*) Let Tag_1, Tag_2 be MACs. We know that at least one of them is UF-CMA, but not which one.
Show how to construct Tag that is UF-CMA using both Tag_1, Tag_2.

Suggestion: $Tag(K, m) = Tag_1(K_1, m) \| Tag_2(K_2, m)$

A $\quad$ A_1 $\quad$ C_1 $\quad$ K_1

$A \xrightarrow{m} A_1 \xrightarrow{m} C_1$

$\tau_1 = Tag_1(K_1, m)$

$\tau = \tau_1 \| \tau_2$

$K_2 \leftarrow V_\lambda$

$\tau_2 = Tag_2(K_2, m)$

$\tau^* = \tau_1^* \| \tau_2^*$

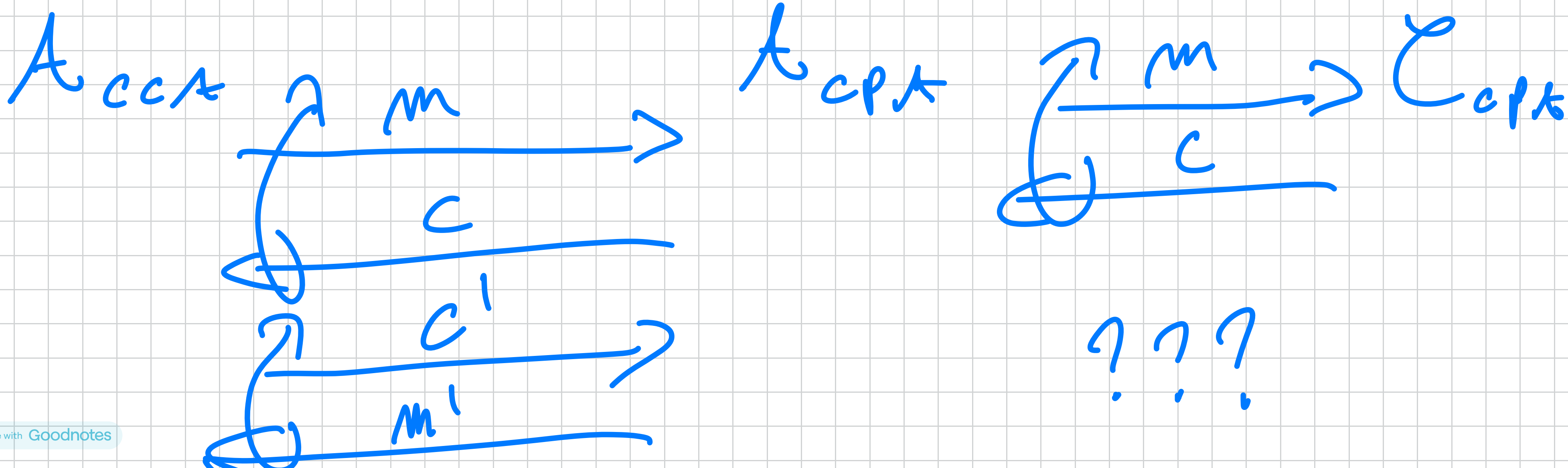
τ_1^* ✓

CCA SECURITY (cont'd).

For the proof of approach 3), we need a lemma:

LEMMA Assuming $\Pi = (Enc, Dec)$ satisfies both CPA and AUTH, then Π is CCA-sec.

