Anamorphic-Resistant Encryption

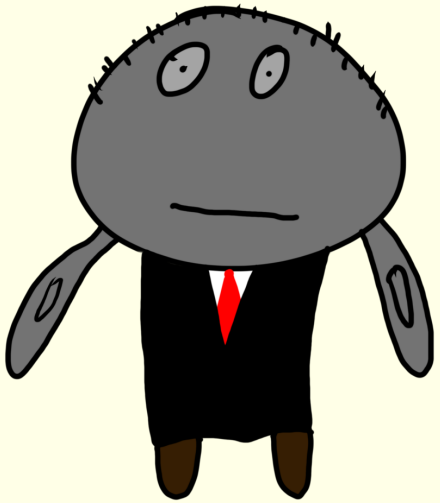
Or, why the encryption debate is still alive

Yevgeniy Dodis, **Eli Goldin**

Encryption Debate

Privacy Advocate

Mr. Government

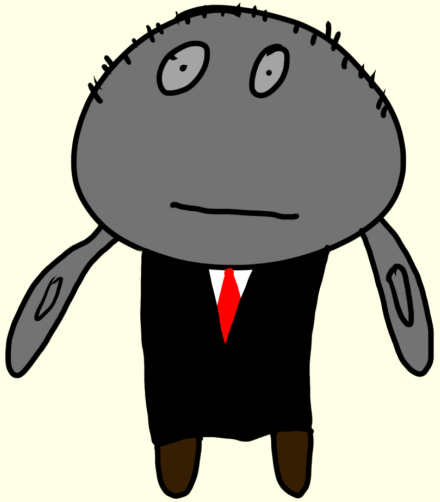


Encryption Debate

Privacy Advocate

- Everything should be encrypted

Mr. Government



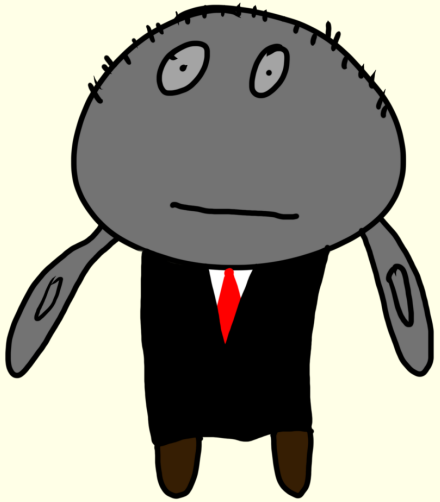
Encryption Debate

Privacy Advocate

- Everything should be encrypted

As long as I can read it.

Mr. Government



Encryption Debate

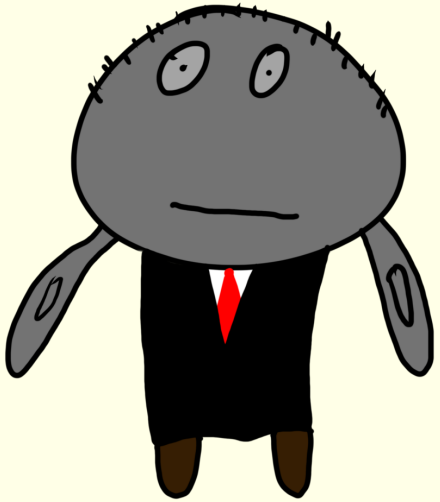
Privacy Advocate

Everything should be encrypted

As long as I can read it.

Add backdoor?

Mr. Government



Encryption Debate

Privacy Advocate

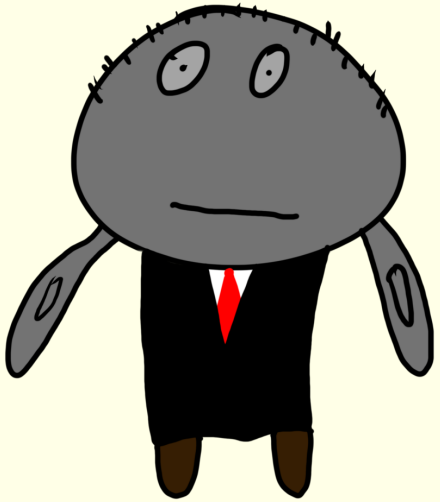
- Everything should be encrypted

As long as I can read it.

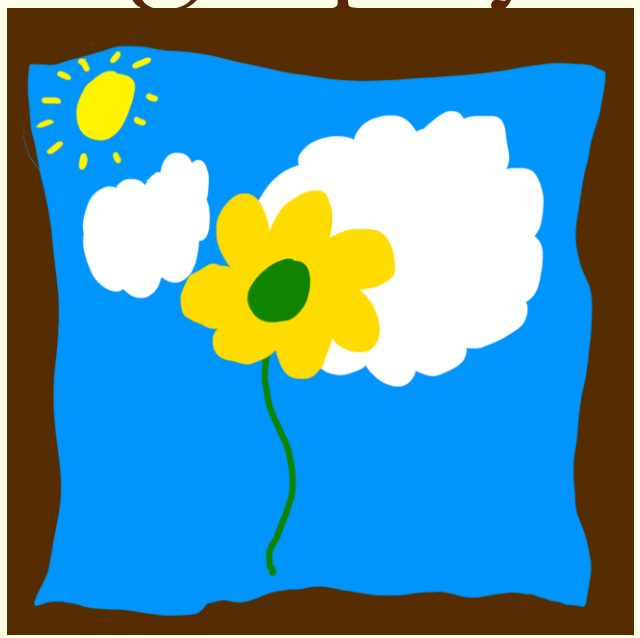
Add backdoor?

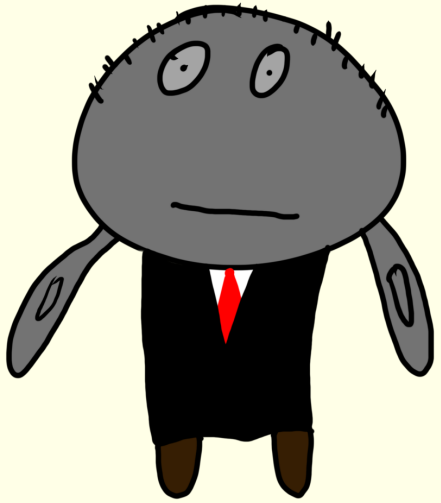
- No way, bad idea

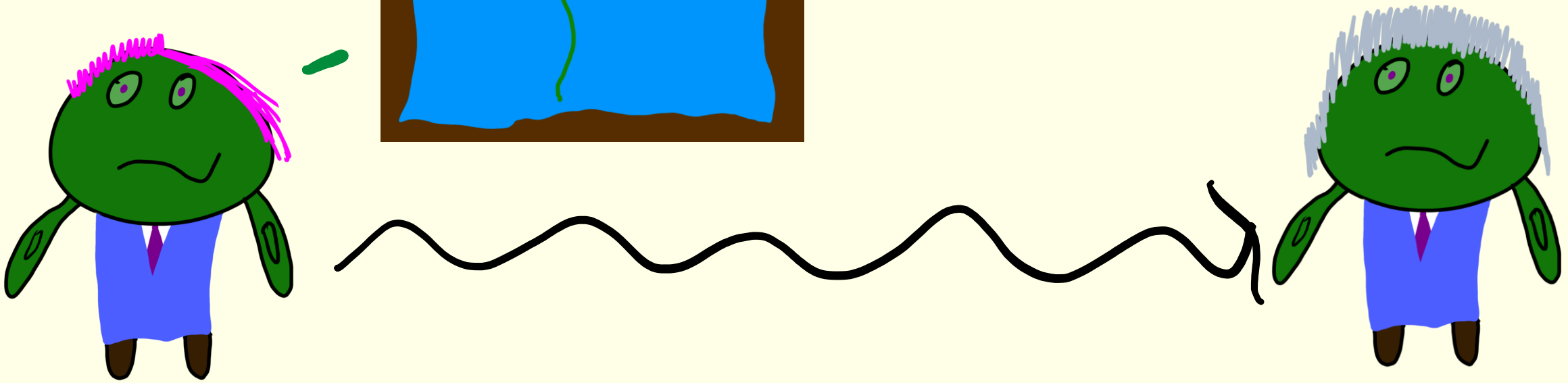
Mr. Government



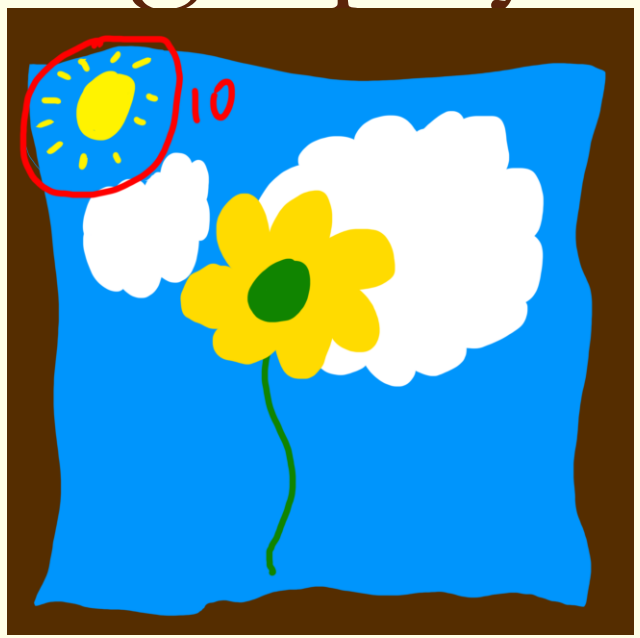
Steganography

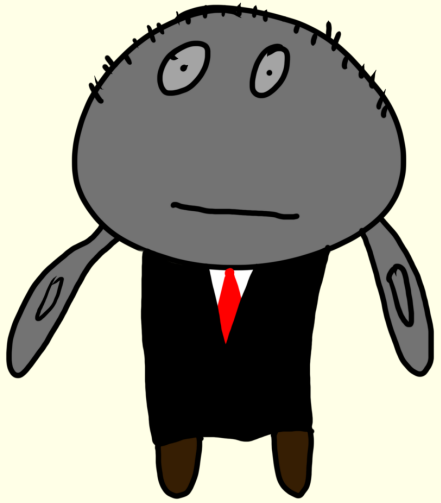




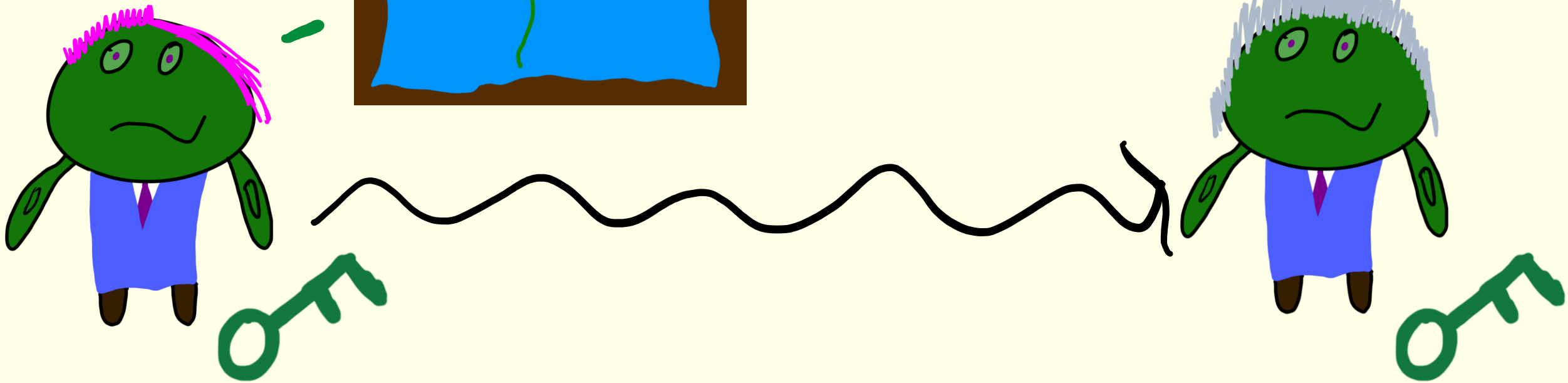


Steganography

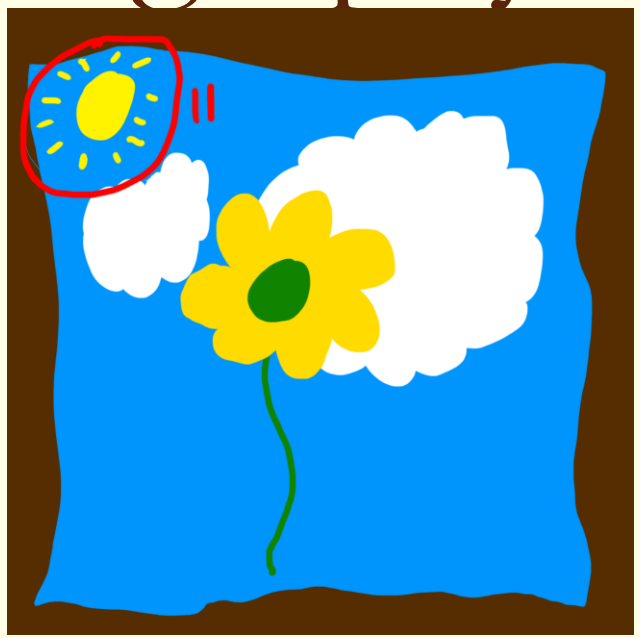


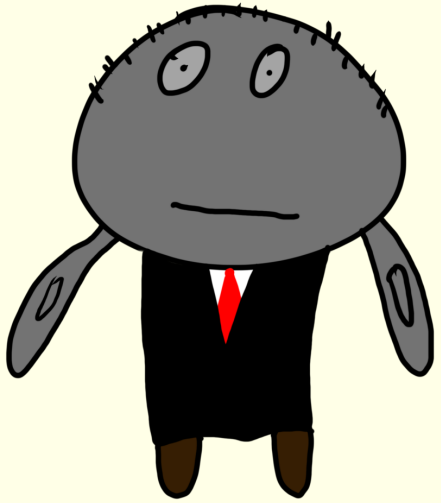


Overthrow the govt. at midnight

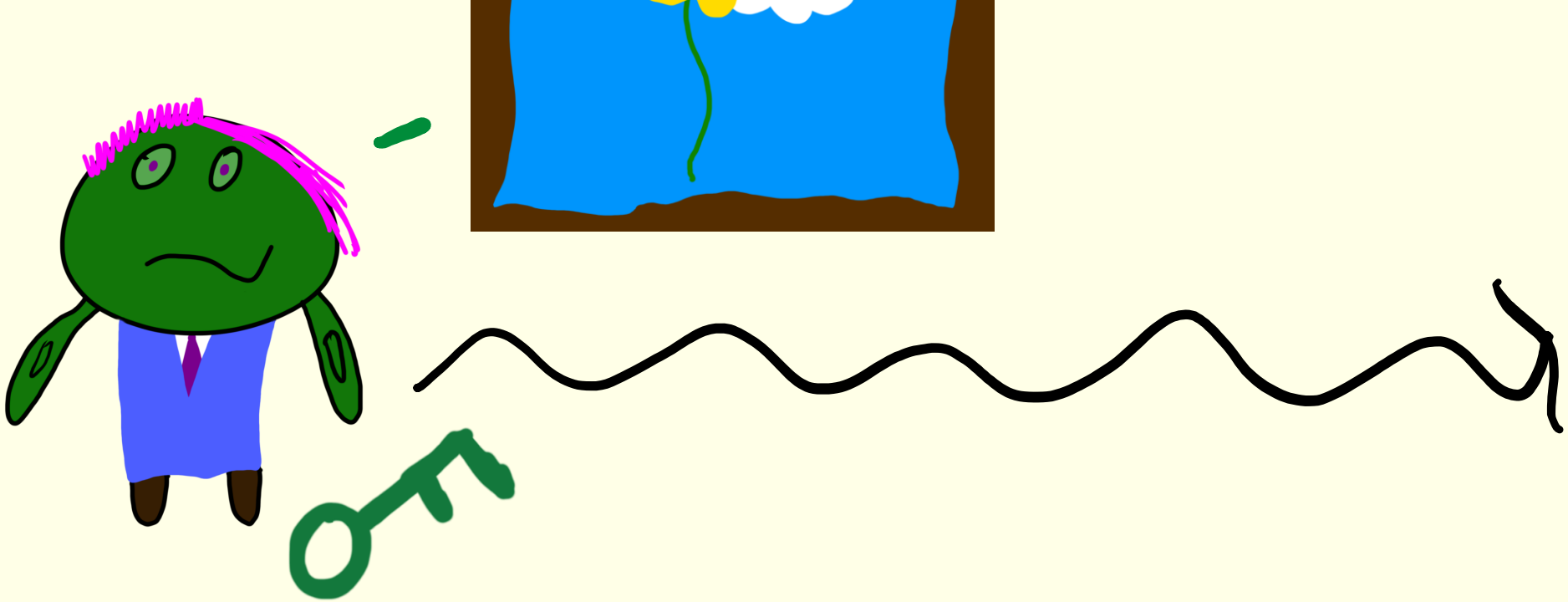


Steganography

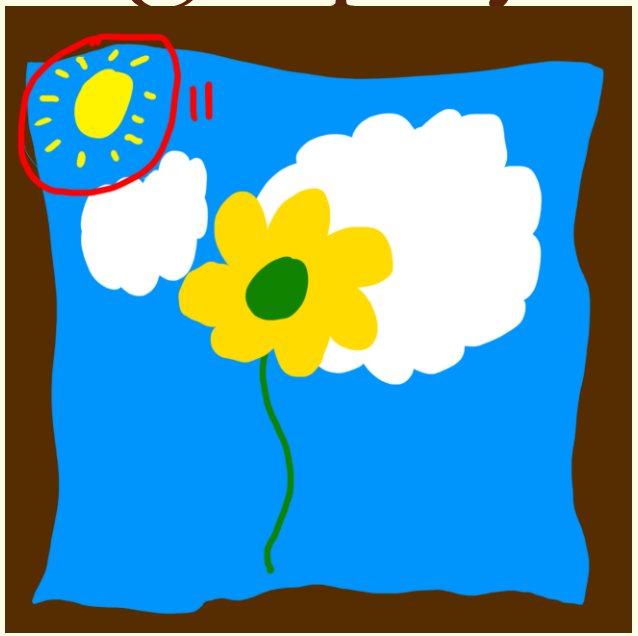


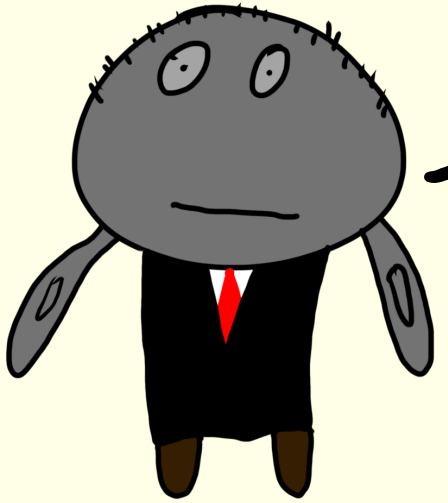


Overthrow the govt. at noon



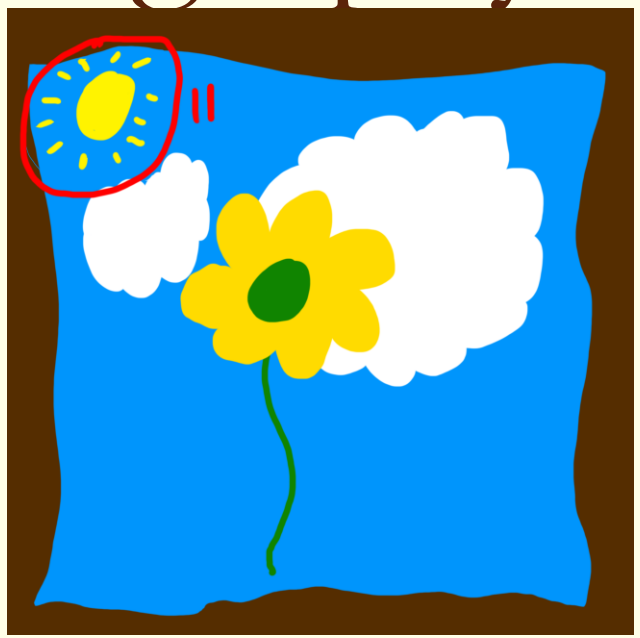
Steganography

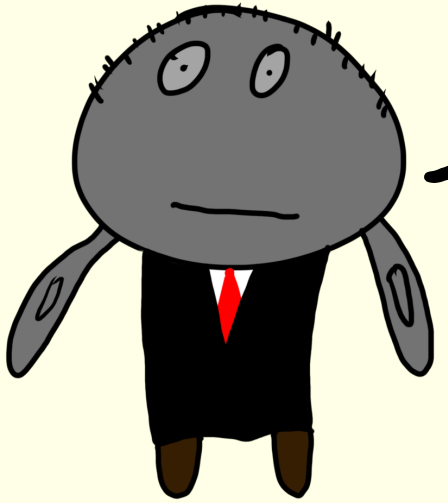




Looks like (comp. indis.) an honest photo to me.

Steganography





Looks like (comp. indis.) an honest photo to me.

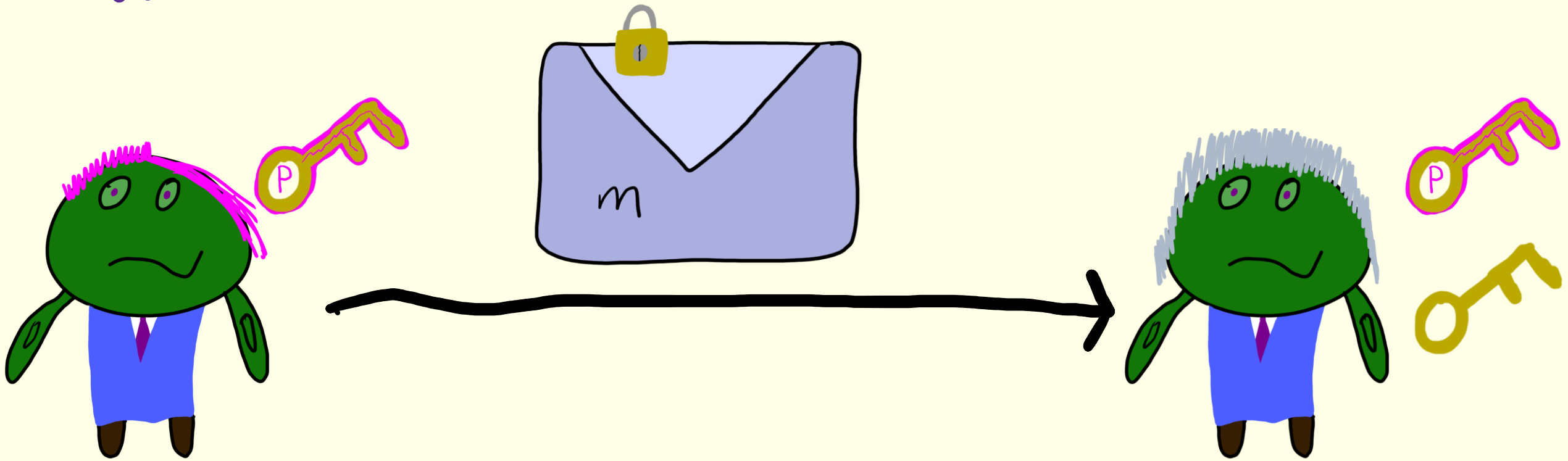
ALWAYS possible (w/ enough randomness)

Steganography

What must always be random?

Steganography

What must always be random?



Anamorphic Encryption

If you backdoor encryption, people will just use the subliminal channel.

[HPRV19, PPYZ22]

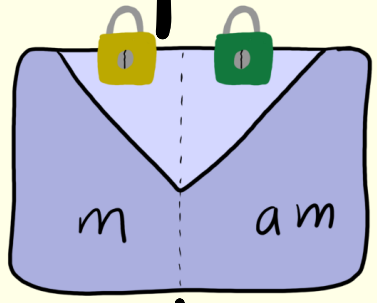
Anamorphic Encryption

Given encryption scheme
(Gen, Enc, Dec)
Anamorphic instantiation is protocol
(AGen, AEnc, ADec) formalizing
steganographic channel

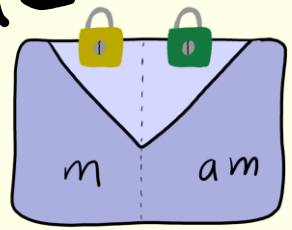
Anamorphic Encryption

(Gen, Enc, Dec)

$A.Gen \rightarrow$ 

$AEnc(m, am) \rightarrow$ 

act

$ADec($  $,$ 

$) \rightarrow am$

ak , act

Anamorphic Encryption (Gen, Enc, Dec)

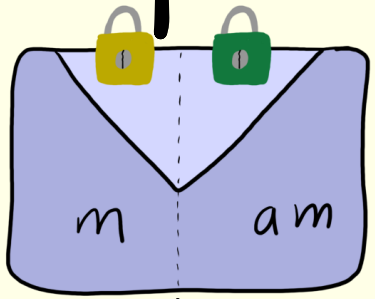
A Gen →

pk

sk

ak

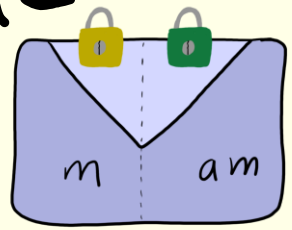
AEnc(m, am) →



act

ADec(

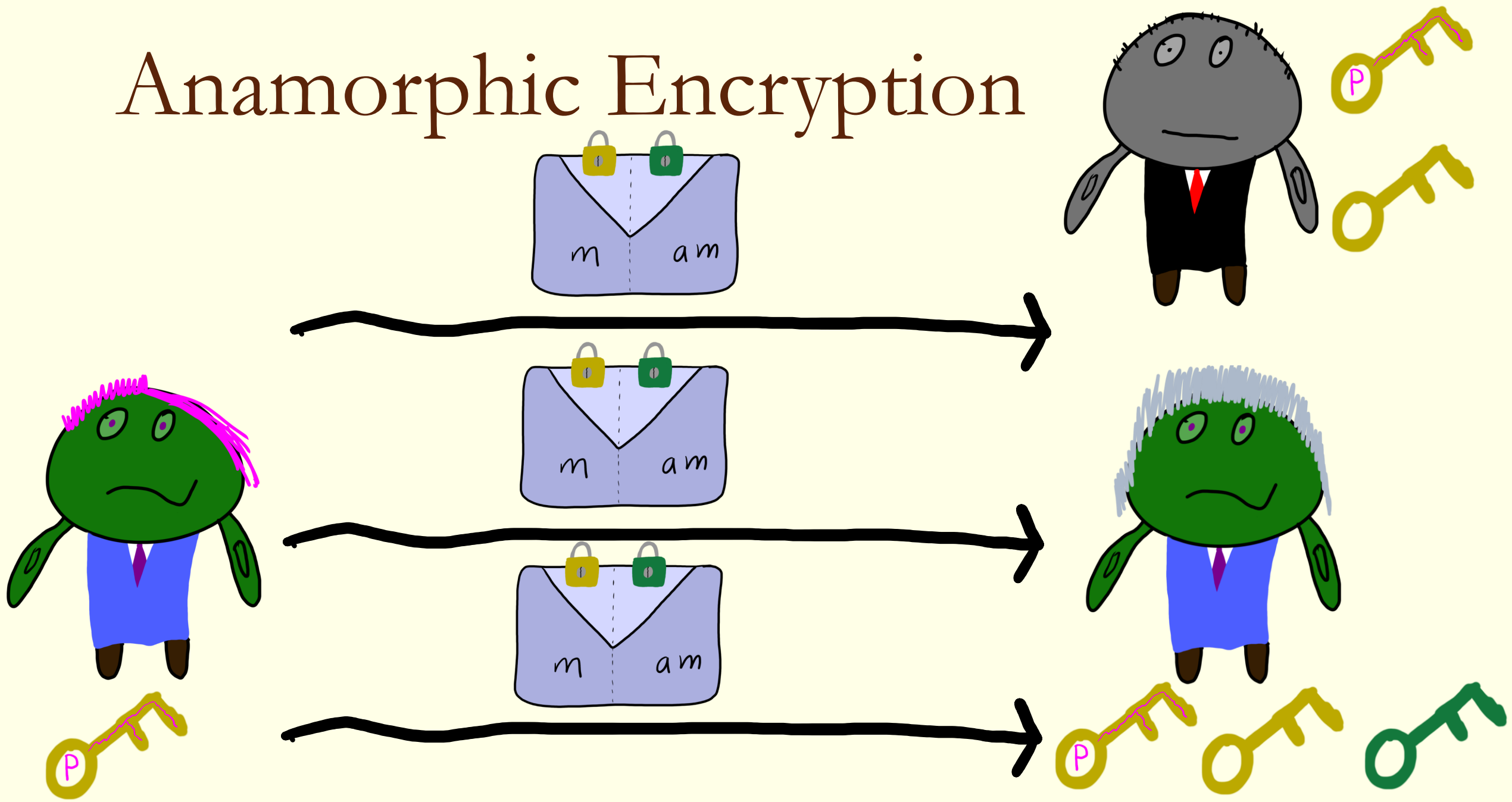
ak ,



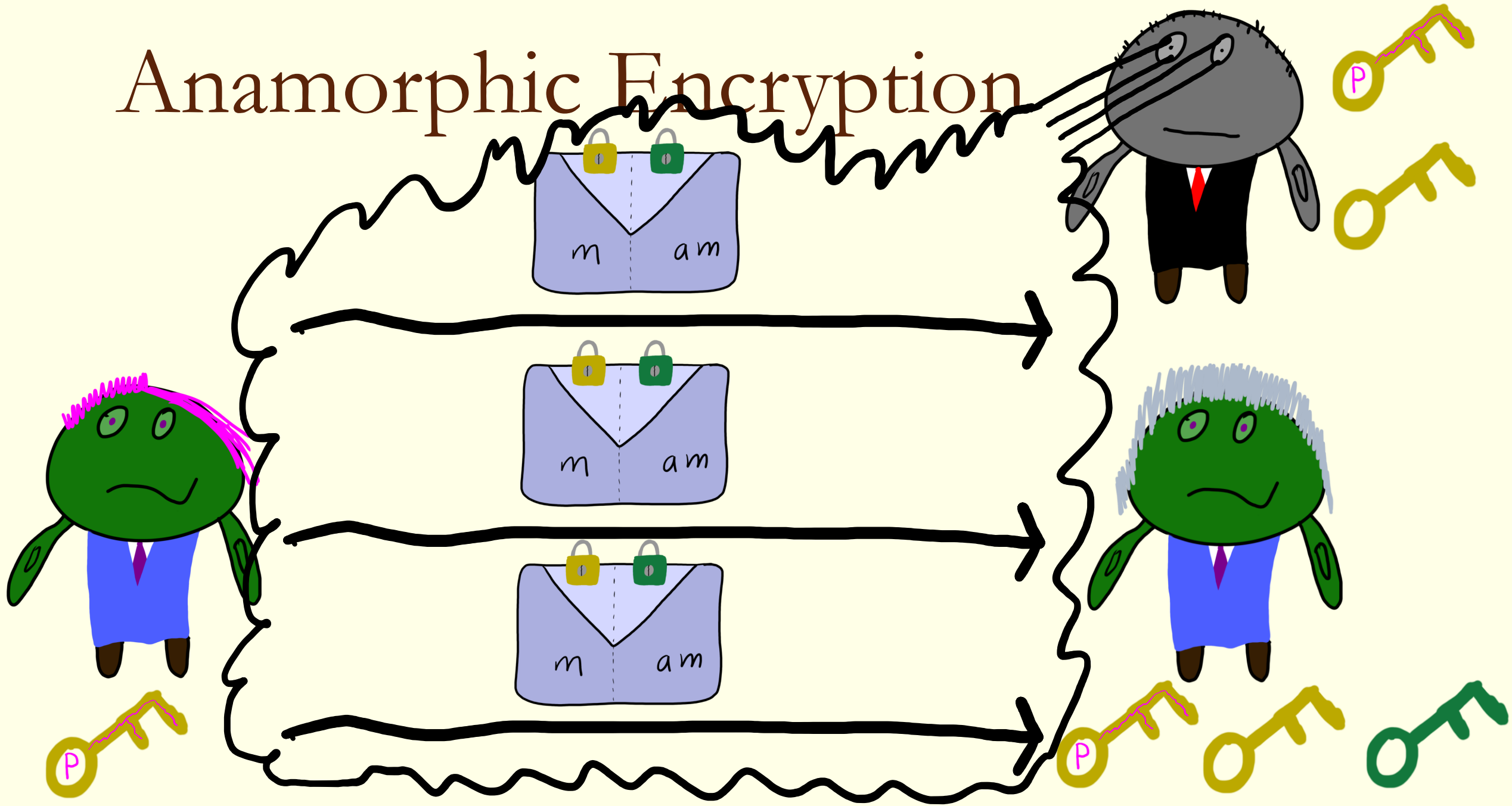
act) → am

"anamorphic instantiation of (Gen, Enc, Dec)"

