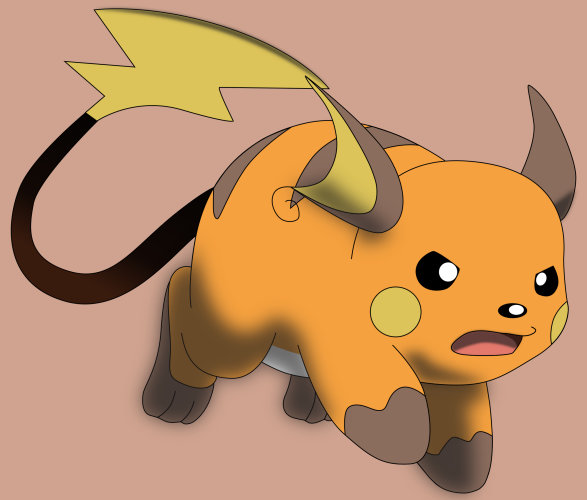
CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Rai-Choo!
Evolving Blind Signatures to the Next Level



Source: https://tinyurl.com/bdy34fzc

Lucjan Hanzlik

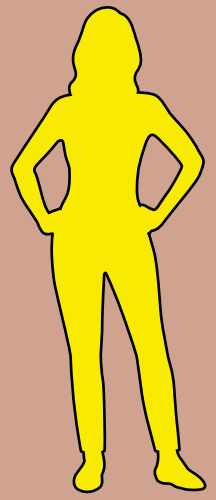
Julian Loss

Benedikt Wagner

Blind Signatures

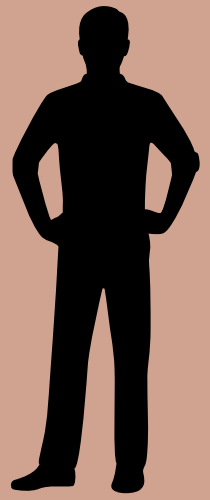
Blind Signatures

pk, sk



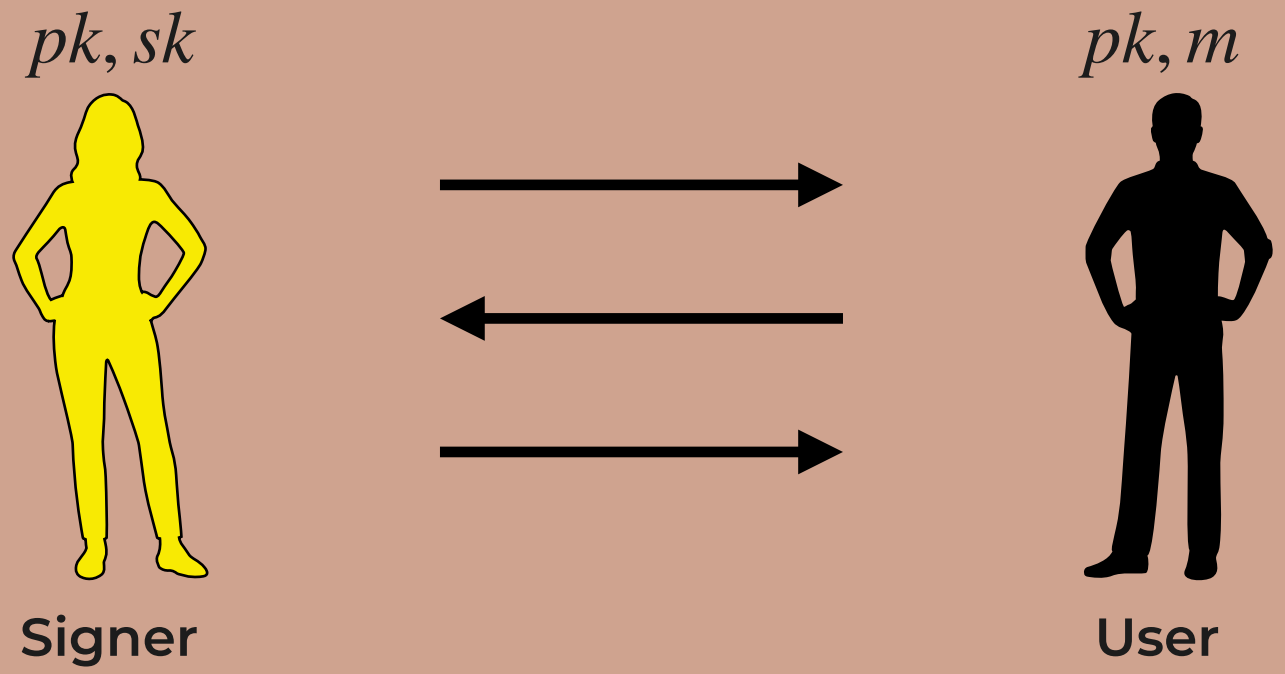
Signer

pk, m

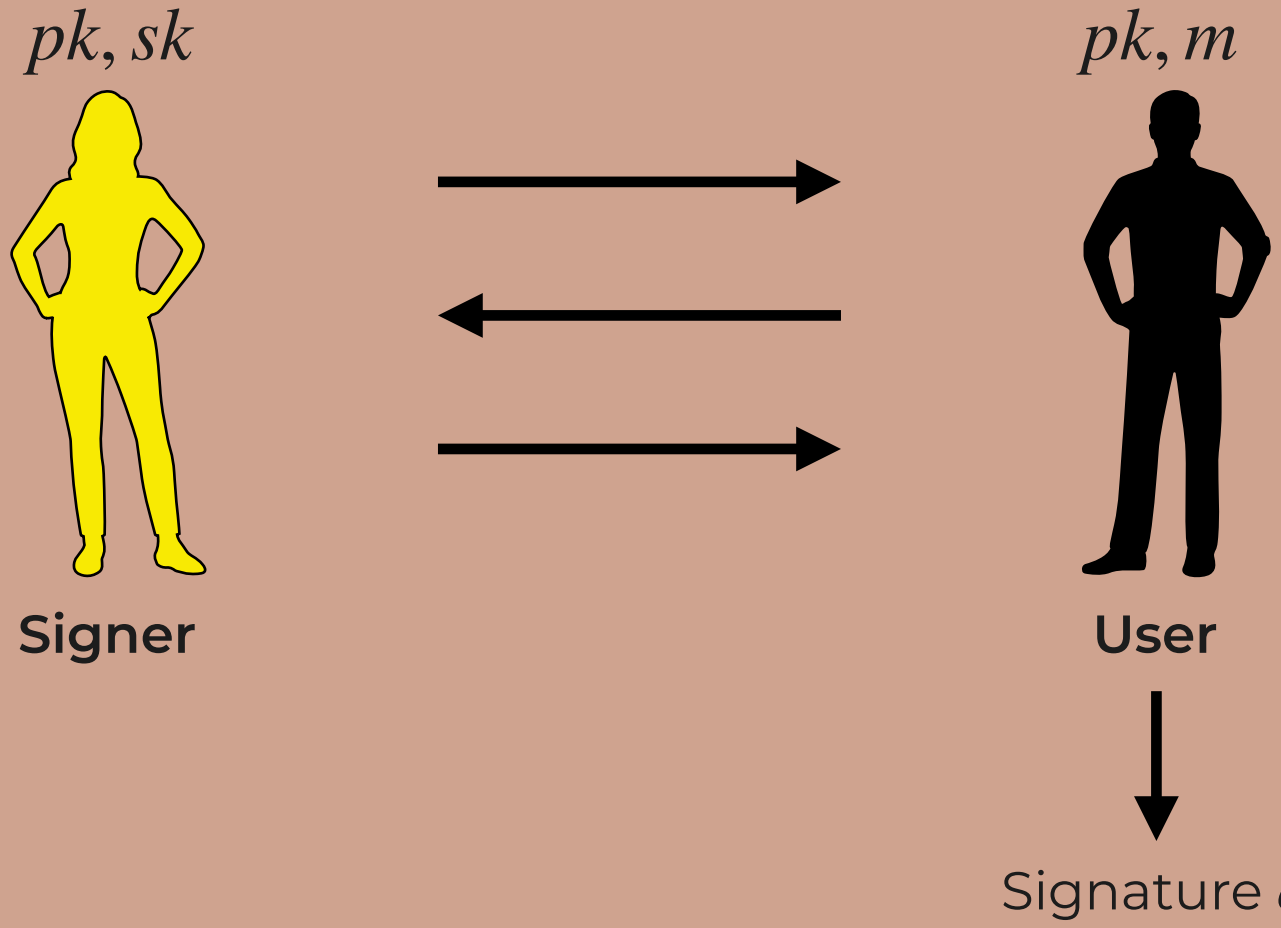


User

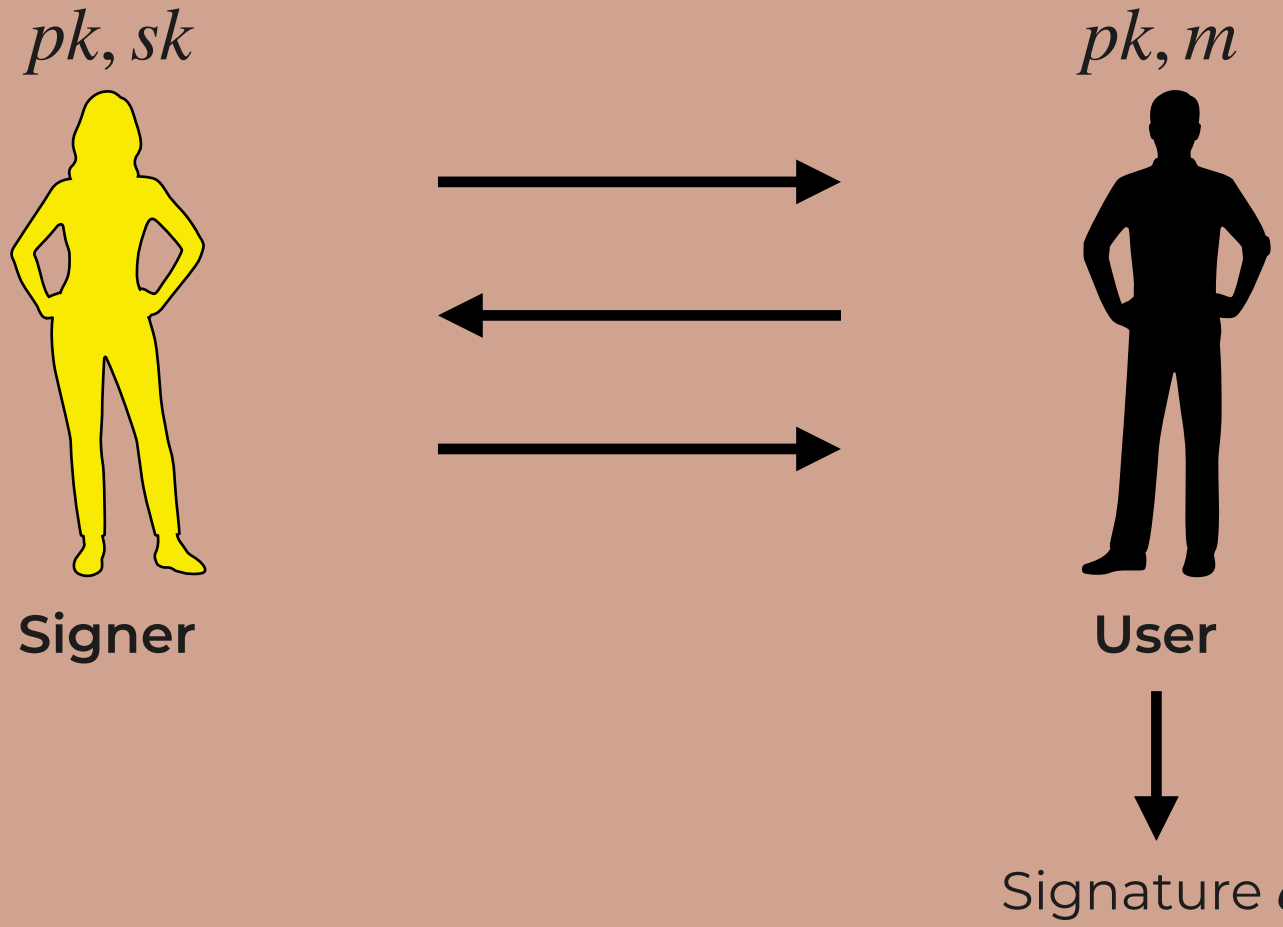
