4 Symmetrische Kryptosysteme und ihre Analyse

4.1 Produktchiffren

Produktchiffren erhält man durch die sequentielle Anwendung mehrerer Verschlüsselungsverfahren. Sie können extrem schwer zu brechen sein, auch wenn die einzelnen Komponenten leicht zu brechen sind.

Definition 87 (**Produktkryptosystem**)

Seien $S_1 = (M_1, C_1, E_1, D_1, K_1)$ und $S_2 = (M_2, C_2, E_2, D_2, K_2)$ Kryptosysteme mit $C_1 = M_2$. Dann ist das **Produktkryptosystem** von S_1 und S_2 definiert als $S_1 \times S_2 = (M_1, C_2, E, D, K_1 \times K_2)$ mit

$$E(k_1, k_2; x) = E_2(k_2, E_1(k_1, x)) \text{ und } D(k_1, k_2; y) = D_1(k_1, D_2(k_2, y))$$

für alle $x \in M_1$, $y \in C_2$ und $(k_1, k_2) \in K_1 \times K_2$.

Der Schlüsselraum von $S_1 \times S_2$ umfasst also alle Paare (k_1, k_2) von Schlüsseln $k_1 \in K_1$ und $k_2 \in K_2$, wobei wir voraussetzen, dass die Schlüssel unabhängig gewählt werden (d.h. es gilt $p(k_1, k_2) = p(k_1)p(k_2)$).

Beispiel 88 *Sei $A = \{a_0, \ldots, a_{m-1}\}$. Man sieht leicht, dass die affine Chiffre $S = (M, C, K, E, D)$ mit $M = C = \mathcal{A}$ und $K = \mathbb{Z}_m^* \times \mathbb{Z}_m$ das Produkt $S = S_1 \times S_2$ der multiplikativen Chiffre $S_1 = (M, C, K_1, E_1, D_1)$ mit der additiven Chiffre $S_2 = (M, C, K_2, E_2, D_2)$ ist, da für jeden Schlüssel $k = (k_1, k_2) \in K = \mathbb{Z}_m^* \times \mathbb{Z}_m$ gilt:*

$$E(k, x) = k_1 x + k_2 = E_2(k_2, E_1(k_1, x)).$$

Für $S' = S_2 \times S_1$ erhalten wir das Kryptosystem $S' = (M, C, K', E', D')$ mit $K' = \mathbb{Z}_m \times \mathbb{Z}_m^*$ und

$$E'(k_1, k_2; x) = k_2(x + k_1) = k_2 x + k_2 k_1 = E(k_2, k_2 k_1; x)$$

für jeden Schlüssel $(k_1, k_2) \in K'$. Da die Abbildung

$$(k_1, k_2) \mapsto (k_2, k_2 k_1)$$

eine Bijektion zwischen den Schlüsselräumen K' und K ist und der Schlüssel (k_1, k_2) im System S' die gleiche Chiffrierfunktion realisiert wie der Schlüssel $(k_2, k_2 k_1)$ in S, sind die Kryptosysteme $S = S_1 \times S_2$ und $S' = S_2 \times S_1$ als gleich (genauer: äquivalent, siehe Übungen) anzusehen, d.h. S_1 und S_2 kommutieren. ◁

Definition 89 (**endomorph, idempotent**)

Ein Kryptosystem $S = (M, C, K, D, E)$ mit $M = C$ heißt **endomorph**.
Ein endomorphes Kryptosystem S heißt **idempotent**, falls $S \times S = S$ ist.

Beispiel 90 *Eine leichte Rechnung zeigt, dass die additive, die multiplikative und die affine Chiffre idempotent sind. Ebenso die Blocktransposition sowie die Vigenère- und Hill-Chiffre.*

Will man durch mehrmalige Anwendung (Iteration) derselben Chiffriermethode eine höhere Sicherheit erreichen, so darf diese nicht idempotent sein. Man kann beispielsweise versuchen, ein nicht idempotentes System S durch die Kombination $S = S_1 \times S_2$ zweier idempotenter Verfahren S_1 und S_2 zu erhalten. Wegen

$$\begin{aligned}(S_1 \times S_2) \times (S_1 \times S_2) &= S_1 \times (S_2 \times S_1) \times S_2 \\ &= S_1 \times (S_1 \times S_2) \times S_2 \\ &= (S_1 \times S_1) \times (S_2 \times S_2) \\ &= S_1 \times S_2\end{aligned}$$

dürfen hierbei S_1 und S_2 jedoch nicht kommutieren.

Im Rest dieses Kapitels werden wir nur noch das Binäralphabet $A = \{0,1\}$ als Klar- und Kryptotextalphabet benutzen und auch der Schlüsselraum wird von der Form $\{0,1\}^k$ sein, wobei k die Schlüssellänge bezeichnet.

Eine iterierte Blockchiffre wird typischerweise durch eine Rundenfunktion (*round function*) g und einen Schlüsselgenerator (*key schedule algorithm*) f beschrieben. Ist N die Rundenzahl, so erzeugt f bei Eingabe eines Schlüssels K eine Folge $f(K) = (K^1, \ldots, K^N)$ von N Rundenschlüsseln K^i für g. Mit diesen wird ein Klartext $x = w^0$ durch N-malige Anwendung der Rundenfunktion g zu einem Kryptotext $y = w^N$ verschlüsselt:

$$\begin{aligned}&w^1 \leftarrow g(K^1, w^0) \\ &\quad\vdots \\ &w^N \leftarrow g(K^N, w^{N-1})\end{aligned}$$

Um y wieder zu entschlüsseln, muss die inverse Rundenfunktion g^{-1} mit umgekehrter Rundenschlüsselfolge $K^N, \ldots, K^1$ benutzt werden:

$$\begin{aligned}&w^{N-1} \leftarrow g^{-1}(K^N, w^N) \\ &\quad\vdots \\ &w^0 \leftarrow g^{-1}(K^1, w^1)\end{aligned}$$

Beispiele für iterierte Chiffren sind der aus 16 Runden bestehende DES-Algorithmus und der AES mit einer variablen Rundenzahl $N \in \{10, 12, 14\}$, die wir in späteren Abschnitten behandeln werden.

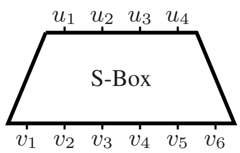
4.2 Substitutions-Permutations-Netzwerke

In diesem Abschnitt betrachten wir den prinzipiellen Aufbau von iterierten Blockchiffren. Als Basisbausteine für die Rundenfunktion eignen sich Substitutionen und Transpositionen besonders gut. Aus Effizienzgründen sollten die Substitutionen nur eine relativ kleine Blocklänge l haben.

Definition 91 (**Teilwort**)

Für ein Wort $u = u_1 \cdots u_n \in \{0,1\}^n$ und Indizes $1 \leq i \leq j \leq n$ bezeichne $u[i,j]$ das **Teilwort** $u_i \cdots u_j$ von u. Im Fall $n = lm$ bezeichnen wir das Teilwort $u[(i-1)l+1, il]$ auch einfach mit $u_{(i)}$, d.h. es gilt $u = u_{(1)} \cdots u_{(m)}$, wobei $|u_{(i)}| = l$.