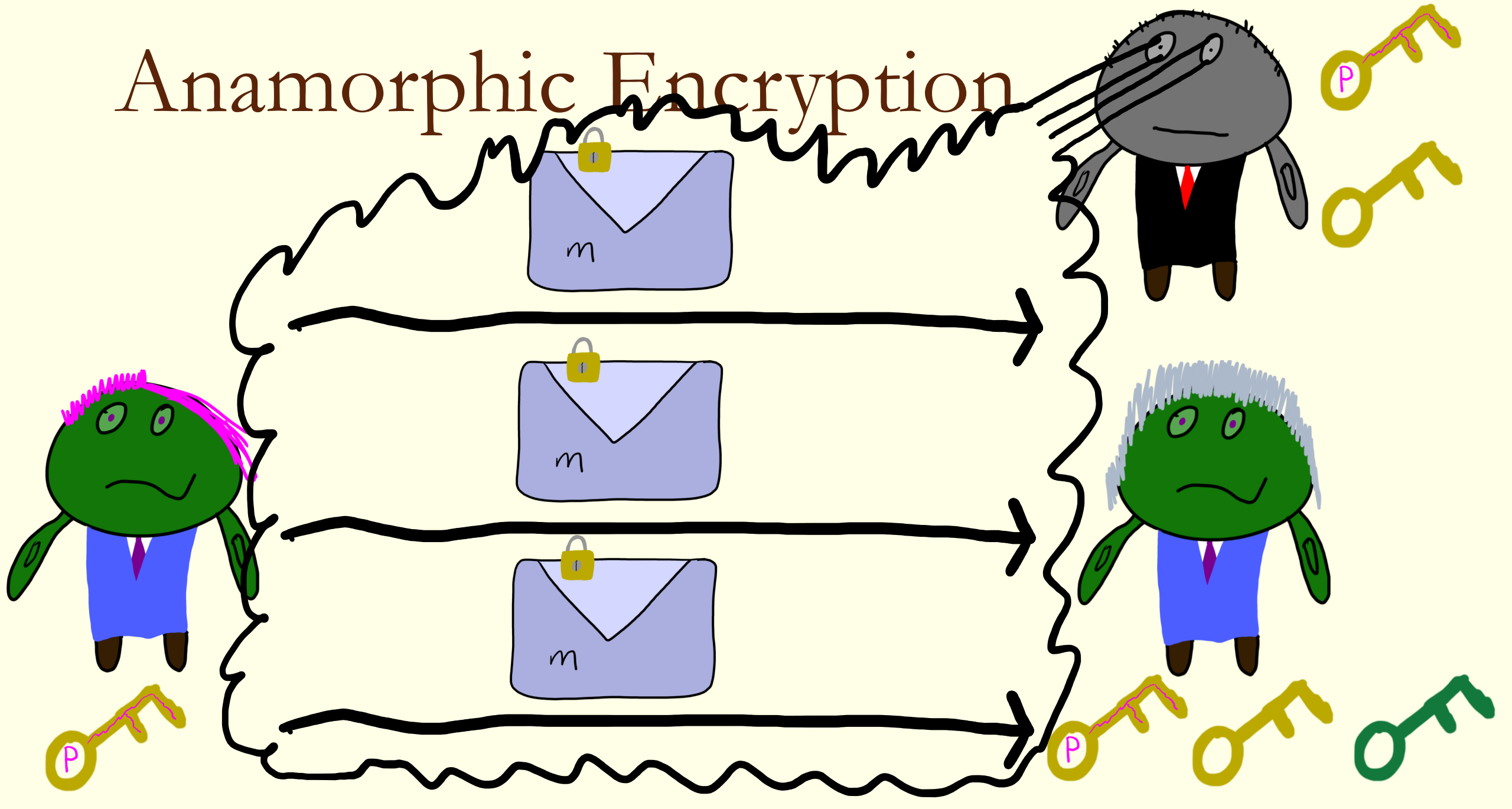
Anamorphic Encryption



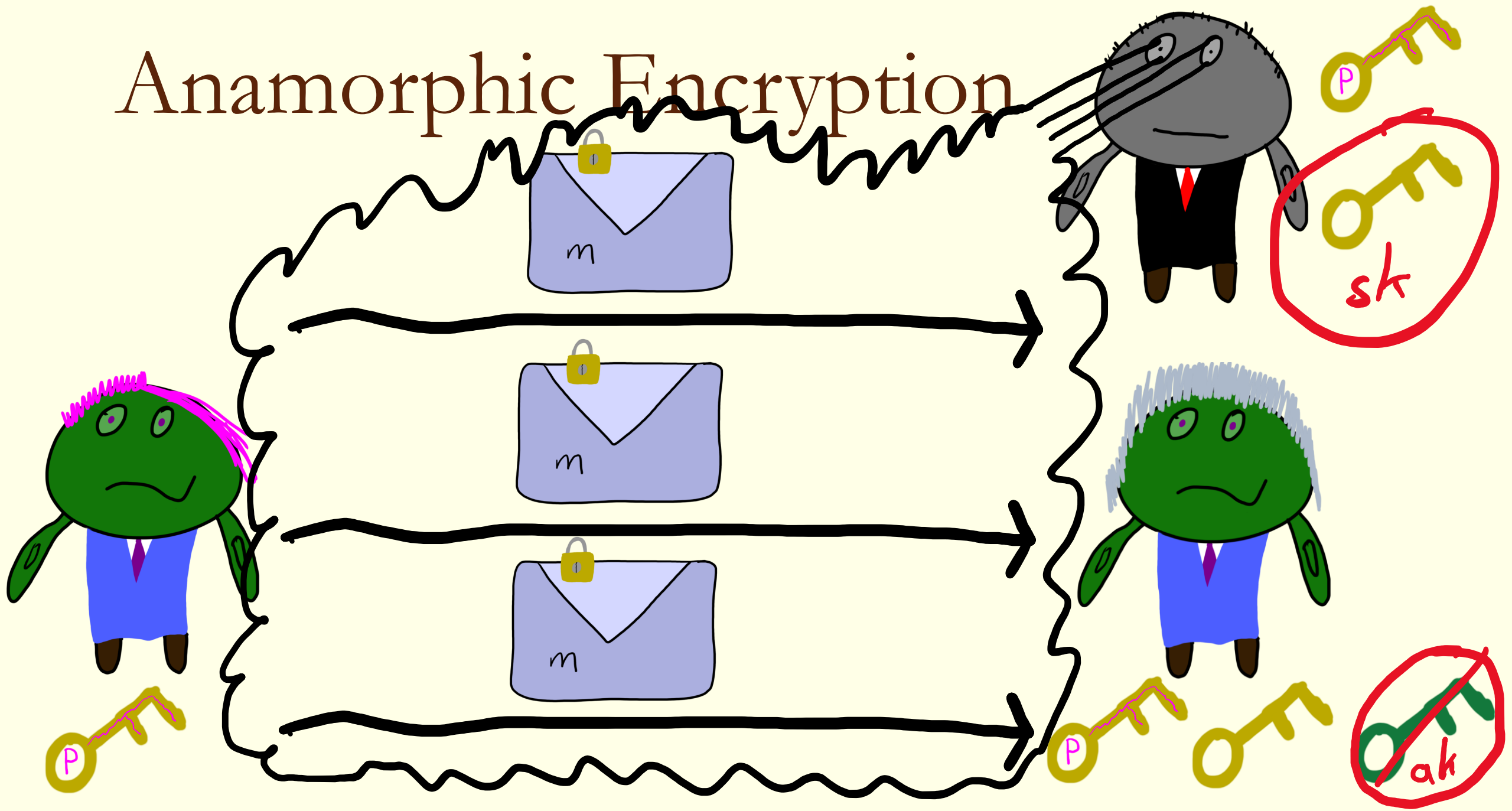
Anamorphic Encryption



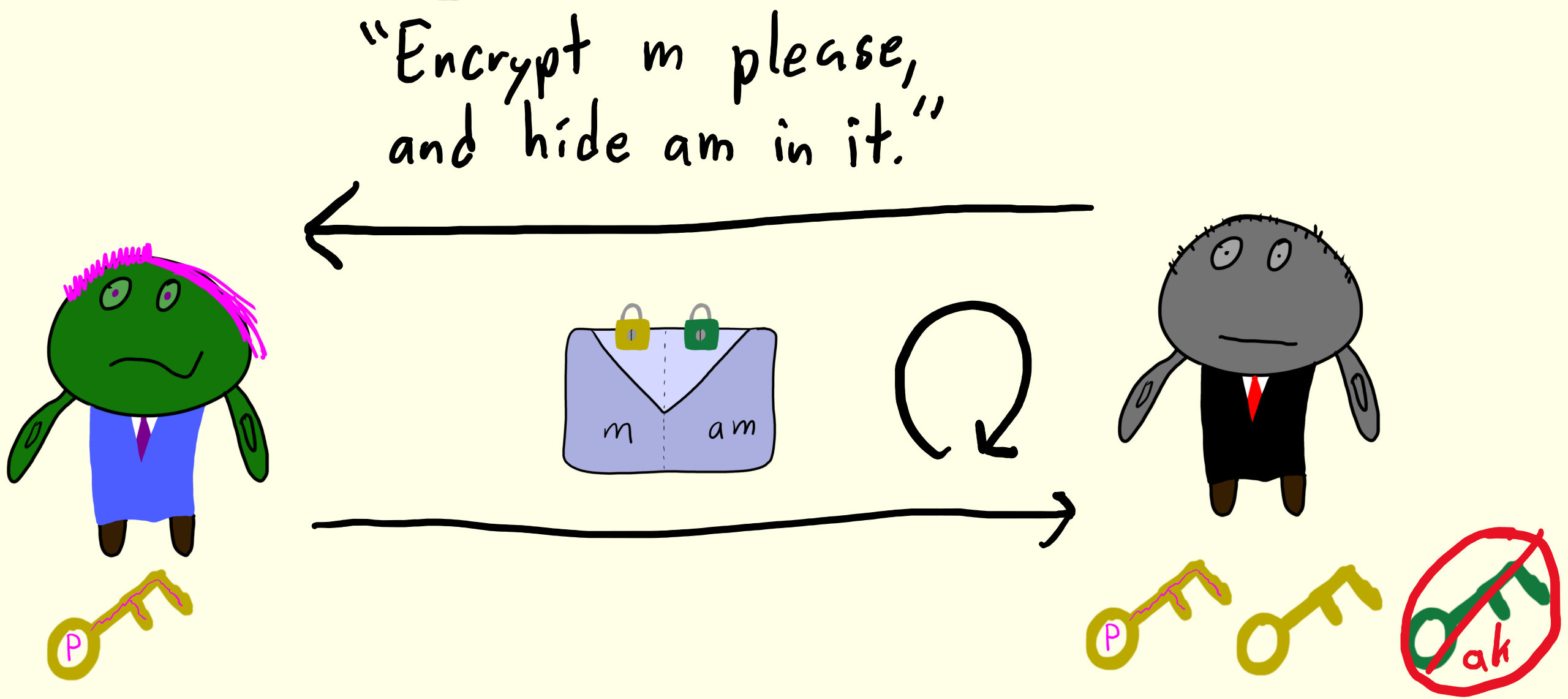
Anamorphic Encryption



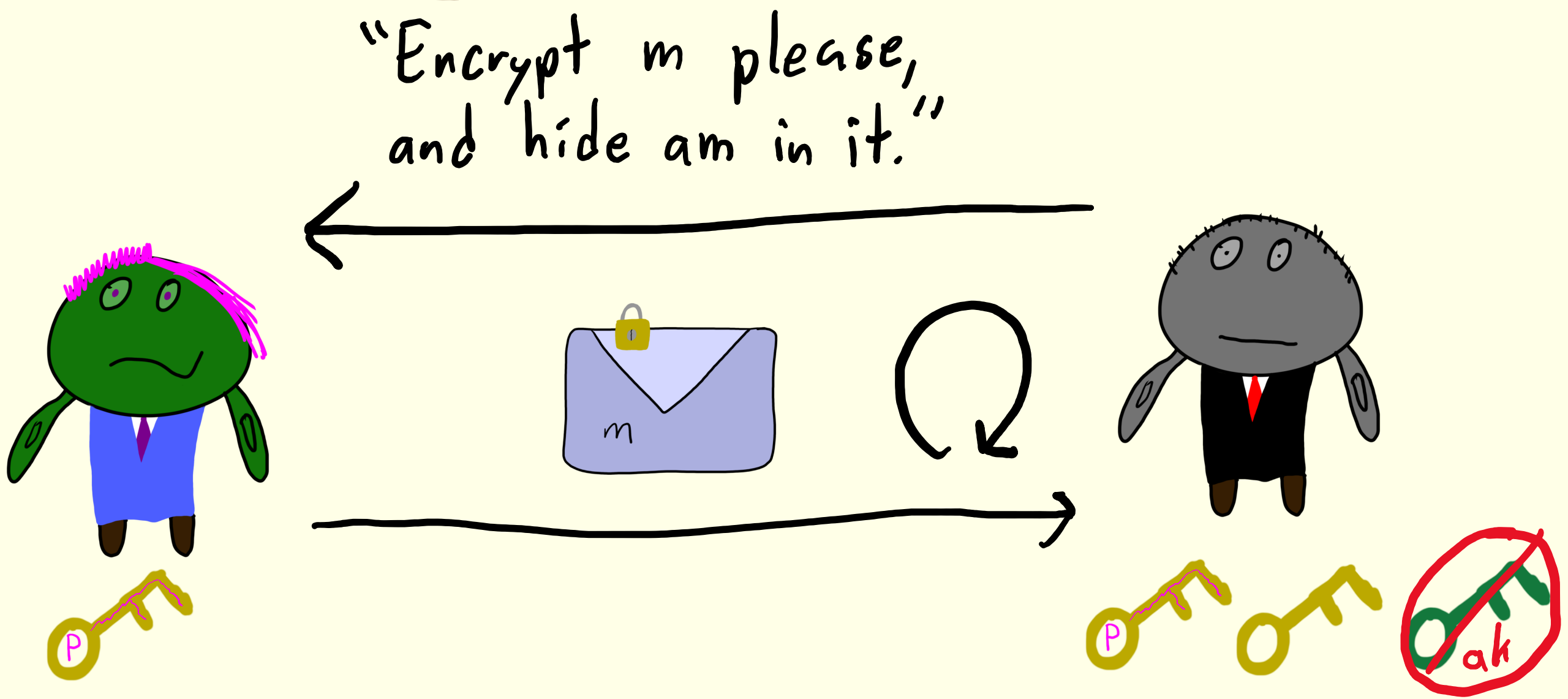
Anamorphic Encryption



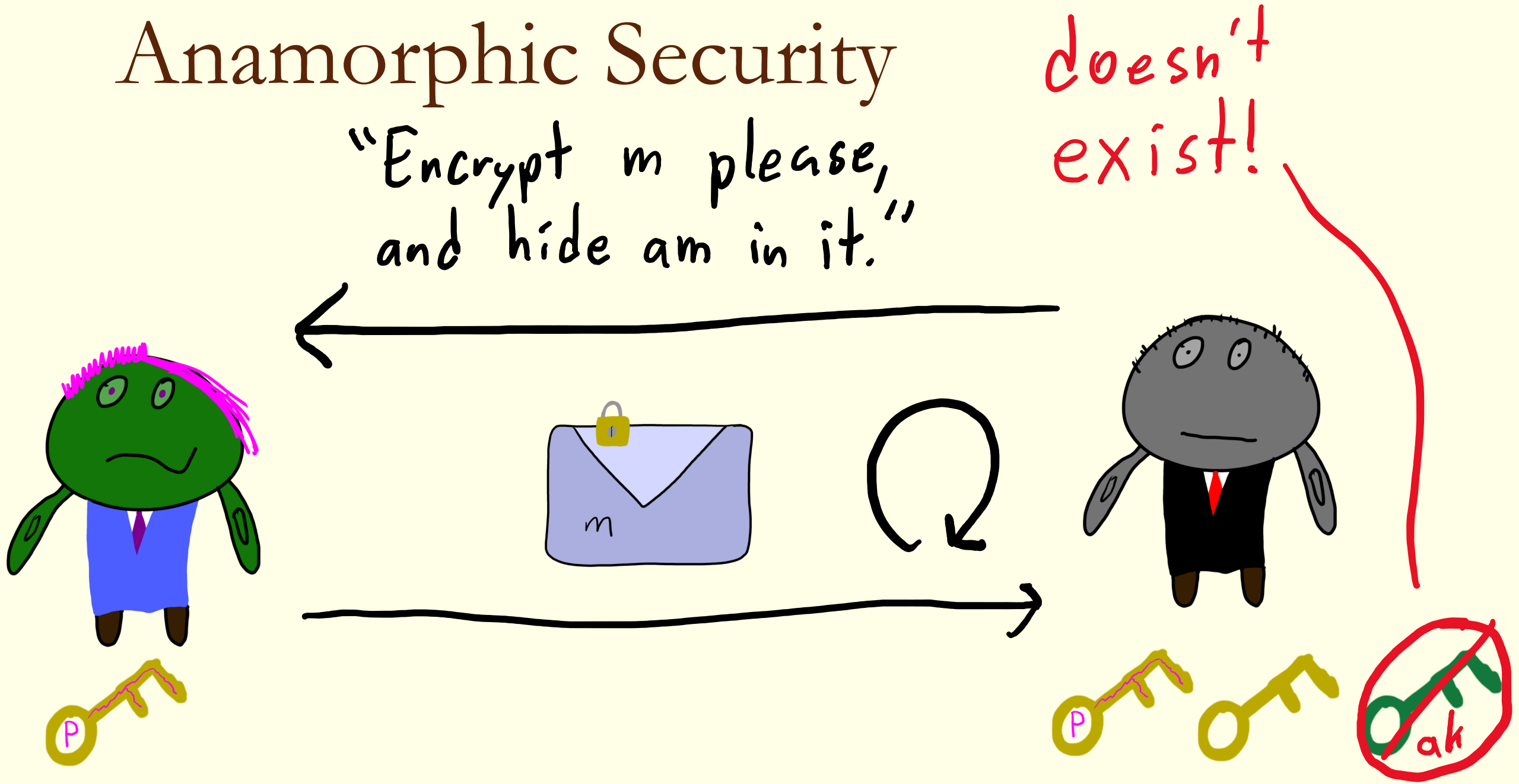
Anamorphic Security



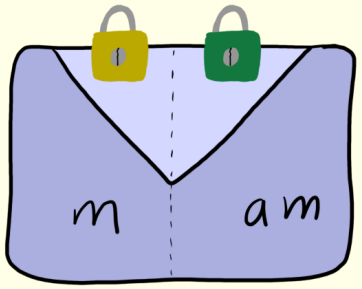
Anamorphic Security



Anamorphic Security

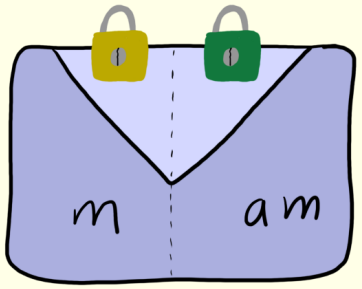


Anamorphic Encryption

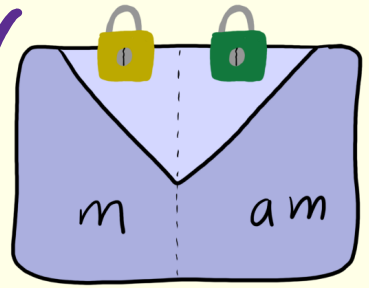


always possible by rejection sampling

Anamorphic Encryption

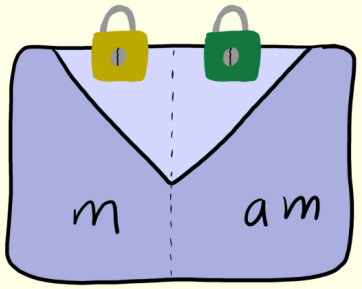


always possible by rejection sampling

$H_{\text{🔑}}$(

) = am

Anamorphic Encryption



always possible by rejection sampling

Bandwidth: $O(\log \lambda)$

Anamorphic Encryption

Generic

$O(\log \lambda)$ bandwidth

[PPY22]

Linear Bandwidth

Anamorphic Encryption

Generic

$O(\log \lambda)$ bandwidth [PPY22]

Linear Bandwidth

Paper	Applicable Schemes	Properties			
[PPY22]	Naor-Yung	—	—	—	—

Anamorphic Encryption

Generic	Linear Bandwidth		
	Applicable Schemes		Properties
	Paper		
$O(\log \lambda)$ bandwidth [PPY22]	[PPY22]	Naor-Yung	robust
	[BGH+24]	randomness recoverable	

Anamorphic Encryption

Generic

$O(\log \lambda)$ bandwidth

[PPY22]

Linear Bandwidth

Paper	Applicable Schemes	Properties
[PPY22]	Naor-Yung	robust
[BGH+24]	randomness recoverable	
[PPY24]	many CCA PKEs	public-key anamorphism
⋮		

Anamorphic Encryption

Generic

$O(\log \lambda)$ bandwidth [PPY22]

Linear Bandwidth

Paper	Applicable Schemes	Properties
[PPY22]	Naor-Yung	
[BGH+24]	randomness recoverable	robust
[PPY24]	many CCA PKEs	public-key anamorphism
⋮	([KPP+23]×2, [CGM24], ...)	

Anamorphic Encryption

Hypothesis: linear bandwidth anamorphic instantiations are always possible.

Anamorphic Encryption

Hypothesis: linear bandwidth anamorphic instantiations are always possible.

At least for CCA schemes? [PPYZ24]

Anamorphic Encryption

Hypothesis: linear bandwidth anamorphic instantiations are always possible.

At least for CCA schemes? [PPYZ24]

Spoiler: NO

Philosophy

Philosophical Question

Who picks what PKE schemes are legal?

Dictatoria

Wants to read all messages (universal backdoor)

Dictatoria

Wants to read all messages (universal backdoor)

Privacy against foreign nations

Dictatoria

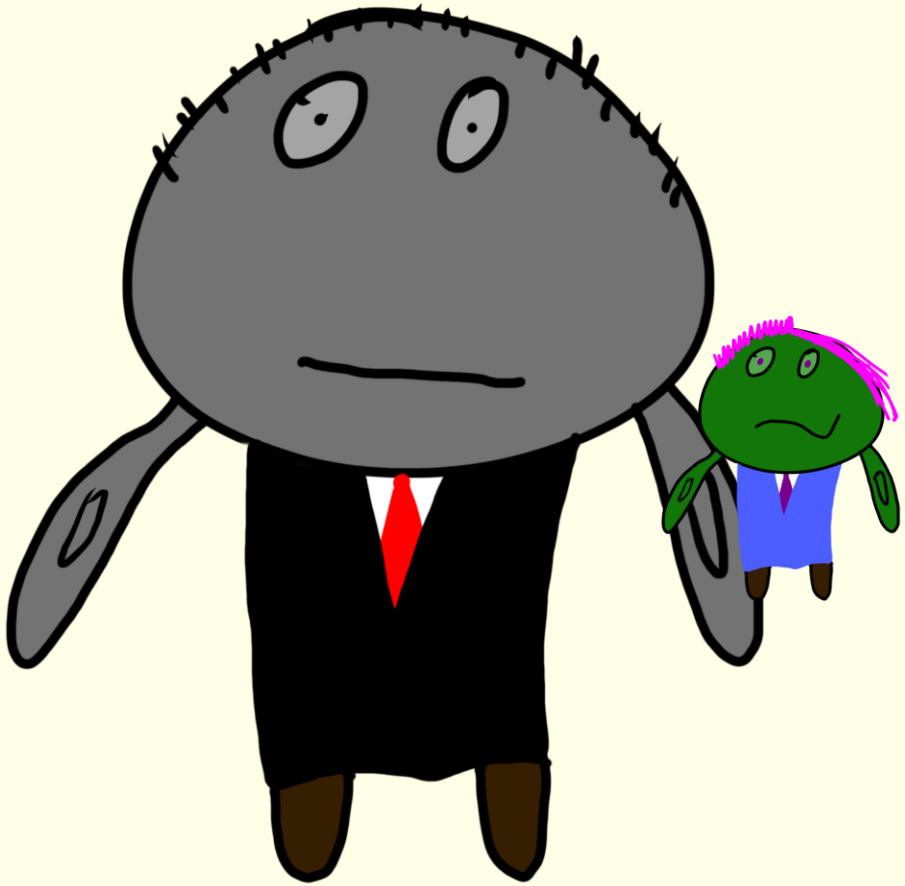
Wants to read all messages (universal backdoor)

Privacy against foreign nations

Outlaw non-trivial anamorphism

Warrantland

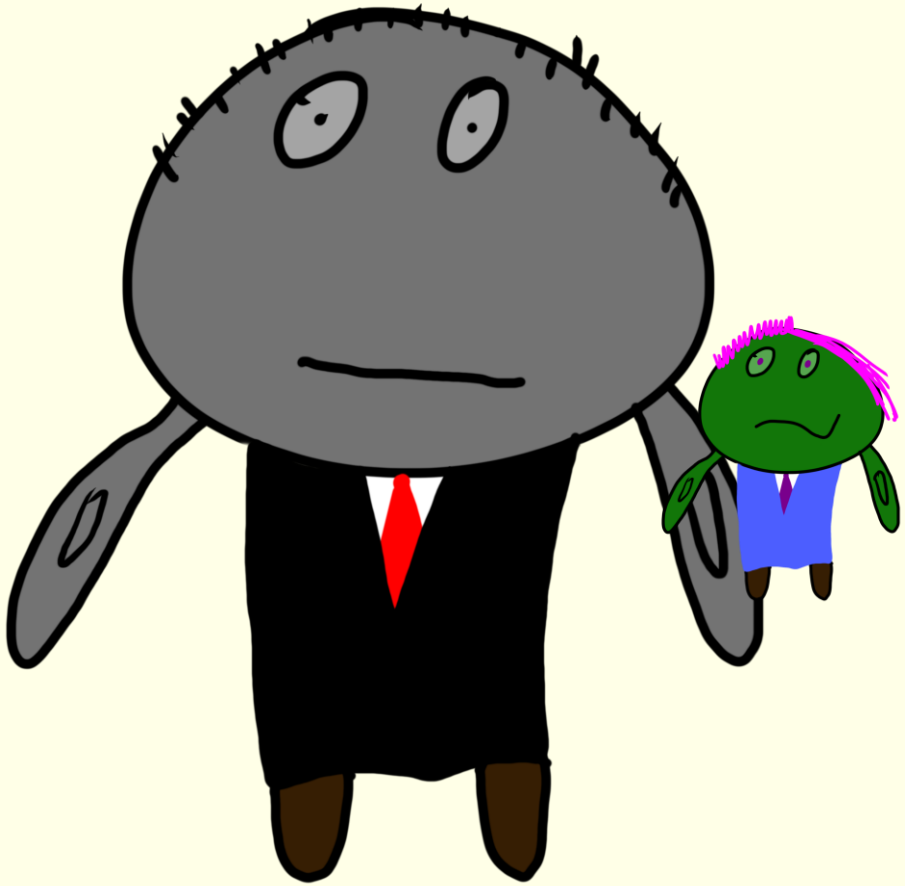
Wants to read messages when warrant issued



