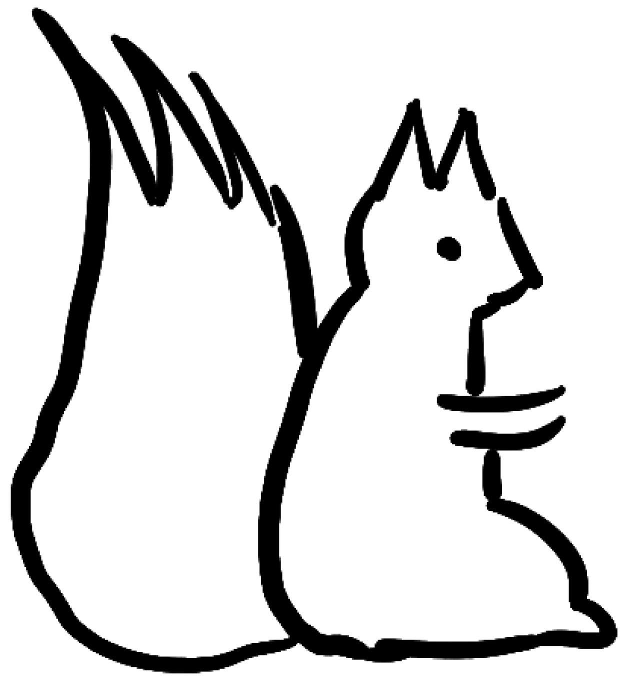
The Abe-Okamoto Partially Blind Signature Scheme Revisited

Julia Kastner
ETH Zürich
Switzerland

Julian Loss
CISPA
Germany

Jiayu Xu
Oregon State University
USA

Partially Blind Signatures [AF96]



e.g. 1$, "best before 24.12."

