Quick Recap: OWFs => PRGs => PRFs.

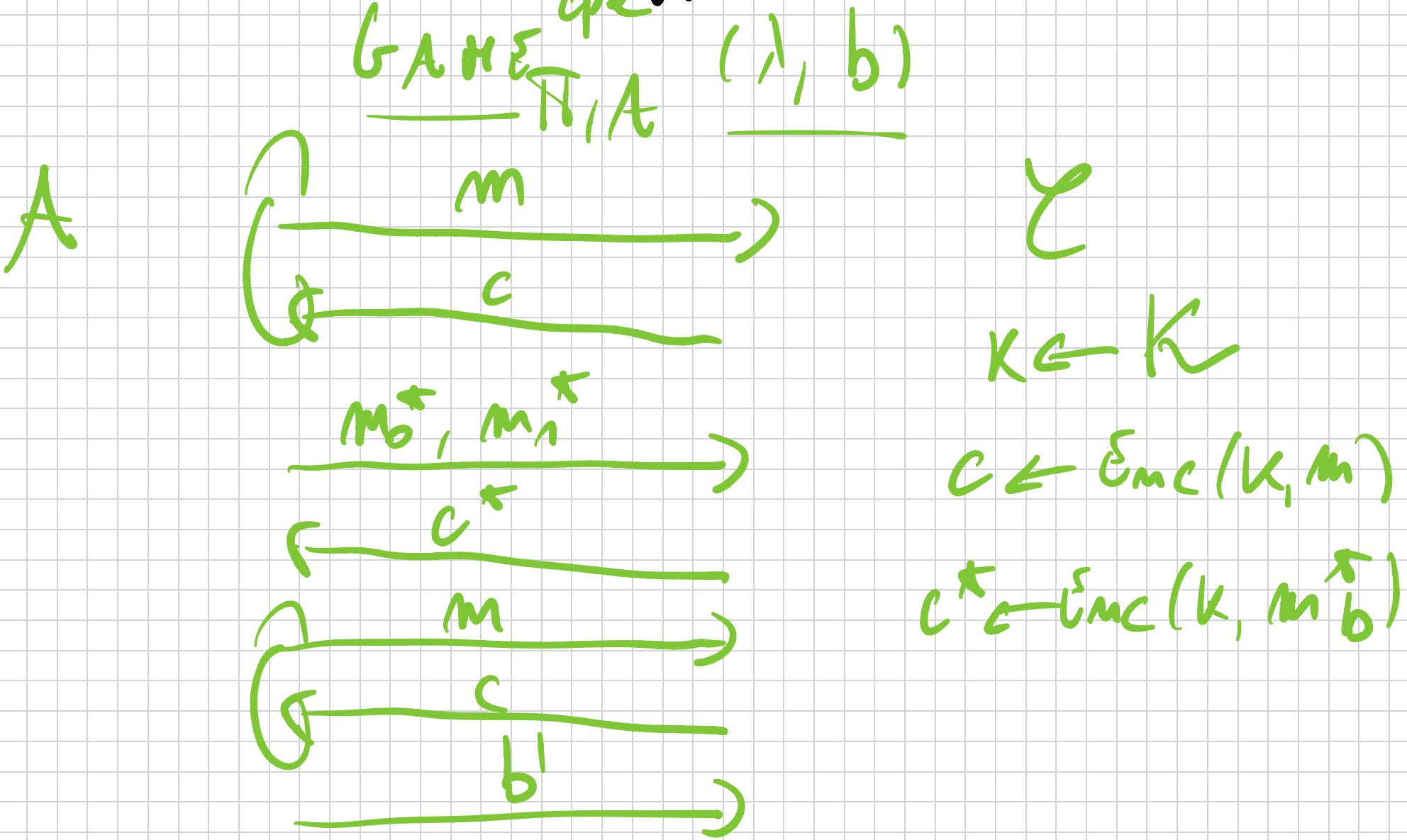
In the next few lectures we'll see that PRFs are enough to do practical symmetric crypto:

- CPA-secure SKE for msg of variable input length. (VIL) ✓
- Secure MACs for msg of VIL
- Non-malleable SKE (e.g. CCA-secure SKE), which is basically

equivalent to combining encryption and msg authentication.