Warrantland

Wants to read messages when warrant issued

Privacy against itself with no warrant

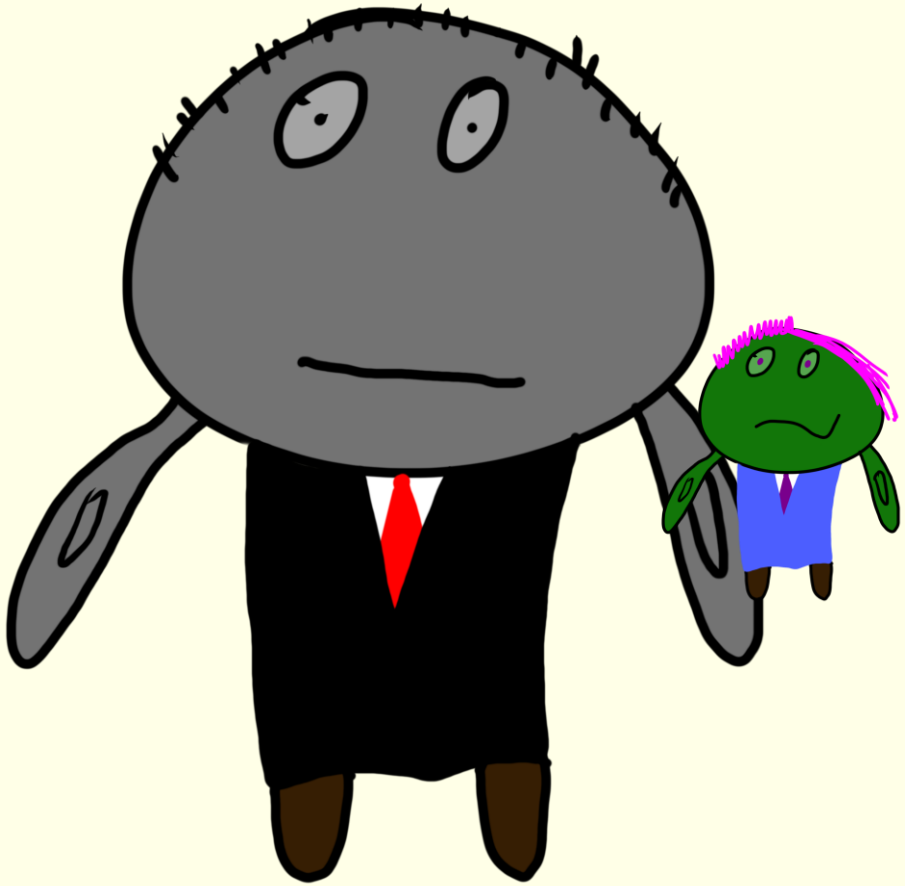


Warrantland

Wants to read messages when warrant issued

Privacy against itself with no warrant

Outlaw non-trivial anamorphism



Privatopia

Privacy against itself now and in the future

Privatopia

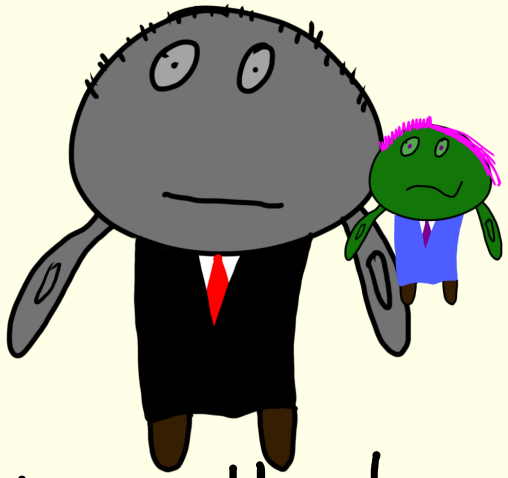
Privacy against itself now and in the future

Wants to standardize anamorphic encryption

Main Philosophical Question

Are there schemes which make

Dictatoria



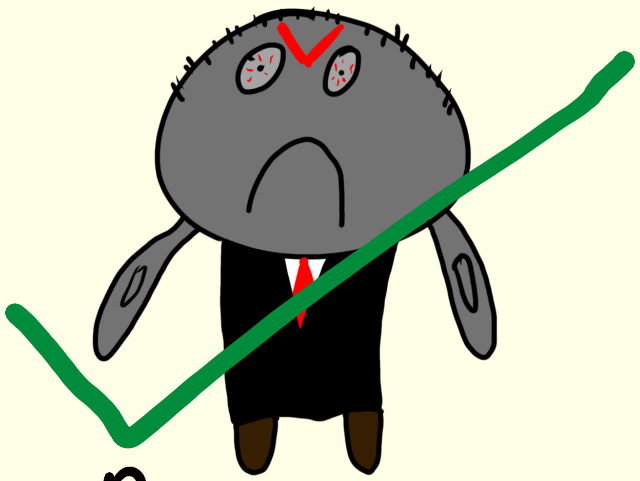
Warrantland

Prioritopia

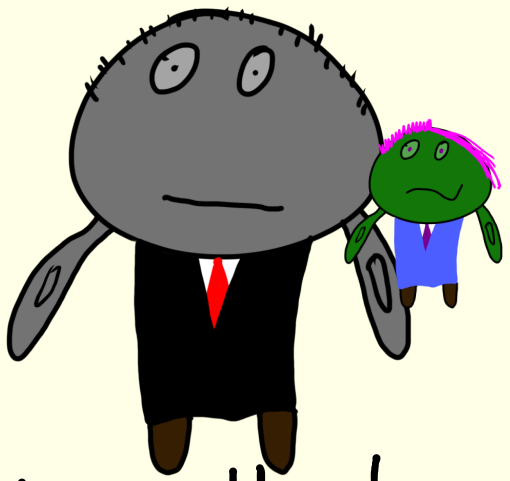
happy?

Main Philosophical Question

Are there schemes which make



Dictatoria



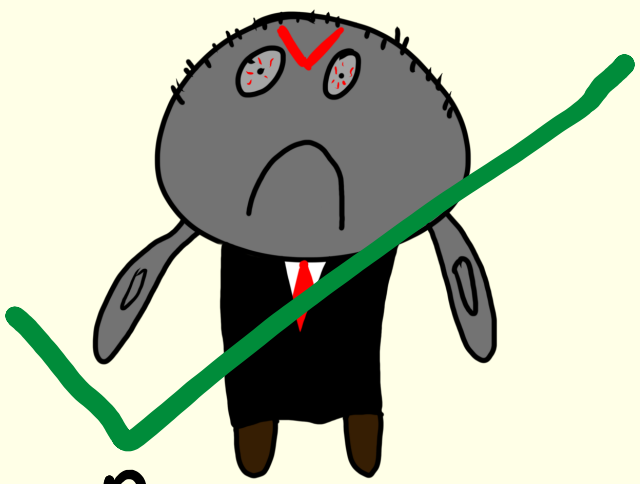
Warrantland

Prioritopia

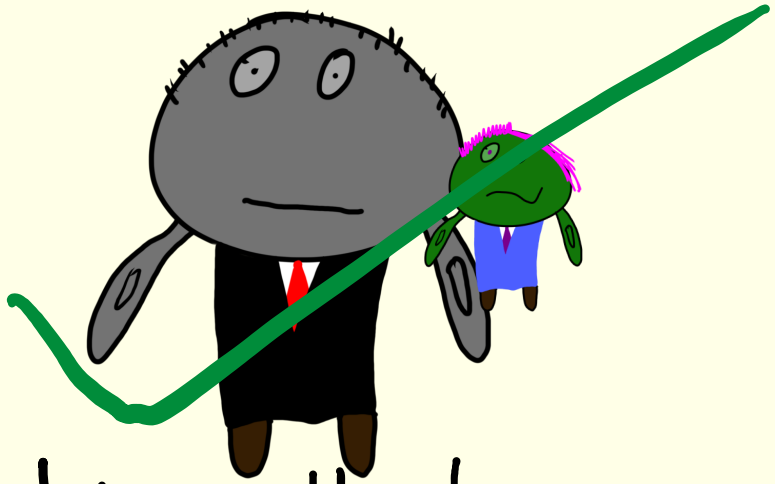
happy?

Main Philosophical Question

Are there schemes which make



Dictatoria



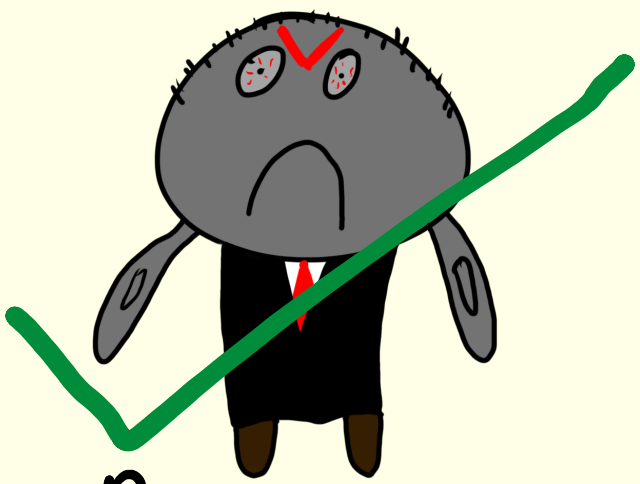
Warrantland

Prioritopia

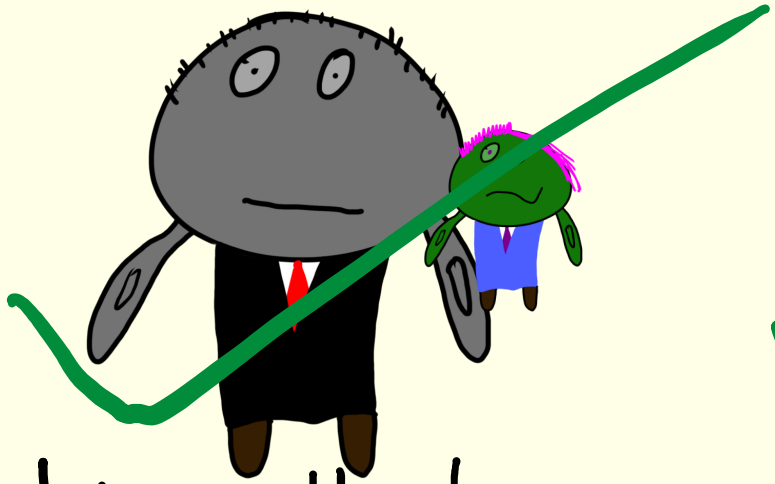
happy?

Main Philosophical Question

Are there schemes which make



Dictatoria



Warrantland

Prioritopia

happy?

Results

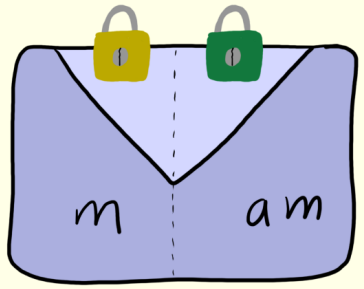
Anamorphic Resistant Encryption

(Gen, Enc, Dec) such that no non-trivial anamorphic instantiation exists.

Anamorphic Resistant Encryption

(Gen, Enc, Dec) such that no non-trivial anamorphic instantiation exists.

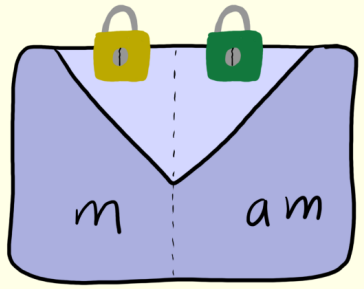
ALL (AGen, AEnc, ADec) have $|am| = O(\log \lambda)$



Anamorphic Resistant Encryption

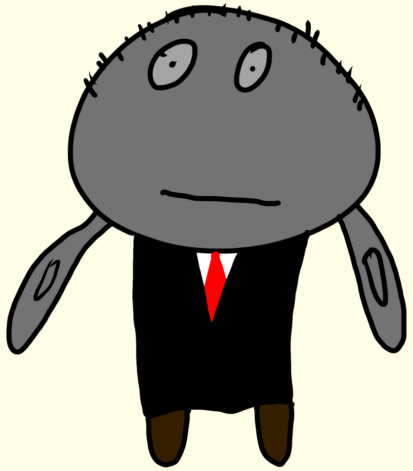
(Gen, Enc, Dec) such that no non-trivial anamorphic instantiation exists.

ALL (AGen, AEnc, ADec) have $|am| = O(\log \lambda)$

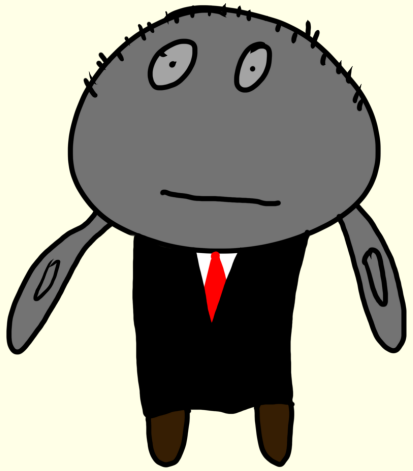


ARE

Anamorphic Resistant Encryption

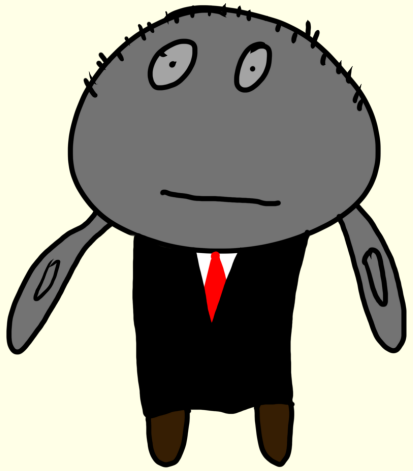
We allow  to control public parameters.

Anamorphic Resistant Encryption

We allow  to control public parameters.

(Init, Gen, Enc, Dec)

Anamorphic Resistant Encryption

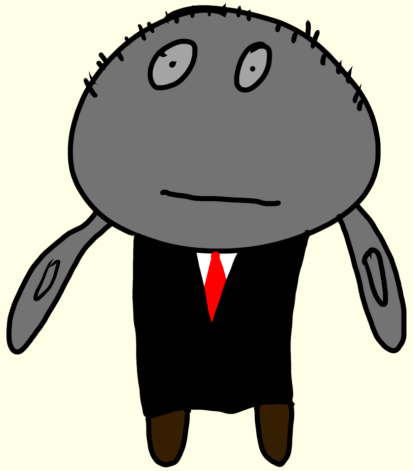
We allow  to control the

(Init → 

public parameters.

(Init, Gen, Enc, Dec)

Anamorphic Resistant Encryption

We allow  to control

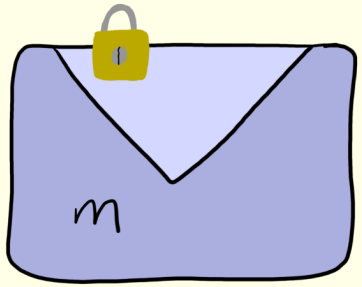
public parameters.

(Init, Gen, Enc, Dec)

Init → 

Enc(

, pp, m, dk)

→ 

Anamorphic Resistant Encryption

We allow  to control

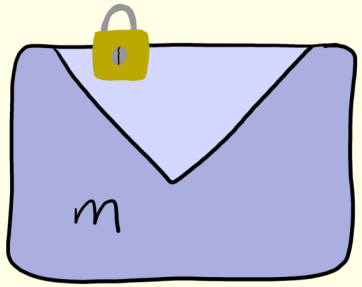
$(Init, Gen, Enc, Dec)$

public parameters.

$Init \rightarrow$ 

pp dk

$Enc($ 

$, m) \rightarrow$ 

Dictatoria

Dictatoria

ARE w/

universal backdoor.

Dictatoria

ARE w/ universal backdoor.

Dictator can read all messages & detect anamorphism <u>without</u> secret-key access

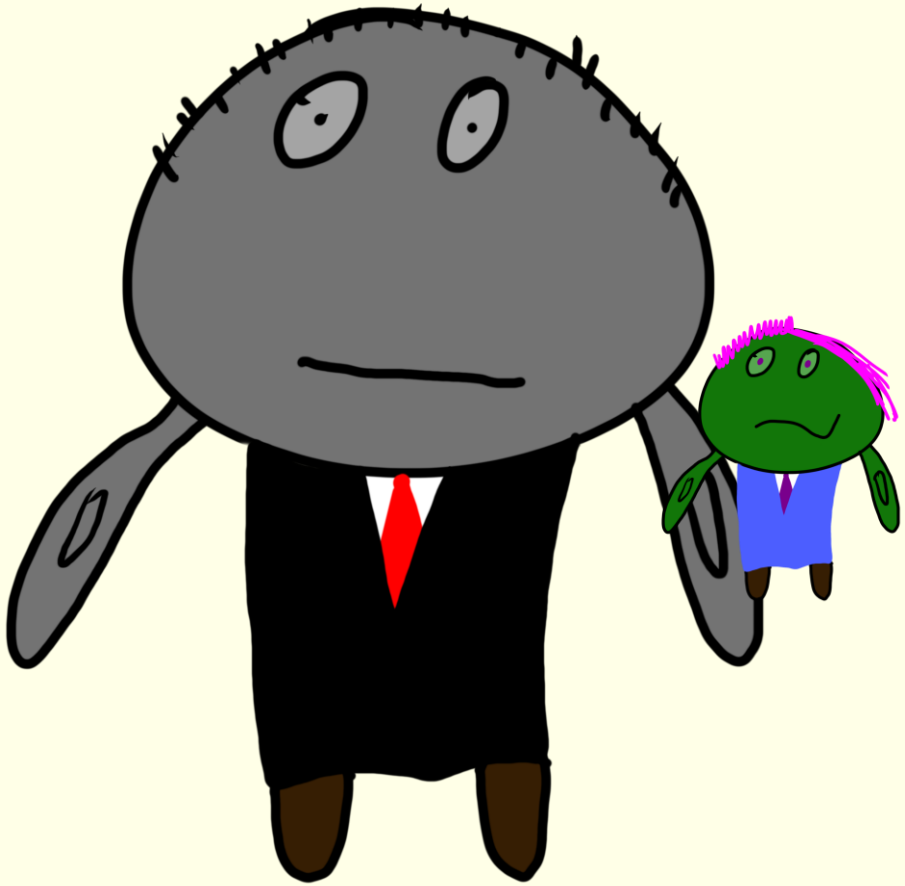
Dictatoria

ARE w/ in ROM

universal backdoor.

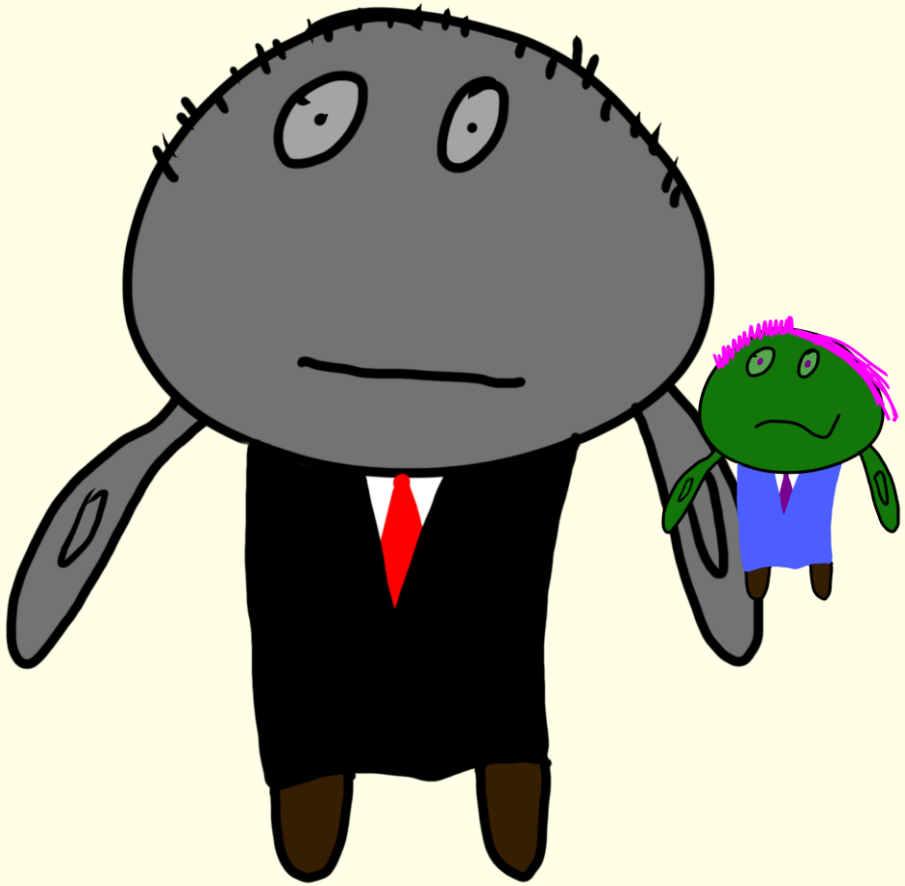
Dictator can read all messages & detect anamorphism without secret-key access

Warrantland



Warrantland

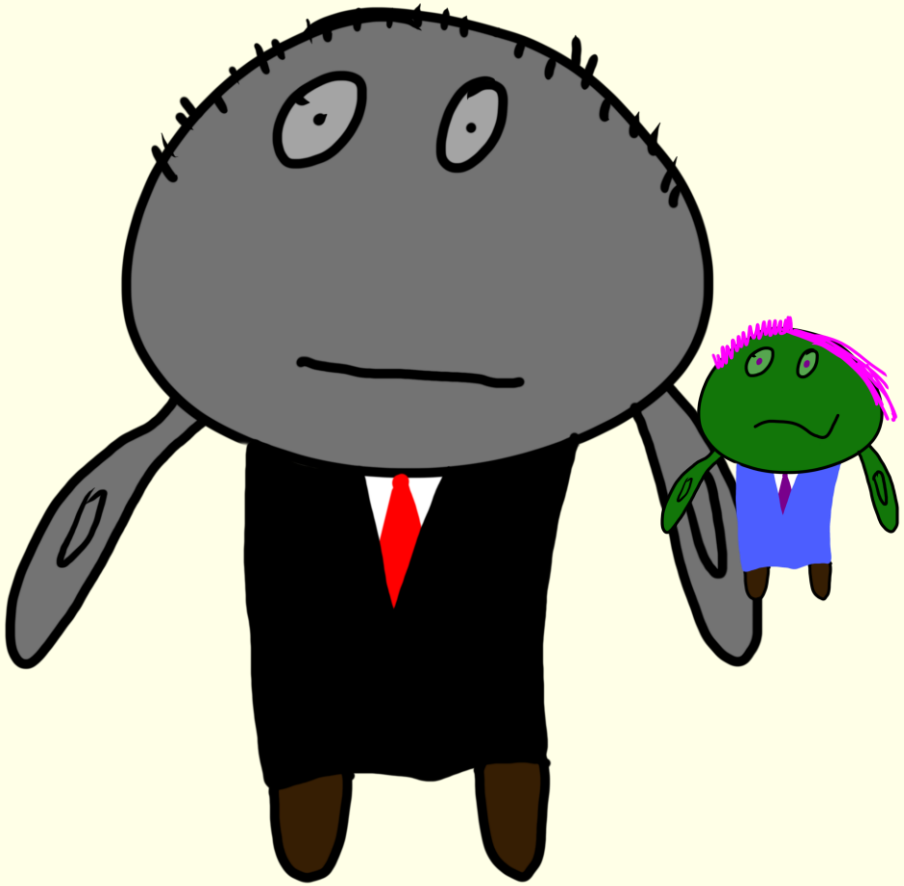
ARE which needs secret key access.



Warrantland

ARE which needs secret key access.

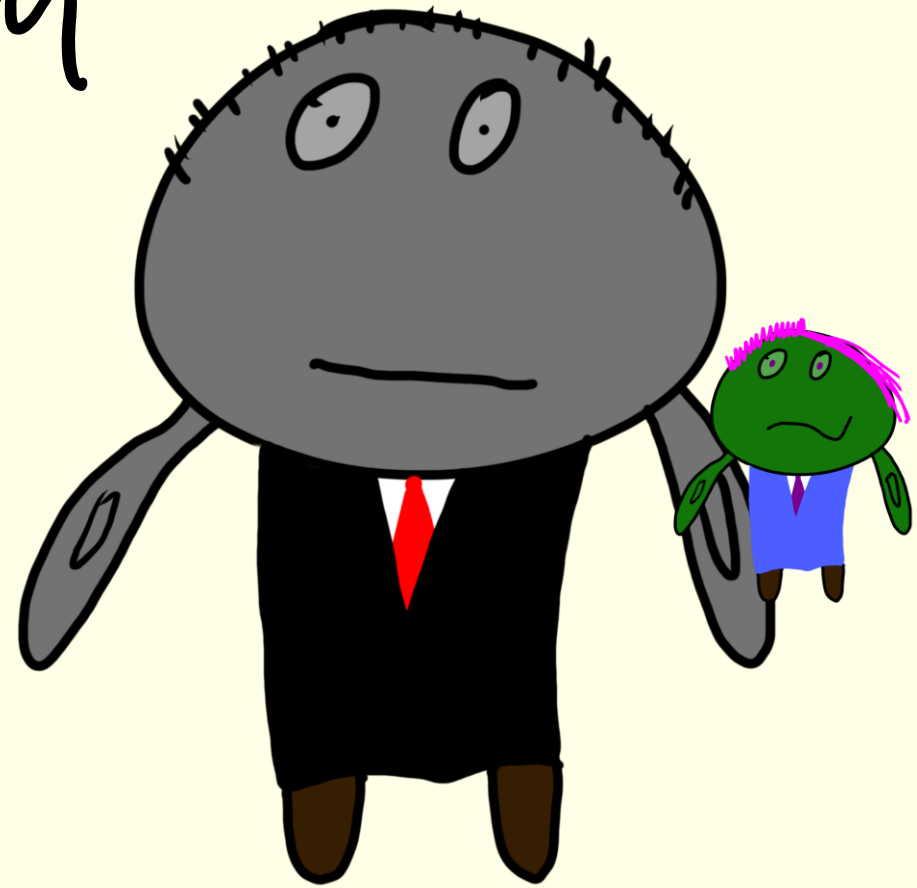
If  hidden, secure against govt. w/ 



Warrantland in ROM

ARE which needs secret key access.

