Übungsblatt 7

Aufgabe 34 ***mündlich***

Sei E die Chiffrierfunktion einer Blockchiffre S mit Blocklänge l und Schlüssellänge k. Wir betrachten einen Angriff bei *bekanntem Klartext*, d. h. es steht eine ausreichende Zahl von Klartext-Kryptotext-Paaren $(x_i, y_i), i = 1, \ldots, t$, zur Verfügung, die alle mit demselben unbekannten Schlüssel K generiert wurden.

(a) Bestimmen Sie heuristisch die erwartete Anzahl von Schlüsseln K, die zu allen Paaren (x_i, y_i) »passen«, d. h. es gilt $E_K(x_i) = y_i$ für $i = 1, \ldots, t$.

Hinweis: Gehen Sie davon aus, dass für einen zufällig gewählten Schlüssel K die Wahrscheinlichkeit $\Pr[E_K(x) = y]$, dass K einen gegebenen Klartext x durch den Kryptotext y chiffriert, gleich 2^{-l} ist (selbst dann, wenn bereits bekannt ist, dass K gewisse Klartexte $x_i \neq x$ durch gewisse Kryptotexte y_i chiffriert).

(b) Wie lässt sich im Fall $t \geq k/l$ der benutzte Schlüssel K mittels $t2^k$ Verschlüsselungen bestimmen?

(c) Um die Sicherheit zu erhöhen wird nun das Kryptosystem $S \times S$ verwendet, d. h. die Schlüssellänge verdoppelt sich auf $2k$. Zeigen Sie, dass sich dadurch die benötigte Anzahl t an Klartext-Kryptotext-Paaren (x_i, y_i) ebenfalls (auf $2k/l$) verdoppelt.

(d) Wie lässt sich im Fall $t \geq 2k/l$ der benutzte Schlüssel (K, K') unter Verwendung eines Speichers der Größe $(lt + k)2^k$ mittels $t2^{k+1}$ Ver- und Entschlüsselungen bestimmen?

(e) Überlegen Sie, wie sich der Platzbedarf in (d) auf Kosten der Rechenzeit reduzieren lässt. Suchen Sie nach einer möglichst allgemeinen Beziehung für diesen so genannten *Time-Memory-Tradeoff*.

Aufgabe 35 ***mündlich***

Sei $\pi_S\colon \{0,1\}^l \to \{0,1\}^{l'}$ eine S-Box und für $(a, b) \in \{0,1\}^l \times \{0,1\}^{l'}$ sei $L(a, b)$ die Anzahl der Paare $(x, y) \in \left\{(x, \pi_S(x)) \,\middle|\, x \in \{0,1\}^l\right\}$, für die $\bigoplus_{i=1}^{l} a_i x_i = \bigoplus_{j=1}^{l'} b_j y_j$ ist. Zeigen Sie:

(a) $L(0^l, 0^{l'}) = 2^l$,

(b) $L(a, 0^{l'}) = 2^{l-1}$ für alle $a \in \{0,1\}^l - \{0^l\}$,

(c) $\sum_{a \in \{0,1\}^l} L(a, b) = 2^{2l-1} \pm 2^{l-1}$ für alle $b \in \{0,1\}^{l'}$,

(d) $\sum_{\substack{a \in \{0,1\}^l \\ b \in \{0,1\}^{l'}}} L(a, b) = \begin{cases} 2^{2l+l'-1} + 2^{l+l'-1} & \pi_S(0^l) = 0^{l'}, \\ 2^{2l+l'-1} & \text{sonst.} \end{cases}$

Aufgabe 36 ***mündlich***

Zeigen Sie, dass eine S-Box genau dann linear ist, wenn für alle $a \in \{0,1\}^l$ und $b \in \{0,1\}^{l'}$ der Bias $\varepsilon(U_a \oplus V_b)$ einen der drei Werte in $\left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}$ annimmt.

Aufgabe 37 ***mündlich***

Wir betrachten ein SPN mit der S-Box S' (aus Aufgabe 32)

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_{S'}(z)$	8	4	2	1	C	6	3	D	A	5	E	7	F	B	9	0

und der Permutation π_P aus der Vorlesung. Überlegen Sie, wie sich durch lineare Approximation von drei S-Boxen $S^r_{i_r}$, $r = 1, 2, 3$, die lineare Approximation $X_{16} \oplus U_1^4 \oplus U_9^4$ für die Abbildung $x \mapsto u^4$ gewinnen läßt, so dass diese (bei Verwendung des Piling-up Lemmas) einen hypothetischen Bias-Absolutwert von $1/16$ hat.

Aufgabe 38 ***10 Punkte***

Schreiben Sie ein Programm, das den in der vorigen Aufgabe skizzierten Angriff auf ein SPN mittels linearer Kryptoanalyse ausführt. Testen Sie Ihr Programm mit zufällig generierten Klartext-Kryptotext-Paaren, um die Anzahl t der zur Bestimmung des korrekten Subkey benötigten Paare herauszufinden.

Aufgabe 39 ***mündlich***

(a) Wir betrachten das SPN aus der Vorlesung, wobei die S-Box $\pi_{S''}$ mit

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_{S''}(z)$	E	2	1	3	D	9	0	6	F	4	5	A	8	C	7	B

benutzt wird. Bestimmen Sie die Werte $D(a, b)$ für $a, b \in \{0,1\}^4$.

(b) Finden Sie geeignete Differentiale für die vier S-Boxen S_1^1, S_4^1, S_4^2 und S_4^3, um eine Differentialspur mit einem Weitergabequotienten von $27/2048$ zu bilden.

Aufgabe 40 ***10 Punkte***

Schreiben Sie ein Programm, das den in der vorigen Aufgabe skizzierten Angriff auf ein SPN mittels differentieller Kryptoanalyse ausführt. Testen Sie Ihr Programm mit zufällig generierten Klartext-Kryptotext-Doppelpaaren, um die Anzahl t der zur Bestimmung des korrekten Subkey benötigten Doppelpaare herauszufinden.