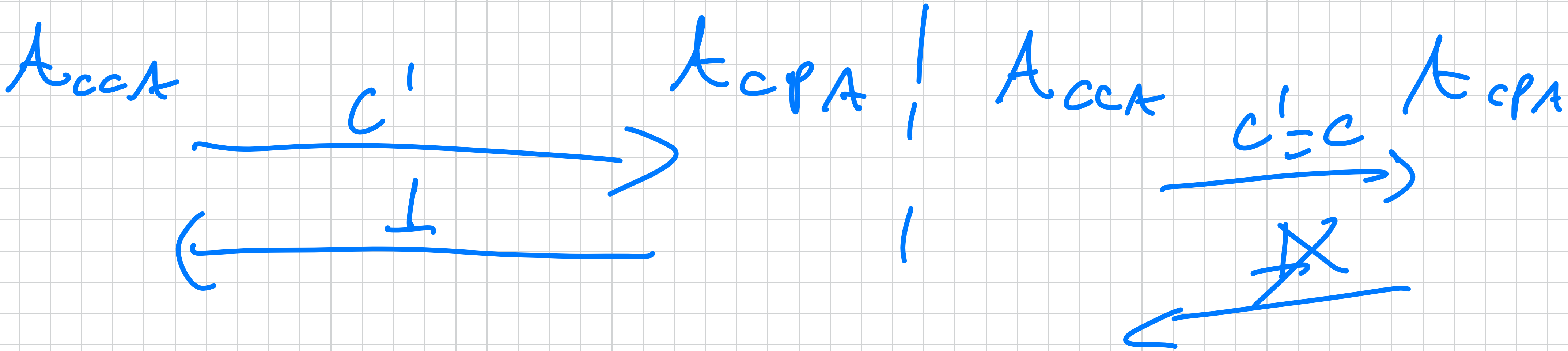
Proof. Sketch of proof. Main idea: Make a reduction from CPA to CCA.



Made with Goodnotes

Intuition: A_{cpa} needs to answer decryption queries exploiting AUTCt property.

AUTCt means no A can make valid $\tilde{c}$ so just answer Dec query with $\perp$.



Upon decryption query c':