Blind Signatures



Blind Signatures



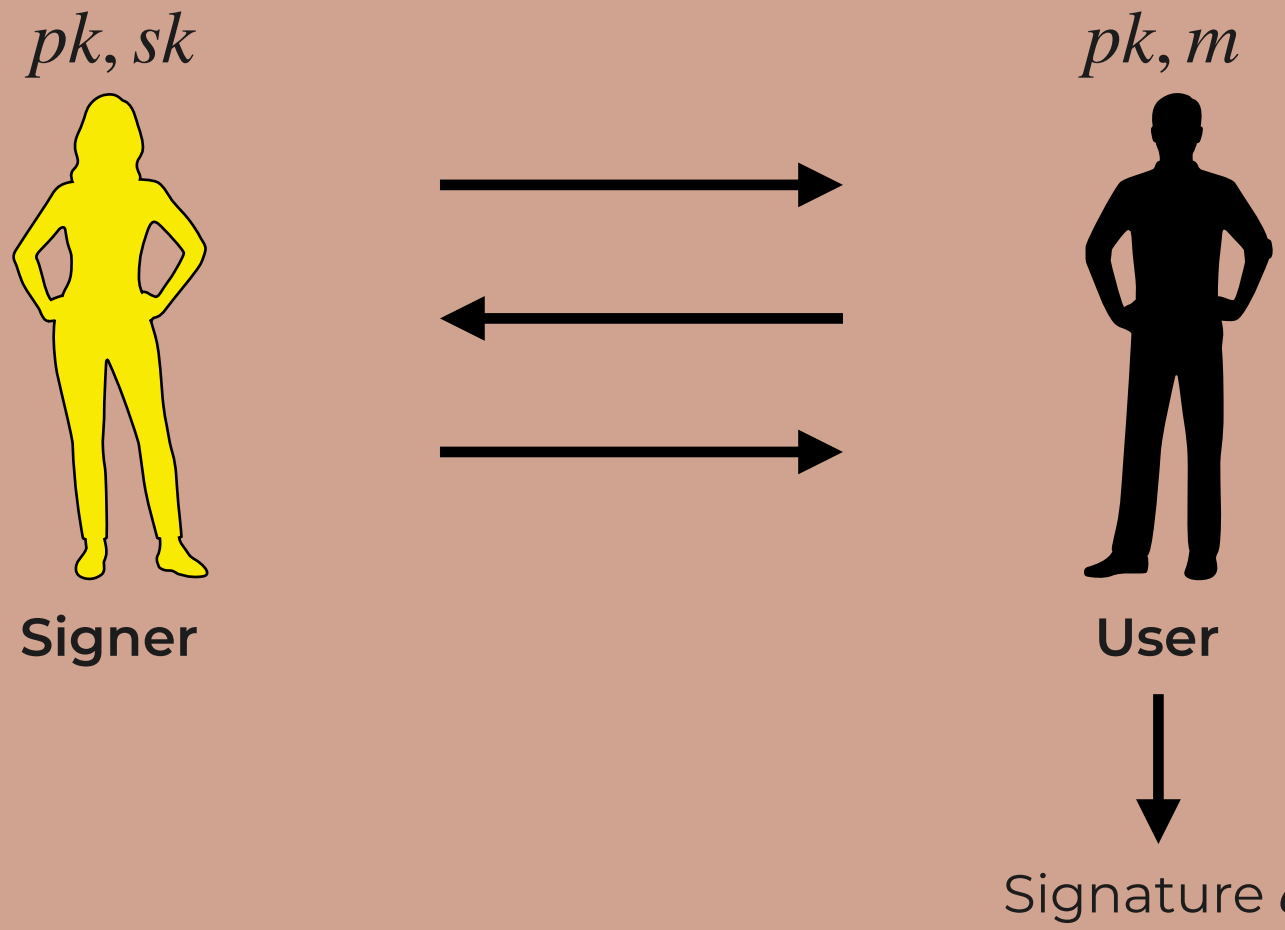
Blind Signatures



Blindness

Signer does not learn message

Blind Signatures



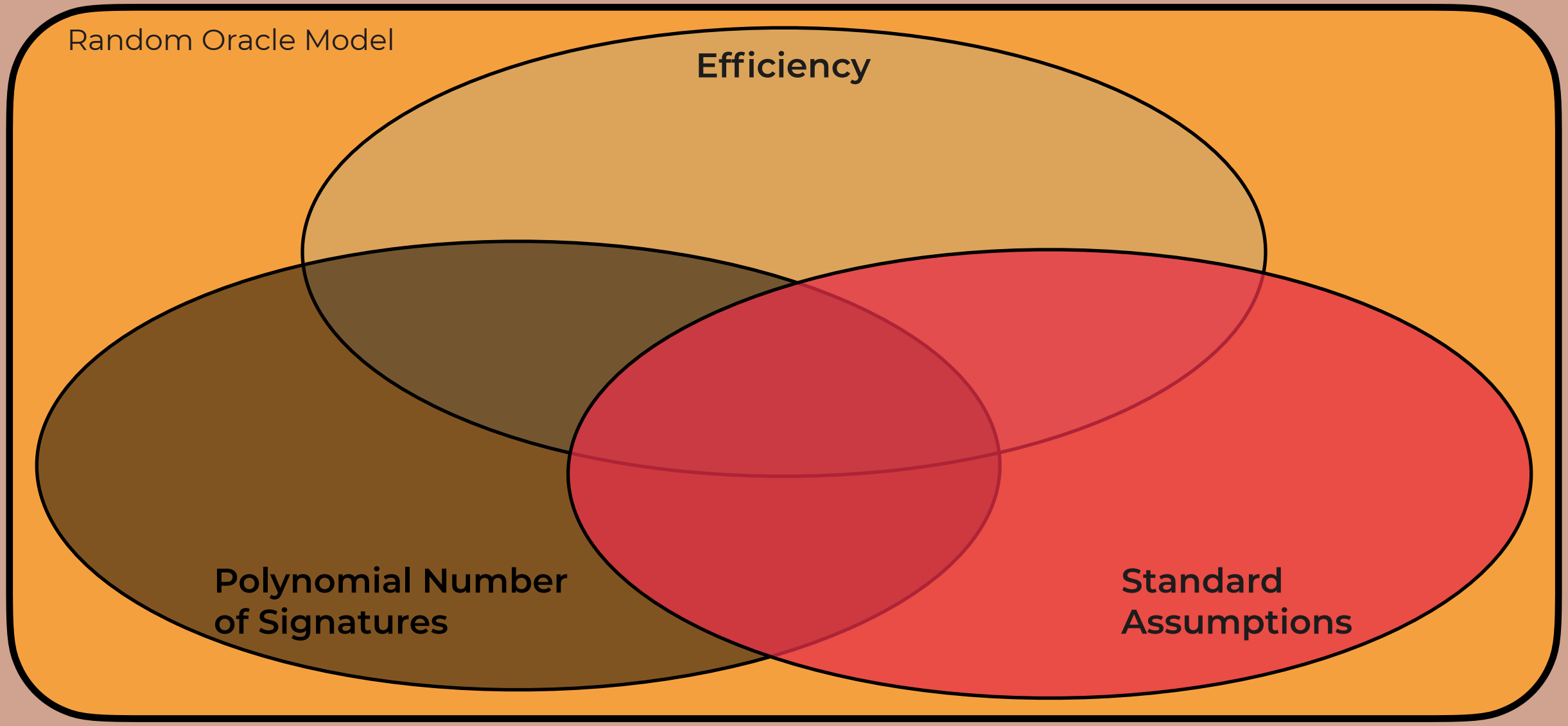
Blindness

Signer does not learn message

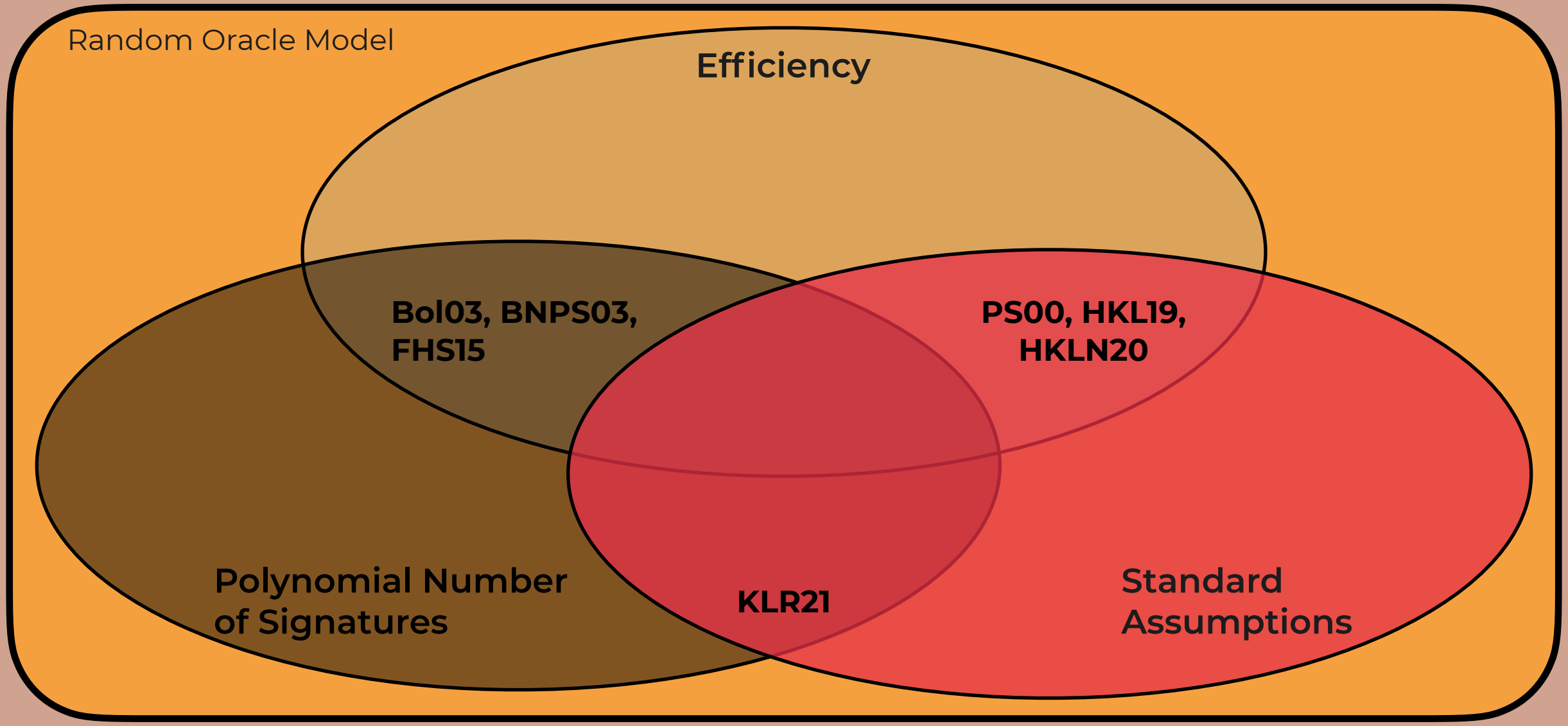
Unforgeability

User needs Signer to get signatures

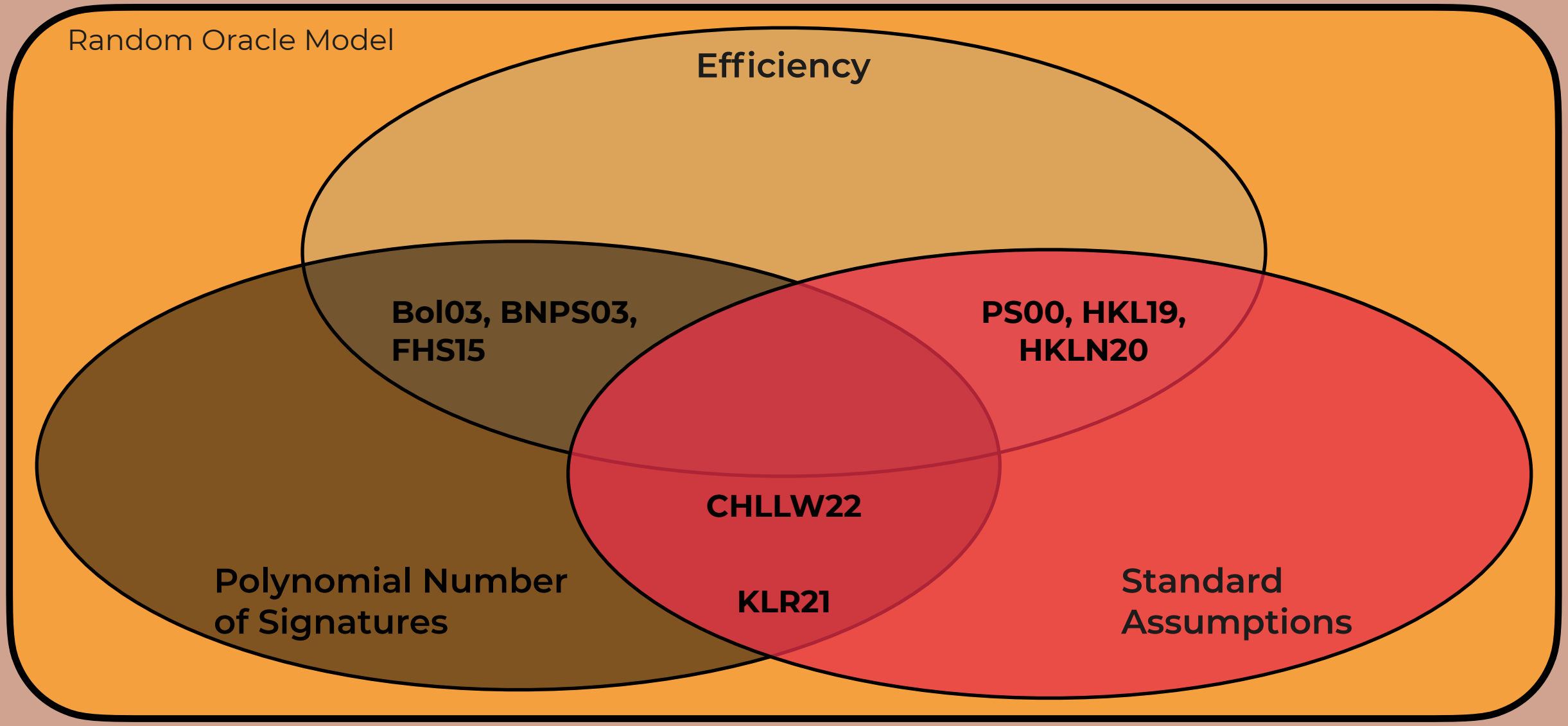
State-of-the-Art



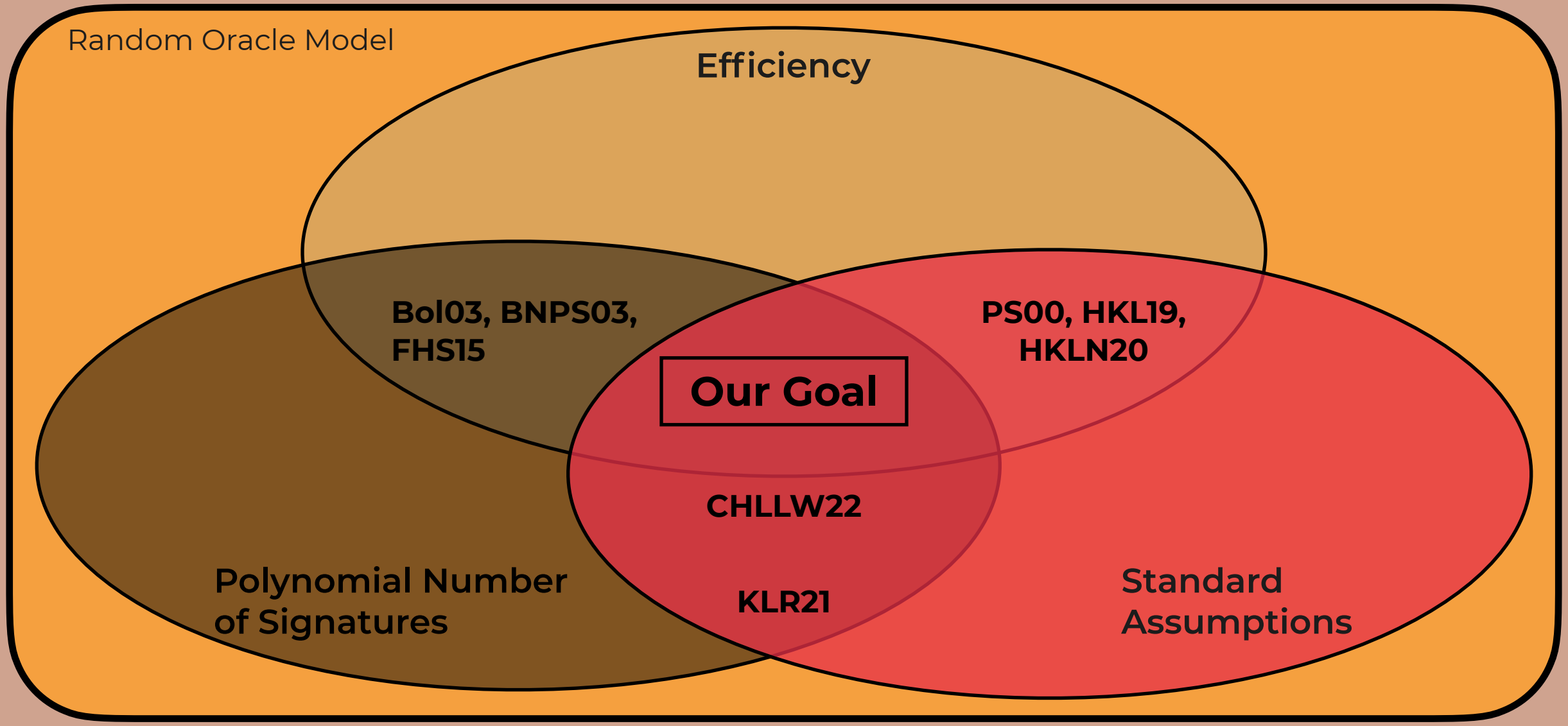
State-of-the-Art



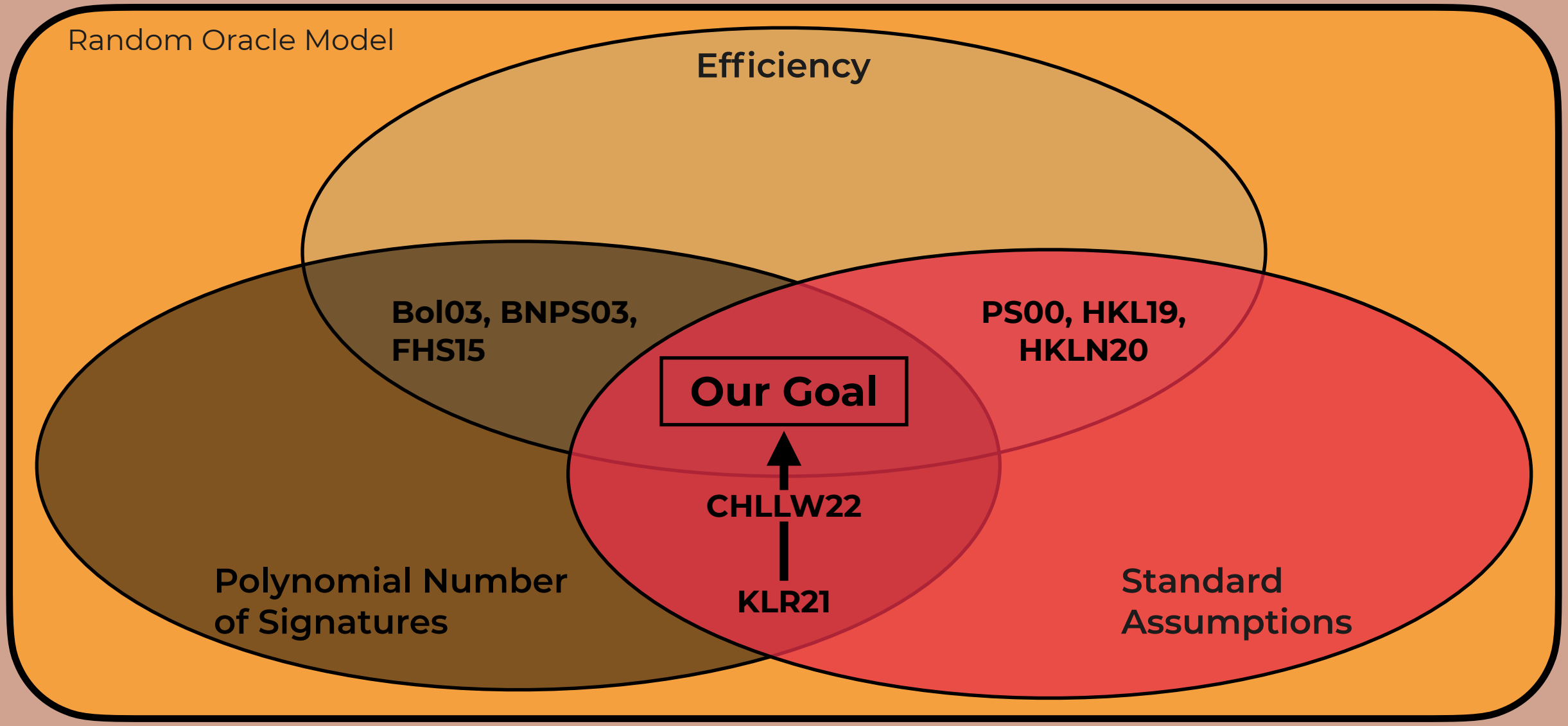
