

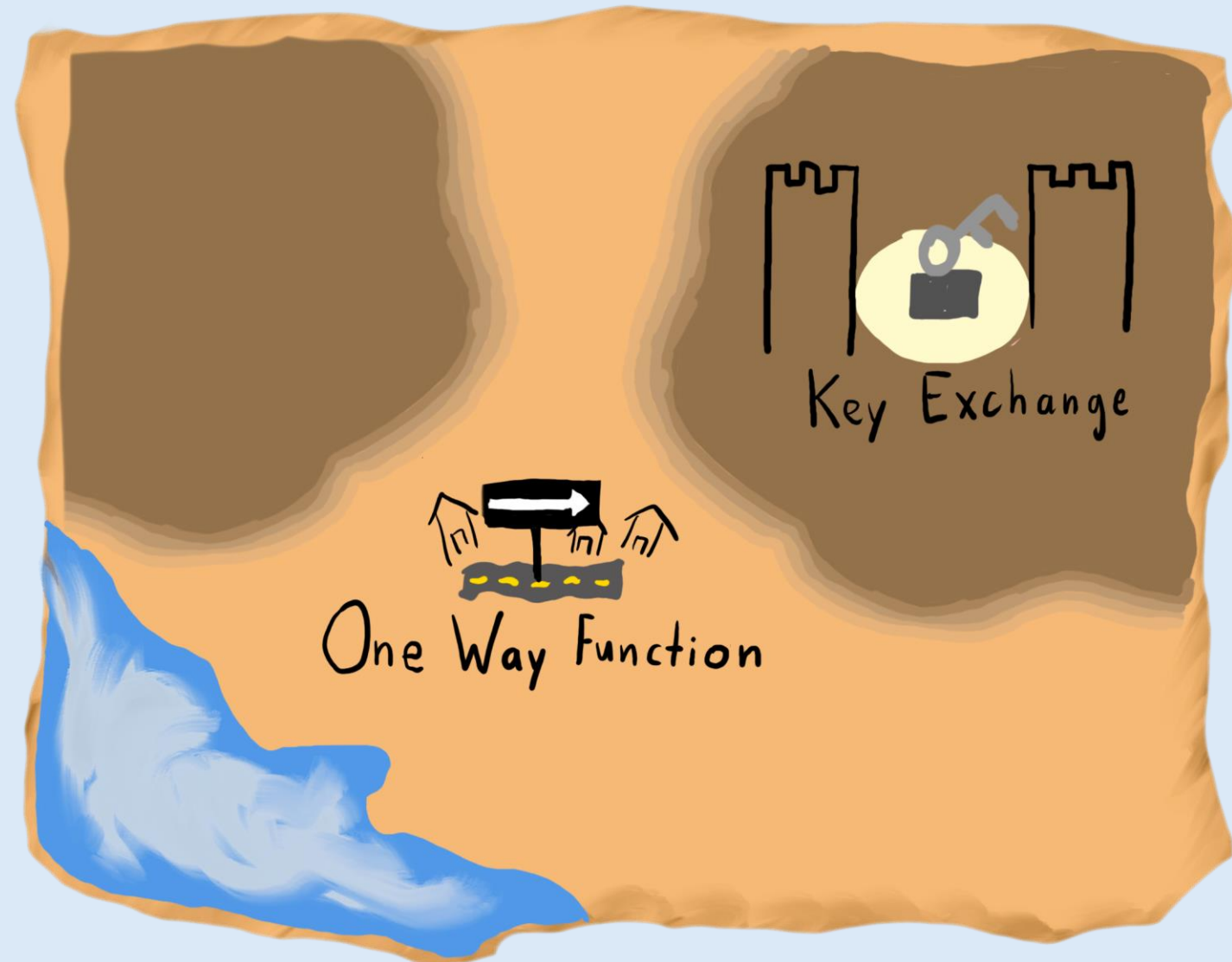
# On Central Primitives for Quantum Cryptography with Classical Communication

Kai-Min Chung, **Eli Goldin**, Matthew Gray

# (Classical) Cryptography Topography

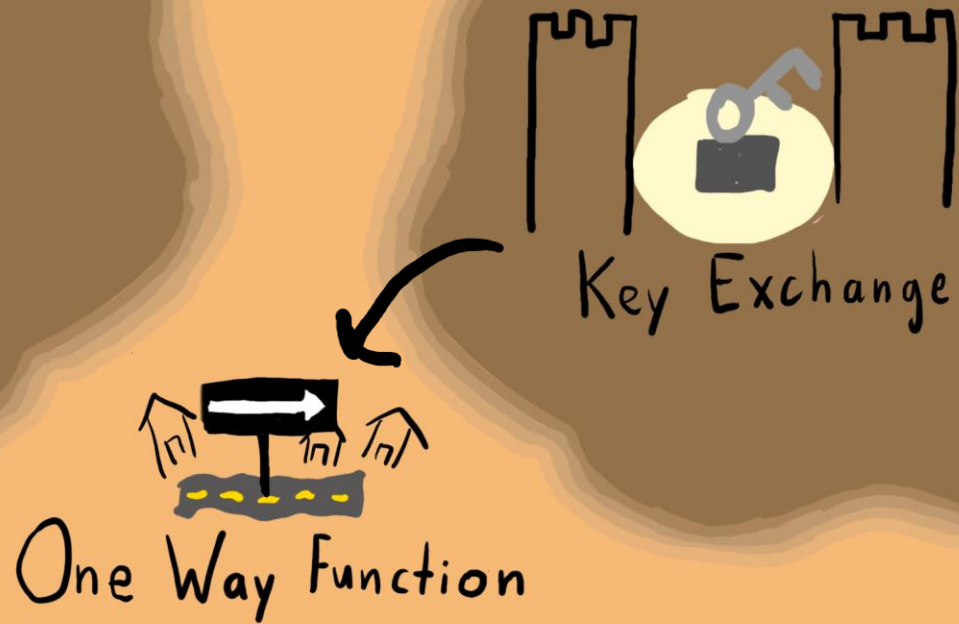


# (Classical) Cryptography Topography

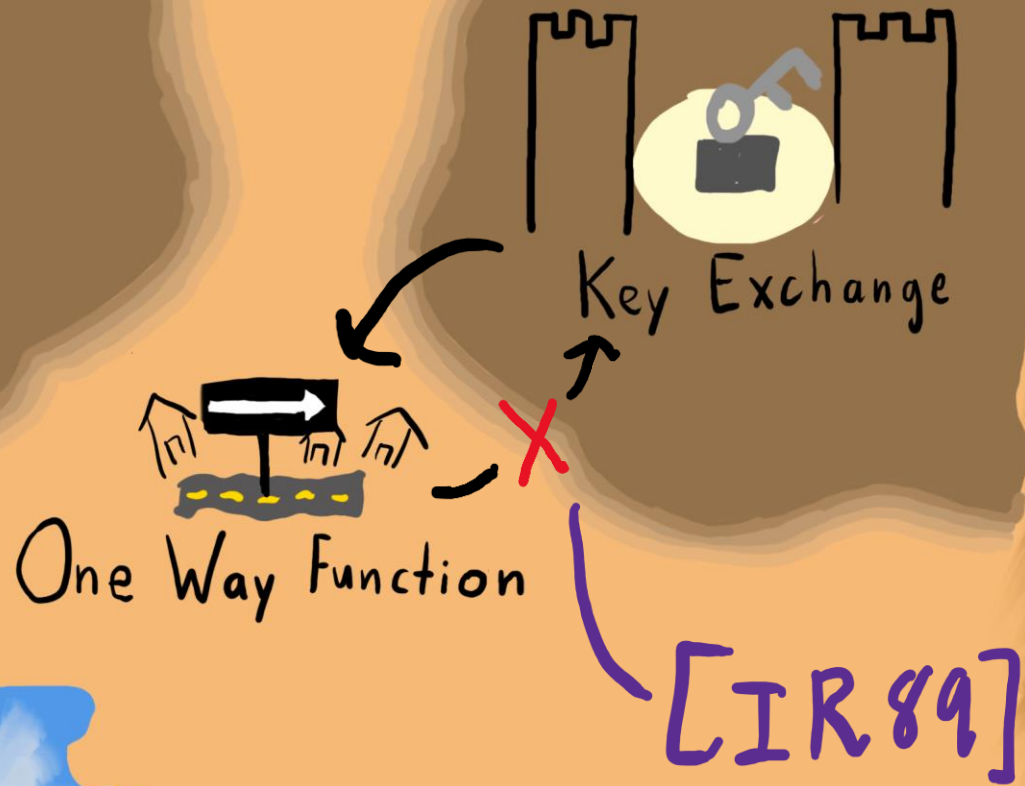


# (Classical) Cryptography Topography

Key Exchange can be used to build one way functions



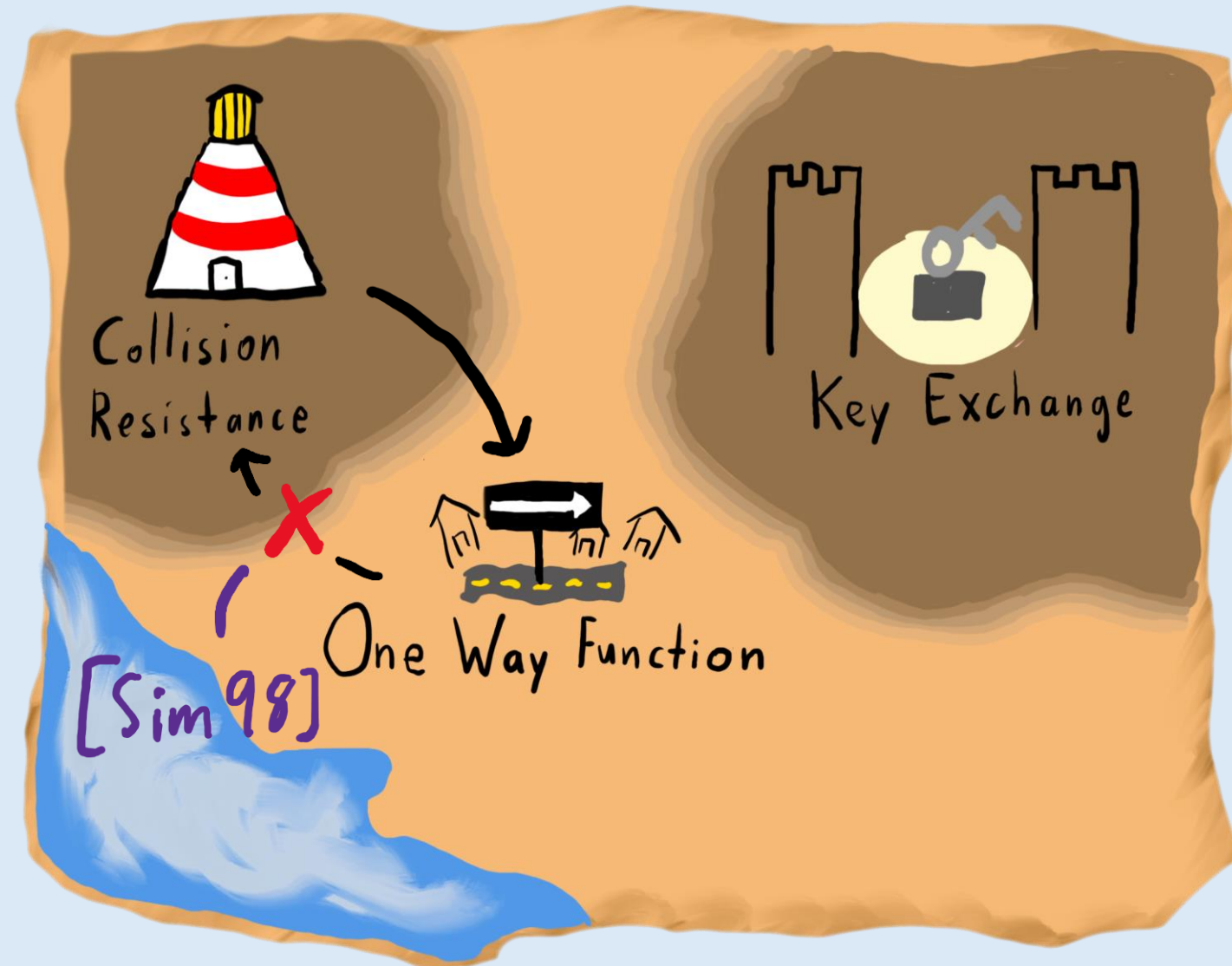
# (Classical) Cryptography Topography



Key Exchange can be used to build one way functions

Hard to build KE from OWF

# (Classical) Cryptography Topography



# (Classical) Cryptography Topography



Collision  
Resistance



Key Exchange



One Way Function



Pseudorandom  
Generator



# (Classical) Cryptography Topography



Collision  
Resistance



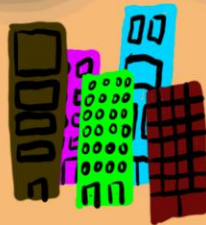
Key Exchange



One Way Function



Secret  
Sharing



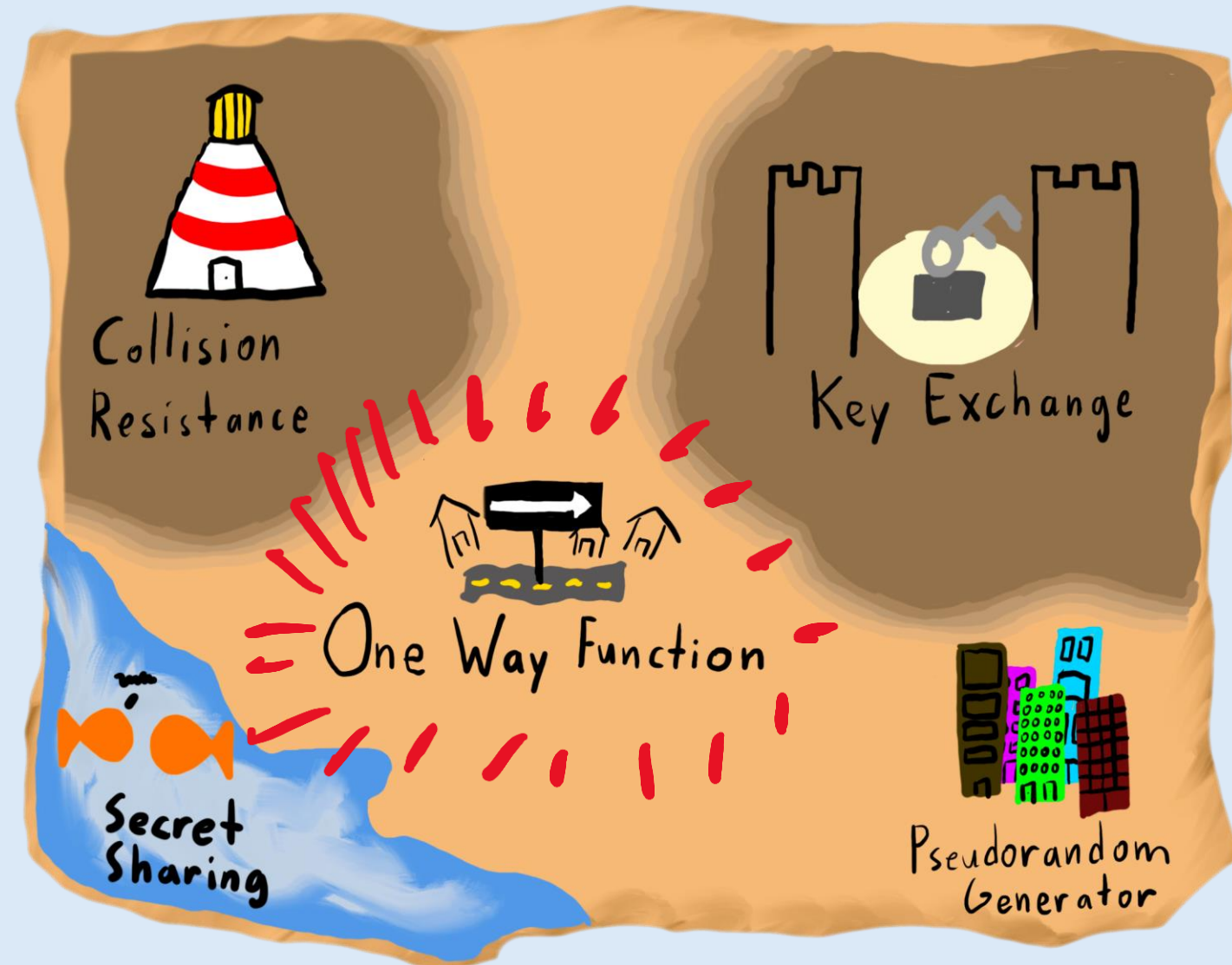
Pseudorandom  
Generator

Secret sharing exists unconditionally (and has information theoretic security)



# The Center of the World: OWFs

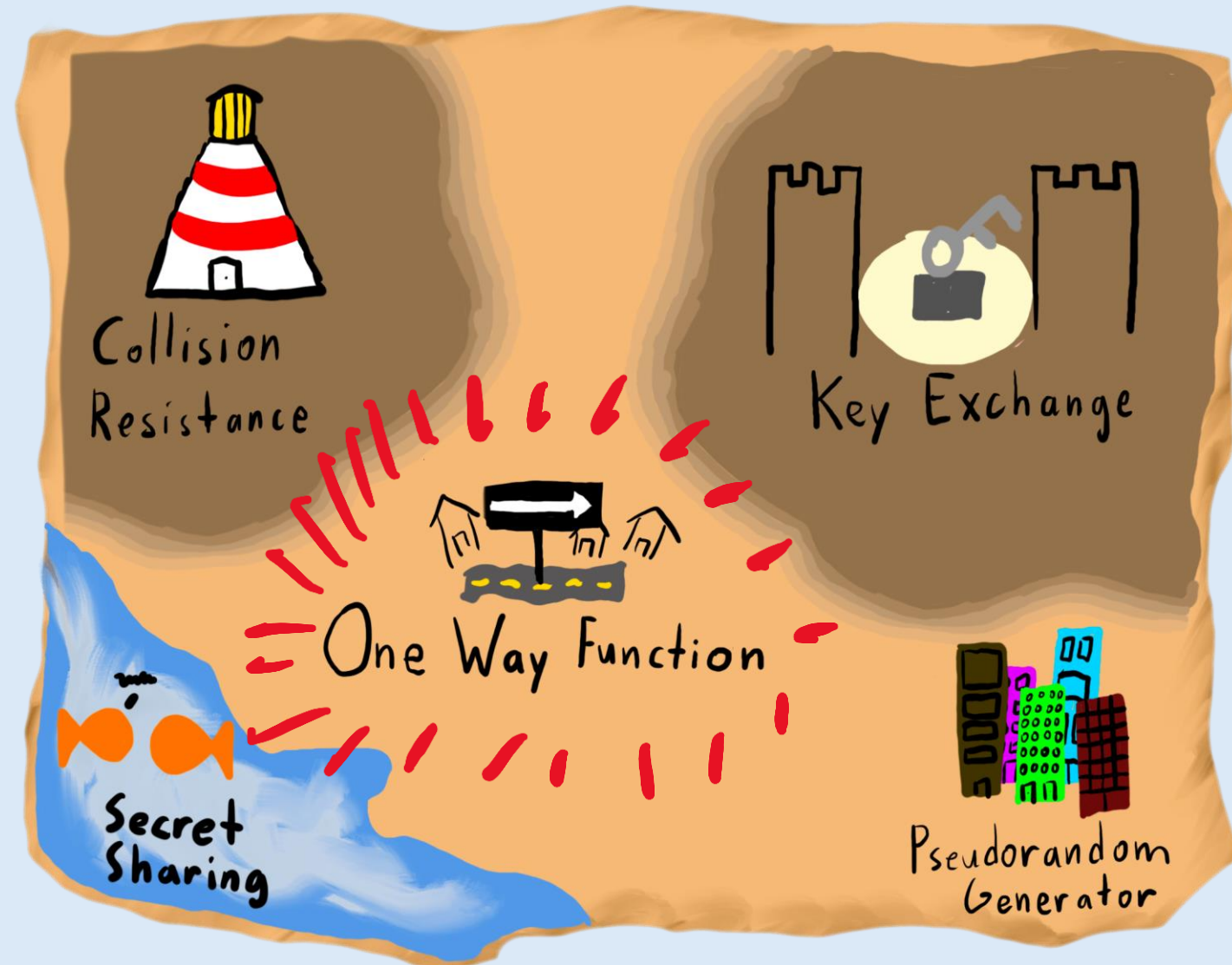
OWFs are special. Very useful starting place for understanding the map of Cryptography.