- If $c' \in \{c\}$ returned in a previous encryption query m, return m ✓

- Else, answer $\perp$.

BAD event: A_{CCA} makes $\tilde{c}$ dec. query s.t. $\tilde{c} \notin \{c\}$ and $\mathrm{Dec}(k, \tilde{c}) \neq \perp$.

By AUTH: $\Pr[\mathrm{BAD}] \leq \mathrm{negl}(\lambda)$. ∎

LEMMA Approach 3) satisfies both CPA and AUTH*.

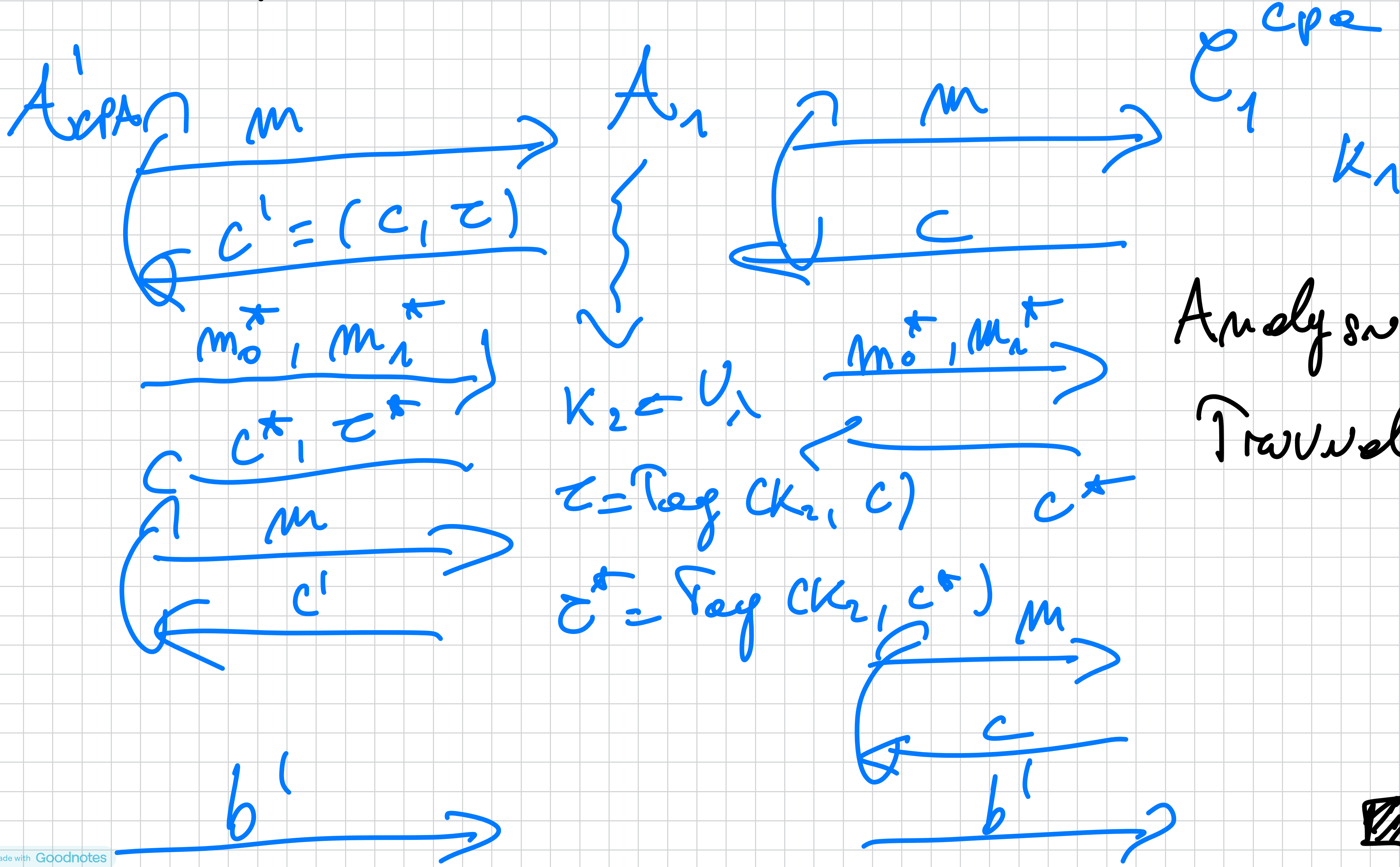
Proof. Approach 3):

