergegeben, und die Untersuchung von etwa 17 Millionen Klartextkandidaten ist kein Problem für einen Standard-PC.

Der hauptsächliche Rechenaufwand liegt in der Voruntersuchung der Blöcke durch die ASICs; je nachdem, wie viele von diesen parallel arbeiten, kann dies mehr oder weniger schnell gehen.

Die tatsächlich gebaute Maschine enthält $1536 = 3 \times 2^9$ ASICs, von denen jedes aus 24 parallel arbeitenden Sucheinheiten besteht; insgesamt können also jeweils 36 864 Schlüssel parallel untersucht werden. Jede Sucheinheit kann zweieinhalb Millionen Schlüssel pro Sekunde untersuchen, die gesamte Maschine also etwas über 92 Milliarden.

Insgesamt müssen 2^{56} Schlüssel untersucht werden; im Mittel wird man nach 2^{55} Versuchen den richtigen gefunden haben. Dafür braucht man

$$\frac{2^{55}}{92\,160\,000\,000} \approx 390\,937 \text{ Sekunden} \approx 108{,}5 \text{ Stunden} \approx 4{,}5 \text{ Tage}\,.$$

Die Maschine ist skalierbar: Jeweils 64 Chips sitzen auf einem Board und 12 Boards in einer Chassis (einer ehemaligen SUN); die gebaute Maschine besteht aus dem steuernden PC zusammen mit zwei Chassis; der PC könnte aber auch mit deutlich mehr als zwei Chassis arbeiten und auch mehrere PCs wären denkbar.

In der gebauten Version kostete DES Cracker 210 000 $, wovon 80 000 $ Entwicklungskosten waren; der Bau eines zweiten Exemplars wäre also für 130 000 $ möglich, wobei der Preis bei Serienproduktion wohl noch deutlich gesenkt werden könnte. Mit einer Investition in Größenordnung einer Million Dollar käme man also wohl in den Bereich weniger Stunden für das Auffinden eines Schlüssels.

Da eine Million Dollar auch für Geheimdienste kleiner Länder und (etwas kreative Buchhaltung vorausgesetzt) Großunternehmen kein Problem sind, sollte damit klar sein, daß DES auf keinen Fall mehr heutigen Sicherheitsanforderungen genügt.

Die EFF veröffentlichte sowohl die komplette Hardware-Spezifikation als auch die Software von DES-Cracker; in gedruckter Form findet man sie im Buch

ELECTRONIC FRONTIER FOUNDATION: *Cracking DES. Secrects of Encryption Research, Wiretap Politics & Chip Design,* O'Reilly, 1998

Online ist das Buch unter anderem verfügbar unter

`http://cryptome.org/cracking-des.htm` ;

die DES Cracker Seite der EFF ist

`www.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/` .

§4: Modifikationen

a) Mehrfacher DES

Viele Unternehmen, insbesondere im Bankenbereich, haben viel Geld in DES-Hardware investiert und haben daher wenig Interesse, auf ein neues Verfahren umzusteigen – ganz abgesehen davon, daß 1998 noch kein allgemein anerkannter Nachfolgealgorithmus zur Verfügung stand. (Mit der heutigen Situation werden wir uns im nächsten Kapitel beschäftigen.)

Deshalb bot sich ein Verfahren an, das bereits kurz nach Normierung des DES vorgeschlagen wurde, um den kleinen Schlüsselraum zu vergrößern: Man verschlüssele mehrfach mit verschiedenen Schlüsseln.

Eine zweifache Verschlüsselung hängt von 2^{112} statt von 2^{56} Schlüsselbit ab; der Aufwand für eine Durchsuchung des gesamten Schlüsselraums steigt also um den Faktor 2^{56} gegenüber dem gewöhnlichen DES, was eine Maschine nach Art des DES Crackers nach heutigen Stand der Technik nicht mehr in akzeptabler Zeit durchführen kann.

Eine mehrfache Verschlüsselung bietet allerdings nur dann Vorteile, wenn die Hintereinanderausführung zweier DES-Verschlüsselungen nicht äquivalent zur einfachen DES-Verschlüsselung mit einem anderen Schlüssel ist.

Um dies zu untersuchen, müssen wir noch einmal zurück zur grundsätzlichen Struktur von Blockchiffren: Blockchiffren sind Permutationen auf der Menge aller Blöcke; im Falle von DES also auf der Menge aller bijektiver Abbildungen von der Menge aller 64-Bit-Blöcke auf sich selbst.

Die 2^{56} durch DES definierten Permutationen bilden natürlich nur eine winzige Teilmenge der Gruppe aller $2^{64}!$ solcher Permutationen; falls diese Teilmenge eine Gruppe sein sollte (oder in einer relativ kleinen Untergruppe liegt), kann die mehrfache Anwendung von DES keine (oder nur wenig) zusätzliche Sicherheit bieten.