State-of-the-Art



State-of-the-Art



State-of-the-Art



Efficiency

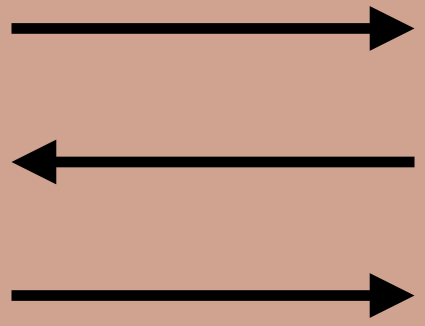
Efficiency

Signature Size

Efficiency

Signature Size

Communication

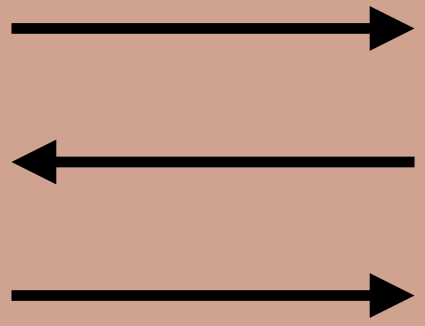


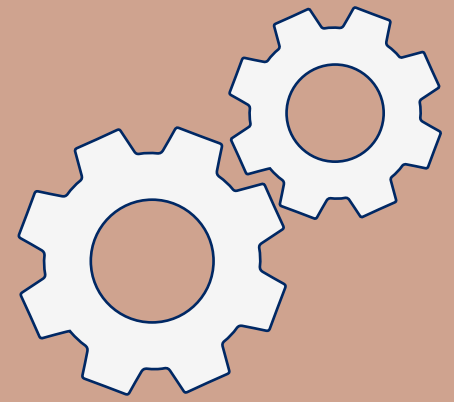
Efficiency

Signature Size

Communication

Computation





The World of Boosting

Boosting Blind Signatures [KLR21]

Boosting Blind Signatures [KLR21]

Log-Secure
Blind Signature

Boosting Blind Signatures [KLR21]

