

Verifier  
(Reader)

$K$

Generate  $R_V$  randomly

$R_V$

$R_P$

Prover  
(Card)

$K$

Generate  $R_P$  randomly

Compute  $(1 + q) \cdot 2N$  session keys

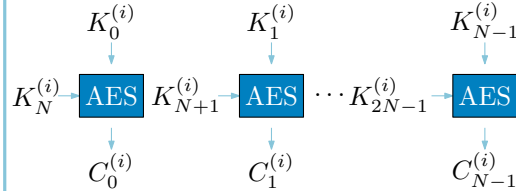
$K$

$R_V || R_P \rightarrow \text{KDF} \rightarrow (K_0^{(0)}, K_1^{(0)}, \dots, K_{2N-1}^{(0)})$

$\forall i \in [1, q] : (K_0^{(i)}, K_1^{(i)}, \dots, K_{2N-1}^{(i)}) \leftarrow \text{random keys}$

For  $i \in [0, q]$ :

- Execute  $N$  AES and store intermediate results



- Compute the “likelihood”  $\mathcal{L}_i$  between each series of  $N$  AES and the leakage acquired during the Prover computation.

Let  $j = \text{argmax} \{\mathcal{L}_i\}$

If  $(j \neq 0 \text{ or } \mathcal{L}_j \text{ does not stand out from } \{\mathcal{L}_{i \neq j}\})$

Then reject the authentication

Else choose  $k$  randomly in  $[0, N - 1]$

$C_k^{(0)}$

If  $\exists l \in [0, N - 1] : l \neq k \text{ and } C_l^{(0)} = C_m$

Then authenticate the Prover

Else reject the authentication

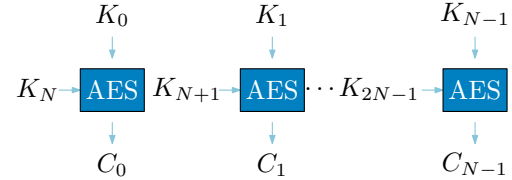
$C_m$

Compute  $2N$  session keys

$K$

$R_V || R_P \rightarrow \text{KDF} \rightarrow (K_0, K_1, \dots, K_{2N-1})$

Execute  $N$  *leaky* AES computations



If  $\exists l \in [0, N - 1] : C_l = C_k^{(0)}$

Then accept it and send back  $C_m$  with  $m \neq l$

Else reject the authentication