Sei $\pi_S : A^l \to A^{l'}$ eine Substitution, die Binärblöcke u der Länge l in Binärblöcke $v = \pi_S(u)$ der Länge l' überführt (engl. auch als **S-Box** bezeichnet).



Durch parallele Anwendung von m dieser S-Boxen erhalten wir folgende Substitution $S : A^{lm} \to A^{l'm}$,

$$S(u_1 \cdots n_{lm}) = \pi_S(u_{(1)}) \cdots \pi_S(u_{(m)}).$$

Für die Speicherung einer S-Box $\pi_S : \{0,1\}^l \to \{0,1\}^{l'}$ auf einem Speicherchip werden $l'2^l$ bit Speicherplatz benötigt (im Fall $l = l'$ also $l2^l$ bit). Für $l = l' = 16$ wären dies beispielsweise 2^{20} bit, was Smartcard-Anwendungen bereits ausschließen würde.

Für eine Transposition P auf A^{lm} bezeichnen wir die zugehörige Permutation auf $\{1, \ldots, lm\}$ mit π_P, d.h.

$$P(u_1 \cdots u_{lm}) = u_{\pi_P(1)} \cdots u_{\pi_P(m)}.$$

Definition 92 (**Substitutions-Permutations-Netzwerk**)

Sei $A = \{0,1\}$ und sei $M = C = A^{lm}$ für natürliche Zahlen $l, m \geq 1$. Ein **Substitutions-Permutations-Netzwerk** (SPN) wird durch Permutationen $\pi_S : \{0,1\}^l \to \{0,1\}^l$ und $\pi_P : \{1, \ldots, lm\} \to \{1, \ldots, lm\}$ sowie durch einen Schlüsselgenerator $f : \{0,1\}^k \to \{0,1\}^{lm(N+1)}$ beschrieben. Der Generator f erzeugt aus einem (externen) Schlüssel $K \in \{0,1\}^k$ eine Folge $f(K) = (K^1, \ldots, K^{N+1})$ von $N+1$ Rundenschlüsseln K^r, unter denen ein Klartext $x \in \{0,1\}^{lm}$ gemäß folgendem Algorithmus in einen Kryptotext $y = E_{f,\pi_S,\pi_P}(K,x) \in \{0,1\}^{lm}$ überführt wird.

Algorithmus 93 $E_{f,\pi_S,\pi_P}(K,x)$

$w^0 \leftarrow x$
for $r \leftarrow 1$ ***to*** $N-1$ ***do***
  $u^r \leftarrow w^{r-1} \oplus K^r$
  $v^r \leftarrow S(u^r)$
  $w^r \leftarrow P(v^r)$
end
$u^N \leftarrow w^{N-1} \oplus K^N$
$v^N \leftarrow S(u^N)$
$y \leftarrow v^N \oplus K^{N+1}$

Zu Beginn jeder Runde $r \in \{1, \ldots, N\}$ wird w^{r-1} zunächst einer XOR-Operation mit dem Rundenschlüssel K^r unterworfen (dies wird *round key mixing* genannt), deren Resultat u^r den S-Boxen zugeführt wird. Auf die Ausgabe v^r der S-Boxen wird in jeder Runde $r \leq N-1$ die Transposition P angewendet, was die Eingabe w^r für die nächste Runde $r+1$ liefert.

Am Ende der letzten Runde $r = N$ wird nicht die Transposition P angewandt, sondern der Rundenschlüssel K^{N+1} auf v^N addiert. Durch diese (*whitening* genannte) Vorgehensweise wird einerseits erreicht, dass auch für den letzten Chiffrierschritt der Schlüssel benötigt und somit der Gegner von einer partiellen Entschlüsselung des Kryptotexts abgehalten wird. Zum Zweiten ermöglicht dies eine (legale) Entschlüsselung nach fast demselben Verfahren (siehe Übungen).

Beispiel 94 *Sei* $l = m = N = 4$ *und sei* $k = 32$*. Für* f *wählen wir die Funktion* $f(K) = (K^1, \ldots, K^5)$ *mit* $K^r = K[4(r-1)+1, 4(r-1)+16]$*. Weiter seien* $\pi_S : \{0,1\}^4 \to \{0,1\}^4$ *und* $\pi_P : \{1, \ldots, 16\} \to \{1, \ldots, 16\}$ *die folgenden Permutationen (wobei die Argumente und Werte von* π_S *hexadezimal dargestellt sind; siehe auch Abbildung 1):*

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

und

z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_P(z)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

Für den Schlüssel $K = 0011\,1010\,1001\,0100\,1101\,0110\,0011\,1111$ *liefert* f *beispielsweise die Rundenschlüssel* $f(K) = (K^1, \ldots, K^5)$ *mit*

$$\begin{aligned}
K^1 &= 0011\,1010\,1001\,0100,\\
K^2 &= 1010\,1001\,0100\,1101,\\
K^3 &= 1001\,0100\,1101\,0110,\\
K^4 &= 0100\,1101\,0110\,0011,\\
K^5 &= 1101\,0110\,0011\,1111,
\end{aligned}$$

