Pairing-Free Blind Signatures from Standard Assumptions in the ROM

Julia Kastner[1], Ky Nguyen[2], Michael Reichle[3]

[1] Centrum Wiskunde & Informatica, Amsterdam, Netherlands - *Work done while at ETH Zurich*
[2] DIENS, École normale supérieure, CNRS, PSL University, Paris, France
[3] Department of Computer Science, ETH Zurich, Switzerland - *Work done partially while at ENS and Inria, Paris*

Motivations & Contributions

Round-optimal schemes [*see our paper for extensive ref's*]

- Generic and inefficient, or
- need pairings or lattices, or
- rely on interactive assumptions (+ ROM).

Motivations & Contributions

Round-optimal schemes [*see our paper for extensive ref's*]

- Generic and inefficient, or
- need pairings or lattices, or
- rely on interactive assumptions (+ ROM).

Can we have efficient round-optimal blind signatures in ROM, from non-interactive assumptions, w/out pairings nor lattices?

Motivations & Contributions

Round-Optimal Blind Signatures Against Malicious Signers from DDH & sRSA

Motivations & Contributions

Relaxed
Range
Proofs w/
Subversion-ZK

Round-Optimal Blind Signatures Against Malicious Signers from DDH & sRSA

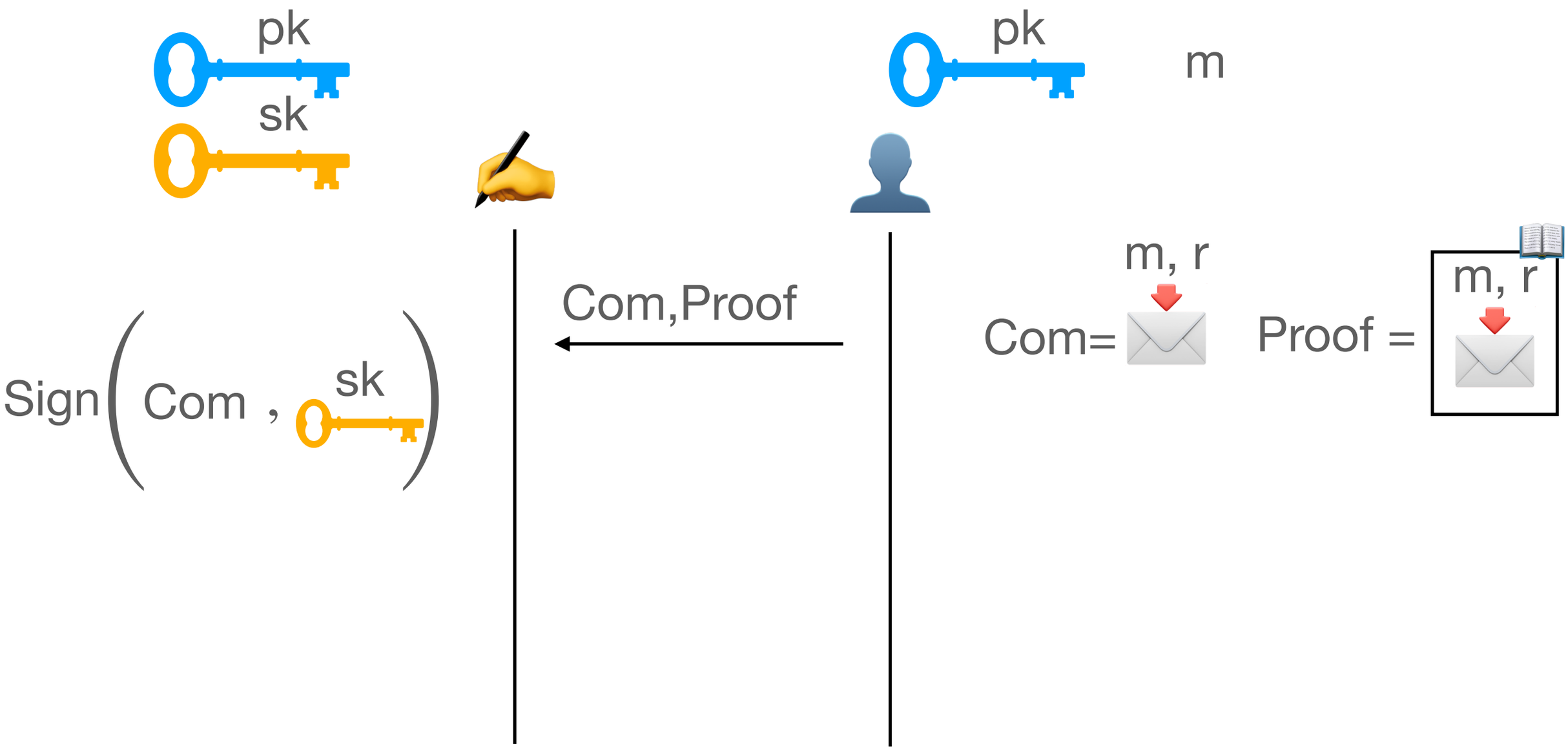
Compact
Commitments
In Arbitrary
Groups

Differences from Previous Talk

	Our work	Previous Talk
Setting	Hidden-Order & Prime-order Pairing-free Groups	Prime-order Pairing-free Groups
Goal	Efficient, Round-optimal, from standard non-interactive assumptions in ROM	In ROM w/out AGM nor GGM
Starting point	Fischlin Framework	Blind Schnorr & Blind BLS
Main challenges	Blindness against malicious signers, efficiency	Concurrent one-more unforgeability

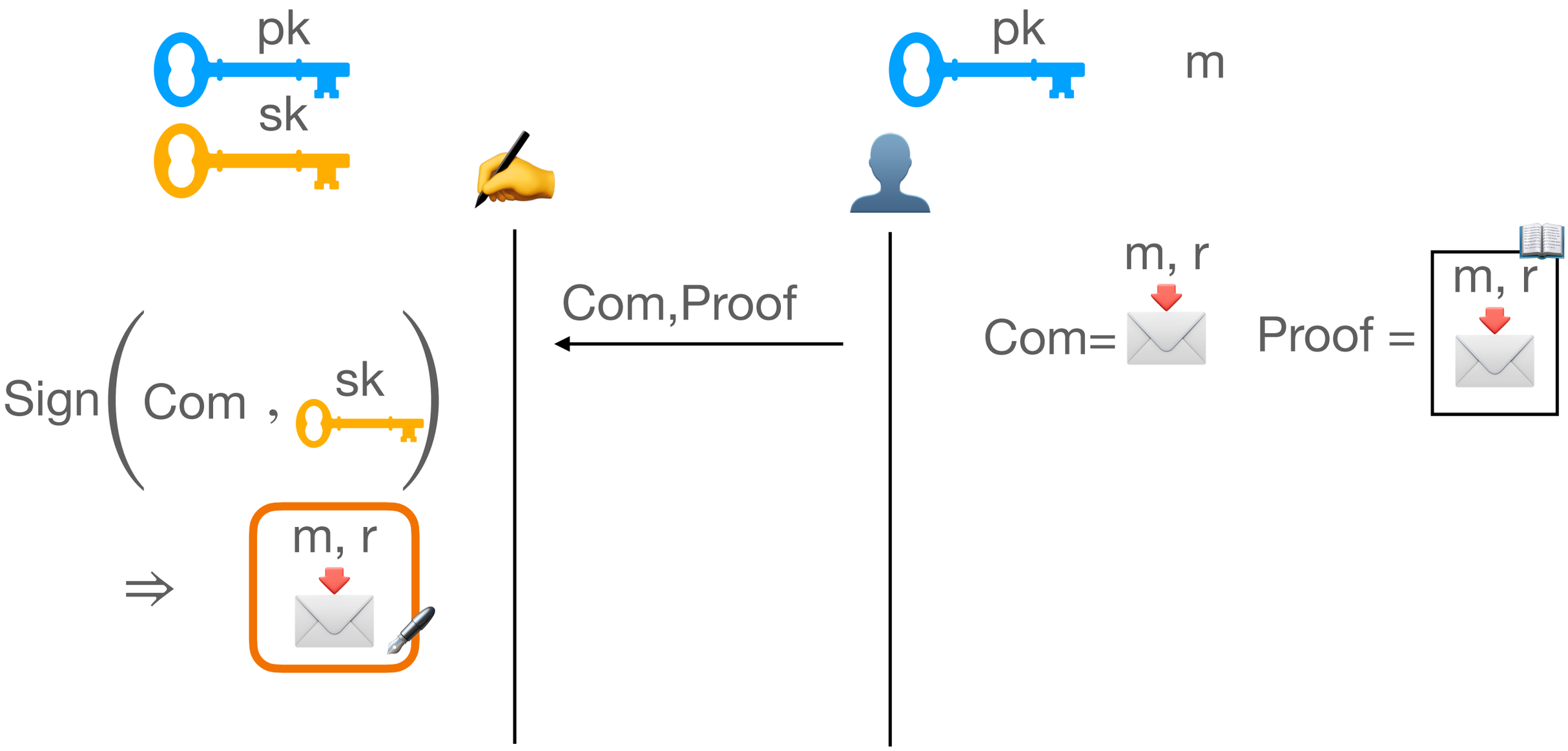
Recall: Fischlin Framework (1)

[Fischlin06, Katsumata-Reichle-Sakai23]



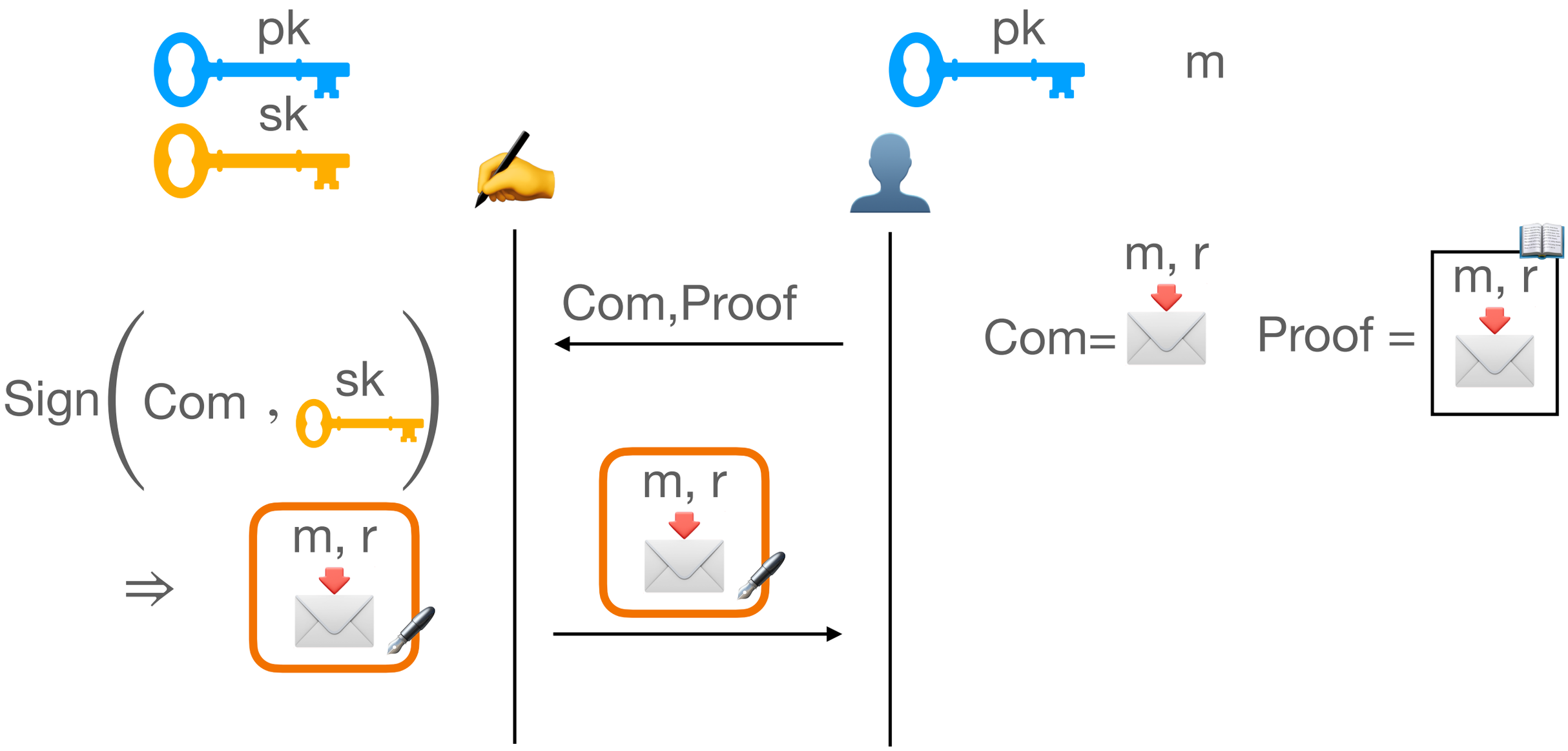
Recall: Fischlin Framework (1)

[Fischlin06, Katsumata-Reichle-Sakai23]



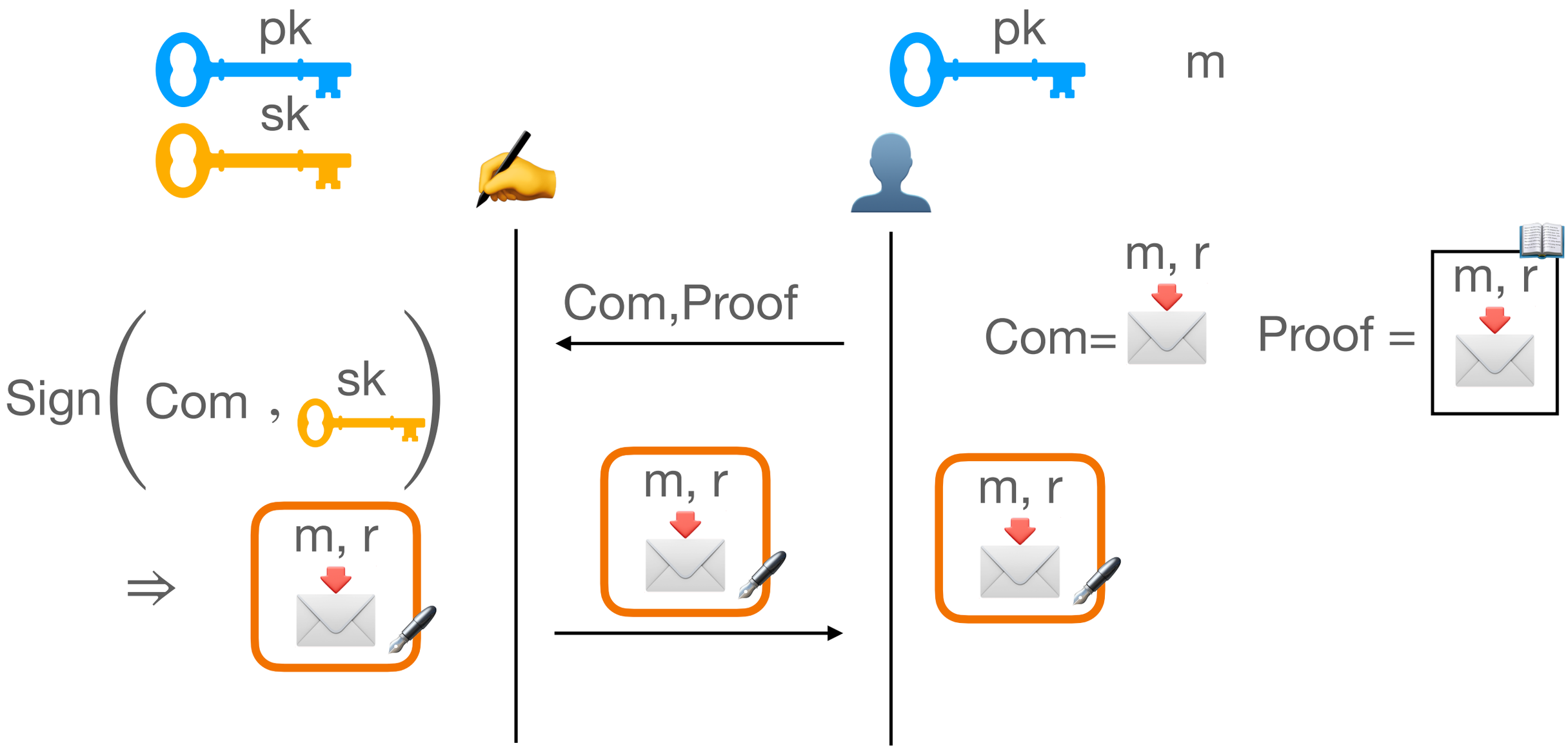
Recall: Fischlin Framework (1)