*) As we'll see for some applications it will be important that F is a PRP (Pseudorandom Permutation) namely it is invertible given the key. In practice, we call it a BLOCKCIPHER. We will show: PRFs ⇒ PRPs. This will also explain real-world designs

of some blockciphers (DES, AES).
Let's start with encryption (CPA).



VIL: $|m|$ is any poly (λ)

$|m_0^*| = |m_1^*|$.

We will need a MODE OF OPERATION: A standardized way to encrypt a msg $m = (m_1, \ldots, m_d)$ where $d \in \mathbb{N}$ and $m_i \in \{0,1\}^n$. (e.g. $n = 256$) using a PRF $F = \{F_k : \{0,1\}^n \to \{0,1\}^n\}$

Remark: We can't just output

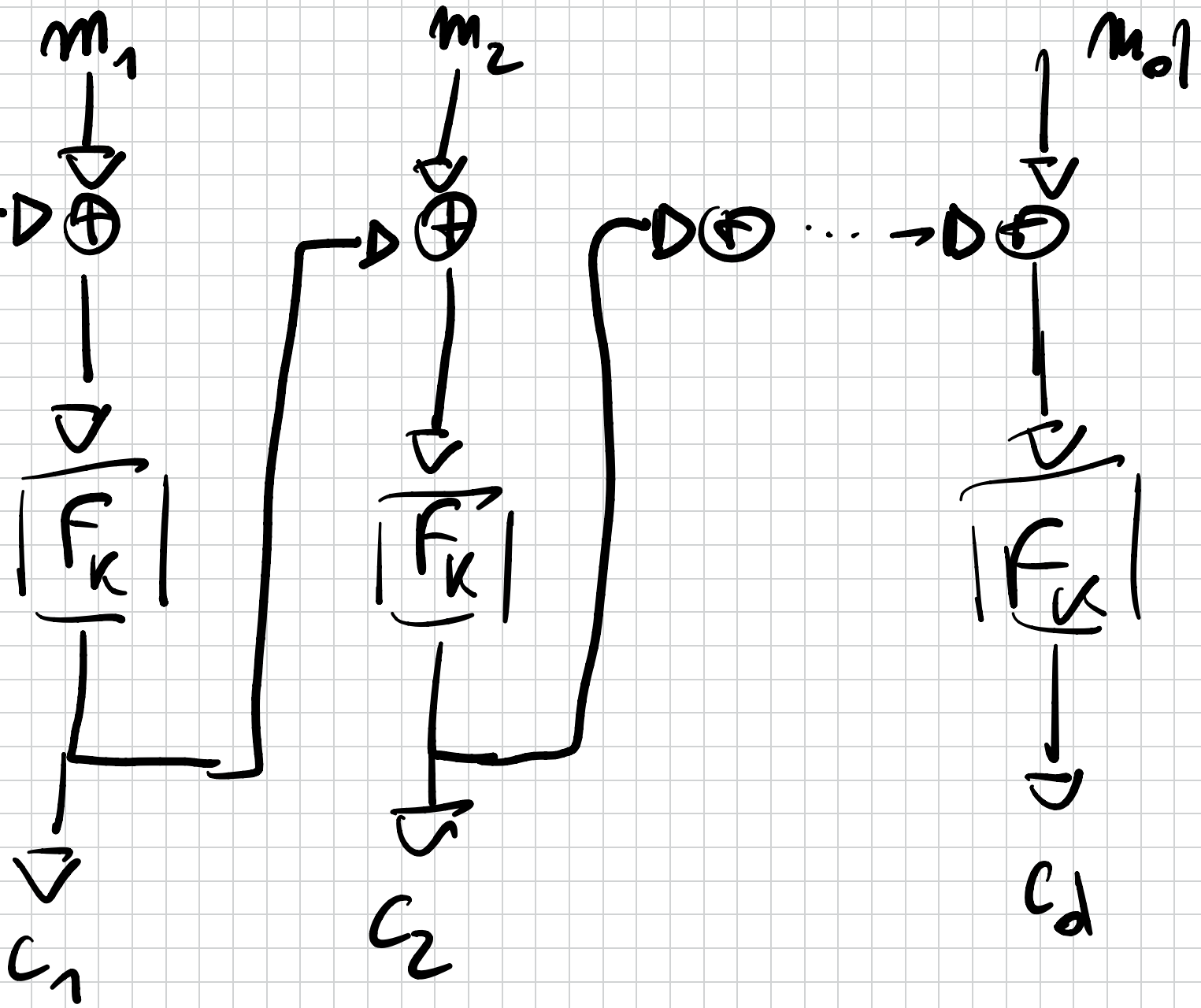
$C = (F_k(m_1), \ldots, F_k(m_d))$ ECB

even assuming F is a PRP.