info

sk — pk, m

$\xrightarrow{R}$

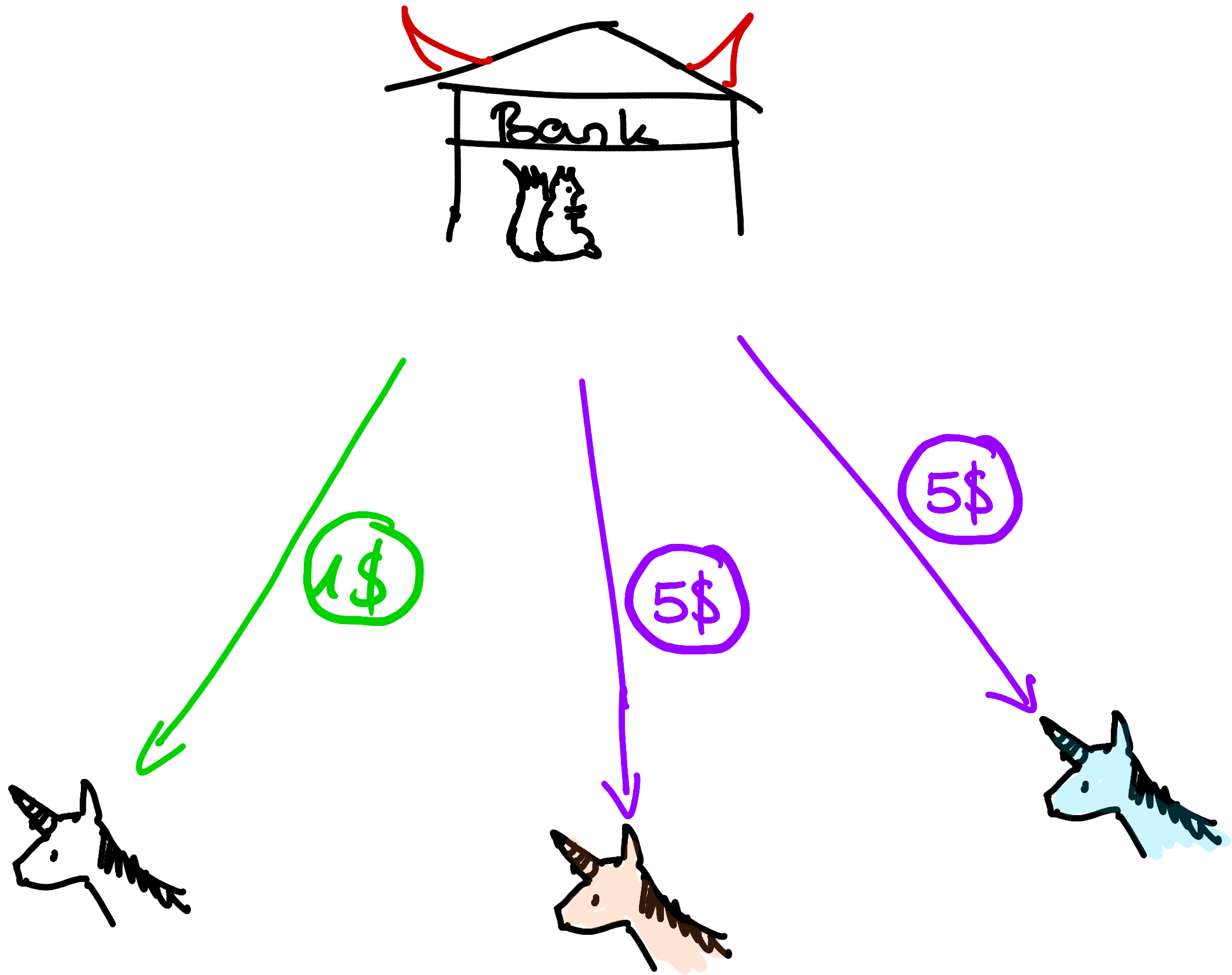
$\xleftarrow{e}$

$\xrightarrow{s}$

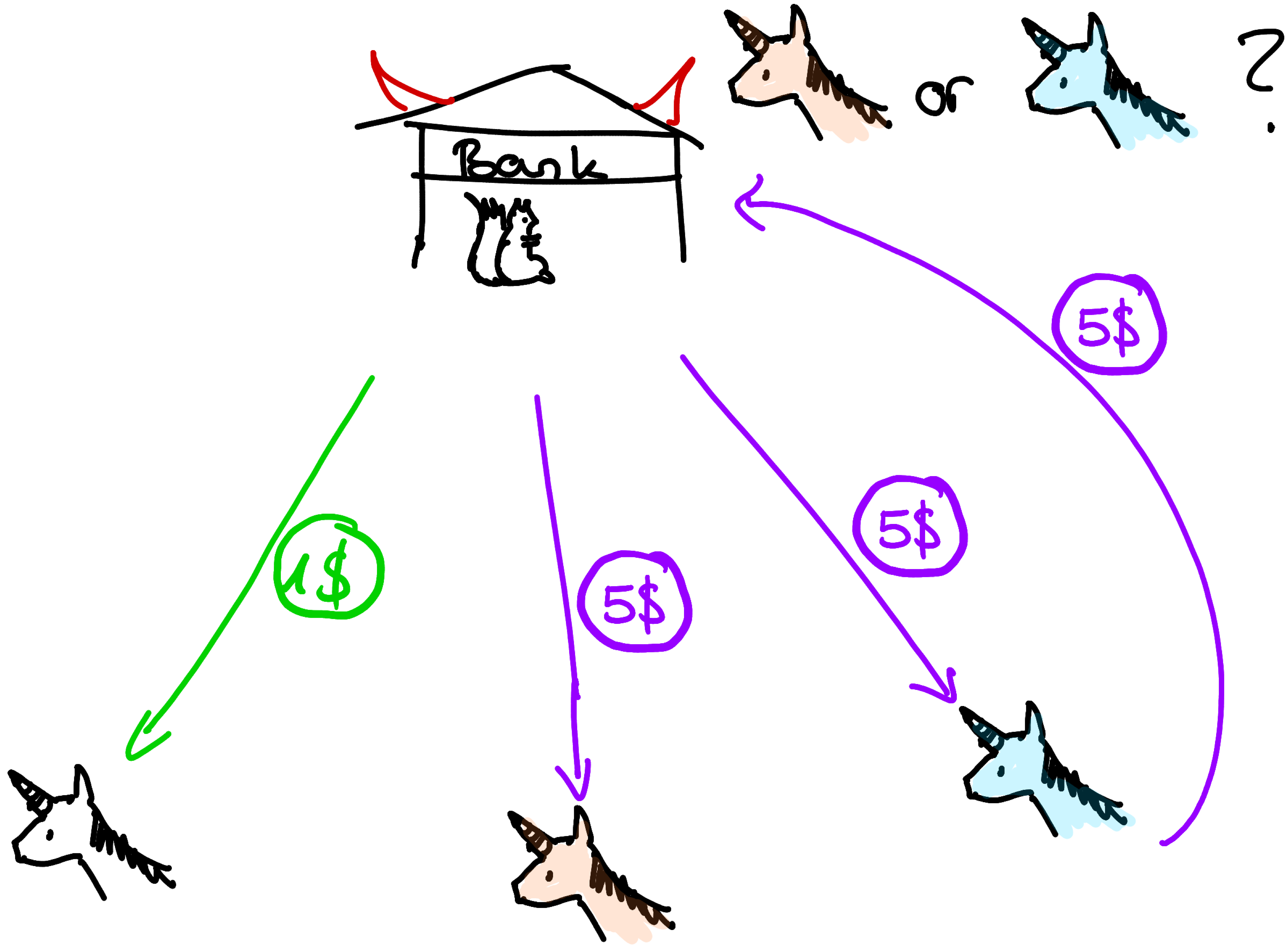
$\downarrow$

σ

Electronic Cash

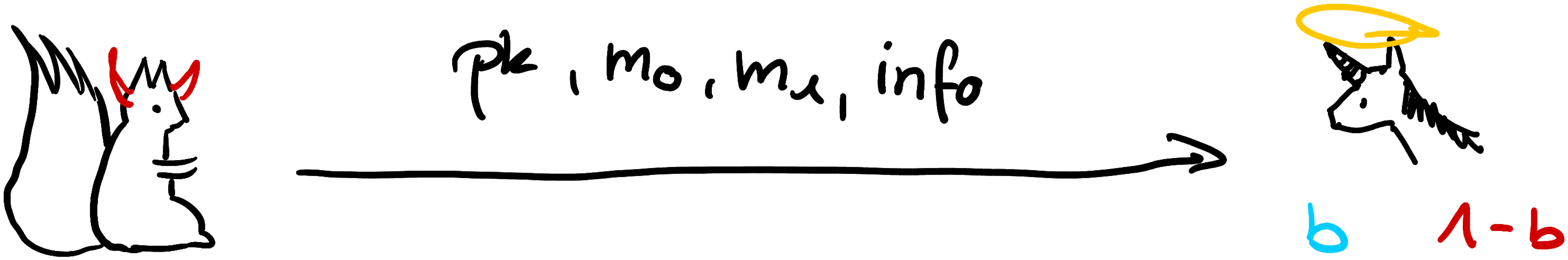


Electronic Cash

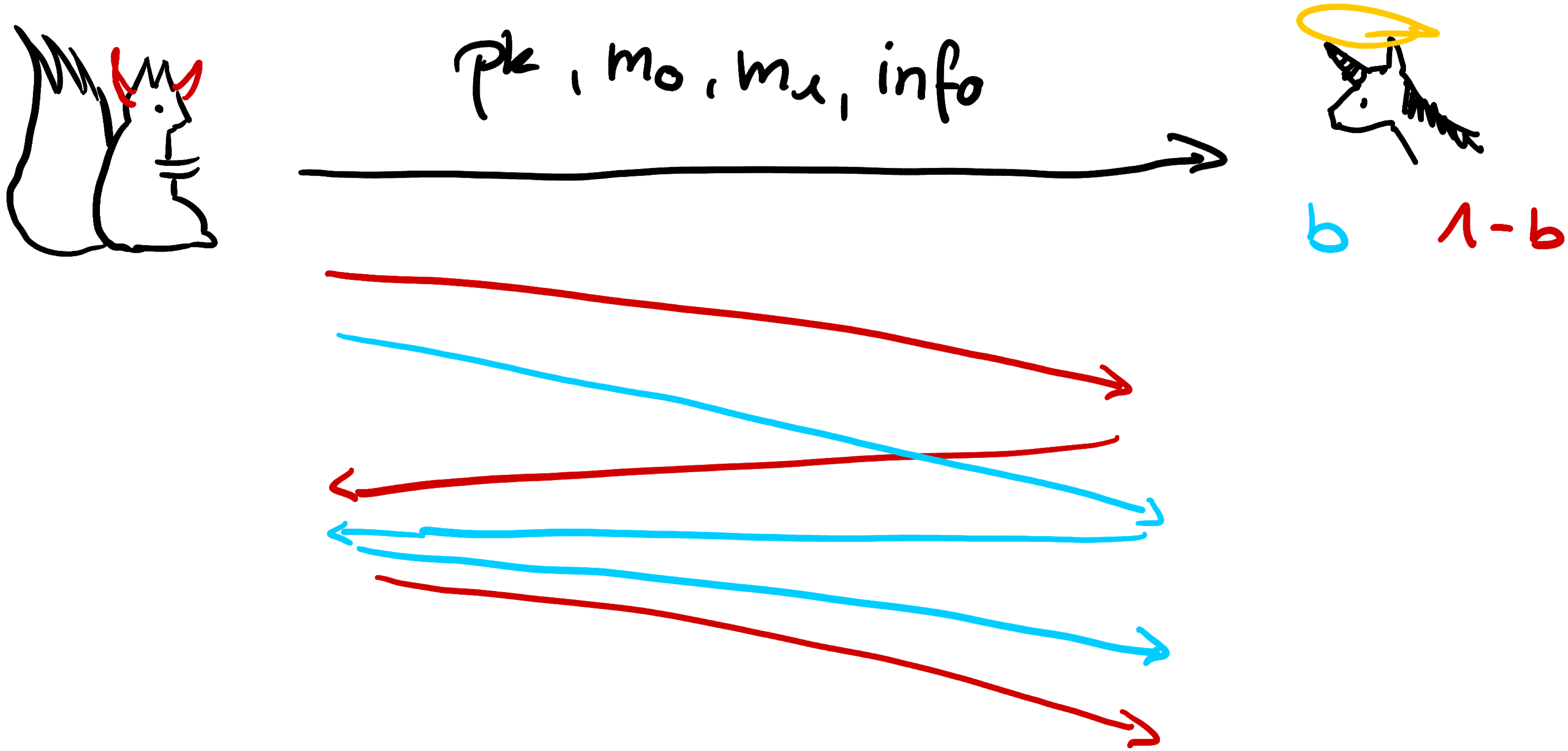


Partial Blindness

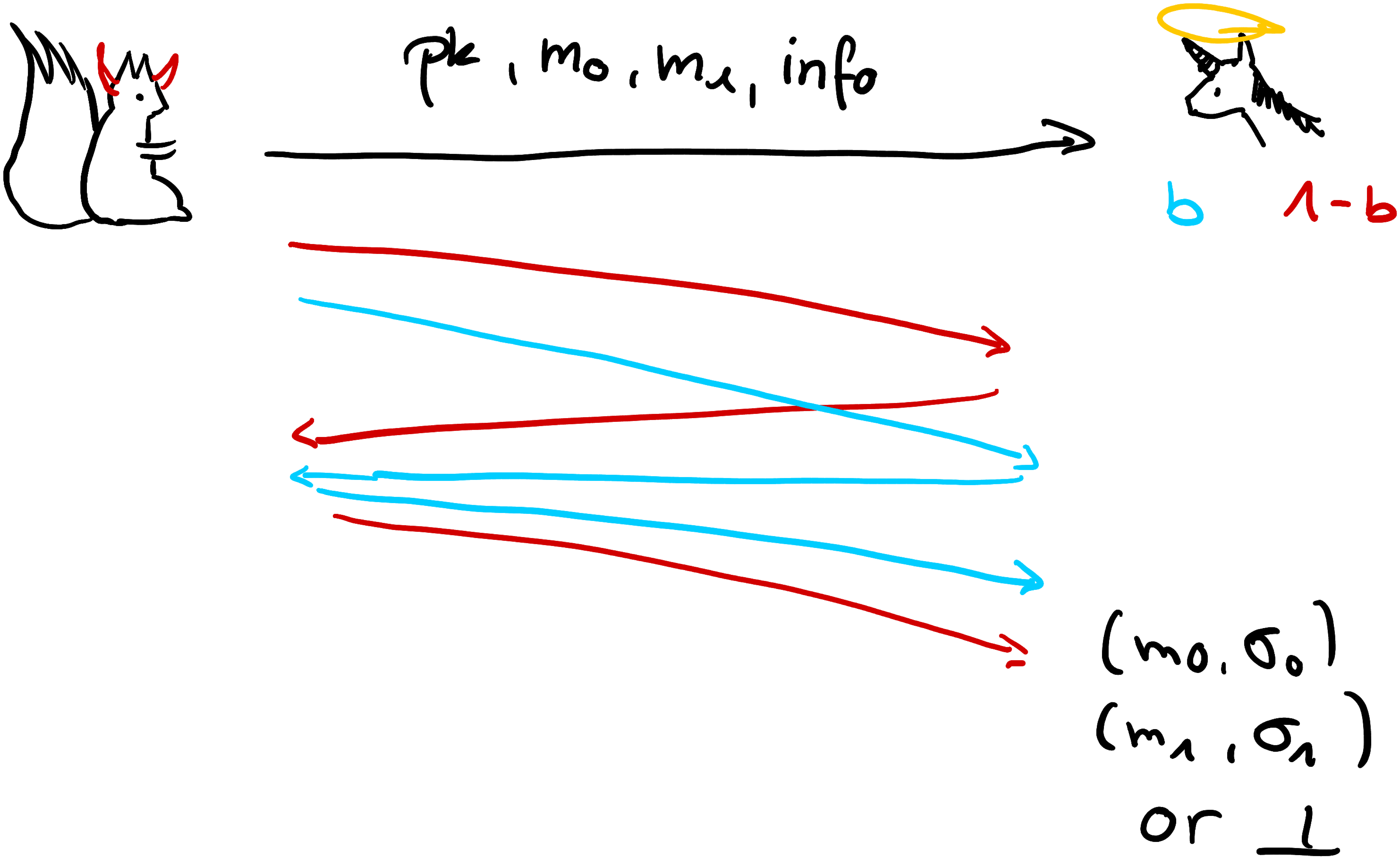
Partial Blindness



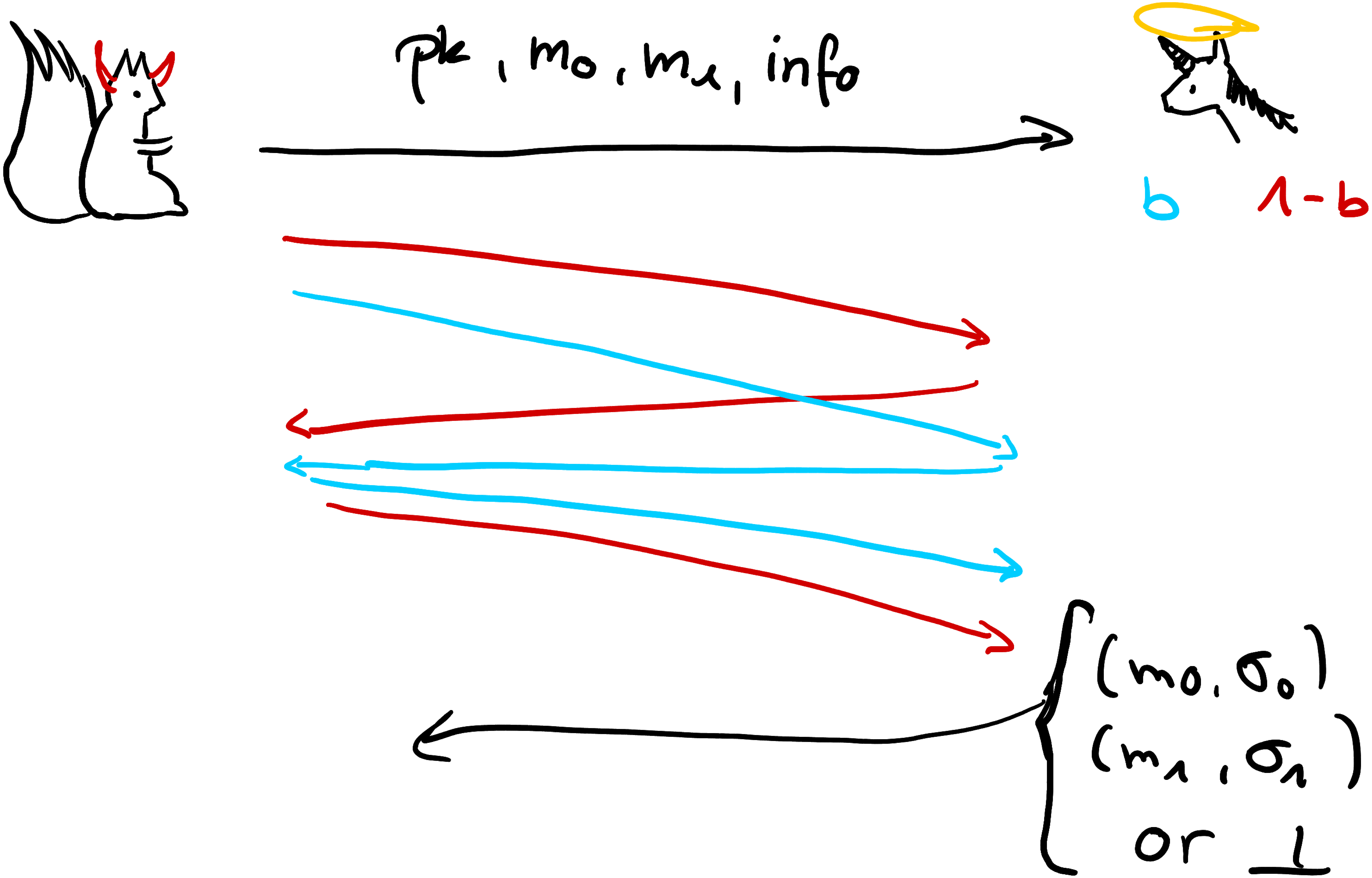
Partial Blindness



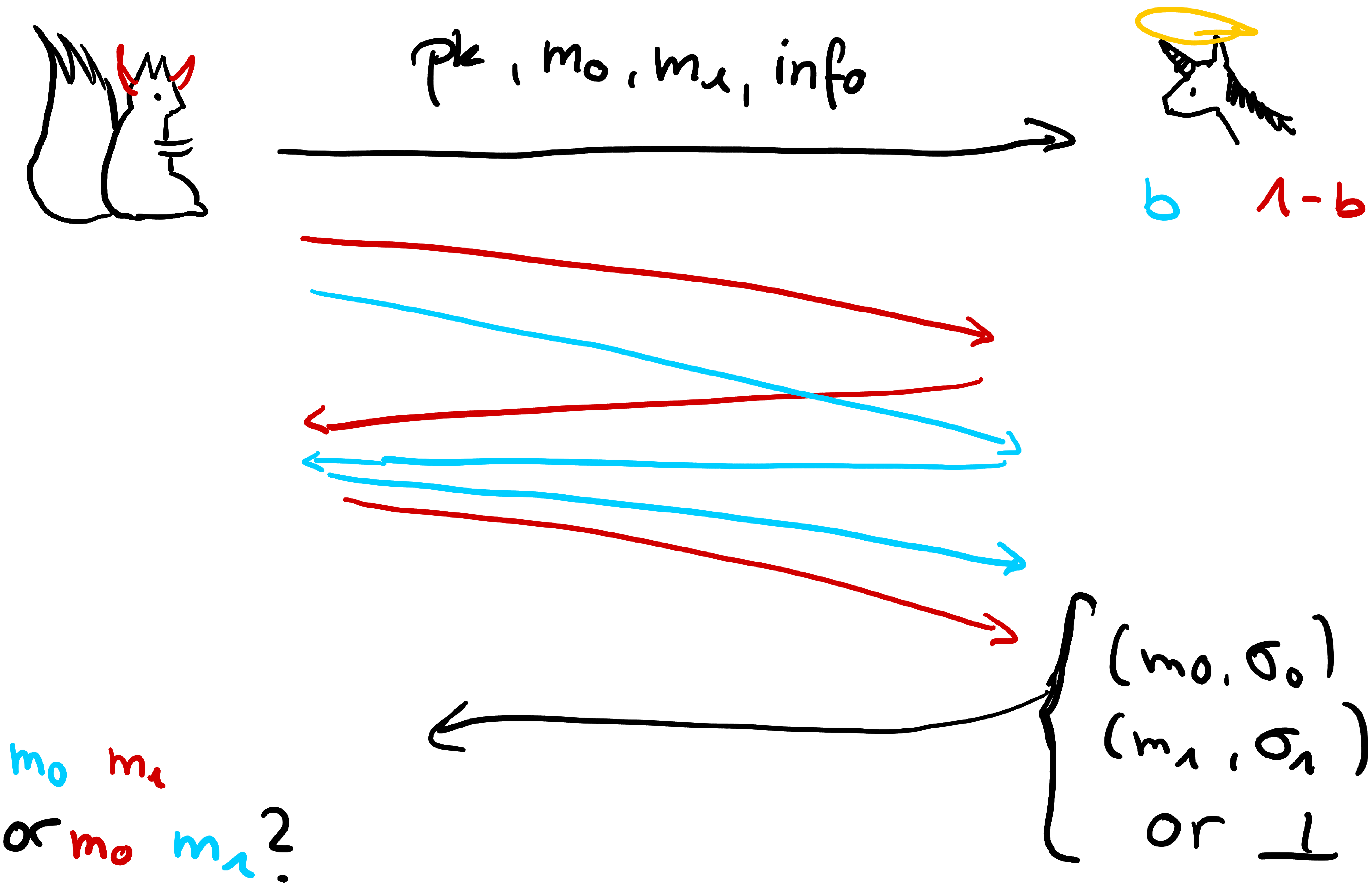
Partial Blindness



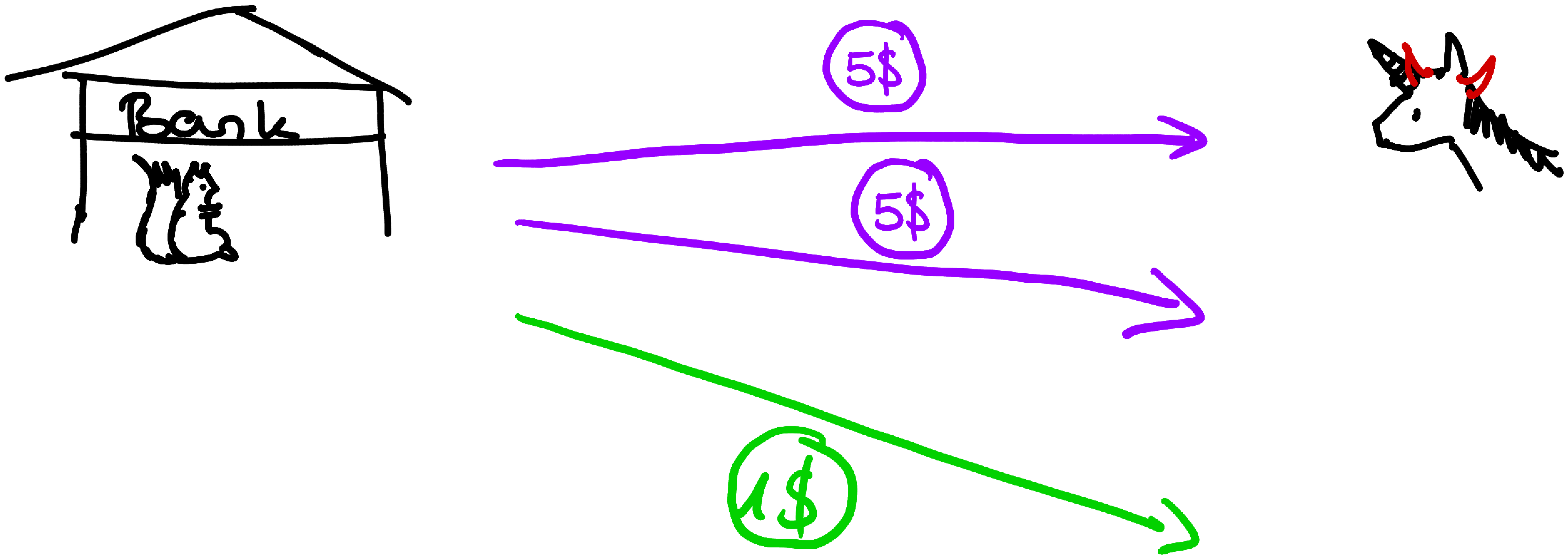
Partial Blindness



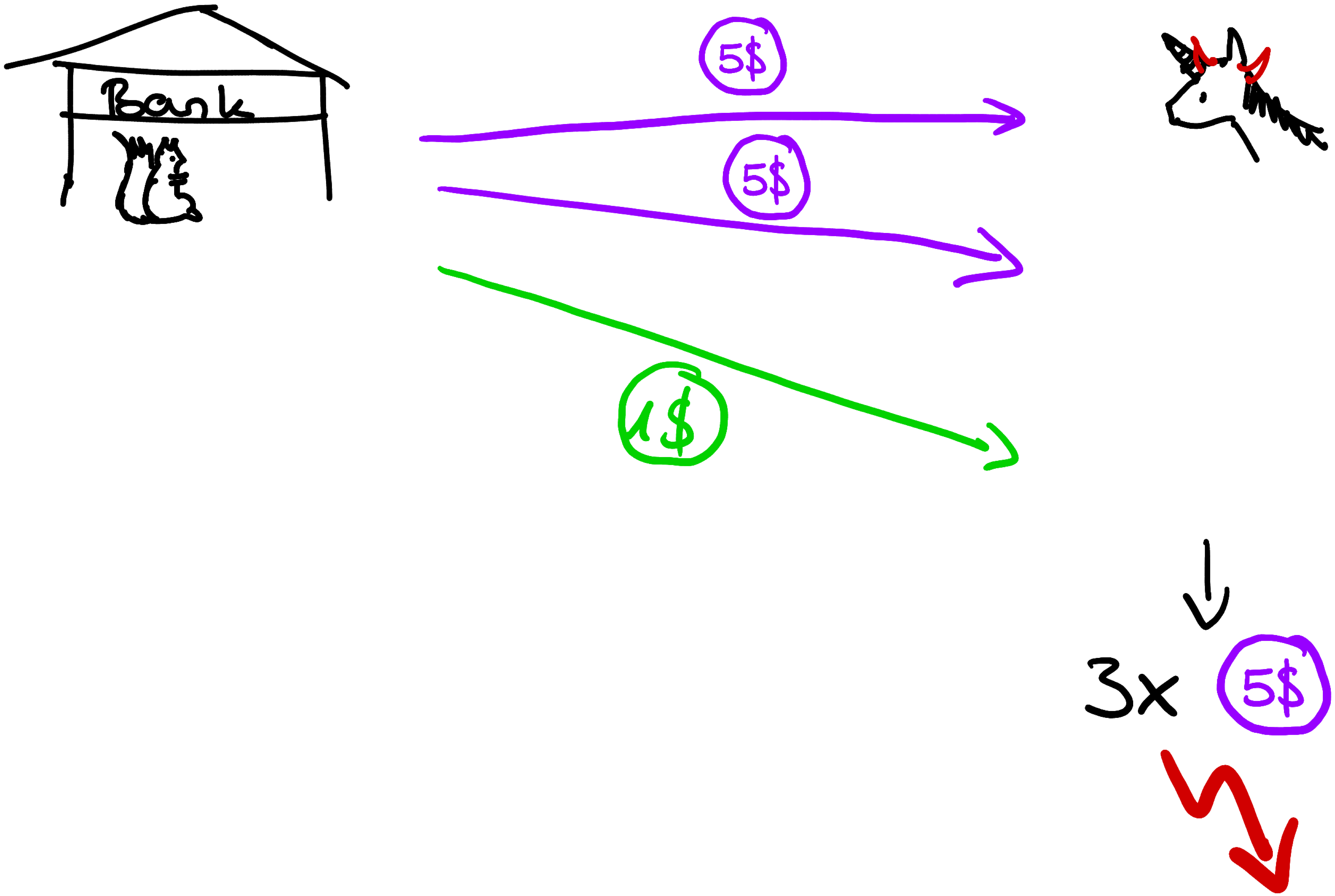