[Fischlin06, Katsumata-Reichle-Sakai23]



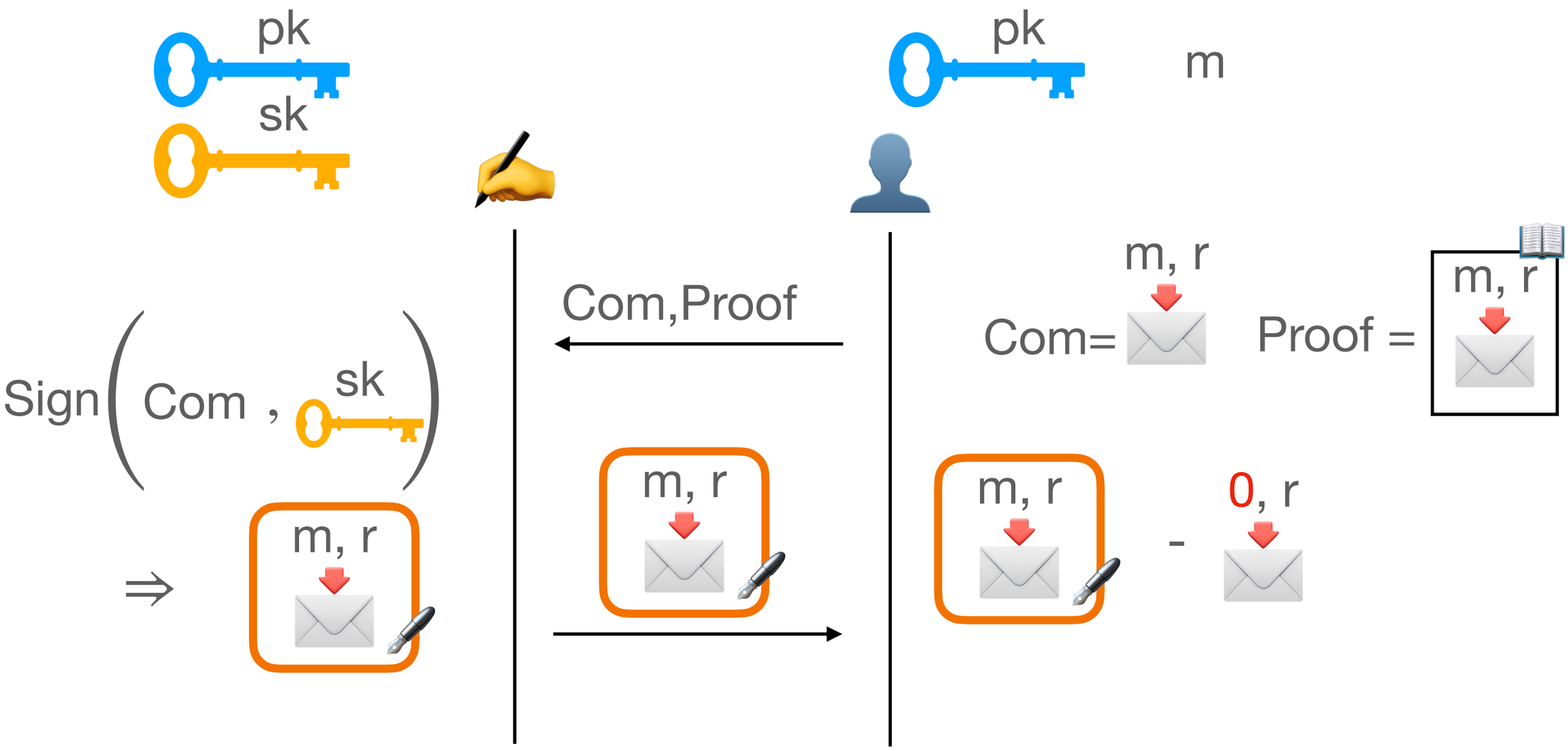
Recall: Fischlin Framework (1)

[Fischlin06, Katsumata-Reichle-Sakai23]



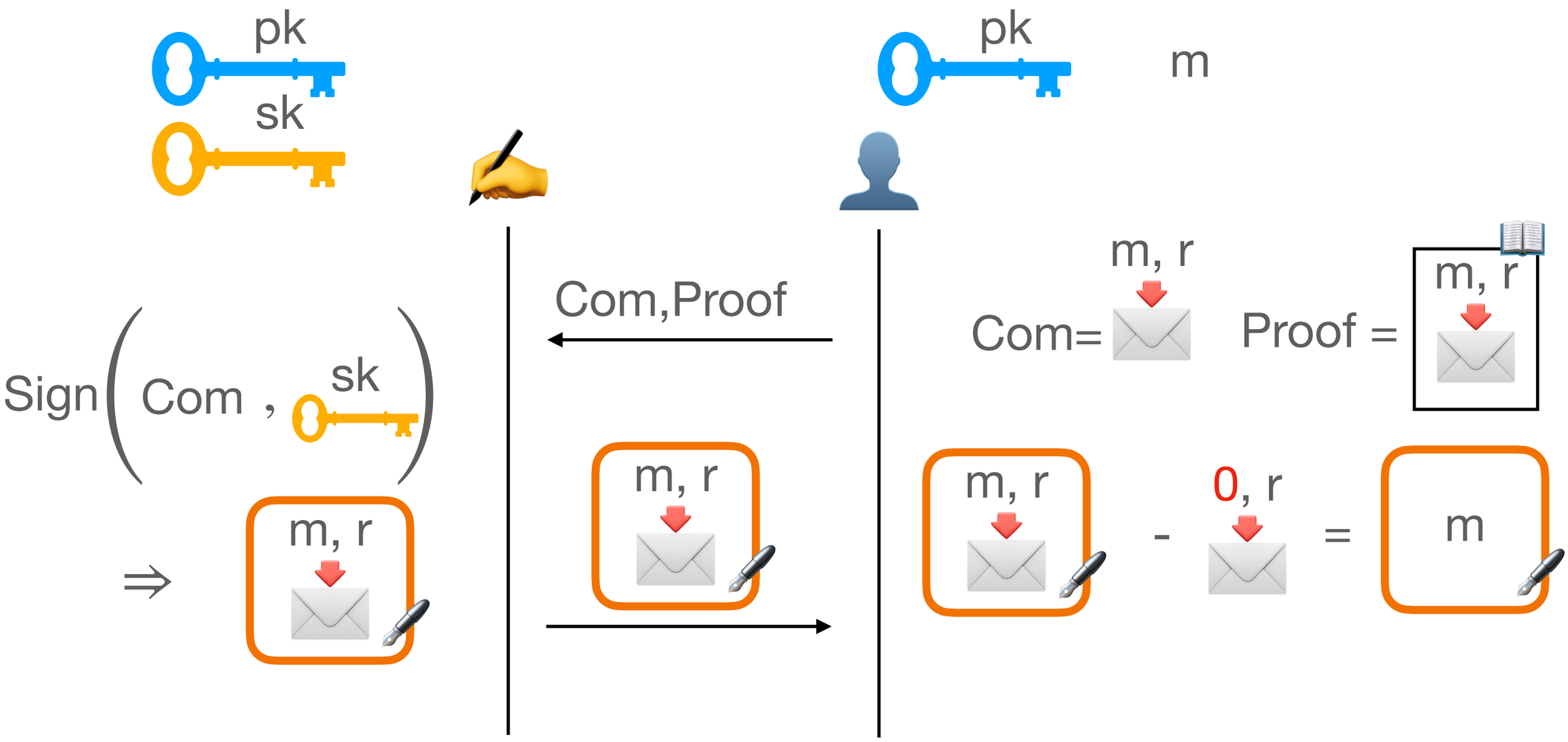
Recall: Fischlin Framework (1)

[Fischlin06, Katsumata-Reichle-Sakai23]



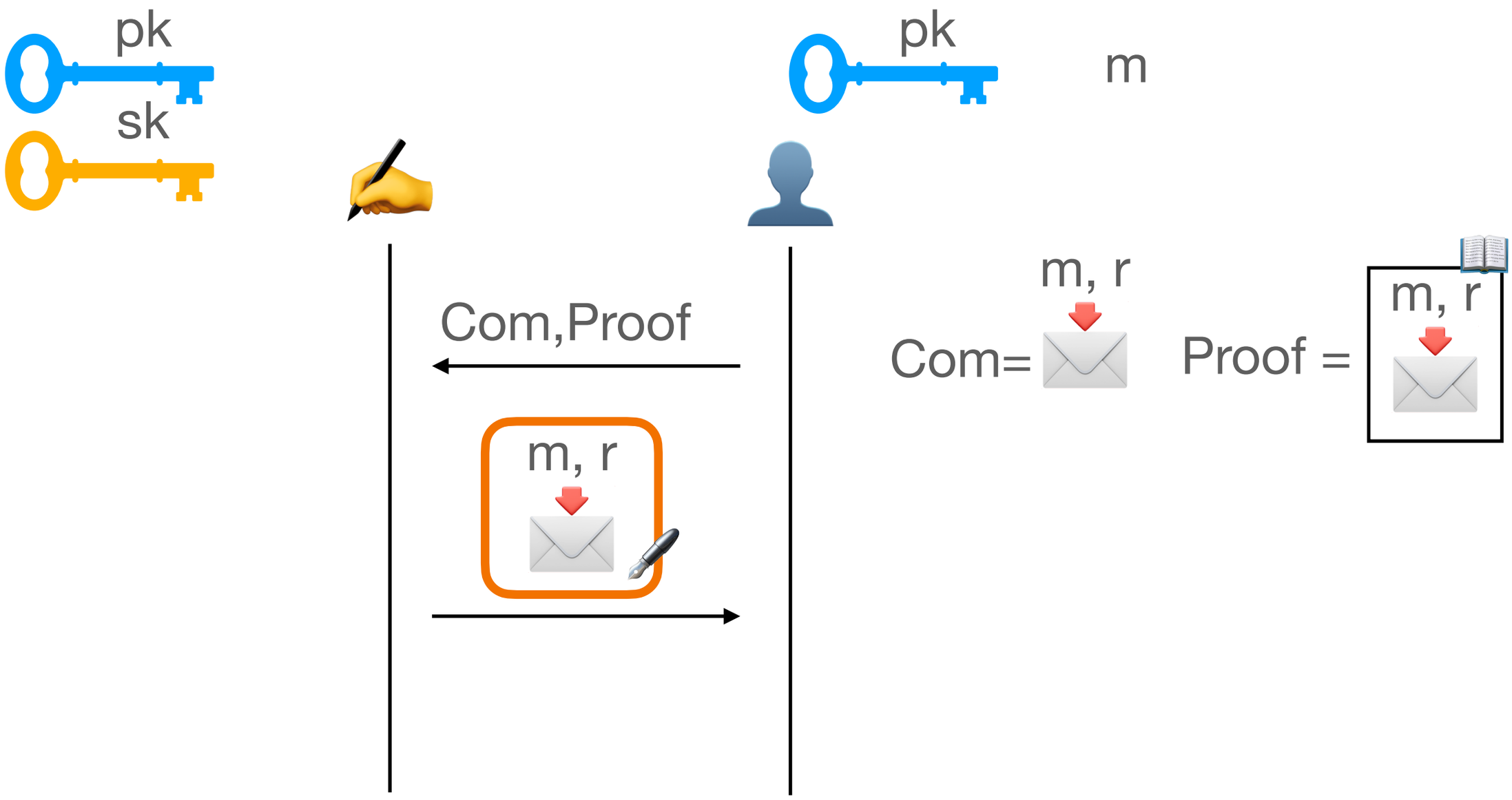
Recall: Fischlin Framework (1)

[Fischlin06, Katsumata-Reichle-Sakai23]



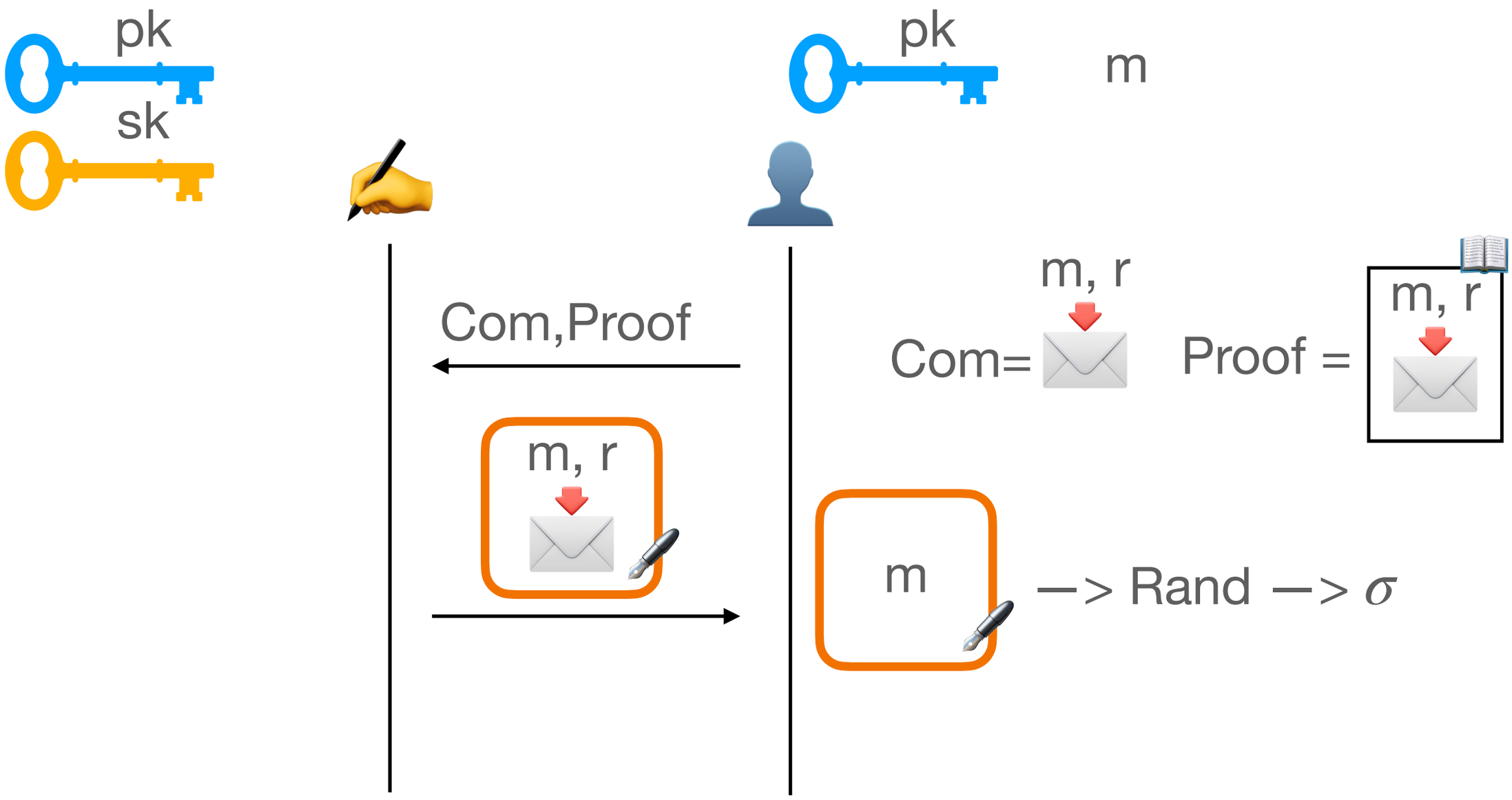
Recall: Fischlin Framework (2)

[Fischlin06, Katsumata-Reichle-Sakai23]