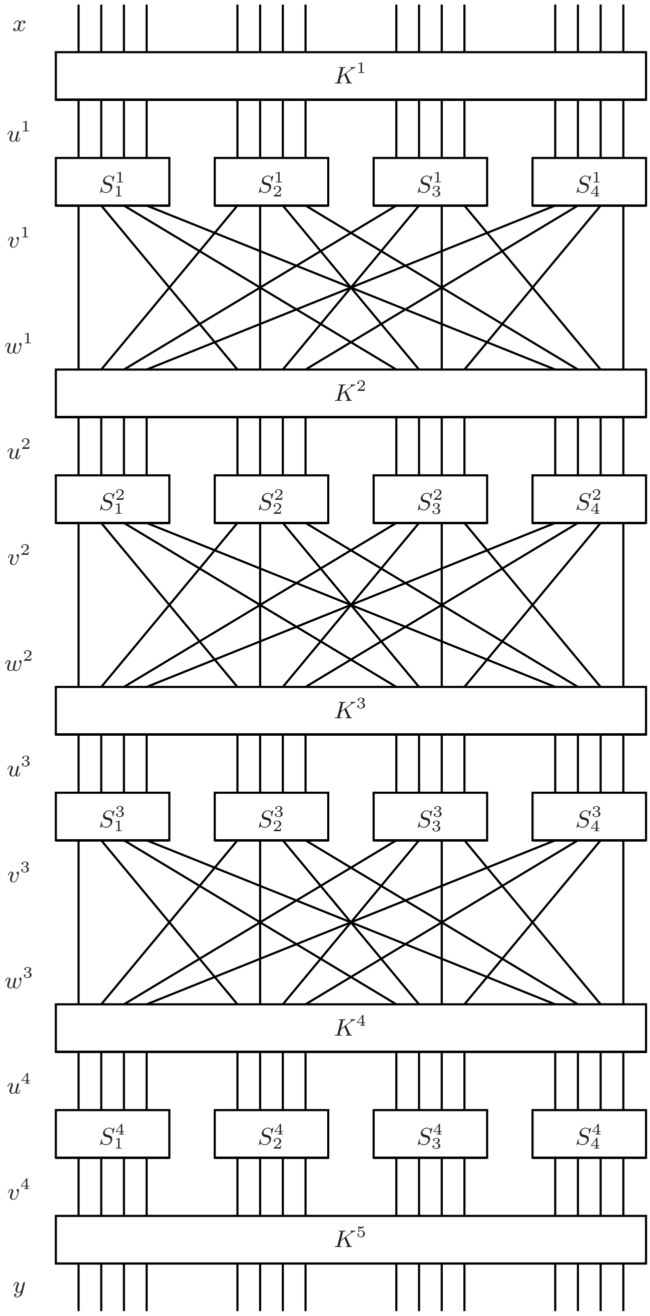


Abbildung 1 Ein Substitutions-Permutations-Netzwerk.

unter denen der Klartext $x = 0010\,0110\,1011\,0111$ *die folgenden Chiffrierschritte durchläuft:*

$$\begin{aligned} x &= 0010\,0110\,1011\,0111 = w^0 \\ w^0 \oplus K^1 &= 0001\,1100\,0010\,0011 = u^1 \\ S(u^1) &= 0100\,0101\,1101\,0001 = v^1 \\ P(v^1) &= 0010\,1110\,0000\,0111 = w^1 \\ &\ \ \vdots \\ P(v^3) &= 1110\,0100\,0110\,1110 = w^3 \\ w^3 \oplus K^4 &= 1010\,1001\,0000\,1101 = u^4 \\ S(u^4) &= 0110\,1010\,1110\,1001 = v^4 \\ u^4 \oplus K^5 &= 1011\,1100\,1101\,0110 = y. \end{aligned}$$

4.3 Lineare Kryptoanalyse von SPNs

Zuerst stellen wir eine Reihe von Ergebnissen aus der Wahrscheinlichkeitstheorie bereit. Für eine Zufallsvariable X mit Wertebereich $W(X) = \{0,1\}$ und $p = Prob[X = 0]$ bezeichne $\varepsilon(X)$ den Wert $\varepsilon(X) = p - 1/2$ (auch *bias* von X genannt). Sind X_1, X_2 unabhängige Zufallsvariablen mit Wertebereich $W(X_i) = \{0,1\}$ und bias-Wert $\varepsilon_i = \varepsilon(X_i)$, dann ist

$$\begin{aligned} Prob[X_1 \oplus X_2 = 0] &= Prob[X_1 = X_2 = 0] + Prob[X_1 = X_2 = 1] \\ &= (1/2 + \varepsilon_1)(1/2 + \varepsilon_2) + (1/2 - \varepsilon_1)(1/2 - \varepsilon_2) \\ &= 1/2 + 2\varepsilon_1\varepsilon_2 \end{aligned}$$

und $Prob[X_1 \oplus X_2 = 1] = 1/2 - 2\varepsilon_1\varepsilon_2$, d.h. es gilt $\varepsilon(X_1 \oplus X_2) = 2\varepsilon_1\varepsilon_2$. Diese Beobachtung lässt sich leicht verallgemeinern.

Lemma 95 (Piling-up Lemma)
Seien $X_1, \ldots, X_n$ unabhängige $\{0,1\}$-wertige Zufallsvariablen mit bias-Werten $\varepsilon_i = \varepsilon(X_i)$. Dann gilt

$$\varepsilon(X_1 \oplus \cdots \oplus X_n) = 2^{n-1} \prod_{i=1}^{n} \varepsilon_i.$$

Beweis: Wir führen den Beweis durch Induktion über n.

Induktionsanfang ($n = 1$): Klar.

Induktionsschritt ($n \rightsquigarrow n+1$): Nach IV hat die Zufallsvariable $Z = X_1 \oplus \cdots \oplus X_n$ den bias-Wert $\varepsilon(Z) = 2^{n-1}\varepsilon(X_1) \cdots \varepsilon(X_n)$ und daher folgt

$$\varepsilon(X_1 \oplus \cdots \oplus X_{n+1}) = \varepsilon(Z \oplus X_{n+1}) = 2\varepsilon(Z)\varepsilon_{n+1} = 2^n \varepsilon_1 \cdots \varepsilon_{n+1}.$$

■

Beispiel 96 *Seien X_1, X_2, X_3 Zufallsvariablen mit $\varepsilon(X_i) = 1/4$ für $i = 1, 2, 3$. Dann gilt nach obigem Lemma $\varepsilon_{i,j} = \varepsilon(X_i \oplus X_j) = 1/8$ für $1 \leq i < j \leq 3$. Man beachte, dass die Zufallsvariablen $X_1 \oplus X_2$ und $X_2 \oplus X_3$ nicht unabhängig sind, und daher das Piling-up-Lemma nicht anwendbar ist. Dieses würde nämlich für die Zufallsvariable*

$$(X_1 \oplus X_2) \oplus (X_2 \oplus X_3) = X_1 \oplus X_3$$

einen bias-Wert von $\varepsilon = 2(1/8)^2 = 1/32$ ergeben, was dem tatsächlichen Wert $\varepsilon(X_1 \oplus X_3) = \varepsilon_{1,3} = 1/8$ widersprechen würde.

Lineare Approximationen

Sei $f : \{0,1\}^l \to \{0,1\}^{l'}$ eine Abbildung. Wählen wir für f eine zufällige Eingabe $U = U_1 \cdots U_l$ unter Gleichverteilung, so gilt für die zugehörige Ausgabe $V = f(U) = V_1 \cdots V_{l'}$,

$$Prob[V = v \mid U = u] = \begin{cases} 1 & \pi_S(u) = v, \\ 0 & \text{sonst} \end{cases}$$

für alle $u \in \{0,1\}^l$ und $v \in \{0,1\}^{l'}$. Wegen $Prob[U = u] = 2^{-l}$ folgt

$$Prob[V = v, U = u] = \begin{cases} 2^{-l} & \pi_S(u) = v, \\ 0 & \text{sonst.} \end{cases}$$