Partial Blindness



Electronic Cash

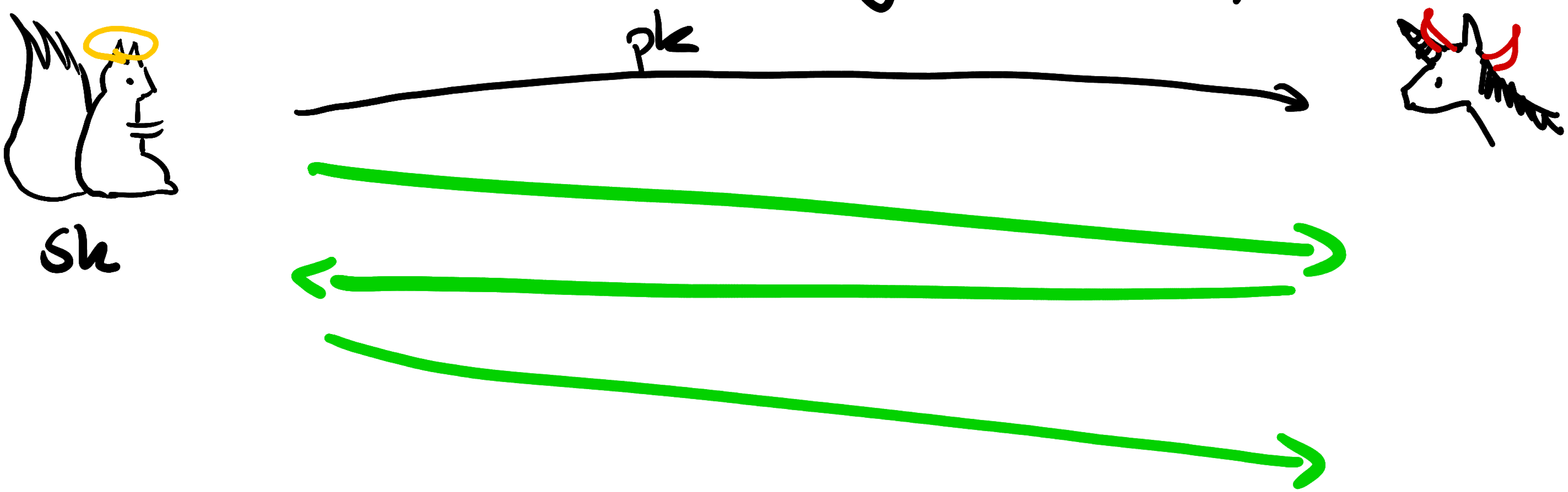


Electronic Cash

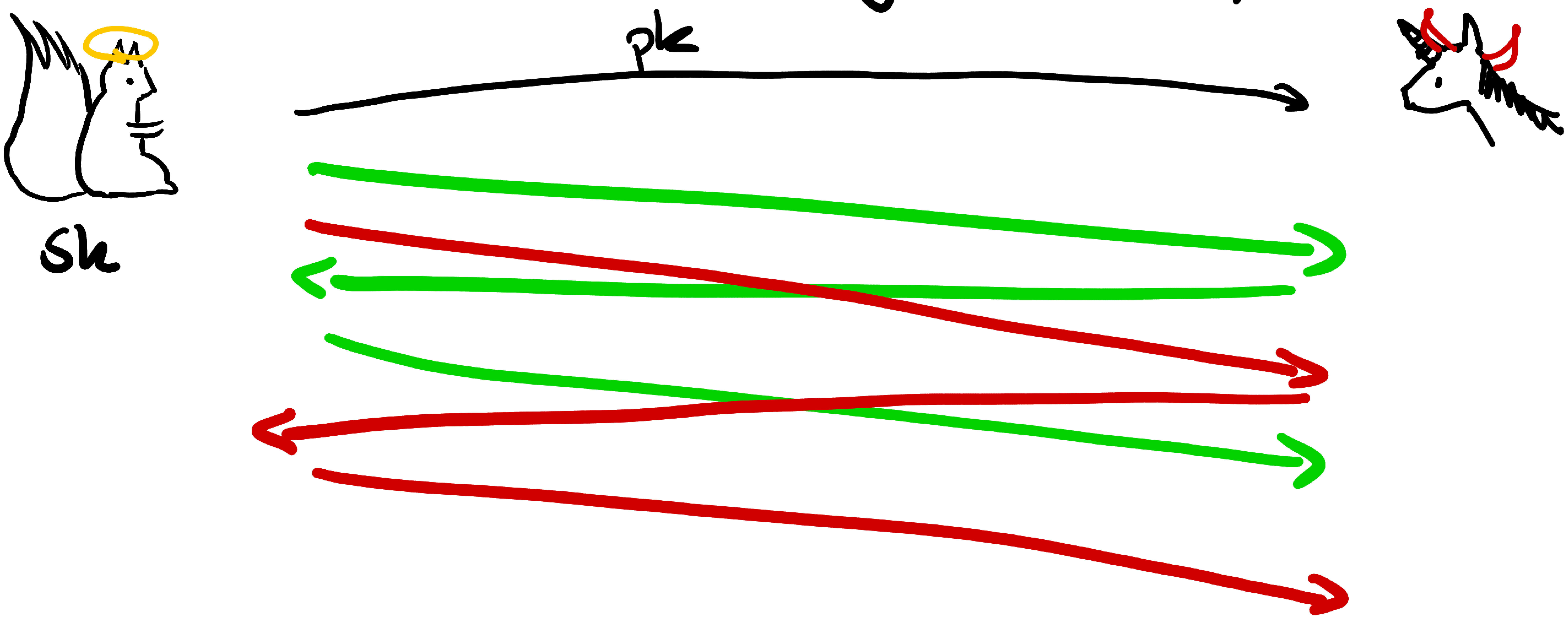


One-more Unforgeability

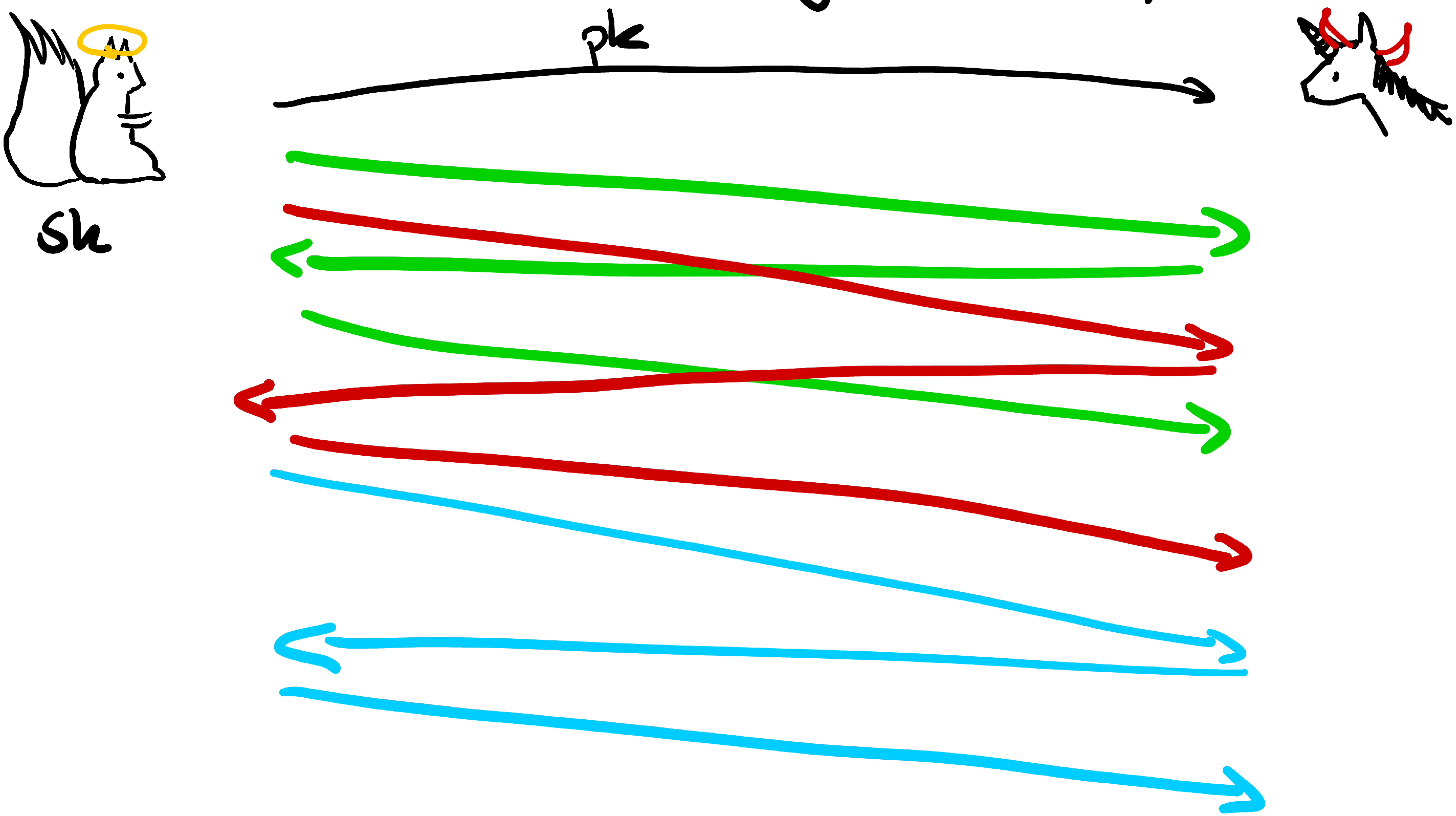
One-more Unforgeability



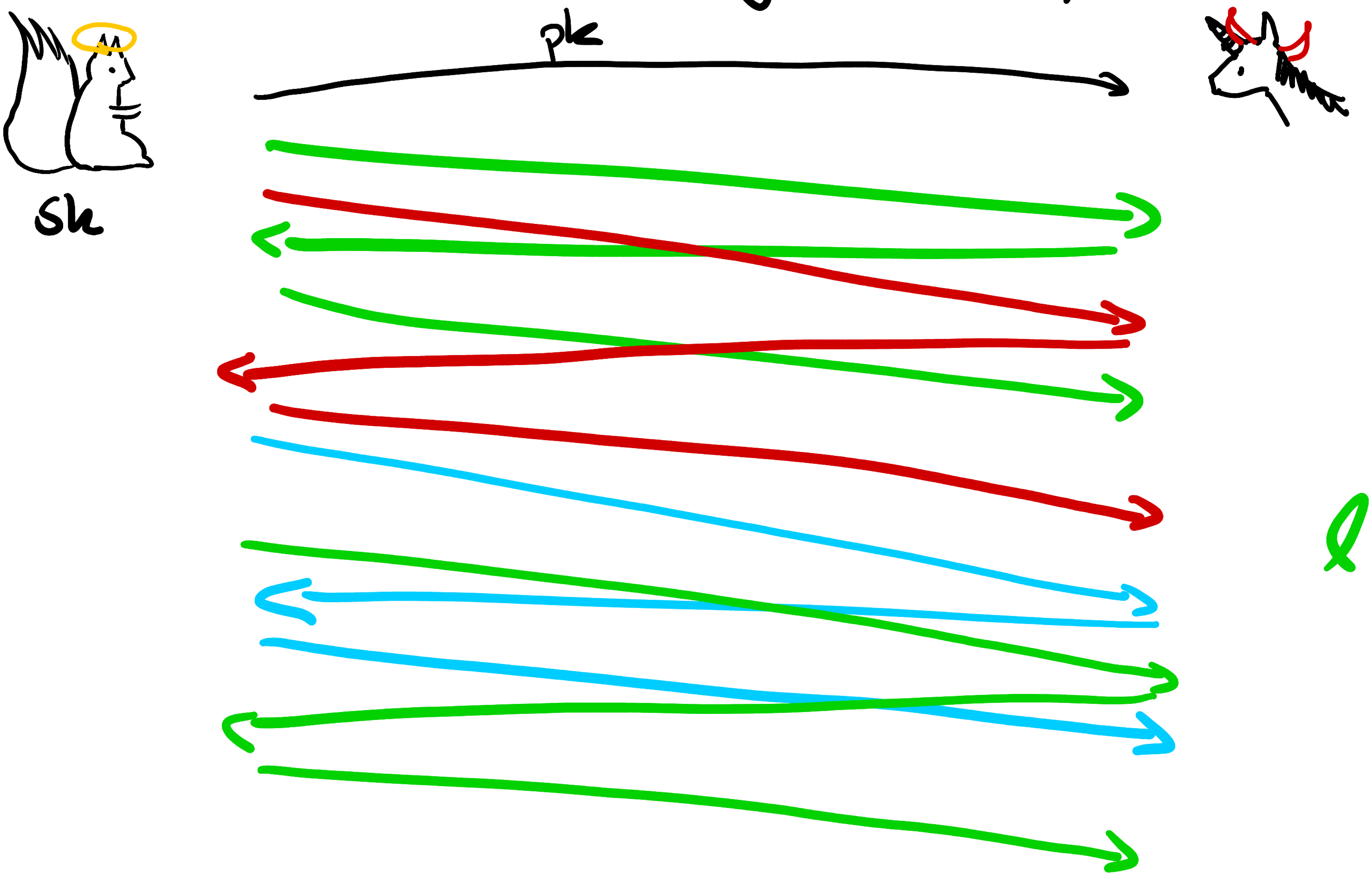
One-more Unforgeability



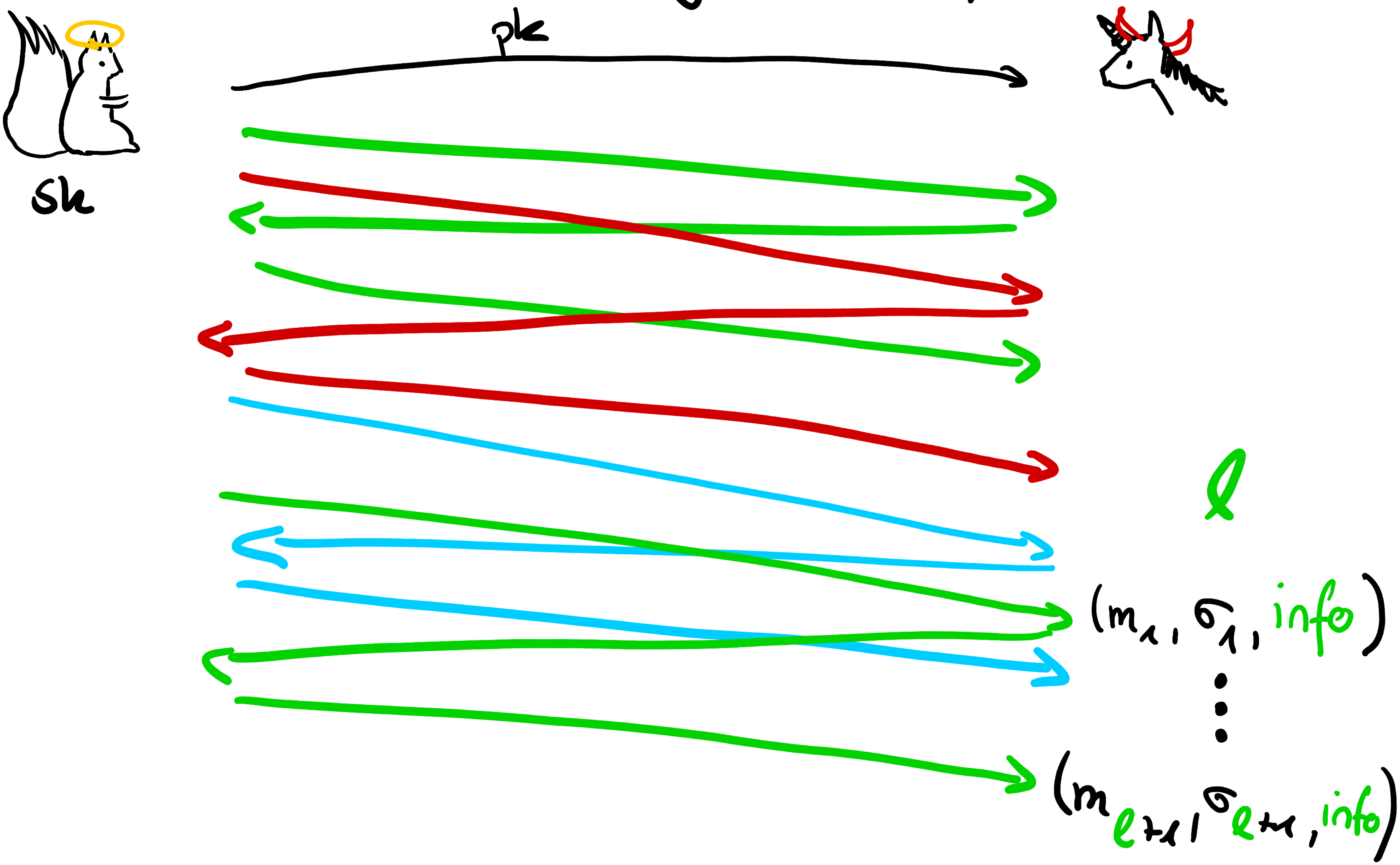
One-more Unforgeability



One-more Unforgeability



One-more Unforgeability



Motivation

- [AO'00] efficient DL-based PBS
- inspiration for several other schemes
  e.g. [Abe01], [BL13], [AHJ21]
- Proof strategy of interest for other similar schemes