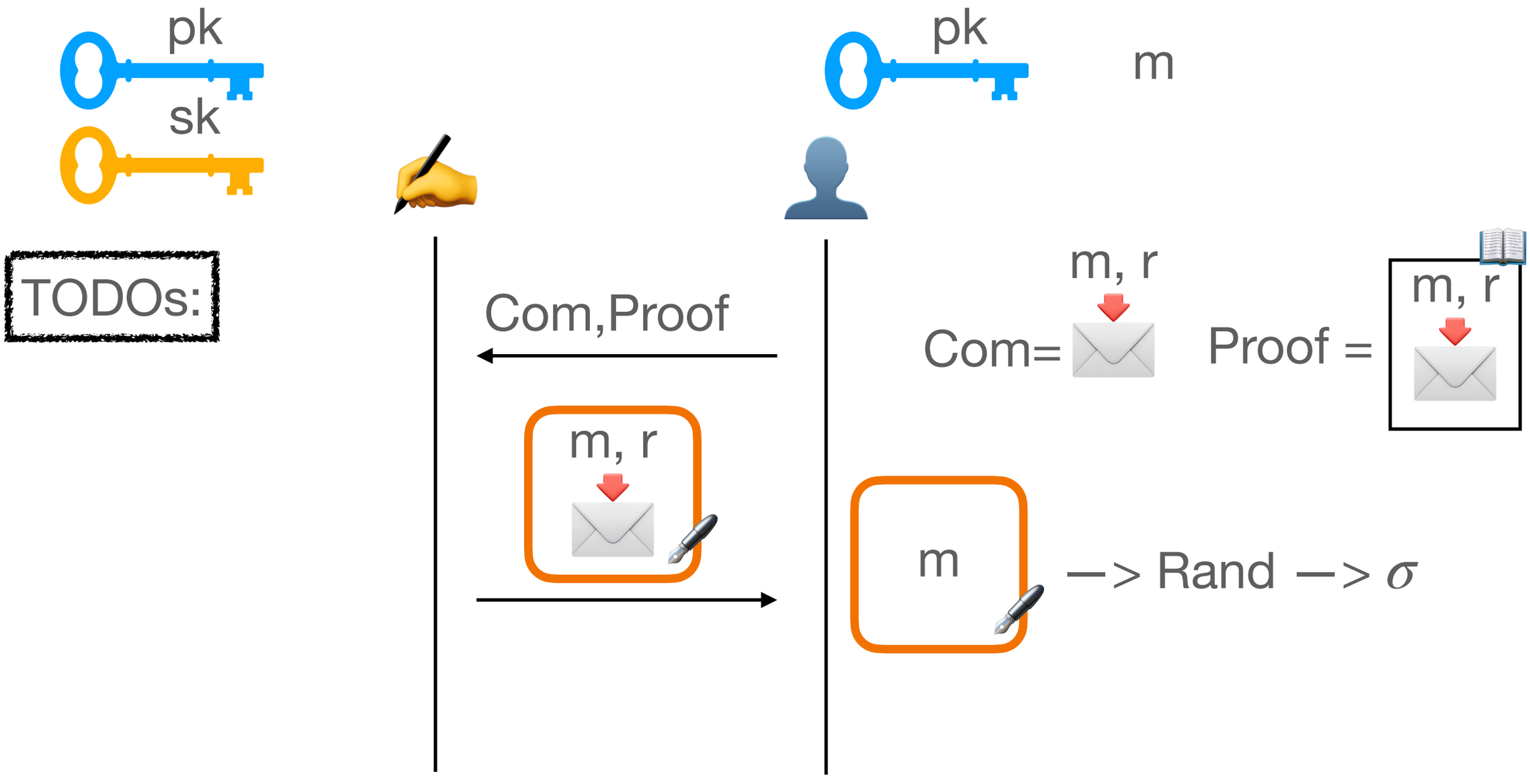
Recall: Fischlin Framework (2)

[Fischlin06, Katsumata-Reichle-Sakai23]



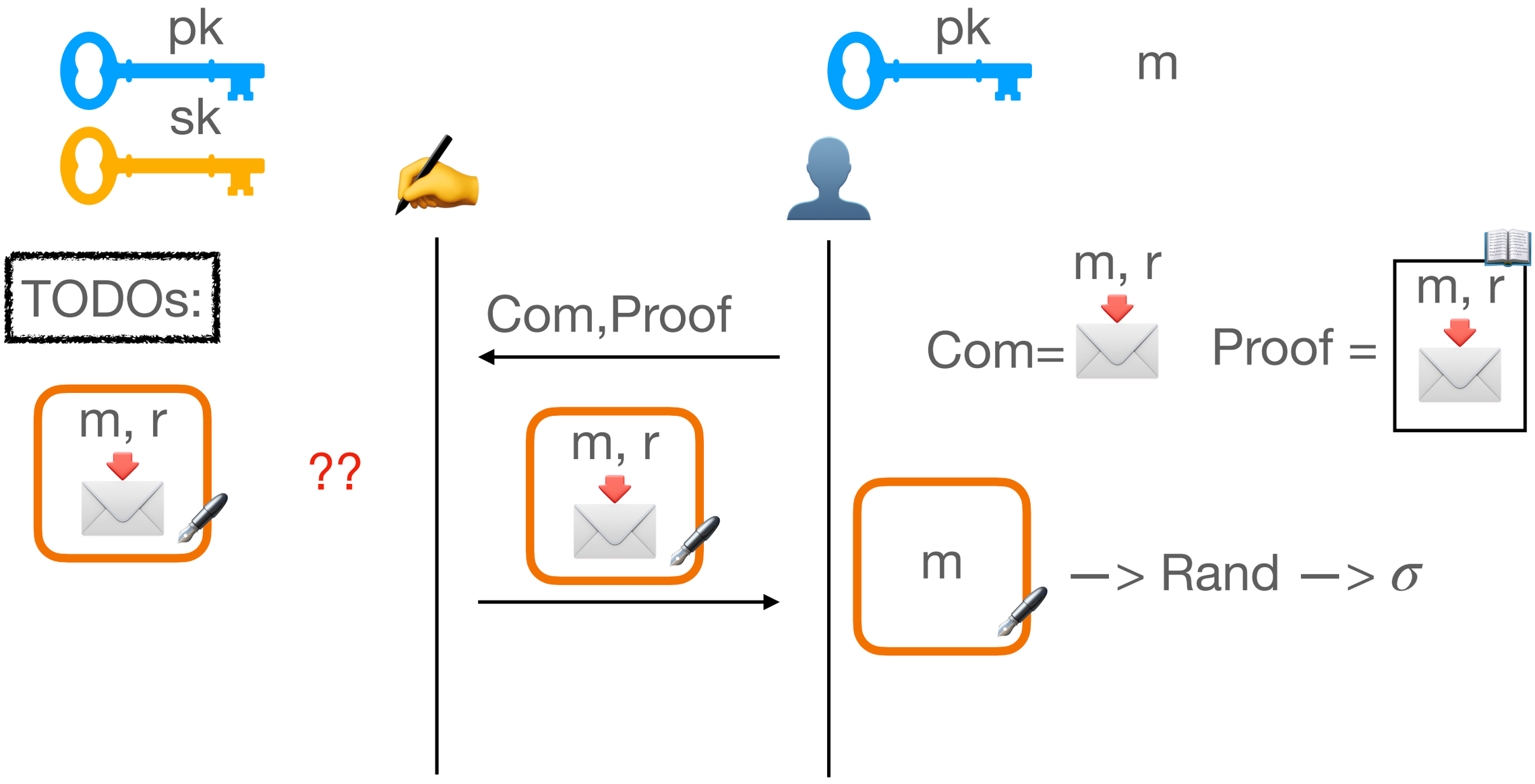
Recall: Fischlin Framework (2)

[Fischlin06, Katsumata-Reichle-Sakai23]



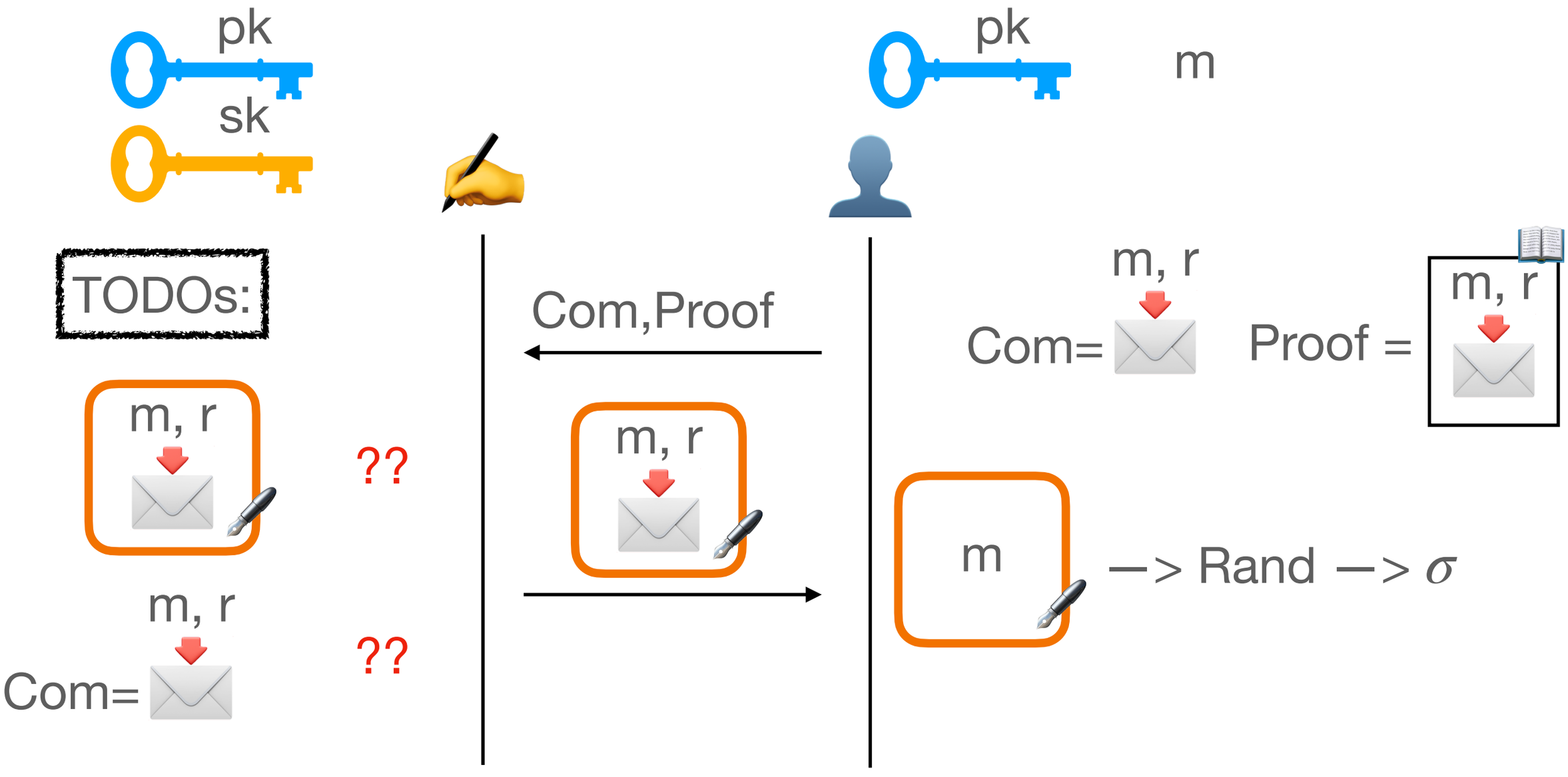
Recall: Fischlin Framework (2)

[Fischlin06, Katsumata-Reichle-Sakai23]



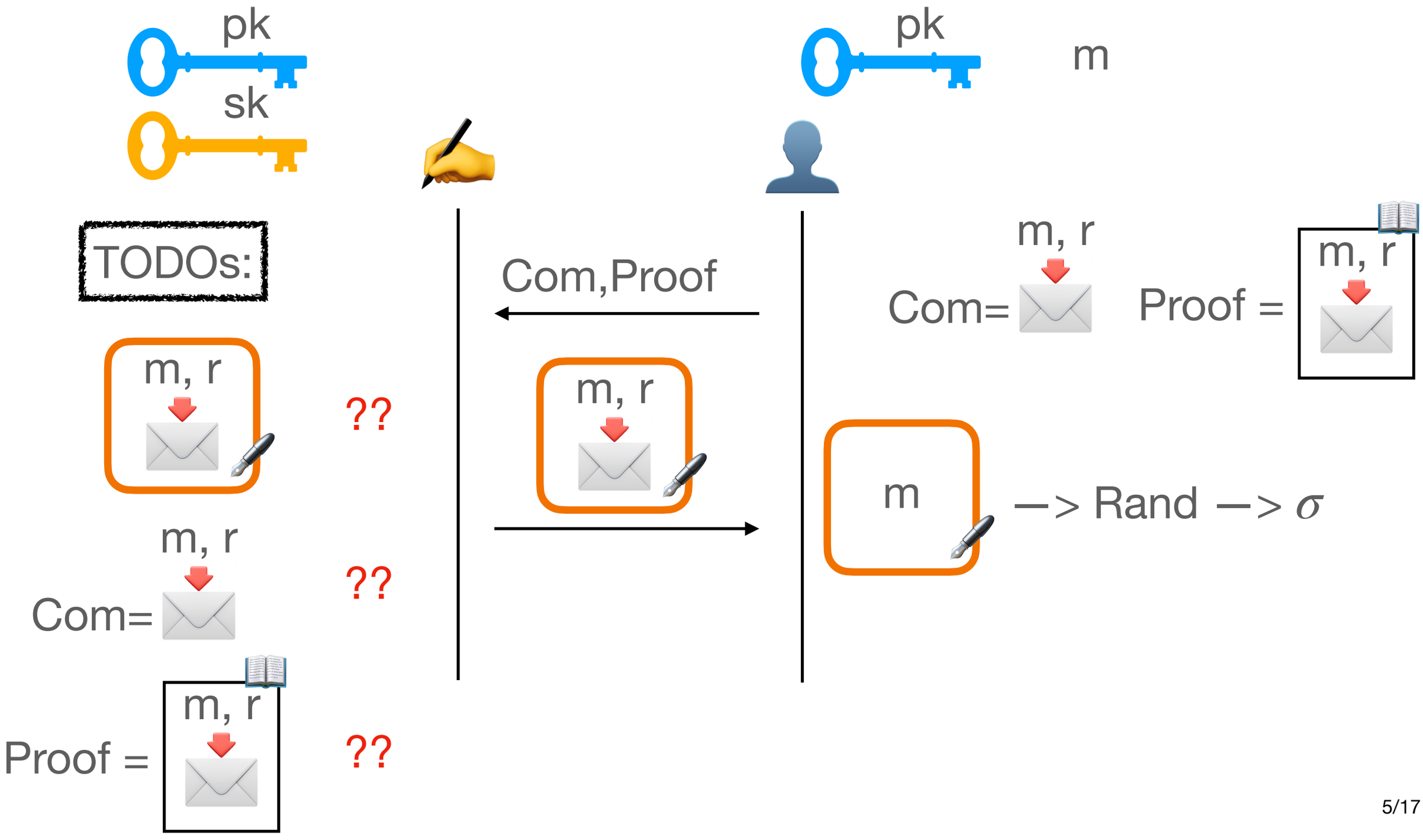
Recall: Fischlin Framework (2)

[Fischlin06, Katsumata-Reichle-Sakai23]



Recall: Fischlin Framework (2)

[Fischlin06, Katsumata-Reichle-Sakai23]



Our Signature for Fischlin

??

Our Signature for Fischlin

??

—> Modified Fischlin-Cramer-Shoup [Fischlin03,Cramer-Shoup99]
Based on sRSA

Our Signature for Fischlin

??

—> Modified Fischlin-Cramer-Shoup [Fischlin03,Cramer-Shoup99]
Based on sRSA

Properties:

(NIZK-friendly) Fixing m,
have NIZKPoK

(Setup w/ trapdoors)
Simulated signing
by trapdoors in

Our Signature for Fischlin

??

—> Modified Fischlin-Cramer-Shoup [Fischlin03,Cramer-Shoup99]
Based on sRSA

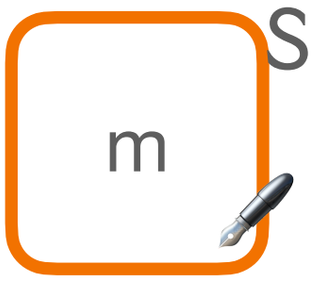
Properties:

(NIZK-friendly) Fixing m,
have NIZKPoK

(Setup w/ trapdoors)
Simulated signing
by trapdoors in

Our S_{fis} in Sect. 4

Our Commitment for Fischlin



S_{fis}

; Com= 

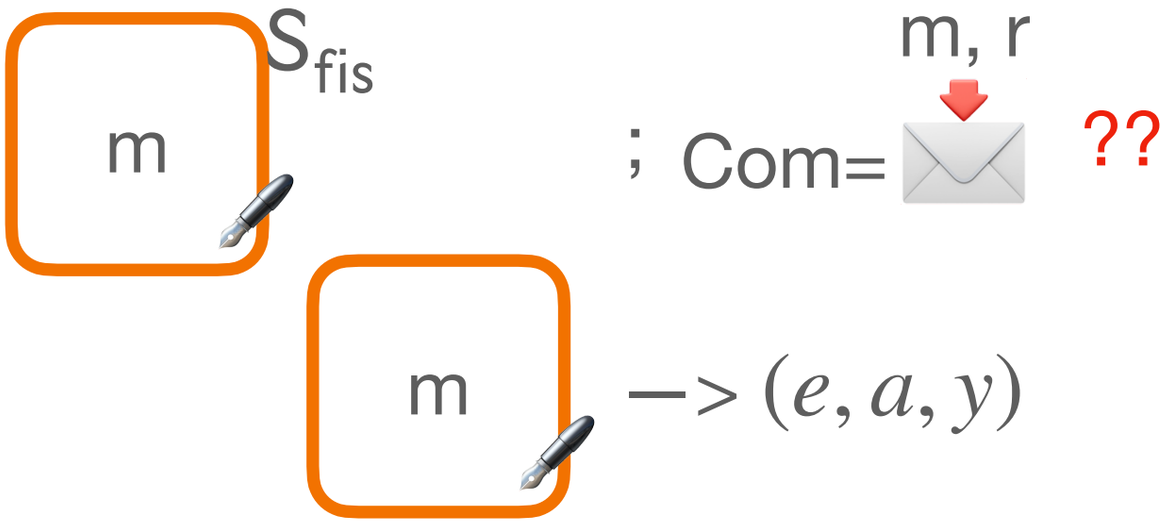
m, r

??


