r is an integer mod 2^n and addition is also mod 2^n.

Thm. If F a PRF then CTR mode is CPA secure for VIL.

Proof. We start with original CPA game.

$\mathcal{A}$

$m = (m[1], \dots, m[t]) \longrightarrow$ Cpa

$\longleftarrow c = (r, c[1], \dots, c[t])$

$m_0^*, m_1^* \longrightarrow$

$\longleftarrow c^* = (r^*, c^*[1] \dots c^*[t^*])$

$m \longrightarrow$

$\longleftarrow c$

$H_0(\lambda, b)$ $\quad$ $H_1(\lambda, b)$

R $\quad$ $H_2(\lambda, b)$

$k \leftarrow U_\lambda$

$r \leftarrow U_n$ $\quad \forall i \in [t]$

$c[i] = F_k(r+i-1) \oplus m[i]$

$r^* \leftarrow U_n$; $\forall i \in [t^*]$

$c^*[i] = F_k(r^* + i - 1) \oplus m_b^*[i]$

$\xrightarrow{b'}$

$H_1(\lambda, b)$: The same as $H_0(\lambda, b)$ but use R instead of F_K.

$H_2(\lambda, b)$: The same as $H_1(\lambda, b)$ but c^* is UNIFORM.

LEMMA $\forall b,\ H_0(\lambda, b) \approx_c H_1(\lambda, b)$

Proof. Reduction to PRF security.

$\mathcal{A}$ $\qquad$ $\mathcal{B}$ $\qquad$ $\mathcal{C}_{PRF}$

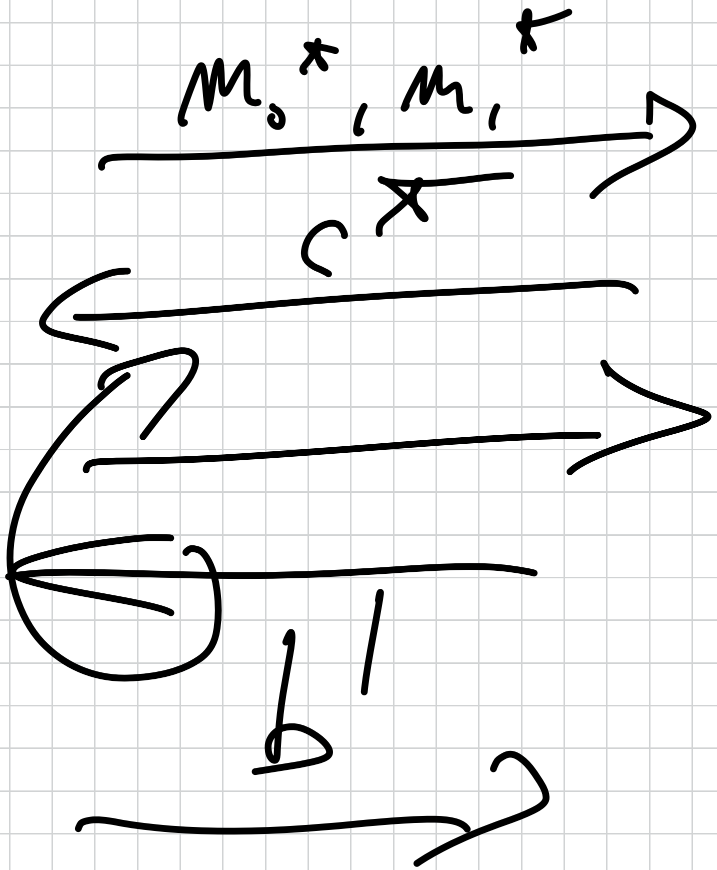
$\mathcal{A} \xleftarrow{\text{START}} \mathcal{B}$

