CRYPTO2-Übersicht: VL05+06: CPA gegen Chiffren mit 0-3 S-Boxen

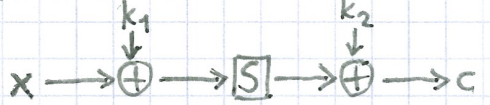
↳ DDT einer S-Box / Differenzielle Kryptoanalyse
↳ r-Runden-Charakteristik/-Differential

VL5 →

- **Chiffre 0:**

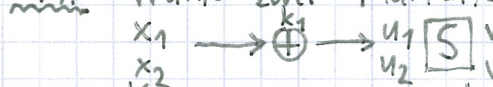
≈ OTP, aber k konstant
CPA: 1 chosen plaintext genügt:
Wähle x, erhalte $c = x + k$. Es gilt: $k = c + x$ „traurig"

- **Chiffre 1:**



wobei bekannt: x (da CPA) und c
allgemein bekannt: S-Box S
(verzichte sogar auf Schlüsselableitung)

CPA: Wähle zwei Klartexte x_1, x_2 und erhalte c_1, c_2:

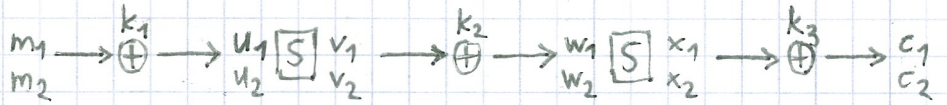


Kenne: $u_1 + u_2 = (x_1 + k_1) + (x_2 + k_1) = x_1 + x_2$
$v_1 + v_2 = (c_1 + k_2) + (c_2 + k_2) = c_1 + c_2$

Rate k_2: Kandidat k_2^* (k1 muss n. mehr geraten werden ⇒ Bits an Sicherh. halbiert!
↳ $v_1^* = c_1 + k_2^* \Rightarrow u_1^* = S^{-1}(v_1^*)$
↳ $v_2^* = c_2 + k_2^* \Rightarrow u_2^* = S^{-1}(v_2^*)$

(falsch in VL 4/5) Ist $u_1^* + u_2^* = x_1 + x_2$? → Nein: k_2^* kein Kandidat, rate erneut!
→ Ja: Berechne k_1^* als $k_1^* = x_1 + u_1^* = x_2 + u_2^*$
↳ Prüfe mit mehreren x_i, ob tatsächlich $(k_1, k_2) = (k_1^*, k_2^*)$

VL6 → • **Chiffre 2:**



CPA: Rate k_3: $k_3^* \rightarrow w_1 = S^{-1}(c_1 + k_3^*)$, $w_2 = S^{-1}(c_2 + k_3^*)$
$v_1 + v_2 = w_1 + w_2$, $u_1 + u_2 = m_1 + m_2$
Wenn $S(u_1 + u_2) = v_1 + v_2$, dann ist k_3^* ein Kandidat.
↳ eigentlich falsch, da S() kein Homomorphismus ist, aber DDT:

Achtung: bitw. Difference = ⊕

DDT: Sei S:

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S(x)	6	4	c	5	0	7	2	e	1	f	3	d	8	a	9	b

(bewusst schlecht gewählt!)

DDT v. S: (← Δ_{out})

Δ_{in} ↓	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	6	0	0	0	0	2	0	2	0	0	2	0	4	0
...																
f	0	0	0	0	2	0	2	0	0	0	0	0	10	0	2	

Δ_{in} = 0001 = 0000+0001 = 1100+1101 = 1110+1111 | Δ_{out} = 6+4 = 2 = 8+a = 2 = 9+b = 2

Bei einer optimalen S-Box wären überall (bis auf ganz oben links) nur 0en und 2en. (vgl. UE14)

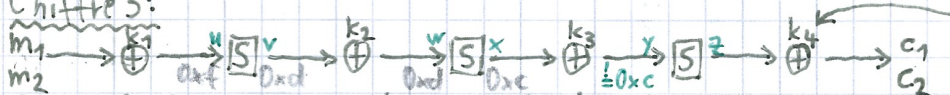
⇒ Wähle m_1, m_2 mit $m_1 + m_2 = u_1 + u_2 = 0xf$. Dann ist in 10/16 Fällen $v_1 + v_2 = w_1 + w_2 = 0xd$.

Schreibe: $\Delta_{in} \xrightarrow{S} \Delta_{out}$

Charakteristik von S: $C(S; \Delta_{in}, \Delta_{out}) = |\{(x,y) \mid x+y = \Delta_{in} \wedge S(x)+S(y) = \Delta_{out}\}| \in 2\mathbb{Z}$

Sage: `sage.crypto.sbox.SBox(6,4,...).difference_distribution_table()`

VL6 → • **Chiffre 3:**



Selbe Strategie: Man rät wieder k_4: k_4^* ⇒ Probl. auf Chiffre 2 zurückgeführt

Zunächst: Zwei S-Boxen lassen sich NICHT zusammenführen, da ein $\oplus k_i$ dazw. ist.
Problem: Man kennt Differenz an Stelle y nicht. ⇒ Also muss auch diese geraten werden!
Wir haben nun 2-Runden-Charakteristik: $0xf \xrightarrow{S} 0xd \xrightarrow{S} 0xe$ (gehe 2 Schritte in DDT)
Es gibt allerdings noch eine zweite 2-Runden-Charakteristik: $0xf \xrightarrow{S} 0x6 \xrightarrow{S} 0xe$
r-Runden-Differential: die Menge aller r-Runden-Charakteristiken mit Eingang Δ_{in} und Ausgang Δ_{out}: $\Delta_{in} \xrightarrow{S} ? \xrightarrow{S} ? \dots \xrightarrow{S} ? \xrightarrow{S} \Delta_{out}$

⇒ Thema heißt: „Differenzielle Kryptoanalyse"
↳ Man versucht, Ausgabedifferenzen über Eingabedif. zu kontrollieren! → gute → d.h. $c_1 + c_2 \in S(0xc) = \{0x1, \dots\}$

⇒ **Wie bestimme ich geeignete Kandidaten k_4^*?:** Wähle mehrere $(m_1, m_2), (m_1', m_2'), \dots$-Paare und generiere für jedes eine Kandidatenliste $K(m_1, m_2)$. Der wirkliche Schlüssel k_4 liegt in allen Kandidatenlisten. $K(m_1, m_2) = \{k^* \mid S^{-1}(c_1 + k^*) + S^{-1}(c_2 + k^*) = \Delta = 0xc\}$.