Req1: S_{fis}-compatible

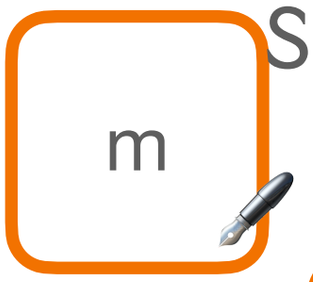

Req2: add. homomorphic

Our Commitment for Fischlin



- Req1: S_{fis}-compatible
- Req2: add. homomorphic

Our Commitment for Fischlin



S_{fis}

; Com= 

m, r

??

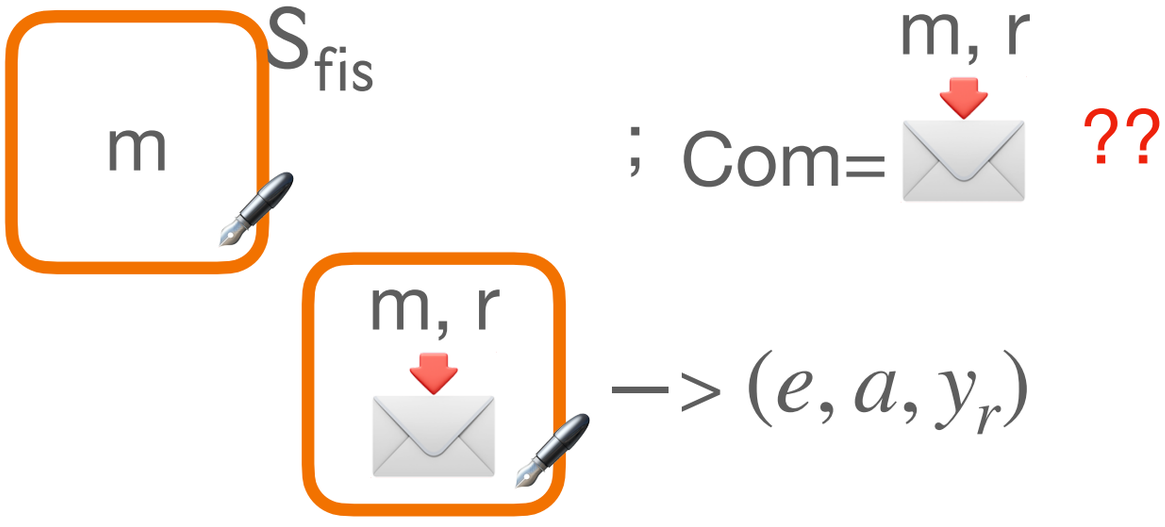
$\longrightarrow (e, a, y)$


Req1: S_{fis}-compatible


Req2: add. homomorphic

$$y^e = h \cdot h_1^a \cdot h_2^{a+m} \mod N$$

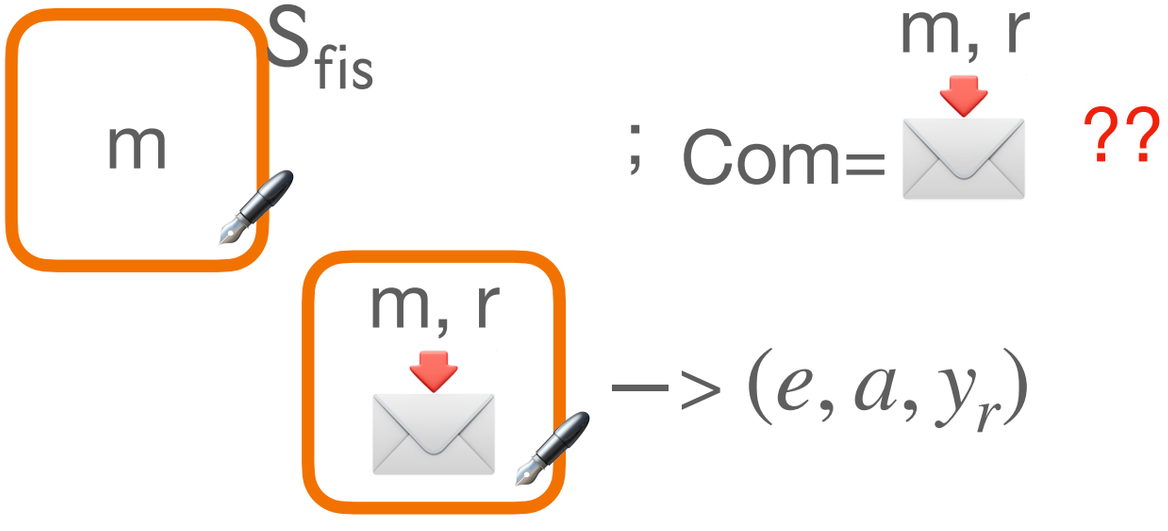
Our Commitment for Fischlin



- Req1: S_{fis}-compatible
- Req2: add. homomorphic

$$y^e = h \cdot h_1^a \cdot h_2^{a+m} \mod N$$

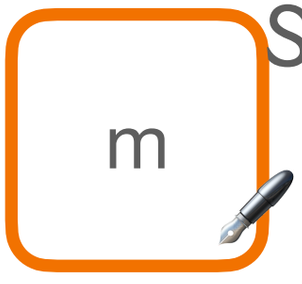
Our Commitment for Fischlin



- Req1: S_{fis}-compatible
- Req2: add. homomorphic

$$y_r^e = h \cdot h_1^a \cdot h_2^a \cdot \underbrace{h_2^m g^r} \mod N$$

Our Commitment for Fischlin



S_{fis}

; Com=

m, r

??

$\longrightarrow (e, a, y_r)$

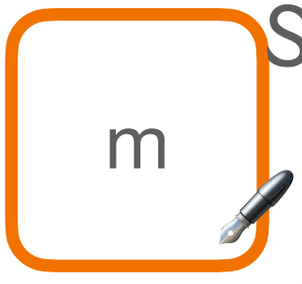
Req1: S_{fis}-compatible

Req2: add. homomorphic

$$y_r^e = h \cdot h_1^a \cdot h_2^a \cdot \underbrace{h_2^m g^r}_{\text{m, r}} \mod N$$

1st try:  $= h_2^m \cdot g^r \mod N$

Our Commitment for Fischlin



S_{fis}

; Com=

m, r

??

$\longrightarrow (e, a, y_r)$

Req1: S_{fis}-compatible

Req2: add. homomorphic

$$y_r^e = h \cdot h_1^a \cdot h_2^a \cdot \underbrace{h_2^m g^r}_{\text{m, r}} \mod N$$

1st try:  $= h_2^m \cdot g^r \mod N$

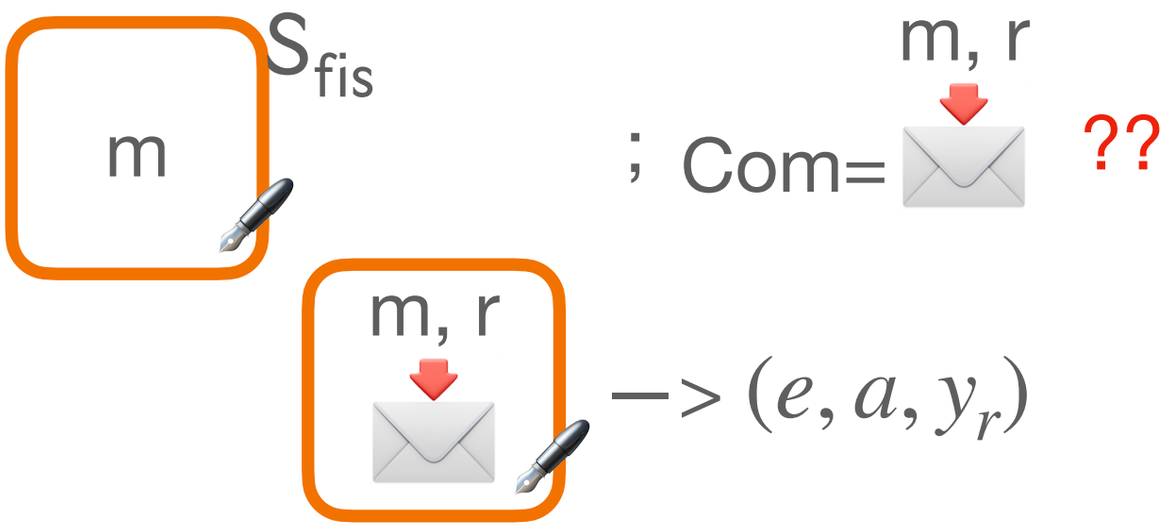
“

-

=

”

Our Commitment for Fischlin

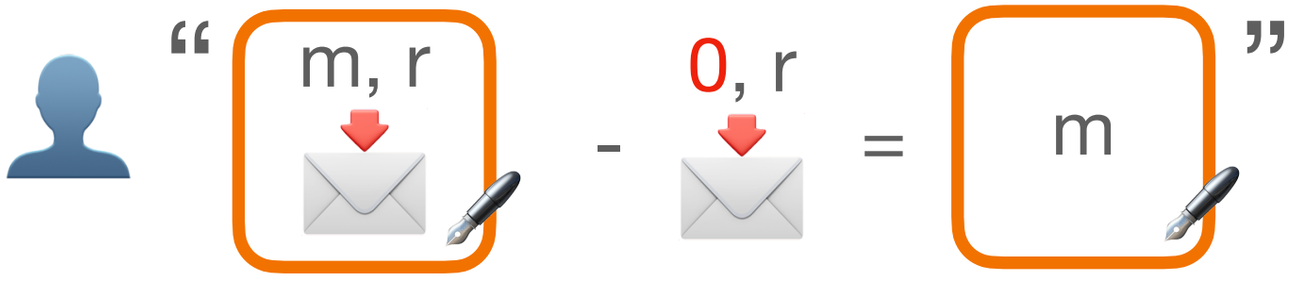


Req1: S_{fis}-compatible

Req2: add. homomorphic

$$y_r^e = y^e g^r \mod N$$

1st try: m, r $= h_2^m \cdot g^r \mod N$



Our Commitment for Fischlin

m S_{fis} ; Com= m, r ??

- Req1: S_{fis}-compatible
- Req2: add. homomorphic

m, r $\rightarrow (e, a, y_r)$

$$y_r = \boxed{\text{m}} \cdot g^{\frac{r}{e}} \mod N$$

1st try: m, r $= h_2^m \cdot g^r \mod N$

“ m, r - 0, r = m ”

Our Commitment for Fischlin

m S_{fis} ; Com= m, r ??

Req1: S_{fis}-compatible

Req2: add. homomorphic

m, r $\rightarrow (e, a, y_r)$

$$y_r = \boxed{\text{m}} \cdot g^{\frac{r}{e}} \mod N$$

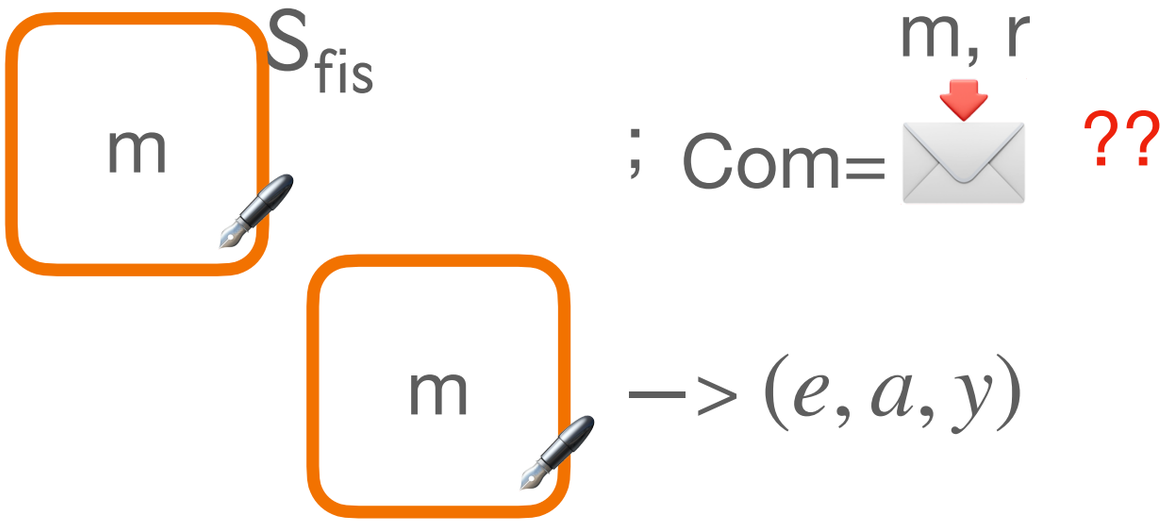
1st try: m, r $= h_2^m \cdot g^r \mod N$

“ m, r - 0, r = m ” $\Rightarrow$ m $:= y_r \cdot g^{-\frac{r}{e}} \mod N$

Our Commitment for Fischlin

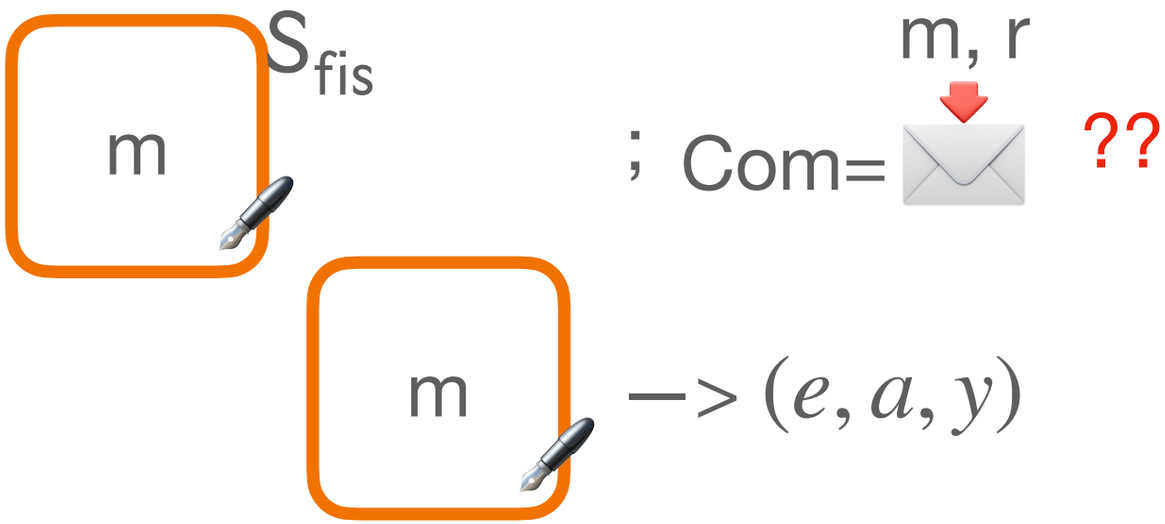
- Req1: S_{fis}-compatible
- Req2: add. homomorphic