If  hidden, secure against govt. w/ 



Privatopia

New notion - unforgeability

Unforgeability

Prior work [BGH+24]

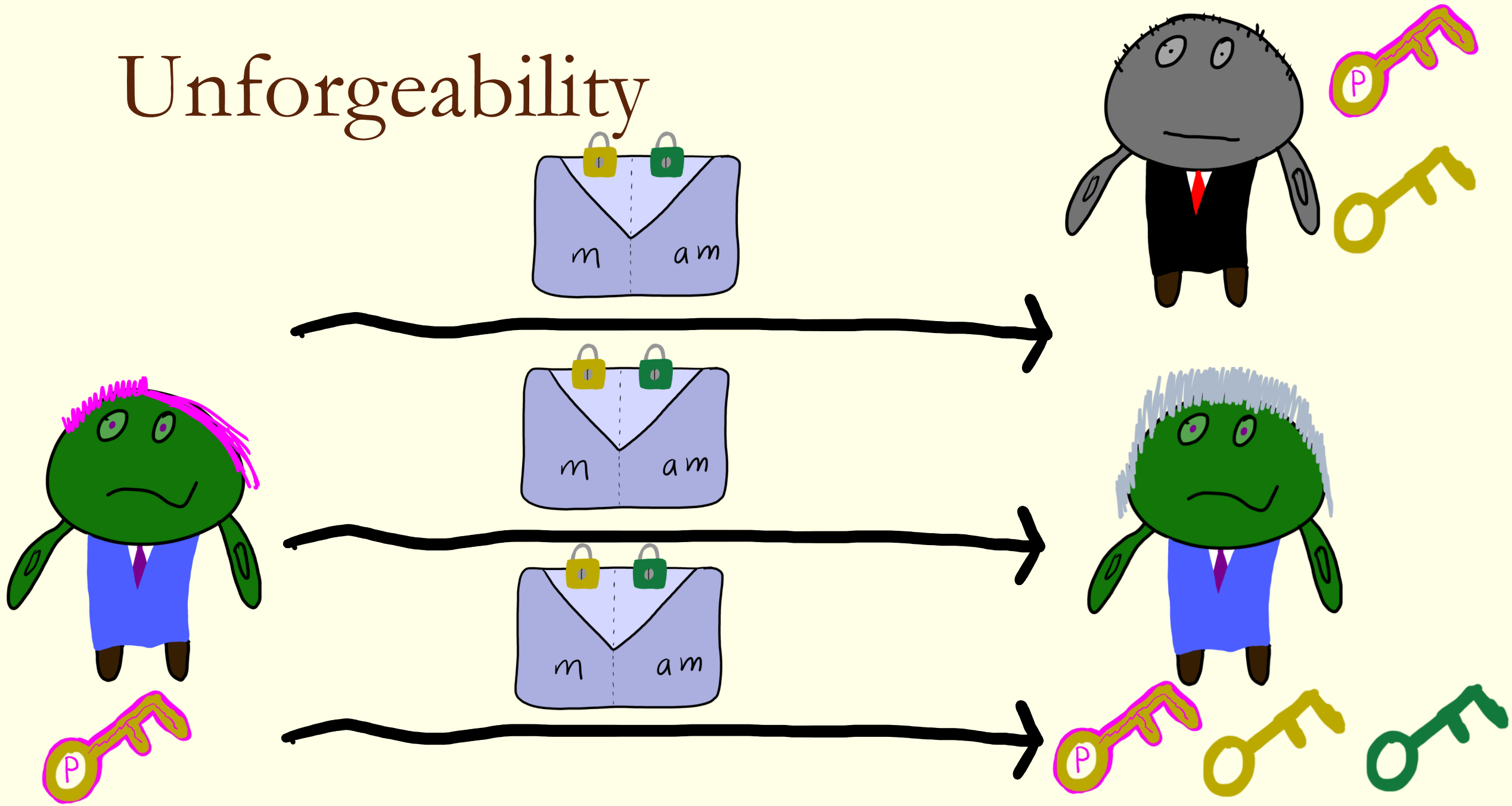
Robustness

Unforgeability

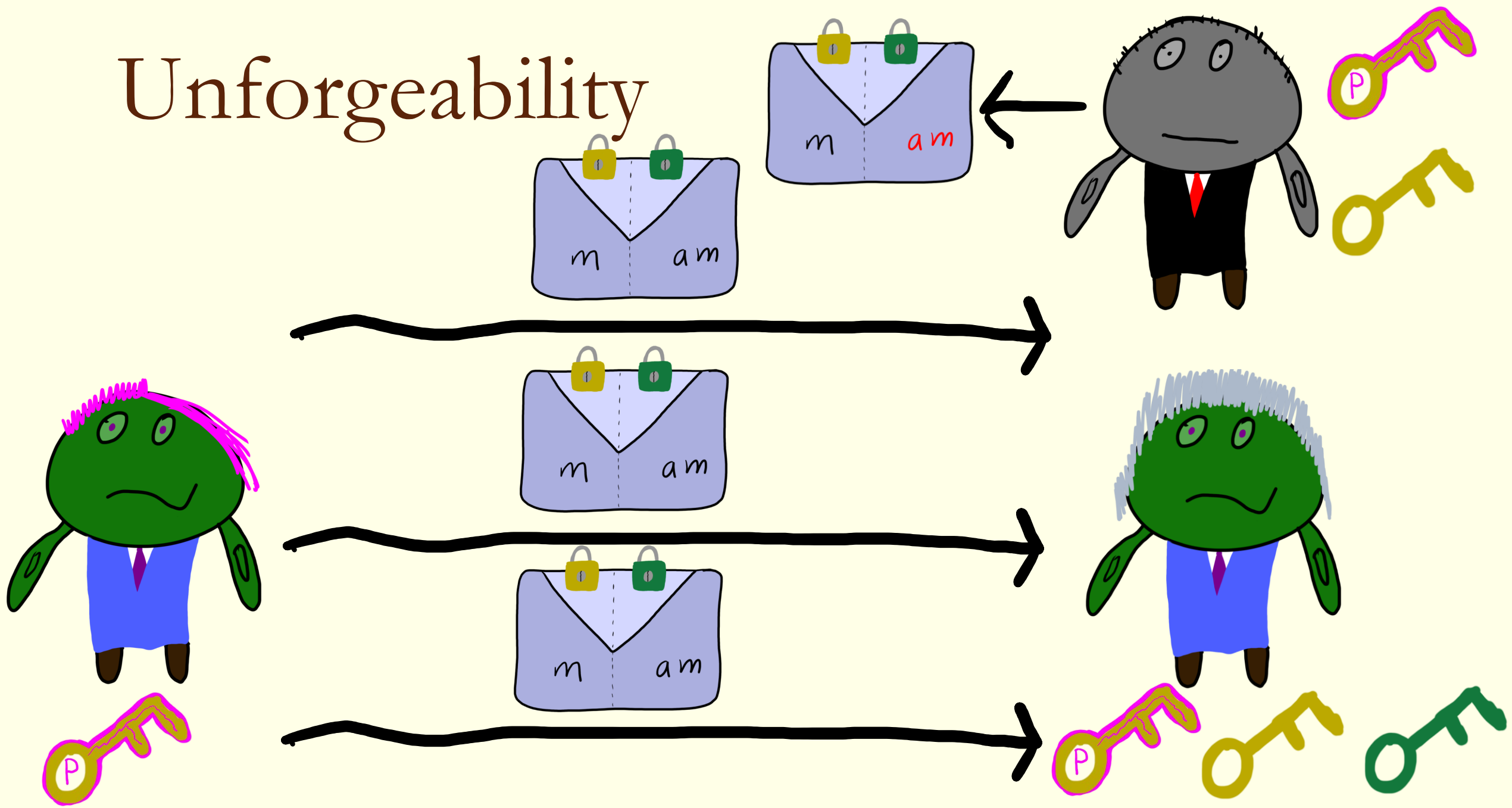
Prior work [BGH+24]

Robustness

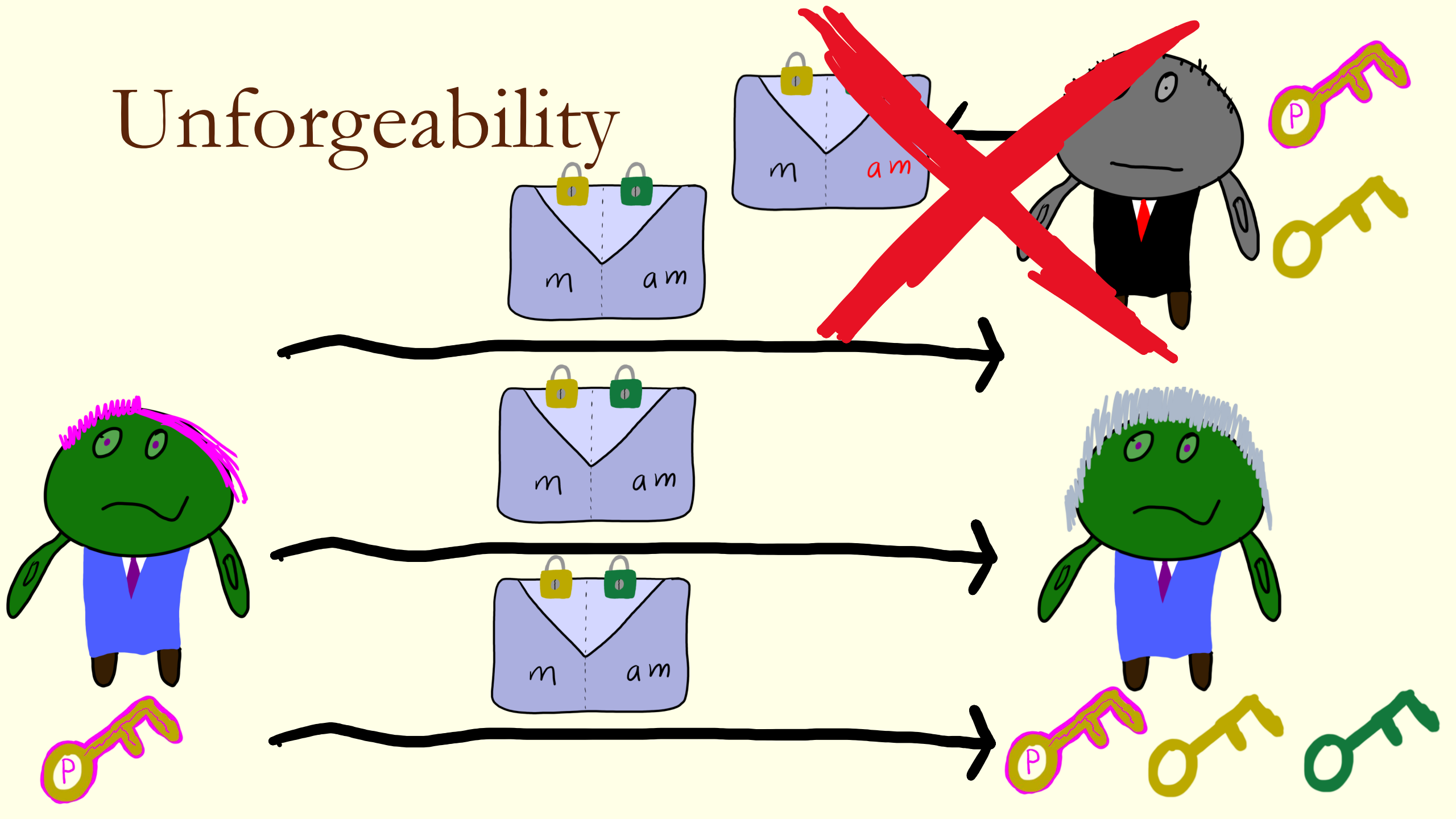
Unforgeability



Unforgeability



Unforgeability



Privatopia

New notion - unforgeability

Privatopia

New notion - unforgeability

Generic const. for already anamorphic schemes.

Privatopia

New notion - unforgeability

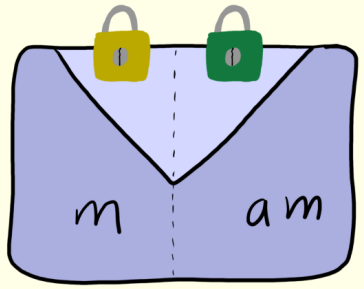
Generic const. for already anamorphic schemes.

Stronger const. for randomness-recoverable, El-Gamal, Naor-Yung

Anamorphic Resistant Encryption

(Gen, Enc, Dec) such that no non-trivial anamorphic instantiation exists.

ALL (AGen, AEnc, ADec) have $|am| = O(\log \lambda)$

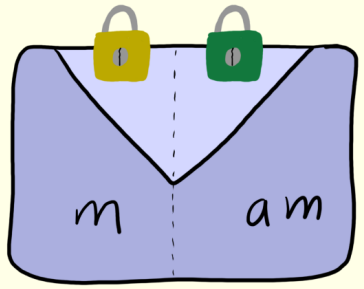


ARE

Anamorphic Resistant Encryption

(Gen, Enc, Dec) such that no non-trivial anamorphic instantiation exists.

ALL (AGen, AEnc, ADec) have $|am| = O(\log \lambda)$



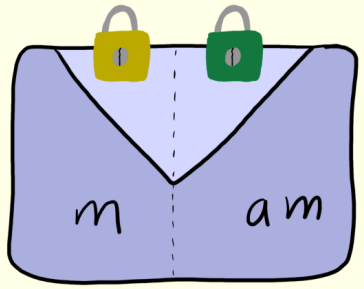
NOT robust

ARE

Anamorphic Resistant Encryption

(Gen, Enc, Dec) such that no non-trivial anamorphic instantiation exists.

ALL (AGen, AEnc, ADec)



have $|am| = O(\log \lambda)$

NOT robust (even 1-bit messages)

ARE

Anamorphic Encryption

Hypothesis: linear bandwidth anamorphic instantiations are always possible.

At least for CCA schemes? [PPYZ24]

Anamorphic Encryption

Hypothesis: linear bandwidth anamorphic instantiations are always possible.

At least for CCA schemes? [PPYZ24]

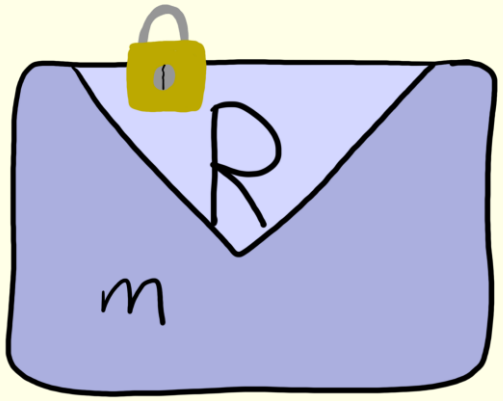
Answer: NO - our schemes are CCA

ARE

Construction

Anamorphic Resistant Encryption

Key plan: "only way to bias

$Enc(pk, m; R) \rightarrow$ 

is to do rejection sampling"

Anamorphic Resistant Encryption

Idea #1: Replace R w/ RO(R)

$$Enc(pk, m; R) = E(pk, m; RO(R))$$

Anamorphic Resistant Encryption

Idea #1: Replace R w/ RO(R)

(Only way to bias RO(R) is rej. sampling)

$$Enc(pk, m; R) = E(pk, m; RO(R))$$

Anamorphic Resistant Encryption

If

$AEnc(m, am) =$

Anamorphic Resistant Encryption

If

$AEnc(m, am) =$ 

then $RO(R_{am})$ can only provide

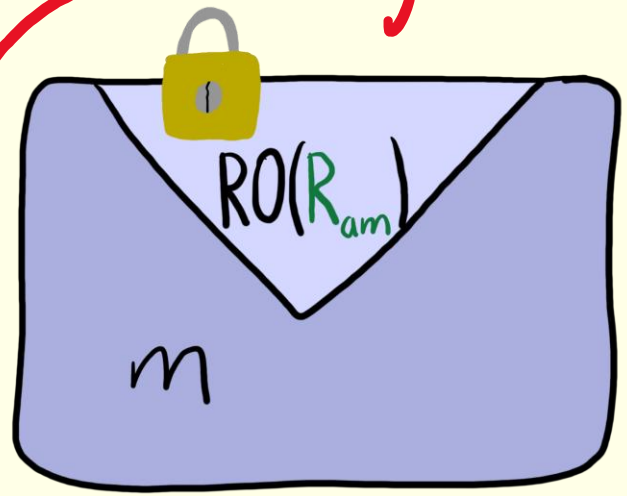
$\approx \log \lambda$ bits of info on am

Anamorphic Resistant Encryption

If

$AEnc(m, am) =$

might not be the case.

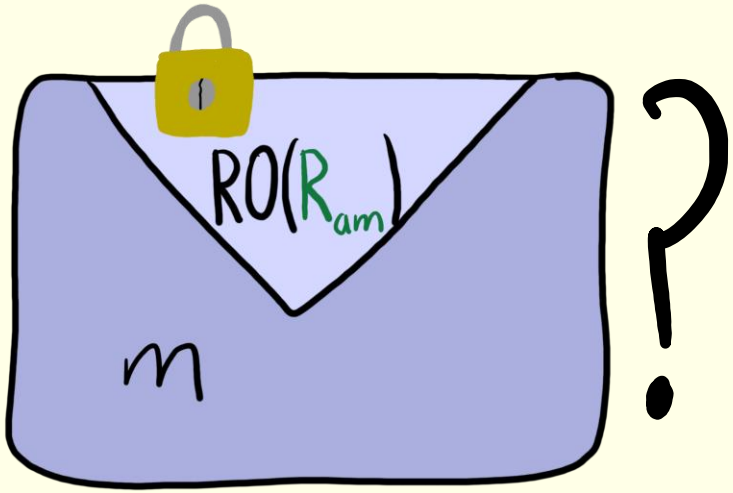


then $RO(R_{am})$ can only provide

$\approx \log \lambda$ bits of info on am

Anamorphic Resistant Encryption

Key question: how to force

$AEnc(m, am) =$ 

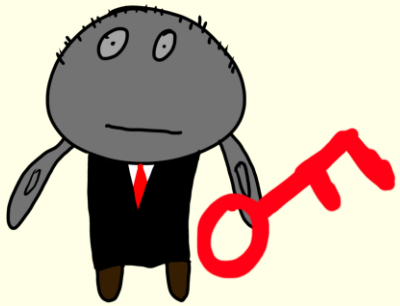
?

Anamorphic Resistant Encryption

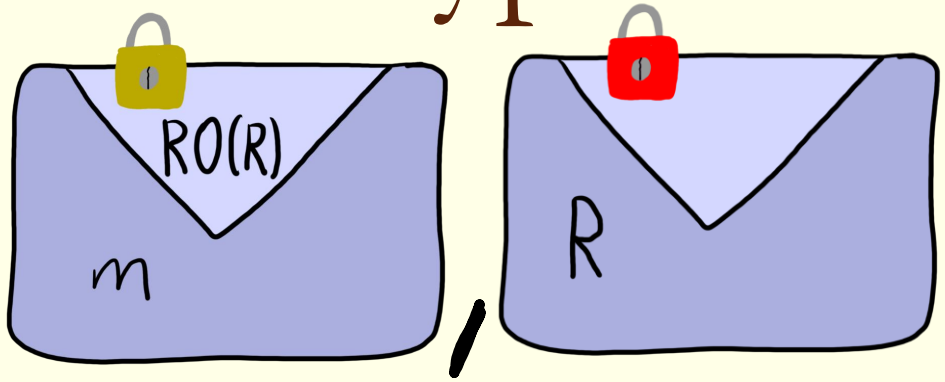
Key question: how to force

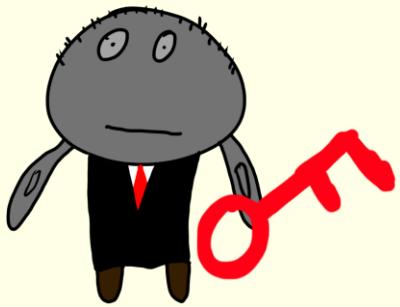
$AEnc(m, am) =$ 

?

Idea 2: tell  R

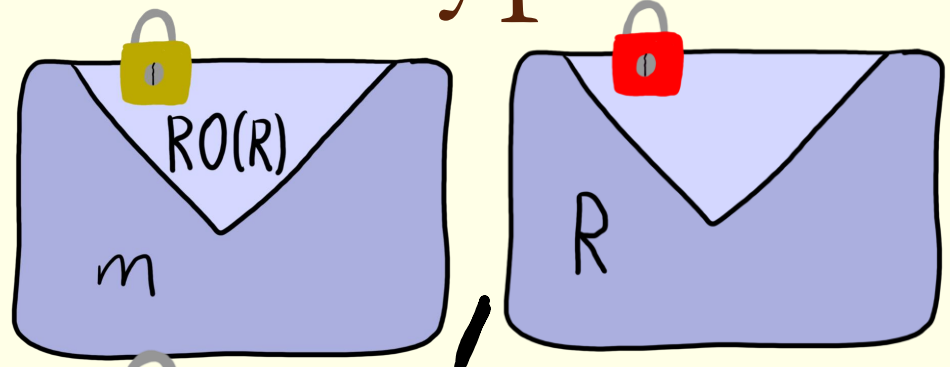
Anamorphic Resistant Encryption

$Enc(pp, pk, m; R) =$ 

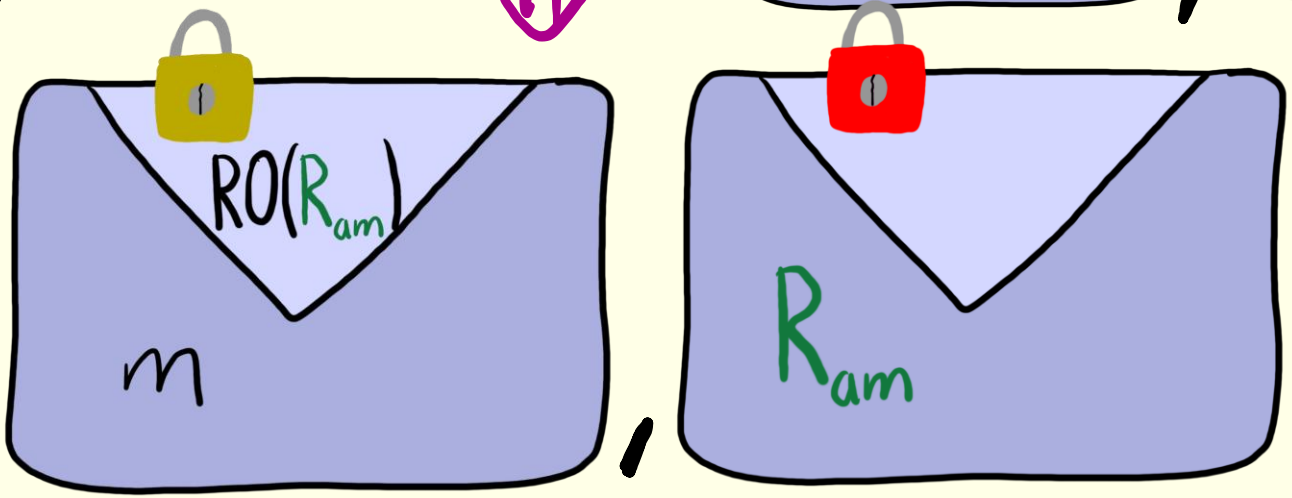
Idea 2: tell  R

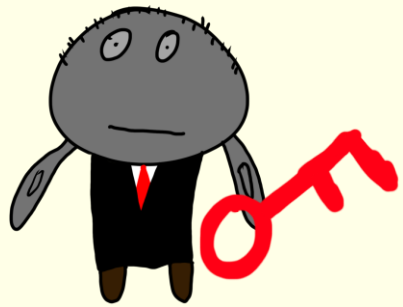
Anamorphic Resistant Encryption

$Enc(pp, pk, m; R) =$



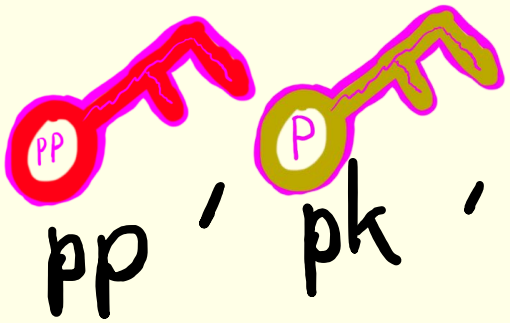
$AEnc(m, am) =$

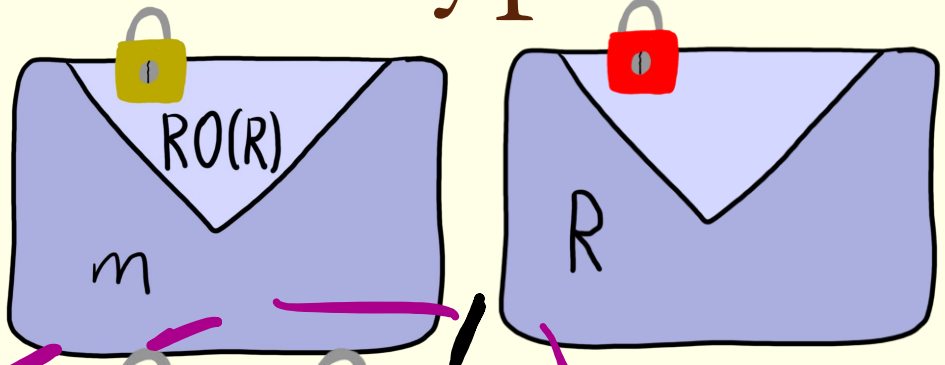


Idea 2: tell  R

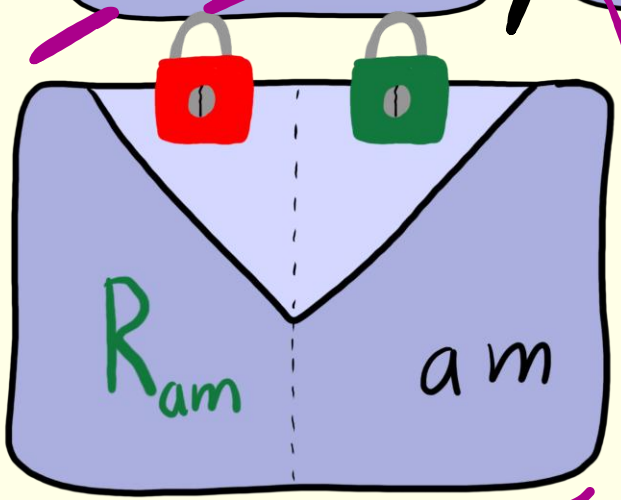
Anamorphic Resistant Encryption

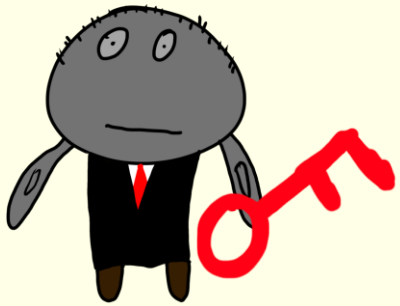
$Enc(pp, pk, m; R) =$





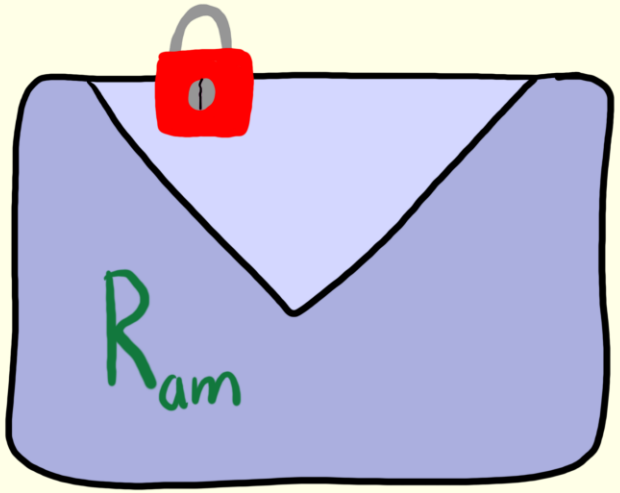
$AEnc(m, am) =$



Idea 2: tell  R

(can contain covert messages itself

Deterministic Encryption

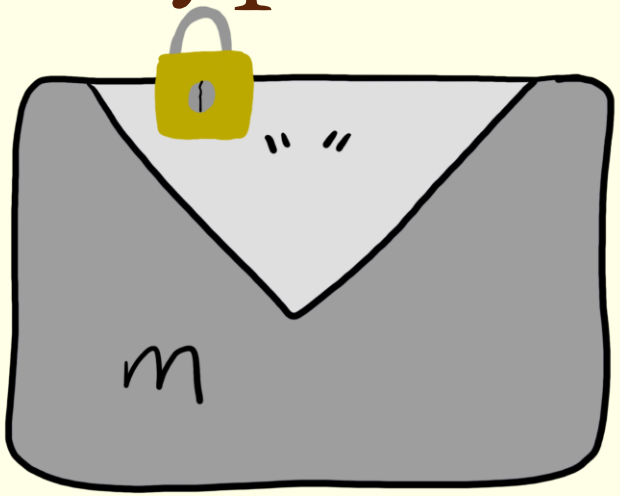
If 

is not random,

can't contain any anamorphic message.