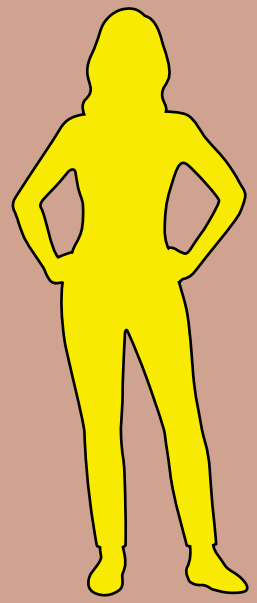
Boosting Blind Signatures [KLR21]

Log-Secure Blind Signature

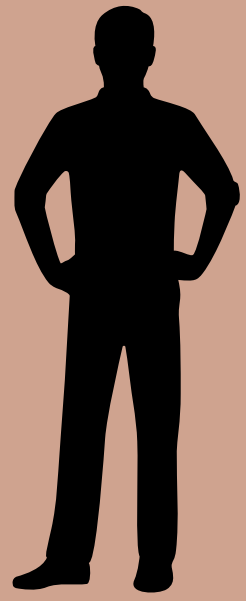
Boosting

KLR21

Fully-Secure Blind Signature



Signer



User

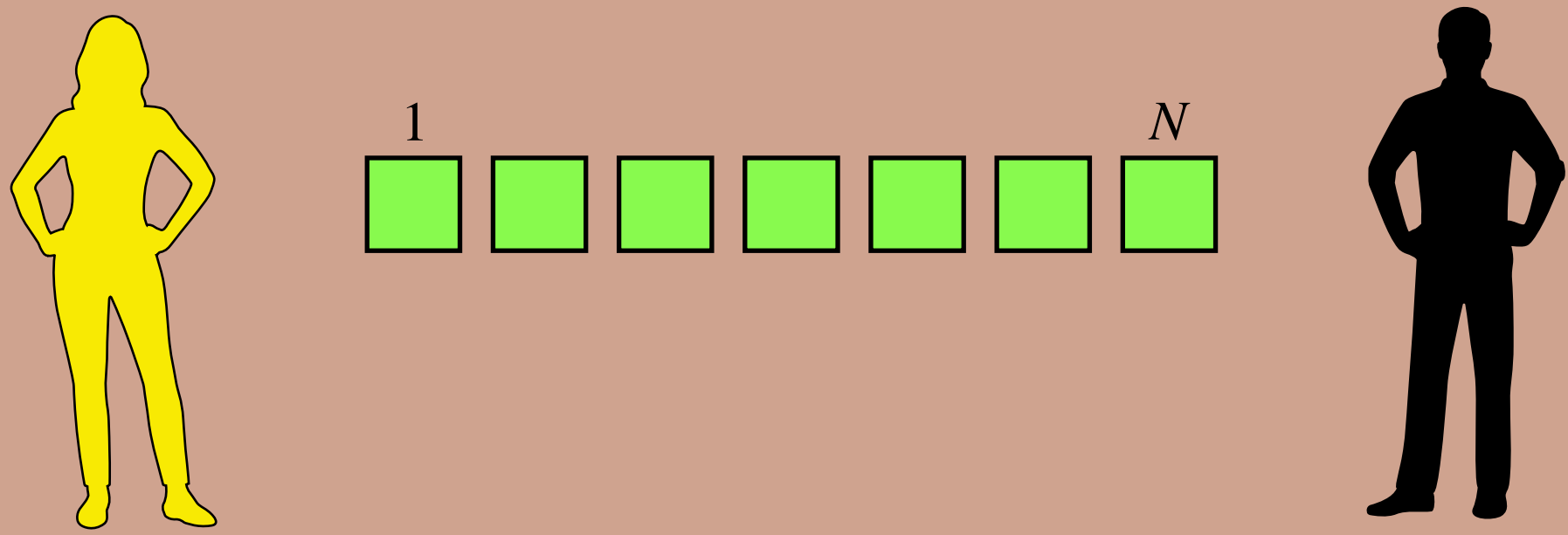
Boosting Blind Signatures [KLR21]

Log-Secure Blind Signature

Boosting

KLR21

Fully-Secure Blind Signature



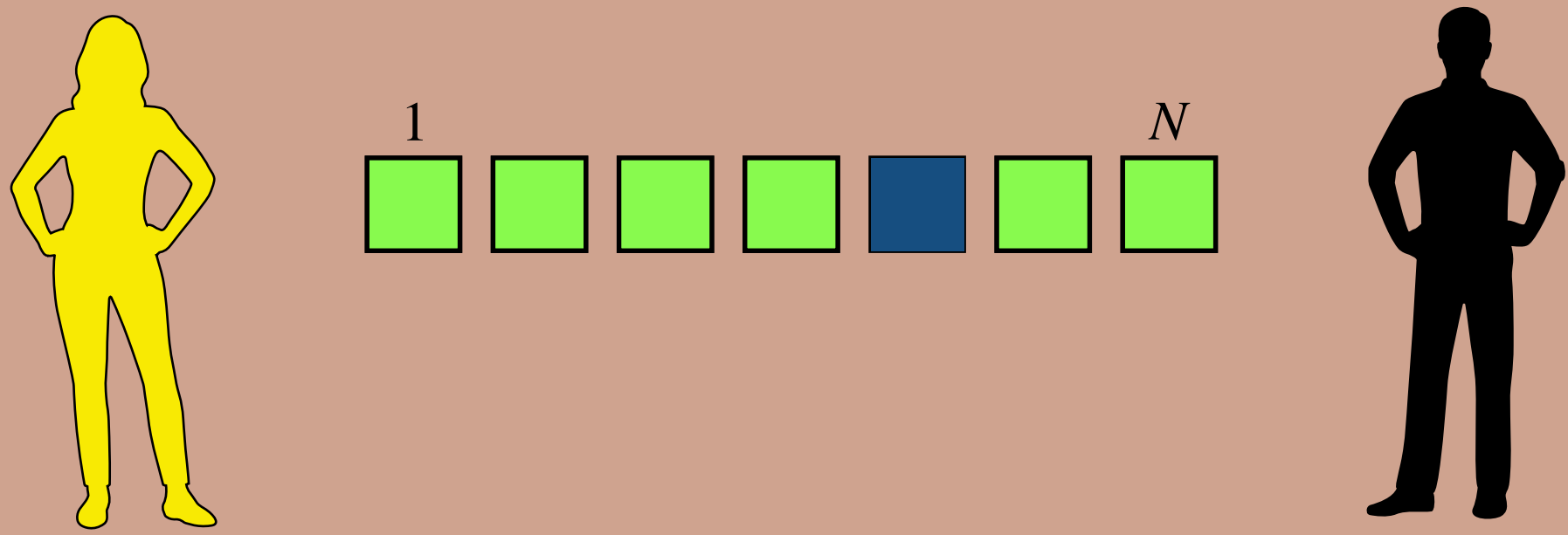
Boosting Blind Signatures [KLR21]

Log-Secure Blind Signature

Boosting

KLR21

Fully-Secure Blind Signature



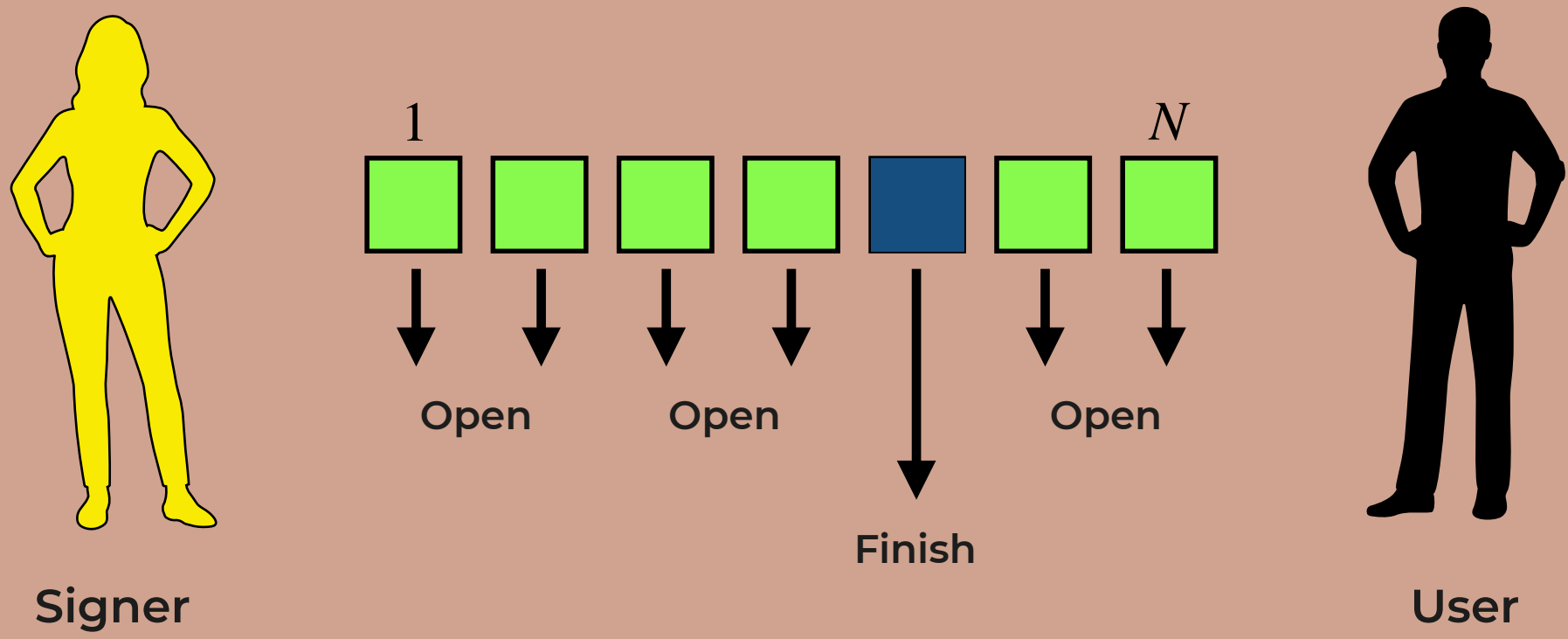
Boosting Blind Signatures [KLR21]

Log-Secure Blind Signature

Boosting

KLR21

Fully-Secure Blind Signature



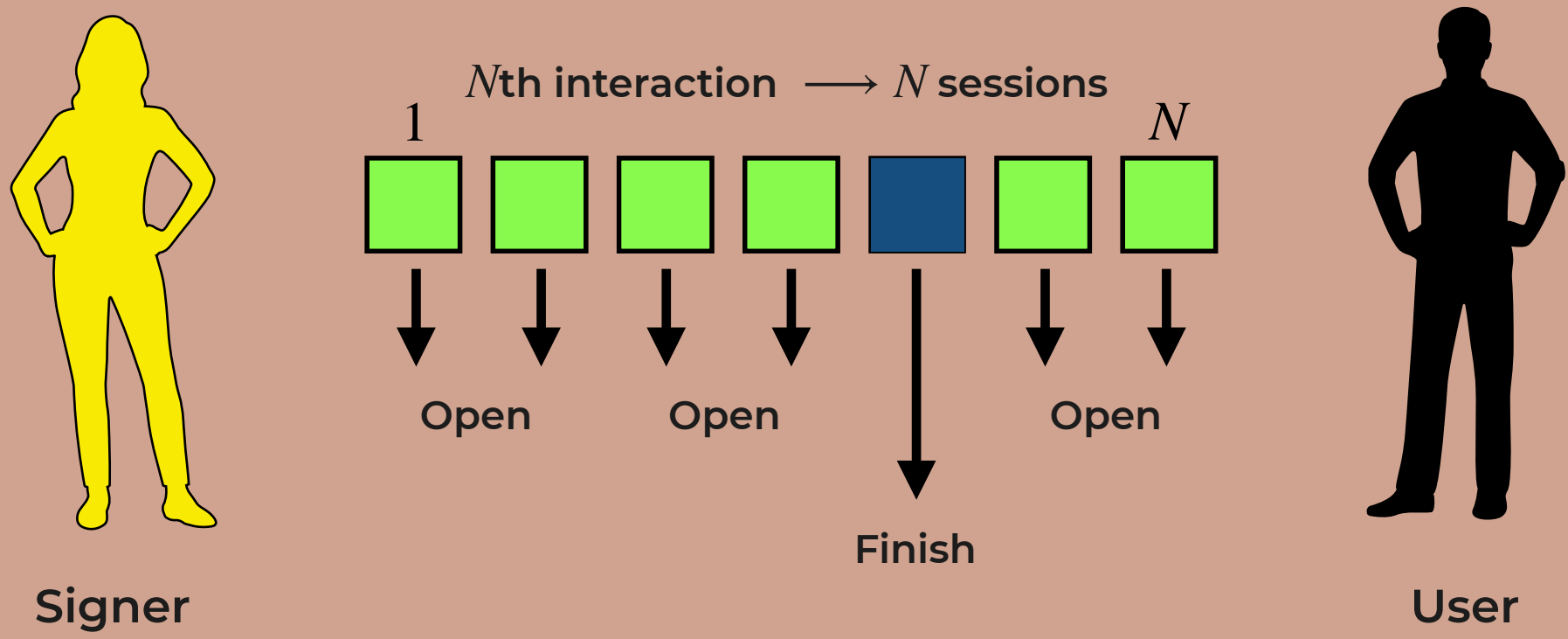
Boosting Blind Signatures [KLR21]