Our Commitment for Fischlin



- Req1: S_{fis}-compatible
- Req2: add. homomorphic

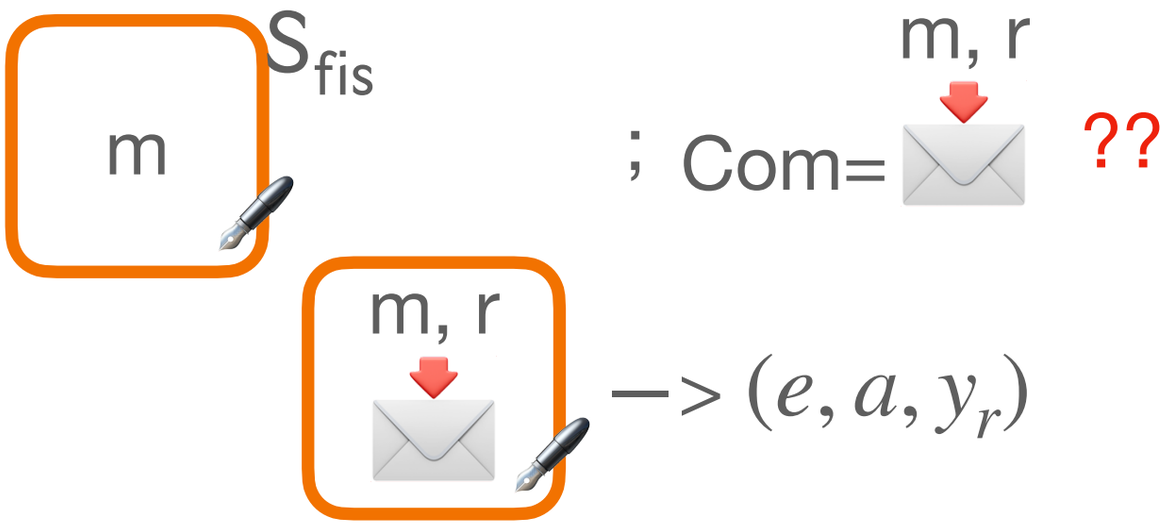
Our Commitment for Fischlin



- Req1: S_{fis}-compatible
- Req2: add. homomorphic

$$y^e = h \cdot h_1^a \cdot h_2^{a+m} \mod N$$

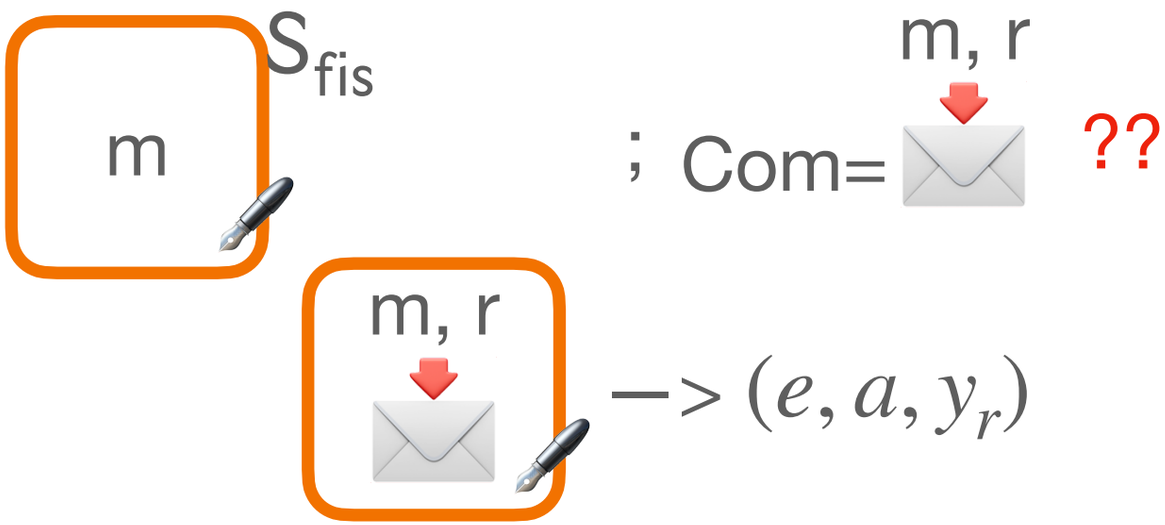
Our Commitment for Fischlin



- Req1: S_{fis}-compatible
- Req2: add. homomorphic

$$y^e = h \cdot h_1^a \cdot h_2^{a+m} \mod N$$

Our Commitment for Fischlin

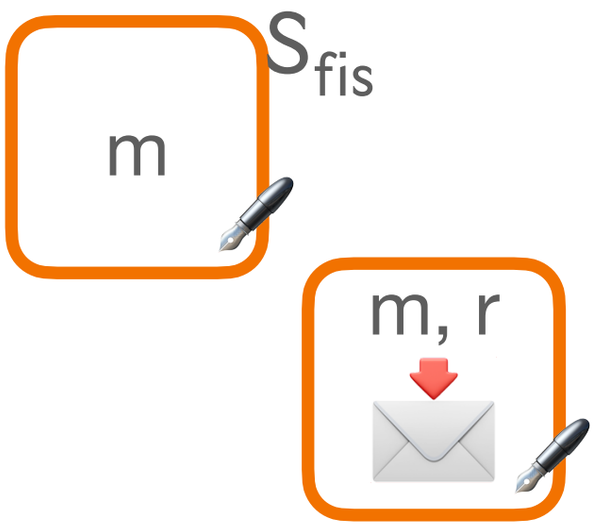


Req1: S_{fis}-compatible

Req2: add. homomorphic

$$y_r^e = h \cdot h_1^a \cdot h_2^a \cdot h_2^m g^{e \cdot r} \mod N$$

Our Commitment for Fischlin

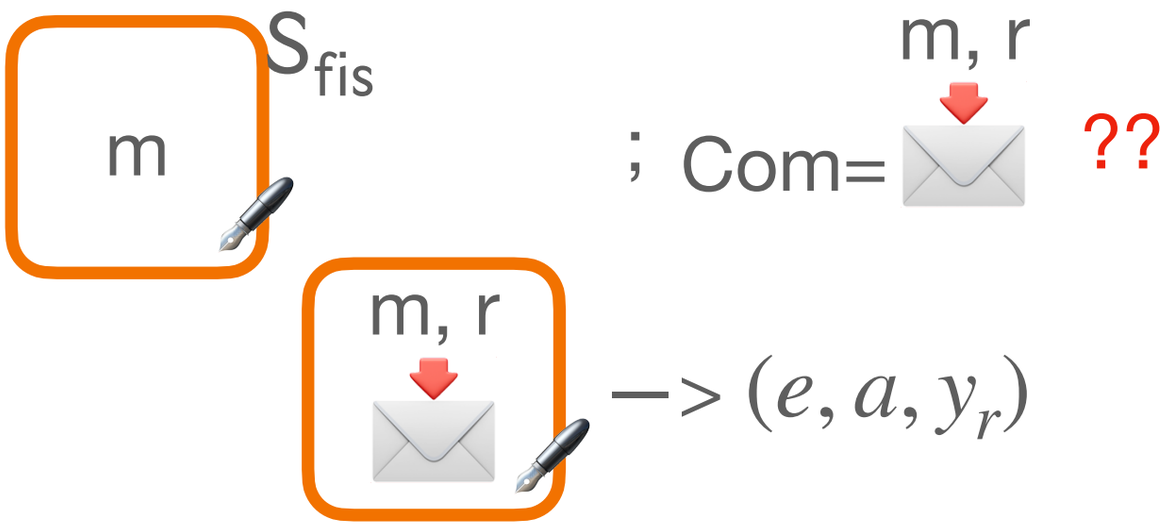


Req1: S_{fis}-compatible
Req2: add. homomorphic

$$y_r^e = h \cdot h_1^a \cdot h_2^a \cdot h_2^m g^{e \cdot r} \mod N$$

2nd try: $\overbrace{\text{m, r}}$ $= h_2^m \cdot g^{e \cdot r} \mod N$

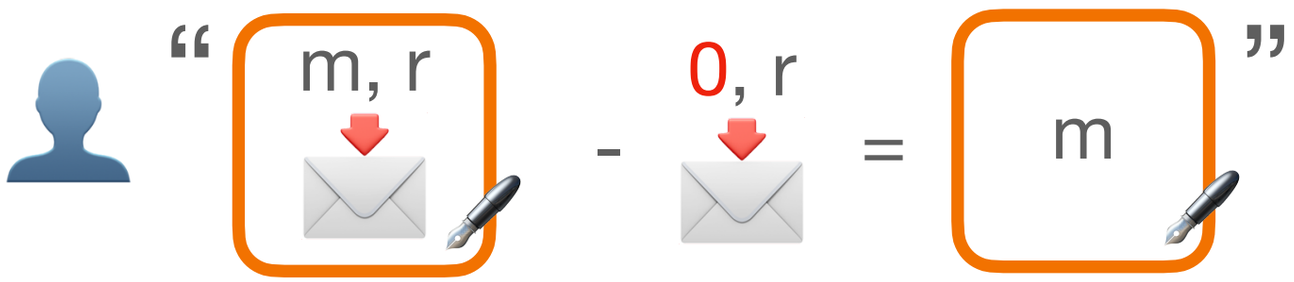
Our Commitment for Fischlin



Req1: S_{fis}-compatible
Req2: add. homomorphic

$$y_r^e = h \cdot h_1^a \cdot h_2^a \cdot h_2^m g^{e \cdot r} \mod N$$

2nd try: $\overbrace{\text{m, r}}$ $= h_2^m \cdot g^{e \cdot r} \mod N$



Our Commitment for Fischlin

S_{fis}

; Com=

m, r

??

$\rightarrow (e, a, y_r)$

- Req1: S_{fis}-compatible
- Req2: add. homomorphic

$$y_r^e = y^e g^{er} \mod N$$

2nd try: m, r $= h_2^m \cdot g^{e \cdot r} \mod N$

“

-

=

”

Our Commitment for Fischlin

m S_{fis} ; Com= m, r ??

Req1: S_{fis}-compatible

Req2: add. homomorphic

m, r —> (e, a, y_r)

$$y_r = \boxed{\text{m}} \cdot g^r \mod N$$

2nd try: m, r $= h_2^m \cdot g^{e \cdot r} \mod N$

“ m, r - 0, r = m ”

Our Commitment for Fischlin

m S_{fis} ; Com= m, r ??

Req1: S_{fis}-compatible

Req2: add. homomorphic

m, r —> (e, a, y_r)

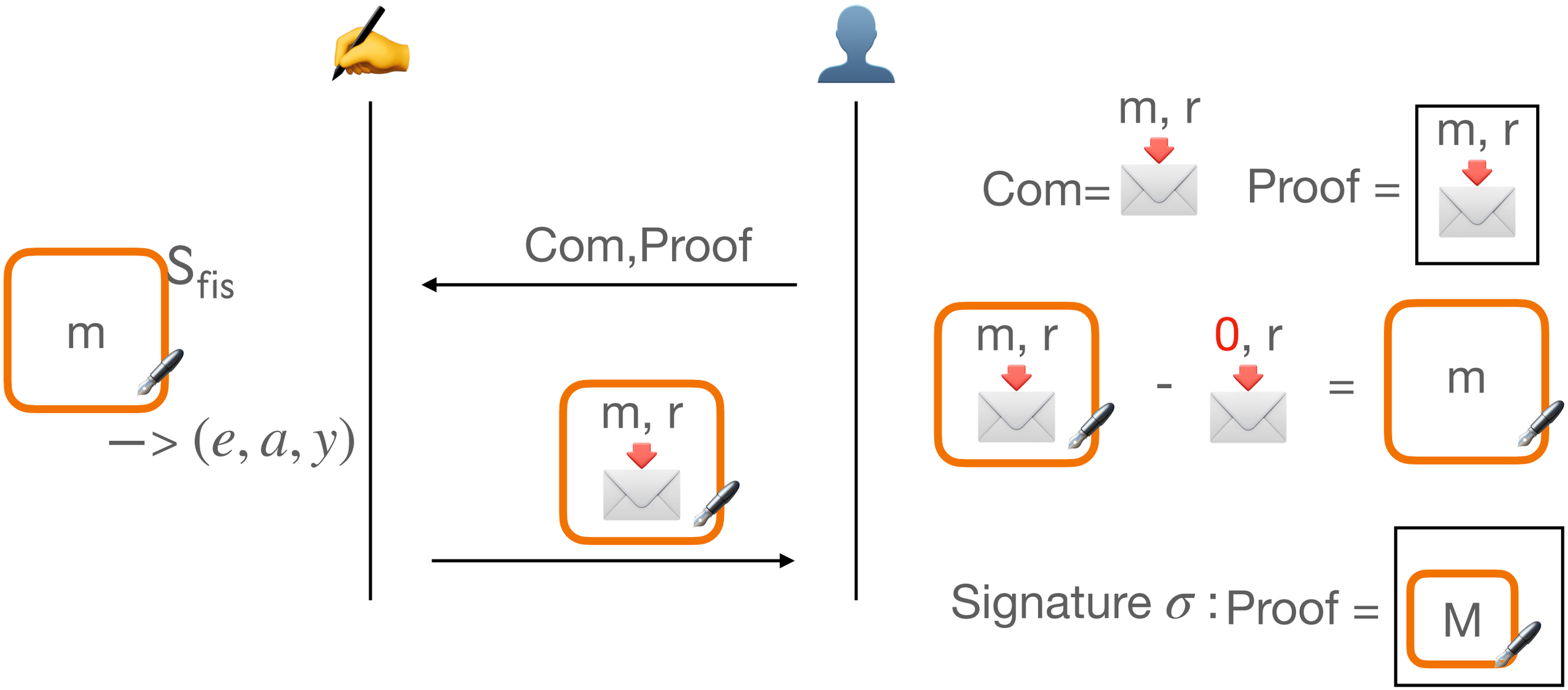
$$y_r = \boxed{\text{m}} \cdot g^r \mod N$$