CBC

$c_0 = IV \leftarrow U_n$

$c_i = F_k(c_{i-1} \oplus m_i)$



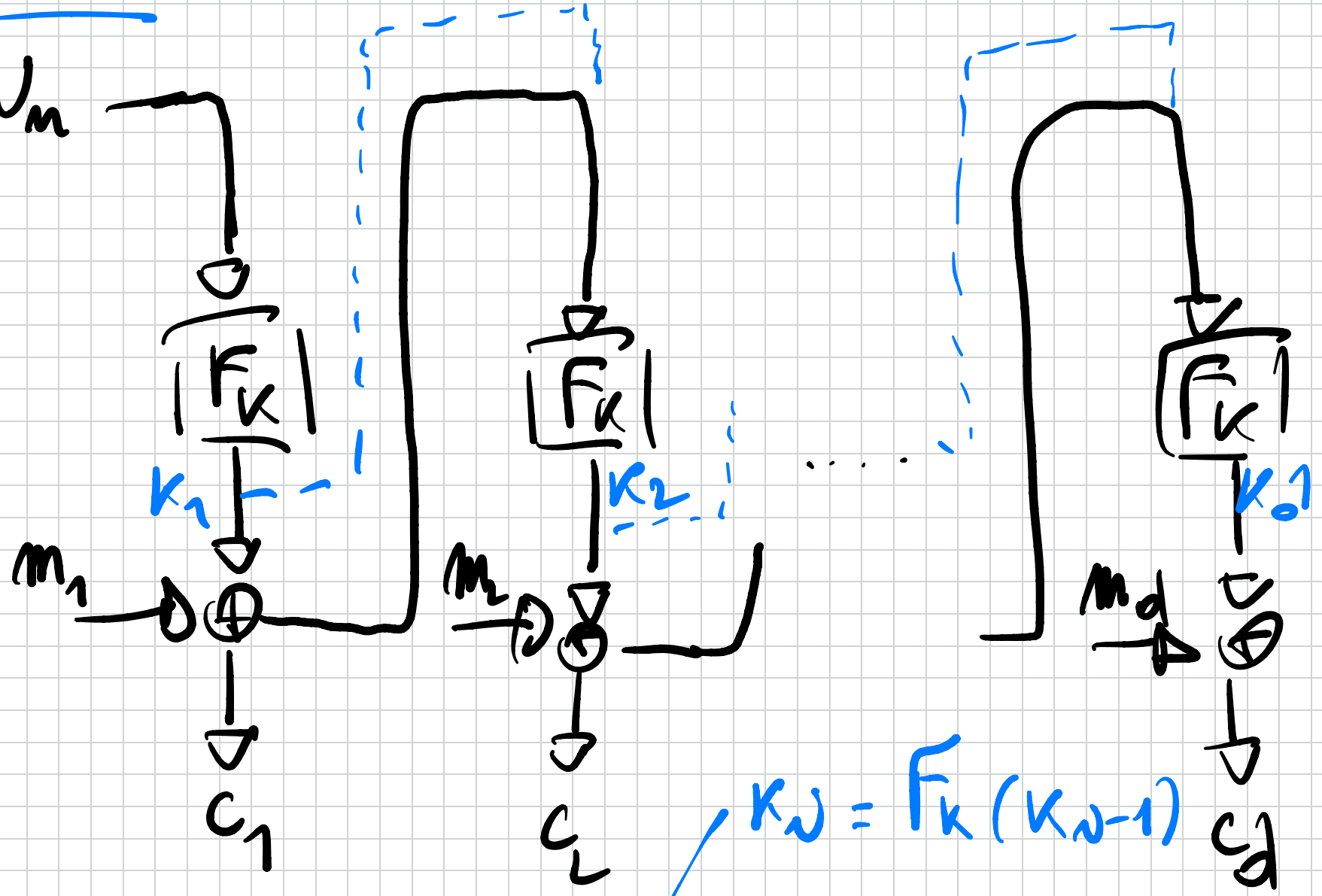
$C = (c_0, c_1, \ldots, c_d)$

Some notes:

- Requires to invert F_K (so needs PRP).
- It's sequential.
- It's secure: If F is a PRP, then CBC-MODE is CPA-secure for VIL.