# The Center of the World: OWFs

OWFs are special. Very useful starting place for understanding the map of Cryptography.

Why?



# The Center of the World: OWFs

Why are one way functions so important?

1. Simple
2. Minimal
3. Useful
4. Flexible

# The Center of the World: OWFs

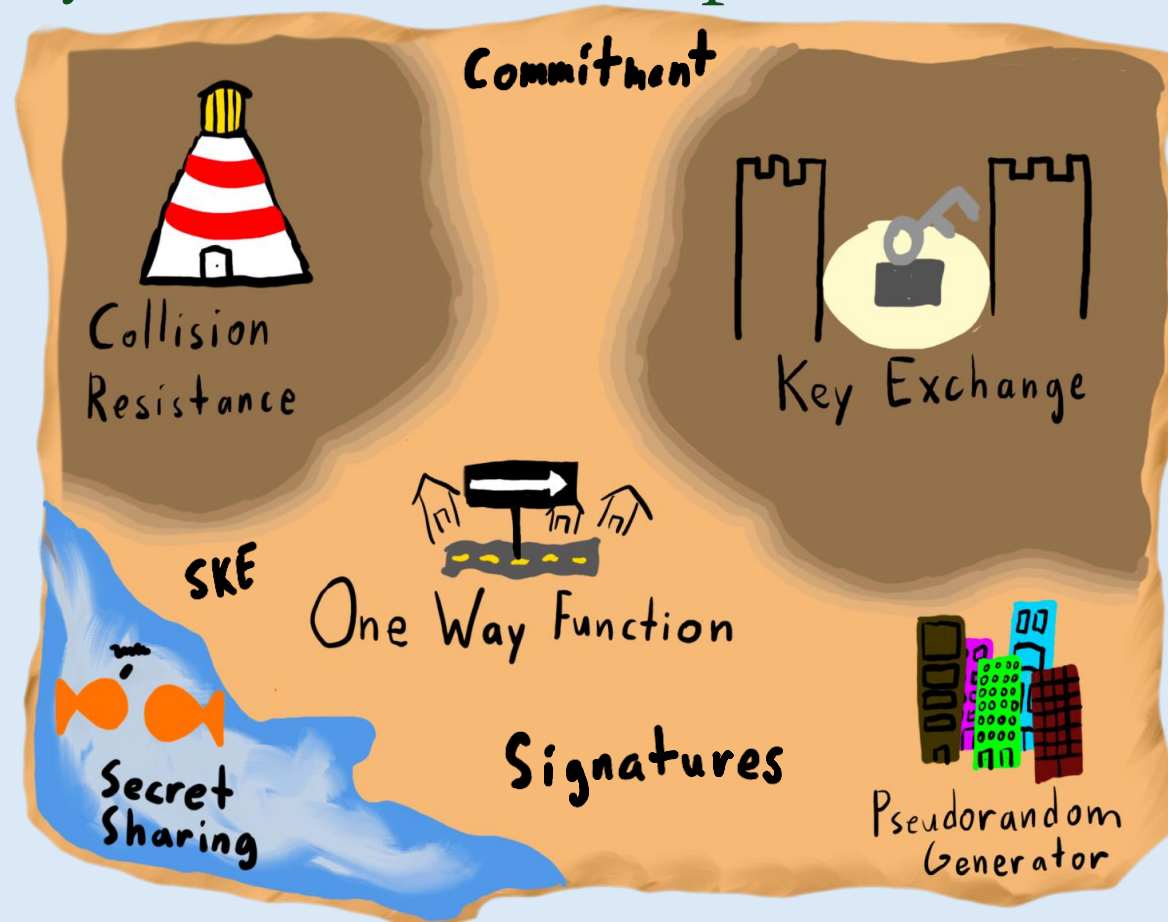
Why are one way functions so important?

1. **Simple:** “Easy to compute, hard to invert”
2. Minimal
3. Useful
4. Flexible

# The Center of the World: OWFs

Why are one way functions so important?

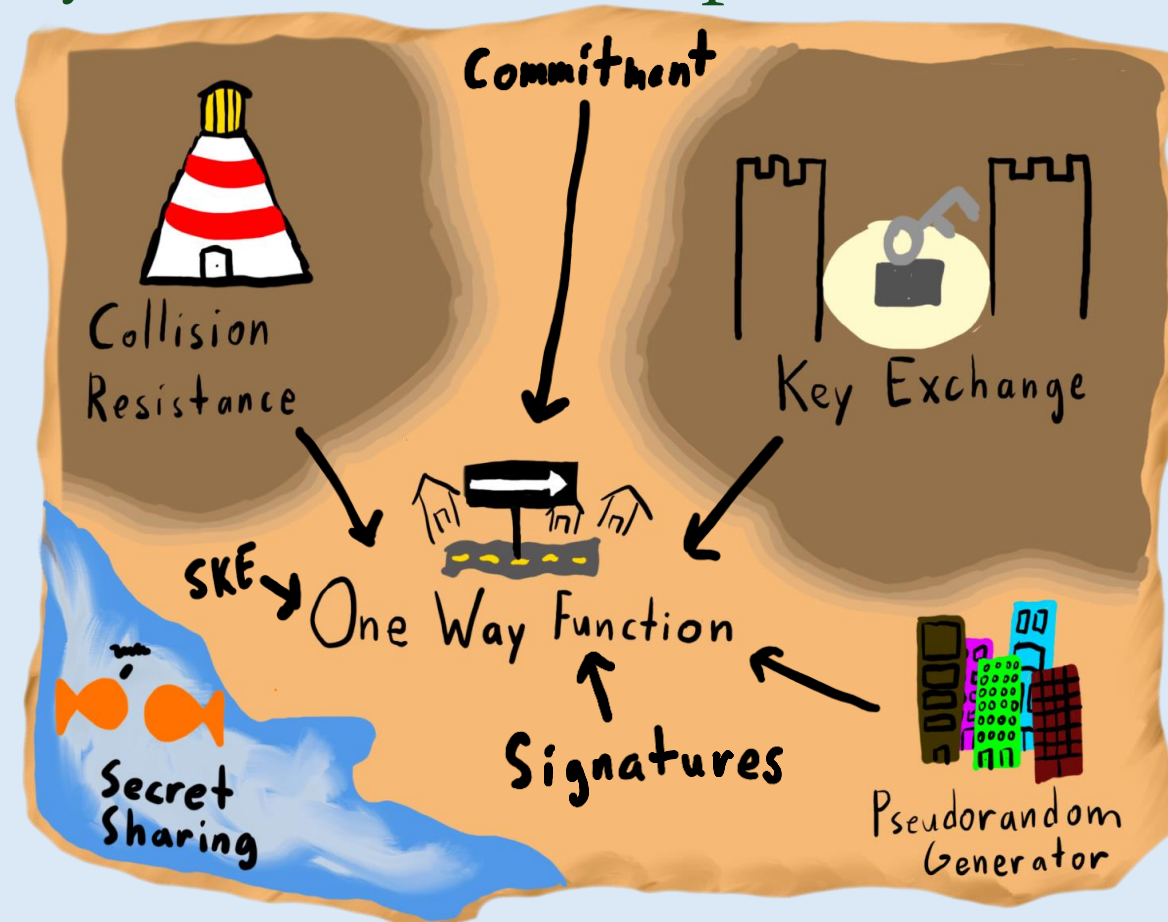
1. Simple
2. Minimal
3. Useful
4. Flexible



# The Center of the World: OWFs

Why are one way functions so important?

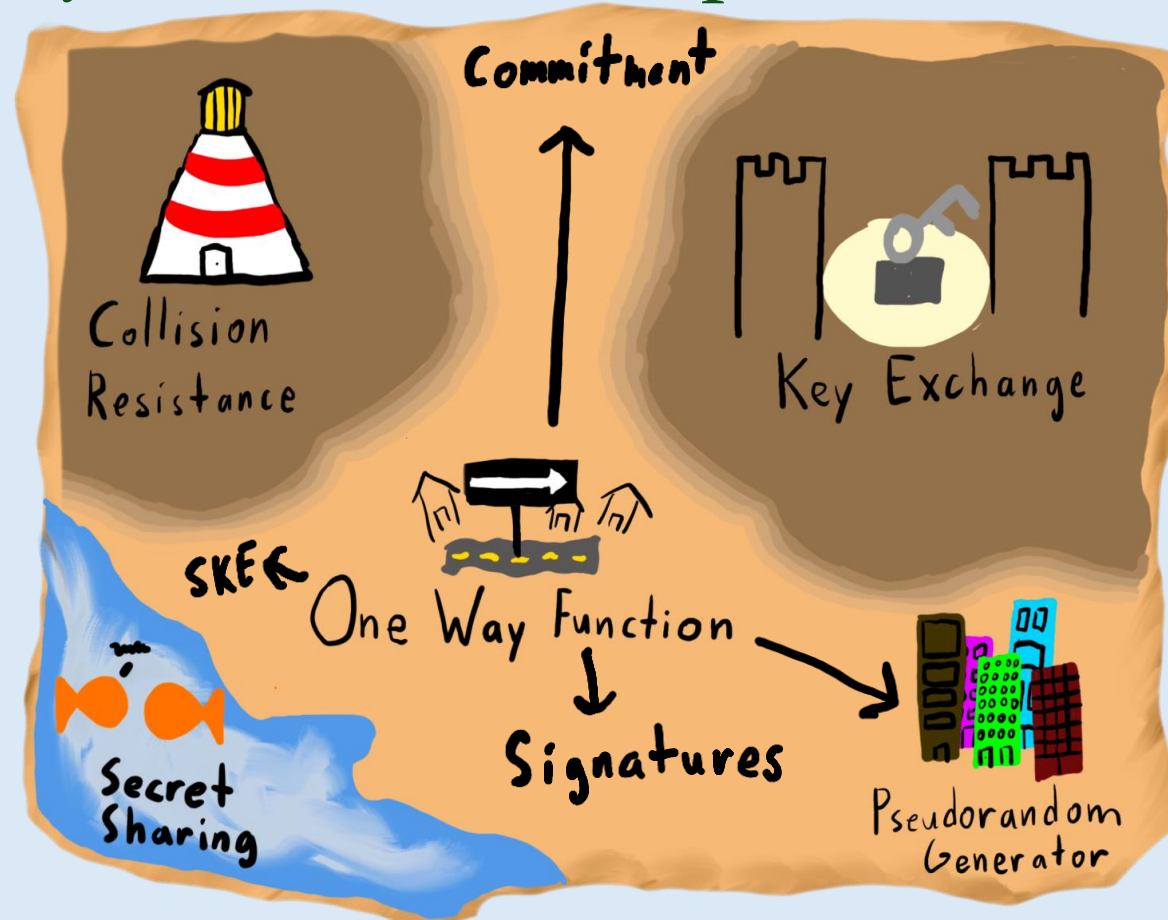
1. Simple
2. **Minimal**
3. Useful
4. Flexible



# The Center of the World: OWFs

Why are one way functions so important?

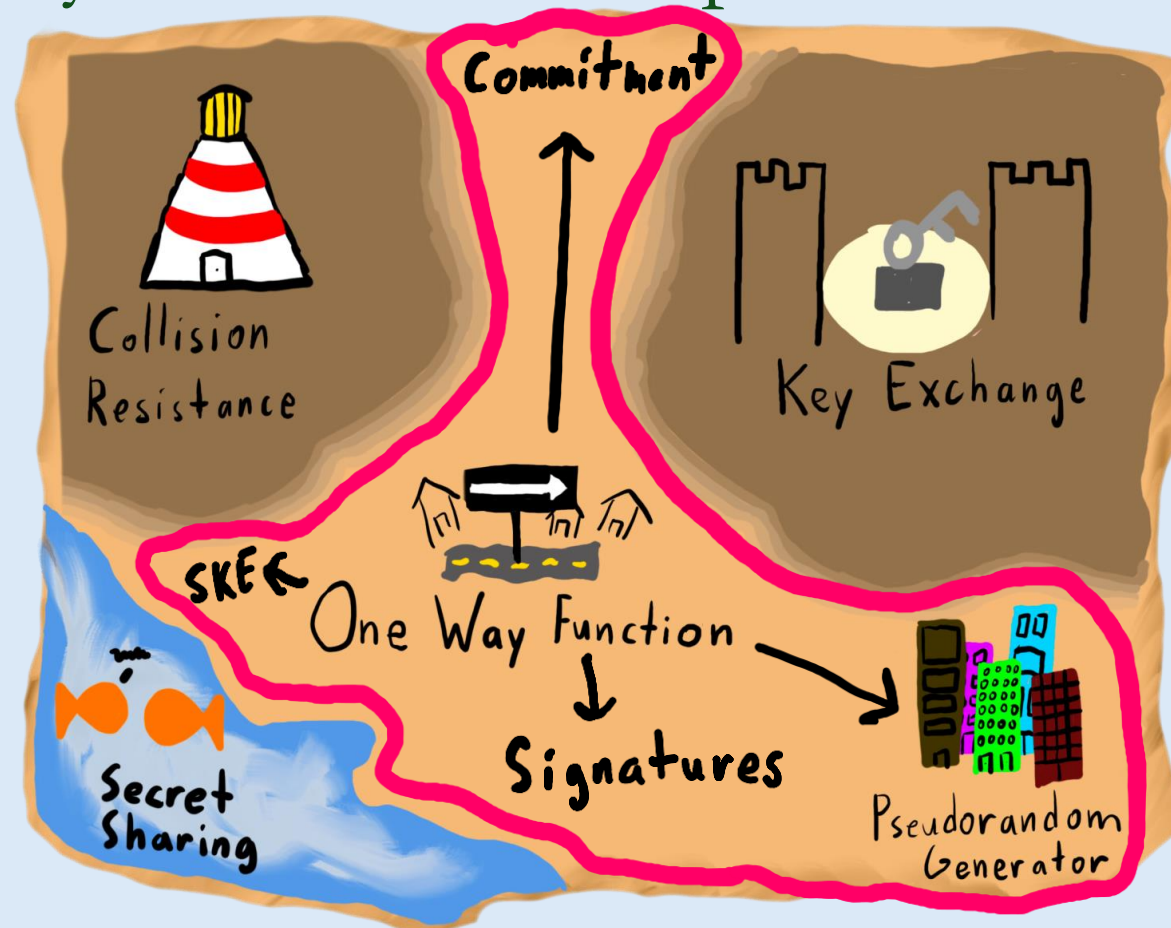
1. Simple
2. Minimal
3. Useful
4. Flexible



# The Center of the World: OWFs

Why are one way functions so important?

1. Simple
2. Minimal
3. Useful
4. Flexible



MiniCrypt



# The Center of the World: OWFs

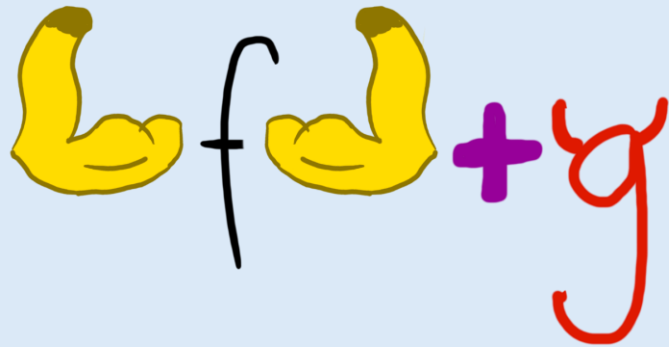
Why are one way functions so important?

1. Simple
2. Minimal
3. Useful
4. **Flexible**

# The Center of the World: OWFs

Why are one way functions so important?

1. Simple
2. Minimal
3. Useful
4. **Flexible**

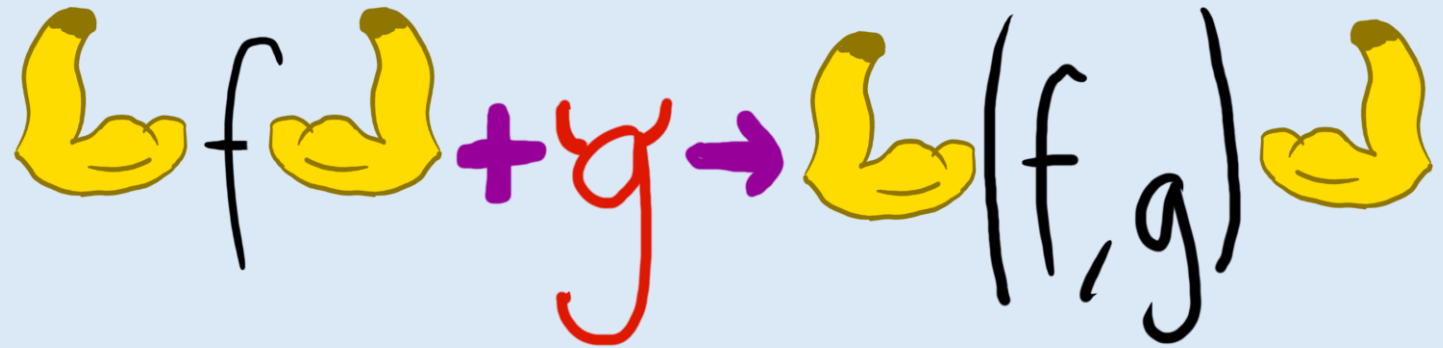


Combiners

# The Center of the World: OWFs

Why are one way functions so important?