Ist f linear, so sind die Zufallsvariablen V_j in der Form

$$V_j = U_{i_1} \oplus \cdots \oplus U_{i_k}$$

für geeignete Indizes $1 \leq i_1 < \cdots < i_k \leq l$ darstellbar. Die Idee hinter der linearen Kryptoanalyse ist nun, Gleichungen der Form

$$V_{j_1} \oplus \cdots \oplus V_{j_{k'}} = U_{i_1} \oplus \cdots \oplus U_{i_k} \oplus c$$

mit $1 \leq i_1 < \cdots < i_k \leq l$, $1 \leq j_1 < \cdots < j_{k'} \leq l'$ und $c \in \{0,1\}$ zu finden, die mit großer WK gelten. Definieren wir für $a \in \{0,1\}^l$ und $b \in \{0,1\}^{l'}$ die Zufallsvariablen

$$U_a = \bigoplus_{i=1}^{l} a_i U_i \text{ und } V_b = \bigoplus_{i=1}^{l'} b_i V_i,$$

so sind wir also an solchen Werten für a, b und c interessiert, für die das Ereignis $V_b = U_a \oplus c$ (oder gleichbedeutend: $U_a \oplus V_b \oplus c = 0$) eine möglichst große Wahrscheinlichkeit besitzt. Dies bedeutet, dass die Zufallsvariable $U_a \oplus V_b \oplus c$ einen möglichst großen bias-Wert $\varepsilon(U_a \oplus V_b \oplus c)$ haben sollte. Wegen

$$\varepsilon(U_a \oplus V_b \oplus 1) = -\varepsilon(U_a \oplus V_b)$$

ist die durch a und b beschriebene lineare Approximation $U_a \oplus V_b$ also um so besser, je größer der Absolutbetrag $|\varepsilon(U_a \oplus V_b)|$ des bias-Wertes dieser Approximation ist.

Beispiel 97 *Wir betrachten die S-Box* $\pi_S : \{0,1\}^4 \to \{0,1\}^4$ *aus Beispiel 94. Dann nimmt die Zufallsvariable* $(U_1,\ldots,U_4,V_1,\ldots,V_4)$ *die folgenden 16 Werte jeweils mit Wahrscheinlichkeit* $2^{-4} = 1/16$ *an.*

U_1	U_2	U_3	U_4	V_1	V_2	V_3	V_4	$U_3 \oplus U_4 \oplus V_1 \oplus V_4$
0	0	0	0	1	1	1	0	1
0	0	0	1	0	1	0	0	1
0	0	1	0	1	1	0	1	1
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	1	0	0
0	1	0	1	1	1	1	1	1
0	1	1	0	1	0	1	1	1
0	1	1	1	1	0	0	0	1
1	0	0	0	0	0	1	1	1
1	0	0	1	1	0	1	0	0
1	0	1	0	0	1	1	0	1
1	0	1	1	1	1	0	0	1
1	1	0	0	0	1	0	1	1
1	1	0	1	1	0	0	1	1
1	1	1	0	0	0	0	0	1
1	1	1	1	0	1	1	1	1

Um nun $\varepsilon(U_a \oplus V_b)$ *zu berechnen, genügt es, die Anzahl* $L(a,b)$ *der Zeilen zu bestimmen, für die* $U_a = V_b$ *ist. Dann gilt* $Prob[U_a \oplus V_b = 0] = Prob[U_a = V_b] = L(a,b)/16$ *und somit*

$$\varepsilon(U_a \oplus V_b) = L(a,b)/16 - 1/2 = (L(a,b)-8)/16.$$

Für $a = 0011$ *und* $b = 1001$ *gibt es z.B.* $L(a,b) = 2$ *Zeilen (Zeile 5 und Zeile 10) mit* $U_a = U_3 \oplus U_4 = V_b = V_1 \oplus V_4$*, d.h.* $\varepsilon(U_3 \oplus U_4 \oplus V_1 \oplus V_4) = (L(a,b)-8)/16 = -3/8$*. Die folgende Tabelle zeigt für alle Werte von* a *und* b *(hexadezimal dargestellt) die Anzahlen* $L(a,b)$*.*

	b															
a	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	6	6	8	8	6	14	10	10	8	8	10	10	8	8
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8	8	8	8	8	8	8	8	10	**2**	6	6	10	10	6	6
$\vdots$									$\vdots$							
F	8	6	4	6	6	8	10	8	8	6	12	6	6	8	10	8

Lineare Kryptoanalyse eines SPN

Wir betrachten nun das SPN aus Beispiel 94 und führen eine lineare Kryptoanalyse durch. Dabei handelt es sich um einen Angriff bei bekanntem Klartext, d.h. es steht eine Menge M von t Klartext-Kryptotext-Paaren (x,y) zur Verfügung, die alle mit dem gleichen unbekannten Schlüssel K erzeugt wurden.

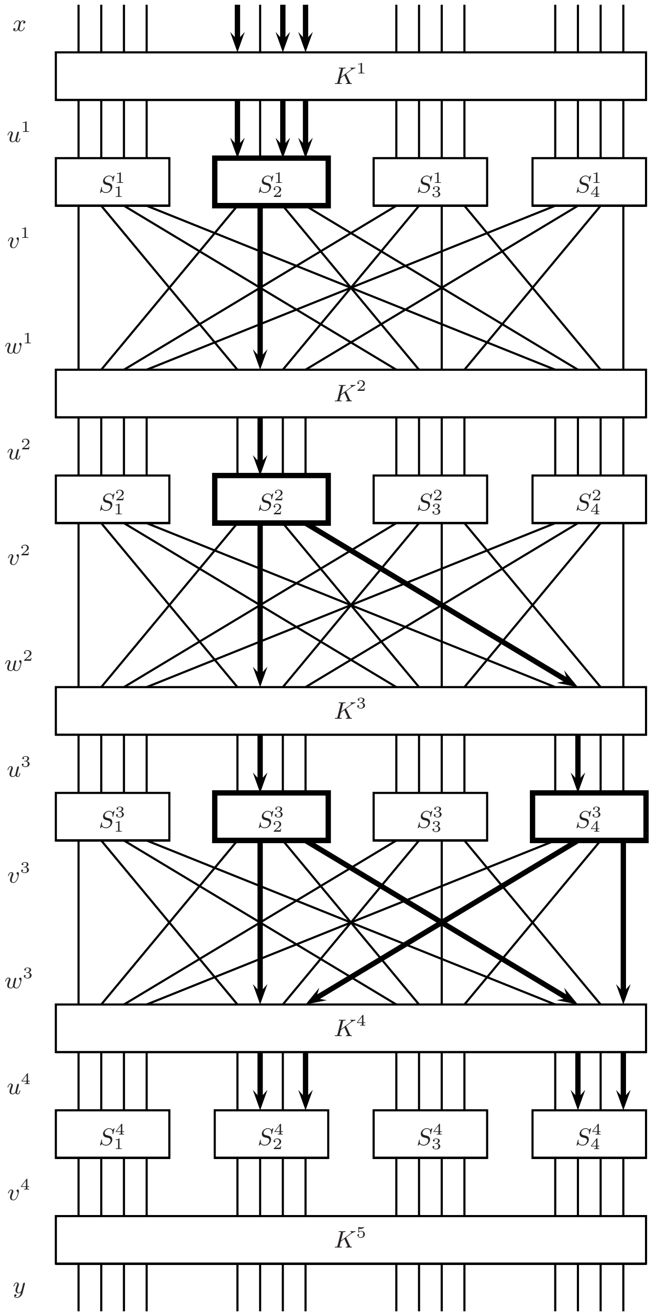


Abbildung 2 Eine lineare Approximation an ein Substitutions-Permutations-Netzwerk.