Our Contribution

- identify gap ⚠ in original   OMUF proof
- mend gap ⚠
- achieve similar bounds to original work

The Abe-Okamoto Scheme [AO 00]

$sk = x$

$z = H^*(info)$

e.g. 1\$

$pk = g^x = y$

$z = H^*(info)$

The Abe-Okamoto Scheme [AO 00]

$sk = x$

$z = H^*(info)$

$pk = g^x = y$

$z = H^*(info)$

a, b $\xrightarrow{a,\ b}$

The Abe-Okamoto Scheme [AO 00]

$sk = x$

$z = H^*(info)$

$pk = g^x = y$

$z = H^*(info)$

a, b

$\xrightarrow{a,\ b}$

$a \rightsquigarrow \alpha$

$b \rightsquigarrow \beta$

$e \leftarrow H(\alpha, \beta, m, info)$

$\xleftarrow{e}$

The Abe-Okamoto Scheme [AO 00]

$sk = x$

$z = H^*(\mathrm{info})$

$pk = g^x = y$

$z = H^*(\mathrm{info})$

a, b $\xrightarrow{a,\ b}$

$a \rightsquigarrow \alpha$

$b \rightsquigarrow \beta$

$e \leftarrow H(\alpha, \beta, m, \mathrm{info})$

$\xleftarrow{e}$

$\xrightarrow{c,\ r,\ d,\ s}$

$c, r \rightsquigarrow \omega, \rho$

$d, s \rightsquigarrow \delta, \sigma$

The Abe-Okamoto Scheme [AO 00]

$sk = x$

$z = H^*(\mathrm{info})$

$pk = g^x = y$

$z = H^*(\mathrm{info})$

a, b $\xrightarrow{\quad a,\ b \quad}$

$a \rightsquigarrow \alpha$

$b \rightsquigarrow \beta$

$e \leftarrow H(\alpha, \beta, m, \mathrm{info})$

$\xleftarrow{\quad e \quad}$

$\xrightarrow{\quad c,\ r,\ d,\ s \quad}$

$c, r \rightsquigarrow \omega, \rho$

$d, s \rightsquigarrow \delta, \sigma$

$sig = (\omega, \rho, \delta, \sigma)$

Verification

$$\omega + \delta \stackrel{?}{=} H(y^{\omega} \cdot g^{\rho}, z^{\delta} \cdot g^{\sigma}, m, \mathrm{info})$$

"OR-Proof of two Schnorr-Signatures"

Proving One-more Unforgeability

Idea:

Simulate using x, extract $\mathrm{dlog}\, z$

OR

Simulate using $\mathrm{dlog}\, z$, extract x

Reduction strategy

1. Pick "secret key" x or dlog z

Reduction strategy

1. Pick "secret key" x or dlog z
2. Run adversary once

Reduction strategy

1. Pick "secret key" x or $\mathrm{dlog}\ z$
2. Run adversary once
3. Re-program RO

Reduction strategy

1. Pick "secret key" x or $d\log z$
2. Run adversary once
3. Re-program RO
4. Run adversary another time

Reduction strategy

1. Pick "secret key" x or $\mathrm{dlog}\ z$

2. Run adversary once

3. Re-program RO

4. Run adversary another time

5. Hope to get "other key"

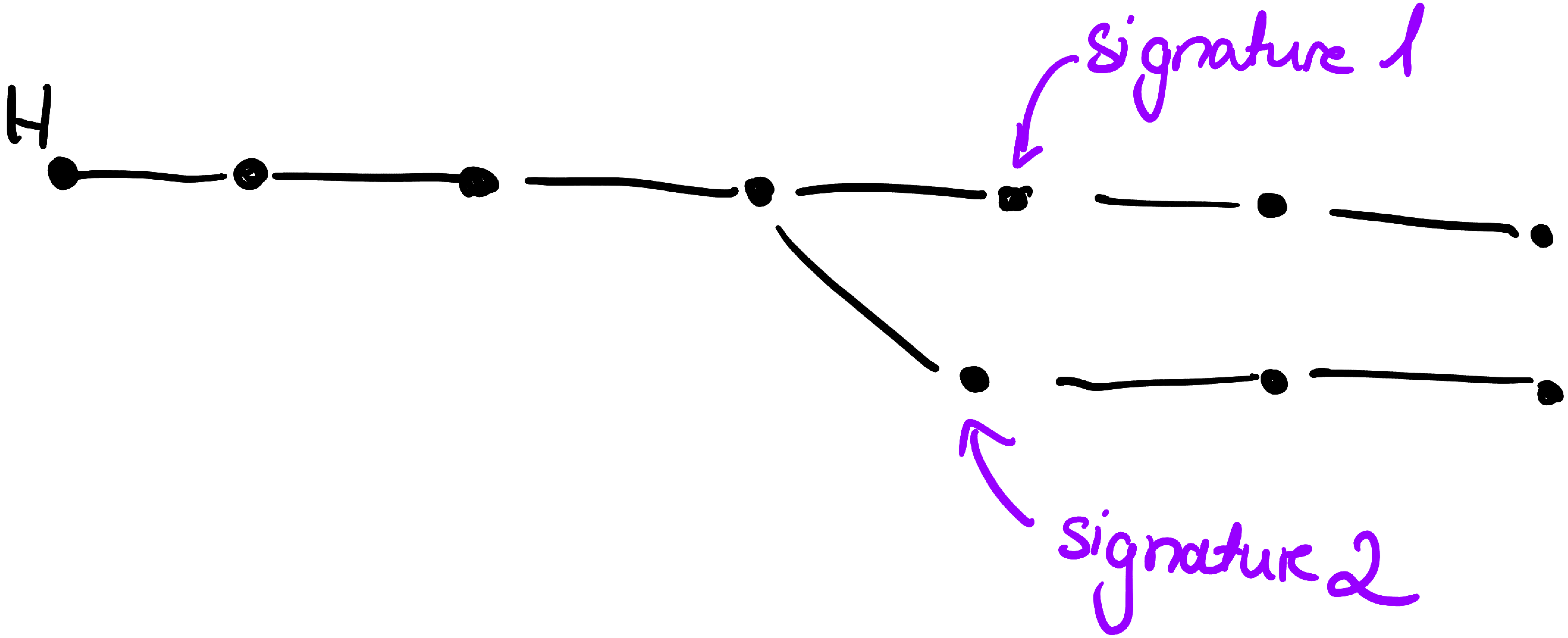
Reduction strategy

1. Pick "secret key" x or $\mathrm{dlog}\ z$
2. Run adversary once
3. Re-program RO
4. Run adversary another time
5. Hope to get "other key"
6. Return "other key" as solution to DL

Reduction strategy

1. Pick "secret key" x or dlog z
2. Run adversary once
3. Re-program RO
4. Run adversary another time
5. Hope to get "other key"
6. Return "other key" as solution to DL

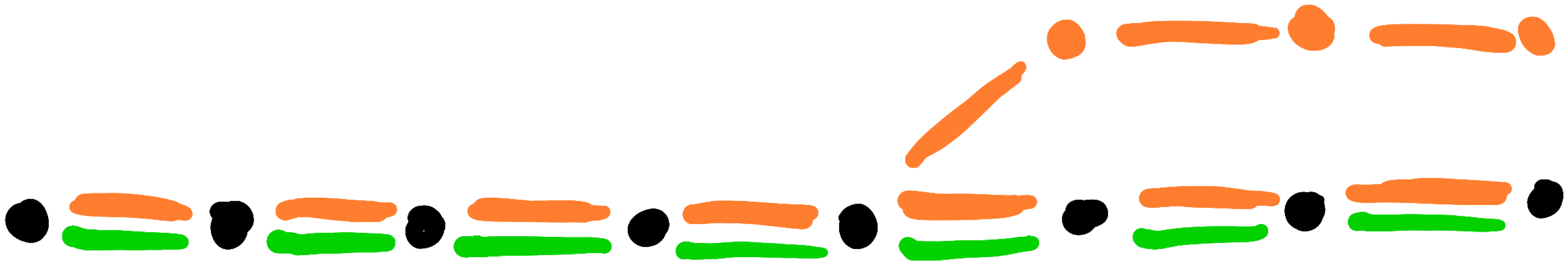
Forking [PS96]



$\mathcal{E}$(signature 1, signature 2) $\rightsquigarrow$ X

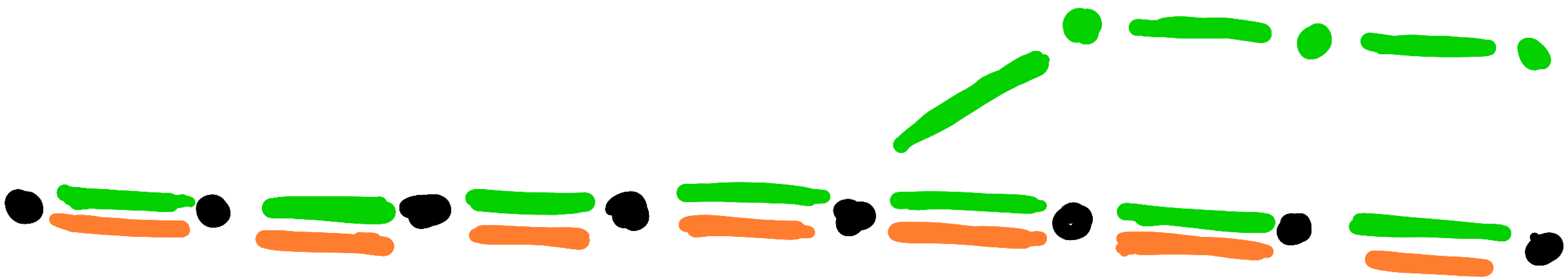
$\rightsquigarrow$ dlog z

Forking with an OR Proof



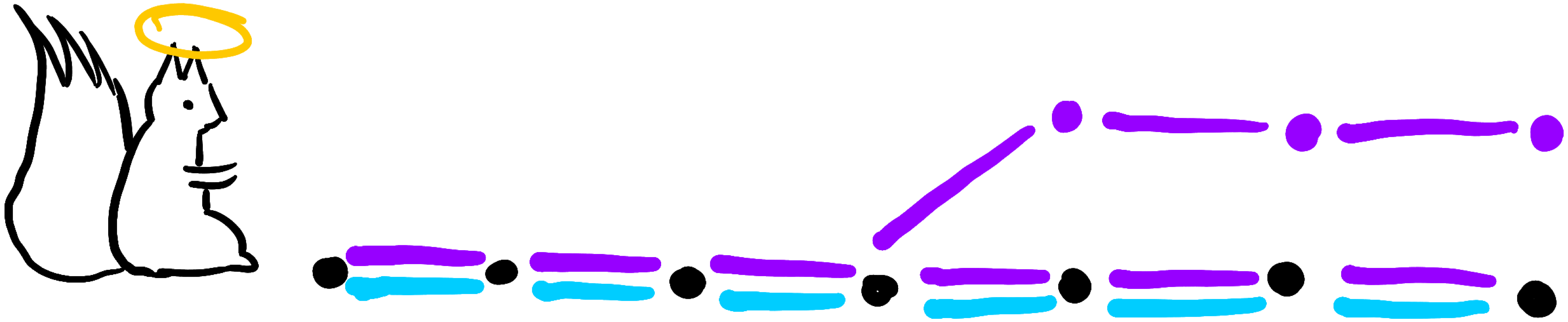
get x

OR

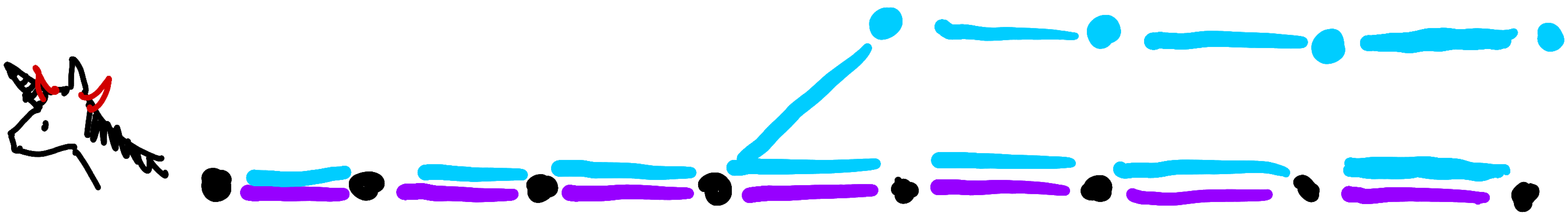


get dlog z

Reduction Goal



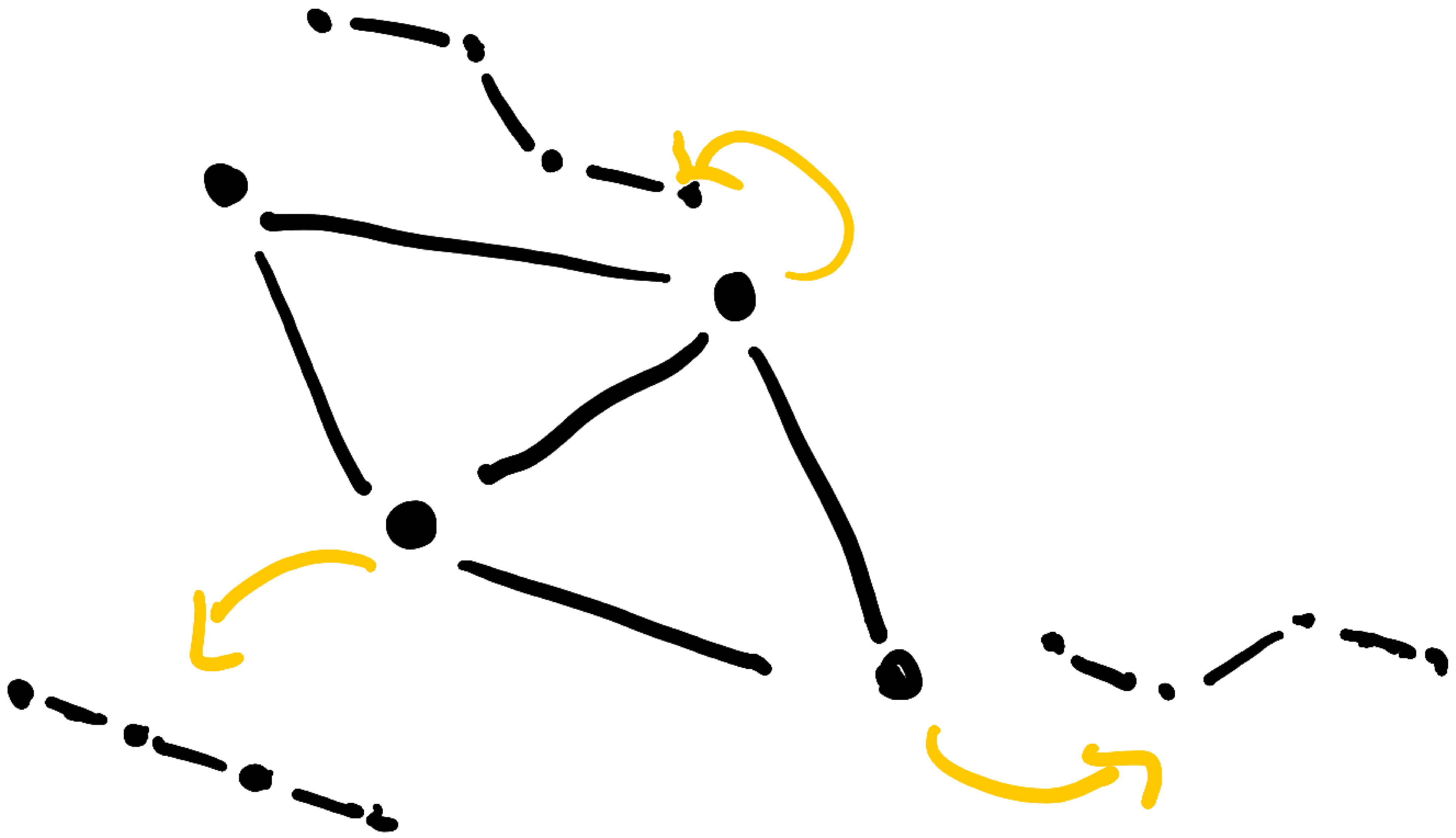
has "secret key"

