

Time-Memory Trade-Offs: False Alarms Detection Using Checkpoints

Gildas Avoine ¹ Pascal Junod ² Philippe Oechslin ³

¹EPFL, Switzerland

²Kudelski Group, Switzerland

³Objectif Sécurité, Switzerland

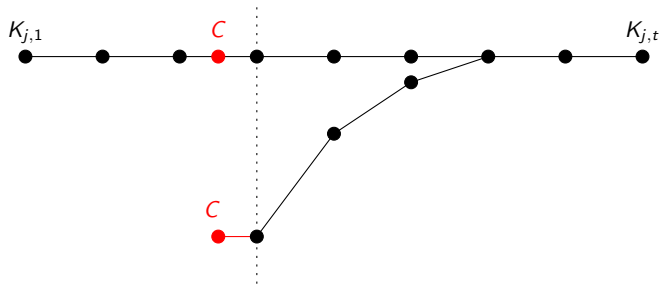
chosen plaintext attack against a block cipher.

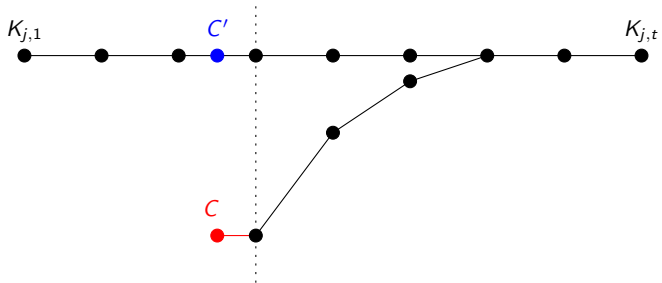
$$\begin{array}{ccccccc}
 \boxed{K_{1,1}} & \xrightarrow{F} & K_{1,2} & \xrightarrow{F} & k_{1,3} & \dots & \xrightarrow{F} & \boxed{k_{1,t}} \\
 \boxed{K_{2,1}} & \xrightarrow{F} & K_{2,2} & \xrightarrow{F} & k_{2,3} & \dots & \xrightarrow{F} & \boxed{k_{2,t}} \\
 \vdots & & & & & & & \vdots \\
 \boxed{K_{m,1}} & \xrightarrow{F} & K_{m,2} & \xrightarrow{F} & k_{m,3} & \dots & \xrightarrow{F} & \boxed{k_{m,t}}
 \end{array}$$

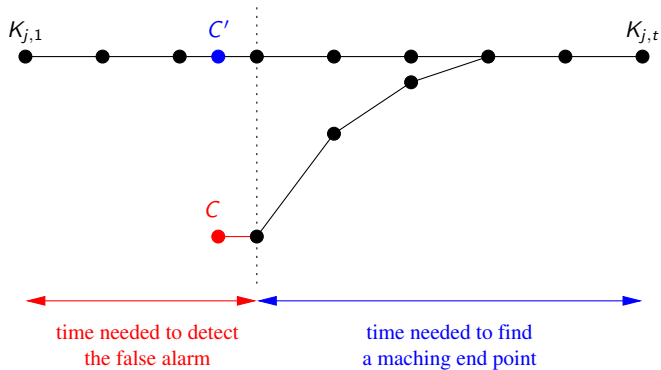
$$F(K) := R(S_K(P))$$

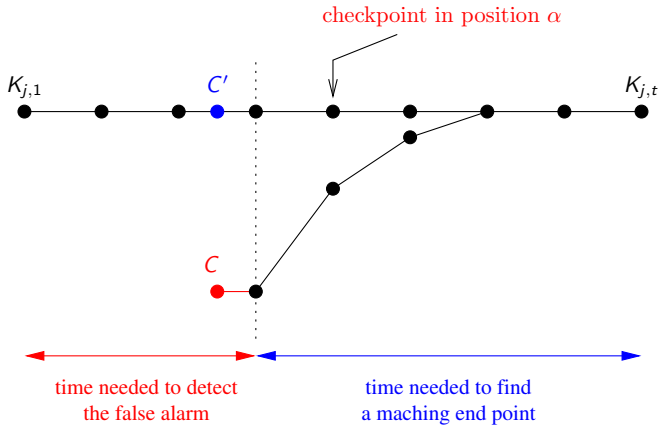
$S_K(*)$ denotes the enciphering operation under key K : $C = S_K(P)$

$R(*)$ is a **reduction** function: $K = R(C)$

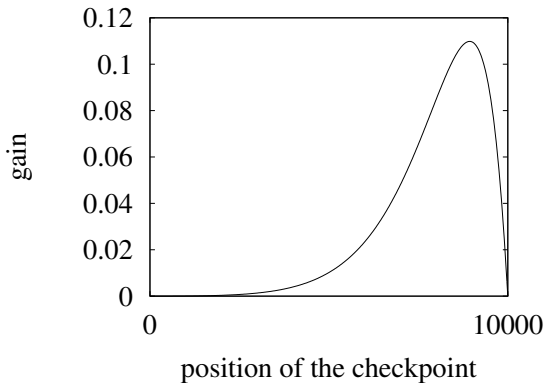








Windows Password Cracking using Rainbow Tables



($t = 10000$, $N = 80.6 \cdot 10^9$, $m = 15408000$, $\ell = 4$)

Gain with 3 checkpoints is about 20%.

- Adding checkpoints is interesting as long as it is more efficient to store checkpoints than using this memory for the trade-off itself.
- Easy to add checkpoints in an existing table (not necessary to re-generate the table)