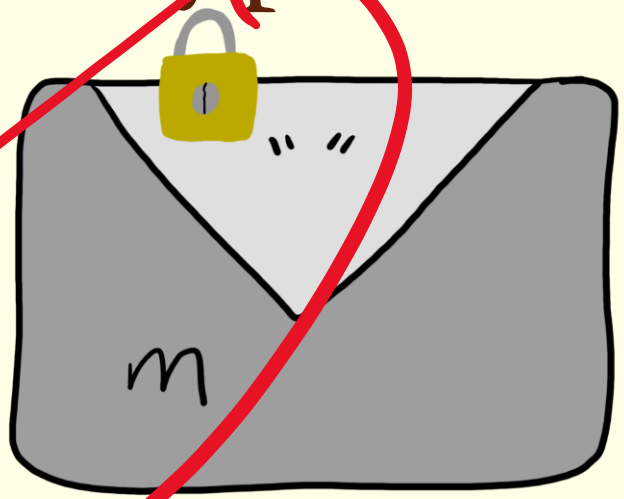
Deterministic Encryption

$DEnc(pk, m) =$



Deterministic Encryption

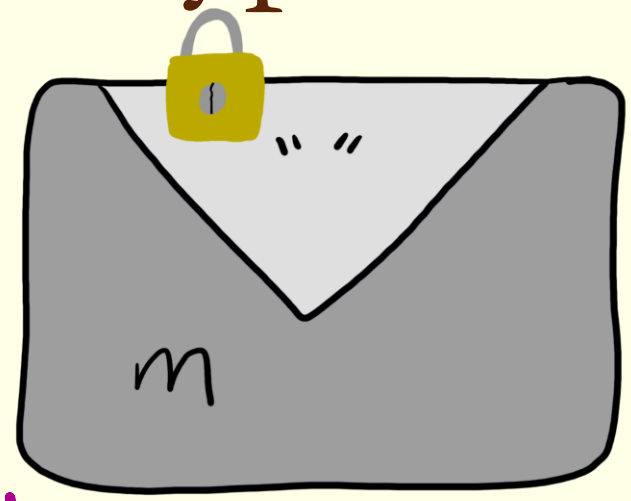
$DEnc(pk, m) =$



usually impossible

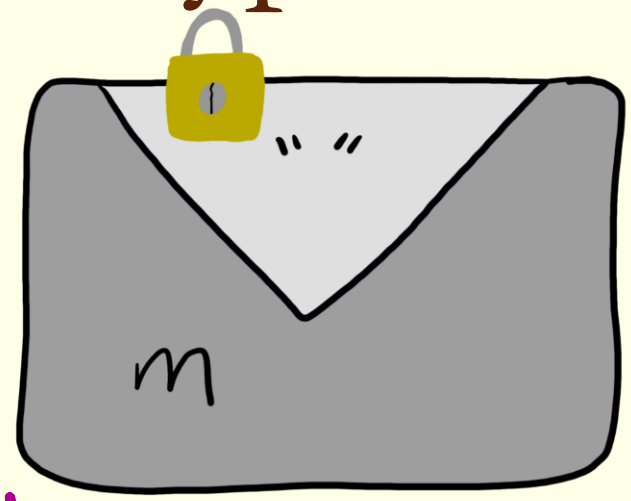
Deterministic Encryption

$DEnc(pk, m) =$



Possible in restricted circumstances

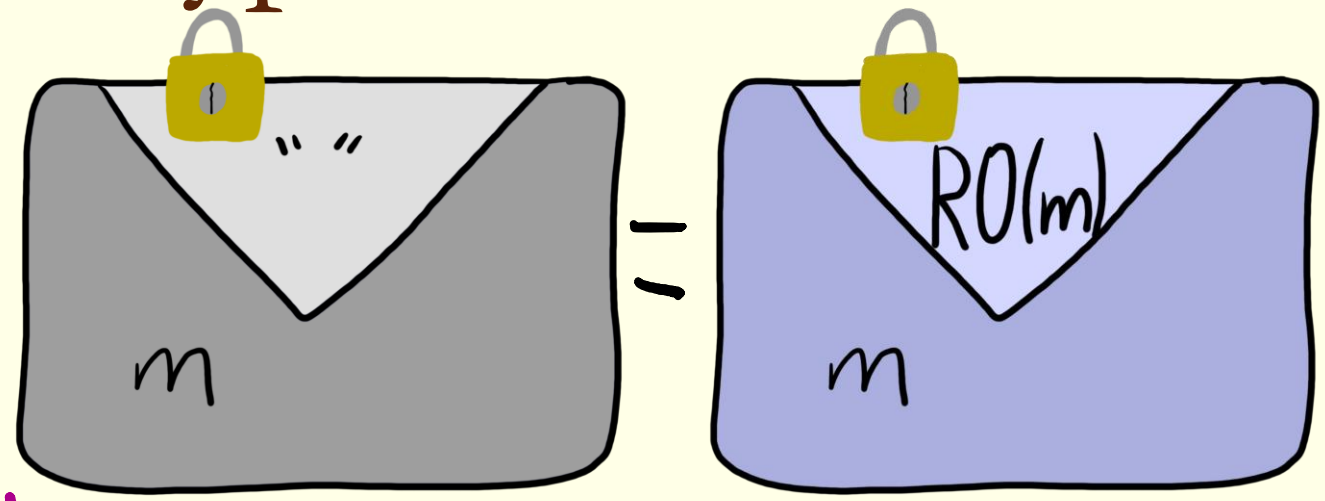
Deterministic Encryption

$DEnc(pk, m) =$ 

Possible in restricted circumstances

Including this one! (with work)

Deterministic Encryption

$DEnc(pk, m) =$ 

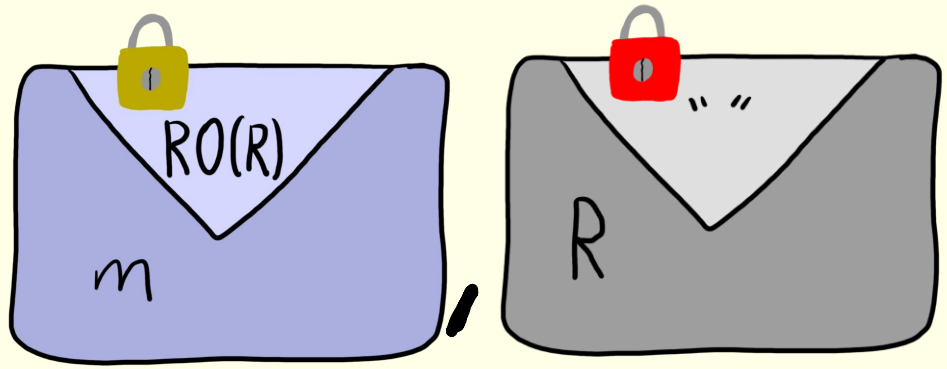
Possible in restricted circumstances

Including this one! (with work)

- Encrypt-with-Hash
- [BBO 06]

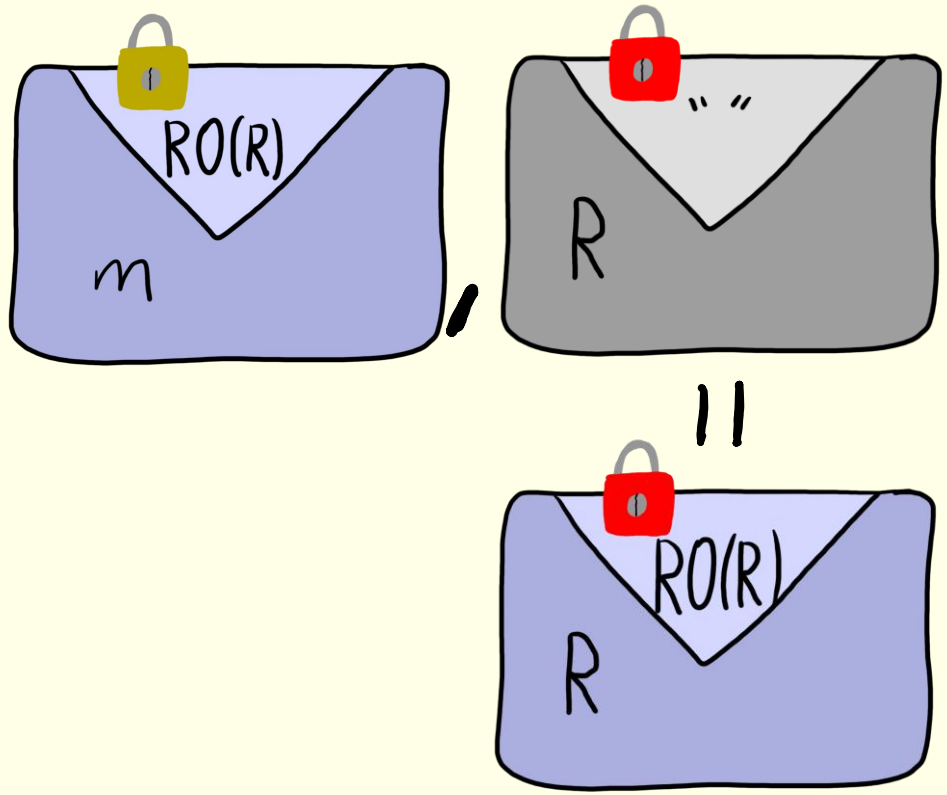
Final Construction

$Enc(pp, pk, m; R) =$



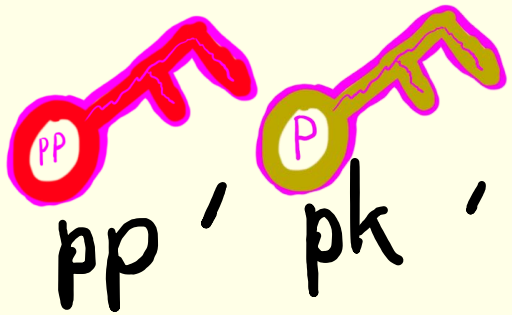
Final Construction

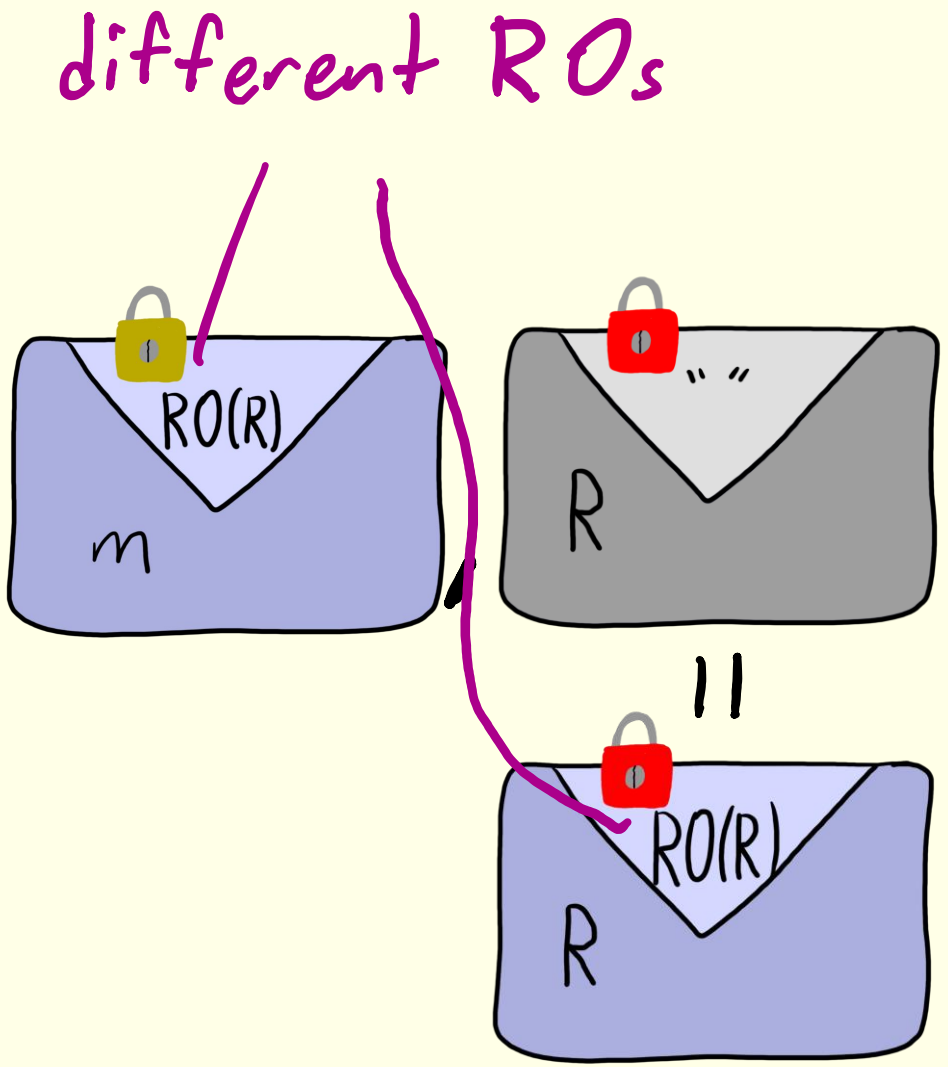
$Enc(pp, pk, m; R) =$



Final Construction

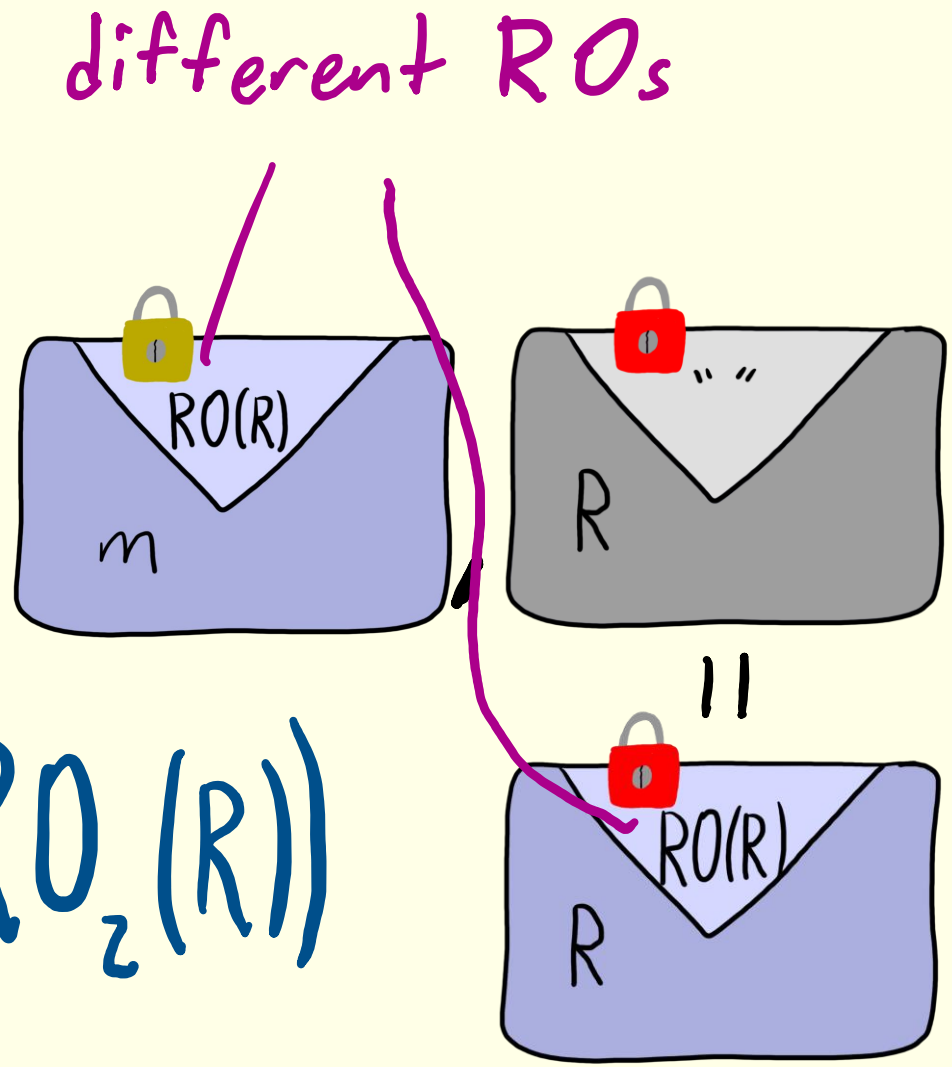
$Enc(pp, pk, m; R) =$





Final Construction

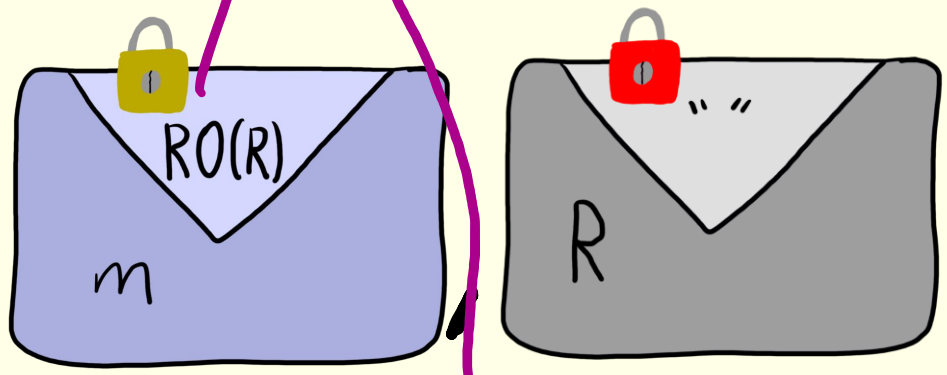
$$Enc(pp, pk, m; R) =$$



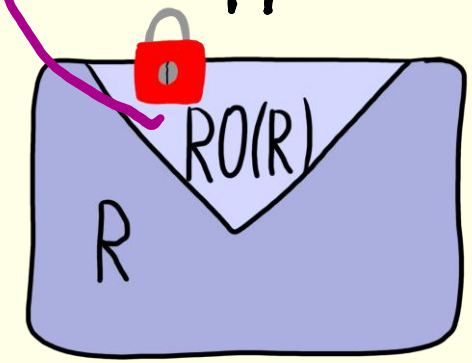
$$= E(pk, m; RO_1(R)), E(pp, R; RO_2(R))$$

Final Construction

$$Enc(pp, pk, m; R) =$$



$$\| \quad E(pk, m; RO_1(R)), E(pp, R; RO_2(R))$$

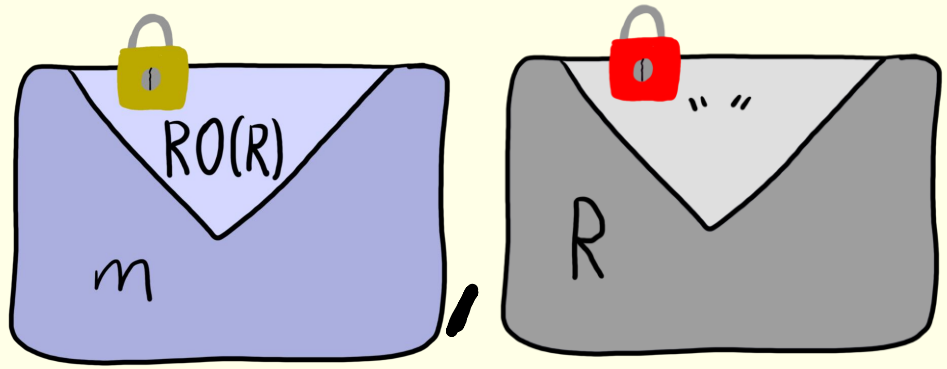


different ROs

can be different

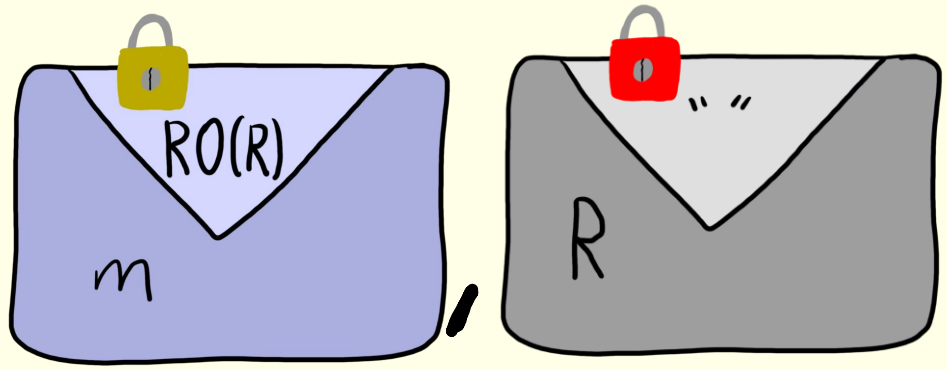
Final Construction

$Enc(pp, pk, m; R) =$



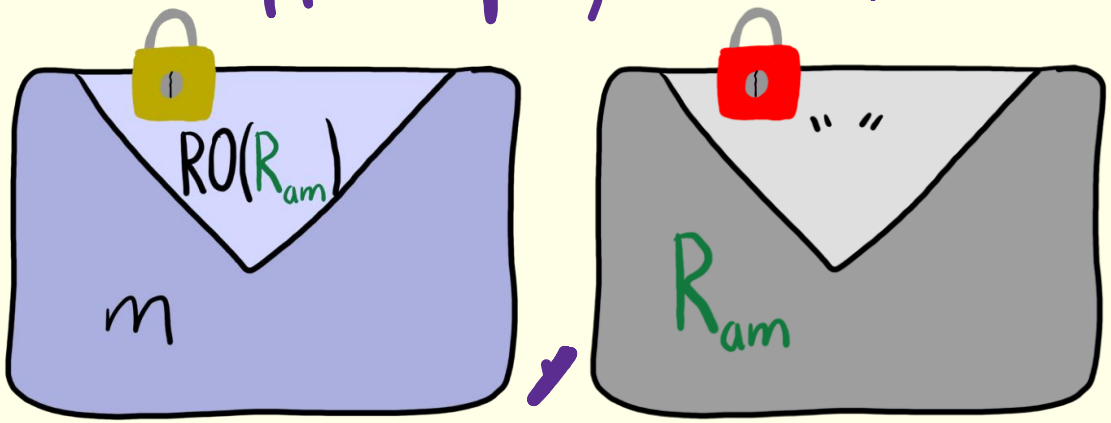
Final Construction

$Enc(pp, pk, m; R) =$



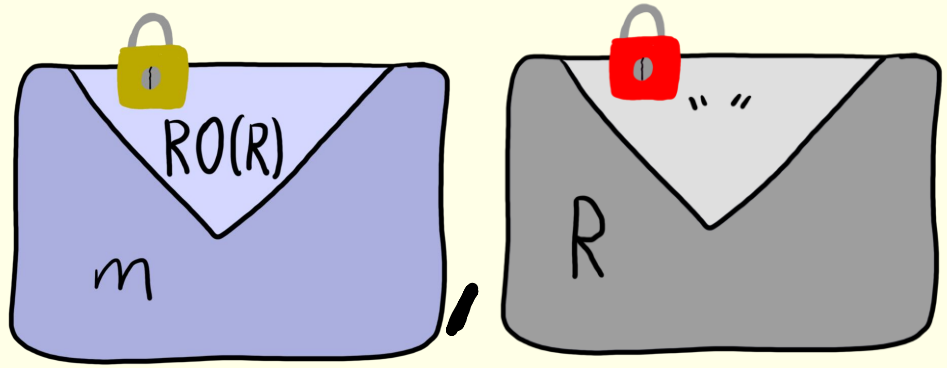
$AEnc(pp, pk, ak, m, am)$

$=$



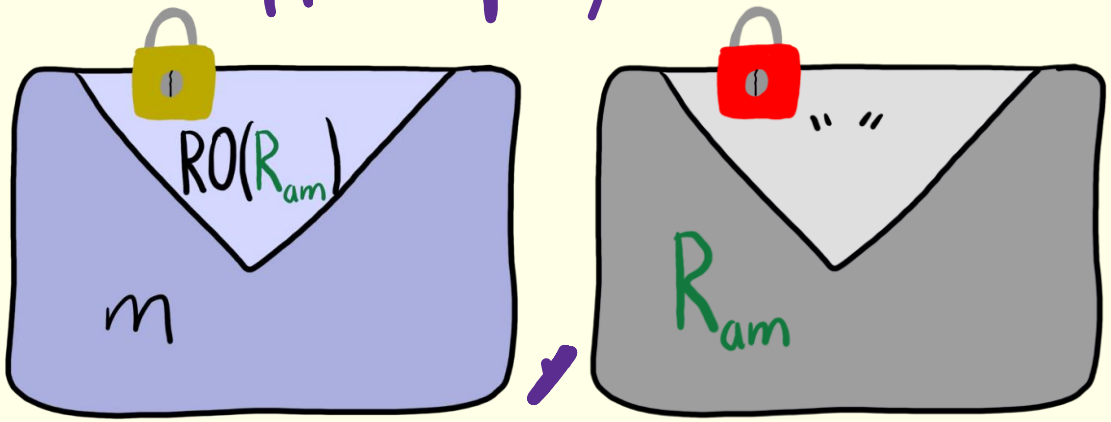
Final Construction

$Enc(pp, pk, m; R) =$



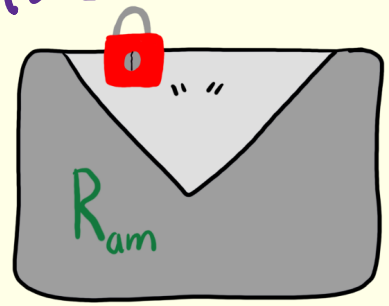
$AEnc(pp, pk, ak, m, am)$

$=$



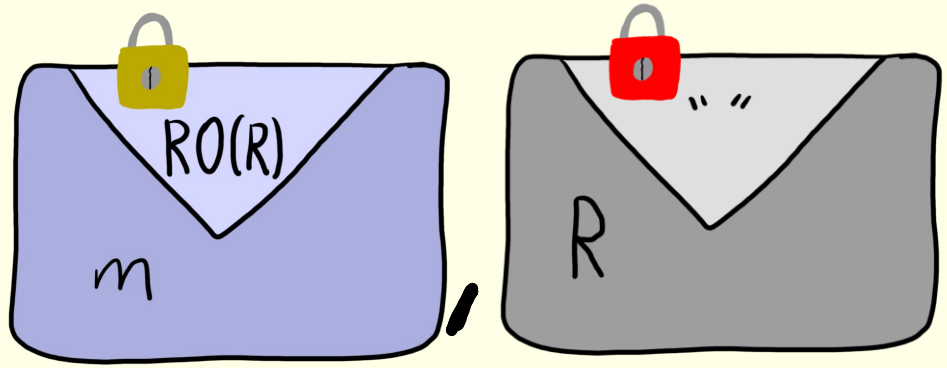
only info on am given to ADec is

$RO(R_{am}) +$

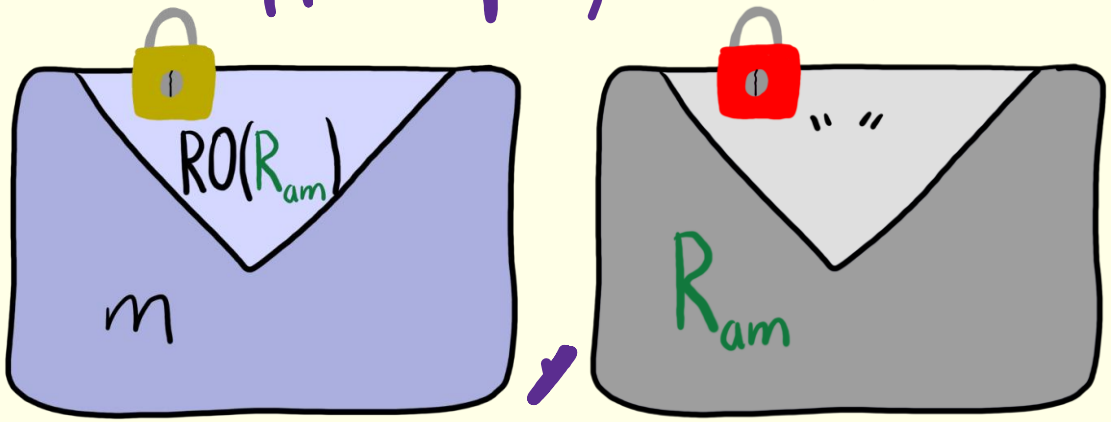


Final Construction

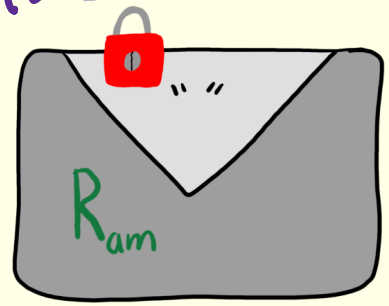
$Enc(pp, pk, m; R) =$



$AEnc(pp, pk, ak, m, am) =$



only info on am given to ADec is

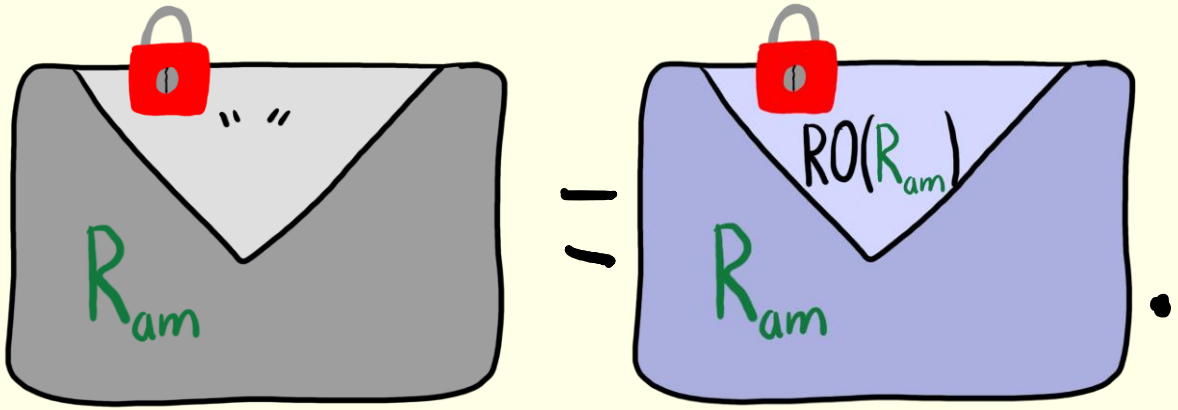
$RO(R_{am})$ + 

log λ bits of info!