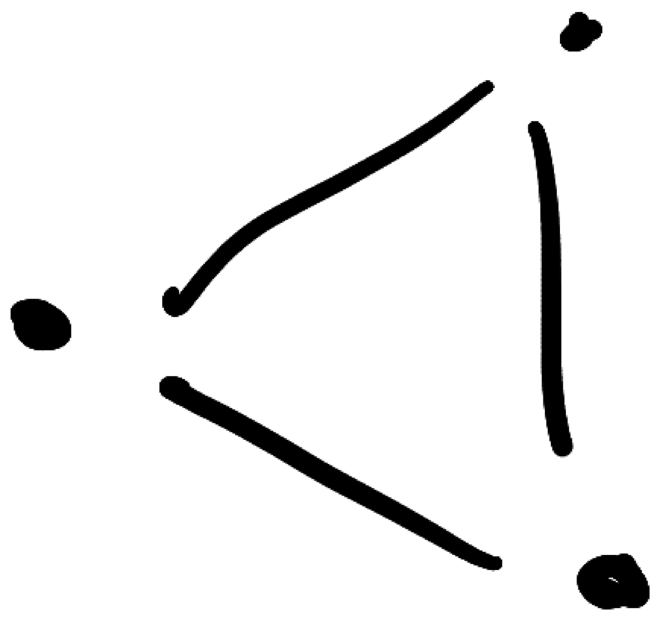


reveals "other key"

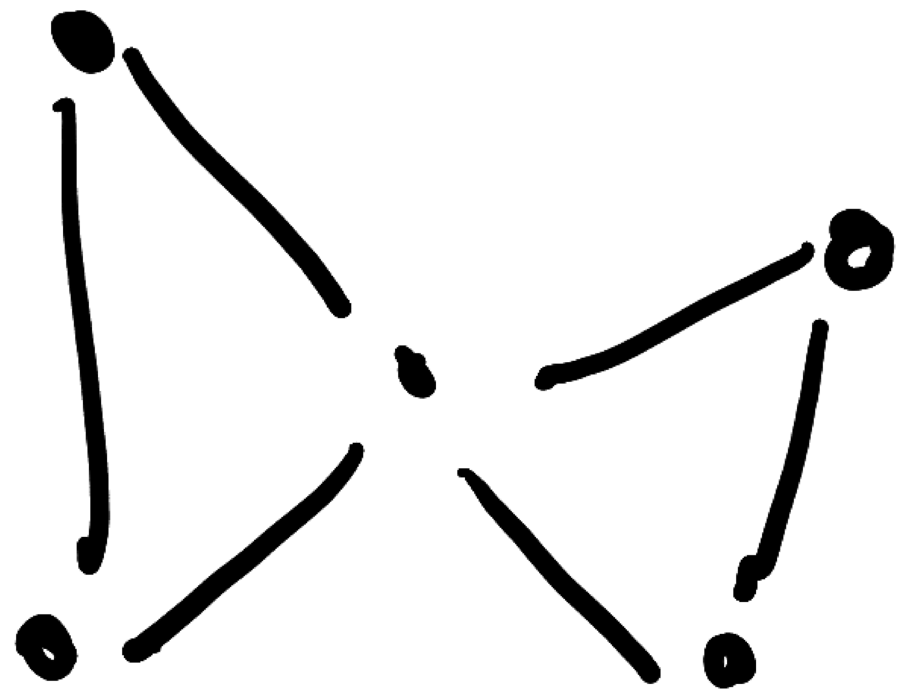
Abstraction in progress...

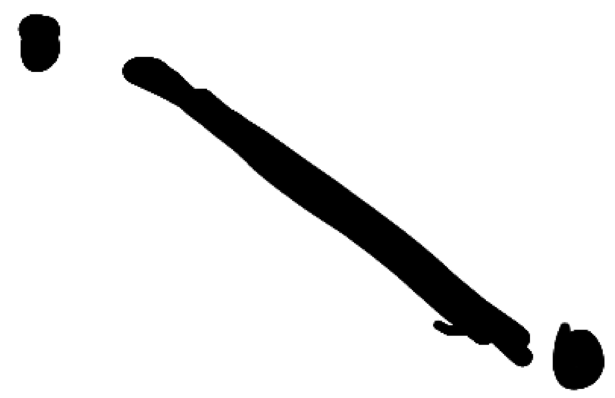
✗



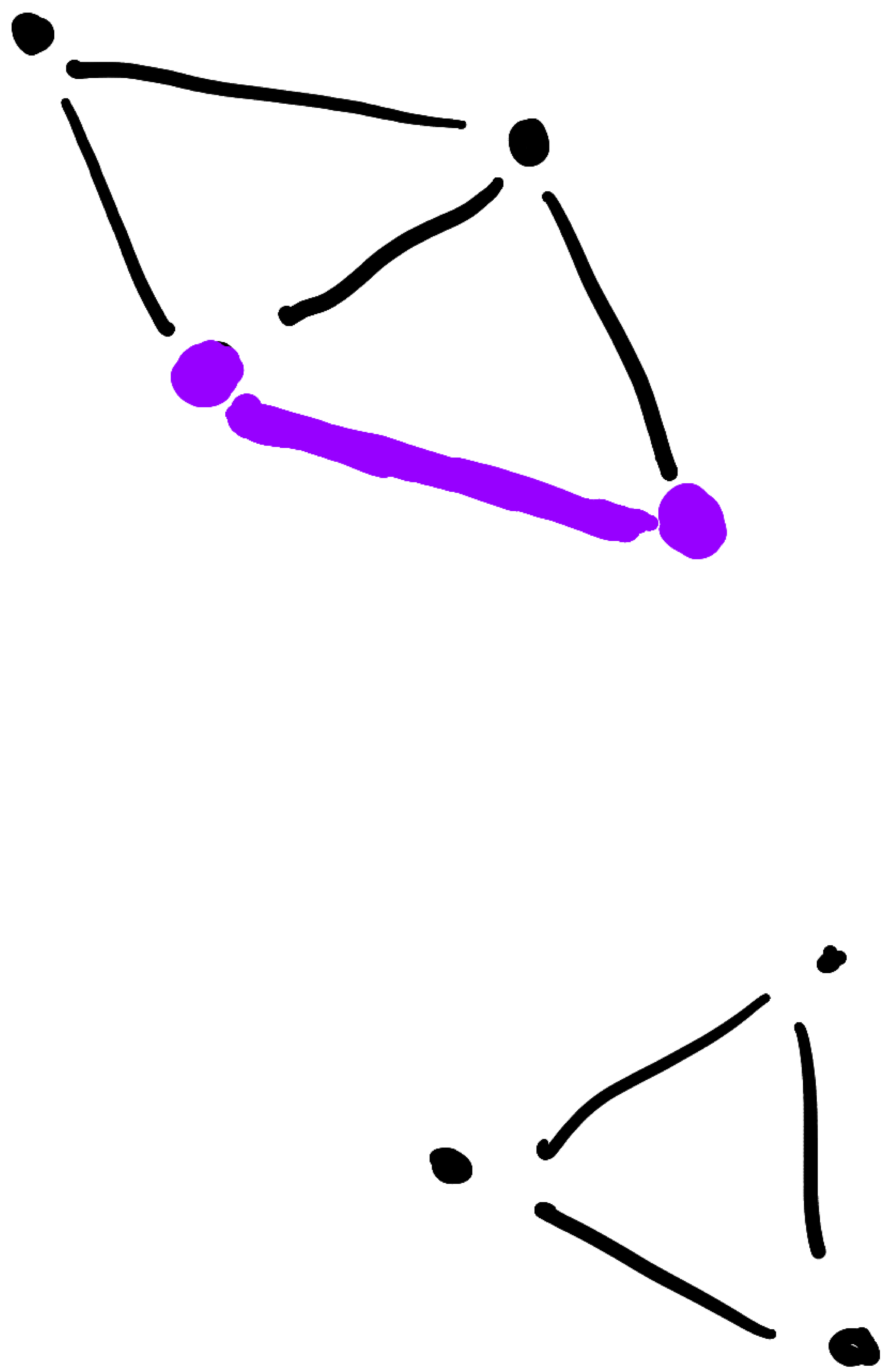


$\mathrm{dlog}\, z$

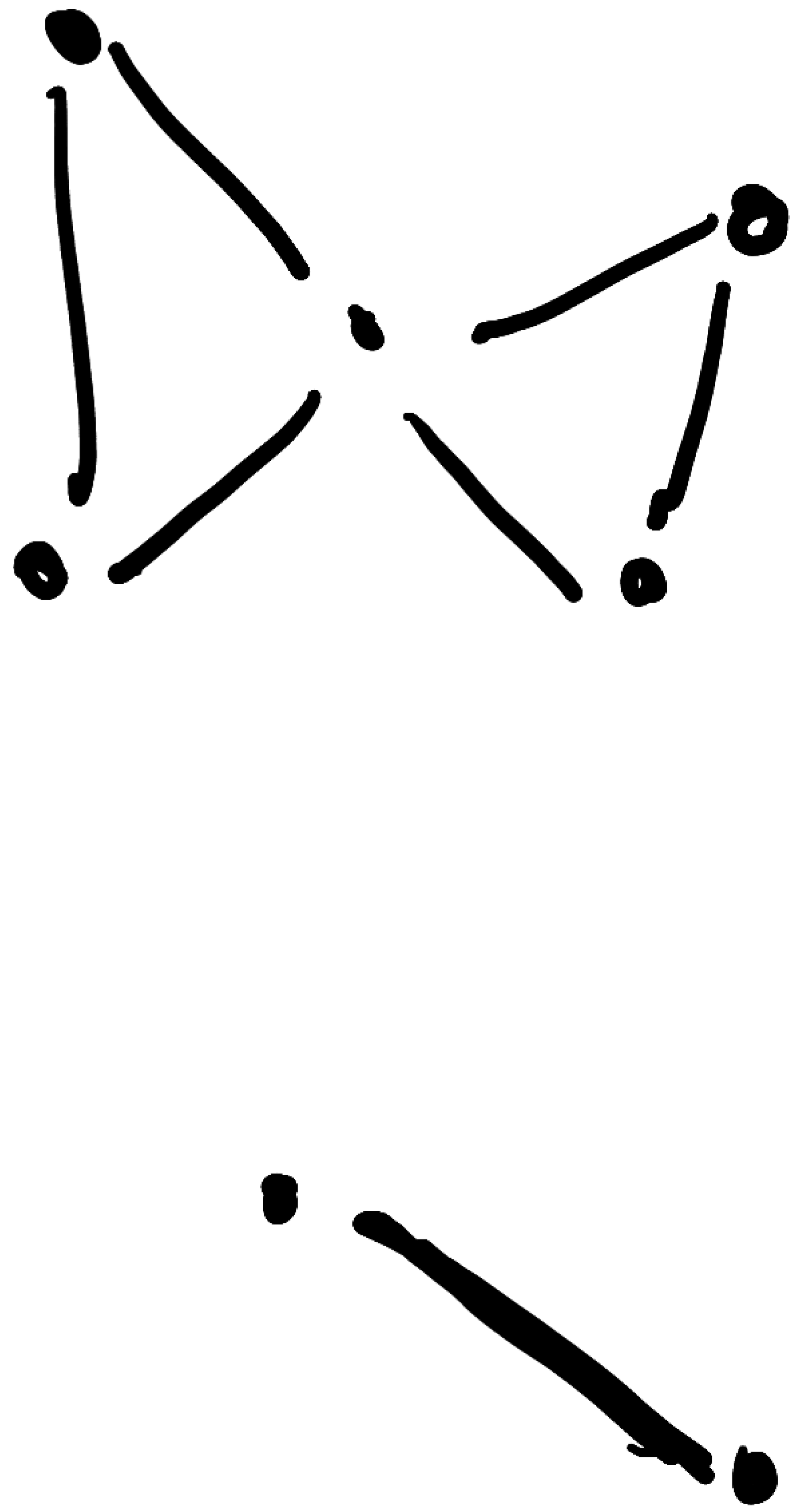




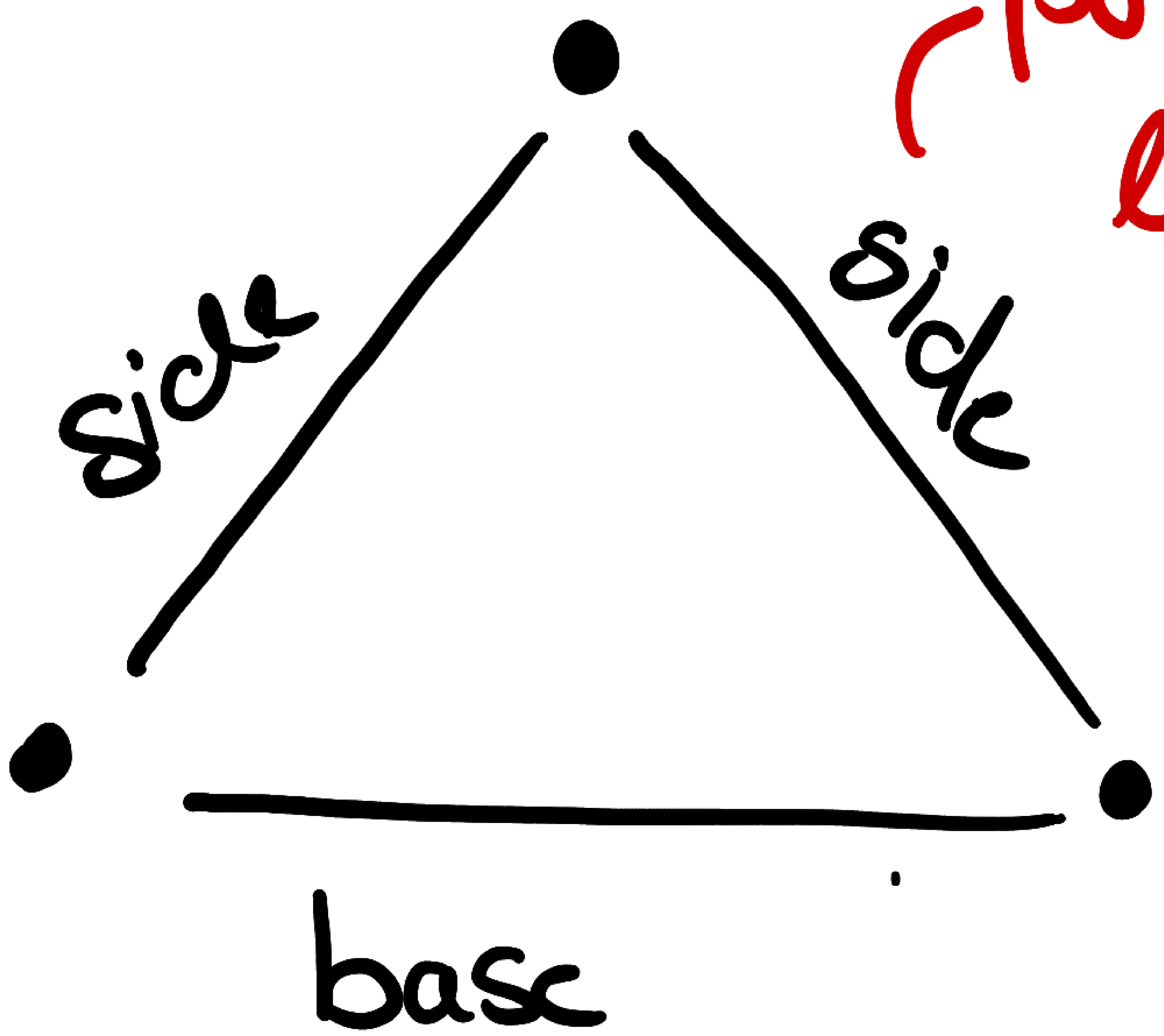
X



dlog z



Triangles [10'00]