$$c' = \mathrm{Enc}'((k_1, k_2), m) = (c, \tau)$$

$$c \leftarrow \mathrm{Enc}(k_1, m);\ \tau = \mathrm{Tag}(k_2, c).$$

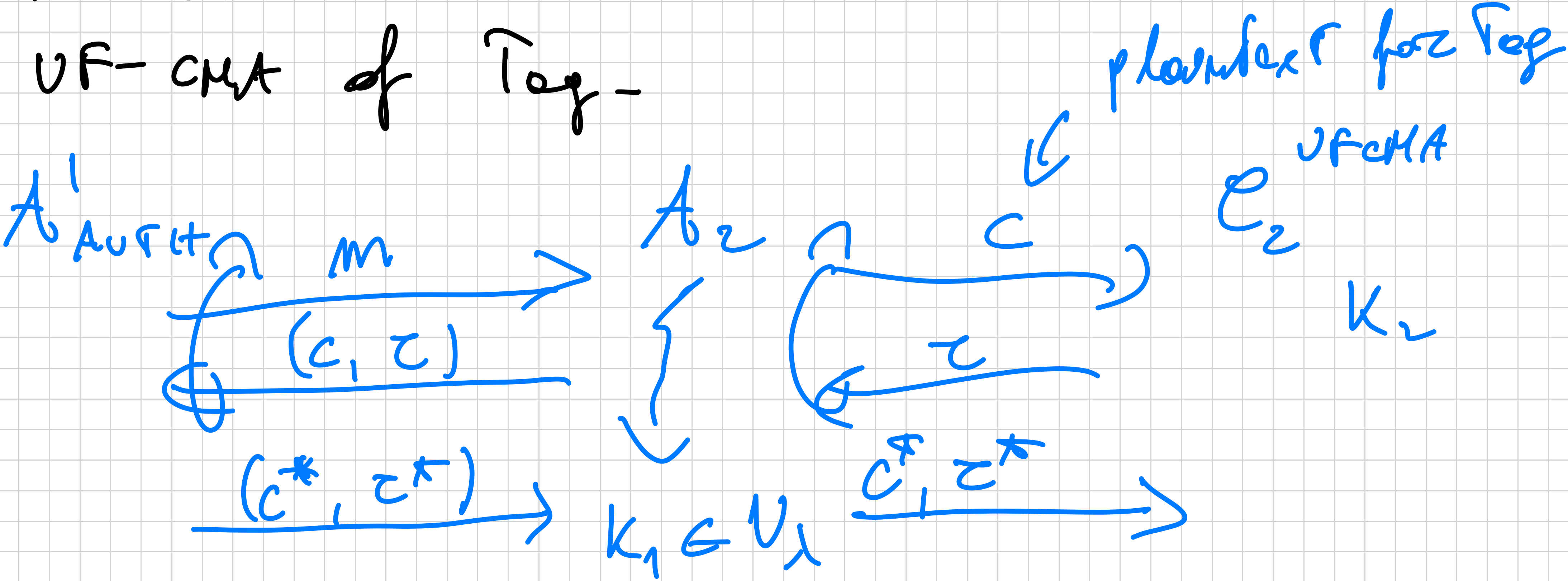
Made with Goodnotes

Let's start with CPA. By reduction to CPA sec. of (Enc, Dec) = Π_1



Analysis: Trivial. $\blacksquare$

It remains to show AUTH. Reduction to ?
UF-CMA of Tag.



$c \leftarrow Enc(k_1, m)$

When does A'_{AUTH} win? If:

1) $Tag(k_2, c^*) = \tau^*$
2) (c^*, τ^*) FRESH : $\neq \{(c, \tau)\}$

Made with Goodnotes

When does A_2 win?: If:

1) $\mathrm{Tag}(k_2, c^*) = \tau^*$ ✓
2) c^* FRESH: $\neq \{c\}$.

What if $c^* \in \{c\}$: A'_{AUTH} still wins, but A_2 does not!

$\hookrightarrow \tau^* \neq \{\tau\}$

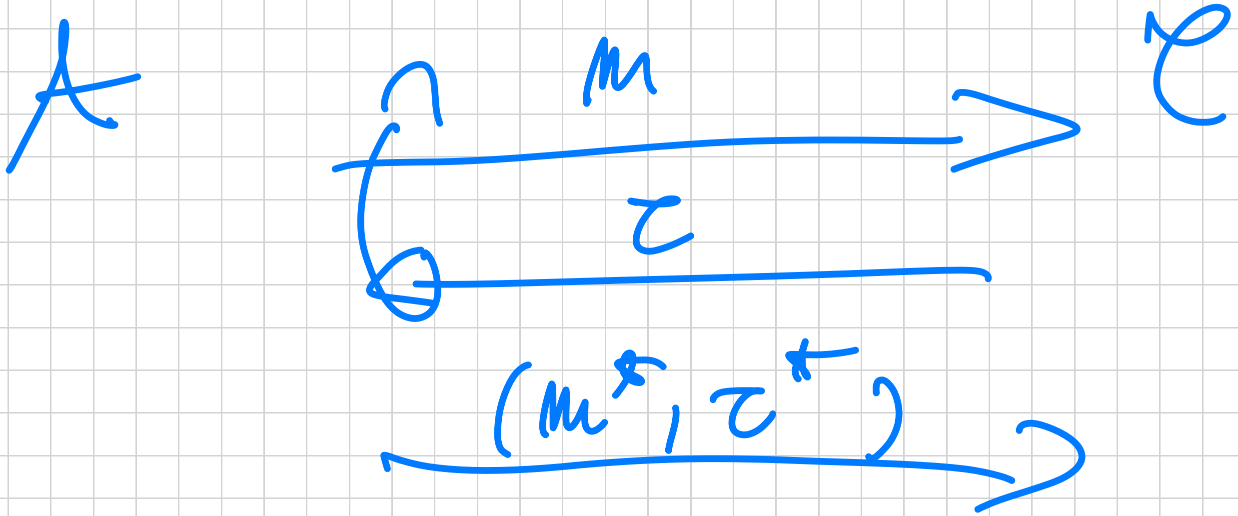
Here is a bad scheme:

$$\widetilde{\mathrm{Tag}}(k, m) = 0 \,\|\, \overbrace{\mathrm{Tag}(k, m)}^{\tau}$$

Bob: Discard first bit and check τ.

Still UF-CMA, because you can forge Tag only on messages for which you already queried the challenger.