Log-Secure Blind Signature

Boosting

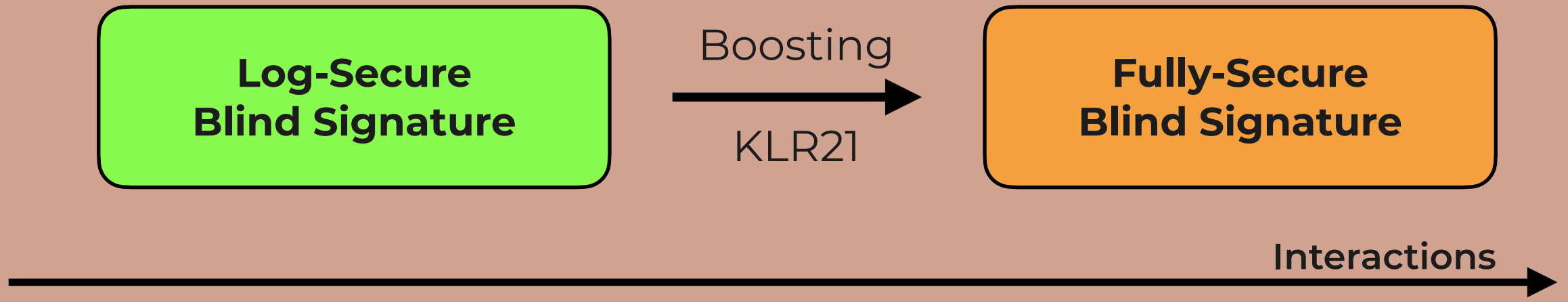
KLR21

Fully-Secure Blind Signature

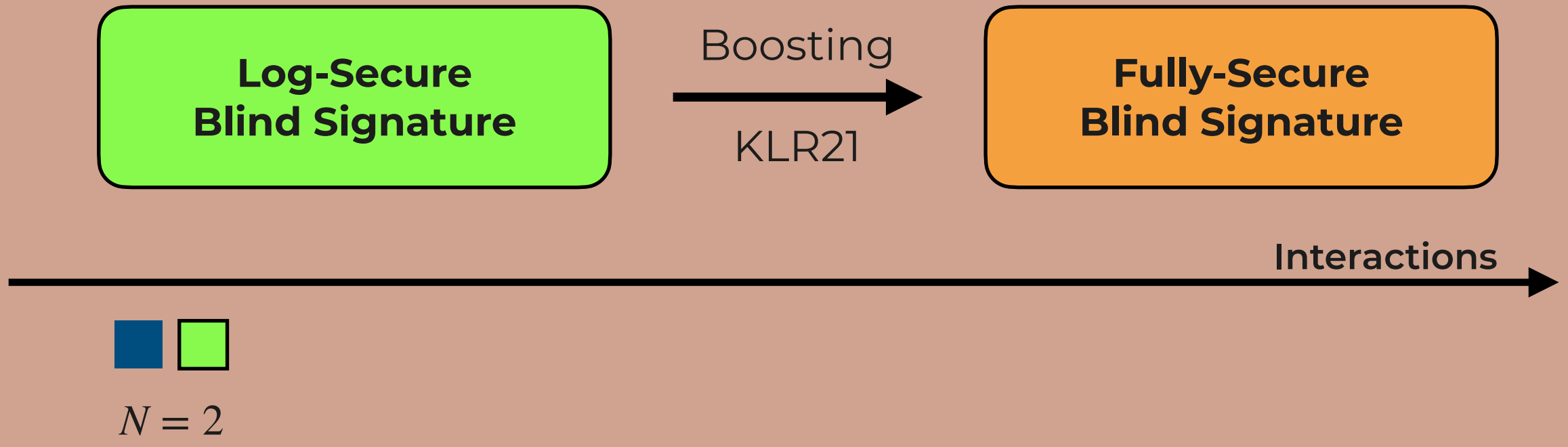


Boosting Blind Signatures [KLR21]

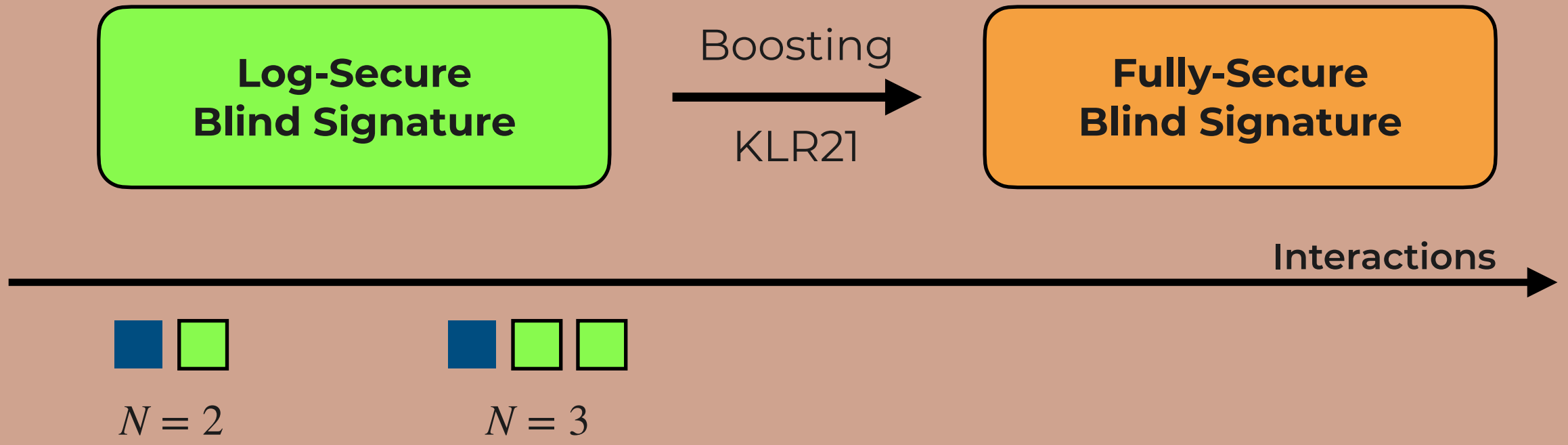
Boosting Blind Signatures [KLR21]



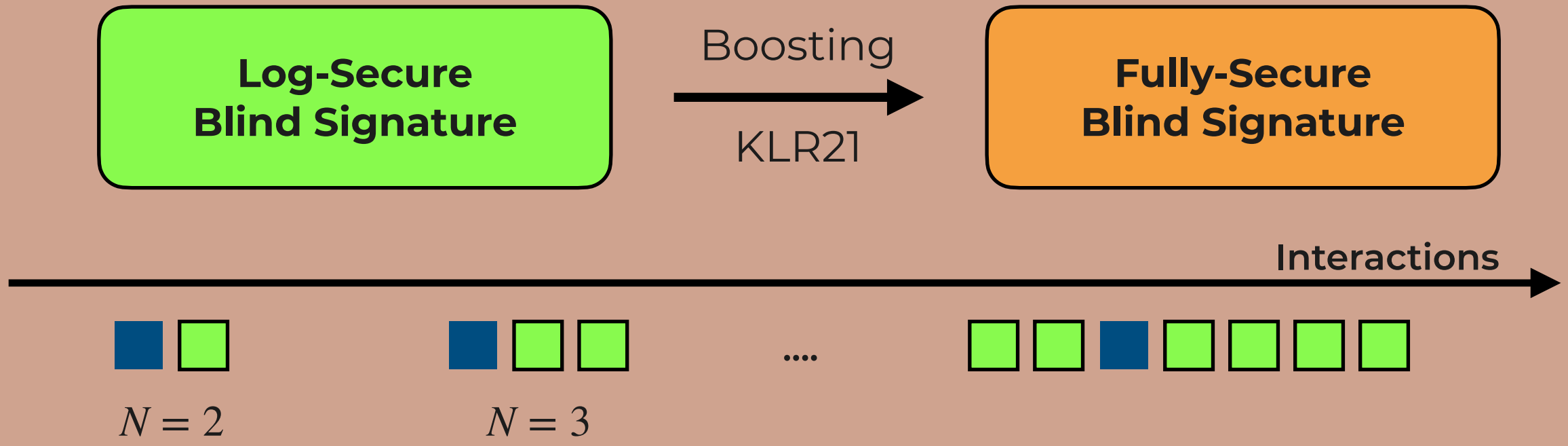
Boosting Blind Signatures [KLR21]



Boosting Blind Signatures [KLR21]



Boosting Blind Signatures [KLR21]



Boosting Blind Signatures [KLR21]

Boosting Blind Signatures [KLR21]

User malforms …

… more than one session

Boosting Blind Signatures [KLR21]

User malforms ...

... more than one session  Detected

Boosting Blind Signatures [KLR21]

Log-Secure Blind Signature

Boosting

KLR21

Fully-Secure Blind Signature

User malforms ...

... more than one session

Detected

... no session

Boosting Blind Signatures [KLR21]

Log-Secure Blind Signature

Boosting

KLR21

Fully-Secure Blind Signature

User malforms ...

... more than one session

Detected

... no session

Can simulate

Boosting Blind Signatures [KLR21]

Log-Secure Blind Signature

Boosting

KLR21

Fully-Secure Blind Signature

User malforms ...

... more than one session

Detected

... no session

Can simulate

... one session

Boosting Blind Signatures [KLR21]

Log-Secure Blind Signature

Boosting

KLR21

Fully-Secure Blind Signature

User malforms ...

... more than one session

Detected

... no session

Can simulate

... one session

Cheat ⟶ Signer Oracle

Boosting Blind Signatures [KLR21]

Log-Secure Blind Signature

Boosting

KLR21

Fully-Secure Blind Signature

User malforms ...