Wir brauchen daher Informationen darüber, wie groß die kleinste Gruppe ist, die alle 2^{56} DES-Permutationen enthält. Über deren genaue Struktur und Elementanzahl ist leider nichts bekannt, man kann aber immerhin untere Schranken angeben: Für jeden einzelnen DES-Schlüssel kann man die zugehörige Substitution so lange wiederholen, bis die Identität entsteht; da $2^{64}!$ eine zwar große, aber endliche Zahl ist, muß dies nach endlich vielen Schritten der Fall sein.

Angenommen, bei solchen Berechnungen mit verschiedenen Schlüsseln ergaben sich die Ordnungen $n_1, n_2, \ldots, n_r$. Dann enthält die kleinste Gruppe, in der alle DES-Substitutionen liegen, zyklische Untergruppen der Ordnungen $n_1, \ldots, n_r$. Nach einem einfachen Satz der Gruppentheorie, dem Satz von LAGRANGE, müssen die Zahlen $n_1, \ldots, n_r$ dann die Ordnung der gesamten Gruppe teilen; diese ist also mindestens gleich dem kleinsten gemeinsamen Vielfachen der n_i.

Experimentelle Untersuchungen zweier Wissenschaftler von Bell Northern Research in Ottawa zeigten, daß die Ordnung der erzeugten Untergruppe größer als ,9 $\times 10^{2499}$ sein muß, was sehr deutlich über 2^{56} liegt; siehe ihre Arbeit

KEITH W. CAMPBELL, MICHAEL J. WIENER: DES is not a Group, *Crypto '92, Springer Lecture Notes in Computer Science* **740** (1993), 512–520

Durch mehrfache Anwendung von DES läßt sich also die Sicherheit erhöhen.

b) Doppelter DES

Beim doppelten DES hat man dann, wie bereits erwähnt, einen Schlüsselraum mit 2^{112} Elementen. Leider muß man diesen aber, falls man genügend viel Speicherplatz hat, nicht vollständig durchsuchen, um die Schlüssel zu finden: Falls man ein Paar aus einander entsprechendem Klartext und Chiffretext hat, reicht es, in einer sogenannten *meet in the middle attack* den Klartext mit allen 2^{56} möglichen Schlüsseln zu *ver*schlüsseln und den Chiffretext mit allen 2^{56} Schlüsseln zu *ent*schlüsseln; sobald man einen Block gefunden hat, der bei beiden Operationen auftritt, hat man mit an Sicherheit grenzender Wahrscheinlichkeit die beiden Schlüssel gefunden.

In der Praxis würde man wohl den beträchtlichen Speicheraufwand für 2^{56} Blöcke scheuen und versuchen, ihn in einem *time-memory-tradeoff* auf Kosten einer höheren Rechenzeit zu kompensieren; trotzdem zeigt die grundsätzliche Möglichkeit einer solchen Attacke, daß der Sicherheitsvorsprung, den ein doppelter DES gegenüber dem einfachen bietet, deutlich geringer ist, als man auf den ersten Blick meint. In der Praxis spielt der doppelte DES daher keine Rolle.

c) Dreifach DES

Der nächste Schritt zu Erhöhung der Komplexität besteht in einer dreifachen Anwendung von DES. Auch hier kann man wieder eine *meet in the middle attack* anwenden, aber auf dem Weg zur Mitte muß von mindestens einer der beiden Seiten aus DES zweimal mit verschiedenen

Schlüsseln anwenden, hat dann also einen Aufwand der Größenordnung 2^{112}. Dreifach DES oder, wie man meist sagt, Triple DES, gilt daher heute noch als sicher: Nachdem die DES-Cracker-Attacke publik geworden war, zog die amerikanische Regierung die Zulassung von DES für weniger geheime Nachrichten im Regierungsbereich zurück und verlangte stattdessen Triple DES, genauso wurden in Deutschland die Euroscheckkarten ersetzt durch neue Karten, die auf Triple DES anstelle des einfachen DES beruhen.

Soweit bekannt, bietet ein dreifacher DES mit drei verschiedenen Schlüsseln keine zusätzliche Sicherheit gegenüber einem mit nur zwei Schlüsseln, wobei die erste und die dritte Verschlüsselung denselben Schlüssel benutzen. Meist wird Triple DES in folgender Weise angewandt: Ein Block wird mit dem ersten Schlüssel *ver*schlüsselt, mit dem zweiten *ent*schlüsselt und dann noch einmal mit dem ersten *ver*schlüsselt.

d) Operationsmodi

Bislang haben wir Blockchiffren stets als Permutationen auf der Menge aller Blöcke betrachtet. Diese einfachste Anwendung einer Chiffre bezeichnet man als den *electronic code book* Modus, kurz ECB, da es zumindest im Prinzip möglich wäre, alle Blöcke und ihre Chiffrierung für einen gegebenen Schlüssel in einem Buch festzuhalten.

Für manche Anwendungen hat dieser ECB-Modus allerdings Schwächen und Sicherheitslücken: Falls beispielsweise nur wenige kurze Nachrichten (etwa „Ja“ und „Nein“) häufig übertragen werden, weiß ein Gegner, welche der gesendeten Nachrichten miteinander übereinstimmen und kann daraus vielleicht auch auf den Inhalt schließen.