* Way out: Assume each msg has a UNIQUE Tag. Alternatively, do not assume that but assume that Tag satisfies STRONG UF-CMA:



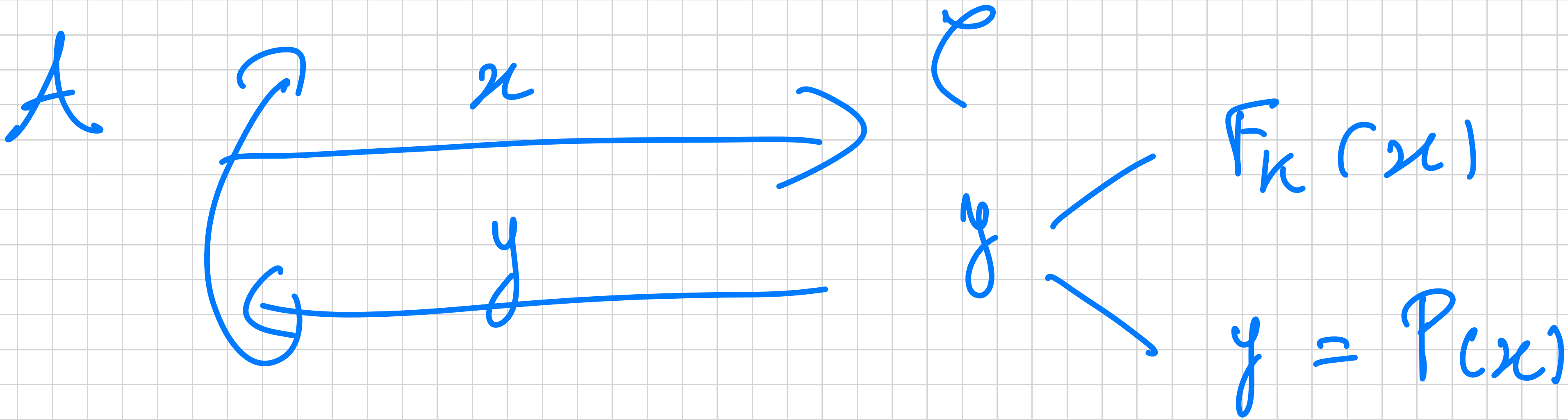
WIN IF:

- τ^* VALID
- τ^* FRESH

BLOCK CIPHERS

In practice: AES, DES, 3DES...

In theory: Pseudorandom permutation (PRP).



$P:\{0,1\}^n \to \{0,1\}^n$ chosen randomly among all permutations over $\{0,1\}^n$.

Made with Goodnotes

PRPs are efficiently invertible: $\exists$ PPT F^{-1} s.t. $F_K^{-1}(F_K(x)) = x \quad \forall x$.

e.g. some modes of operation require this.

How to build a PRP? Two approaches:

-) Provably secure way: Assume hardness of number theoretic problems (FACTORING, DISCRETE LOG, ...) or in fact ANY OWF.

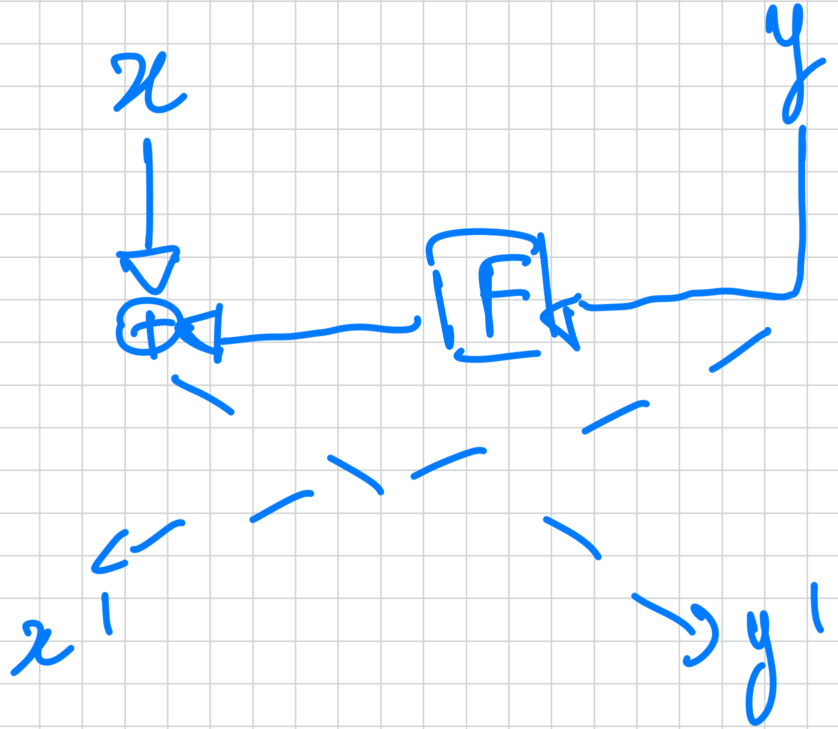
OWF $\Rightarrow$ PRGs $\Rightarrow$ PRF $\Rightarrow$ PRP.

