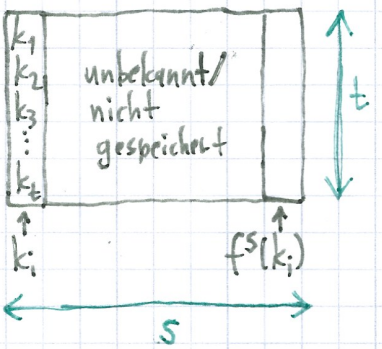
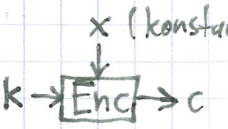
CRYPTO2-Übersicht: VL05: Angriffe gegen sym. Verf.: Brute-Force & TMTO

[1] Brute-Force: Annahme: CPA!
$x \to Enc(k,x)=c$; x,c bekannt; Schlüssel k gesucht
Probiere alle Schlüssel durch ⇒ Finde k_1 mit $Enc(k_1,x)=c$
Teste mit neuem Klartext/Ciphertext-Paar, „ob der auch durchhält": $Enc(k_1,x')\overset{?}{=}c'$
Wenn ja: $k=k_1$; Wenn nein: weitersuchen
Problem: AES: #mögl. Schlüssel $=2^{128}=(2^{10})^{12,8}\approx 10^{3\cdot 12,8}=10^{38,4}$ („ziemlich viel")
($1\,GHz=10^9\,\frac{1}{s}$) Alter des Universums $\approx 10^{17}$ sec ⇒ $\gg 10^{21,4}$ Versuche/Sek. wären nötig... ($=1$ Zettel-Hertz)

[2] Hellman's Time-Memory-Tradeoff (TMTO): →1980 paper
↳ vgl. Babystep; Dyn. Progr.; Rainbow-Tab.
↳ en.wiki: Time/memory/data tradeoff attack: Hellman's attack on block ciphers
Annahme: CPA! sowie dass $n_k=n_b$ (Key-Size = Block-Size), sonst extra Funkt. zum Iter. nötig.
Wähle Klartext x fest. Sei $f: \mathbb{B}^{n_k}\to\mathbb{B}^{n_b}$: $f(k)=Enc(k,x)$.
Wähle Parameter (s,t) geeignet: $s\cdot t\approx 2^{n_k}/t$ ⇒ $s\approx 2^{n_k/3}$, $t\approx 2^{n_k/3}$ ↯ VL ↯
~~sorgt für Wk.: 50% Schlüssel zu finden~~
- Wähle t zufällige Schlüssel: $k_1,\dots,k_t$
- Berechne $f^s(k_i)=f(f(\dots f(k_i)\dots))$ für alle $i\in\{1,\dots,t\}$ und $j\in\{0,\dots,s\}$
  und speichere $(k_i, f^s(k_i))$ für alle $i\in\{1,\dots,t\}$ in einer Tabelle;
  die Zwischenwerte $f^j(k_i)$ für $j\in\{1,\dots,s-1\}$ werden nicht gespeichert:



...wobei jeder Schritt nach rechts so dargestellt werden kann:
$o \xrightarrow{f} o$ oder



Nun sind wir für eine CPA vorbereitet:
- Schicke x an den Challenger und erhalte c.
- Existiert ein i mit $c=f^s(k_i)$? (Binäre Suche in $O(\log t)$ oder Hashtable in $O(1)$)
  Dann ist $y=f^{s-1}(k_i)$ ein guter Schlüsselkandidat, denn: $f(y)=Enc(y,x)=c$.
  Berechne nun y als $y=f^{s-1}(k_i)$.
- Für alle $r\in\{1,\dots,s-1\}$:
  Existiert ein i mit $f^r(c)=f^s(k_i)$? (Binäre Suche in $O(\log t)$ oder Hashtable in $O(1)$)
  Dann ist $c=f^{s-r}(k_i)$
  und $y=f^{s-r-1}(k_i)$ ein guter Schlüsselkandidat, denn: $f(y)=Enc(y,x)=c$.
  Berechne nun y als $y=f^{s-r-1}(k_i)$.

▽ Der Erfolg des Verfahrens hängt von s und t ab ▽

▽ Einordnung:

(Hellman 1980): • Exhaustive Search mit s Operationen hat $P(Success)=s/2^{n_k}$
• Table Lookup mit t Wörtern hat $P(Success)=t/2^{n_k}$
• Sein TMTO mit s Operationen & t Wörtern hat $P(Success)=(st)/2^{n_k}$

▽Achtung: im Original-Paper sind die Buchstaben anders▽

(crypto.stackexchange.com/questions/15233):

- "[...] precomputed tables don't actually speed up the search time, if you count the time taken to generate the precomputed table."
- "[...] if the key/ciphertext pair never occurred during the precomputation phase, you won't find it in the table; [...]"