CFB / OFB

$c_0 = r \leftarrow U_n$

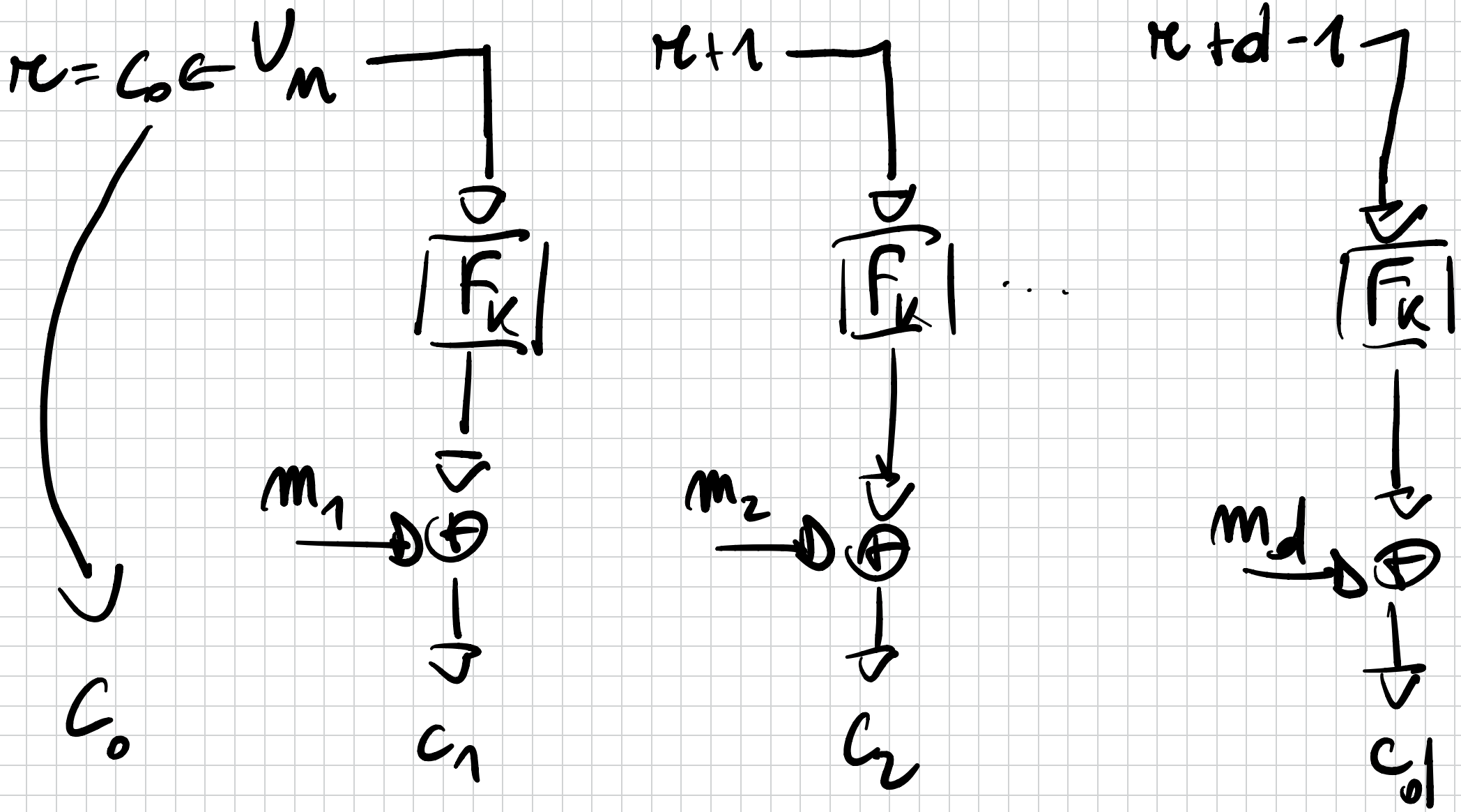


$c_i = F_k(c_{i-1}) \oplus m_i$

$k_i = F_k(k_{i-1})$

$c_i = k_i \oplus m_i$

CTR Let "+" be addition mod 2^n



THM Assuming F is a PRF, CTR mode is a CPA secure SKE for VIL.

Proof. Let $G(\lambda, b) \equiv \text{GAME}^{cpa}_{\Pi, A}(\lambda, b)$. where Π is CTR mode using F.

We need to show $G(\lambda, 0) \stackrel{c}{\approx} G(\lambda, 1)$.

Recall that in $G(\lambda, 0)$:

- 1) Upon input on encryption query $m = (m_1, \ldots, m_d)$, we return

$C = (C_0, C_1, \ldots, C_{d'})$ s.t.

$C_0 = r \xleftarrow{} U_M$; $C_i = F_K(r + i - 1) \oplus m_i$

→) For the challenge $m_b^* = (m_{b,1}^*, \ldots, m_{b,d^*}^*)$

($d^* \in \mathbb{N}$ is the dimension)