2nd try: m, r $= h_2^m \cdot g^{e \cdot r} \mod N$

“ m, r - 0, r = m ” $\Rightarrow$ m $:= y_r \cdot g^{-r} \mod N$

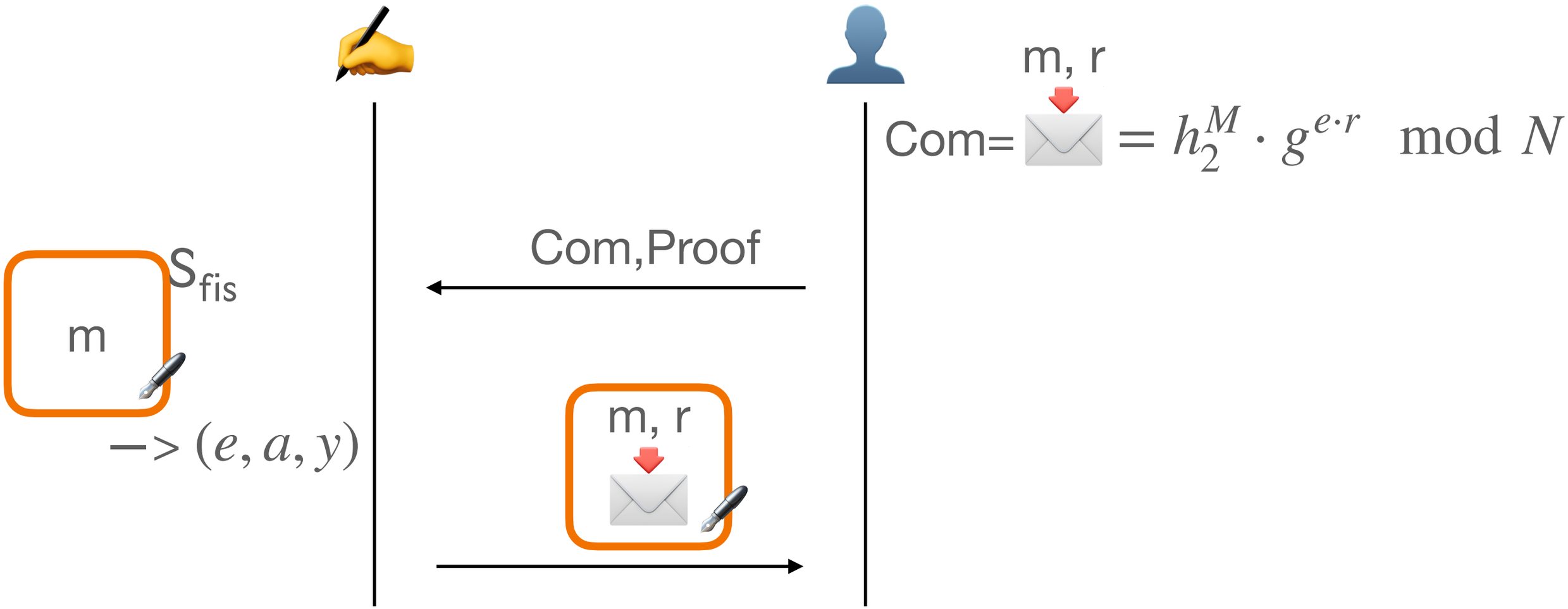
Recall: Fischlin Framework

Dealing with e from S_{fis}



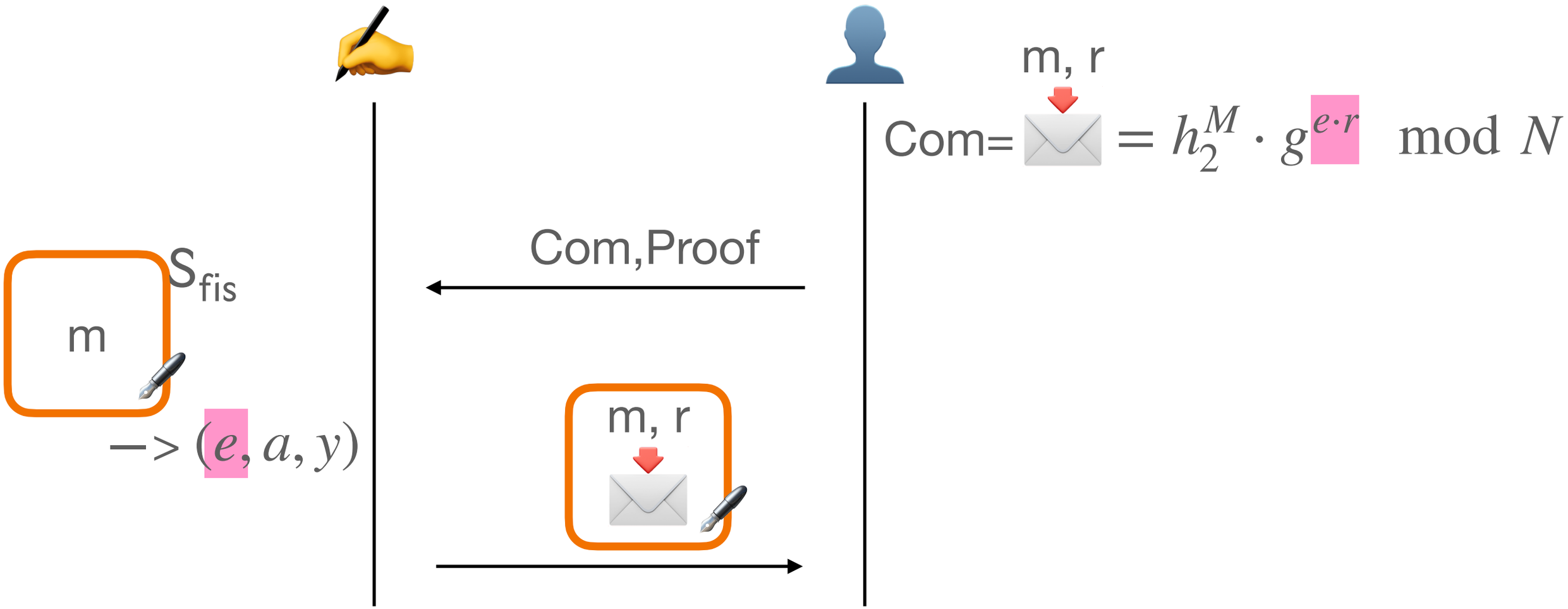
Recall: Fischlin Framework

Dealing with e from S_{fis}



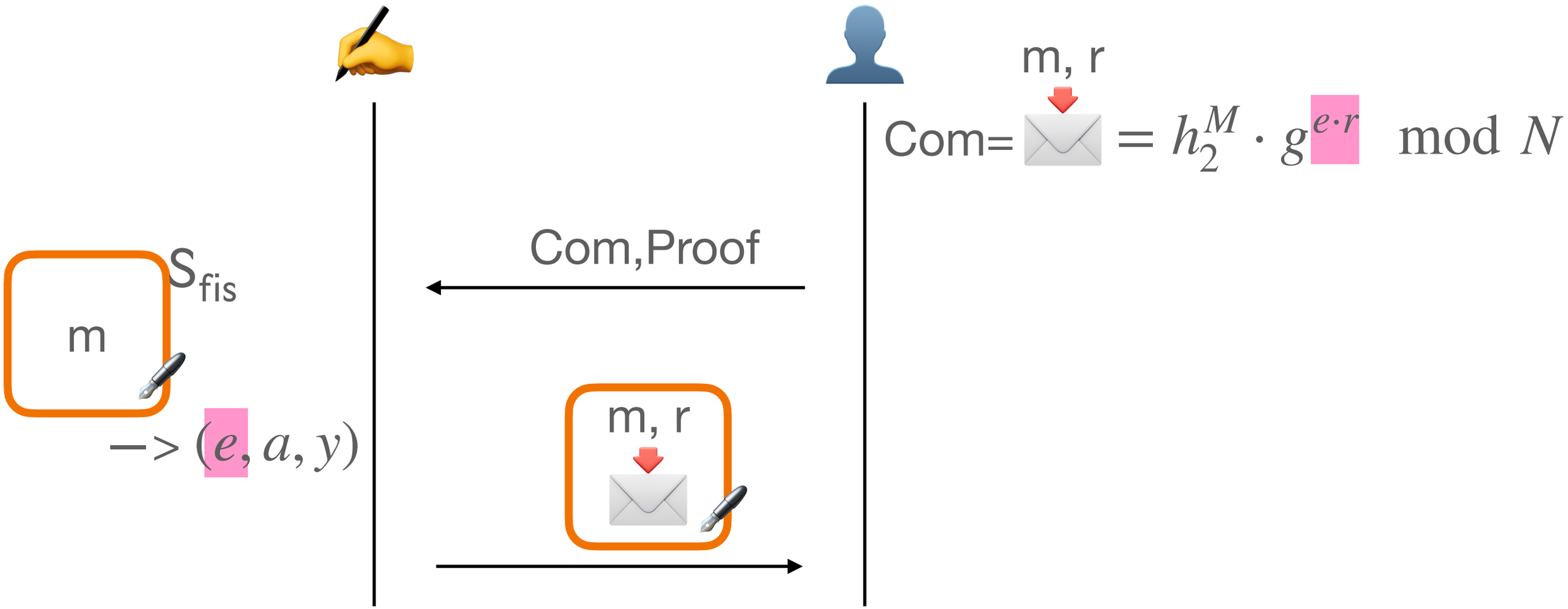
Recall: Fischlin Framework

Dealing with e from S_{fis}



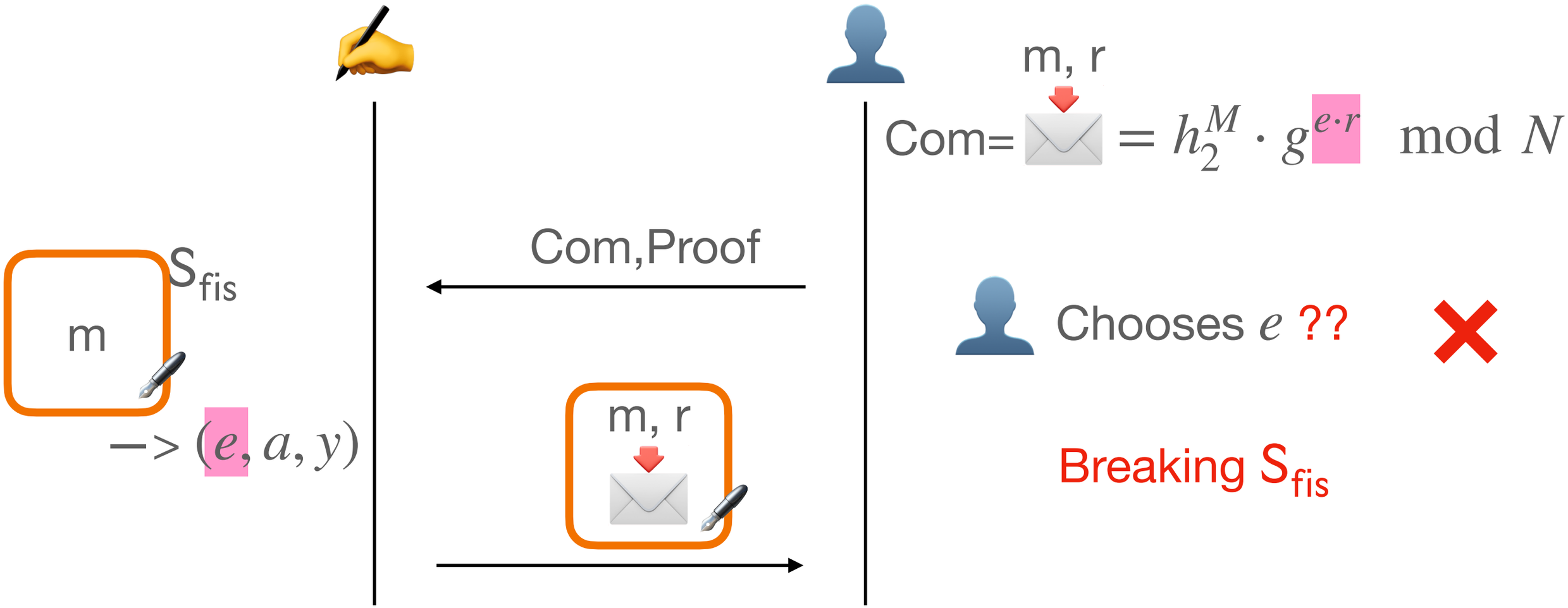
Recall: Fischlin Framework

Dealing with e from S_{fis}



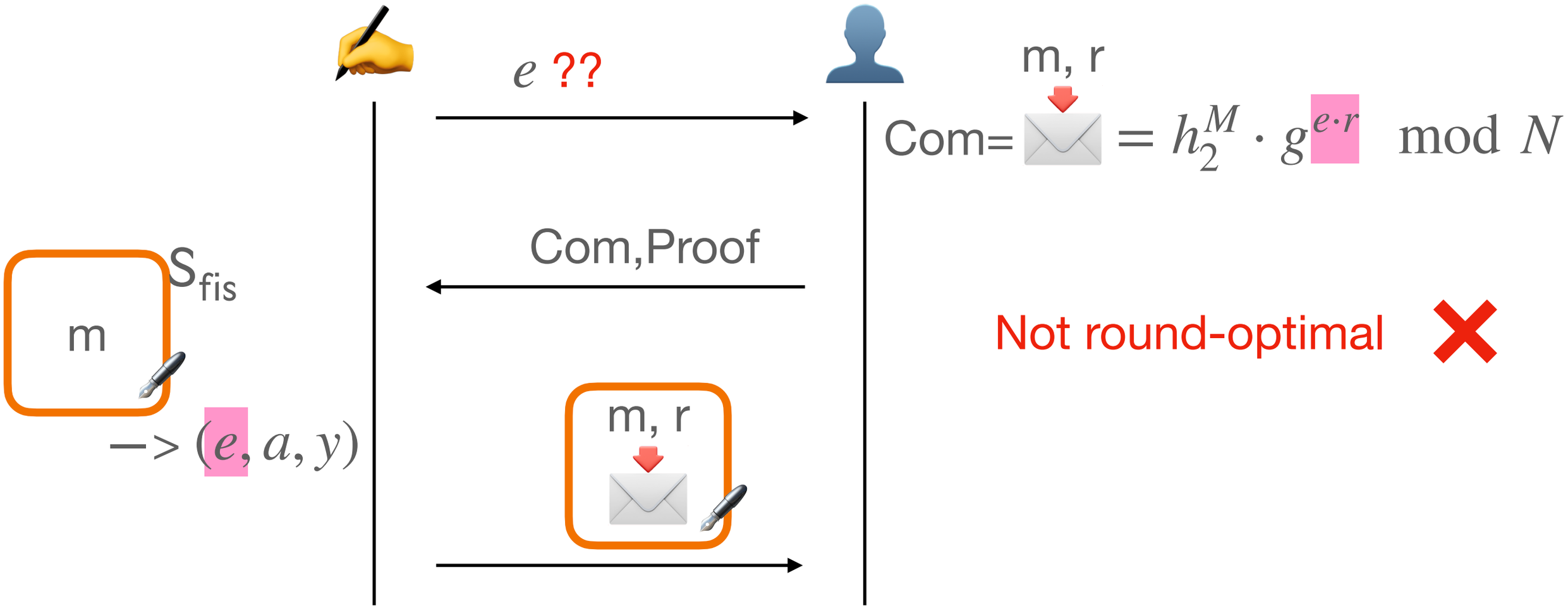
Recall: Fischlin Framework

Dealing with e from S_{fis}



Recall: Fischlin Framework

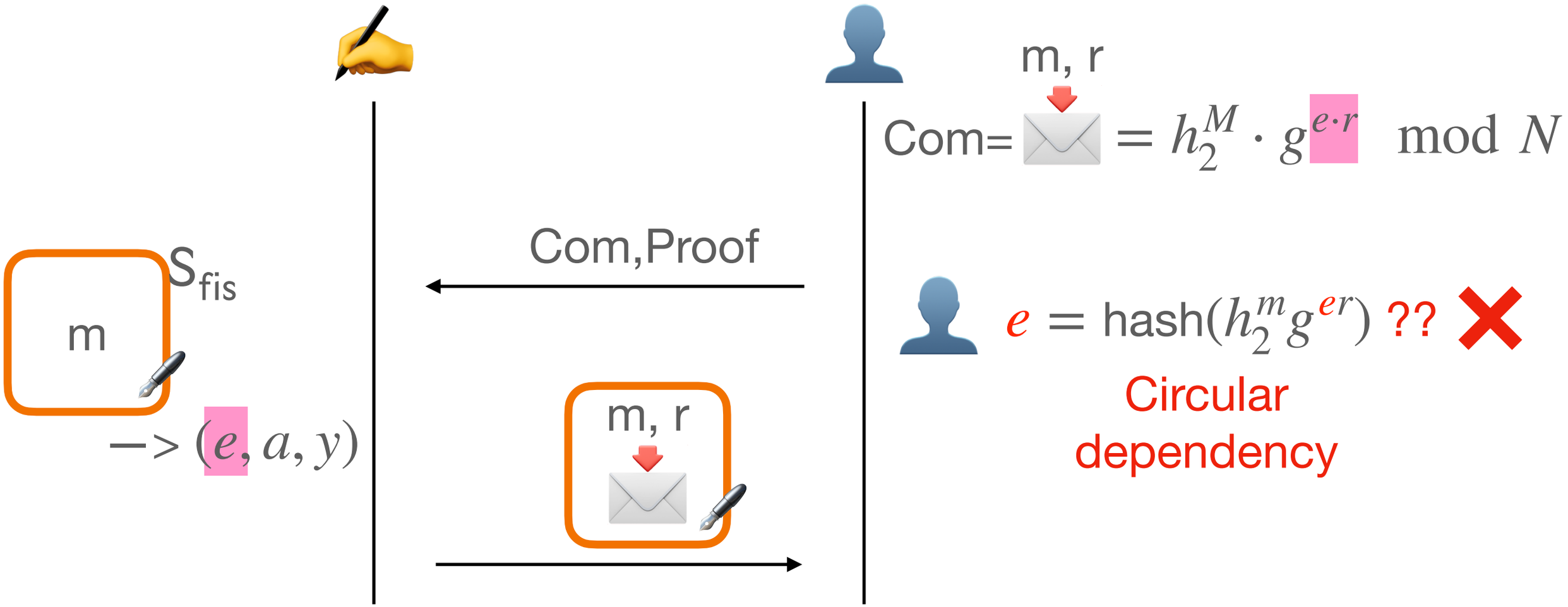
Dealing with e from S_{fis}



Not round-optimal ❌

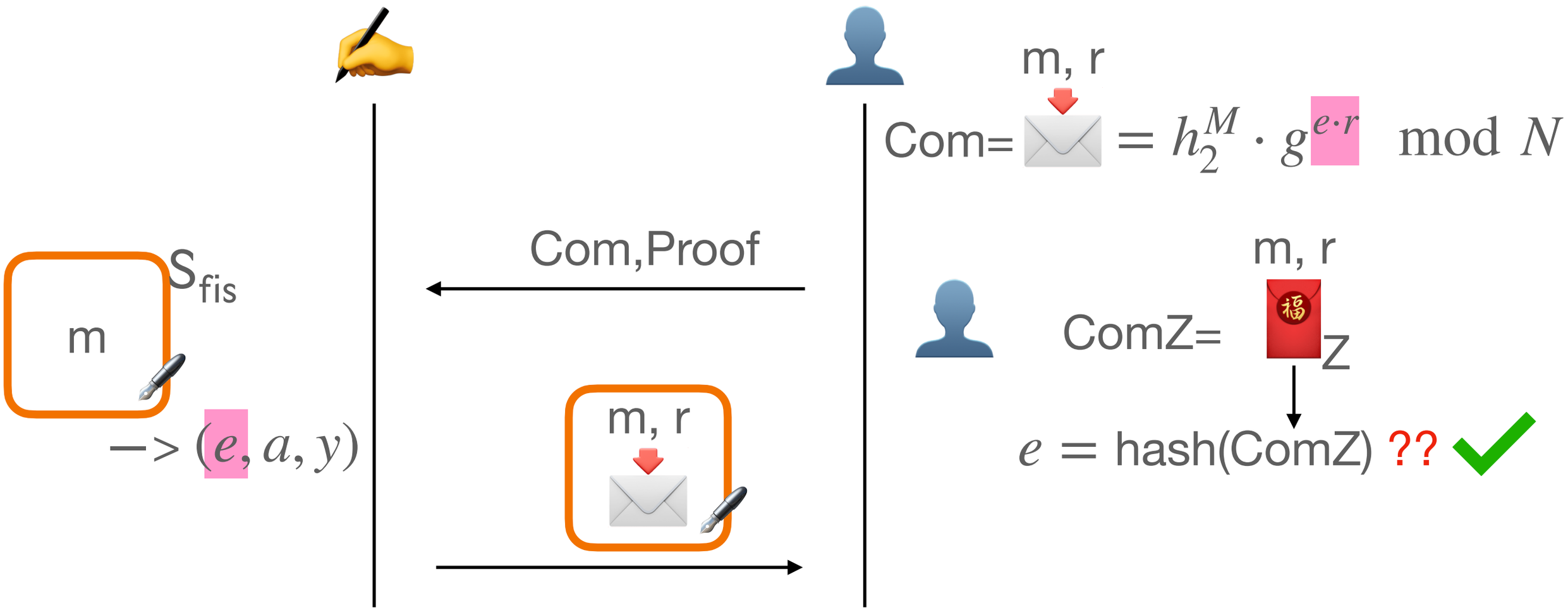