1. Simple
2. Minimal
3. Useful
4. **Flexible**

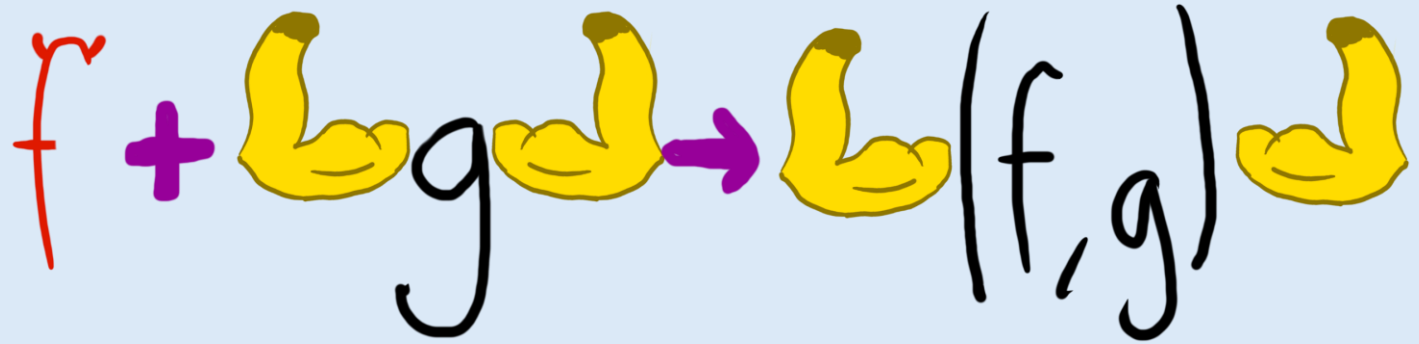


Combiners

# The Center of the World: OWFs

Why are one way functions so important?

1. Simple
2. Minimal
3. Useful
4. **Flexible**

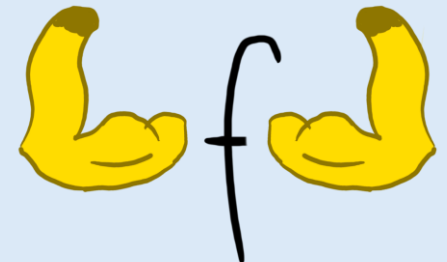


Combiners

# The Center of the World: OWFs

Why are one way functions so important?

1. Simple
2. Minimal
3. Useful
4. **Flexible**

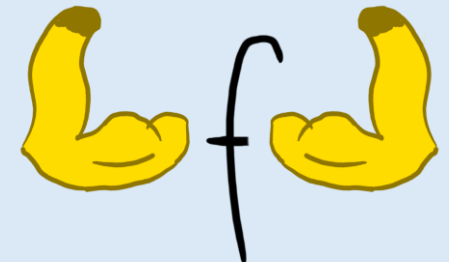


# The Center of the World: OWFs

"strong" OWF

Why are one way functions so important?

1. Simple
2. Minimal
3. Useful
4. **Flexible**



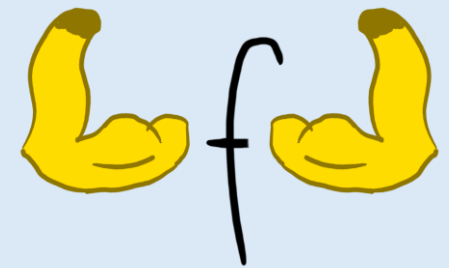
# The Center of the World: OWFs

Why are one way functions so important?

1. Simple
2. Minimal
3. Useful
4. **Flexible**

"strong" OWF

(



"weak" OWF

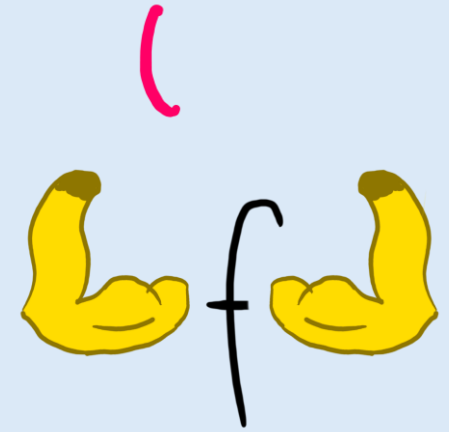


# The Center of the World: OWFs

Why are one way functions so important?

1. Simple
2. Minimal
3. Useful
4. **Flexible**

"strong" OWF



"weak" OWF



"distributional" OWF

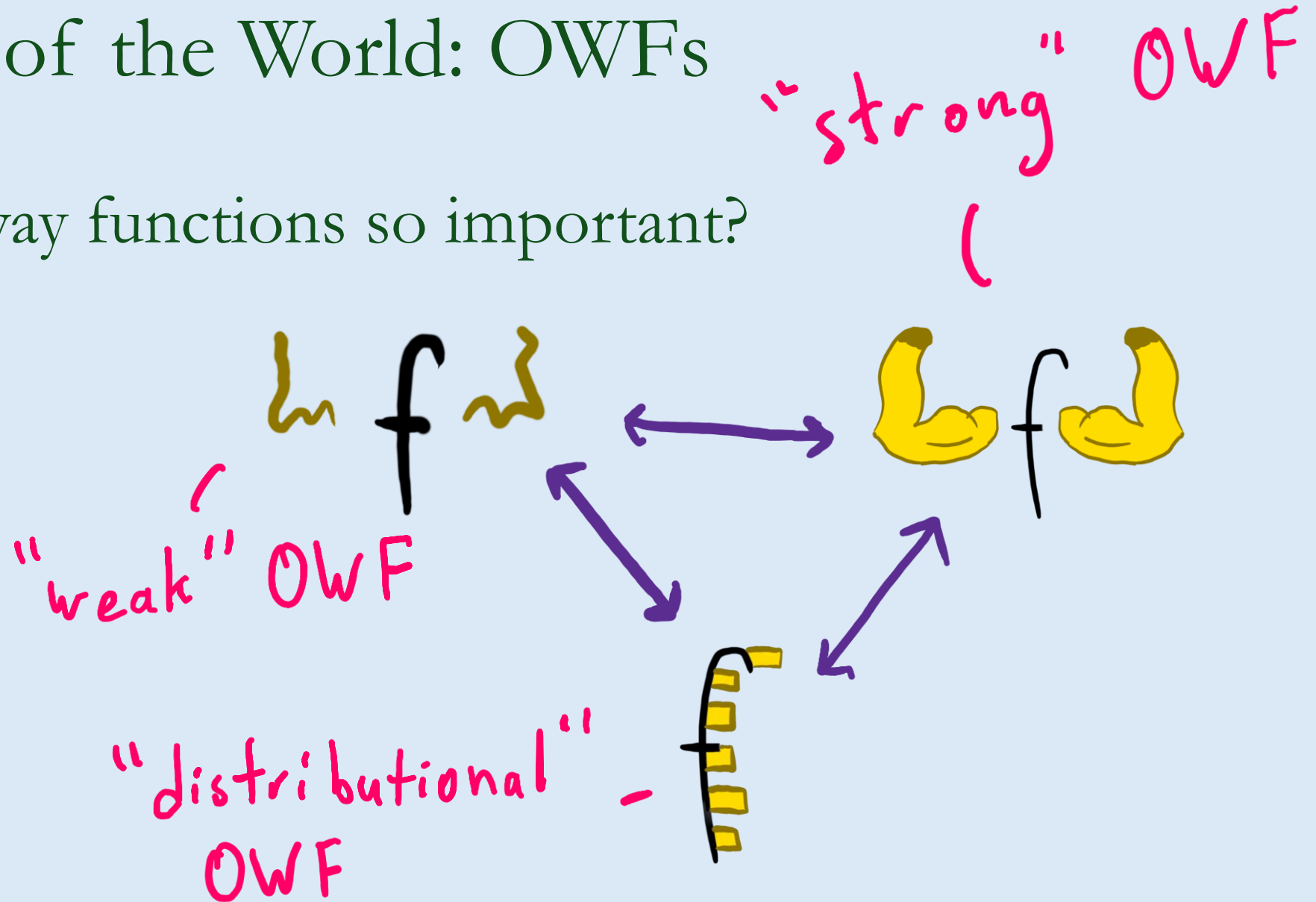




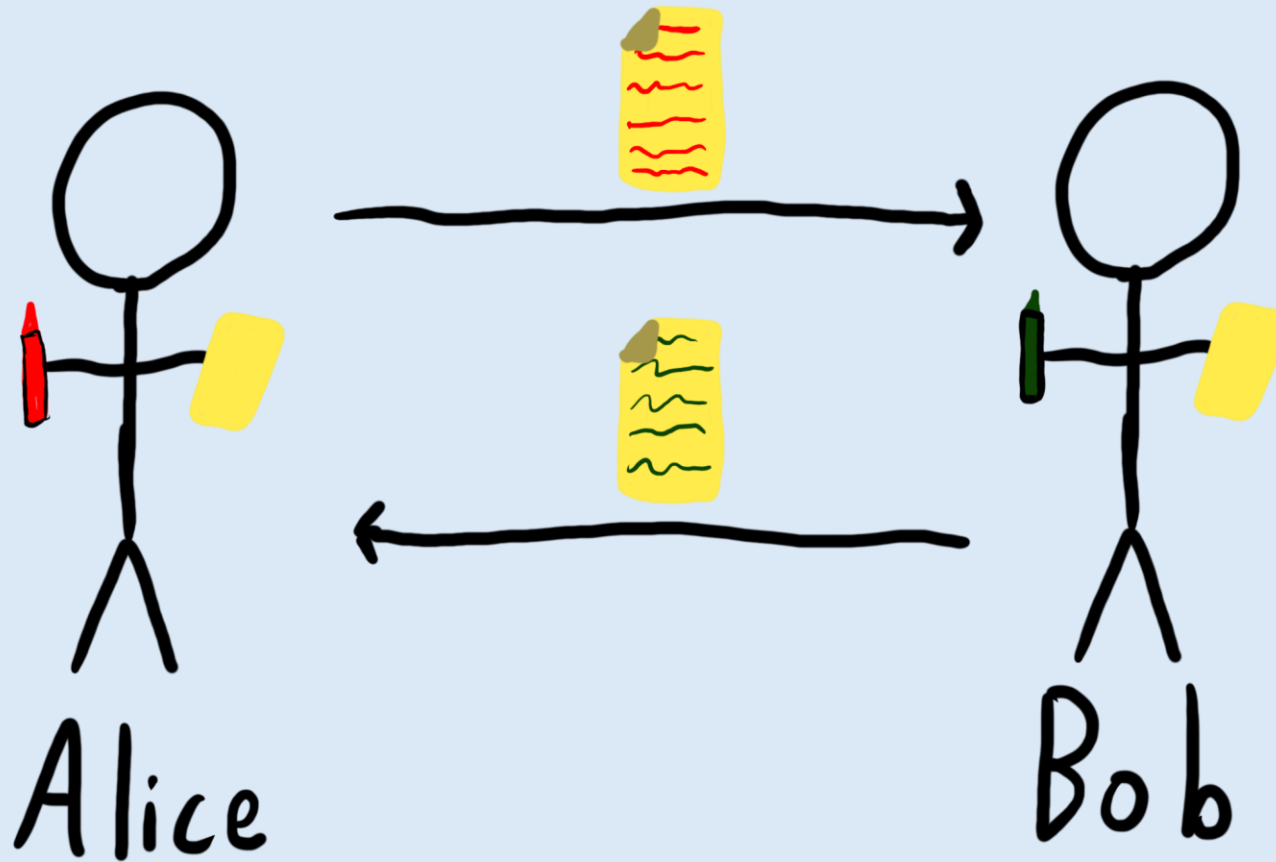
# The Center of the World: OWFs

Why are one way functions so important?

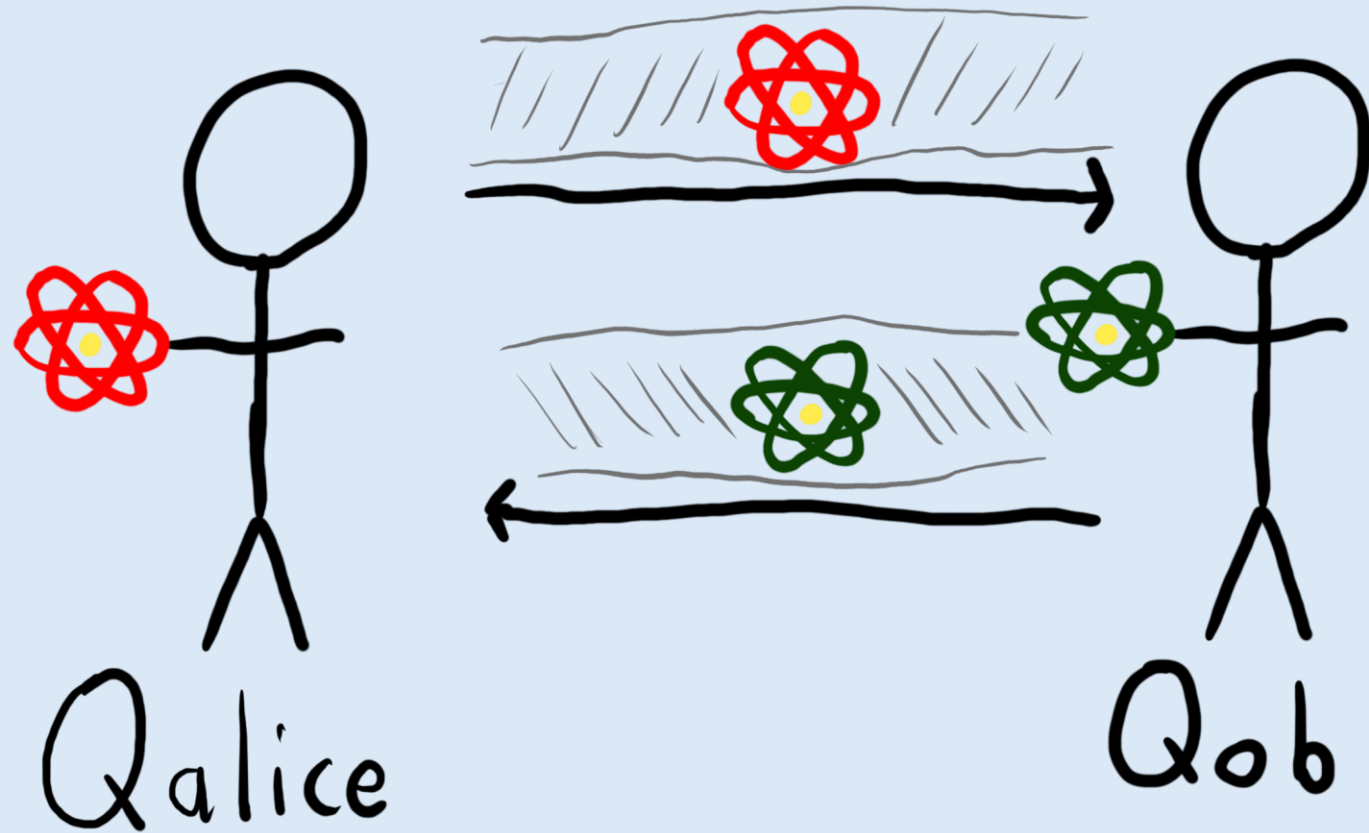
1. Simple
2. Minimal
3. Useful
4. **Flexible**



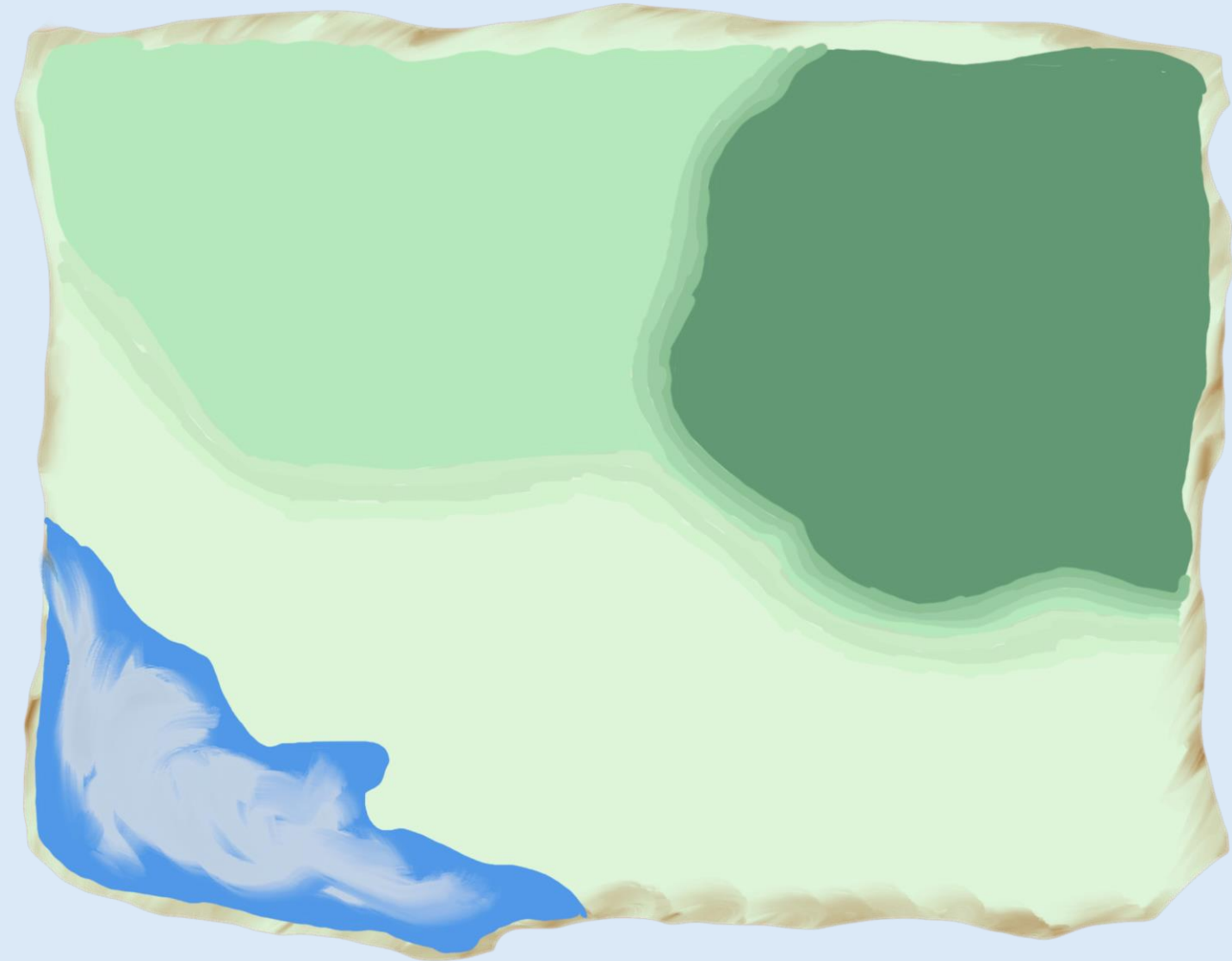
# What about quantum communication?