$C^* = (C_0^*, C_1^*, \ldots, C_{d^*}^*)$

$C_0^* = r^* \xleftarrow{} U_M$; $C_j^* = F_K(r^* + j - 1) \oplus m_{b,j}^*$

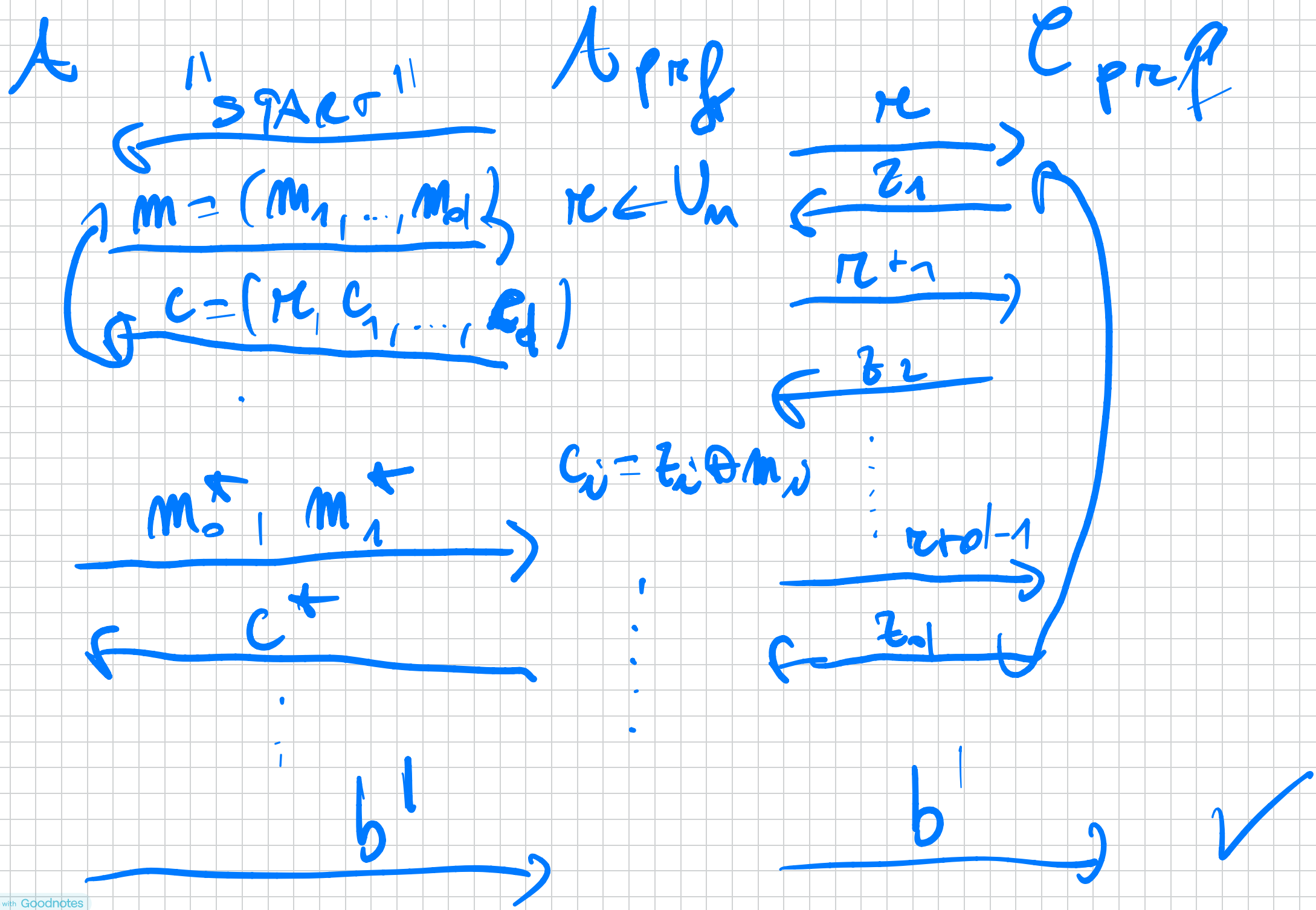
Let $H_1(\lambda, b)$ same as $G(\lambda, b)$ but replace $F_K(\cdot)$ with $R(\cdot)$ s.t.

$R \xleftarrow{} R(\lambda, M \to U)$

$H_2(\lambda)$: The challenge $CT \times C^*$ is UNIFORM and w-independent of b.

LEMMA $\forall b \in \{0,1\}$, $H_1(\lambda, b) \approx_c G(\lambda, b)$.

Proof. A simple exercise. Fix b. Assume $\exists$ PPT A that tells apart $H_1(\lambda, b)$ $G(\lambda, b)$ w.p. $\geq 1/p(\lambda)$ for some poly $p(\lambda)$. We build PPT A_{PRF} against f.


A
"SPACE"
A prf
C prf
m = (m1, ..., md)
r ← Um
r
z1
r+1
c = (r, c1, ..., cd)
z2
ci = zi ⊕ mi
m0, m1
c
zd
b'
b'

LEMMA $H_1(\lambda, b) \approx_s H_2(\lambda)$, $\forall b \in \{0,1\}$

as long as # encryption queries $= q(\lambda) = poly(\lambda)$.