Recall: Fischlin Framework

Dealing with e from S_{fis}



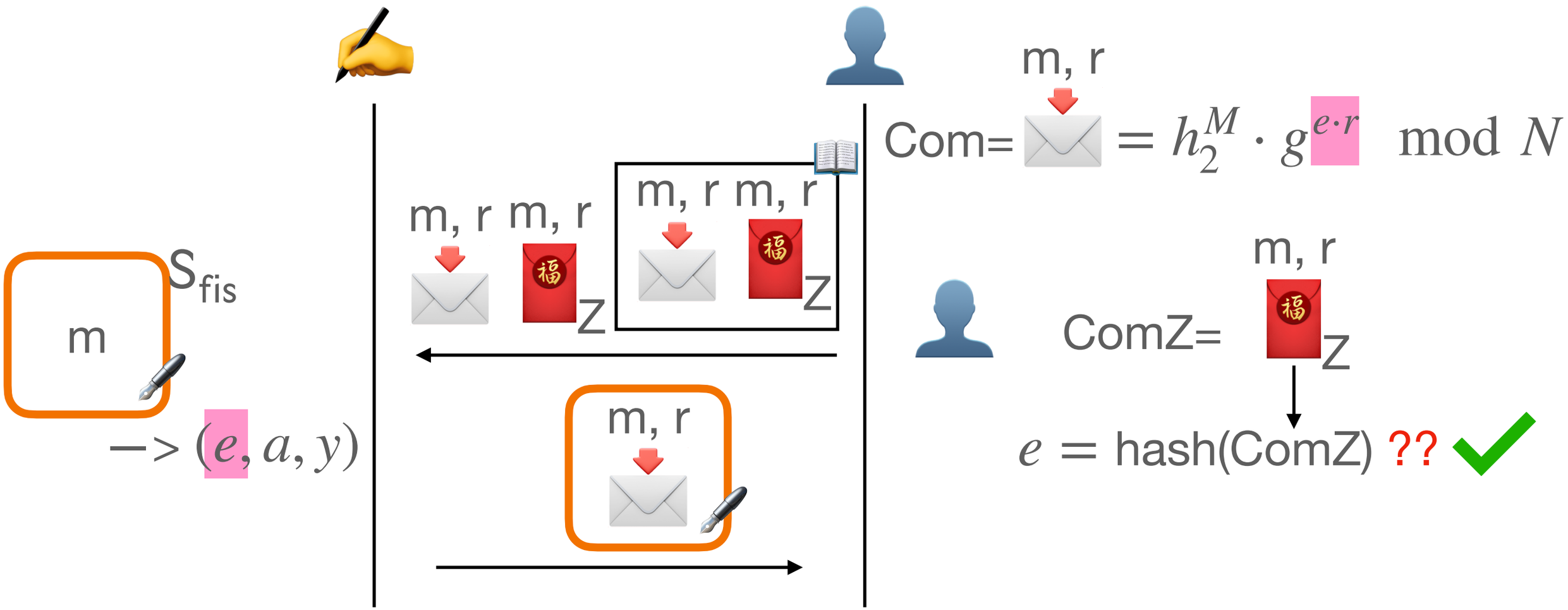
Recall: Fischlin Framework

Dealing with e from S_{fis}



Recall: Fischlin Framework

Dealing with e from S_{fis}

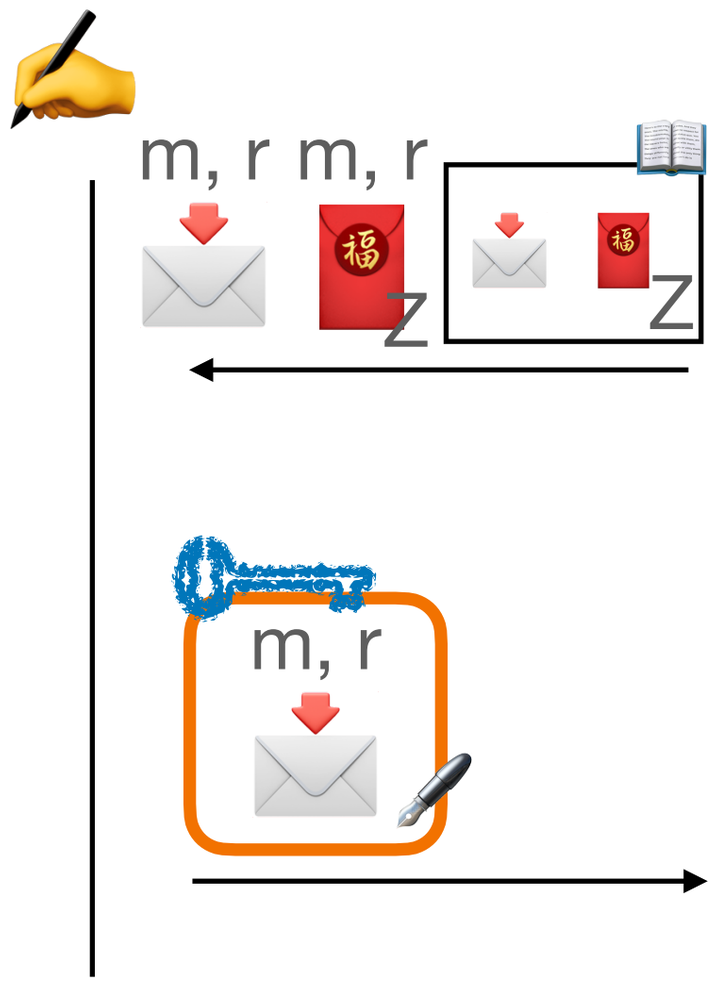


One-More Unforgeability (1)

Simulating signing queries

(Setup w/ trapdoors)

Simulated signing by trapdoors in

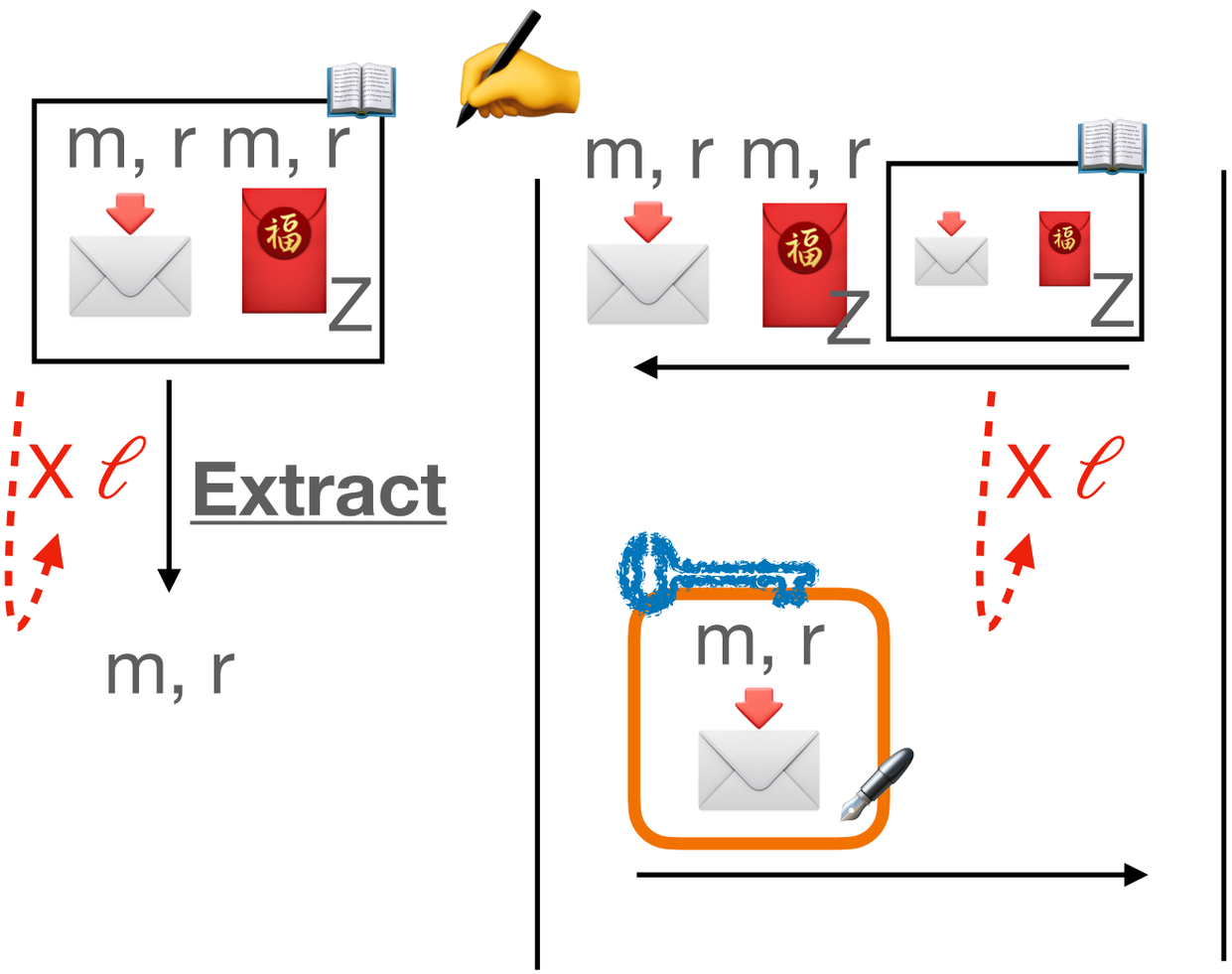


One-More Unforgeability (1)

Simulating signing queries

(Setup w/ trapdoors)
Simulated signing by trapdoors in sim-pk

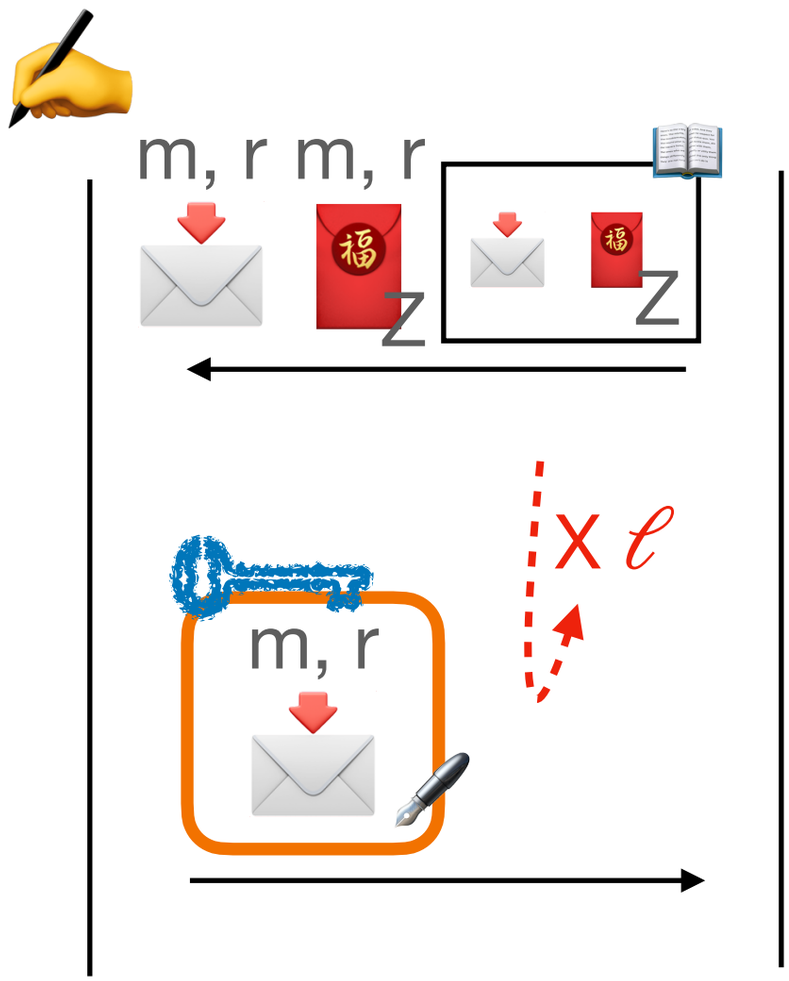
(Online-extractable)



One-More Unforgeability (2)

Exploit the forgery

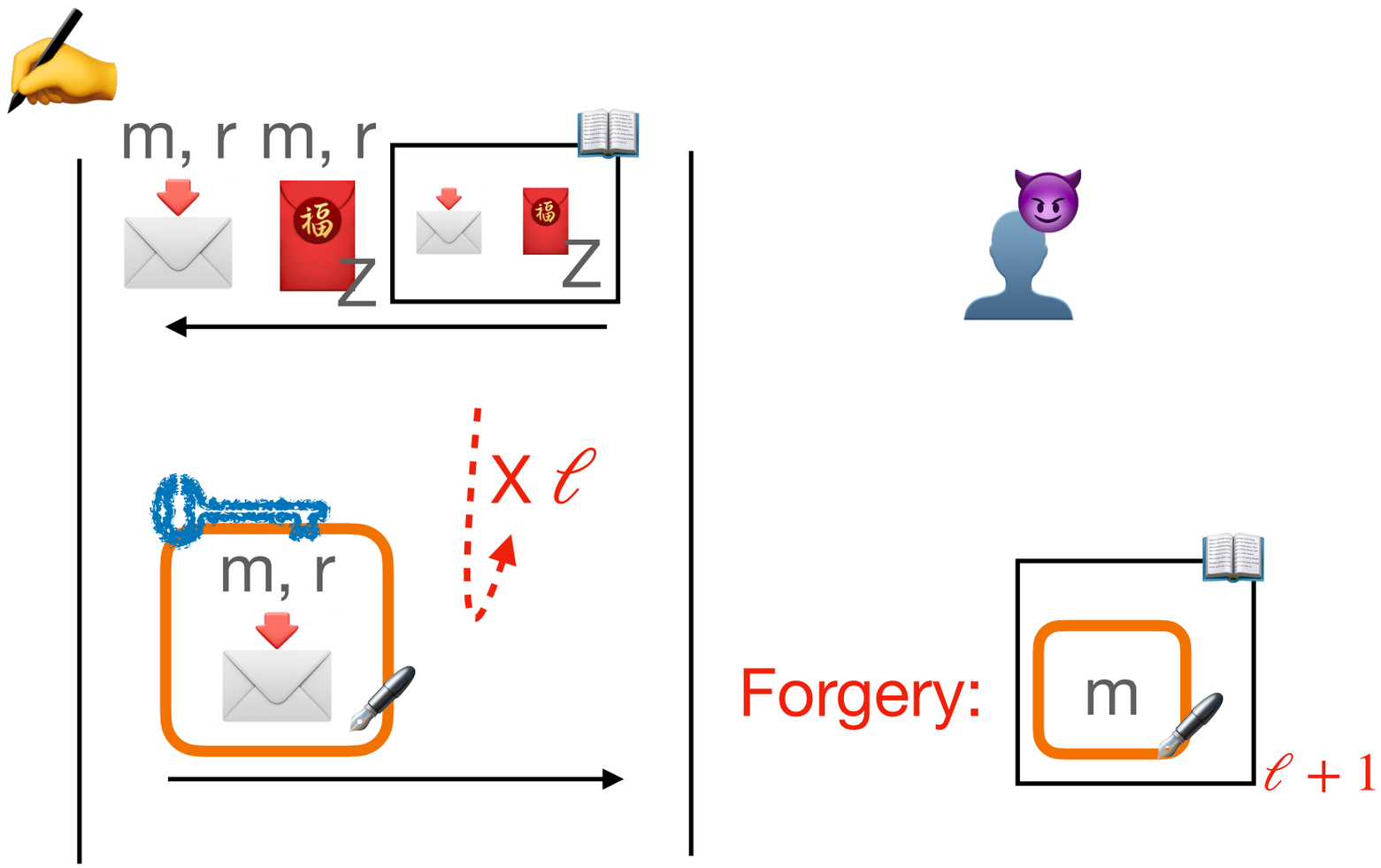
(NIZK-friendly)
Fixing m,
have NIZKPoK



One-More Unforgeability (2)