Seien $K^1, \ldots, K^5$ die zu K gehörigen Rundenschlüssel. Das Ziel besteht zunächst einmal darin, eine lineare Approximation für die Abbildung $x \mapsto u^4$ zu finden, bei der die Rundenschlüssel $K^1, \ldots, K^4$ benutzt werden (siehe Abbildung 2). Hierzu benutzen wir die beiden folgenden linearen Approximationen an die S-Box S:

$$T = U_1 \oplus U_3 \oplus U_4 \oplus V_2$$

mit einem bias-Wert von $\varepsilon(T) = (L(B,4) - 8)/16 = (12 - 8)/16 = 1/4$ und

$$T' = U_2 \oplus V_2 \oplus V_4$$

mit einem bias-Wert von $\varepsilon(T') = (L(4,5) - 8)/16 = (4 - 8)/16 = -1/4$.

Konkret benutzen wir die lineare Approximation T für die S-Box S_2^1,

$$T_1 = U_5^1 \oplus U_7^1 \oplus U_8^1 \oplus V_6^1$$

und die lineare Approximation T' für die S-Boxen S_2^2, S_2^3, S_4^3,

$$\begin{aligned} T_2 &= U_6^2 \oplus V_6^2 \oplus V_8^2, \\ T_3 &= U_6^3 \oplus V_6^3 \oplus V_8^3, \\ T_4 &= U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3. \end{aligned}$$

Indem wir nun die linearen Approximationen $T_1, \ldots, T_4$ der S-Boxen S_2^1, S_2^2, S_2^3 und S_4^3 „zusammen schalten“, erhalten wir für ein $c' \in \{0,1\}$ die gesuchte lineare Approximation

$$X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4 \quad = \quad T_1 \oplus T_2 \oplus T_3 \oplus T_4 \oplus c' \qquad (3)$$

von $x \mapsto u^4$. An dieser Stelle ergeben sich folgende drei Fragen.

1. Warum gilt (3)?
2. Wie gut ist die lineare Approximation $X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4$?
3. Wie können wir mit ihrer Hilfe den Schlüssel bestimmen?

Wir gehen zunächst auf Frage 1 ein. Drücken wir die U-Variablen, die in $T_1, \ldots, T_4$ vorkommen, durch X-Variablen und V-Variablen sowie durch Schlüsselbits K_i^r aus,

$$\begin{aligned} U_5^1 &= X_5 \oplus K_5^1, \\ U_7^1 &= X_7 \oplus K_7^1, \\ U_8^1 &= X_8 \oplus K_8^1, \\ U_6^2 &= W_6^1 \oplus K_6^2 = V_6^1 \oplus K_6^2, \\ U_6^3 &= W_6^2 \oplus K_6^3 = V_6^2 \oplus K_6^3, \\ U_{14}^3 &= W_{14}^2 \oplus K_{14}^3 = V_8^2 \oplus K_{14}^3, \end{aligned}$$

so führt dies mit $c = K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3$ auf die Darstellung

$$T_1 \oplus \cdots \oplus T_4 = X_5 \oplus X_7 \oplus X_8 \oplus V_6^3 \oplus V_8^3 \oplus V_{14}^3 \oplus V_{16}^3 \oplus c.$$

Jetzt müssen wir nur noch die verbliebenen V^3-Variablen durch U^4-Variablen und Schlüsselbits ersetzen,

$$\begin{aligned} V_6^3 &= U_6^4 \oplus K_6^4, \\ V_8^3 &= U_{14}^4 \oplus K_{14}^4, \\ V_{14}^3 &= U_8^4 \oplus K_8^4, \\ V_{16}^3 &= U_{16}^4 \oplus K_{16}^4, \end{aligned}$$

um mit $c' = c \oplus K_6^4 \oplus K_8^4 \oplus K_{14}^4 \oplus K_{16}^4$ die gesuchte Darstellung zu erhalten:

$$T_1 \oplus \cdots \oplus T_4 = X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4 \oplus c'.$$

Nun zu Frage 2: Wären die Zufallsvariablen $T_1, \ldots, T_4$ unabhängig, so würde uns das Piling-up-Lemma den bias-Wert $2^3(1/4)(-1/4)^3 = -1/32$ für $T_1 \oplus \cdots \oplus T_4$ liefern. Obwohl dies nicht der Fall ist, stellt sich in der Praxis heraus, dass sich der tatsächliche Wert $\varepsilon = \varepsilon(T_1 \oplus \cdots \oplus T_4)$ nicht zu sehr von diesem "hypothetischen" Wert unterscheidet, d.h.

$$|\varepsilon(X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4)| \approx {}^1\!/_{32}.$$

Und schließlich zu Frage 3: Wir wissen bereits, dass ein zufälliger Klartext X entweder mit hoher oder mit niedriger Wahrscheinlichkeit auf ein Zwischenresultat U^4 mit

$$X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4 = 0 \tag{4}$$

führt. Gehen wir also davon aus, dass M eine repräsentative Auswahl von Klartext-Kryptotext-Paaren (x, y) darstellt, so wird die Anzahl der Paare (x, y) in M, die (4) erfüllen, ebenfalls eine Mehrheit oder eine Minderheit in M bilden. Man beachte, dass sich für jeden Subschlüssel-Kandidaten (engl. *candidate subkey*) (L_1, L_2) für $(K_{(2)}^5, K_{(4)}^5)$ die zu einem Kryptotext y gehörigen Werte u_6^4, u_8^4, u_{14}^4 und u_{16}^4 leicht berechnen lassen, da π_S^{-1} bekannt ist.

Die Idee besteht nun darin, für jeden Kandidaten (L_1, L_2) die Anzahl $\alpha(L_1, L_2)$ aller Paare (x, y) in M zu bestimmen, die bei Benützung von (L_1, L_2) Gleichung (4) erfüllen. Für den richtigen Kandidaten wird diese Anzahl ungefähr bei ${}^t\!/_2 \pm {}^t\!/_{32}$ liegen, wogegen bei Benutzung eines falschen Subschlüssels mit einer Anzahl von circa ${}^t\!/_2$ zu rechnen ist. Für genügend große Werte von t lassen sich auf diese Weise 8 bit von K^5 (und damit von K) bestimmen.