$\mathcal{A} \xrightarrow{m} \mathcal{B}$

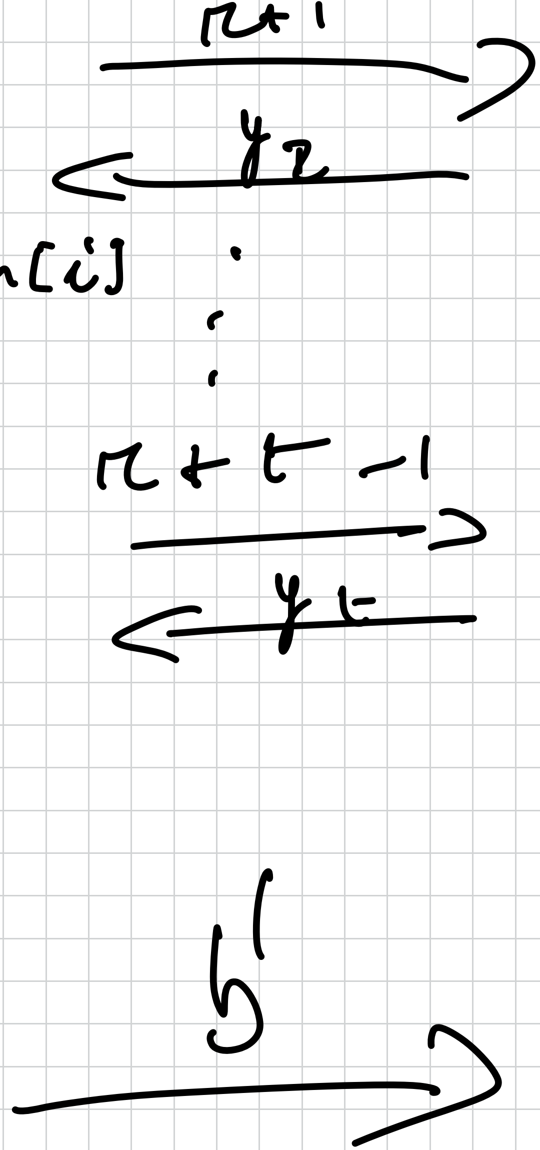
$r \leftarrow U_n$ $\quad \xrightarrow{r} \mathcal{C}_{PRF}$

$\xleftarrow{y'}$

$\mathcal{A} \xleftarrow{c} \mathcal{B}$



$$c[i] = y_i \oplus m[i]$$



$\blacksquare$

LEMMA $\forall b, \ H_1(\lambda, b) \approx_s H_2(\lambda, b)$ as long as A makes $q(\lambda) = poly(\lambda)$ encryption queries.

Proof. Find event E, s.t. when E does not happen $H_1(\lambda, b) \equiv H_2(\lambda, b)$.

The challenge CTX c^* is computed using the sequence:

$$R(r^*), R(r^*+1), \ldots, R(r^*+t^*-1).$$

On the other hand, the other CTXs are computed using the sequence:

$$R(r_i), R(r_i+1), \ldots, R(r_i+t_i-1)$$

Different r_i, t_i for each query!

The event E is the event that the first sequence overlaps with the second sequence (for all encryption queries).

$\bar{E}$: $\exists$ $j, j' \geq 0$; $i \geq 1$

$$r_i + j = r^* + j'$$

$r^* = 2$; $r = 4$; $j' = 2$, $j = 0$

Observe: Conditioning on $\bar{E}$, then c^* will be uniform and $H_1(\lambda, b) = H_2(\lambda, b)$

We only need to bound $\Pr[E]$.

Simplify: Let $q(\lambda)$ be also the max length of any encryption query. Of course $q(\lambda) = \text{poly}$.

$\Rightarrow t_i, t^* = q(\lambda) = \#$ queries.

Consider event E_i: $r_i, \dots, r_i + q - 1$ overlaps with $r^*, \dots, r^* + q - 1$.

$$\Pr[E] \le \sum_{i=1}^{q} \Pr[E_i] \le q(\lambda) \cdot \text{negl}(\lambda)$$
$$= \text{negl}(\lambda).$$

$$r^*, r^*+1, \dots, r^*+q-1$$

$$r_i, r_i+1, \dots, r_i+q-1$$

$$r^* - q + 1 \le r_i \le r^* + q - 1$$

$$\Rightarrow \Pr[E_i] \le \frac{(r^*+q-1) - (r^*-q+1) + 1}{2^n}$$

$$= \frac{2q-1}{2^n} = \mathrm{negl}(\lambda) \quad \blacksquare$$

LEMMA $H_2(\lambda, 0) \equiv H_2(\lambda, 1)$

(Because c^* independent of b in H_2.)

$$\Rightarrow H_0(\lambda, 0) \approx_c H_1(\lambda, 0) \approx_s H_2(\lambda, 0)$$

$$\equiv H_2(\lambda, 1)$$

$$\approx_s H_1(\lambda, 1)$$

$$\approx_c H_0(\lambda, 1) \quad \blacksquare$$

DOMAIN EXTENSION FOR MACs

Recall: PRF $\Rightarrow$ FIL UFCMA MAC.

$$\text{Tag}(k, m) = F_k(m)$$

Some rules that do NOT work:

- $\tau = \text{Tag}_k(\bigoplus_i m_i)$

$$m = (m_1, m_2, \dots)$$

UFCMA (i.e. $AES_k(\cdot)$).

$$(m_1, m_2) = m \Rightarrow \tau$$

$$(m^* = m_1 \oplus m_2, \tau) \qquad \tau = F_k(m_1 \oplus m_2) \quad \checkmark$$

$m = (m_1, m_2)$, let $\tau = F_K(m_1 \oplus m_2)$
$m_1 \neq m_2$

$m^* = (m_2, m_1)$; $\tau^* = \tau$.