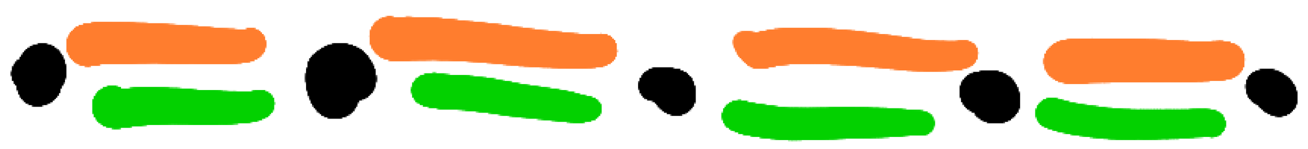
potentially less nice
likely to sample

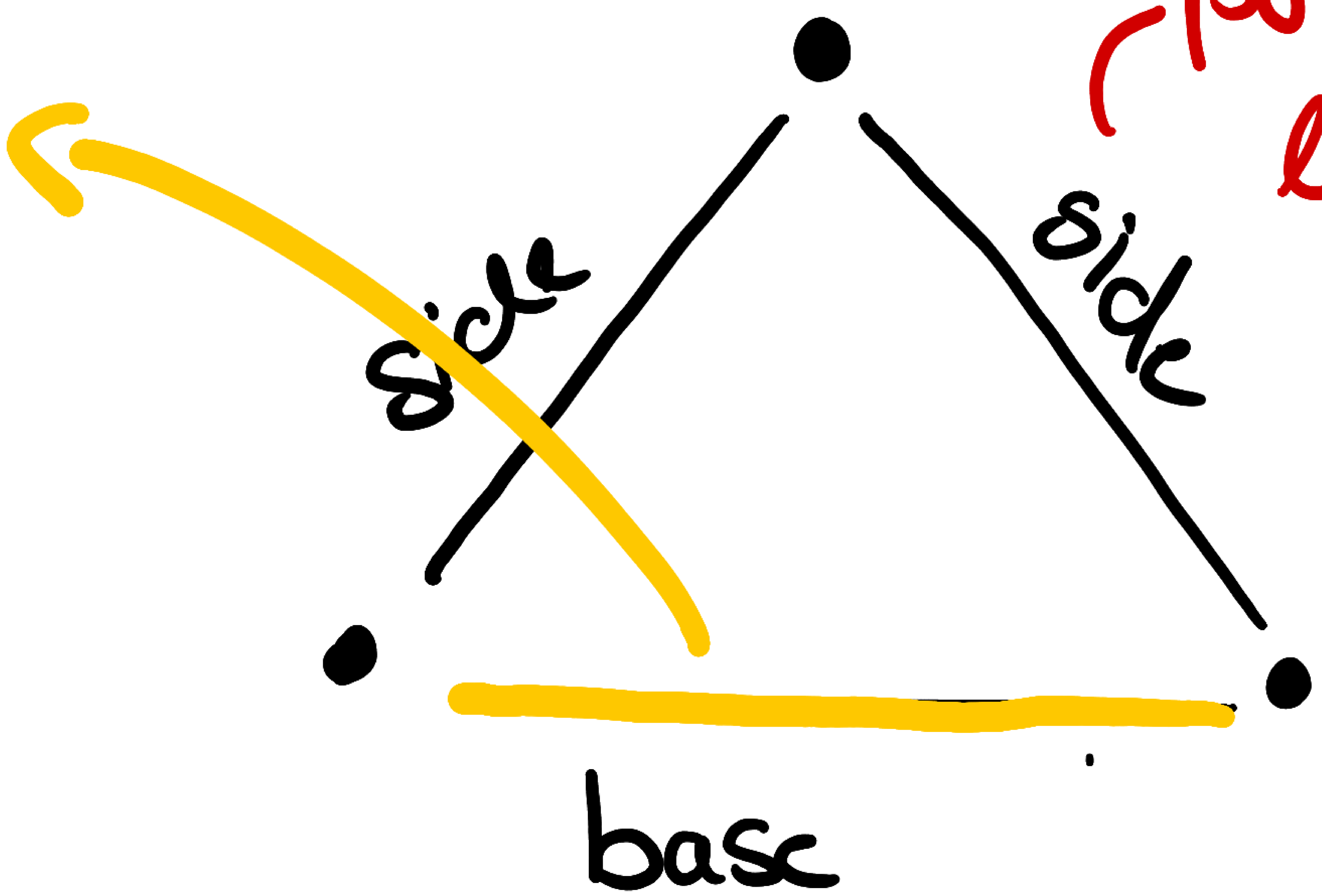
nice properties
unlikely to sample

Triangles [10'00]