# What about quantum communication?

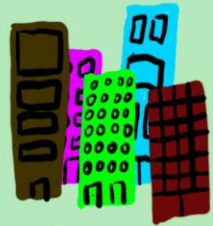


# Quantum Cryptography Topography



# Quantum Cryptography Topography

Map different!



Pseudorandom  
State Generator



Post-Quantum  
One Way Function



Quantum  
Key Distribution

# Quantum Cryptography Topography



A hand-drawn topographic map on a piece of paper with a brown border. The map features three distinct regions: a light green area in the upper left, a dark green area in the upper right, and a blue area in the lower left. The light green area contains an icon of four colorful buildings (brown, pink, green, and red) and the text 'Pseudorandom State Generator'. The dark green area contains an icon of a road with a white arrow pointing right, flanked by three houses, and the text 'Post-Quantum One Way Function'. The blue area contains an icon of a castle with a yellow sun in the center and the text 'Quantum Key Distribution'. The background of the map is a light yellow-green color.

Pseudorandom  
State Generator

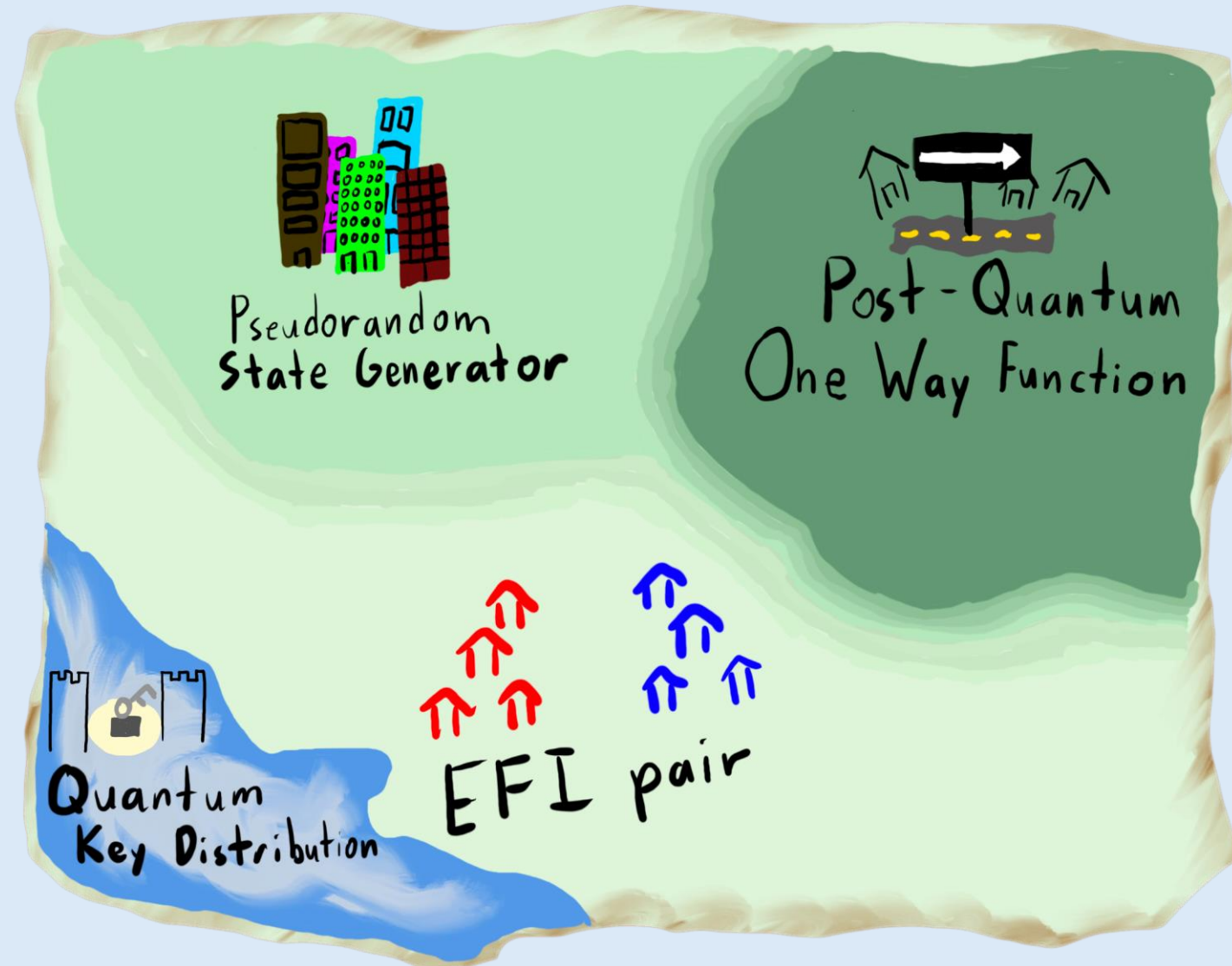
Post-Quantum  
One Way Function

Quantum  
Key Distribution

Map different!

Is there a central  
primitive?

# Quantum Cryptography Topography

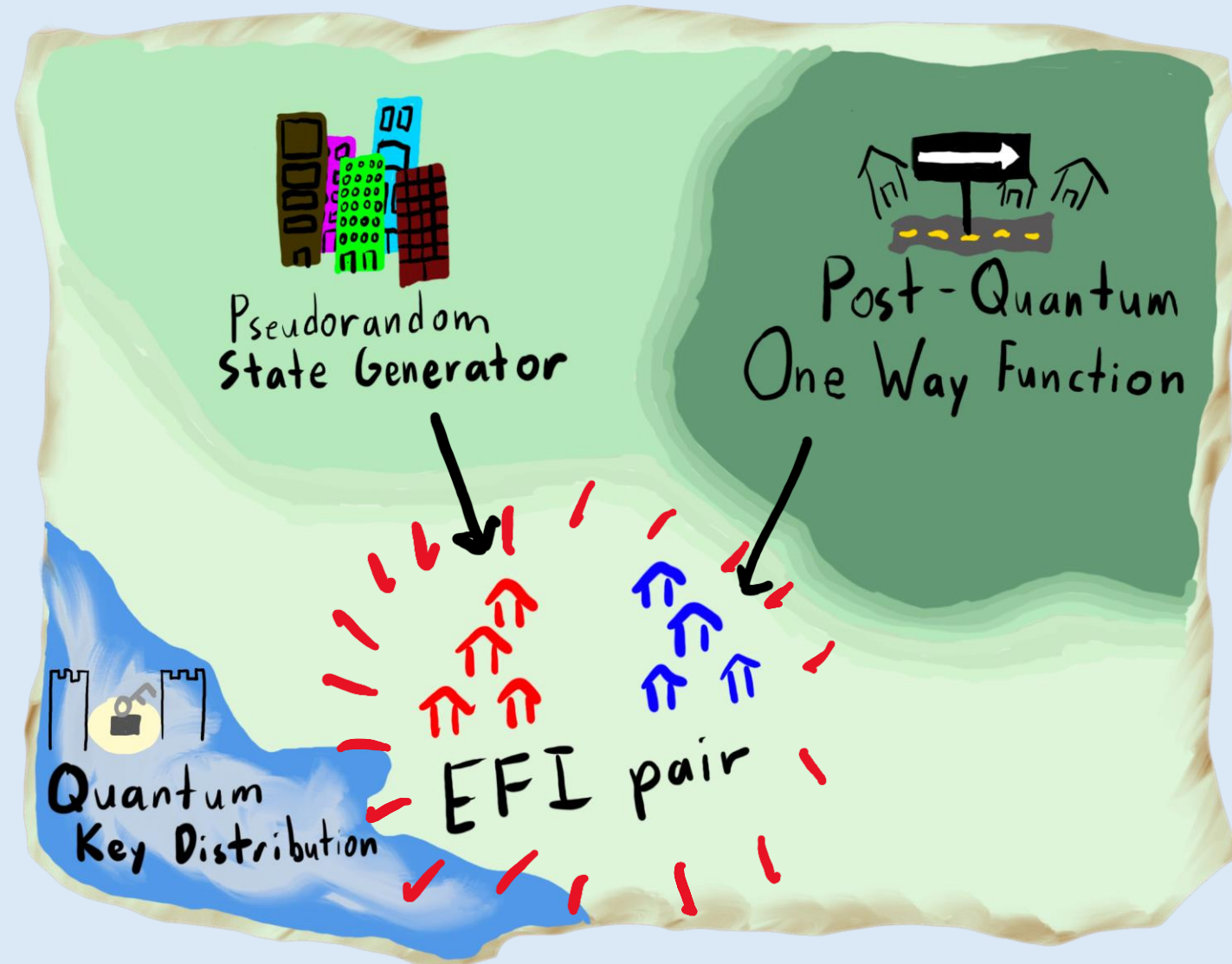


Map different!

Is there a central primitive?

Pretty good candidate:  
EFI pairs

# Quantum Cryptography Topography



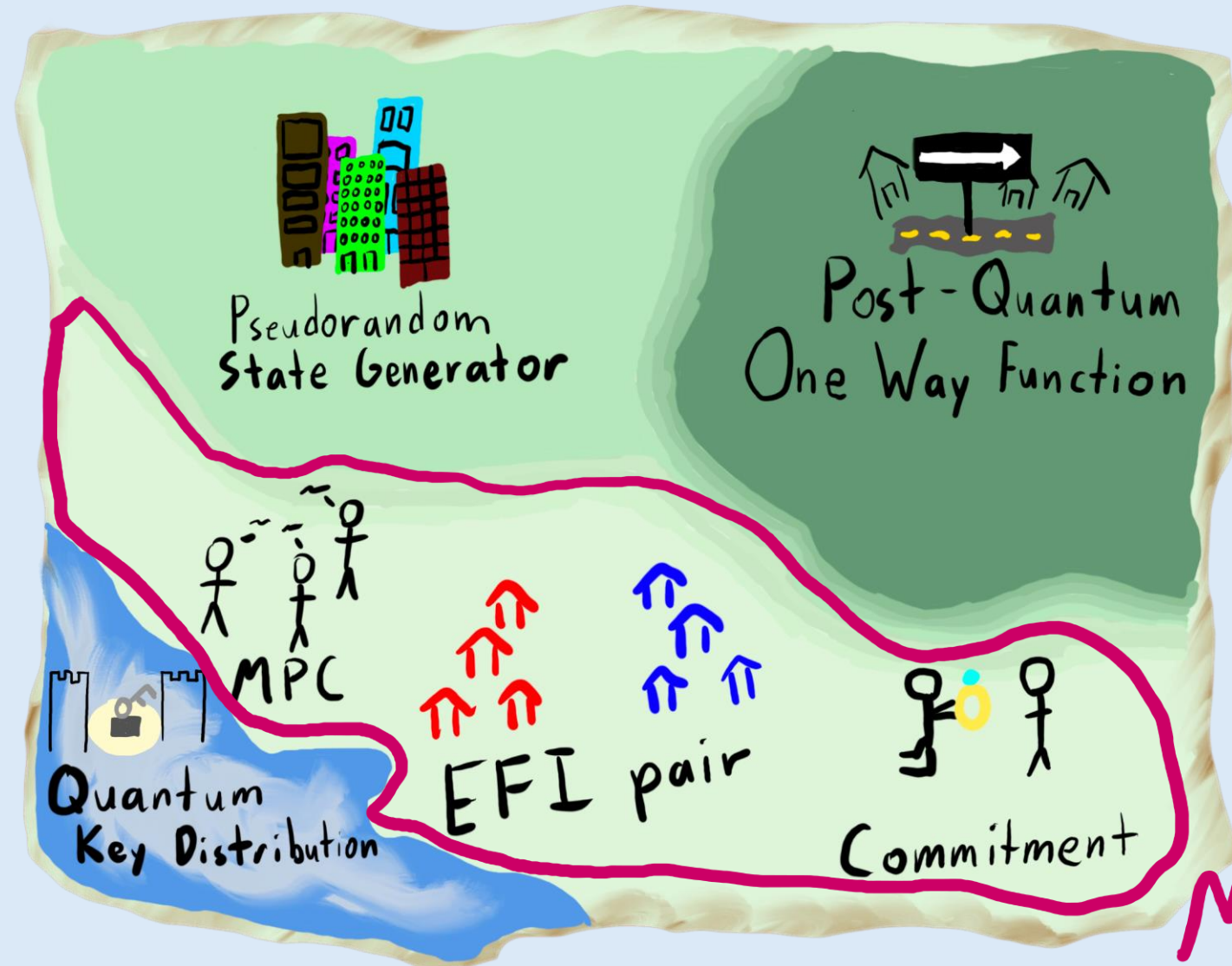
Map different!

Is there a central primitive?

Pretty good candidate:  
EFI pairs



# Quantum Cryptography Topography



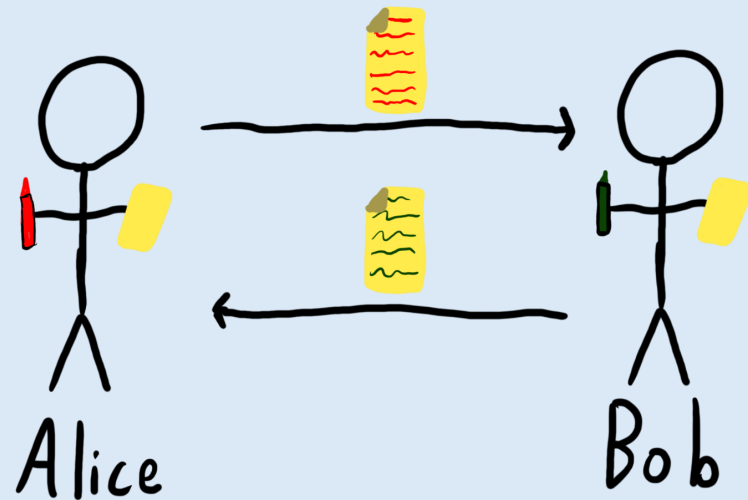
Map different!

Is there a central primitive?

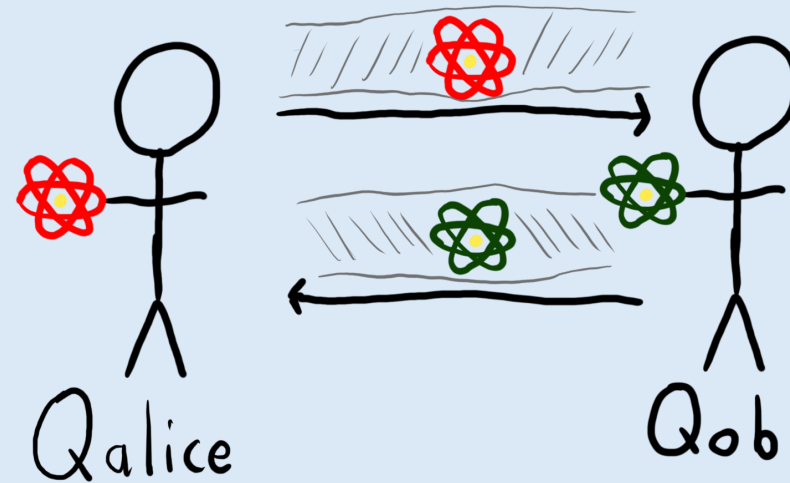
Pretty good candidate:  
EFI pairs

Micro Qrypt

# Our focus: The QCCC (QC) setting



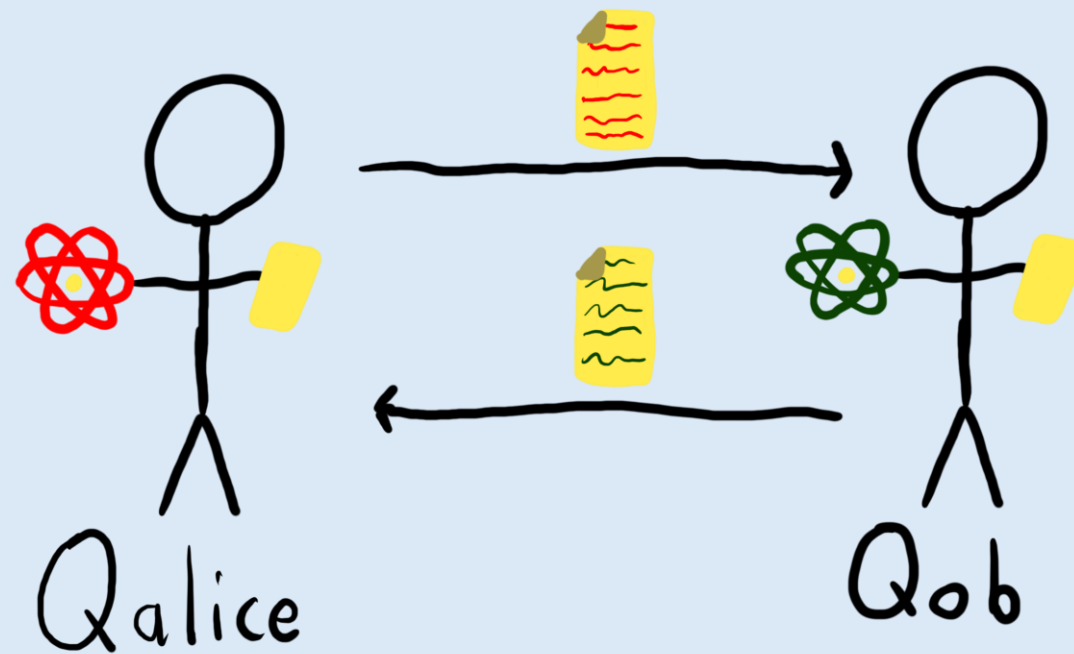
OWF



EFI

Our focus: The QCCC (QC) setting

“Quantum Computation with Classical Communication”



## Our Goals

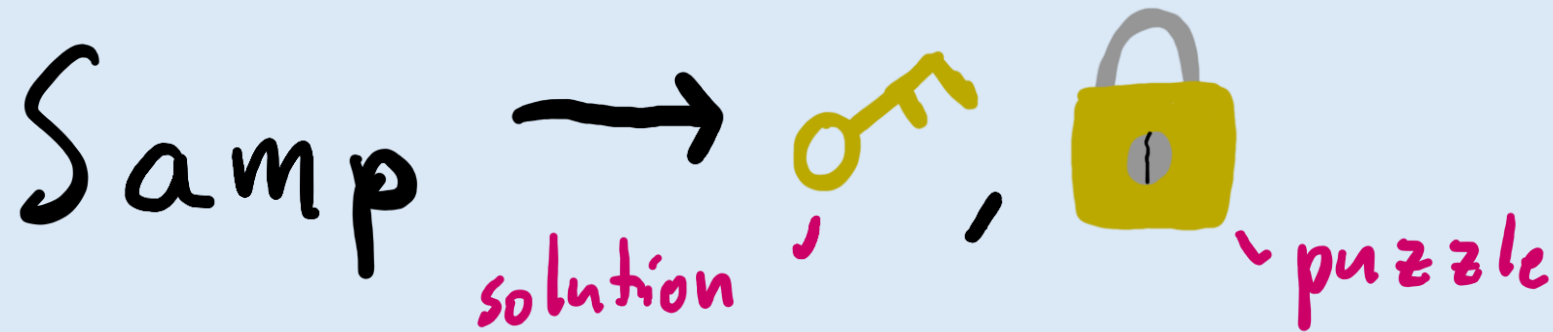
What does the “map” of QC cryptography look like?