Algorithmus 98 *LinearAttack*

for $(L_1, L_2) \leftarrow (0, 0)$ ***to*** *(F,F)* ***do***
  $\alpha(L_1, L_2) \leftarrow 0$ ***end***
for each $(x, y) \in M$ ***do***
  for $(L_1, L_2) \leftarrow (0, 0)$ ***to*** (F, F) ***do***
    $v_{(2)}^4 \leftarrow L_1 \oplus y_{(2)}$
    $v_{(4)}^4 \leftarrow L_2 \oplus y_{(4)}$
    $u_{(2)}^4 \leftarrow \pi_S^{-1}(v_{(2)}^4)$
    $u_{(4)}^4 \leftarrow \pi_S^{-1}(v_{(4)}^4)$

      if $x_5 \oplus x_7 \oplus x_8 \oplus u_6^4 \oplus u_8^4 \oplus u_{14}^4 \oplus u_{16}^4 = 0$ ***then***
        $\alpha(L_1, L_2) \leftarrow \alpha(L_1, L_2) + 1$
    end
  end
  $max \leftarrow -1$
  for $(L_1, L_2) \leftarrow (0, 0)$ ***to*** (F, F) ***do***
    $\beta(L_1, L_2) \leftarrow |\alpha(L_1, L_2) - t/2|$
    if $\beta(L_1, L_2) > max$ ***then***
      $max \leftarrow \beta(L_1, L_2)$
      $maxkey \leftarrow (L_1, L_2)$
  end
Ausgabe: $maxkey$

Im allgemeinen werden für eine erfolgreiche lineare Attacke circa $t \approx c\varepsilon^{-2}$ Klartext-Kryptotext-Paare benötigt, wobei c eine „kleine“ Konstante ist (im Beispielfall reichen $t \approx 8000$ Paare, d.h. $c \approx 8$, da $\varepsilon^{-2} = 1024$ ist.

4.4 Differentielle Kryptoanalyse von SPNs

Bei der differentiellen Kryptoanalyse handelt es sich um einen Angriff bei frei wählbarem Klartext. Genauer gesagt, basiert der Angriff auf einer Menge M von t Klartext-Kryptotext-Doppelpaaren (x, x^*, y, y^*) mit der Eigenschaft, dass alle Klartext-Paare (x, x^*) die gleiche Differenz $x' = x \oplus x^*$ bilden.

Definition 99 (**Eingabe- und Ausgabedifferenz**)

Seien $x, x^* \in \{0,1\}^l$ zwei Eingaben für eine S-Box $\pi_S : \{0,1\}^l \to \{0,1\}^{l'}$ und seien $y = \pi_S(x)$ und $y^* = \pi_S(x^*)$ die zugehörigen Ausgaben. Dann wird $x' = x \oplus x^*$ die **Eingabedifferenz** (engl. input-x-or) und $y' = \pi_S(x) \oplus \pi_S(x^*)$ die **Ausgabedifferenz** (engl. output-x-or) des Paares (x, x^*) genannt. Für eine vorgegebene Eingabedifferenz $a' \in \{0,1\}^l\}$ sei weiter

$$\begin{aligned} \Delta(a') &= \{(x, x^*) \mid x, x^* \in \{0,1\}^l, x \oplus x^* = a'\} \\ &= \{(x, x \oplus a') \mid x \in \{0,1\}^l\} \end{aligned}$$

die Menge aller Eingabepaare, die die Differenz a' realisieren.

Berechnen wir für alle Eingabepaare $(x, x^*) \in \Delta(a')$ die zugehörigen Ausgabedifferenzen, so verteilen sich diese mehr oder weniger gleichmäßig auf die $2^{l'}$ möglichen Werte in $\{0,1\}^{l'}$. Man beachte, dass im Fall einer linearen S-Box nur die Ausgabedifferenz $\pi_S(a')$ auftritt, da dann $\pi_S(x) \oplus \pi_S(x^*) = \pi_S(x \oplus x^*)$ ist. Ist dagegen π_S nicht linear, so kann die Eingabedifferenz a' auf unterschiedliche Ausgabedifferenzen führen, je nachdem, durch welches Eingabenpaar $(x, x^*) \in \Delta(a')$ die Differenz a' realisiert wird. Im Allgemeinen lässt sich eine differentielle Kryptoanalyse um so leichter durchführen, je ungleichmäßiger die auftretenden Ausgabedifferenzen verteilt sind.

Definition 100 (**Differential, Weitergabequotient**)

Sei $a' \in \{0,1\}^l$ eine Eingabe- und sei $b' \in \{0,1\}^{l'}$ eine Ausgabedifferenz für eine S-Box π_S. Dann heißt (a', b') **Differential**. Die Anzahl der Eingabepaare (x, x^*), die die Eingabedifferenz a' in die Ausgabedifferenz b' überführen, bezeichnen wir mit $D(a', b')$, d.h.

$$D(a', b') = \|\{(x, x^*) \in \Delta(a') \mid \pi_S(x) \oplus \pi_S(x^*) = b'\}\|.$$

Der **Weitergabequotient** (engl. propagation ratio) von π_S für ein Differential (a', b') ist

$$Q(a', b') = \frac{D(a', b')}{2^l}.$$

$Q(a', b')$ ist also die (bedingte) Wk

$$Prob[\pi_S(x) \oplus \pi_S(x^*) = b' \mid x \oplus x^* = a'],$$

dass zwei zufällig gewählte Eingaben die Ausgabedifferenz b' erzeugen, wenn sie die Eingabedifferenz a' bilden.

Beispiel 101 *Betrachten wir die S-Box $\pi_S : \{0,1\}^4 \to \{0,1\}^4$ aus Beispiel 94, so erhalten wir für die Eingabedifferenz $a' = 1011$ die Menge*

$$\Delta(a') = \{(0000, 1011), \ldots, (1111, 0100)\}$$

von möglichen Eingabepaaren, die auf folgende Ausgabedifferenzen $y' = y \oplus y^* = \pi_S(x) \oplus \pi_S(x^*)$ führen:

x	x^*	y	y^*	y'
0000	1011	1110	1100	0010
0001	1010	0100	0110	0010
0010	1001	1101	1010	0111
0011	1000	0001	0011	0010
0100	1111	0010	0111	0101
0101	1110	1111	0000	1111
0110	1101	1011	1001	0010
0111	1100	1000	0101	1101
1000	0011	0011	0001	0010
1001	0010	1010	1101	0111
1010	0001	0110	0100	0010
1011	0000	1100	1110	0010
1100	0111	0101	1000	1101
1101	0110	1001	1011	0010
1110	0101	0000	1111	1111
1111	0100	0111	0010	0101

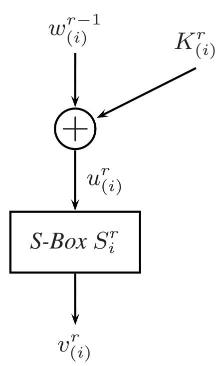
Die Ausgabedifferenz $b' = 0010$ kommt also $D(a', 0010) = 8$ Mal vor, während die Differenzen 0101, 0111, 1101 und 1111 je zwei Mal und die übrigen Werte überhaupt nicht vorkommen (siehe Zeile B in nachfolgender Tabelle). Führen wir diese Berechnungen für jede der $2^4 = 16$ Eingabedifferenzen $a' \in \{0,1\}^4$ aus, so erhalten wir die folgenden Werte für die Häufigkeiten $D(a', b')$ der Ausgabedifferenz b' bei Eingabedifferenz a' (a' und b' sind hexadezimal dargestellt):

	b'															
a'	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
$\vdots$									$\vdots$							
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
$\vdots$									$\vdots$							
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Wie können wir nun die ungleichmäßige Verteilung der Ausgabedifferenzen zur Bestimmung von Schlüsselbits ausnützen? Eine Antwort auf diese Frage geben die folgenden zwei Beobachtungen.