$\mathcal{A}$ $\qquad\qquad\qquad\qquad$ $\mathcal{C}$

$$\xrightarrow{(m_1,\, m_2)}$$

$$\xleftarrow{\tau}$$

$$\xrightarrow{m^*,\ \tau^*}$$

– $\tau_i = \mathrm{Tag}_K(m_i)$ $\qquad$ $\tau = (\tau_1, \dots, \tau_d)$
$m = (m_1, \dots, m_d)$

Permute again!

$$- \quad \tau_i = \mathrm{Tag}_K(i \,\|\, m_i)$$

$$\tau = (\tau_1, \ldots, \tau_d)$$

$$m = (m_1, \ldots, m_d)$$

$$m = (m_1, m_2) \quad ; \quad m' = (m_1', m_2')$$

$$\tau = (\tau_1, \tau_2)$$

$$\tau_1 = F_K(1 \,\|\, m_1)$$

$$m^* = (m_1, m_2')$$

$$\tau^* = (\tau_1, \tau_2')$$

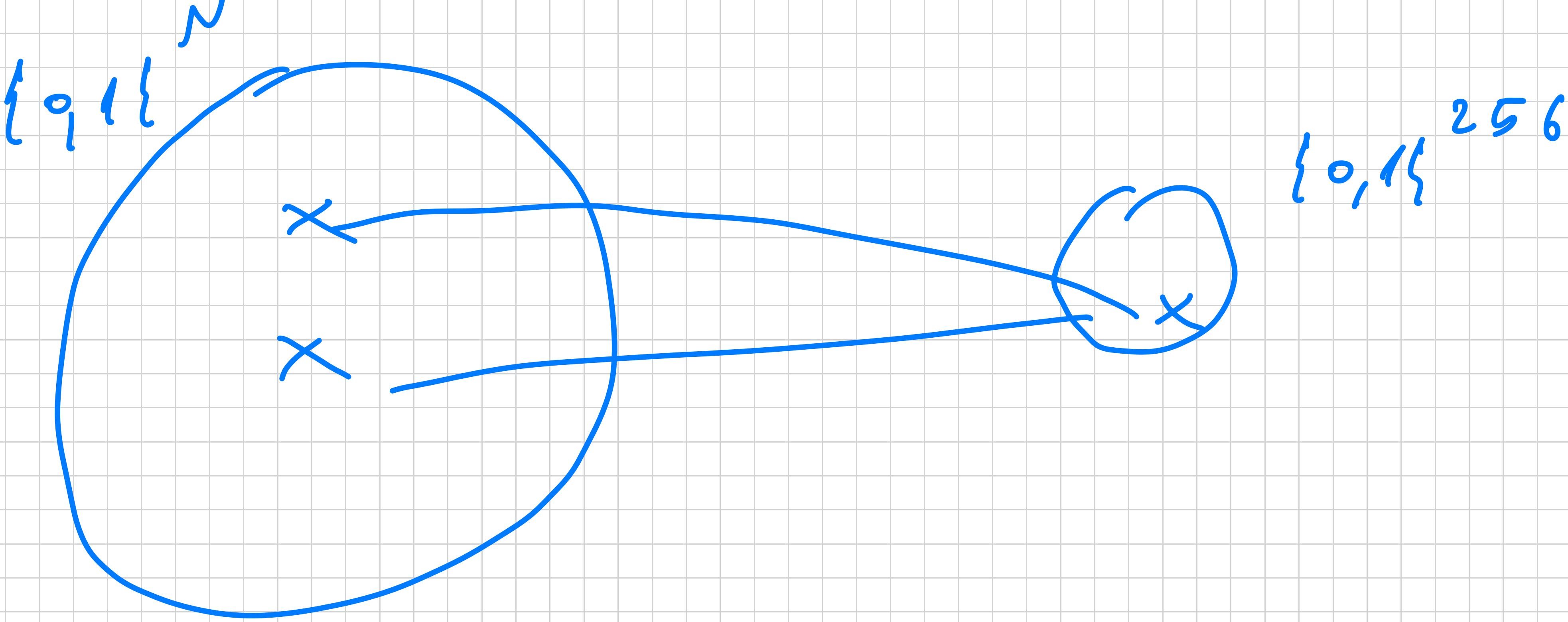
Idea: Design input-shrinking function

$$h : \{0,1\}^N \to \{0,1\}^n$$

$N = n \cdot d$ (d blocks of length n)

Then, output $\tau = F_k(h(m))$

The question: What security from h?



Problem: If we can find COLLISIONS, $h(m) = h(m')$ but $m \neq m'$ we can forge (m', τ) given (m, τ)

Two approaches:

-) let h be SECRET.

→ Let h be PUBLIC. (COLLISION-RES. HASH, SHA)

What does it mean?

$$\mathcal{H} = \{ h_s : \{0,1\}^n \to \{0,1\}^m \}_{s \in \{0,1\}^\lambda}$$

and s is either SECRET or PUBLIC.