Is there a central primitive in the QC setting?

# Prior Work

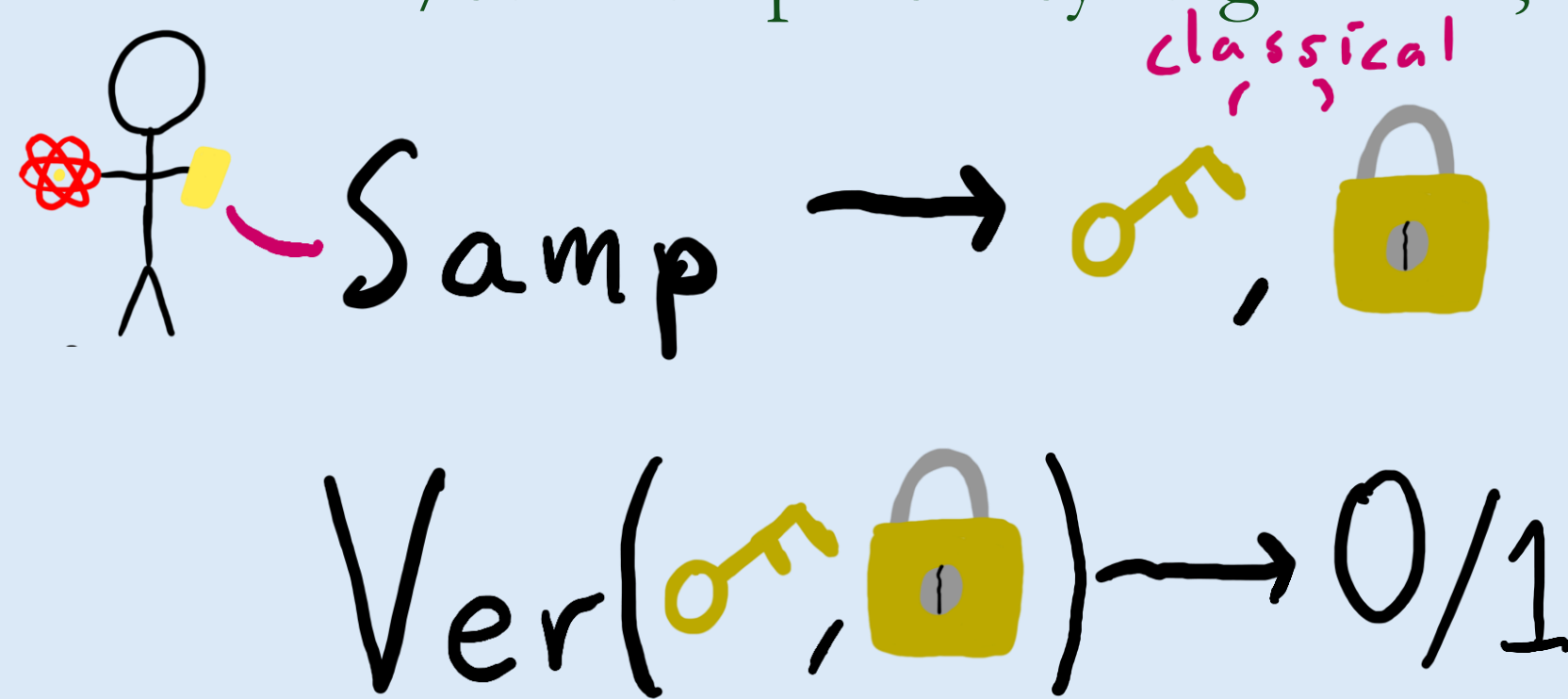
# Candidate: One Way Puzzles [KT24]

Puzzle/solution pairs easy to generate, hard to solve.



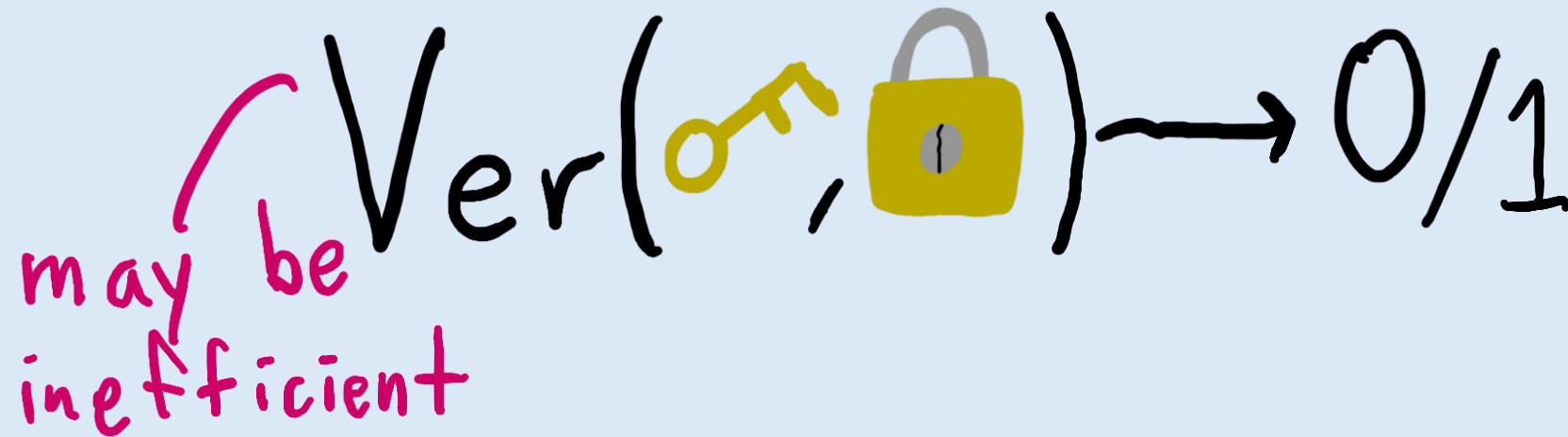
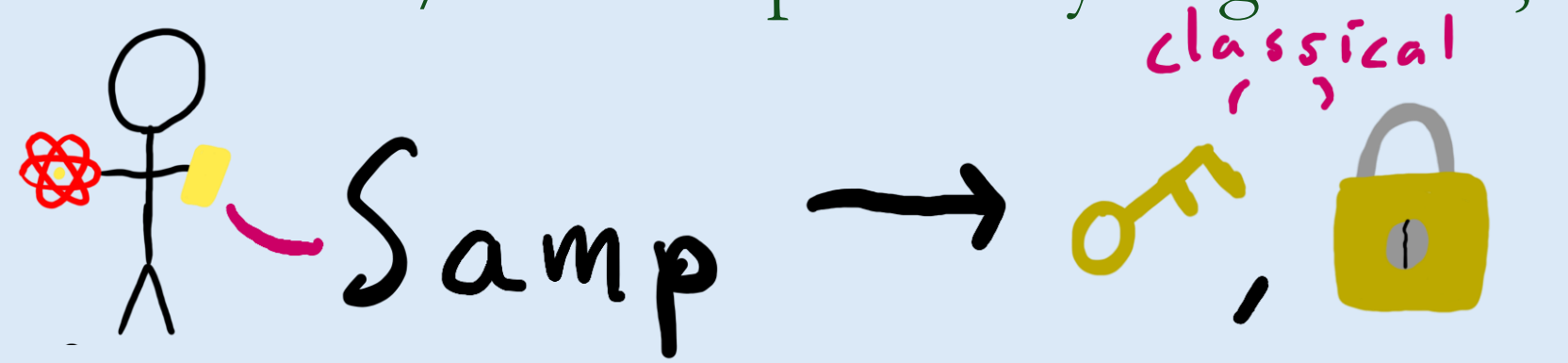
# Candidate: One Way Puzzles [KT24]

Puzzle/solution pairs easy to generate, hard to solve.



# Candidate: One Way Puzzles [KT24]

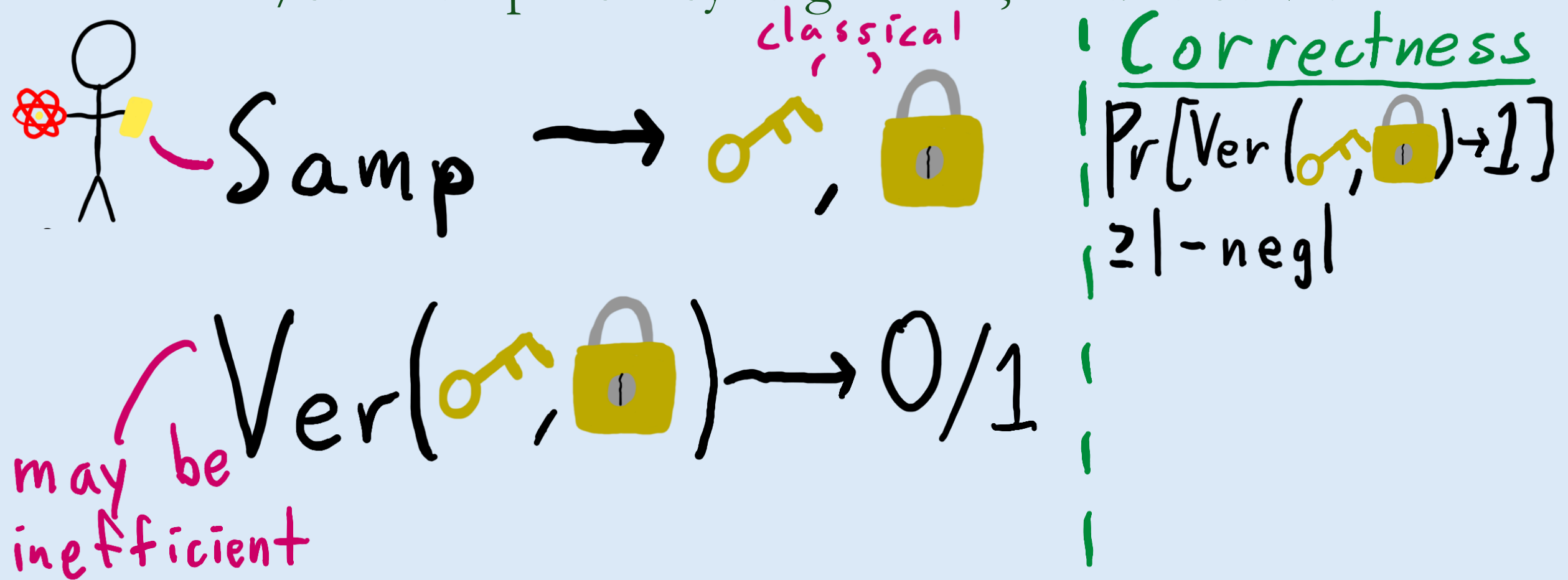
Puzzle/solution pairs easy to generate, hard to solve.





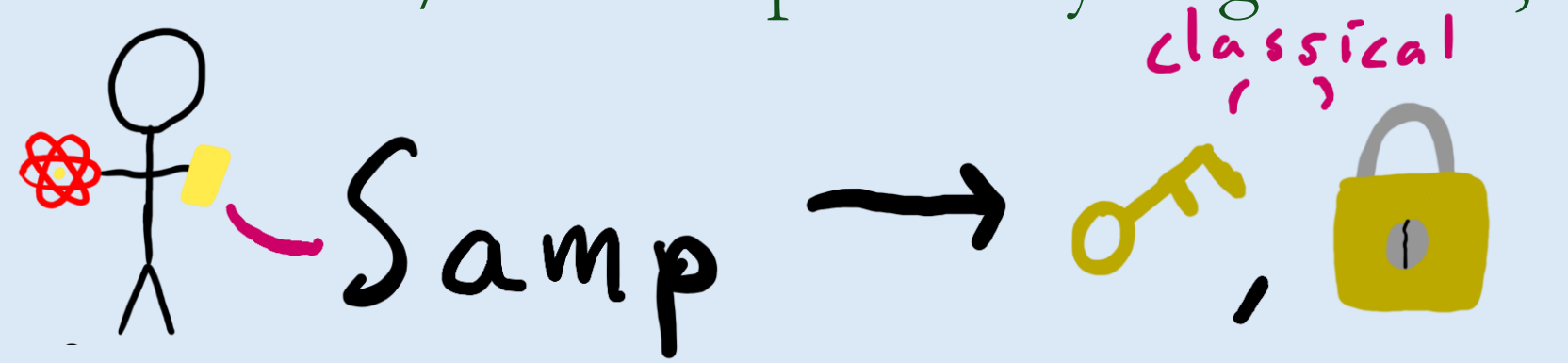
# Candidate: One Way Puzzles [KT24]

Puzzle/solution pairs easy to generate, hard to solve.



# Candidate: One Way Puzzles [KT24]

Puzzle/solution pairs easy to generate, hard to solve.



Correctness

$$\Pr[\text{Ver}(\text{key}, \text{lock}) \rightarrow 1] \geq 1 - \text{negl}$$

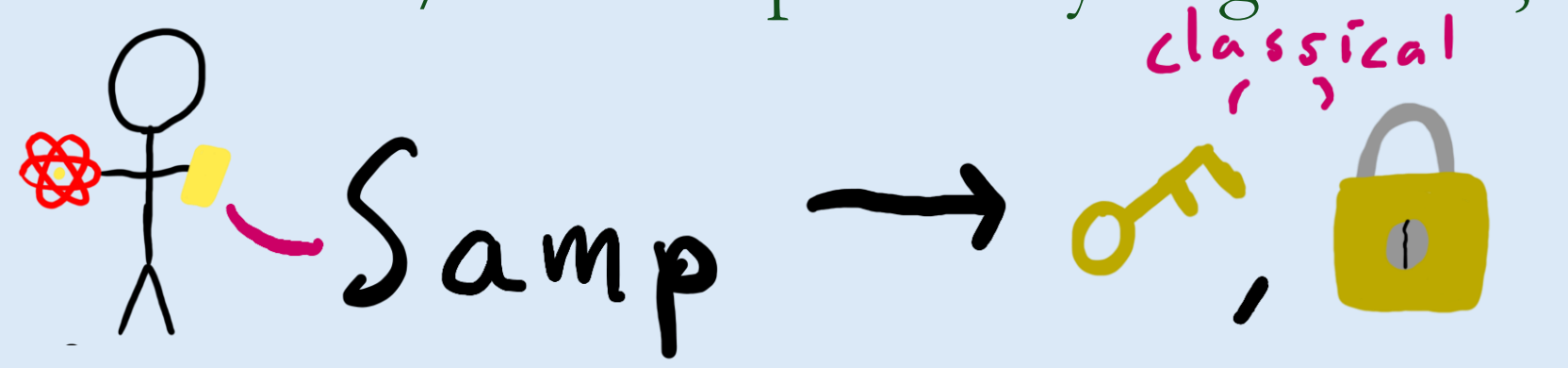
Security

$$A(\text{lock}) \rightarrow \text{key}$$
$$\Pr[\text{Ver}(\text{key}, \text{lock}) \rightarrow 1] \leq \text{negl}$$

# Candidate: One Way Puzzles [KT24]

classically  
 $\approx$  OWF!

Puzzle/solution pairs easy to generate, hard to solve.