Beobachtung 102 *Für die Eingabe $u^r_{(i)}$ der S-Box S^r_i in Runde r gilt*

$$u^r_{(i)} = w^{r-1}_{(i)} \oplus K^r_{(i)}.$$



Daher hängt die Eingabedifferenz

$$(u^r_{(i)})' = u^r_{(i)} \oplus (u^r_{(i)})^* = (w^{r-1}_{(i)} \oplus K^r_{(i)}) \oplus ((w^{r-1}_{(i)})^* \oplus K^r_{(i)}) = w^{r-1}_{(i)} \oplus (w^{r-1}_{(i)})^*$$

von S^r_i nicht von den Schlüsselbits in Runde r ab, während die Ausgabedifferenz

$$(v^r_{(i)})' = v^r_{(i)} \oplus (v^r_{(i)})^* = \pi_S(w^{r-1}_{(i)} \oplus K^r_{(i)}) \oplus \pi_S((w^{r-1}_{(i)})^* \oplus K^r_{(i)})$$

neben der Eingabedifferenz $(u^r_{(i)})'$ auch noch von $w^{r-1}_{(i)} \oplus K^r_{(i)}$ abhängt.

Können wir nun in einem SPN für bestimmte S-Boxen S_i Differentiale (a'_i, b'_i) finden, so dass die Eingabedifferenz dieser Differentiale mit der (permutierten) Ausgabedifferenz der Differentiale in der jeweils vorhergehenden Runde übereinstimmt (siehe Abbildung 3), so können wir diese Differentiale zu einer so genannten **Differentialspur** (engl. differential trail) zusammen setzen. Unter der Annahme, dass sich die beteiligten S-Boxen S_i (diese werden auch als **aktiv** bezeichnet) unabhängig voneinander entscheiden, dem zugehörigen Differential (a'_i, b'_i) zu folgen oder nicht, berechnet sich der Weitergabequotient der Spur als das Produkt der Weitergabequotienten der beteiligten Differentiale. Obwohl diese Annahme i.a. nicht zutrifft, ist in der praktischen Anwendung nicht mit großen Abweichungen von diesem hypothetischen Wert zu rechnen.

Beispiel 103 *Betrachten wir das SPN aus Beispiel 94, so lassen sich folgende Differentiale zu einer Spur für die Abbildung $x \mapsto u^4$ kombinieren (siehe auch Abbildung 3):*

Für S_2^1: *das Differential* $(1011, 0010) = (B, 2)$ *mit* $Q(B, 2) = 1/2$,
für S_3^2: *das Differential* $(0100, 0110) = (4, 6)$ *mit* $Q(4, 6) = 3/8$ *und*
für S_2^3 und S_3^3: *das Differential* $(0010, 0101) = (2, 5)$ *mit* $Q(2, 5) = 3/8$.

Gemäß dieser Spur führt also eine Eingabedifferenz

$$x' = 0000\,1011\,0000\,0000$$

mit hypothetischer Wk $1/2(3/8)^3 = 27/1024 \approx 0,026$ *auf die Differenz*

$$(v^3)' = 0000\,0101\,0101\,0000,$$

welche wiederum mit Wk 1 auf die Differenz

$$(u^4)' = 0000\,0110\,0000\,0110$$

führt. D.h. wir erhalten für die Abbildung $x \mapsto u^4$ die Spur

$$(a', b') = (0000\,1011\,0000\,0000, 0000\,0110\,0000\,0110)$$

mit einem hypothetischen Weitergabequotienten von $\varepsilon = Q(a', b') = 27/1024$.

Sei nun (a', b') eine Spur für die Abbildung $x \mapsto u^4$ mit einem hypothetischen Weitergabequotienten $\varepsilon = Q(a', b')$. Weiter sei M eine Menge von t Klartext-Kryptotext-Doppelpaaren (x, x^*, y, y^*), die alle mit dem gleichen unbekannten Schlüssel K erzeugt wurden und zusätzlich die Eigenschaft haben, dass die Klartextdifferenz $x' = x \oplus x^* = a'$ ist. Dann wird ca. ein ε-Anteil dieser Doppelpaare der Spur (a', b') folgen und daher bei Verschlüsselung mit K Zwischenergebnisse u^4 und $(u^4)^*$ liefern, die die Differenz

$$(u^4)' = u^4 \oplus (u^4)^* = b'$$

aufweisen. Doppelpaare mit dieser Eigenschaft werden **richtige Doppelpaare** (für die Spur (a', b')) genannt. Ein Großteil der falschen Doppelpaare lässt sich daran erkennen, dass die Kryptotext-Differenzen nicht die erwarteten 0^l-Blöcke aufweisen (im aktuellen Beispiel sind dies die Blöcke $y'_{(1)}$ und $y'_{(3)}$). Es empfiehlt sich, diese Doppelpaare auszufiltern, da sie (wie alle falschen Doppelpaare) nur „Hintergrundrauschen" erzeugen und somit die Bestimmung des Schlüssels eher behindern.