Manchmal kann ein Gegner auch einfach dadurch Schaden anrichten, daß er unbemerkt die Reihenfolge von Nachrichtenblöcken vertauscht oder aber einen Nachrichtenblock *mehrfach* übermittelt. Auch gegen solche Attacken bietet ECB keinen Schutz.

Wir wollen daher im folgenden die drei weiteren für DES normierten Operationsmodi kennenlernen, die zumindest in diesen Fällen größere Sicherheit bieten.

Grundsätzlich lassen sich alle diese Modi auf *jede* Blockchiffre anwenden; sie werden hier im Zusammenhang mit DES behandelt, da für die Nachfolgechiffre AES bislang noch keine Operationsmodi festgelegt sind.

Alle drei Operationsmodi hängen außer vom DES-Schlüssel noch von einem Anfangsblock ab; dieser muß nicht geheim sein, kann aber als zusätzlicher Sicherheitsfaktor geheimgehalten werden.

Beim *cipher block chaining* (CBC) Modus wird zunächst der Anfangsblock mittels XOR zum ersten Block der Nachricht addiert; das Ergebnis wird verschlüsselt und übertragen. In jedem folgenden Schritt wird das letzte Verschlüsselungsergebnis zum zu verschlüsselnden Block addiert und diese Summe verschlüsselt und übertragen.

Da somit jeder übertragene Block von *allen* Vorgängerblocks abhängt, kann kein Gegner unbemerkt einzelne Blöcke unterdrücken, duplizieren oder aber in ihrer Reihenfolge verändern.

Diesem Vorteil steht der Nachteil gegenüber, daß ein etwaiger Übertragungsfehler nicht nur *einen* Block betrifft (und damit unentschlüsselbar macht), sondern auch noch den Folgeblock. Man überlegt sich aber leicht, daß danach wieder alles korrekt entschlüsselbar ist. Entsprechend führt ein verlorener Block nur zur Nichtentzifferbarkeit *eines* Folgeblocks, danach ist die Chiffre wieder synchronisiert.

Die beiden noch folgenden Modi sind nützlich, wenn Daten in Echtzeit übertragen werden sollen, die kürzer sind als die Blocklänge; die Länge der zu übertragenen Daten sei k Bit. In Anwendungen des DES ist meist $k = 8$, gelegentlich auch $k = 1$.

Bei *cipher feedback* (CFB) Modus wird zunächst ein Anfangsblock gewählt, bei dem rechts k Bit stehen; der Rest dieses Blocks besteht aus Nullen. Dieser Block kommt in ein Register, das mit DES verschlüsselt wird. Sodann werden die *ersten* k Bit des Ergebnisses zu den ersten k Bit der Nachricht addiert und dies wird übertragen.

In jedem folgenden Schritt wird der Inhalt des Registers um k Bit (nichtzyklisch) nach links verschoben, und die k zuletzt übertragenen Bits

werden am rechten Ende eingesetzt. Das Register wird dann wieder mit DES verschlüsselt, und die ersten k Bit werden zum nächsten Nachrichtenblock addiert und übertragen.

Auch hier hängt wieder jedes übertragene Wort aus k Bit von allen Vorgängern ab; Übertragungsfehler beeinflussen die Folgewörter so lange, wie noch ein fehlerhaft übertragenes Chiffrewort im Register steht.

Bei Satellitenübertragungen, bei denen Bitfehler relativ häufig sind, ist dies nicht tolerierbar; hier verwendet man stattdessen den *output feedback* (OFB) Modus, der im wesentlichen genauso funktioniert wie CFB, nur daß jetzt anstelle des Chiffretexts die linken k Bit des Registers rechts eingefügt werden, d.h. das Register wird zyklisch um k Bit verschoben.

In diesem Modus hängt der Inhalt des Registers weder vom Klartext noch vom Chiffretext ab; DES wird also nur dazu benutzt, einen Schlüsselsgtrom zu erzeugen, der zur Nachricht addiert wird. Im OFB Modus haben wir somit eine Stromchiffre, die zwischen dem *one time pad* und VIGENÈRE liegt: Der Schlüsselstrom hängt zwar nur von wenigen Bit ab, entsteht daraus aber nicht durch bloße Wiederholung, sondern via DES (oder einer anderen Blockchiffre) mit einem geheimen Schlüssel.

Bei dieser Übertragungsart erzeugen Sender und Empfänger die Schlüsselströme unabhängig von der Nachricht, ein Bitfehler bei der Übertragung betrifft also nur ein einziges Bit.

Diesem Vorteil steht der Nachteil entgegen, daß man hier für jede Übertragung einen anderen Anfangsblock und/oder Schlüssel wählen muß, denn ansonsten wird mehrfach derselbe Schlüsselstrom verwendet, ein Gegner weiß also, welche k-Bit-Blöcke jeweils mit demselben Schlüssel chiffriert wurden und kann Häufigkeitsanalysen anwenden.