Correctness

$$\Pr[\text{Ver}(\text{key}, \text{lock}) \rightarrow 1]$$

$$\geq 1 - \text{negl}$$

Security

$$A(\text{lock}) \rightarrow \text{key}$$

$$\Pr[\text{Ver}(\text{key}, \text{lock}) \rightarrow 1]$$

$$\leq \text{negl}$$

$$\text{Ver}(\text{key}, \text{lock}) \rightarrow 0/1$$

may be inefficient

# Are one way puzzles a good central primitive?

## Important properties of central primitives

1. **Simple** ✓
2. Minimal
3. Useful
4. Flexible

# Are one way puzzles a good central primitive?

## Important properties of central primitives

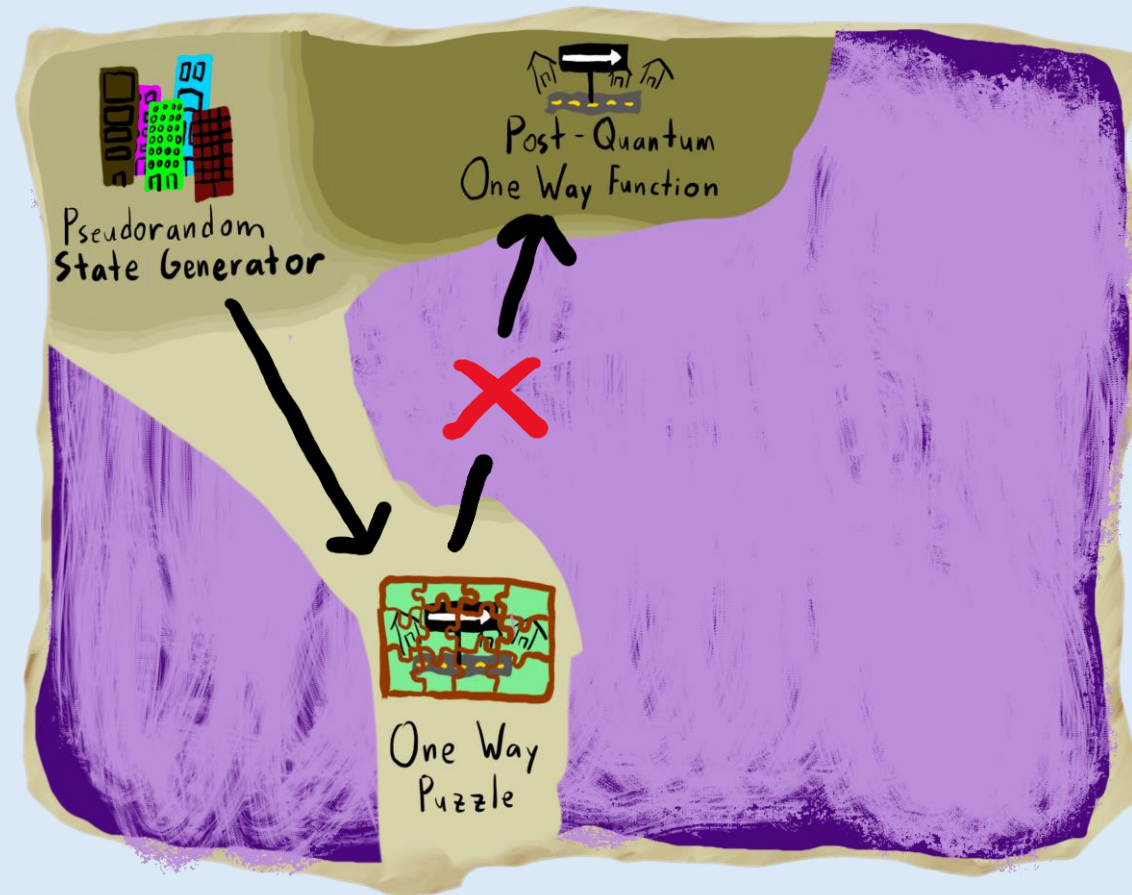
1. Simple ✓
2. **Minimal**
3. Useful
4. Flexible



# Are one way puzzles a good central primitive?

## Important properties of central primitives

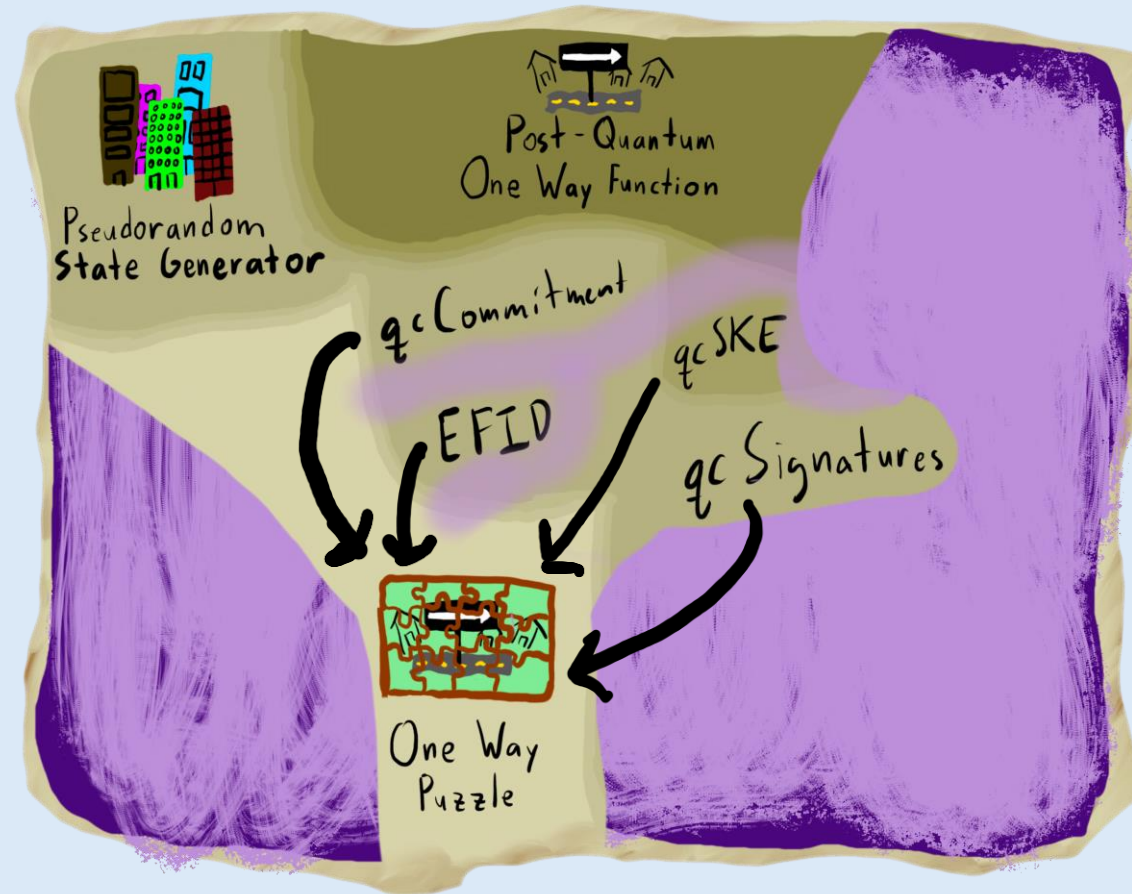
1. Simple ✓
2. Minimal
3. Useful
4. Flexible



# Are one way puzzles a good central primitive?

## Important properties of central primitives

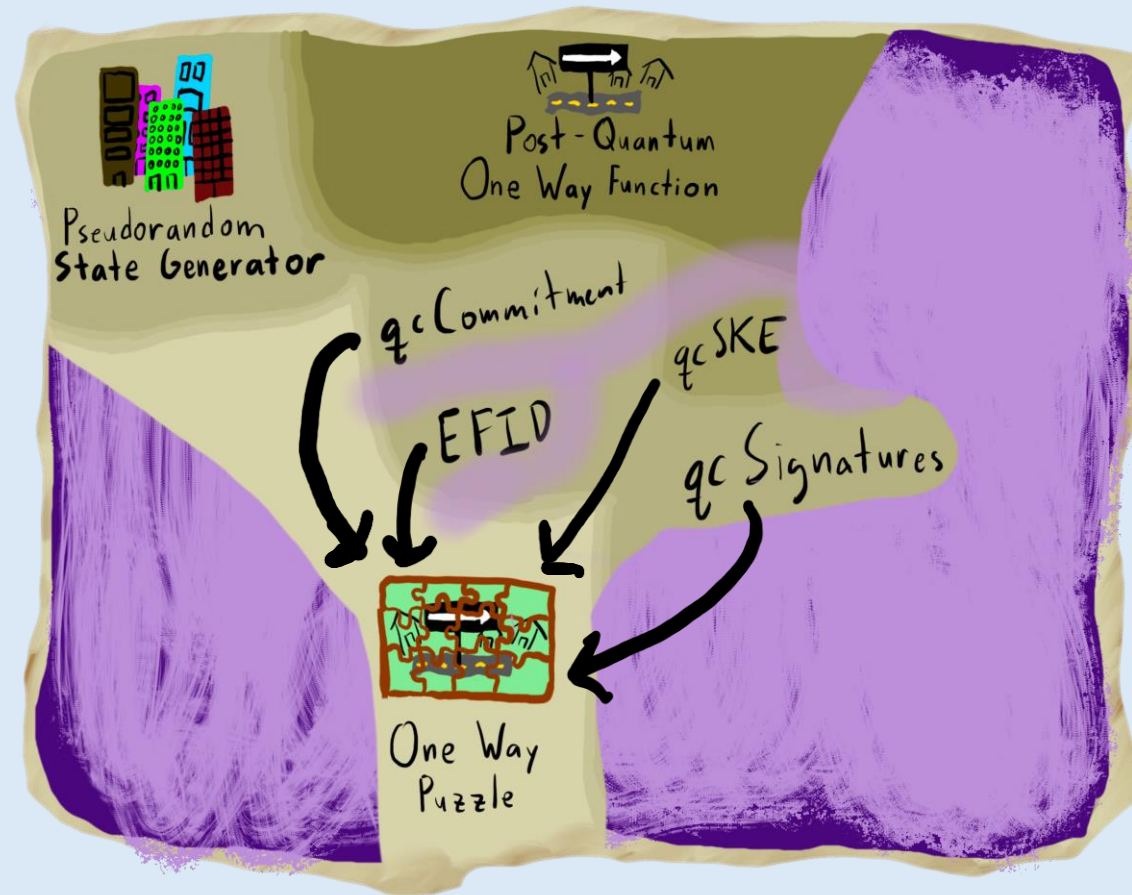
1. Simple ✓
2. Minimal
3. Useful
4. Flexible



# Are one way puzzles a good central primitive?

## Important properties of central primitives

1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible





# Are one way puzzles a good central primitive?

## Important properties of central primitives

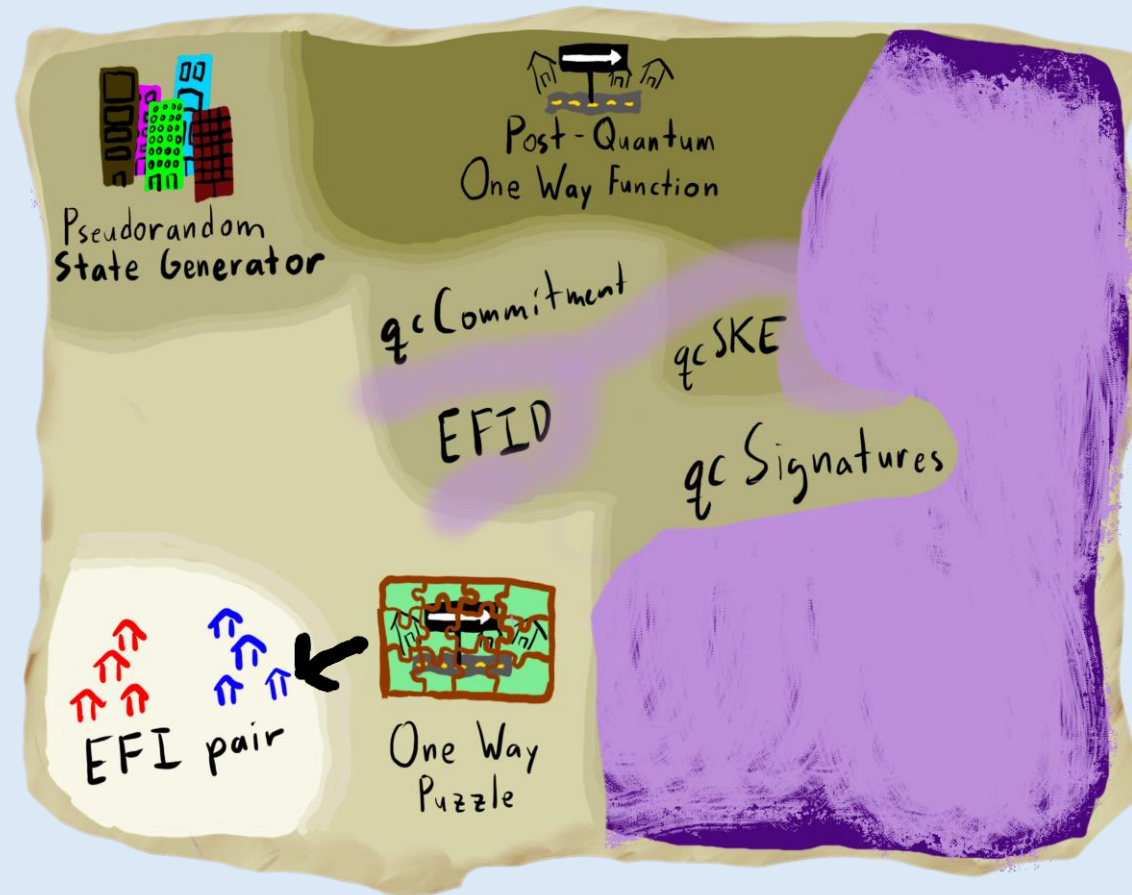
1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible



# Are one way puzzles a good central primitive?

## Important properties of central primitives

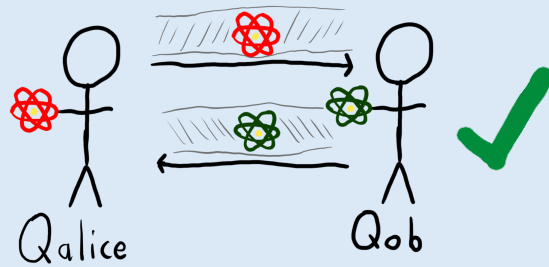
1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible



# Are one way puzzles a good central primitive?

Important properties of central primitives

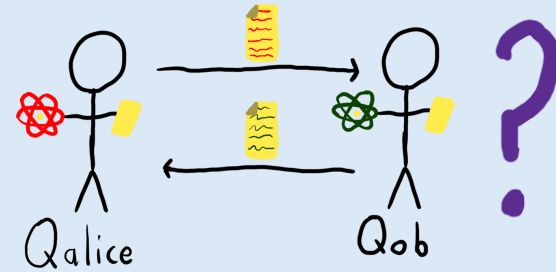
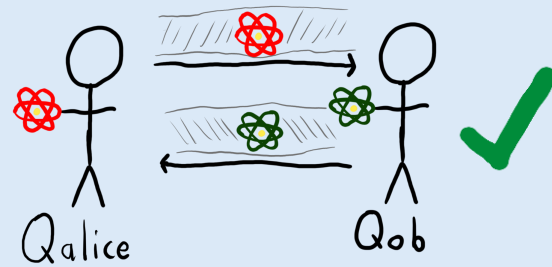
1. Simple ✓
2. Minimal ✓
3. **Useful**
4. Flexible



# Are one way puzzles a good central primitive?

## Important properties of central primitives

1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible ?



# Our Questions

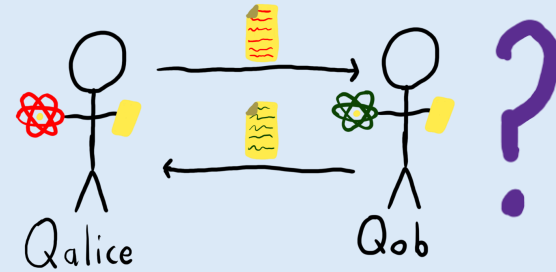
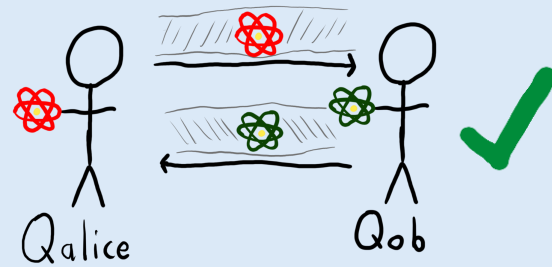
1. Are there any QC primitives implied by one way puzzles?
2. Are one way puzzles well-behaved?

# Our Results

# Are one way puzzles a good central primitive?

## Important properties of central primitives

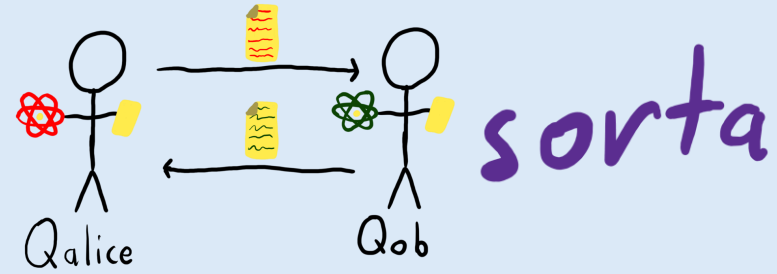
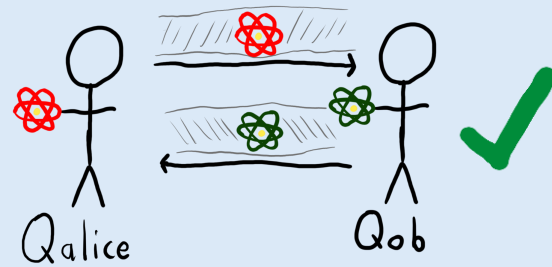
1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible ?



# Are one way puzzles a good central primitive?

## Important properties of central primitives

1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible ?