can use either

x or dlog z

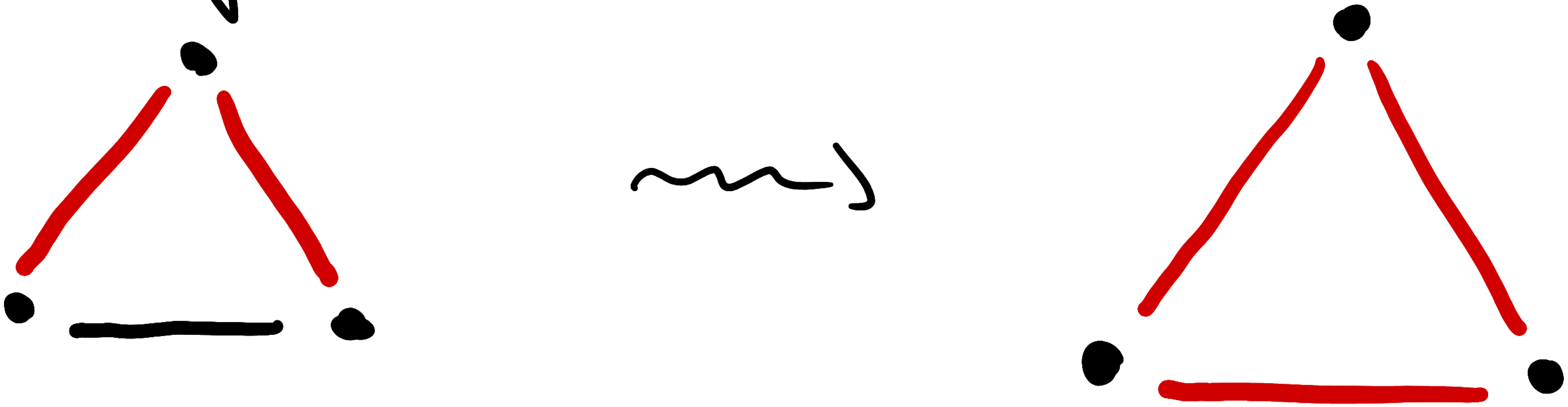




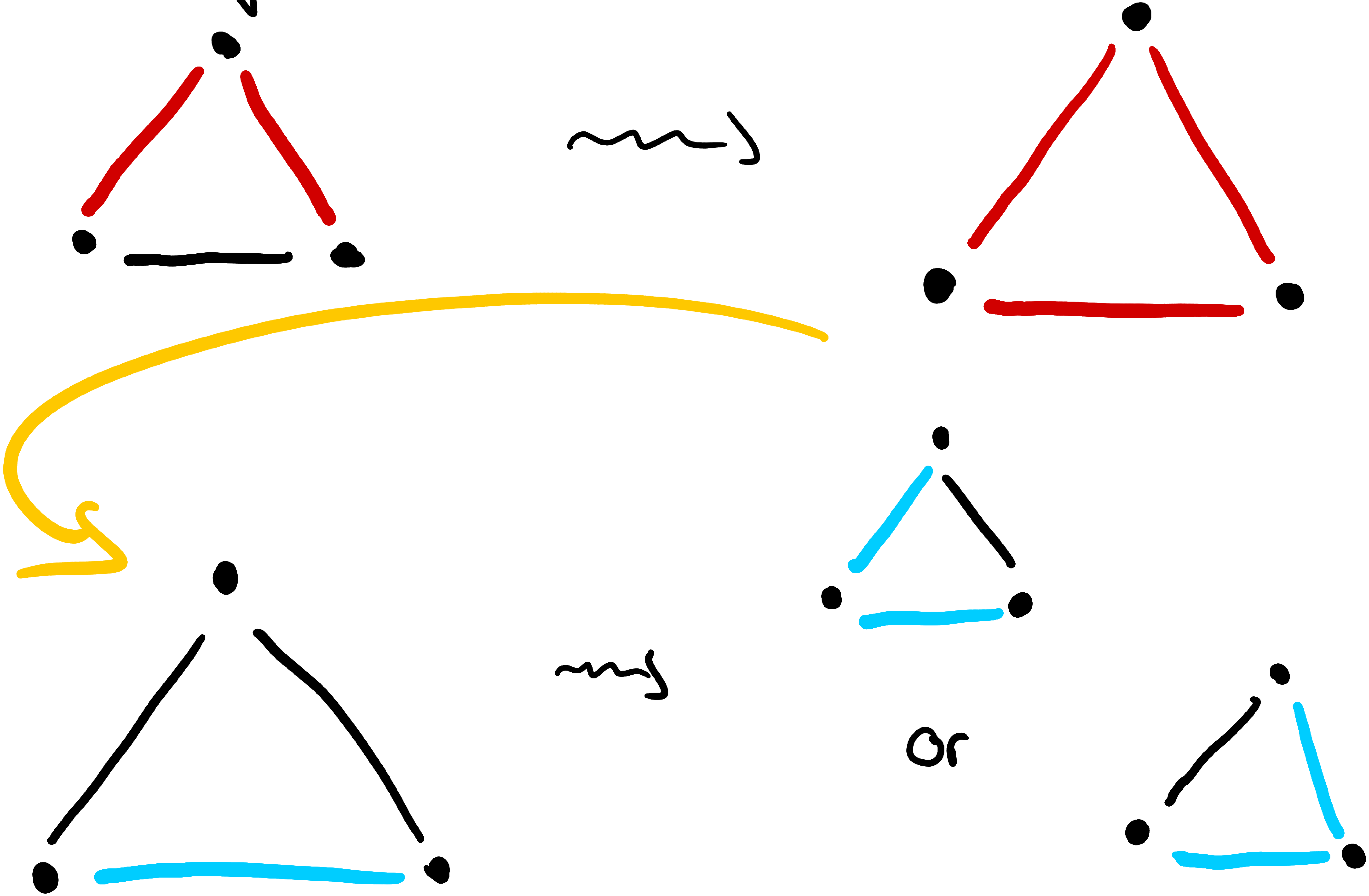
potentially less nice
likely to sample

nice properties
unlikely to sample

Triangle Properties

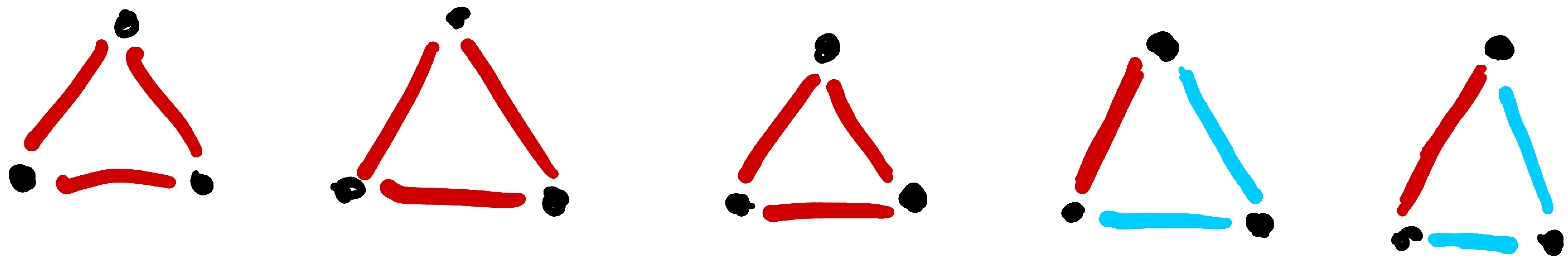


Triangle Properties

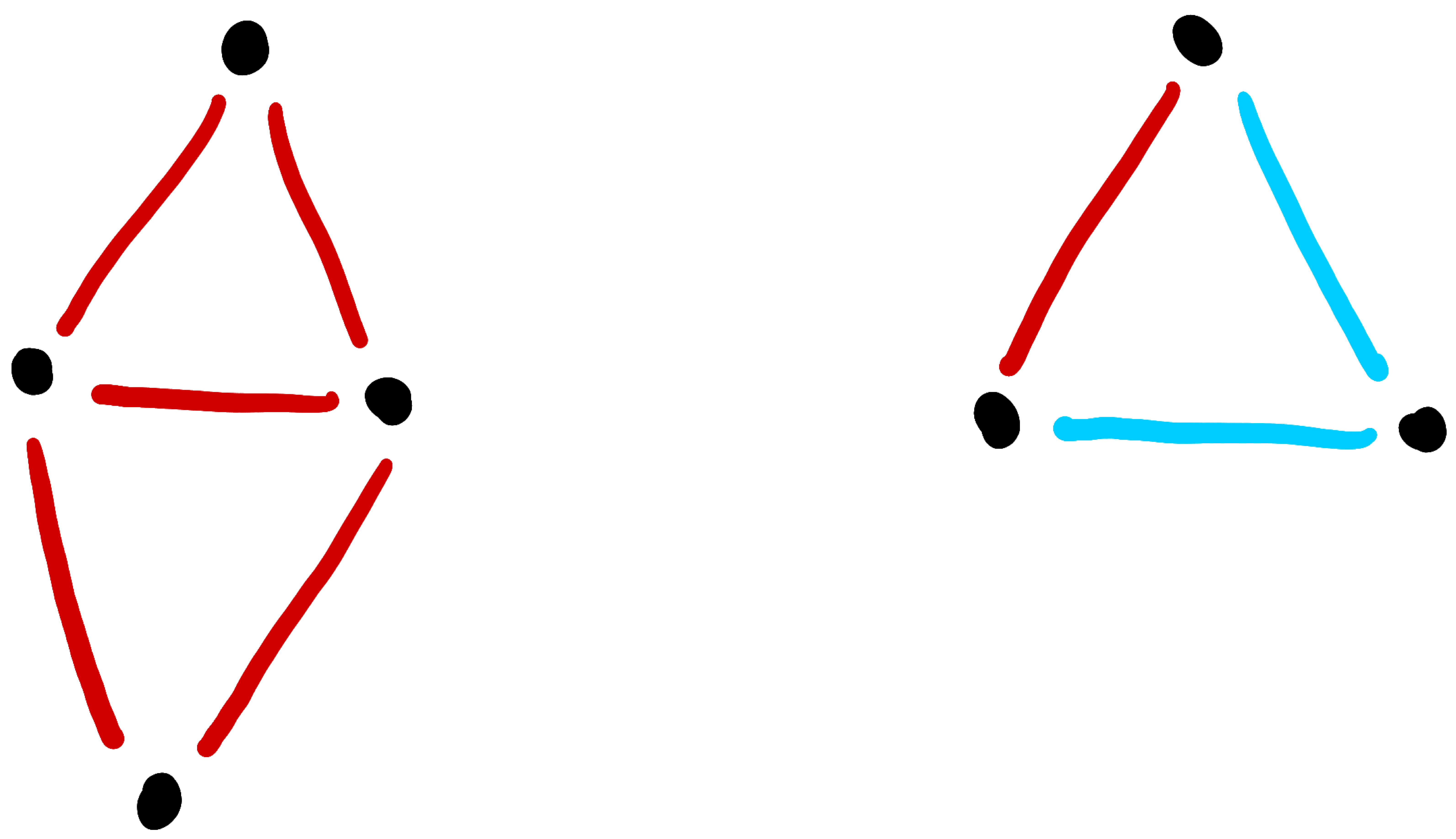


The Gap in [AO'00] ⚠

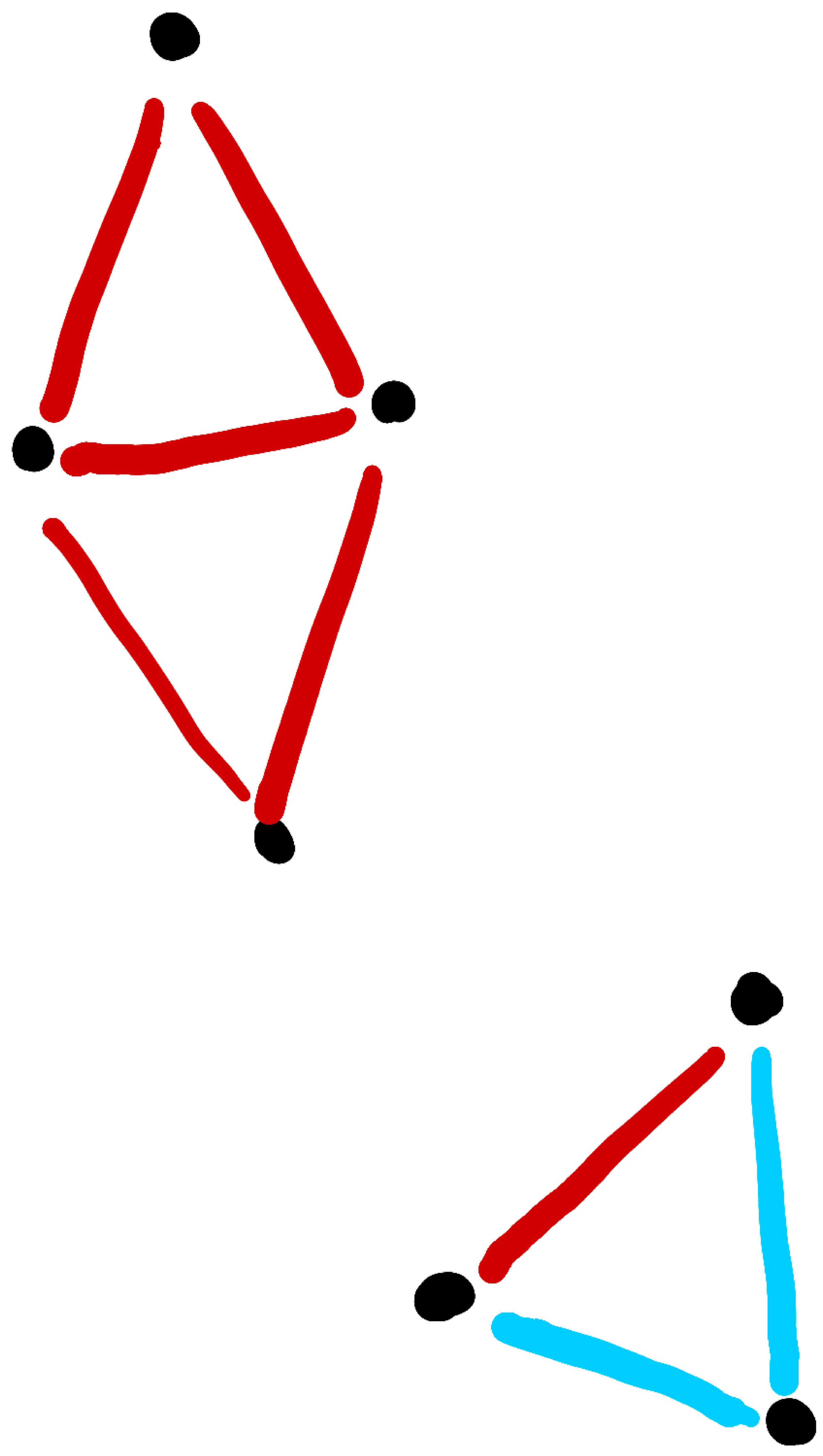
"if $\frac{4}{5}$ of triangle sides yield the bad witness
then $\frac{3}{5}$ of triangle bases yield the bad witness"



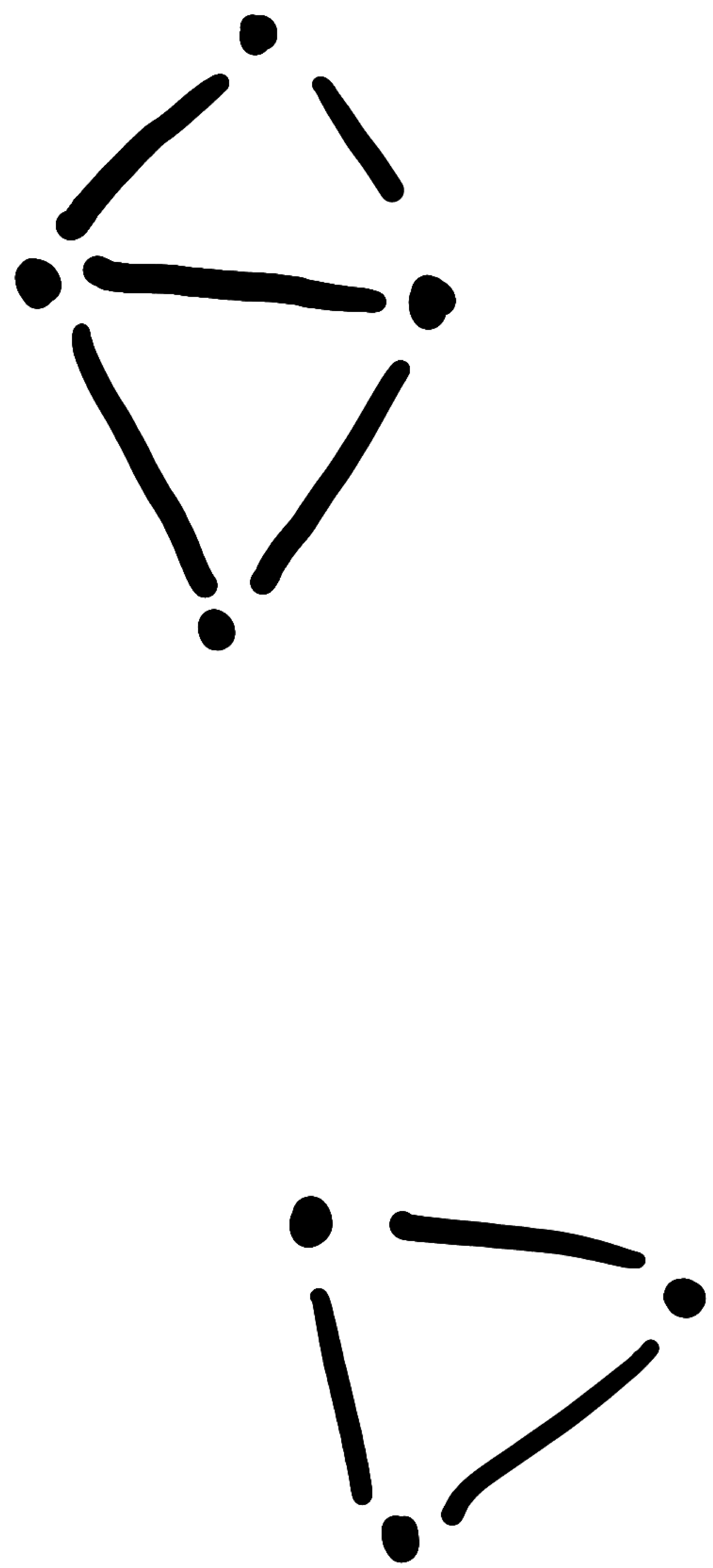
Sharing Triangle "Bases"