-) Heuristic. Heuristically build a PRF and then make it a PRP (e.g. DES).

Made with Goodnotes

as the theoretician would do (almost).
The so-called Feistel Network.
Let $F : \{0,1\}^n \to \{0,1\}^n$ be a function (maybe a PRF). How to make it un-vertible?

$$\Psi_F(x, y) = (y, x \oplus F(y)) = (x', y')$$

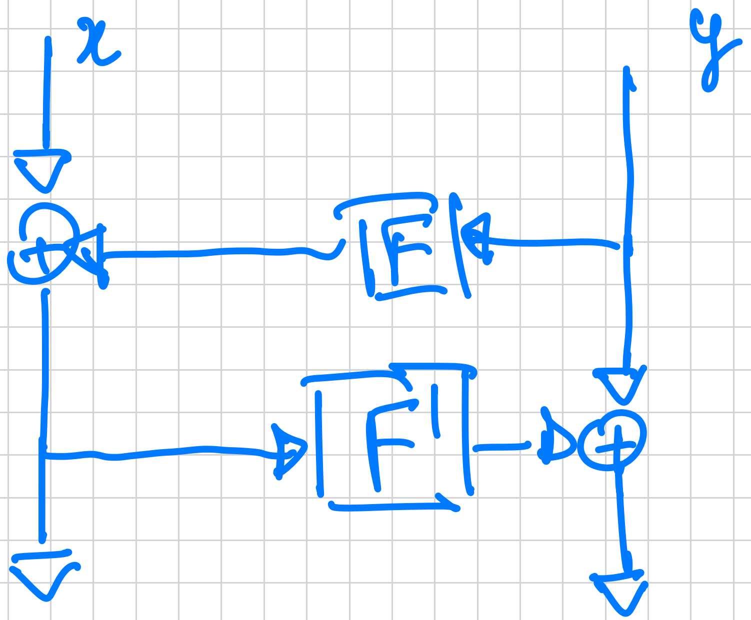


Note:

$$\Psi_F^{-1}(x', y') = (F(x') \oplus y', x') = (x, y)$$

Not a PRP! $\exists$ PPT A that breaks

NP. w. p. $1 - 2^{-n}$. But we can stack it.



$\Psi_{F,F'}(x, y) =$

$= (x'', y'')$

$x'' = x \oplus F(y)$ $\qquad$ $y \oplus F'(x \oplus F(y)) = y''$

still invertible! But not a PRP!

Note: $\Psi_{F,F'}(x, y) \oplus \Psi_{F,F'}(x', y)$

$= (x \oplus x', \underline{\qquad})$

Made with Goodnotes

Okay, do it another time.

THM. $\Psi_{F,F',F''}$ is a PRP assuming F, F', F'' are PRFs.