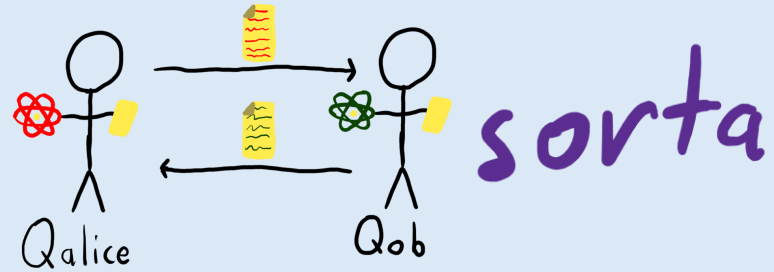
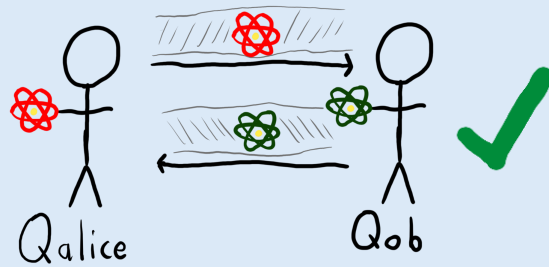




# Are one way puzzles a good central primitive?

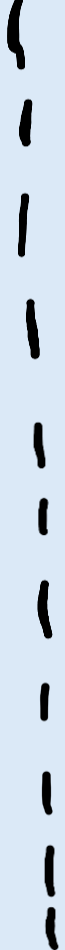
## Important properties of central primitives

1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible ✓



One Way Puzzles are Minimal [KT24]

Signature  $\xrightarrow{\quad}$  One Way Puzzle



# One Way Puzzles are Minimal [KT24]

Signature  $\xrightarrow{\text{!}}$  One Way Puzzle

Gen  $\rightarrow sk, vk$

Sign  $(sk, m) \rightarrow \sigma$

VerSig  $(vk, m, \sigma) \rightarrow 0/1$

# One Way Puzzles are Minimal [KT24]

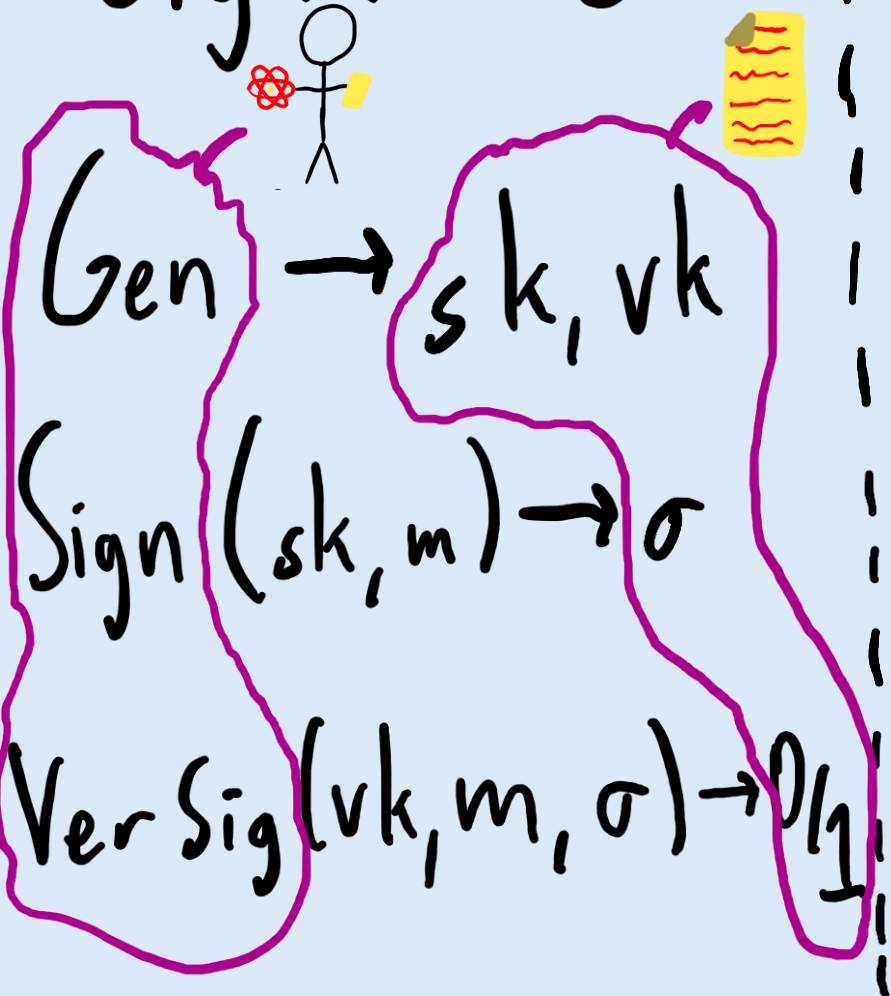
Signature  $\xrightarrow{\quad}$  One Way Puzzle



Gen  $\rightarrow sk, vk$   
Sign  $(sk, m) \rightarrow \sigma$   
VerSig  $(vk, m, \sigma) \rightarrow 0/1$

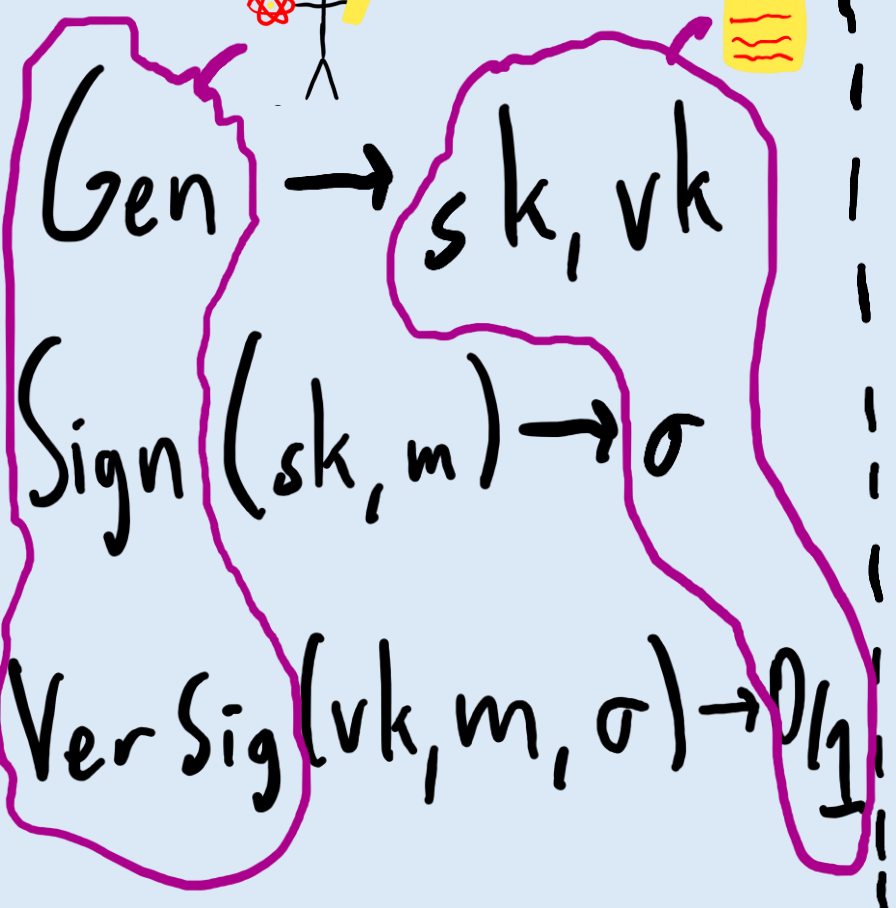
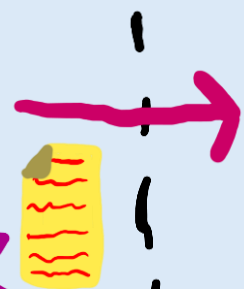
# One Way Puzzles are Minimal [KT24]

Signature  $\rightarrow$  One Way Puzzle

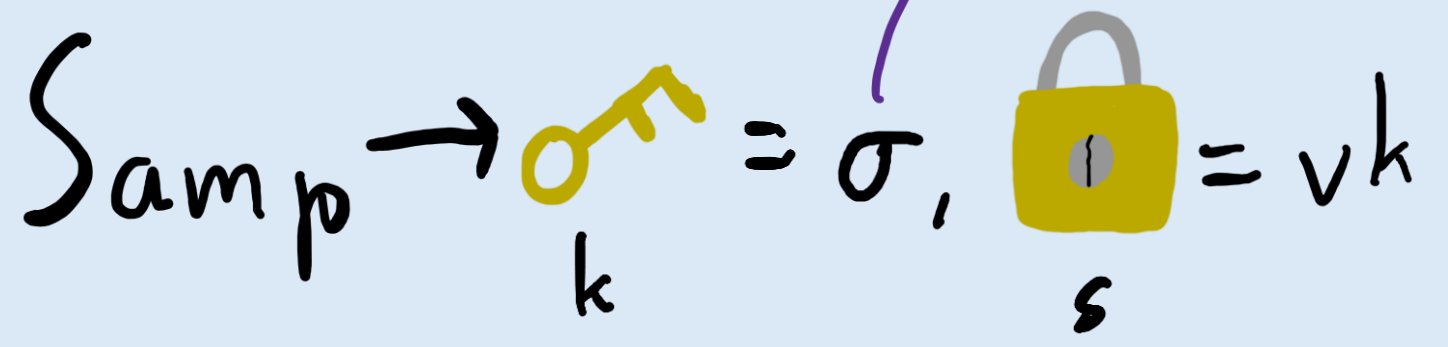


# One Way Puzzles are Minimal [KT24]

Signature



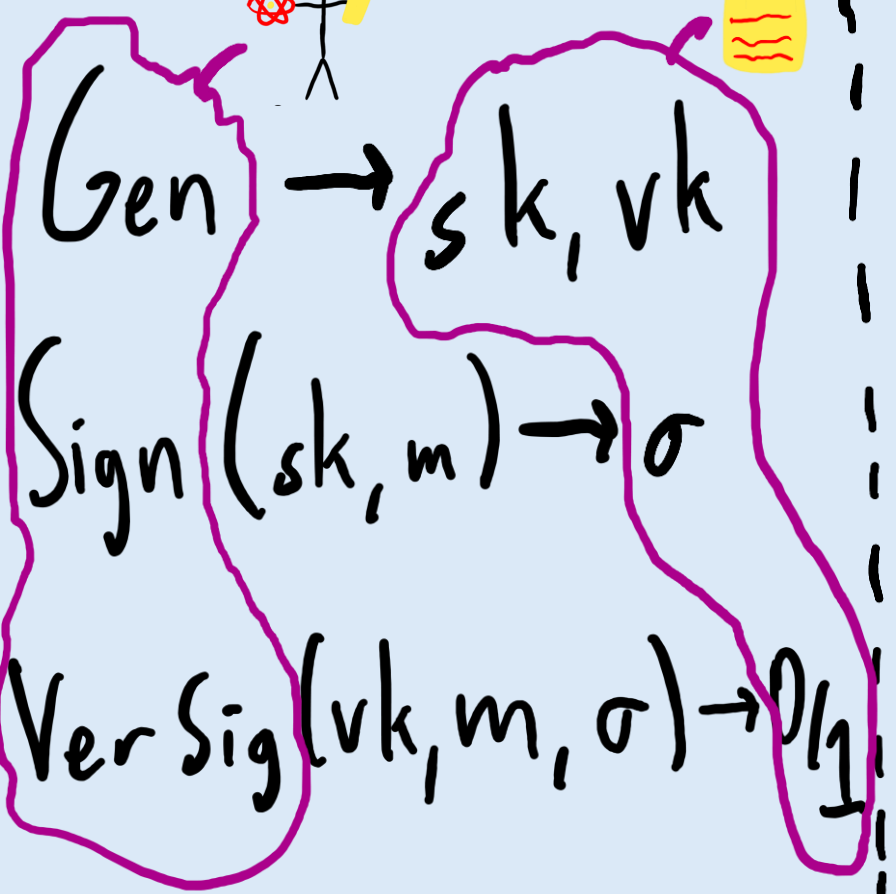
One Way Puzzle



Sign(sk, 0)

# One Way Puzzles are Minimal [KT24]

Signature



One Way Puzzle

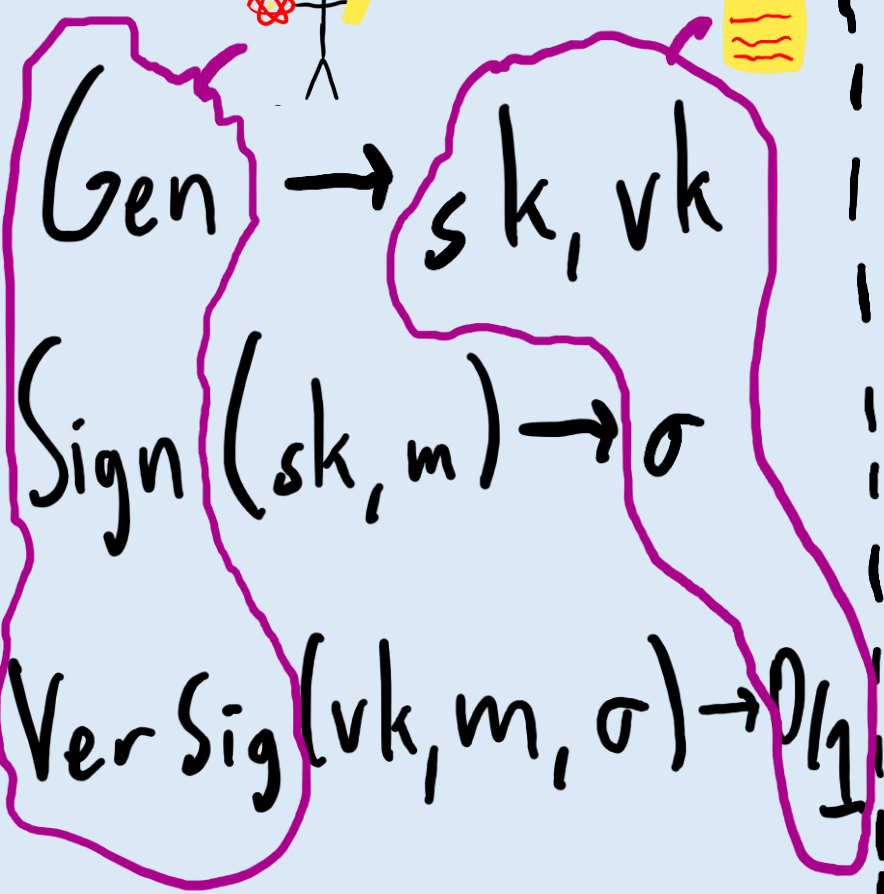
$Sign(sk, 0)$



$$Ver(\sigma, vk) = VerSig(vk, 0, \sigma)$$

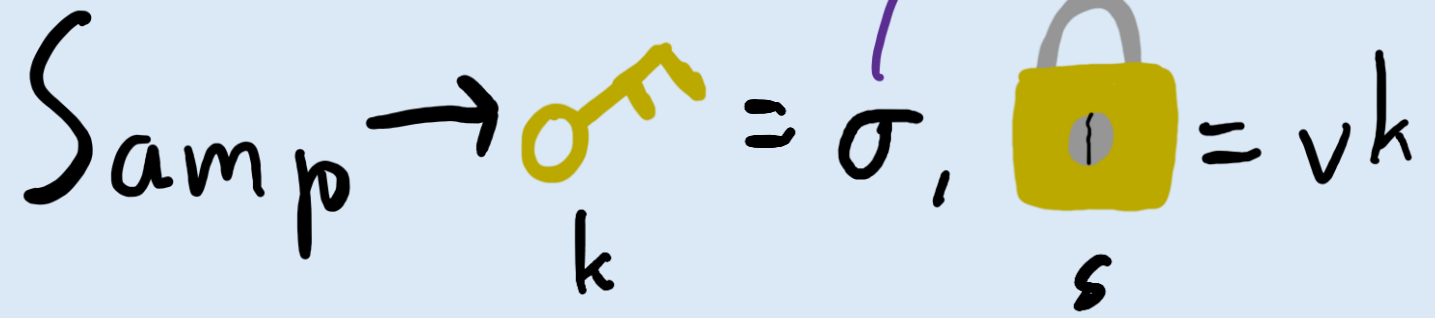
# One Way Puzzles are Minimal [KT24]

Signature



One Way Puzzle

$\text{Sign}(sk, 0)$



$$\text{Ver}(\sigma, vk) = \text{VerSig}(vk, 0, \sigma)$$

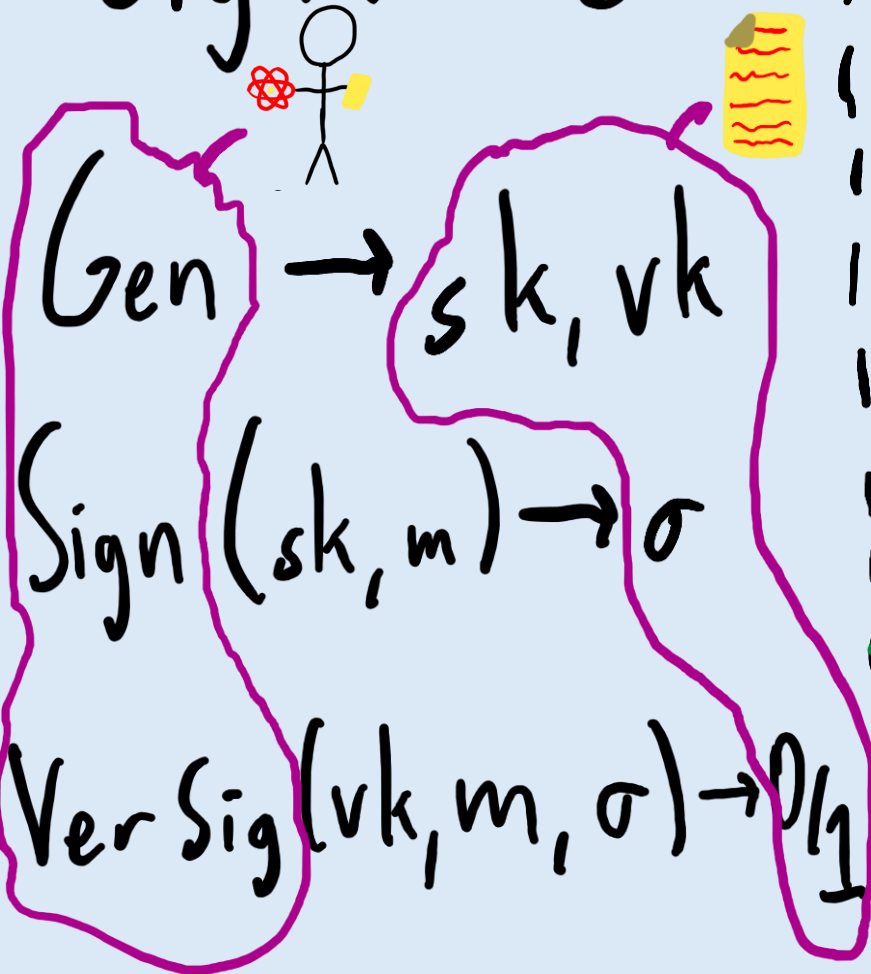
can be inefficient



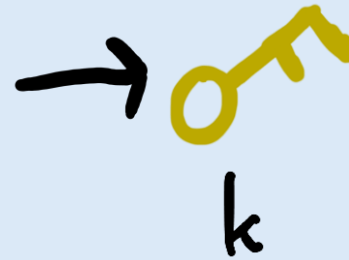
# Are One Way Puzzles Useful?