Exploit the forgery

(NIZK-friendly)
Fixing m,
have NIZKPoK



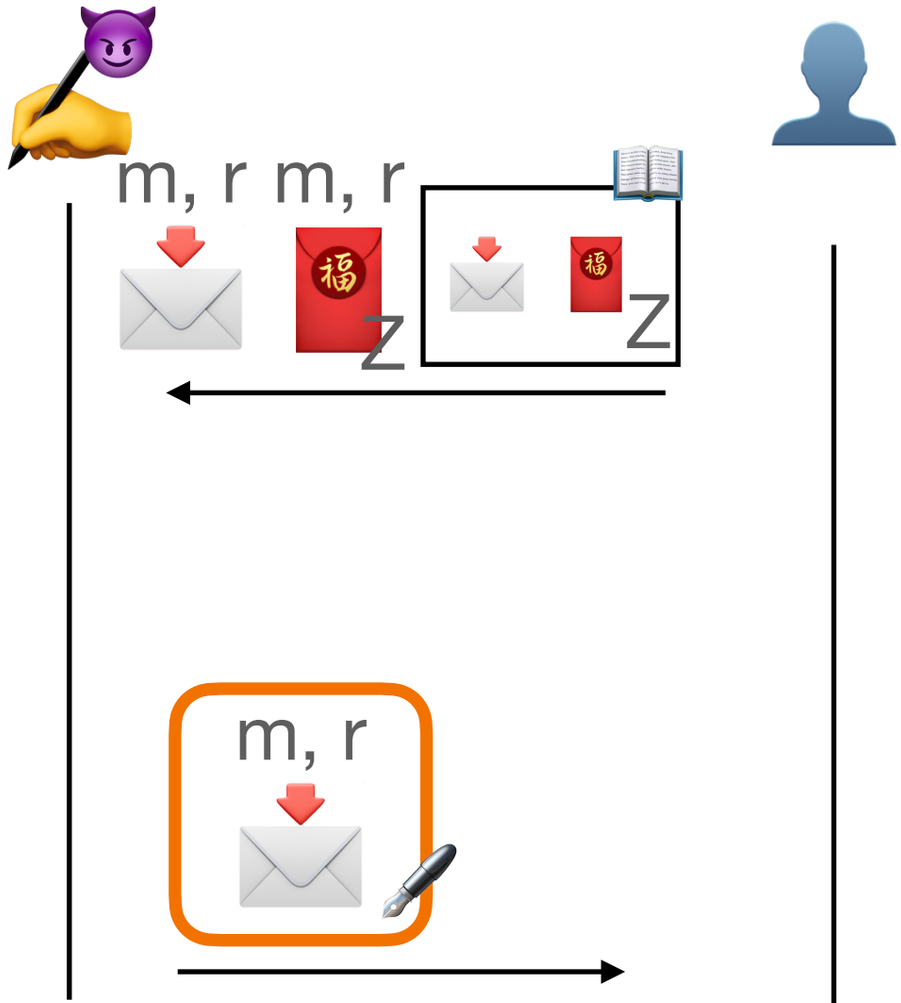
(Rewinding) **ExtractFis**

(e, a, y)

Malicious Blindness (1)

Problems from Malicious Signer - Commitments Not Hiding

pk $= (N, h, h_1, h_2)$; crs



Malicious Blindness (1)

Problems from Malicious Signer - Commitments Not Hiding

pk $= (N, h, h_1, h_2)$; crs

m, r

$= h_2^m \cdot g^{e \cdot r} \mod N$

m, r m, r

Z Z

m, r

m, r

N

m, r

Z

NOT HIDING !!

m, r

Malicious Blindness (2)

Commitments Not Hiding - Our Solution

m, r

$$= h_2^m \cdot g^{e \cdot r} \mod N$$

😈

Malicious Blindness (2)

Commitments Not Hiding - Our Solution

m, r

$= h_2^m \cdot g^{e \cdot r} \mod N$

proves

subversion-SND

$\langle h_2 \rangle = \langle g \rangle$

Malicious Blindness (2)

Commitments Not Hiding - Our Solution

m, r

$$= h_2^m \cdot g^{e \cdot r} \mod N$$


proves

subversion-SND

$$\langle h_2 \rangle = \langle g \rangle$$

$$\Rightarrow \quad \langle h_2 \rangle = \langle g^e \rangle \mod N$$

e prime, arbitrary N

Malicious Blindness (2)

Commitments Not Hiding - Our Solution

m, r

$$= h_2^m \cdot g^{e \cdot r} \mod N$$

😈 proves

subversion-SND

$$\langle h_2 \rangle = \langle g \rangle$$

$$\Rightarrow \quad \langle h_2 \rangle = \langle g^e \rangle \mod N$$

e prime, arbitrary N 😈

$$\Rightarrow$$

Large enough r gives hiding

Malicious Blindness (2)

Commitments Not Hiding - Our Solution

m, r

$= h_2^m \cdot g^{e \cdot r} \mod N$

Compact Commitments In Arbitrary Groups

proves subversion-SND

$\langle h_2 \rangle = \langle g \rangle$

$\Rightarrow$

$\langle h_2 \rangle = \langle g^e \rangle \mod N$

e prime, arbitrary N

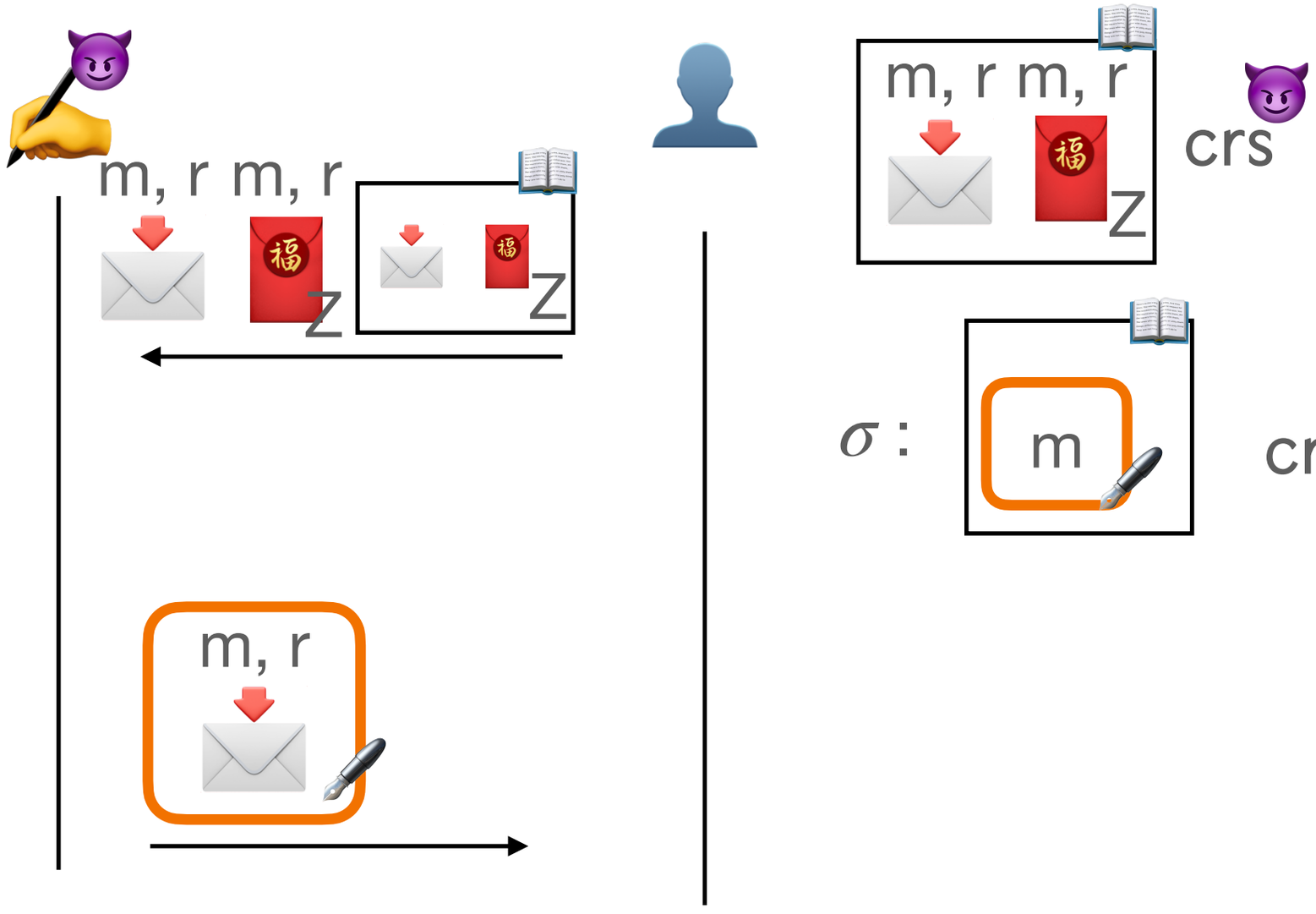
$\Rightarrow$

Large enough r gives hiding

Malicious Blindness (2)

Problems from Malicious Signer - NIZKs Not ZK

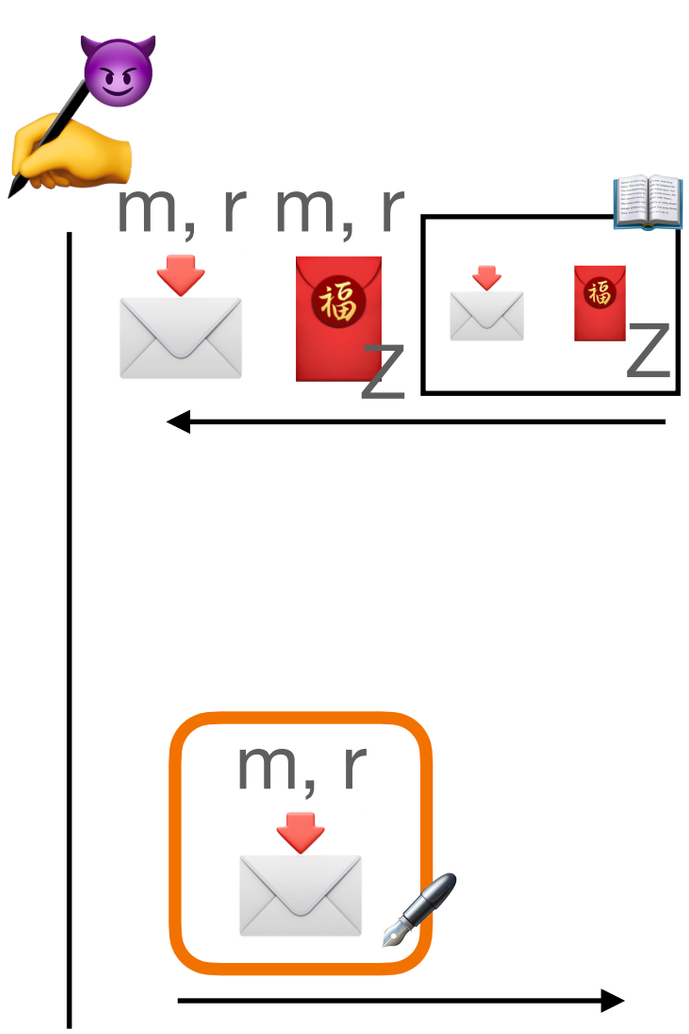
pk $= (N, h, h_1, h_2)$; crs

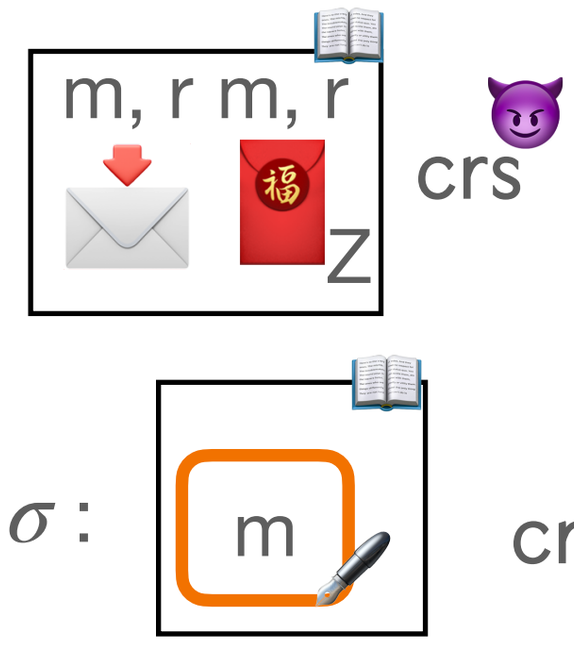


Malicious Blindness (2)

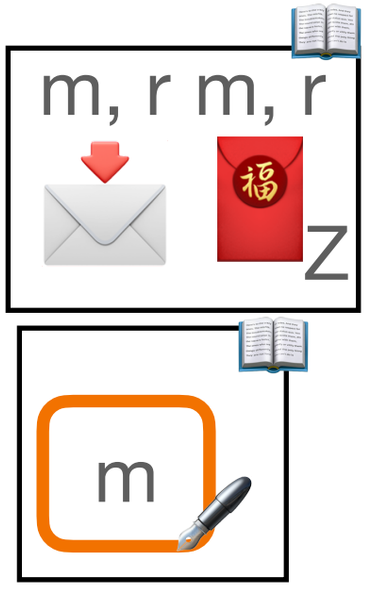
Problems from Malicious Signer - NIZKs Not ZK

pk $= (N, h, h_1, h_2)$; crs





NOT ZK !!



Malicious Blindness (4)

NIZKs Not ZK - Solution

Subversion-ZK in ROM: 😈

1. ZK stills holds with crs
2. ROM to embed
   fresh trapdoors

Malicious Blindness (4)

NIZKs Not ZK - Solution

Subversion-ZK in ROM: 😈

1. ZK stills holds with crs
2. ROM to embed
   fresh trapdoors

😈
crs has

Malicious Blindness (4)

NIZKs Not ZK - Solution

Subversion-ZK in ROM: 😈

1. ZK stills holds with crs
2. ROM to embed fresh trapdoors

unif-rs : prime-order structure for S-ZK in ROM

😈 crs has

😈 struct-rs link the different orders of the groups, e.g. 📩 in $\mathbb{Z}_N^*$, 🧧 Z in $\mathbb{G}$;

Also, we can use efficient RSA-based RPs.

Malicious Blindness (4)

NIZKs Not ZK - Solution

Relaxed Range Proofs w/ Subversion-ZK

Subversion-ZK in ROM: 😈

1. ZK stills holds with crs
2. ROM to embed fresh trapdoors

unif-rs : prime-order structure for S-ZK in ROM

😈 crs has

😈 struct-rs link the different orders of the groups,

e.g. 📩 in $\mathbb{Z}_N^*$, 🧧 Z in $\mathbb{G}$;

Also, we can use efficient RSA-based RPs.

Conclusion

New Blind Signatures:

- *Round-Optimal*
- *Blindness Against Malicious Signers*
- *In ROM from DDH & sRSA*

Conclusion

New Blind Signatures:

- *Round-Optimal*
- *Blindness Against Malicious Signers*
- *In ROM from DDH & sRSA*

1. *Reduce vk's size?*
2. *Better efficiency?*
3. *Remove the need of sRSA?*
4. *Non-hybrid constructions?*