Proof Sketch

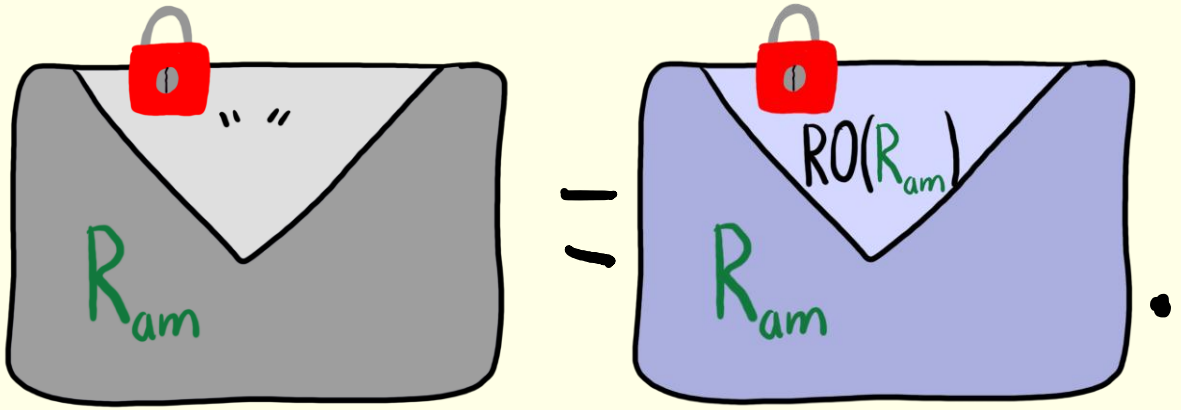
Interesting technical lemma

$RO(R_{am})$ reveals $\log \lambda$ bits of info on am. Need something stronger for



Interesting technical lemma

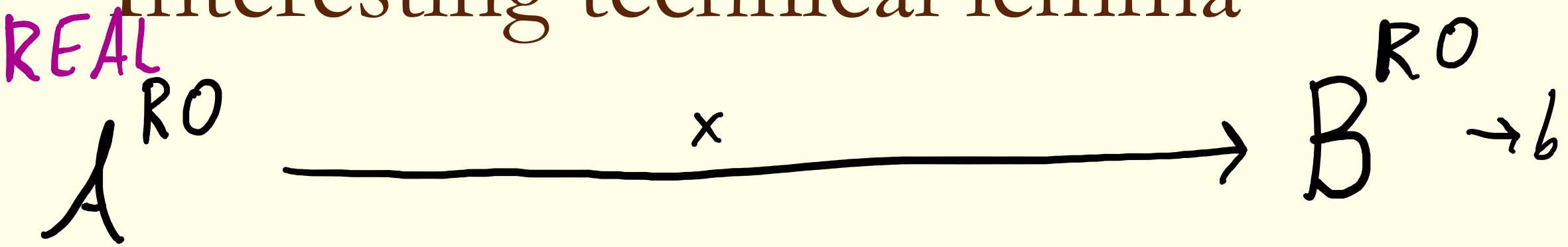
$RO(R_{am})$ reveals $\log \lambda$ bits of info on am. Need something stronger for



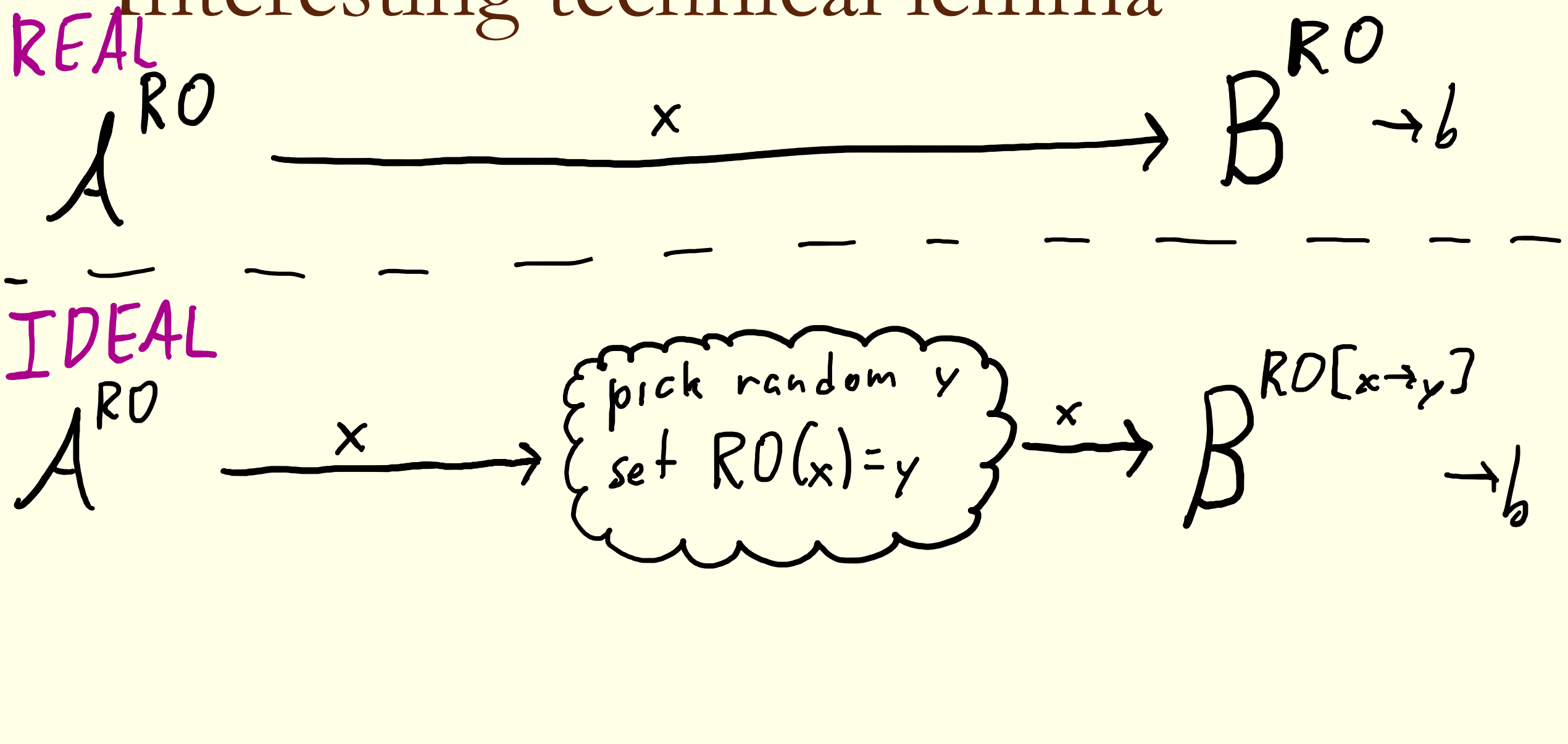
Intuition:

$RO(R_{am})$ should look "fairly random" to A Dec

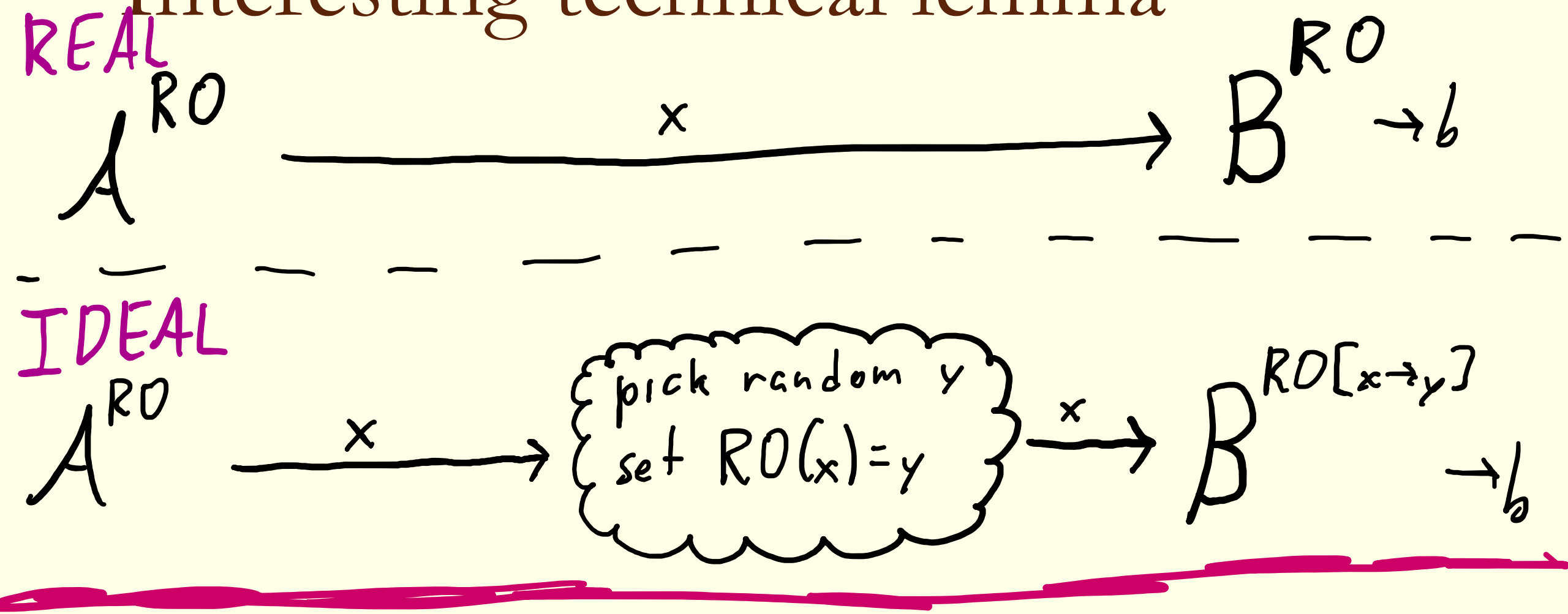
Interesting technical lemma



Interesting technical lemma

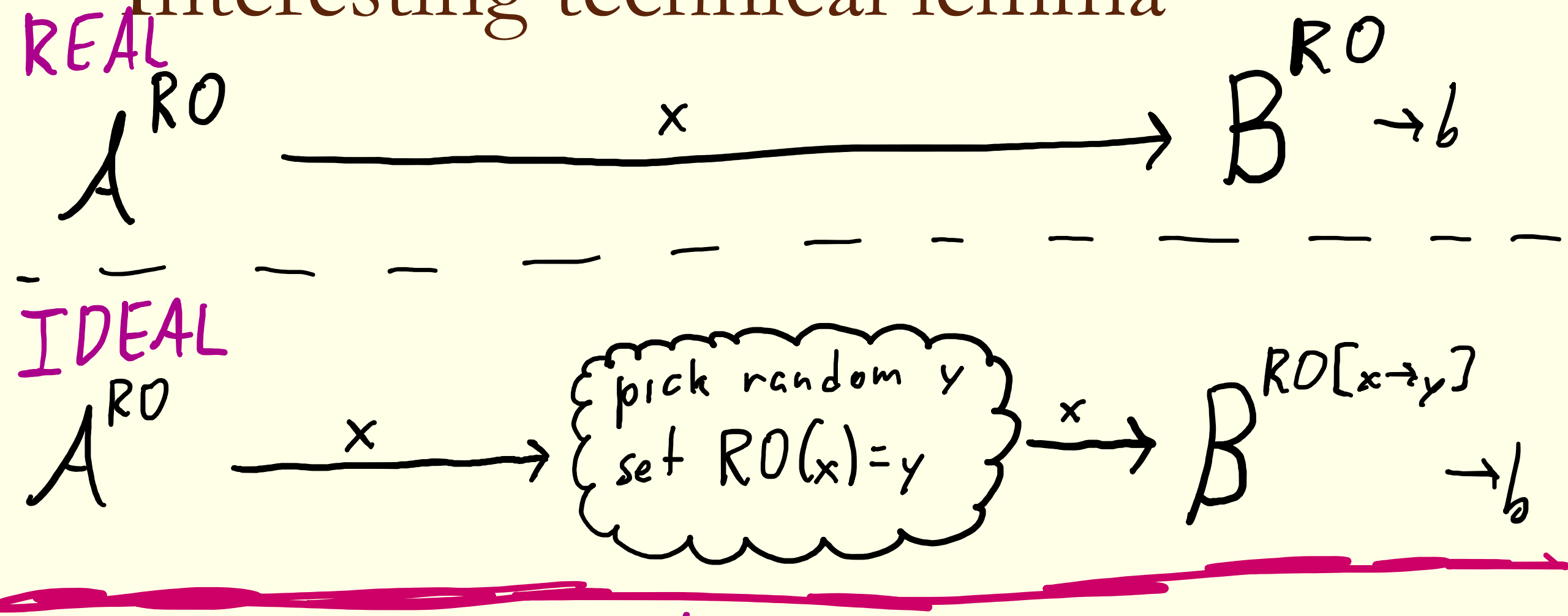


Interesting technical lemma



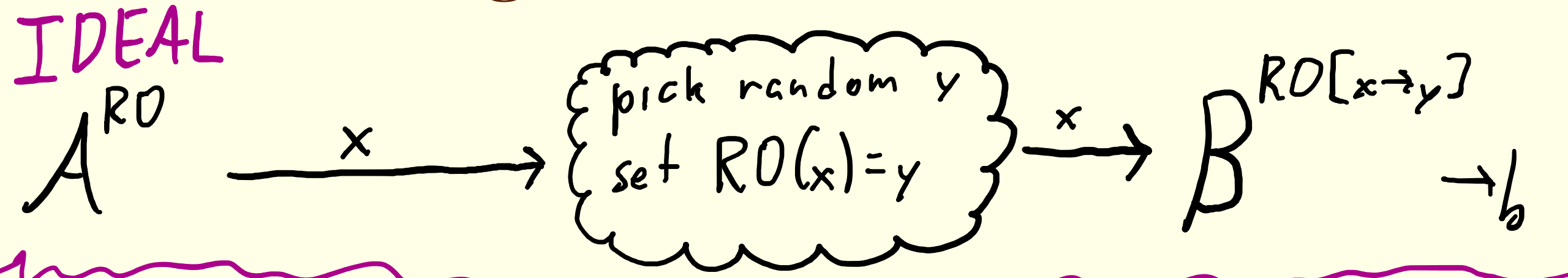
B should not ALWAYS be able to tell

Interesting technical lemma



$$Pr[IDEAL \to 1] \geq \frac{1}{\#queries+1} \cdot Pr[REAL \to 1]$$

Interesting technical lemma

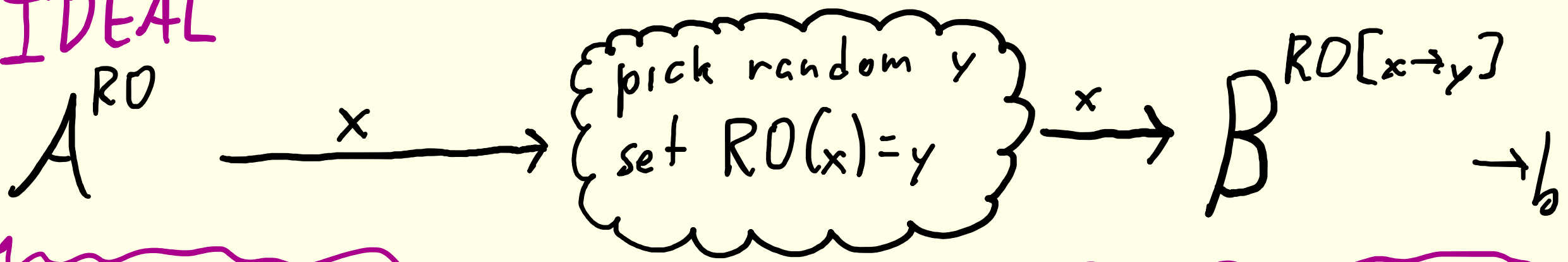


Set $S=\{y : A^{RO[x\to y]} \to x\}$. Conditioned on $y \in S$

REAL=IDEAL

Interesting technical lemma

IDEAL



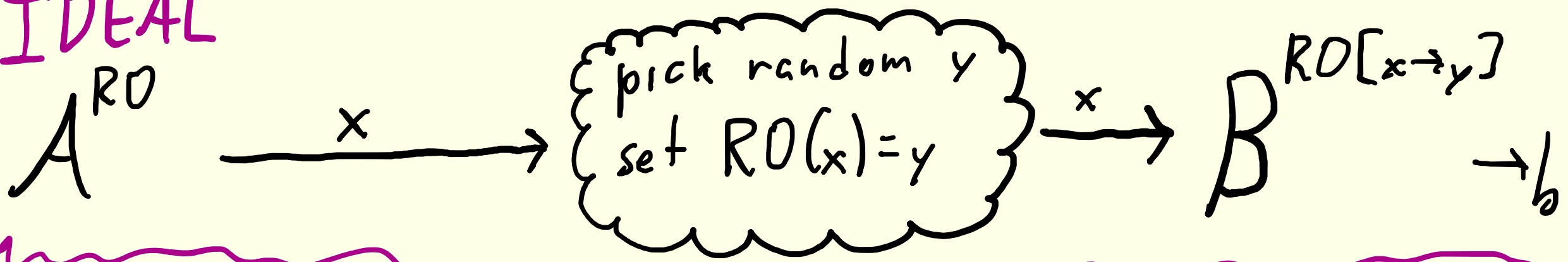
Set $S=\{y : A^{RO[x\to y]} \to x\}$. Conditioned on $y \in S$

REAL=IDEAL

What is $\Pr_y[y \in S]$?

Interesting technical lemma

IDEAL



Set $S=\{y : A^{RO[x\to y]} \to x\}$. Conditioned on $y \in S$

REAL=IDEAL

What is $\Pr_y[y \in S]$? $\frac{1}{\#\text{queries}+1}$

Interesting technical lemma

Set $S = \{y : A^{RO[x \to y]} \to x\}$. Conditioned on $y \in S$

REAL = IDEAL

What is $\Pr_y[y \in S]$? $\frac{1}{\#\text{queries}+1}$

Interesting technical lemma

Set $S = \{y : A^{RO[x \to y]} \to x\}$. Conditioned on $y \in S$

REAL = IDEAL

What is $\Pr_y[y \in S]$? $\frac{1}{\#\text{queries}+1}$

RO can be described by

Interesting technical lemma

Set $S = \{y : A^{RO[x \to y]} \to x\}$. Conditioned on $y \in S$

REAL = IDEAL

What is $\Pr_y[y \in S]$? $\frac{1}{\#\text{queries}+1}$

RO can be described by

1. $H(RO)$ w/ row for x missing

Interesting technical lemma

Set $S = \{y : A^{RO[x \to y]} \to x\}$. Conditioned on $y \in S$

REAL = IDEAL

What is $\Pr_y[y \in S]$? $\frac{1}{\#\text{queries}+1}$

RO can be described by
1. $H(RO)$ w/ row for x missing
2. index of A's query $RO(x)$

Interesting technical lemma

Set $S = \{y : A^{RO[x \to y]} \to x\}$. Conditioned on $y \in S$

REAL = IDEAL

What is $\Pr_y[y \in S]$? $\frac{1}{\#\text{queries}+1}$

RO can be described by

1. H(RO) w/ row for x missing
2. index of A's query RO(x)
3. index of RO(x) in S

Interesting technical lemma

Set $S = \{y : A^{RO[x \to y]} \to x\}$. Conditioned on $y \in S$

REAL = IDEAL

What is $\Pr_y[y \in S]$? $\frac{1}{\#\text{queries}+1}$

RO can be described by

1. $H(RO)$ w/ row for x missing
2. index of A's query $RO(x)$
3. index of $RO(x)$ in S

$\log(\#\text{queries}+1) + \log|S| \geq n$

Interesting technical lemma

Set $S = \{y : A^{RO[x \to y]} \to x\}$. Conditioned on $y \in S$

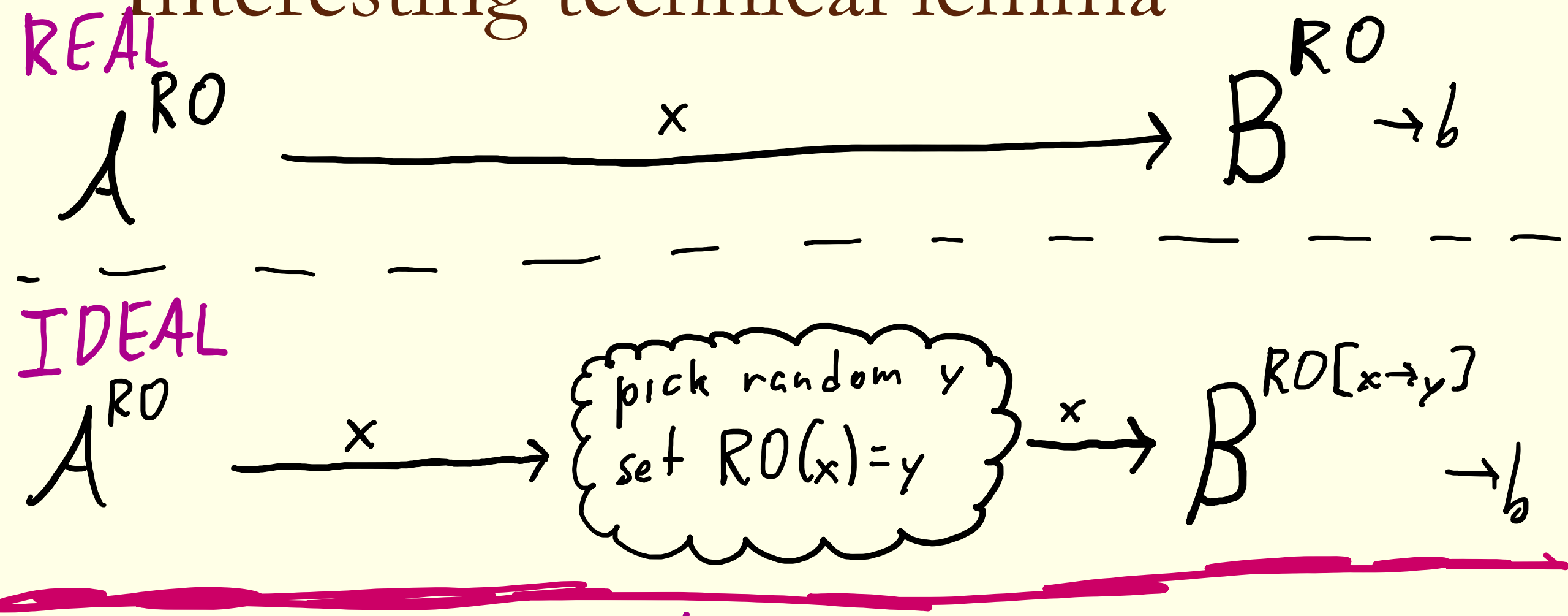
REAL = IDEAL

What is $\Pr_y[y \in S]$? $\frac{1}{\#\text{queries}+1}$

RO can be described by
1. $H(RO)$ w/ row for x missing
2. index of A's query $RO(x)$
3. index of $RO(x)$ in S

$\log(\#\text{queries}+1) + \log|S| \geq n \Rightarrow \frac{|S|}{2^n} \geq \frac{1}{\#\text{queries}+1}$

Interesting technical lemma



$$Pr[IDEAL \to 1] \geq \frac{1}{\#queries+1} \cdot Pr[REAL \to 1]$$

Application of lemma

Only info A_{Dec} gets is

$RO(R_{am})$ +

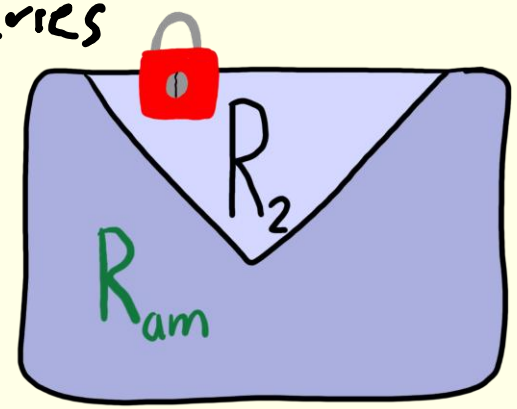
Application of lemma

Only info $ADec$ gets is

$RO(R_{am})$ +

$\approx_{\frac{1}{\#queries}}$

R_1 +

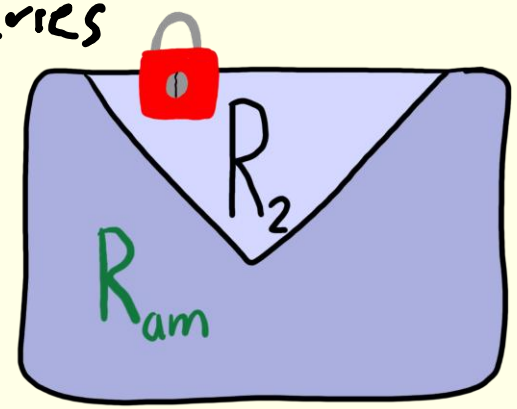


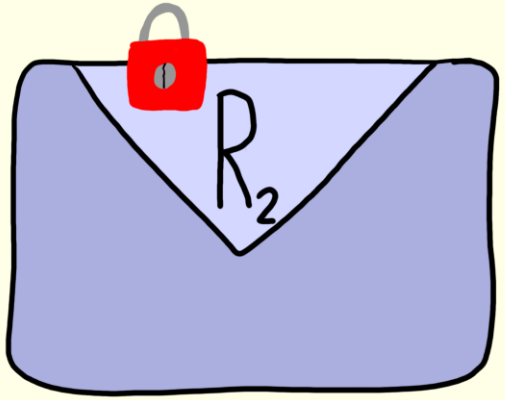
Application of lemma

Only info $\mathcal{ADec}$ gets is

$RO(R_{am})$ + 

$\approx_{\frac{1}{\#\text{queries}}}$

R_1 + 

$\approx$ R_1 + 

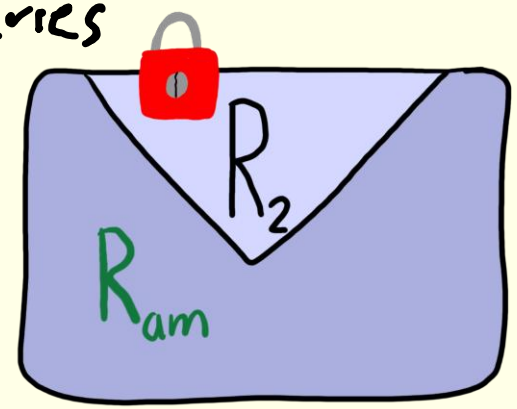
Application of lemma

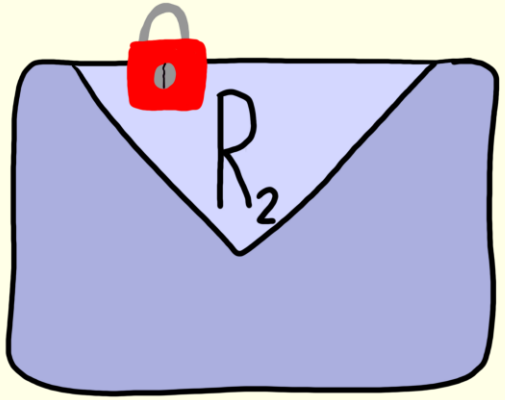
Only info $\mathcal{ADec}$ gets is

$RO(R_{am})$ + 

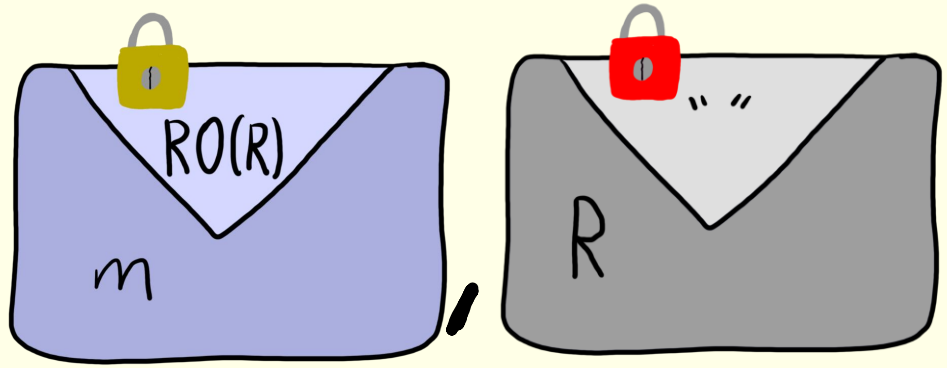
looks honest!

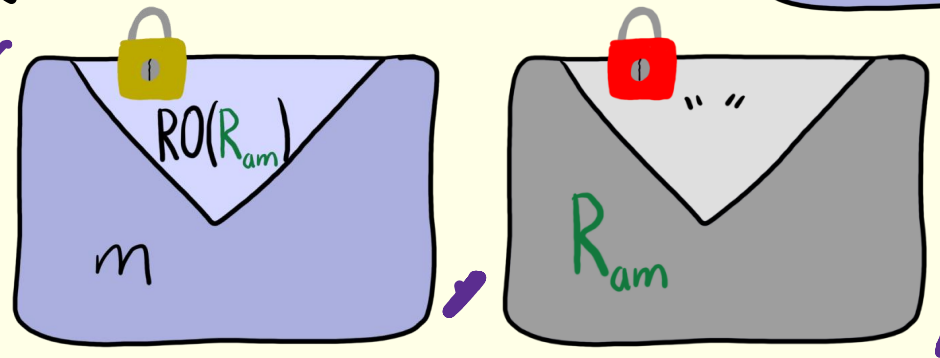
$\approx_{\frac{1}{\#\text{queries}}}$

R_1 + 

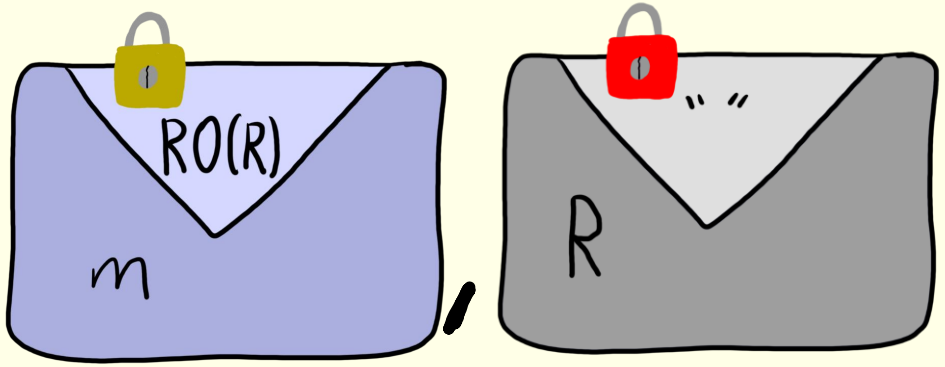
$\approx$ R_1 + 

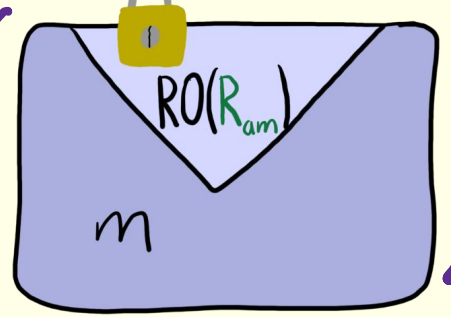
Final Proof

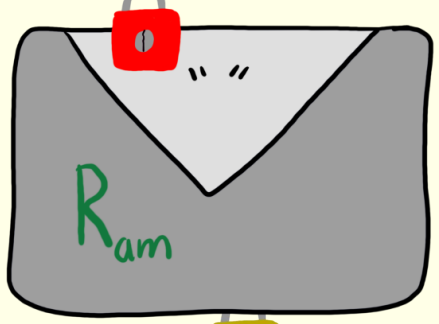
$Enc(pp, pk, m; R) =$ 

If $ADec(\cdot, \cdot) = am$ 

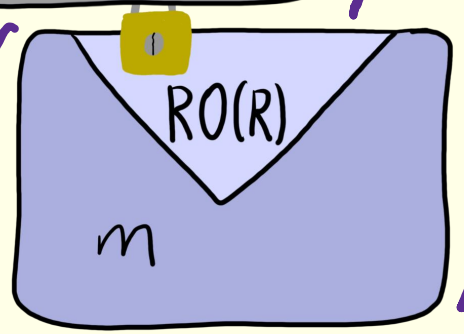
Final Proof

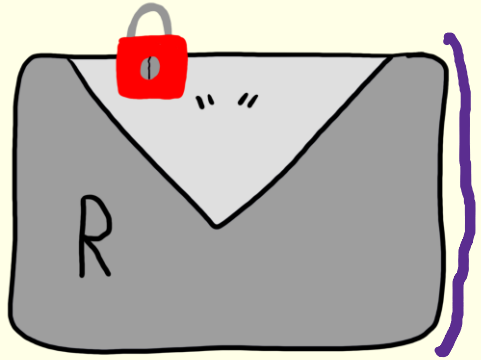
$Enc(pp, pk, m; R) =$ 

If $ADec($ 

, 

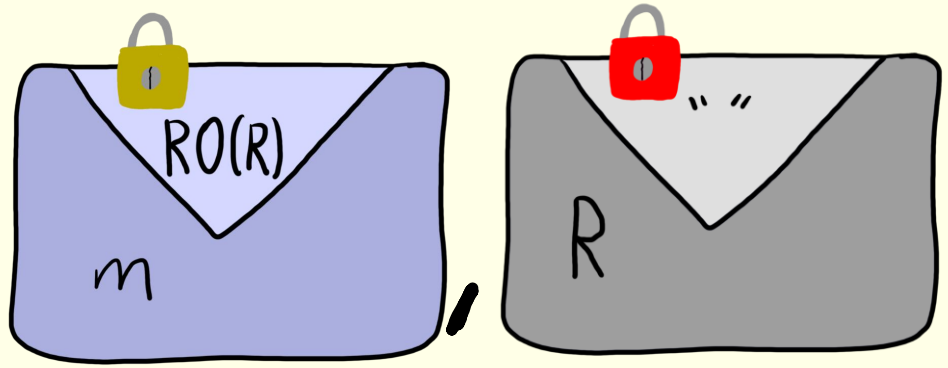
$) = am$

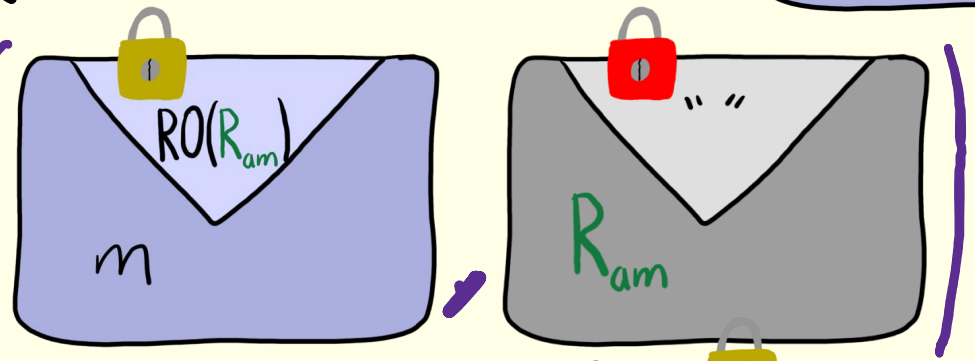
With prob $\frac{1}{\#queries+1}$ $ADec($ 