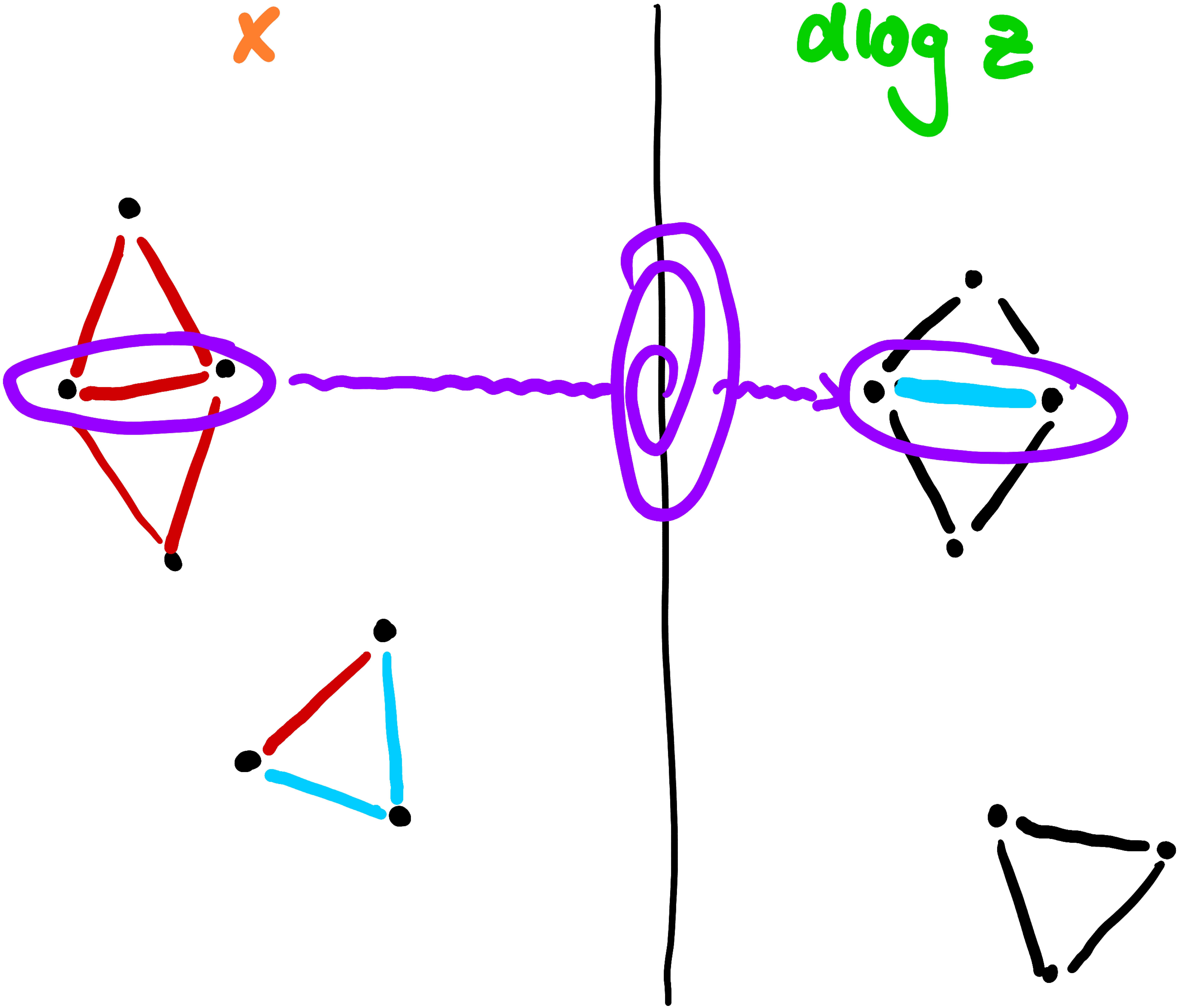


X

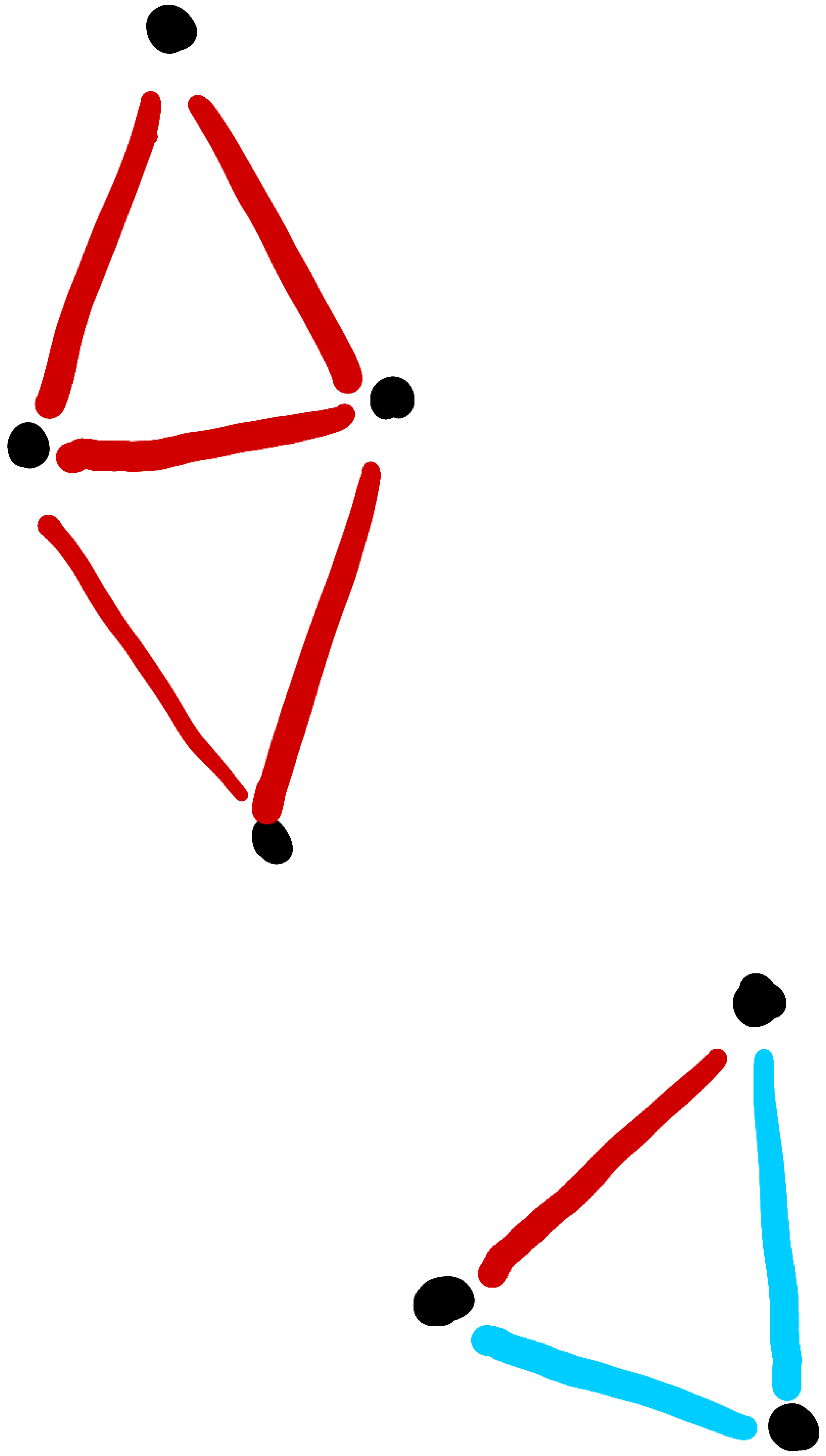


$d\log z$

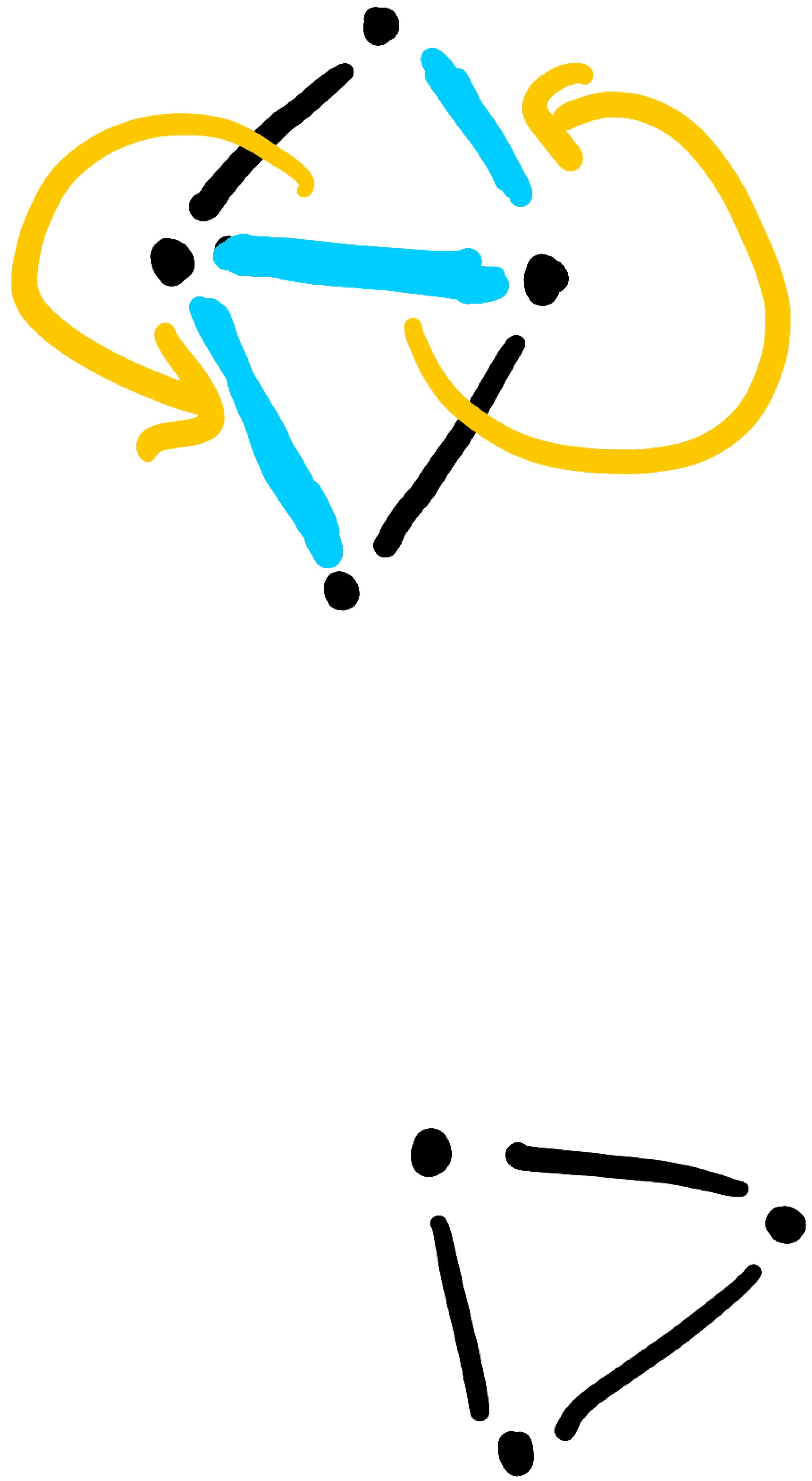



X
dlog z

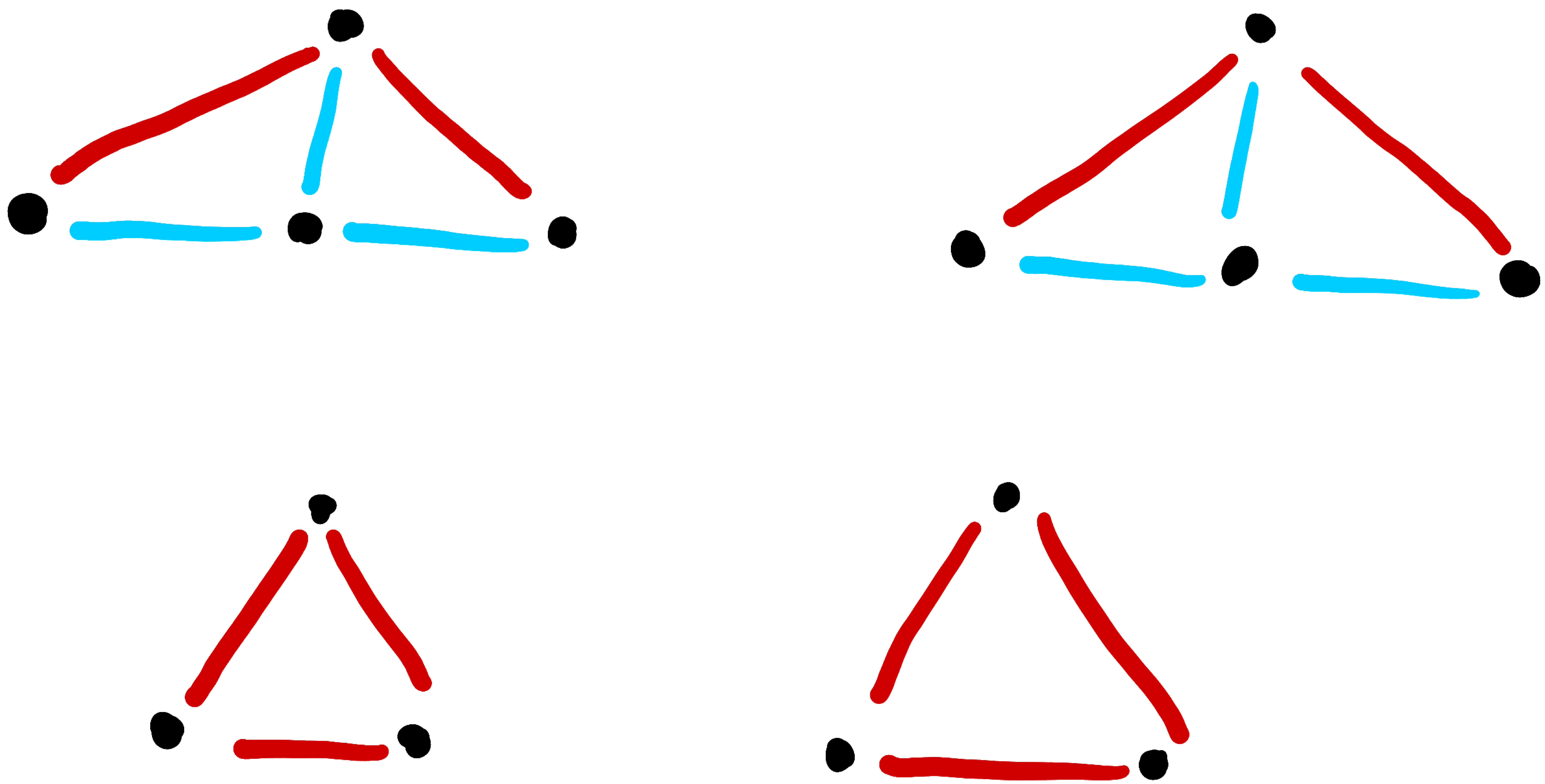
X



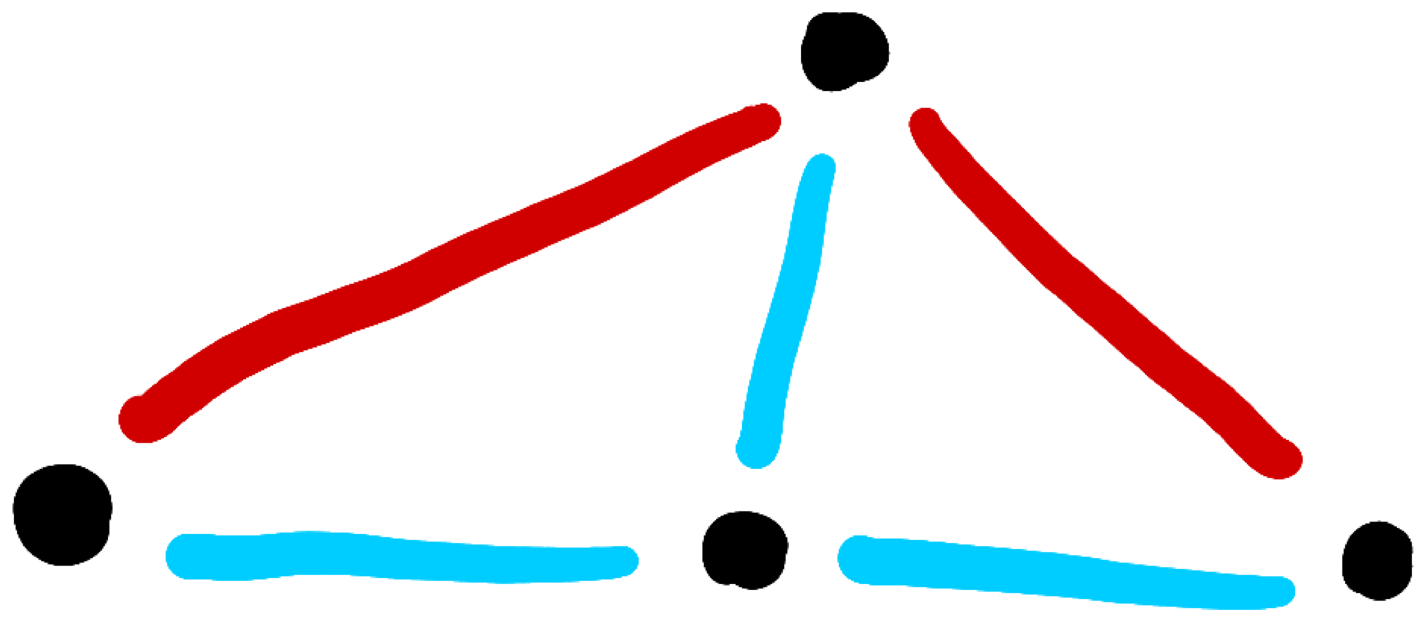
$\mathrm{dlog}\, z$

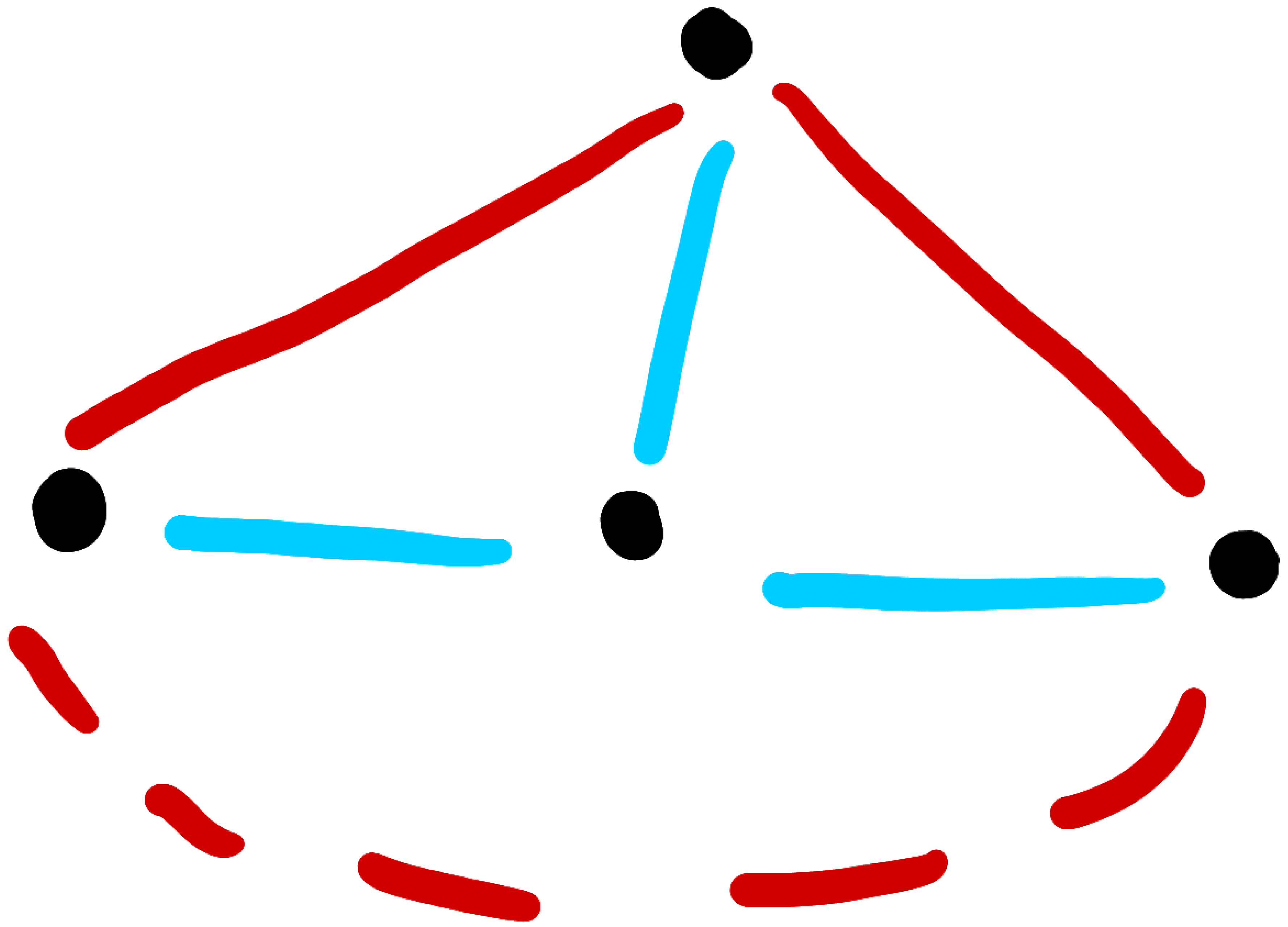


Sharing Triangle "Sides"



Bounding





can't happen "too often"

by combinatorics

Summary

- [AO '00] efficient partially blind signature scheme

Summary

- [AO'00] efficient partially blind signature scheme
- Mending some gaps in proof

Summary

- [AO'00] efficient partially blind signature scheme
- Mending some gaps in proof
- Similar reduction loss to original work

Summary

- [AO '00] efficient partially blind signature scheme
- Mending some gaps in proof
- Similar reduction loss to original work
  - ↳ Small nr of signing sessions

Summary

- [AO '00] efficient partially blind signature scheme
- Mending some gaps in proof
- Similar reduction loss to original work
  ↳ Small nr of signing sessions

Open Questions

- Schemes with polynomial nr of signing sessions?
  ↳ overcome loss

Summary

- [AO '00] efficient partially blind signature scheme
- Mending some gaps in proof
- Similar reduction loss to original work
  ↳ Small nr of signing sessions

Open Questions

- Schemes with polynomial nr of signing sessions?
  ↳ candidate: [Abe 01]