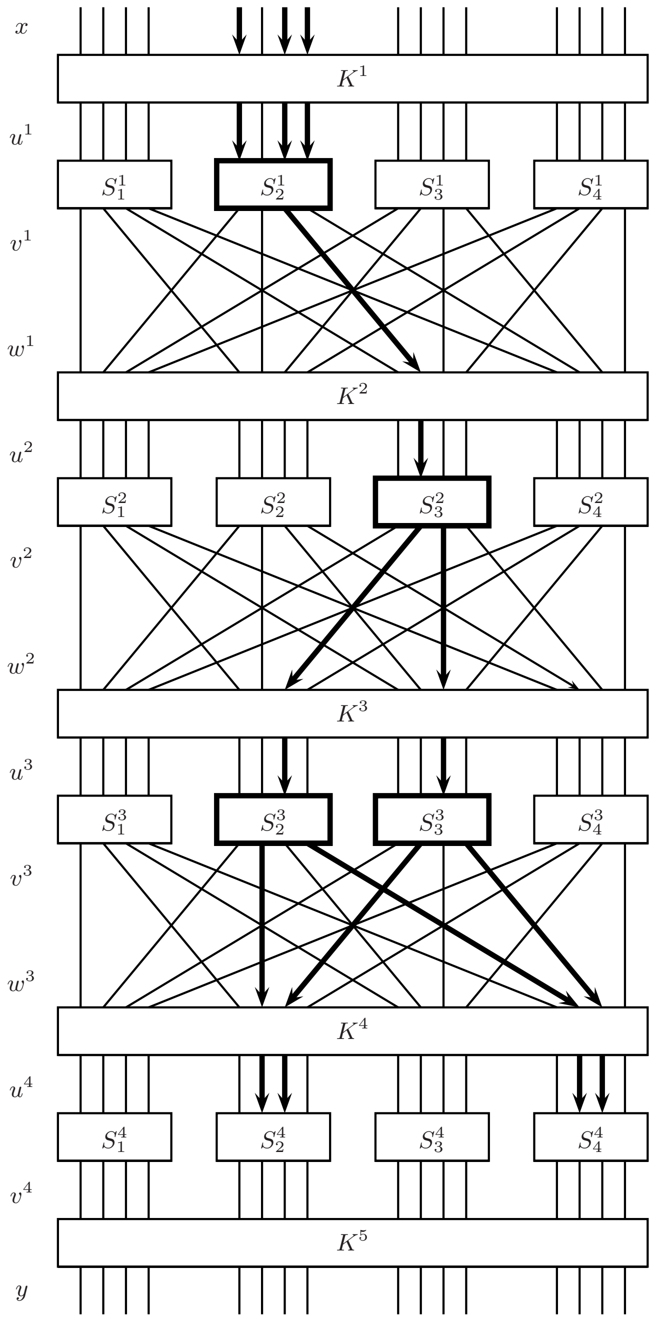


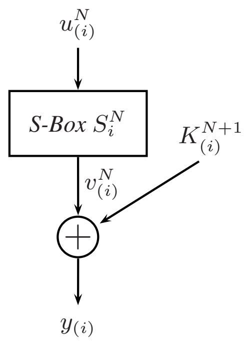
Abbildung 3 Eine Differentialspur für ein Substitutions-Permutations-Netzwerk.

Beobachtung 104 *Für die Ausgabe $v_{(i)}^r$ der S-Box S_i^r in Runde $r = N$ gilt*

$$v_{(i)}^N = y_{(i)} \oplus K_{(i)}^{N+1}$$

und die Eingabe $u_{(i)}^N$ der S-Box S_i^N in Runde N ist

$$u_{(i)}^N = \pi_{S_i^N}^{-1}(v_{(i)}^N) = \pi_{S_i^N}^{-1}(y_{(i)} \oplus K_{(i)}^{N+1})$$



Ist S_i^N nichtlinear, so hängt die aus den Kryptotextblöcken $y_{(i)}$ und $(y_{(i)})^*$ zurückgerechnete Eingabedifferenz

$$(u_{(i)}^N)' = u_{(i)}^N \oplus (u_{(i)}^N)^* = \pi_{S_i^N}^{-1}(y_{(i)} \oplus K_{(i)}^{N+1}) \oplus \pi_{S_i^N}^{-1}((y_{(i)})^* \oplus K_{(i)}^{N+1})$$

von den Schlüsselbits in $K_{(i)}^{N+1}$ ab. Ist also (x, x^*, y, y^*) ein richtiges Doppelpaar, so sind sowohl die Eingabedifferenz $b'_{(i)} = (u_{(i)}^N)'$ von S_i^N als auch die Kryptotextblöcke $y_{(i)}$ und $y_{(i)}^*$ bekannt. Folglich kommen nur solche Subkey-Werte k für $K_{(i)}^{N+1}$ in Frage, für die

$$\pi_{S_i^N}^{-1}(y_{(i)} \oplus k) \oplus \pi_{S_i^N}^{-1}(y_{(i)}^* \oplus k) = b'_{(i)} \tag{5}$$

ist. Erfüllt k Gleichung (5), so sagen wir auch, k ist mit dem Doppelpaar (x, x^*, y, y^*) ***konsistent****.*

Gemäß Beobachtung 104 kann jedes richtige Doppelpaar dazu benutzt werden, die in Frage kommenden Werte für bestimmte Blöcke des Rundenschlüssels K^5 einzuschränken. Ist M hinreichend groß, so wird sich schließlich der richtige Teilschlüssel als derjenige heraus kristallisieren, der mit den meisten Doppelpaaren konsistent ist. Wir benutzen nun die in Beispiel 103 gefundene Spur für einen Angriff mittels differentieller Analyse.

Beispiel 105 *In Algorithmus 106 wird für jeden Subschlüssel-Kandidaten (L_1, L_2) für $(K_{(2)}^5, K_{(4)}^5)$ die Anzahl $\gamma(L_1, L_2)$ aller Doppelpaare (x, x^*, y, y^*) in M bestimmt, die nicht als falsch erkannt werden und mit (L_1, L_2) konsistent sind. Ausgegeben wird der Kandidat (L_1, L_2) mit dem größten γ-Wert.*

Algorithmus 106 *DifferentialAttack*

for $(L_1, L_2) \leftarrow (0, 0)$ ***to*** *(F,F)* ***do***
  $\gamma(L_1, L_2) \leftarrow 0$ ***end***
for each $(x, x^*, y, y^*) \in M$ ***do***
  if $y_{(1)} = y^*_{(1)}$ *und* $y_{(3)} = y^*_{(3)}$ ***then***
    for $(L_1, L_2) \leftarrow (0, 0)$ ***to*** (F, F) ***do***
      $v^4_{(2)} \leftarrow L_1 \oplus y_{(2)}$
      $v^4_{(4)} \leftarrow L_2 \oplus y_{(4)}$
      $u^4_{(2)} \leftarrow \pi_S^{-1}(v^4_{(2)})$
      $u^4_{(4)} \leftarrow \pi_S^{-1}(v^4_{(4)})$
      $(v^4_{(2)})^* \leftarrow L_1 \oplus y^*_{(2)}$
      $(v^4_{(4)})^* \leftarrow L_2 \oplus y^*_{(4)}$
      $(u^4_{(2)})^* \leftarrow \pi_S^{-1}((v^4_{(2)})^*)$
      $(u^4_{(4)})^* \leftarrow \pi_S^{-1}((v^4_{(4)})^*)$
      $(u^4_{(2)})' \leftarrow u^4_{(2)} \oplus (u^4_{(2)})^*$
      $(u^4_{(4)})' \leftarrow u^4_{(4)} \oplus (u^4_{(4)})^*$
      if $(u^4_{(2)})' = 0110$ *und* $(u^4_{(4)})' = 0110$ ***then***
        $\gamma(L_1, L_2) \leftarrow \gamma(L_1, L_2) + 1$
    end
end
$max \leftarrow -1$
for $(L_1, L_2) \leftarrow (0, 0)$ ***to*** (F, F) ***do***
  if $\gamma(L_1, L_2) > max$ ***then***
    $max \leftarrow \gamma(L_1, L_2)$
    $maxkey \leftarrow (L_1, L_2)$
end
Ausgabe: $maxkey$

Im allgemeinen werden für eine erfolgreiche differentielle Attacke circa $t \approx c\varepsilon^{-1}$ Klartext-Kryptotext-Doppelpaare benötigt, wobei ε der Weitergabequotient der benutzten Spur und c eine „kleine“ Konstante ist (im Beispielfall reichen $t \approx 80$ Doppelpaare, wobei $\varepsilon^{-1} \approx 38$ ist).