... more than one session

Detected

... no session

Can simulate

... one session

Cheat ⟶ Signer Oracle

Logarithmic Number of Cheats

Compact Boosting

Log-Secure Blind Signature

Boosting

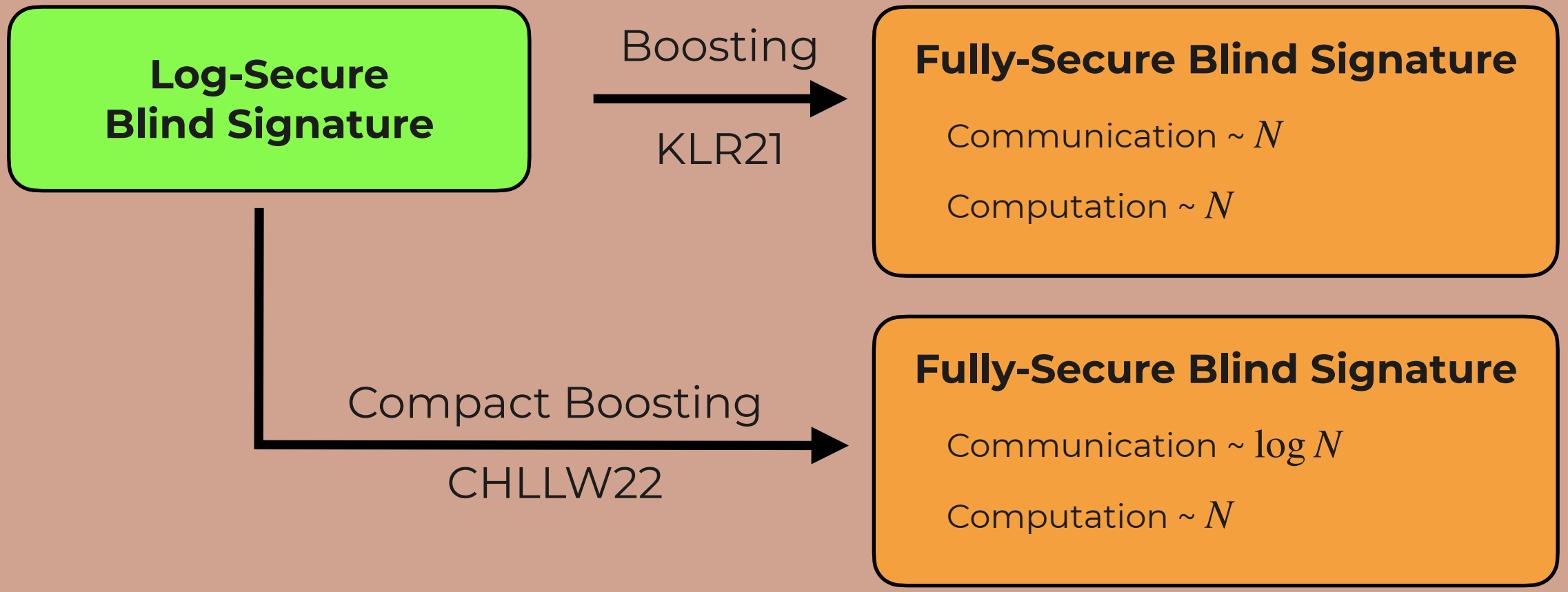
KLR21

Fully-Secure Blind Signature

Communication ~ N

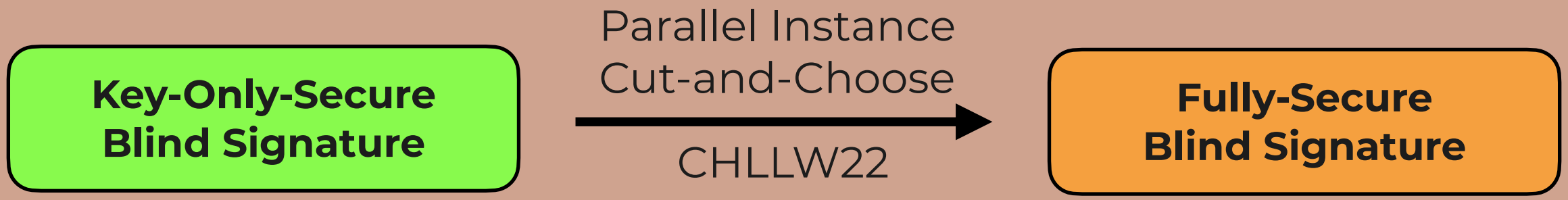
Computation ~ N

Compact Boosting



PI-Cut-Choo [CHLLW22]

PI-Cut-Choo [CHLLW22]



PI-Cut-Choo [CHLLW22]

Key-Only-Secure Blind Signature

Parallel Instance Cut-and-Choose

CHLLW22

Fully-Secure Blind Signature

Interactions

PI-Cut-Choo [CHLLW22]

Key-Only-Secure Blind Signature

Parallel Instance Cut-and-Choose

CHLLW22

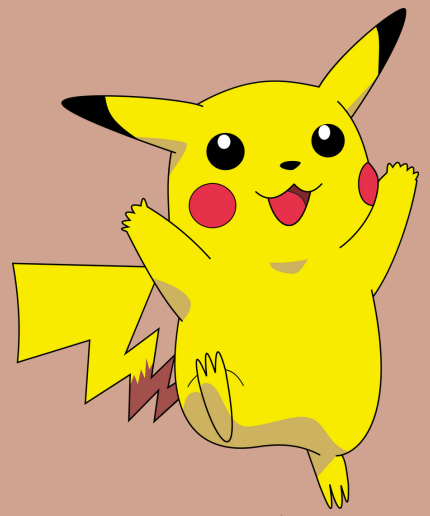
Fully-Secure Blind Signature

Interactions

There is an instance i^* with no cheats.

Our Result: Boosting without State

Our Result: Boosting without State



Source: shorturl.at/cDZ06

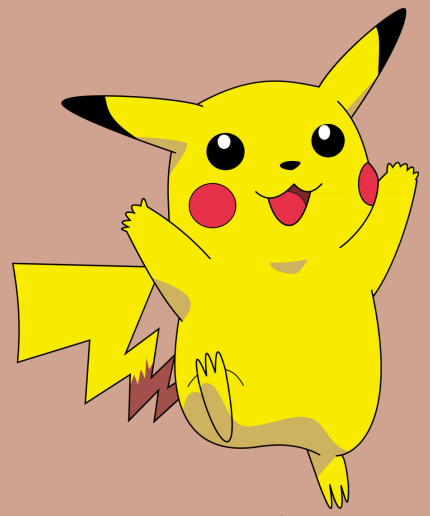
PI-Cut-Choo

7 Rounds

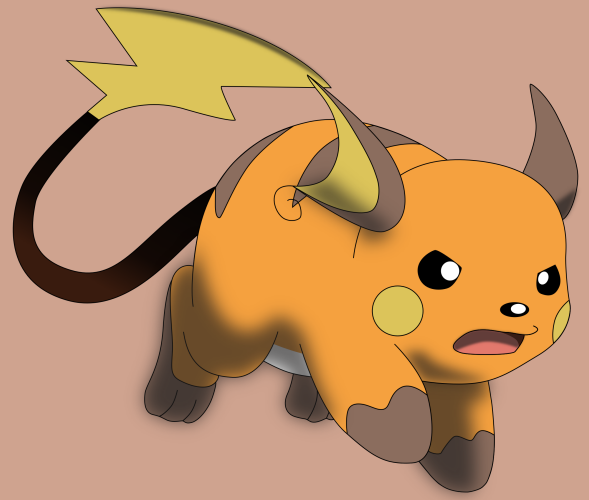
Communication ~ $\log N$

Computation ~ N

Our Result: Boosting without State



Source: shorturl.at/cDZ06



Source: https://tinyurl.com/bdy34fzc

PI-Cut-Choo

7 Rounds

Communication ~ $\log N$

Computation ~ N

Rai-Choo

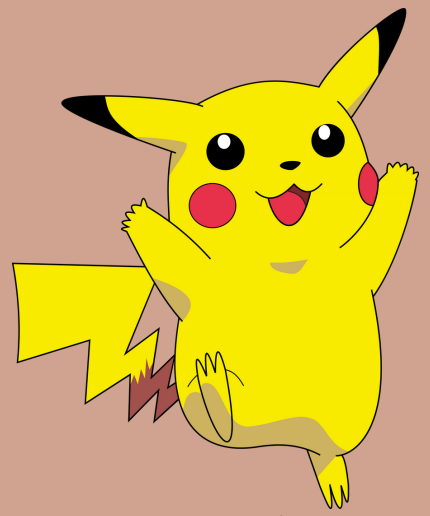
2 Rounds

Stateless

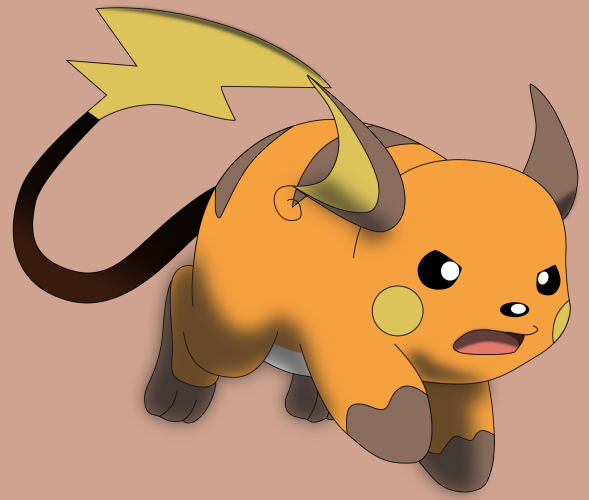
Communication ~ const

Computation ~ const

Our Result: Boosting without State



Source: shorturl.at/cDZ06



Source: https://tinyurl.com/bdy34fzc

PI-Cut-Choo

7 Rounds

Communication ~ $\log N$

Computation ~ N

Rai-Choo

2 Rounds

Stateless

Communication ~ const

Computation ~ const

Additionally:

- **Batching**
- **Partial Blindness**

The World of Boosting

KLR21

7 Rounds

Communication ~ N

Computation ~ N

CHLLW22

5 or 7 Rounds

Communication ~ $\log N$

Computation ~ N

Rai-Choo

2 Rounds

Stateless

Communication ~ const

Computation ~ const

Our Techniques

Attempt I

Attempt I

PI-Cut-Choo

There is an instance i^* with no cheats.

Attempt I

PI-Cut-Choo

There is an instance i^* with no cheats.

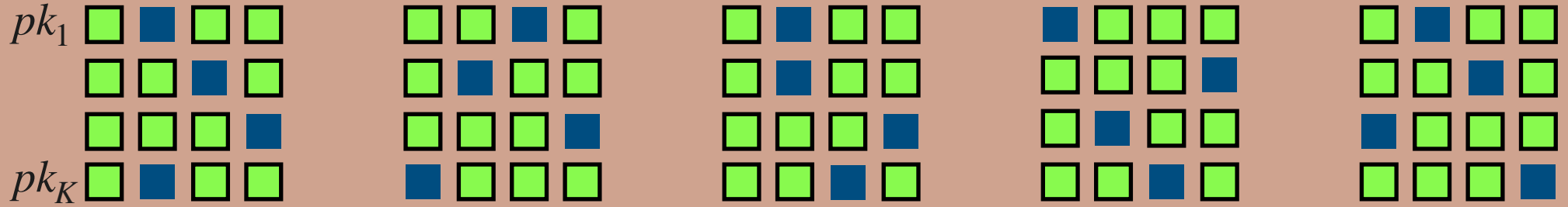
Naive Attempt I: PI-Cut-Choo without State

Attempt I

PI-Cut-Choo

There is an instance i^* with no cheats.

Naive Attempt I: PI-Cut-Choo without State

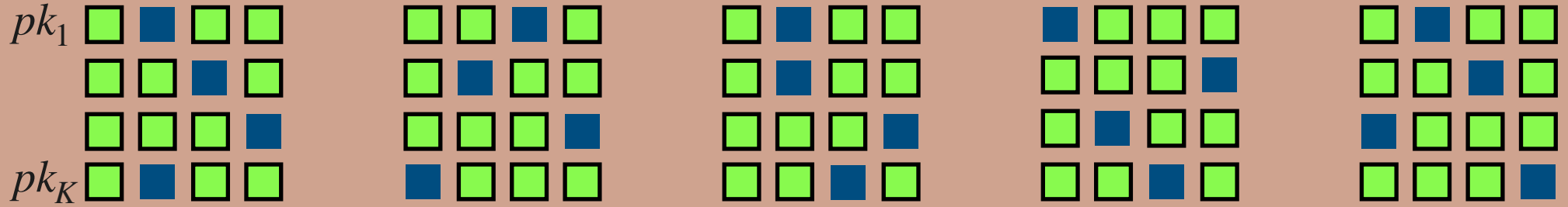


Attempt I

PI-Cut-Choo

There is an instance i^* with no cheats.