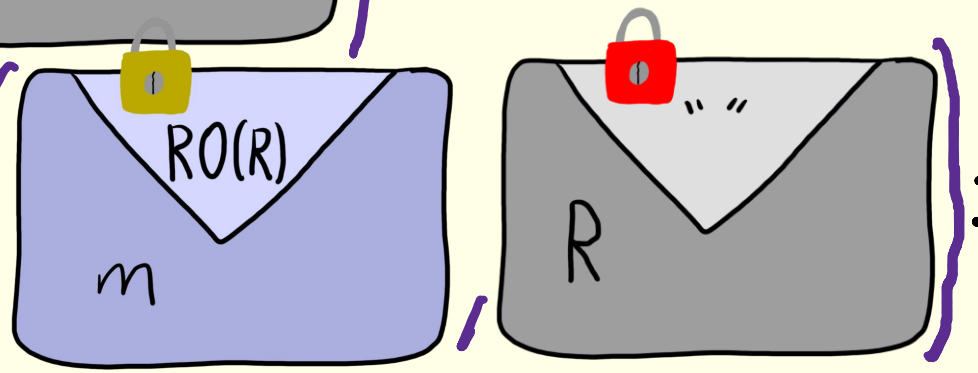
, 

$) = am$

Final Proof

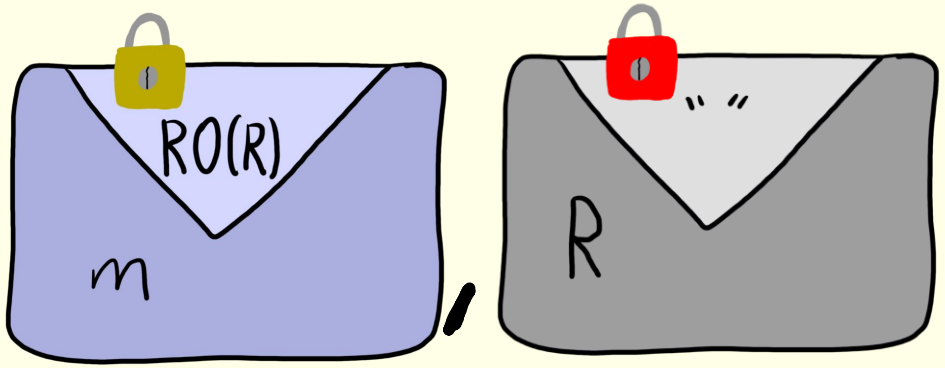
$Enc(pp, pk, m; R) =$ 

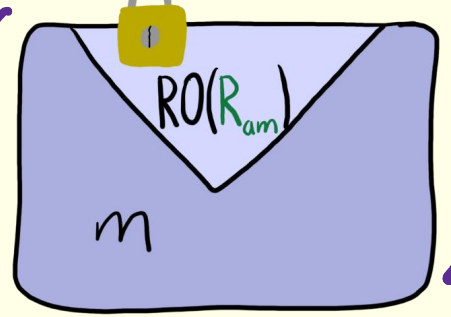
If $ADec(\cdot, \cdot) = am$ 

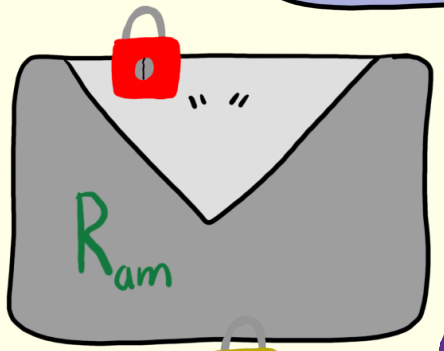
With prob $\frac{1}{\#queries+1}$ $ADec(\cdot, \cdot) = am$ 

Only possible for $\#queries+1 = poly(\lambda)$ different values of am

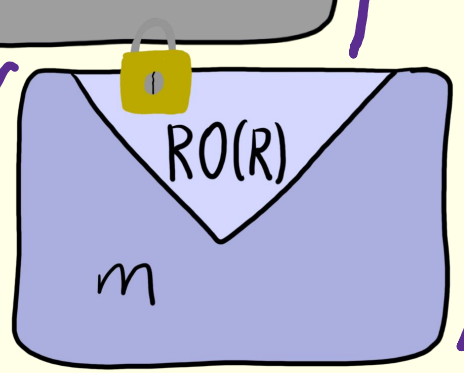
Final Proof

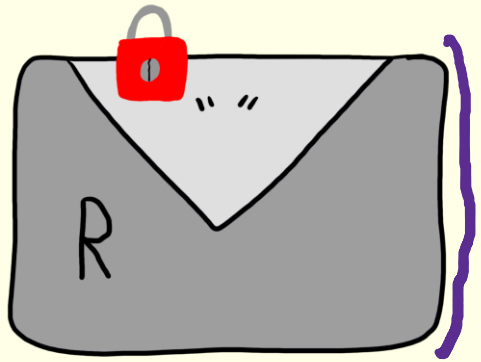
$Enc(pp, pk, m; R) =$ 

If $ADec($ 

, 

$) = am$

With prob $\frac{1}{\#queries+1}$ $ADec($ 

, 

$) = am$

NOT ROBUST! (even for 1-bit messages)

Anamorphic Encryption

Hypothesis: linear bandwidth anamorphic instantiations are always possible.

At least for CCA schemes? [PPYZ24]

Anamorphic Encryption

Hypothesis: linear bandwidth anamorphic instantiations are always possible. NO: ARE

At least for CCA schemes? [PPYZ24]

Anamorphic Encryption

Hypothesis: linear bandwidth anamorphic instantiations are always possible. NO: ARE

At least for CCA schemes? [PPYZ24]

Our construction can be made CCA secure!

Dictatoria

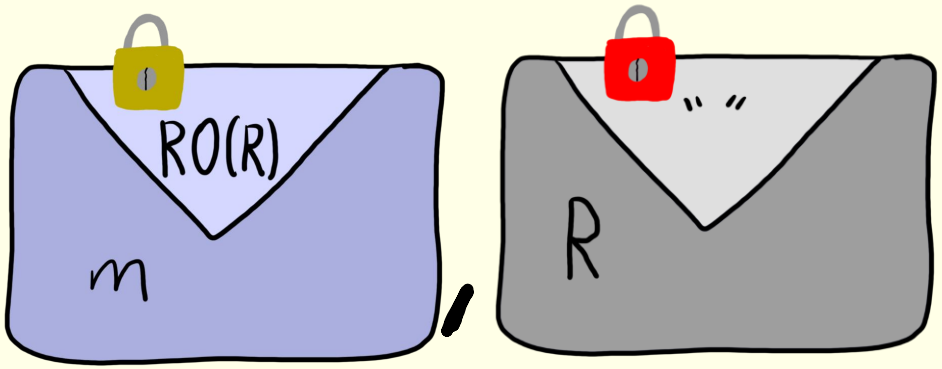
ARE w/ in ROM

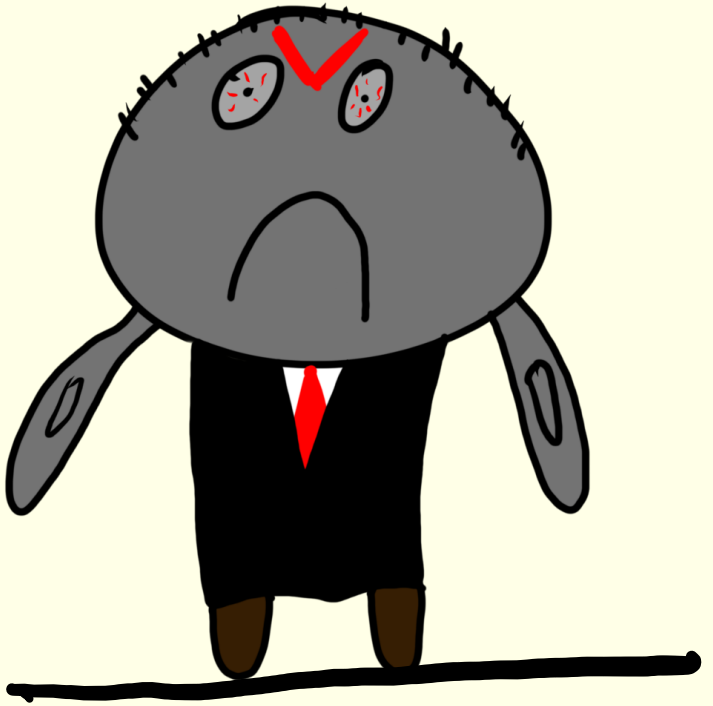
universal backdoor.

Dictator can read all messages & detect anamorphism without secret-key access

Dictatoria

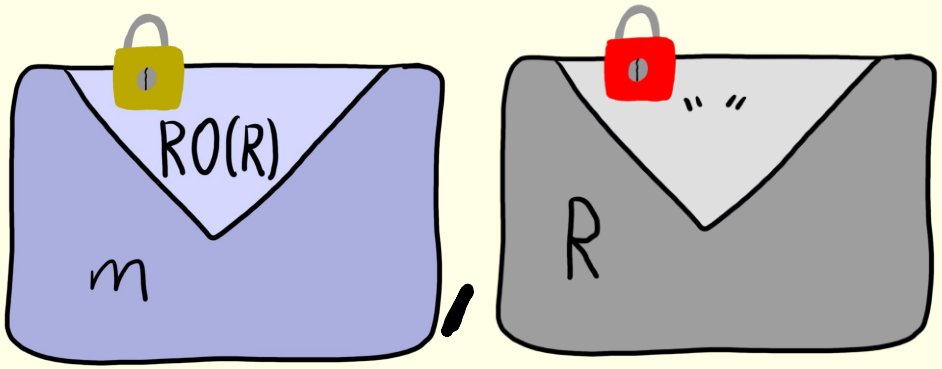
ARE

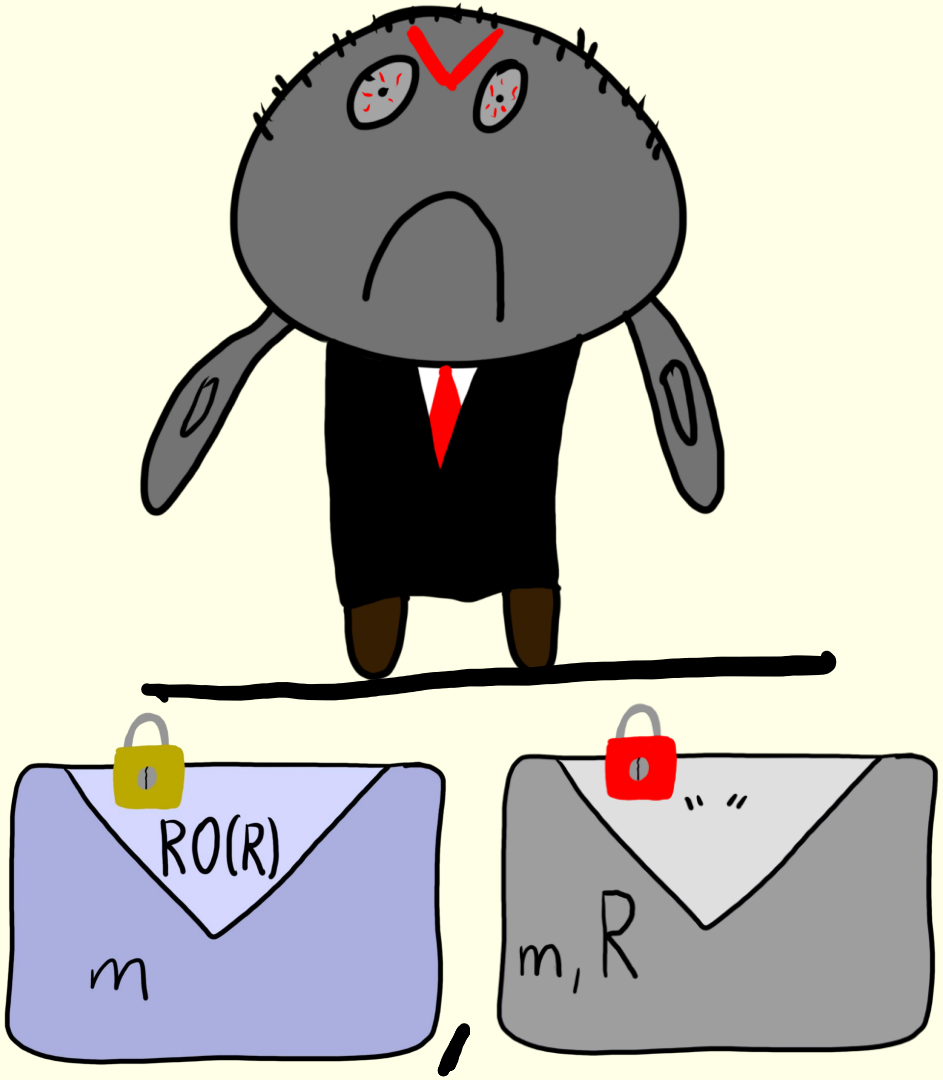




Dictatoria

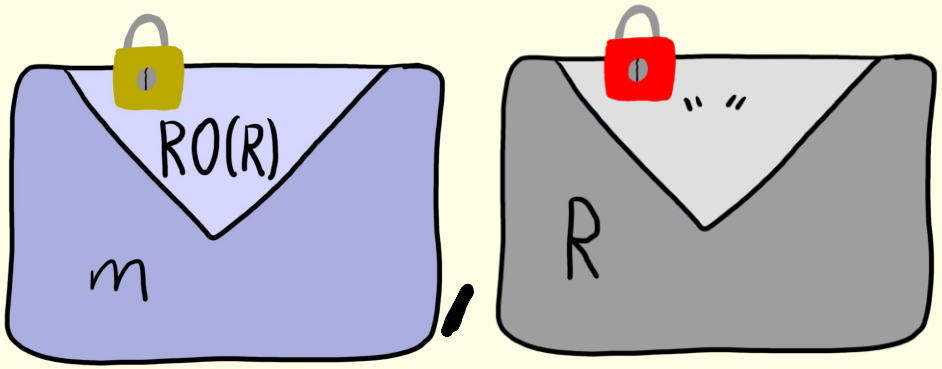
ARE

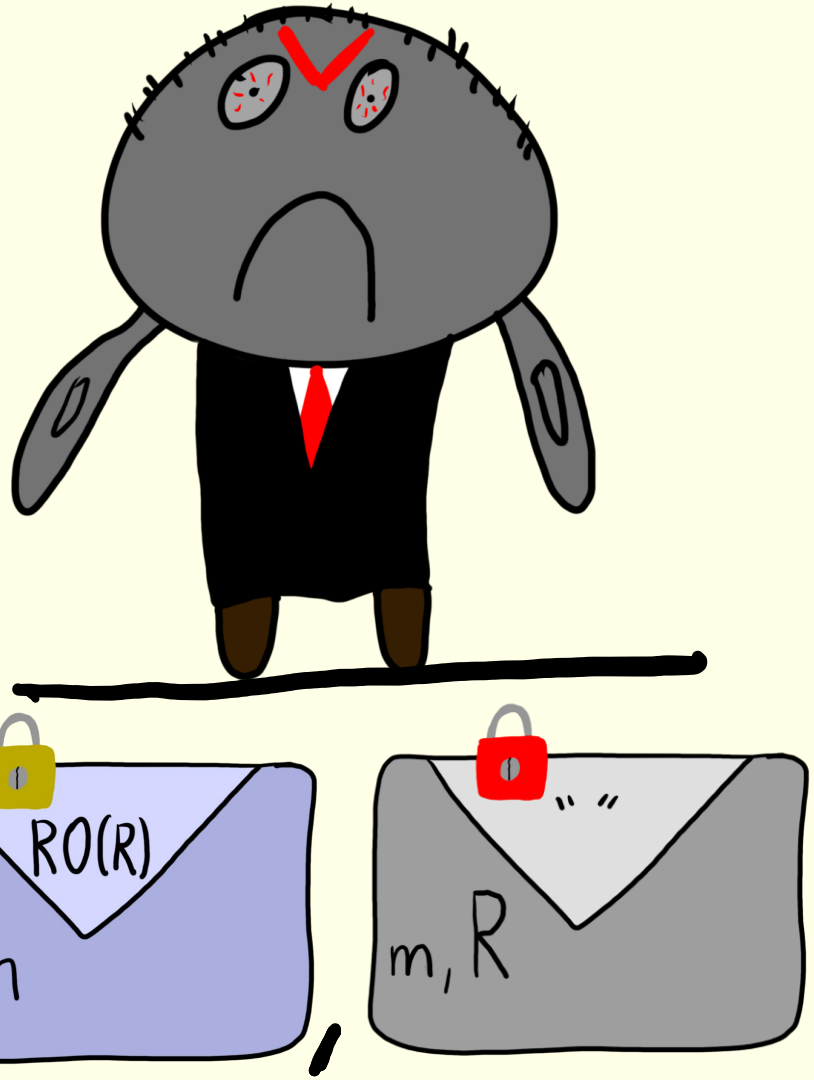




Dictatoria

ARE





Dictator can read all m to detect covert messages

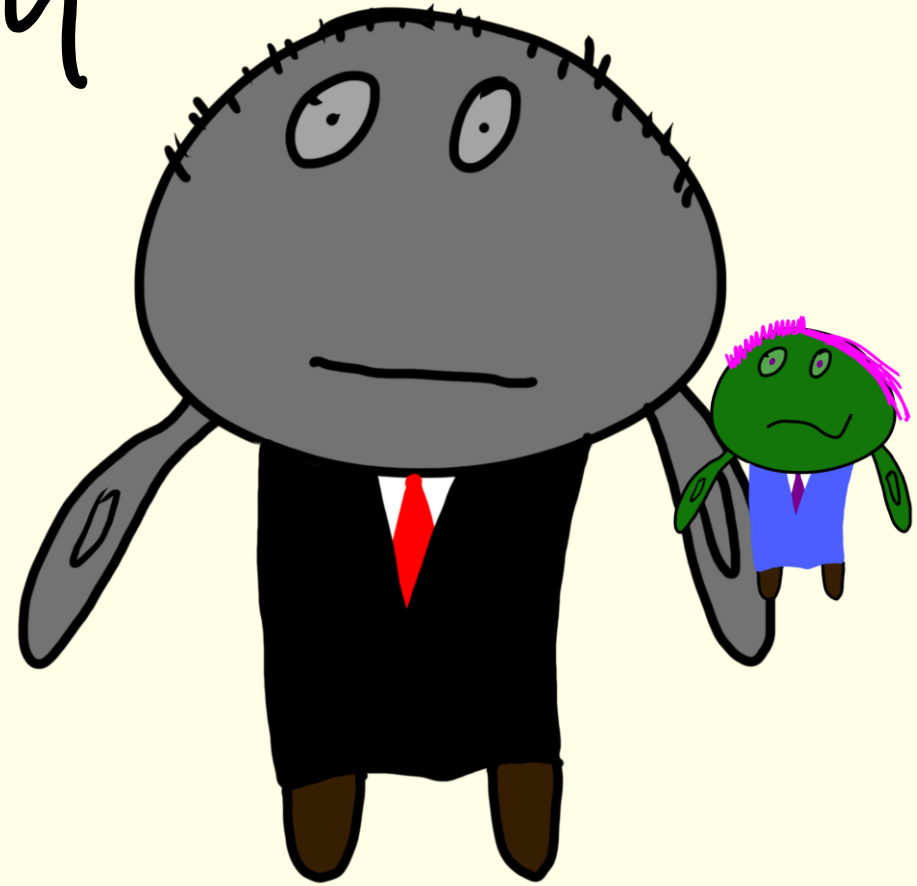
Warrantland in ROM

ARE which needs secret key access.