$\hookrightarrow$ $\mathcal{F} = \{ F_K : \{0,1\}^n \to \{0,1\}^n \}$

$$F \equiv F_{K_1} \, ; \, F' \equiv F_{K_2} \, ; \, F'' \equiv F_{K_3}$$

$$K_1, K_2, K_3 \leftarrow U_\lambda$$

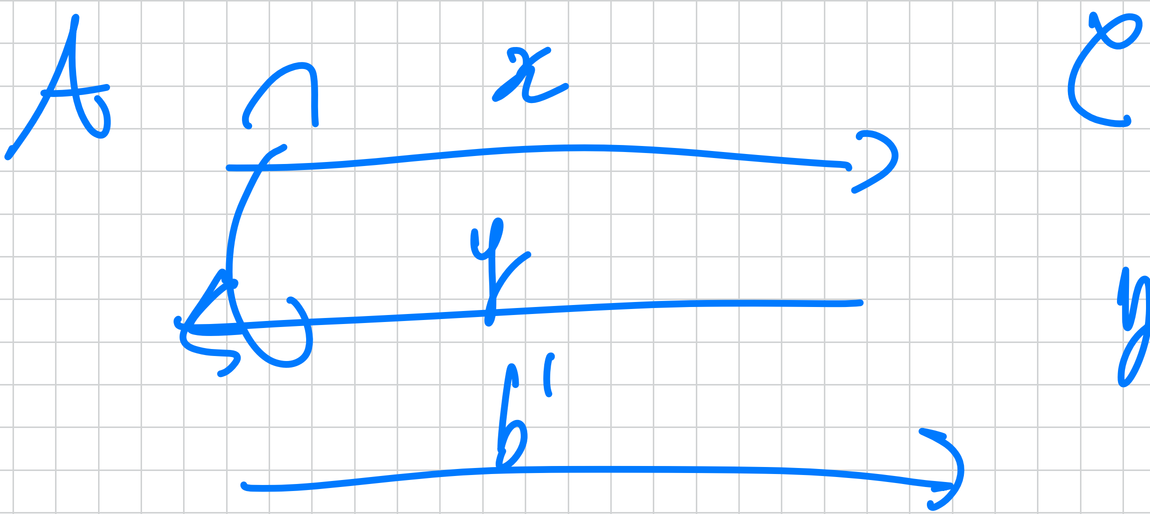
DES: $r = 18$ rounds! F is heuristic (confusion + diffusion); $K_1, K_2, K_3 \dots K_{18}$

Made with Goodnotes

derived from some K (using heuristic PRG).

Intuition for the proof:

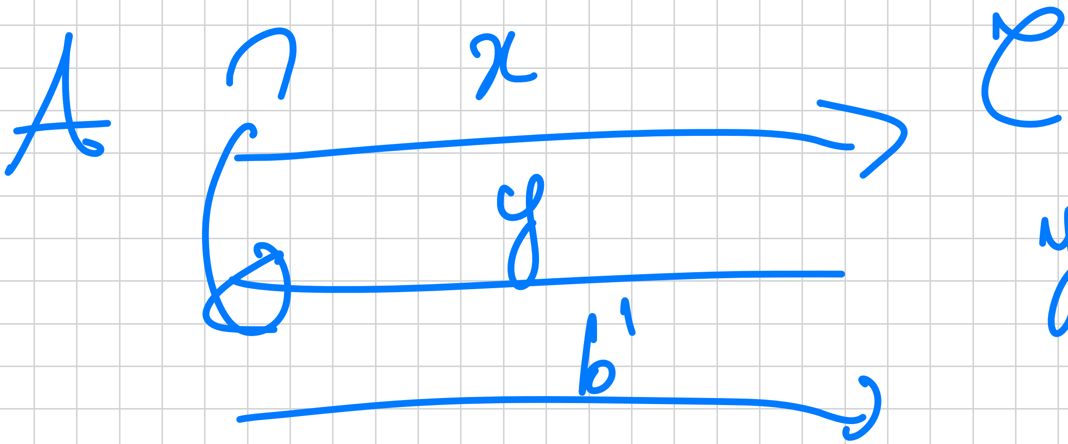
1)



T

$$y = \psi_{F_{k_1}, F_{k_2}, F_{k_3}}(x)$$

2)



S

$$y = \psi_{F, F', F''}(x)$$

3) $A \xrightarrow{x} \mathcal{C}$

$A \xleftarrow{y} \mathcal{C}$

$A \xrightarrow{b'} \mathcal{C}$

R

$y = R(x)$

R RANDOM FUNCTION

4) $A \xrightarrow{x} \mathcal{C}$

$A \xleftarrow{y} \mathcal{C}$

$A \xrightarrow{b'} \mathcal{C}$

P

$y = P(x)$

P RANDOM PERMUT.