Naive Attempt I: PI-Cut-Choo without State



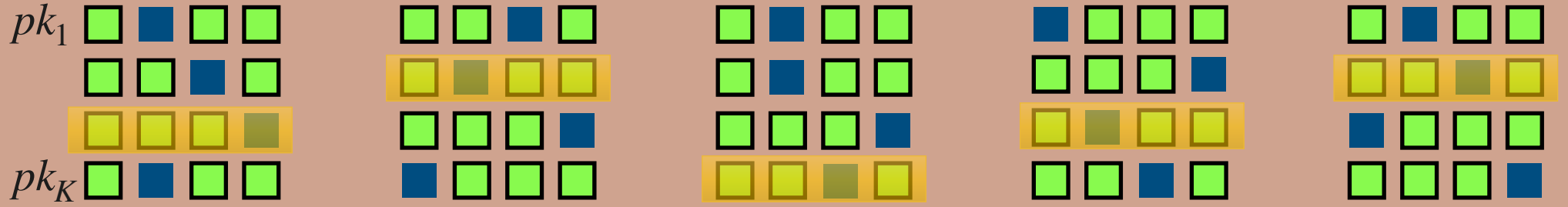
For every interaction, there is an instance i^* with no cheat.

Attempt I

PI-Cut-Choo

There is an instance i^* with no cheats.

Naive Attempt I: PI-Cut-Choo without State



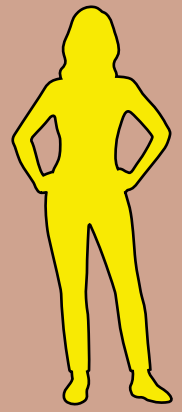
For every interaction, there is an instance i^* with no cheat.

Attempt II

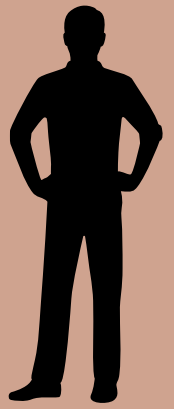
Naive Attempt II: New keys in each interaction

Attempt II

Naive Attempt II: New keys in each interaction



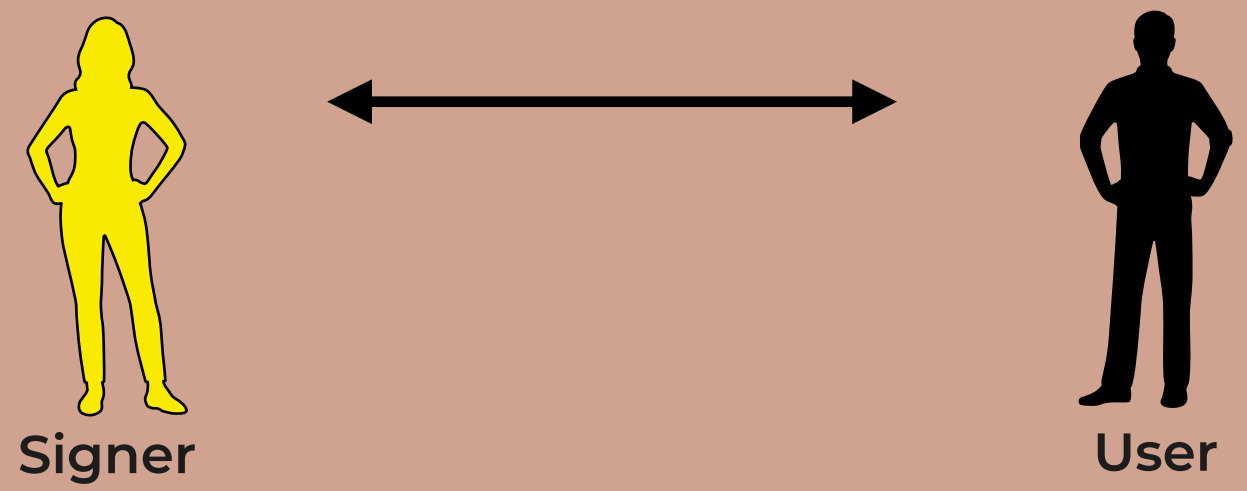
Signer



User

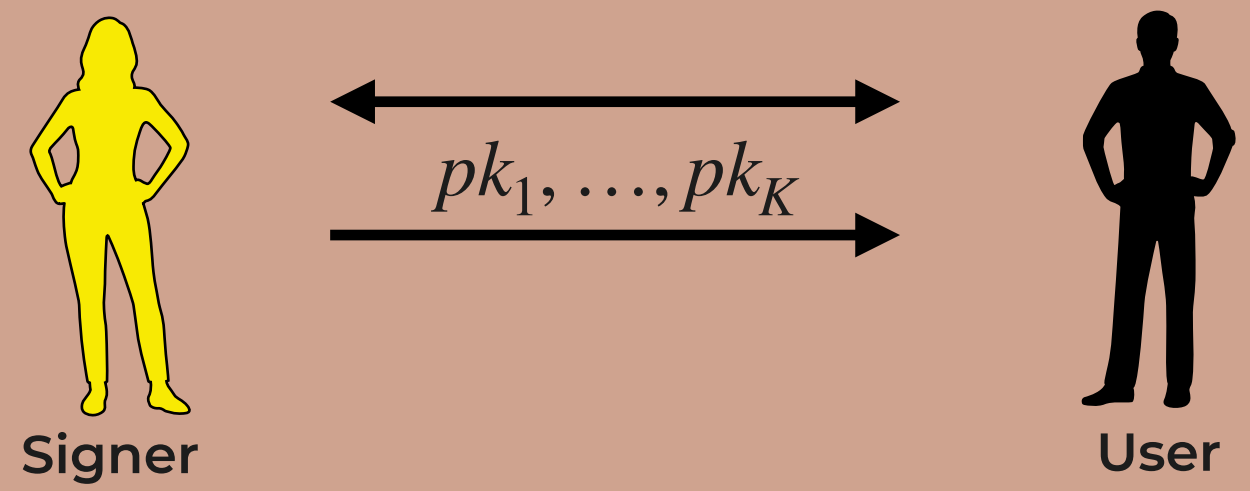
Attempt II

Naive Attempt II: New keys in each interaction



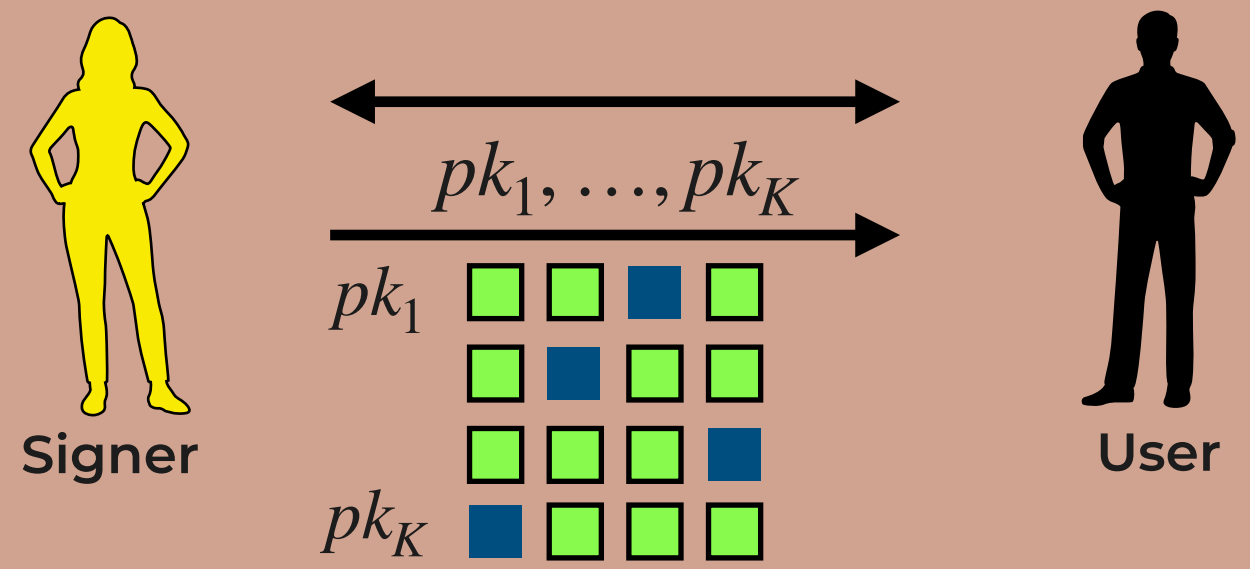
Attempt II

Naive Attempt II: New keys in each interaction



Attempt II

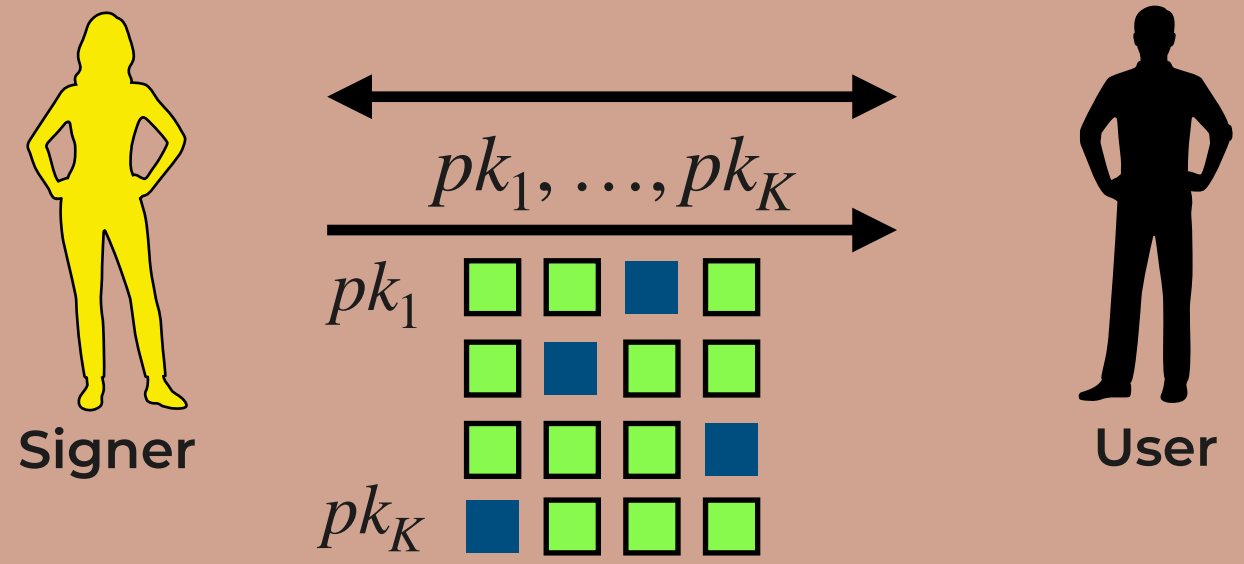
Naive Attempt II: New keys in each interaction