If  hidden, secure against govt. w/ 

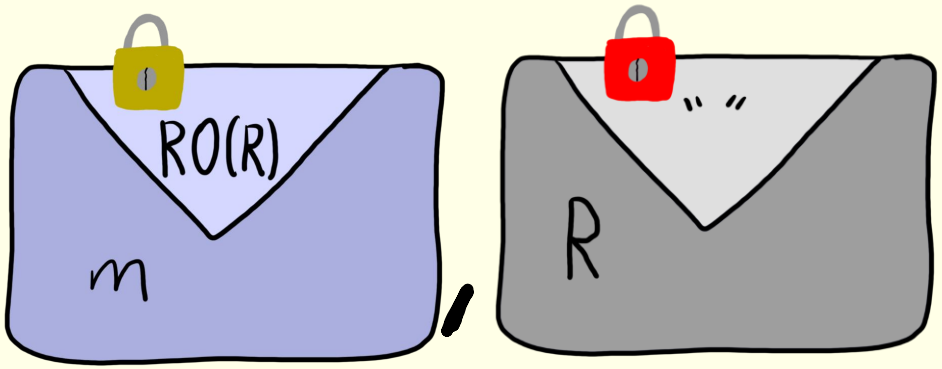
sk

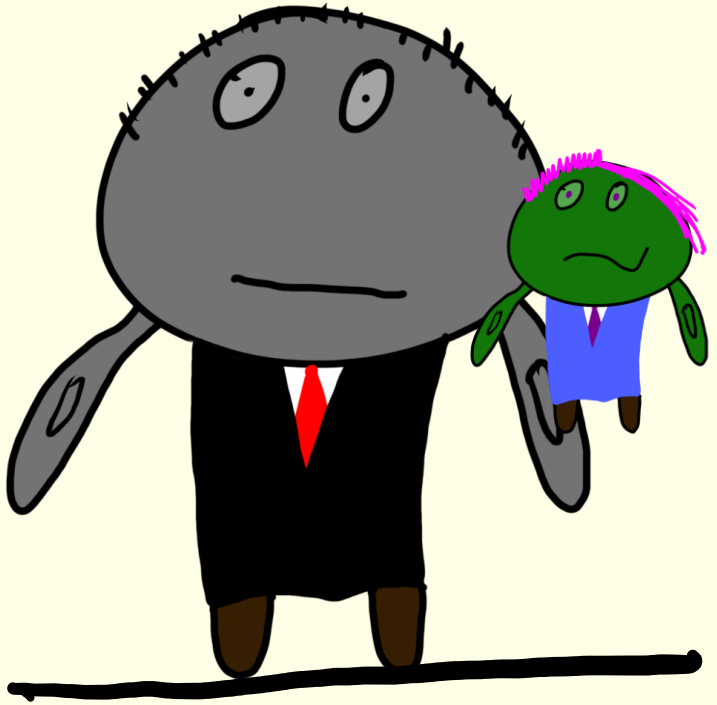
backdoor



Warrantland

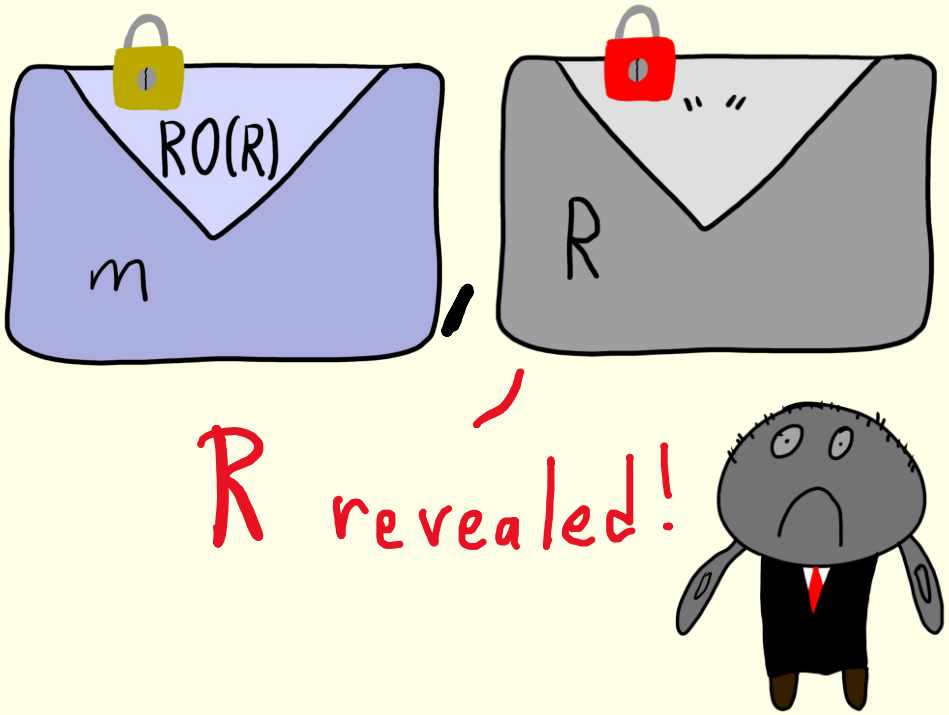
ARE

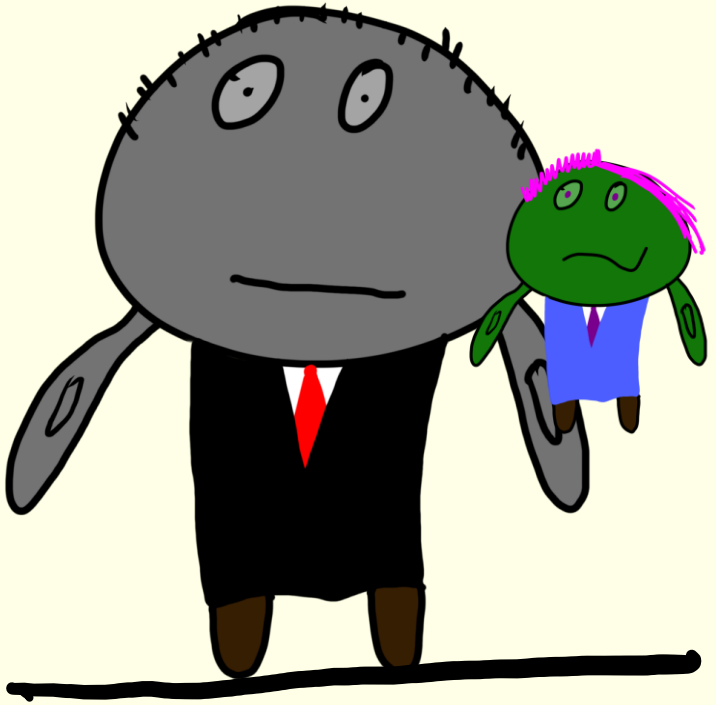




Warrantland

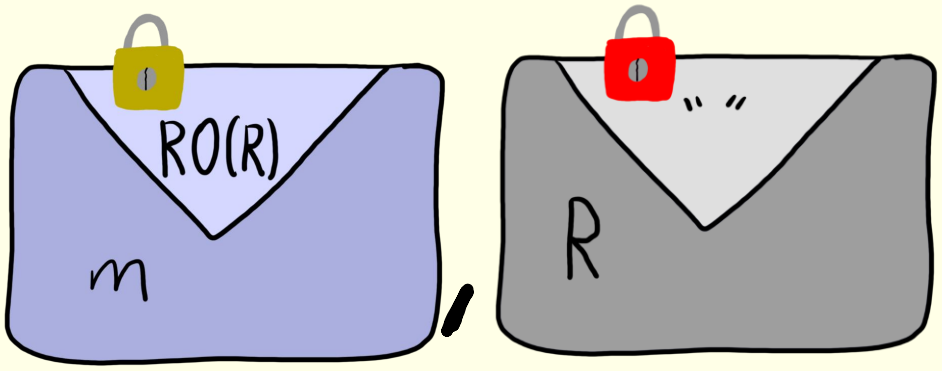
ARE

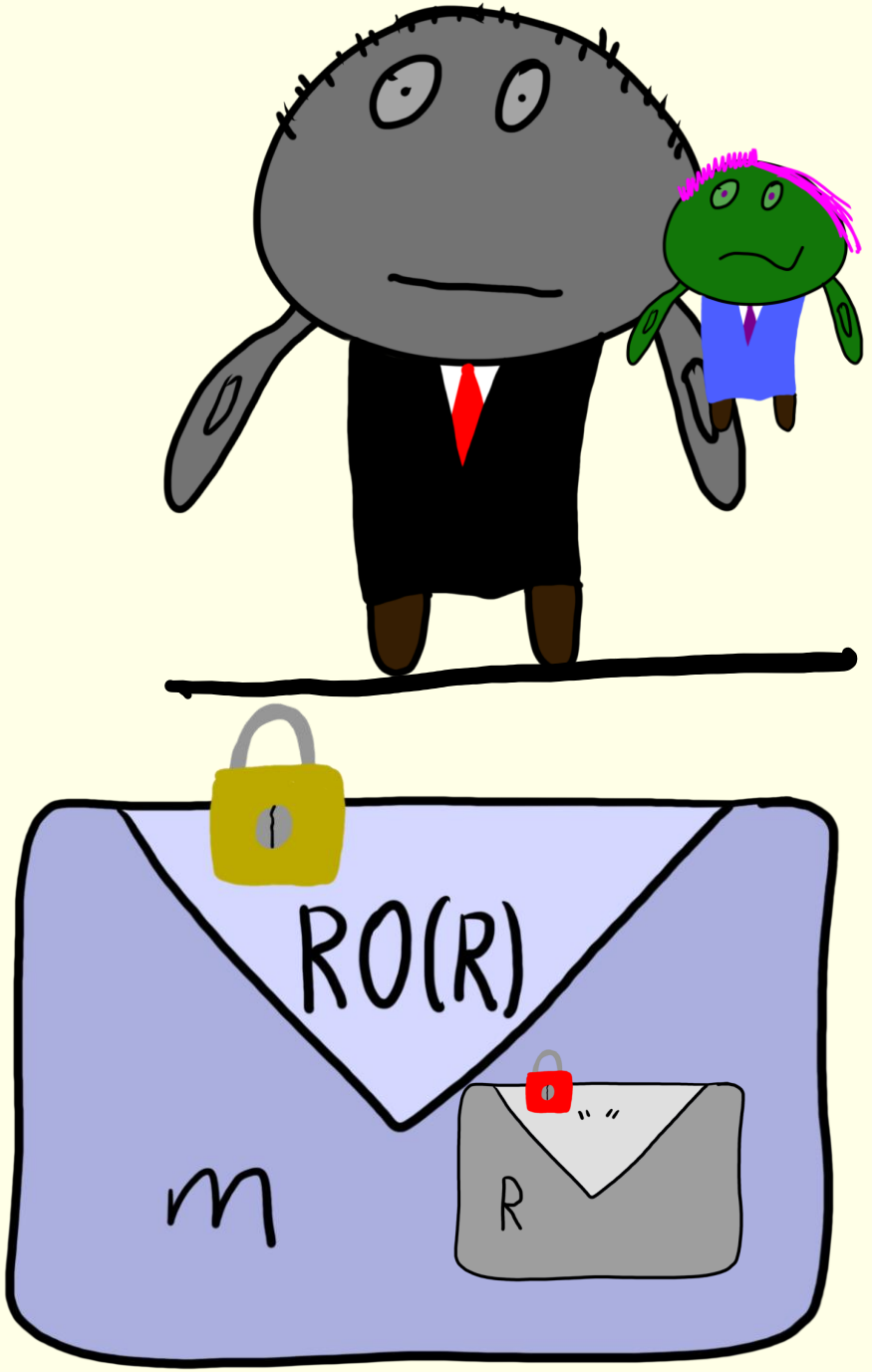




Warrantland

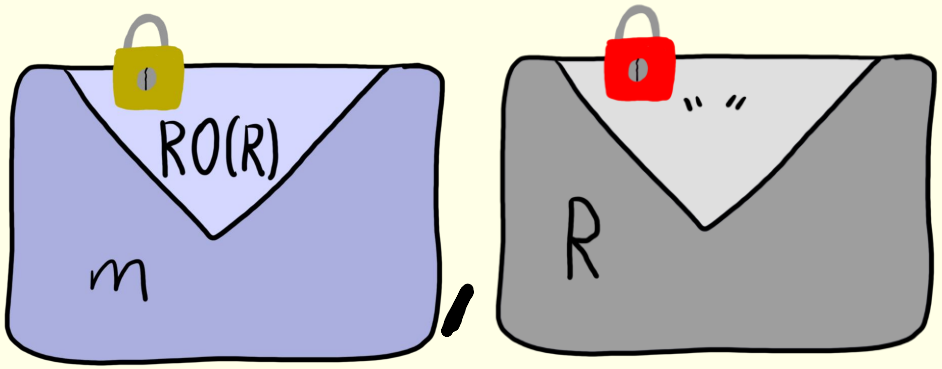
ARE



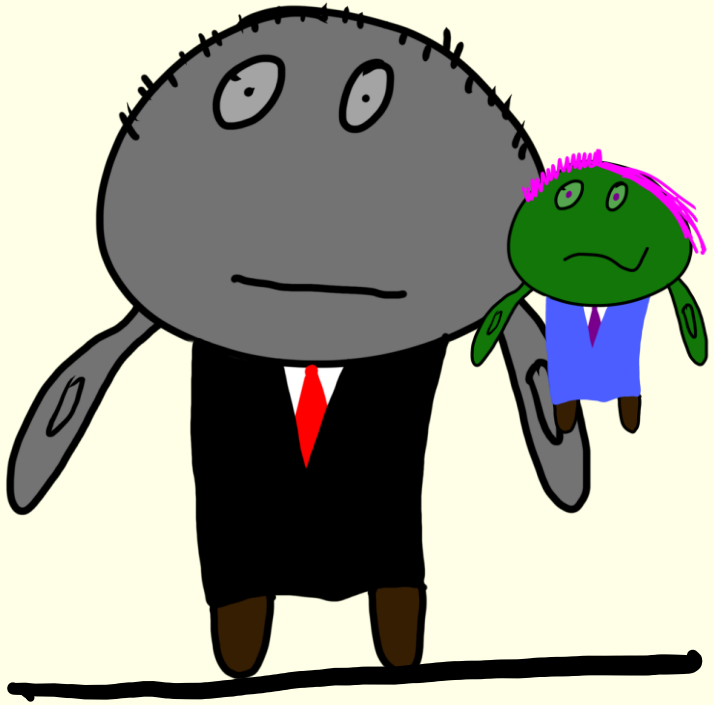


Warrantland

ARE

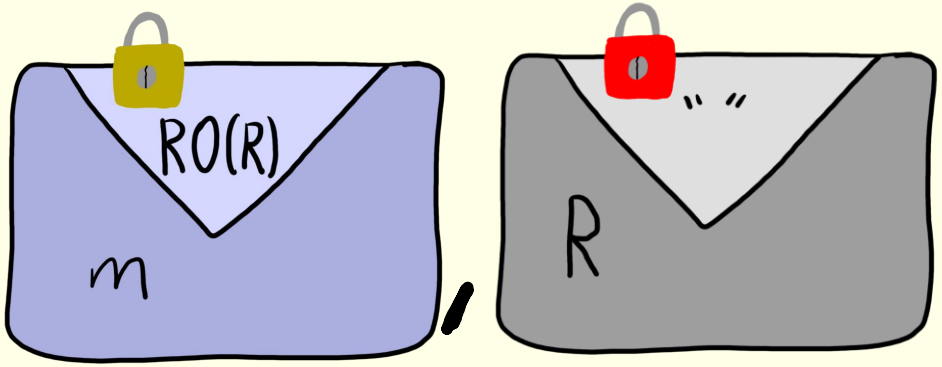


Govt. needs  to detect anamorphism or break PKE.



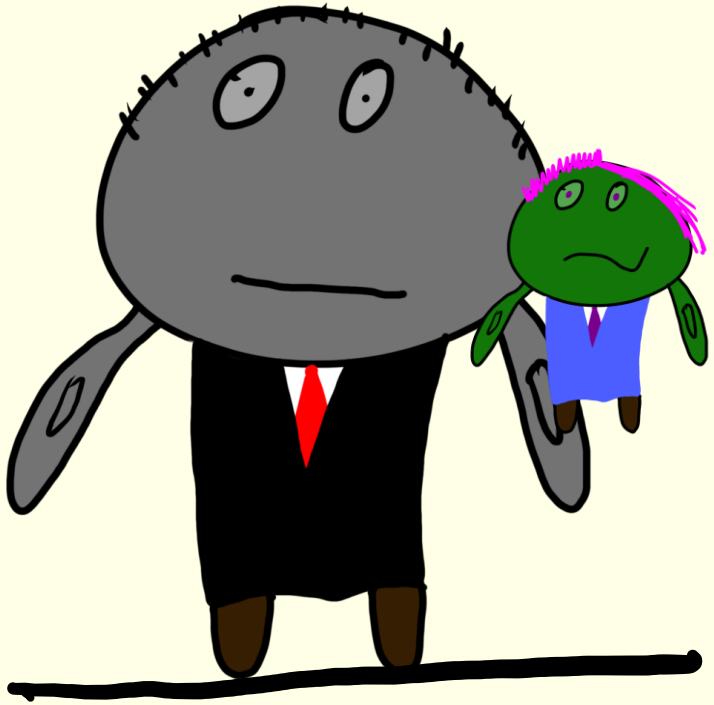
Warrantland

ARE

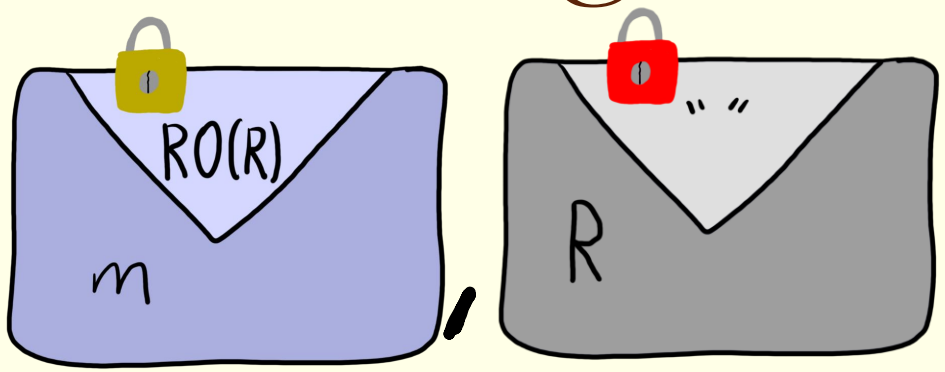


Govt. needs  to detect anamorphism or break PKE.

CCA

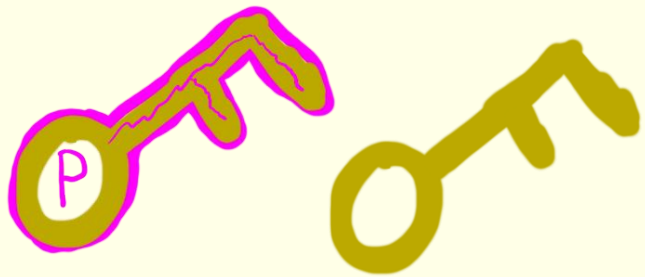


PKI Agnostic



Our ARE scheme

- augments any PKE
- does not require changing



Open Questions

1. Standard Model ARE

Open Questions

1. Standard Model ARE
   [ABG+25] from sub-exponential DDH

Open Questions

1. Standard Model ARE
   [ABG+25] from sub-exponential DDH
2. Getting rid of public parameters.

Open Questions

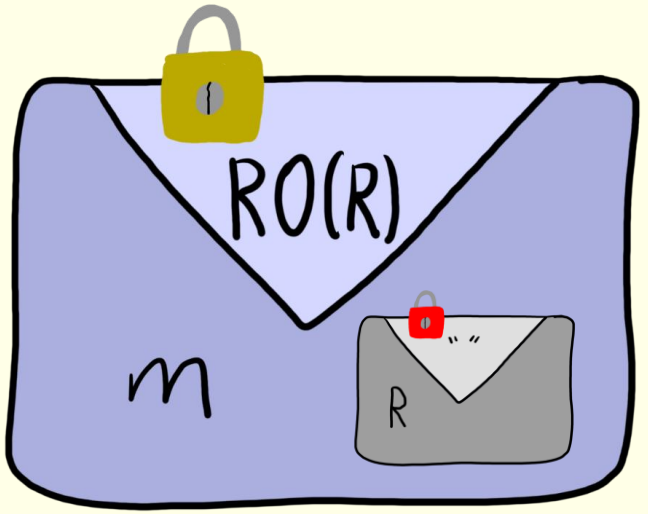
1. Standard Model ARE
   [ABG+25] from sub-exponential DDH
2. Getting rid of public parameters.
   [CCGM25] does for weaker notion of ARE

Open Questions

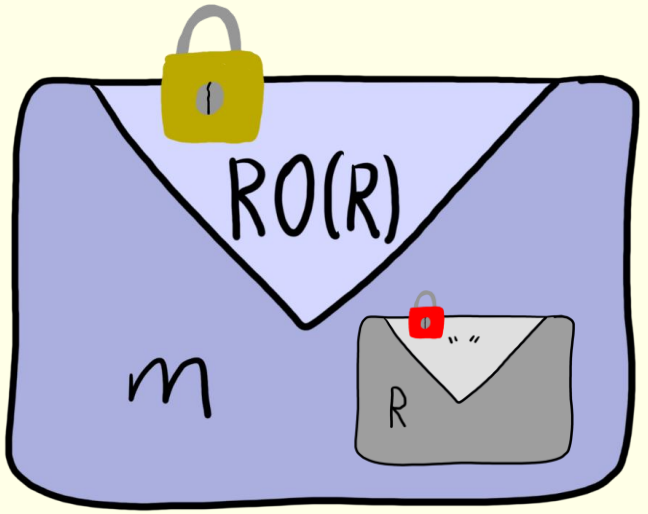
1. Standard Model ARE
   [ABG+25] from sub-exponential DDH
2. Getting rid of public parameters.
   [CCGM25] does for weaker notion of ARE
3. Generic construction of strong unforgeability.

Game-based ARE

So, no anamorphic instantiation of



Game-based ARE



So, no anamorphic instantiation of

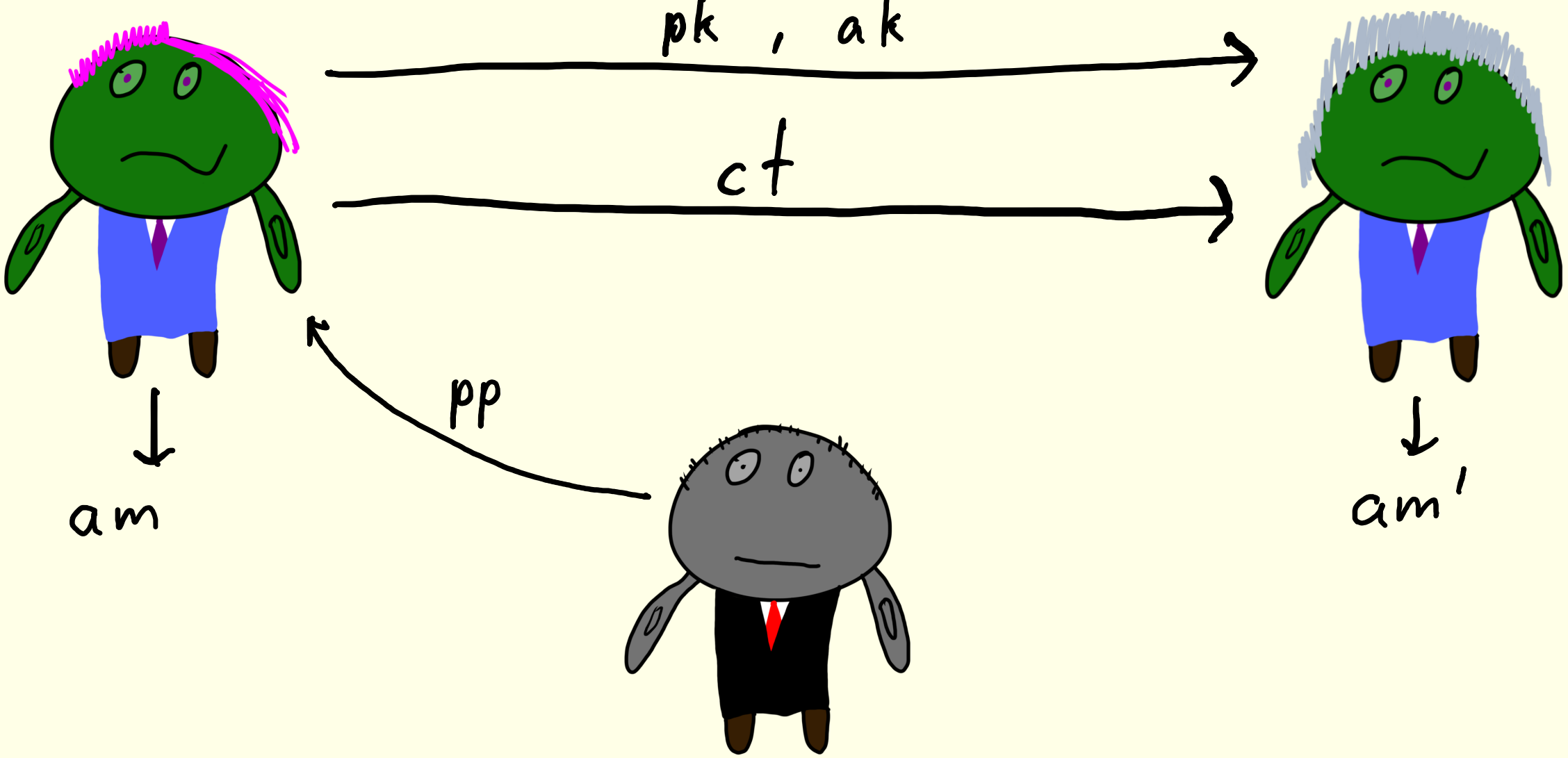
What we *really* want:

algorithm to detect anyone sending anamorphic messages.

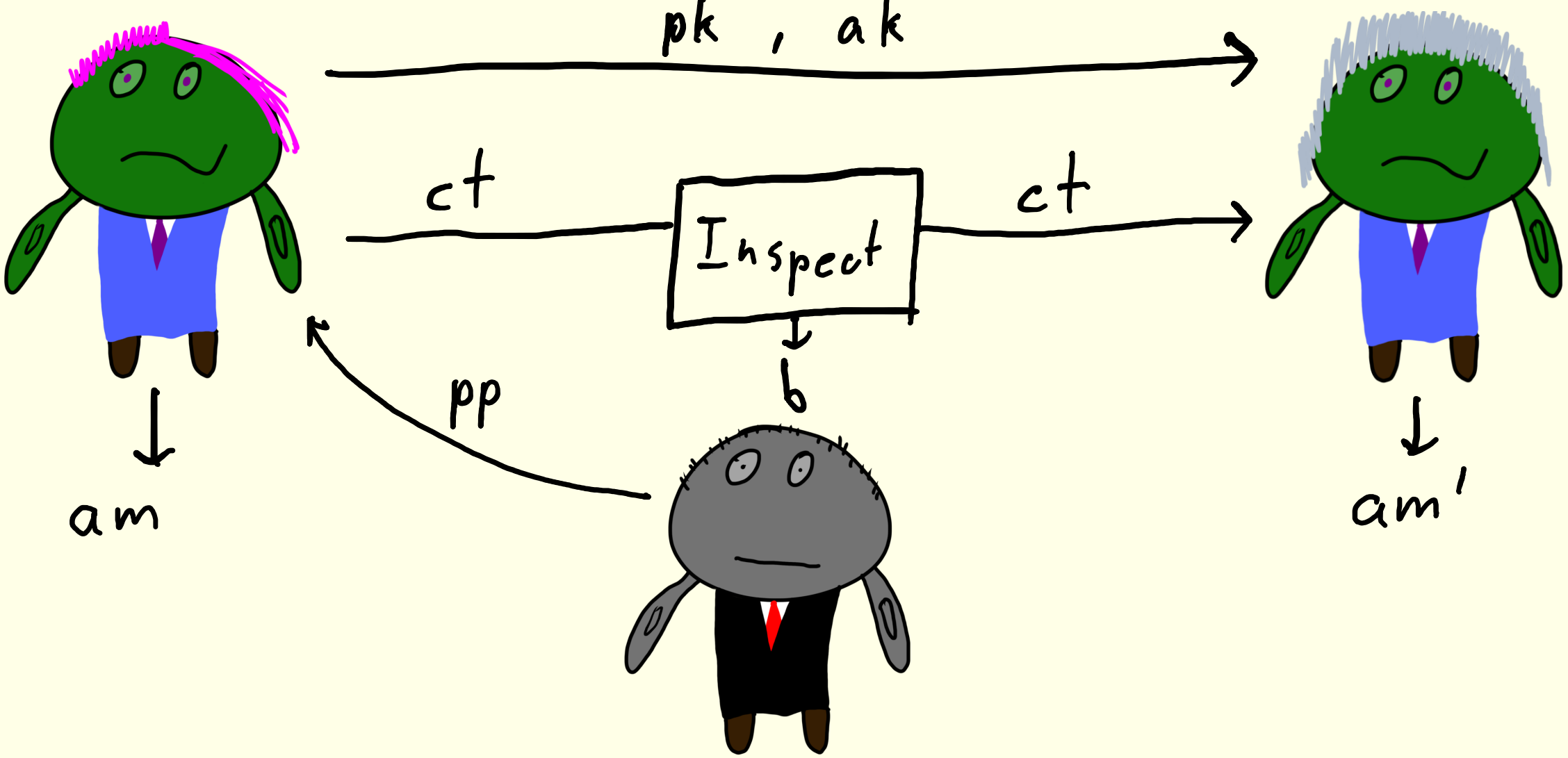
Intuitively: check ct of right form!

Ongoing work: "right" definition

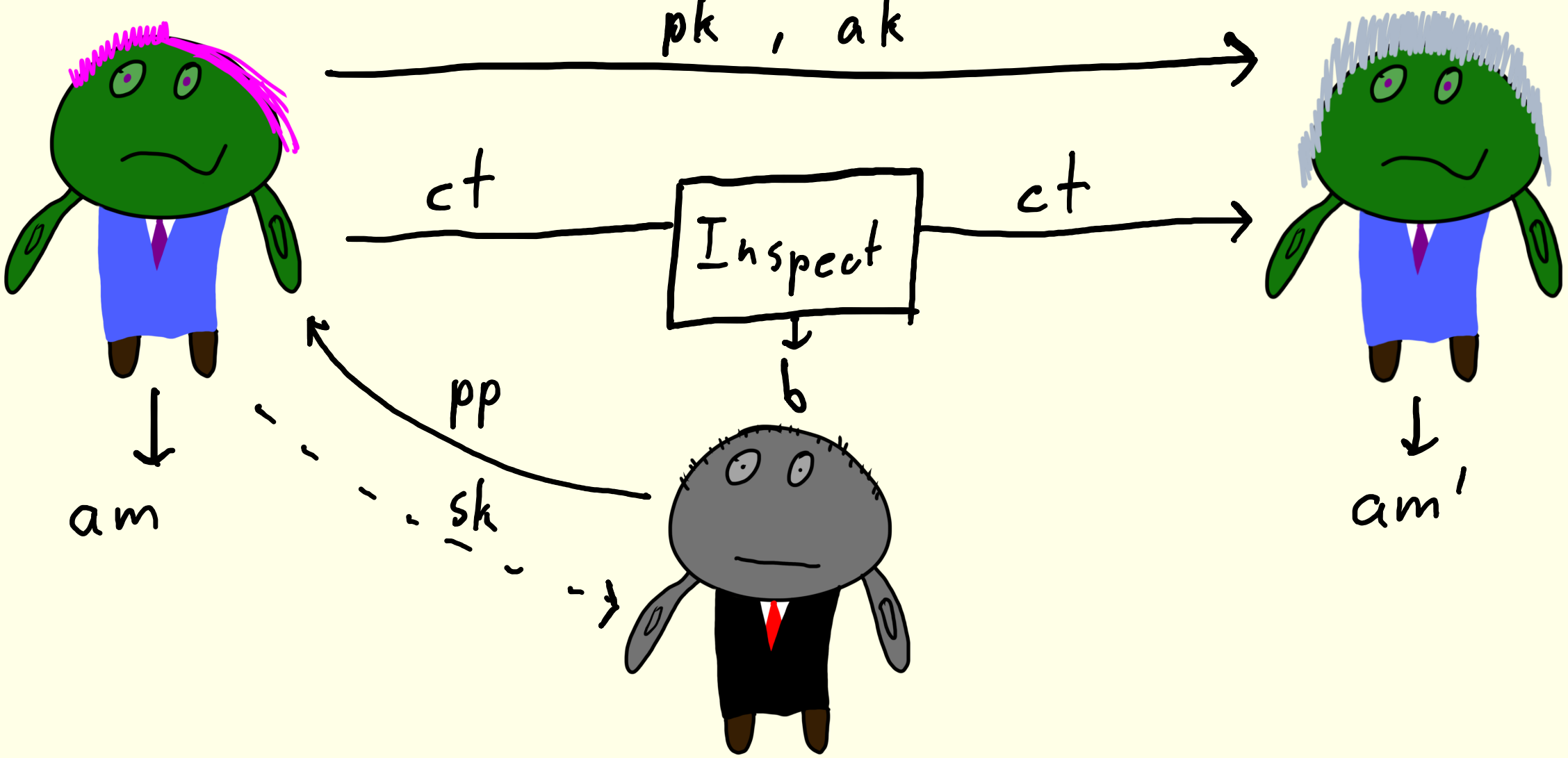
Game-based ARE



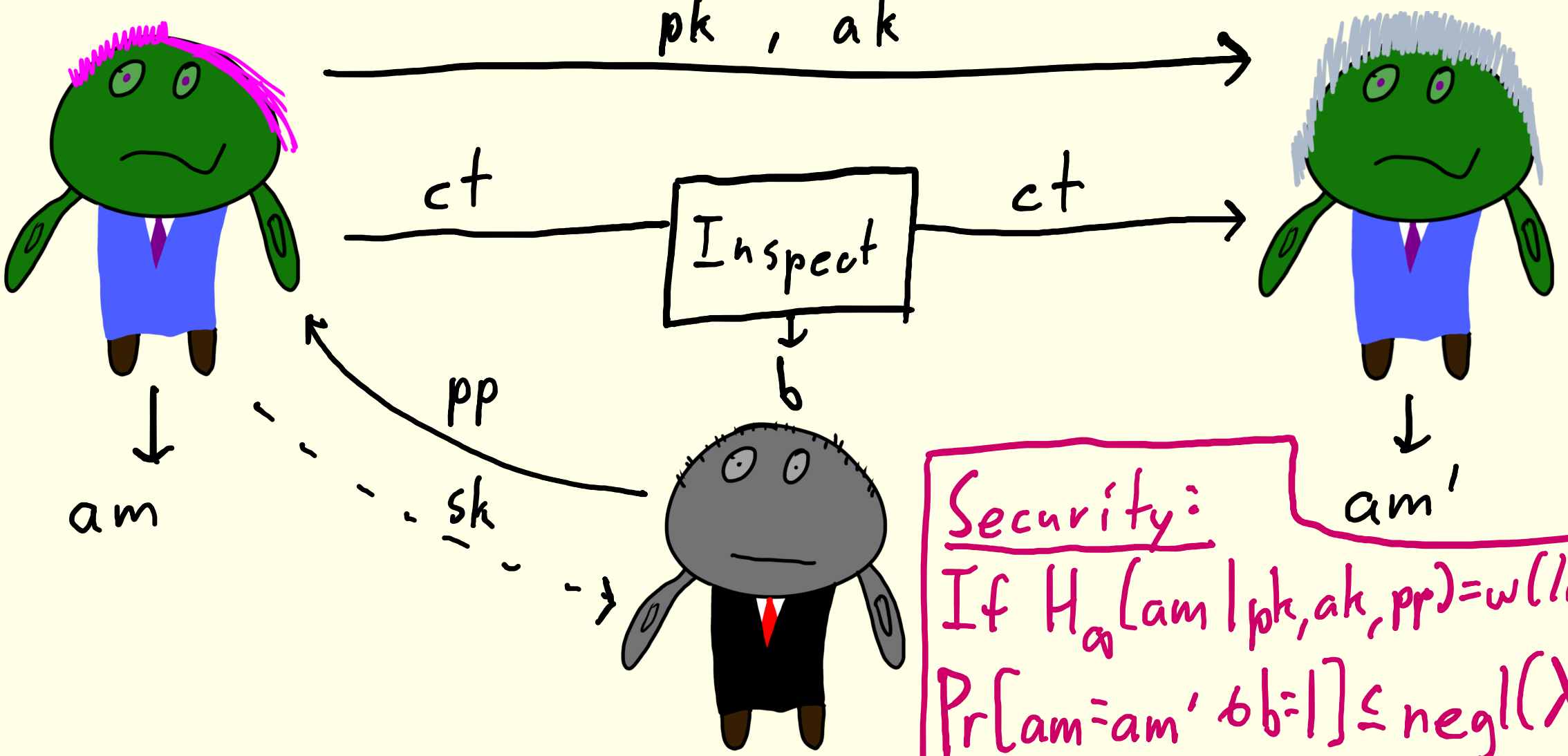
Game-based ARE



Game-based ARE



Game-based ARE

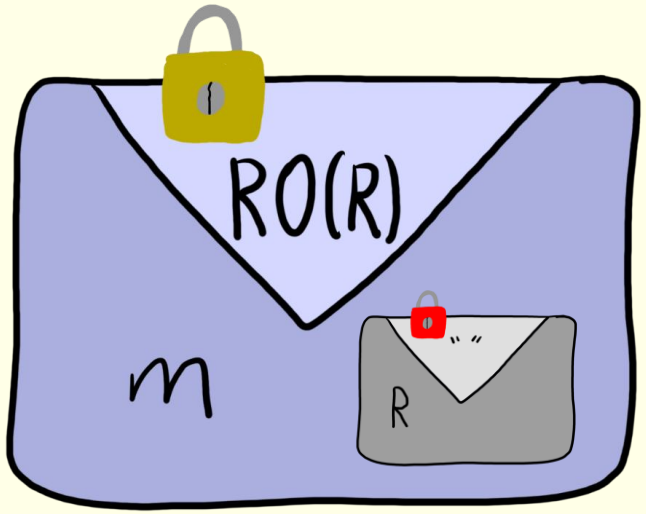


Security:

If $H_\infty(am \mid pk, ak, pp) = \omega(\log \lambda)$

$\Pr[am = am' \wedge b = 1] \leq \mathrm{negl}(\lambda)$

Game-based ARE

Does 

satisfy this

for Inspect: check if right format?

Thanks
for
listening!