This implies the theorem as then:

$$G(\lambda, 0) \approx_c H_1(\lambda, 0) \approx_c H_2(\lambda)$$

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

$$\approx_c G(\lambda, 1)$$

Proof (of LSKRA). Let's look at the sequence of values that are xored to create the challenge in $H_1(\lambda, b)$:

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

In contrast, during encryption query $i \in [q]$ we generate the sequence:

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

We define a BAD event: The event becomes true if $\exists\, i, j, j' \geq 1$ s.t.

$$r_{i+j} = r^*_{j'}$$

$$(j = 0, \ldots, \omega(N) - 1 \; ; \; j' = 0, \ldots, d^* - 1)$$

Now, conditioning on $\overline{BAD}$ then the sequence $r^*, r^*+1, \ldots, r^*+d^*-1$ never overlaps, which means the challenge ciphertext is distributed like:

$$m^*_{b,1} \oplus u_1, \ldots, m^*_{b,d^*} \oplus u_{d^*}$$

$$u_1, \ldots, u_{d^*} \leftarrow U_M$$

which is identical to U_{real} as in $H_2(\lambda)$. A consequence of this is:

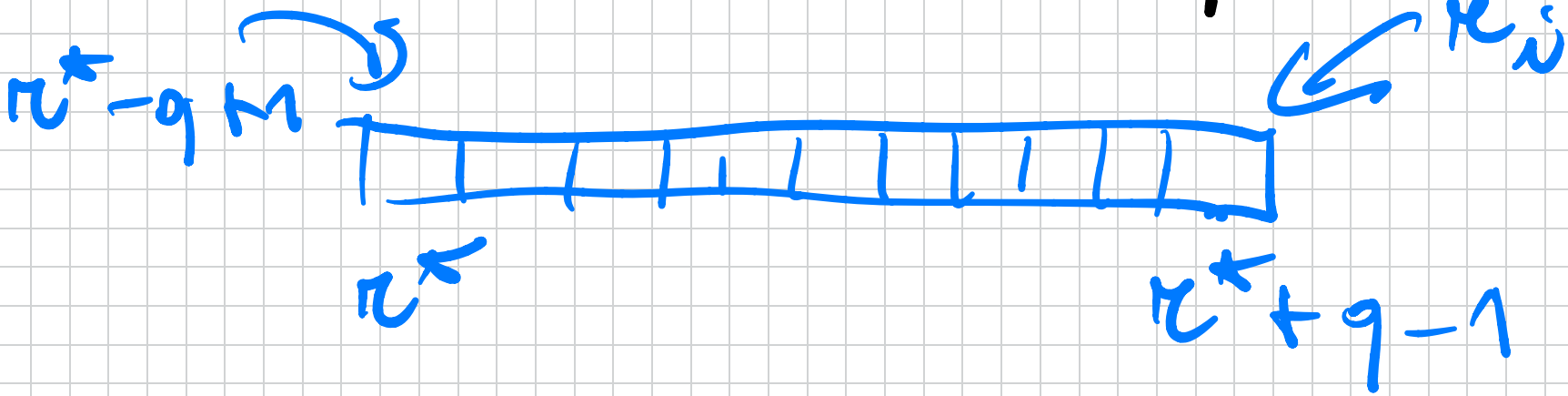
$$SD(H_1(\lambda, b); H_2(\lambda)) \leq Pr[BAD]$$

It remains to compute $Pr[BAD]$.

Fix i, and let BAD_i be the event that the r^* sequence overlaps with the r_i sequence. Why assume that $d_i = d^* = q = poly(\lambda)$. The event BAD_i is the event that:

$r^*, r^*+1, \ldots, r^*+q-1$ overlaps

$r_i, r_{i+1}, \ldots, r_i+q-1.$



$$\Rightarrow r^*-q+1 \le r_{i+1} \le r^*+q-1$$

$$\Rightarrow \Pr[\text{BAD}_i] \le \frac{1+(r^*+q-1)-(r^*-q+1)}{2^m}$$

$$= \frac{2q-1}{2^n}$$

Finally, by the union bound:

$$\Pr[\text{BAD}] = \Pr[\exists i \in [q] : \text{BAD}_i]$$

$$\leq \sum_{i=1}^{q} \Pr[\text{BAD}_i]$$

$$\leq q \cdot \frac{2q-1}{2^n} \leq \frac{2q^2}{2^n}$$

$$= \text{negl}(\lambda)$$ □

We now switch to the problem of message authentication:

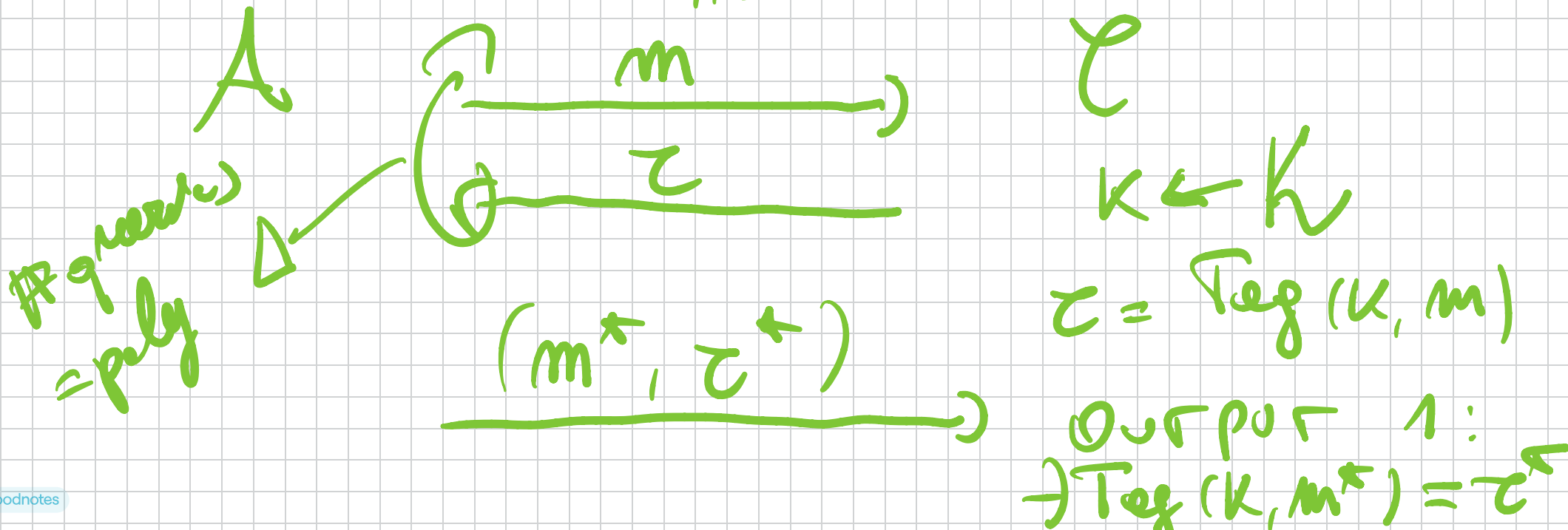
$$\text{Tag} : K \times \mathcal{M} \to \mathcal{T}$$

What's the security of Tag in the computational setting? We'll require that it is hard to forge m^*, τ^* s.t. $\text{Tag}(K, m^*) = \tau^*$ without knowing K.

DEF We say $\Pi = (\mathsf{Tag}, \mathsf{Vf})$ is UFCMA (UNFORGEABILITY UNDER CHOSEN-MESSAGE ATTACKS) if $\forall$ PPT $\mathcal{A}$:

$$\Pr\left[\mathsf{GAME}^{\mathrm{ufcma}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] \leq \mathrm{negl}(\lambda)$$

$\mathsf{GAME}^{\mathrm{ufcma}}_{\Pi,\mathcal{A}}(\lambda)$



-) $m^* \neq 4m$?

THM If $\mathcal{F} = \{f_k\}$ is a PRF, then

$\text{Tag}(k, m) = f_k(m)$ is UF CMA

(for $\mathcal{F}(L)$).