Attempt II

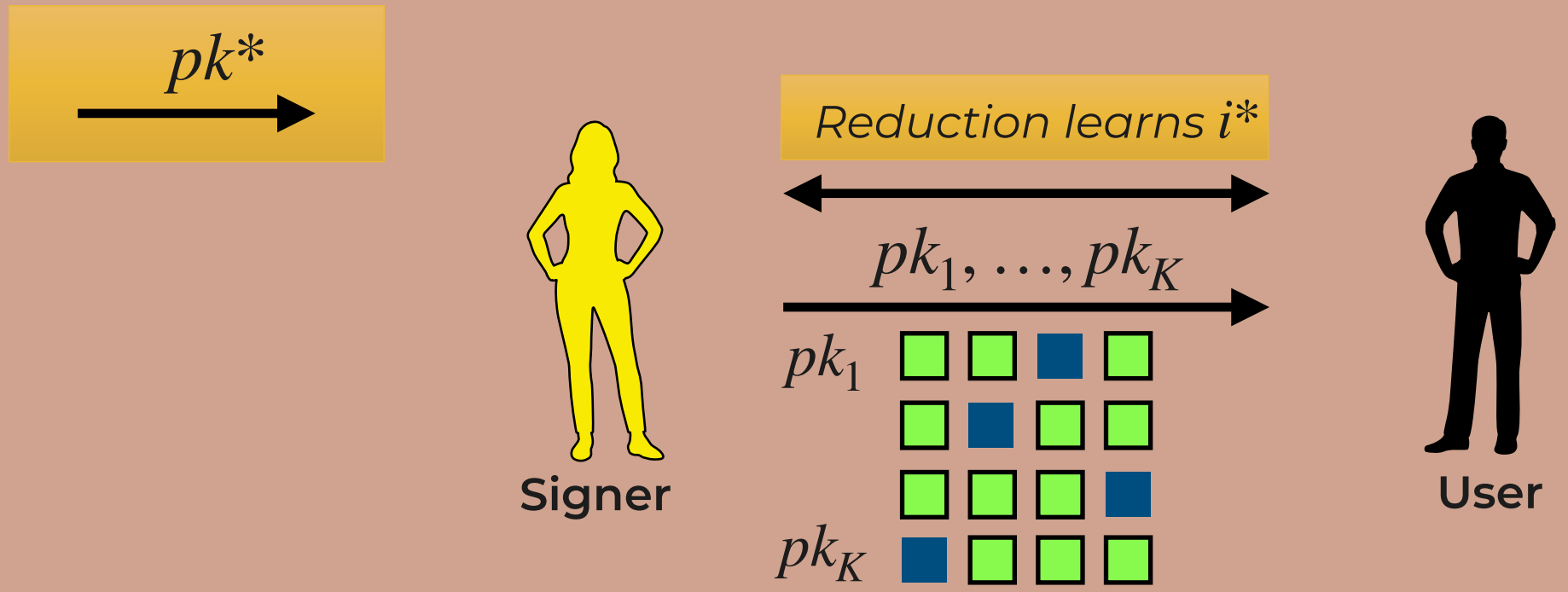
Naive Attempt II: New keys in each interaction

pk^* →



Attempt II

Naive Attempt II: New keys in each interaction

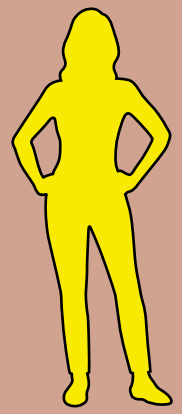


Attempt II

Naive Attempt II: New keys in each interaction

pk^*

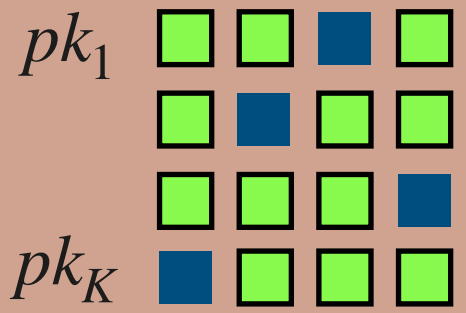
$pk_{i^*} := pk^*$

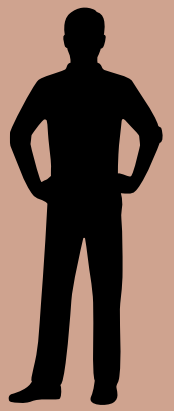


Signer

Reduction learns i^*

$pk_1, \ldots, pk_K$





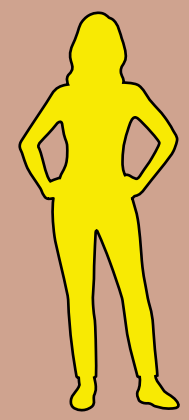
User

Attempt II

Naive Attempt II: New keys in each interaction

pk^*

$pk_{i^*} := pk^*$



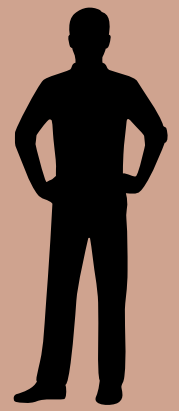
Signer

Reduction learns i^*

$pk_1, \ldots, pk_K$

pk_1

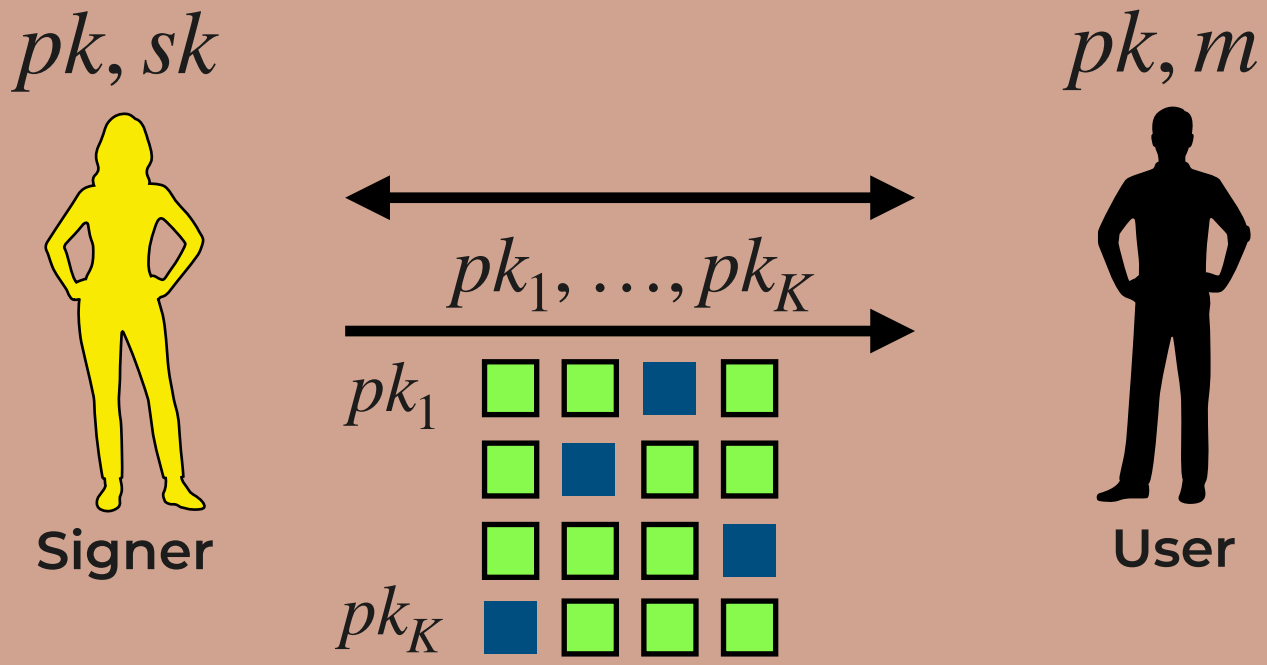
pk_K



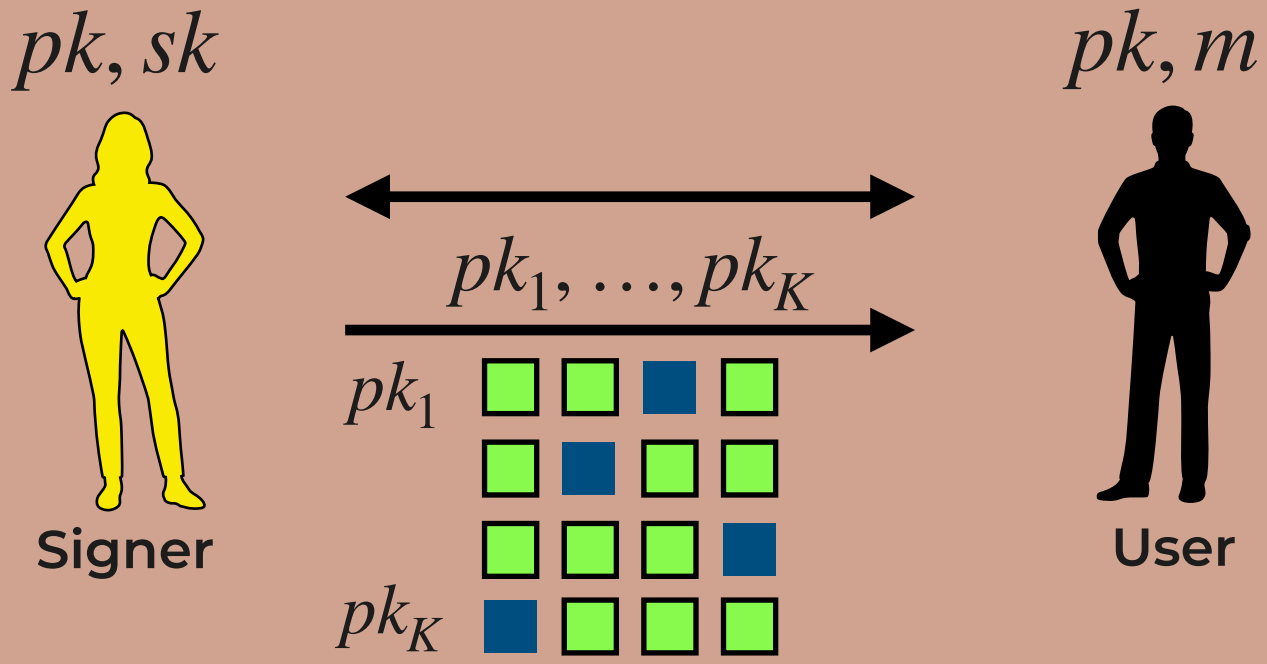
User

Relation to long-term pk, sk ?

Boosting without State: Rai-Choo



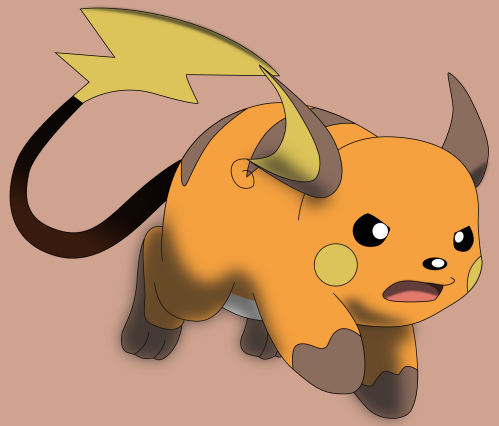
Boosting without State: Rai-Choo



$$pk_i = g^{sk_i}, \qquad \prod_i pk_i = pk$$

Summary

Summary and Open Problems



Source: https://tinyurl.com/bdy34fzc

Rai-Choo

CDH Assumption + Pairings

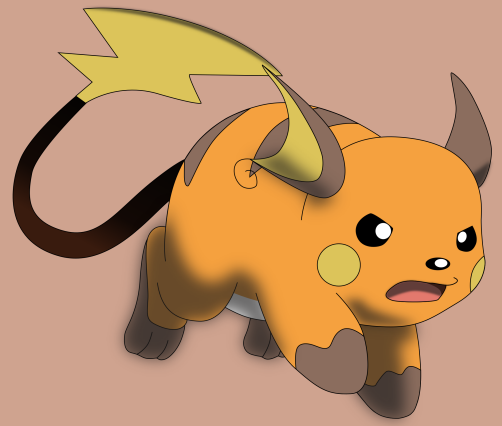
2 Rounds

Stateless

Communication ~ const

Computation ~ const

Summary and Open Problems



Source: https://tinyurl.com/bdy34fzc

Rai-Choo

CDH Assumption + Pairings

2 Rounds

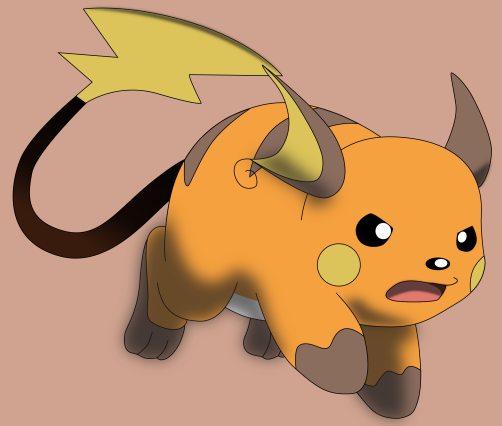
Stateless

Communication ~ const

Computation ~ const

Signature	Communication	Batched	Signing	Verification
9 KB	36 KB	14 KB	169 ms	36 ms
6 KB	72 KB	34 KB	333 ms	22 ms

Summary and Open Problems



Source: https://tinyurl.com/bdy34fzc

Rai-Choo

CDH Assumption + Pairings

2 Rounds

Stateless

Communication ~ const

Computation ~ const

Signature	Communication	Batched	Signing	Verification
9 KB	36 KB	14 KB	169 ms	36 ms
6 KB	72 KB	34 KB	333 ms	22 ms

Pairing-Free Construction?