Signature

One Way Puzzle



Sample



$\text{Sign}(sk, 0)$

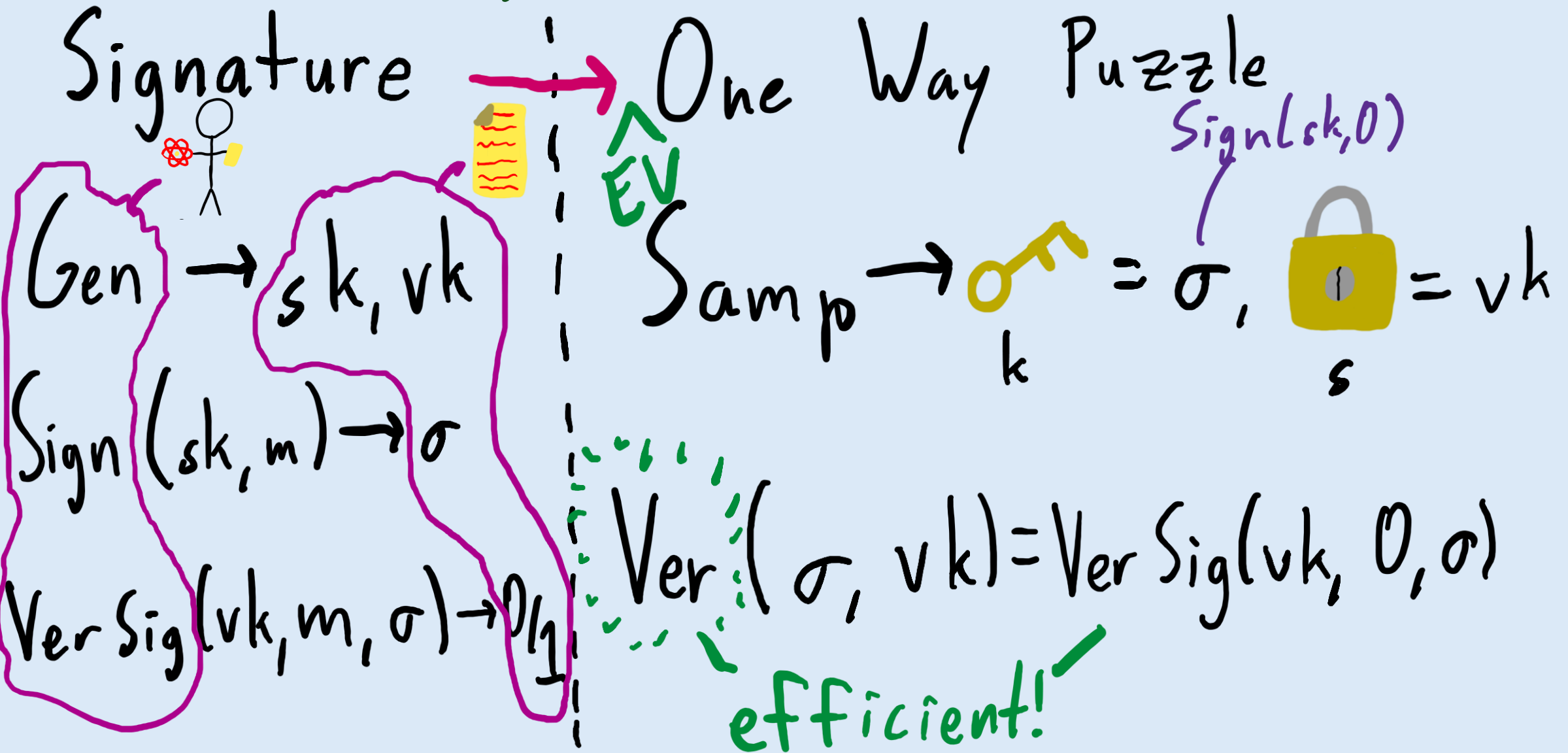


$= vk$

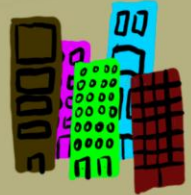
$\text{Ver}(\sigma, vk) = \text{VerSig}(vk, 0, \sigma)$

efficient!

# Are One Way Puzzles Useful?



# Are One Way Puzzles Useful?



Pseudorandom  
State Generator



Post-Quantum  
One Way Function

qc Commitment

EFID

qcSKE

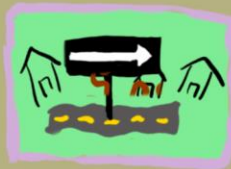
qc Signatures



EFI pair



One Way  
Puzzle

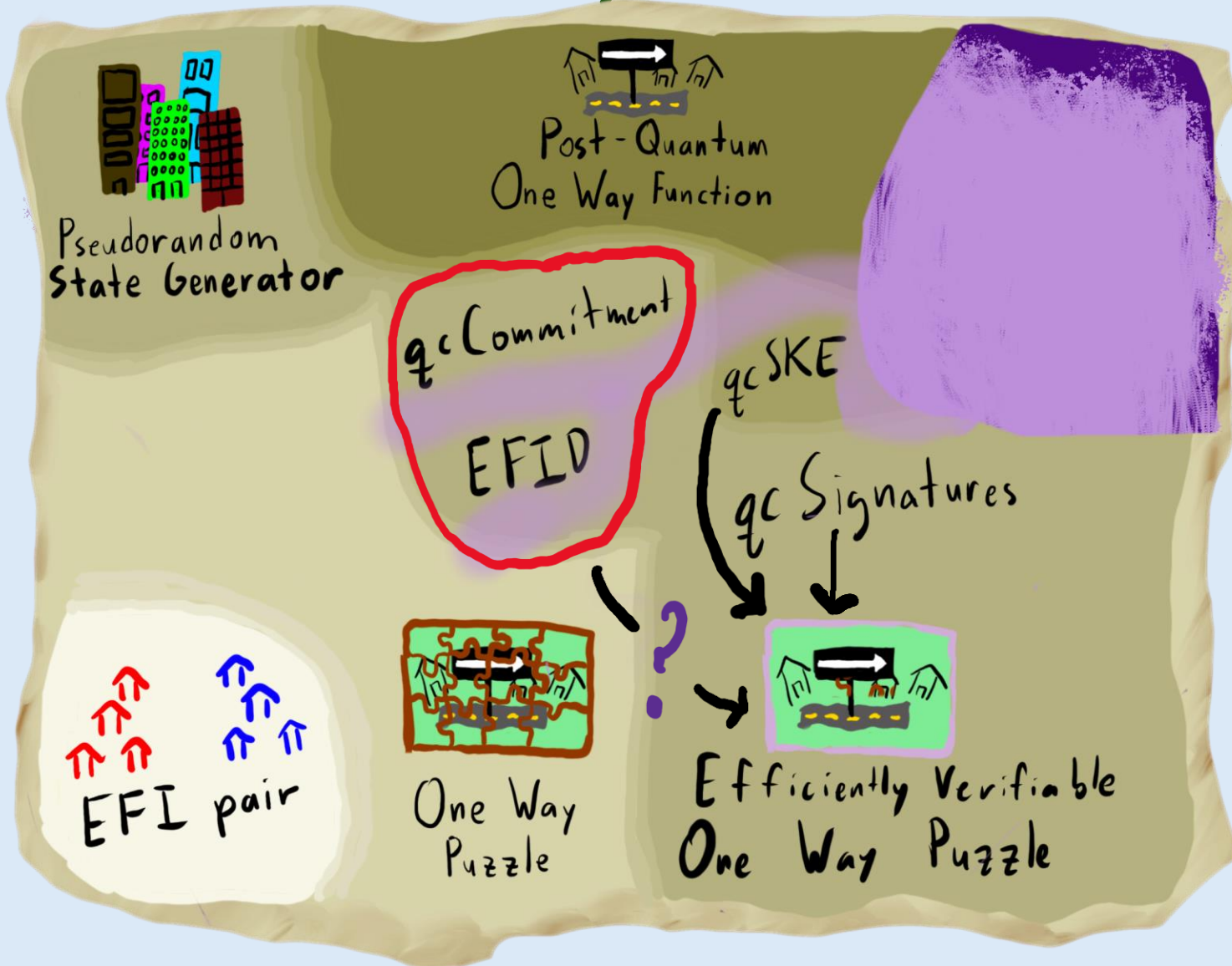


Efficiently Verifiable  
One Way Puzzle

# Are One Way Puzzles Useful?



# Are One Way Puzzles Useful?



Notable exception:  
commitments

# Are One Way Puzzles Useful?



EV-OWPuzz are  
(= 1-time sigs)

# Are One Way Puzzles Useful?



Necessary and sufficient condition for usefulness of OWPuzz

# Are One Way Puzzles Useful?

- Can we build other QC primitives from OWPuzz?

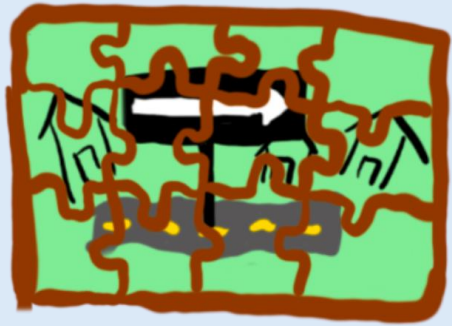
**NO**

exception: maybe commitments

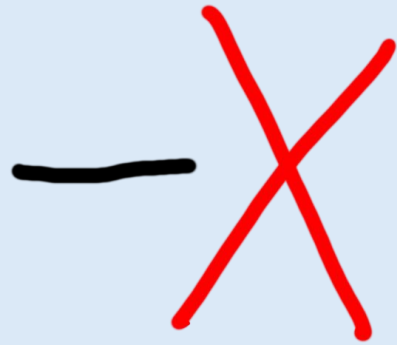


# Are One Way Puzzles Useful?

- Can we build other QC primitives from OWPuzz?



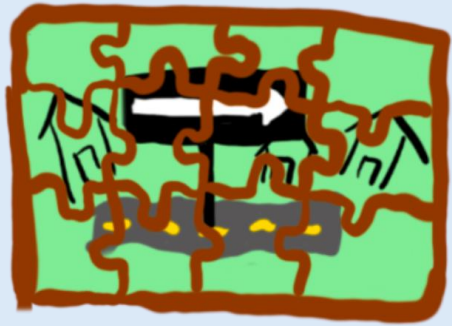
One Way  
Puzzle



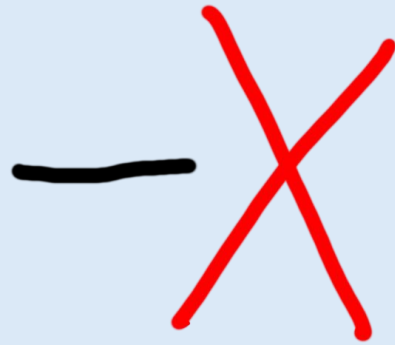
Efficiently Verifiable  
One Way Puzzle

# Are One Way Puzzles Useful?

- Can we build other QC primitives from OWPuzz?



One Way  
Puzzle



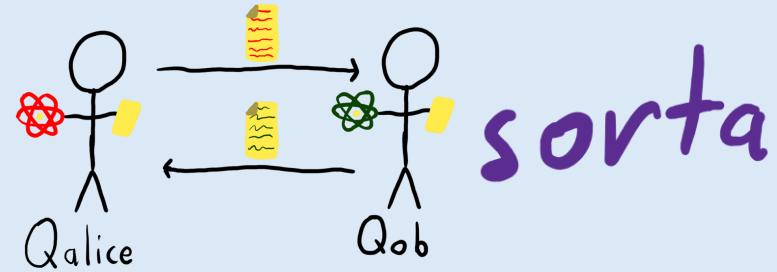
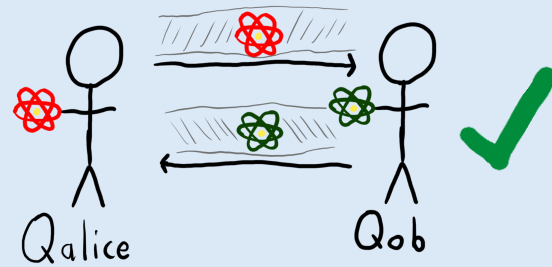
Efficiently Verifiable  
One Way Puzzle

- OWPuzz can exist when  $BQP = QCMA$  [Kreschmer21]
- EV-OWPuzz broken in  $QCMA$  [INNRY22][ABOBS22]

# Are one way puzzles a good central primitive?

## Important properties of central primitives

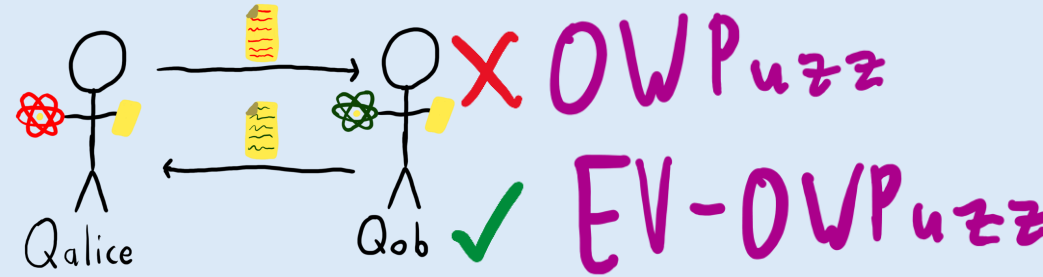
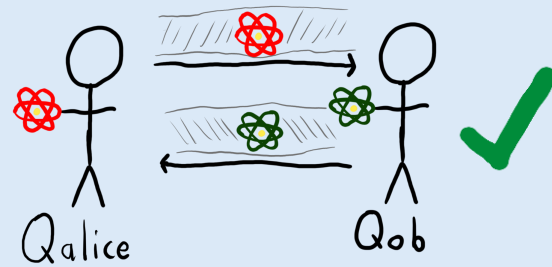
1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible ?



# Are one way puzzles a good central primitive?

## Important properties of central primitives

1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible ?



# Are one way puzzles a good central primitive?

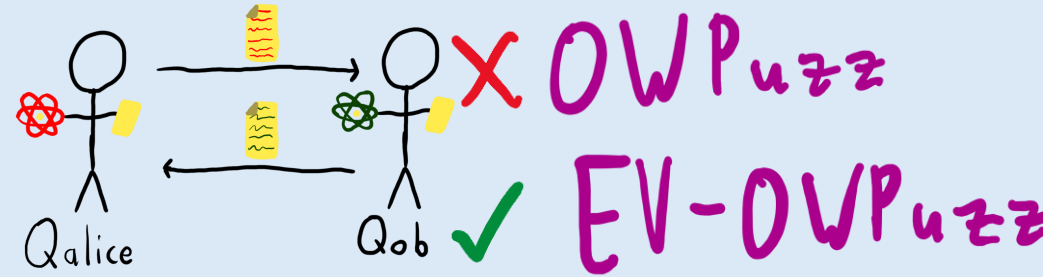
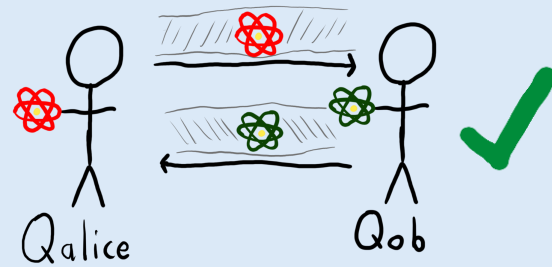
## Important properties of central primitives

1. Simple ✓

2. Minimal ✓

3. Useful

4. Flexible?

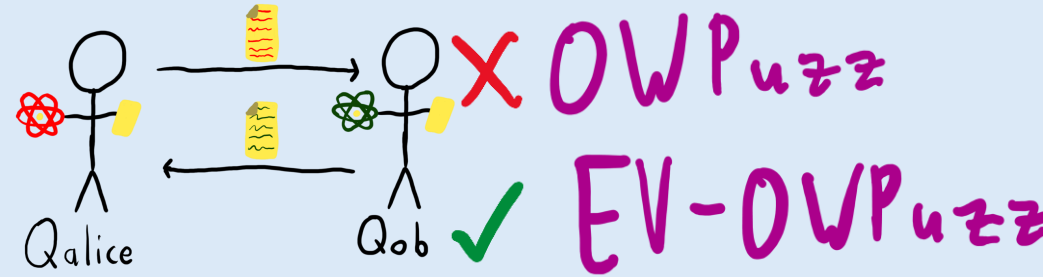
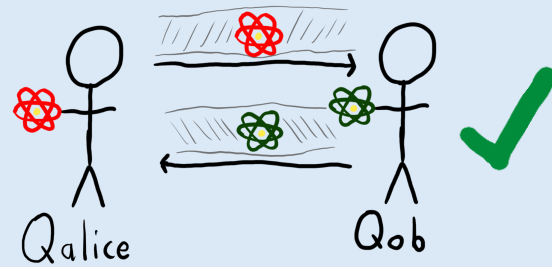


$$(Samp, Ver) + (Samp', Ver') \rightarrow (\widetilde{Samp}, \widetilde{Ver})$$

# Are one way puzzles a good central primitive?

## Important properties of central primitives

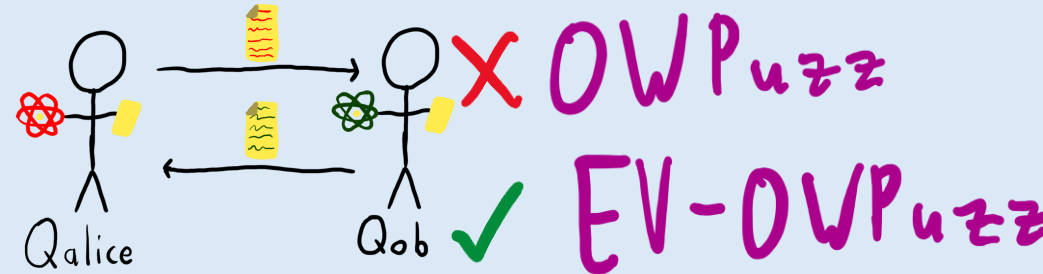
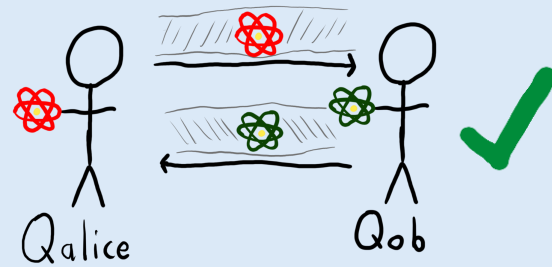
1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible? ?



# Are one way puzzles a good central primitive?

## Important properties of central primitives

1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible?



"Dist OWPuzz"



"Weak OWPuzz"

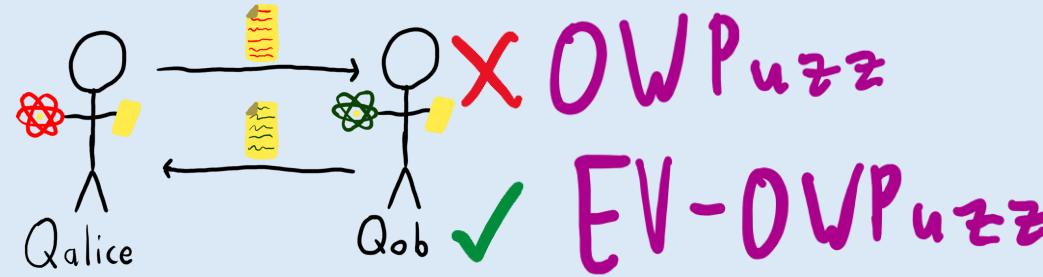
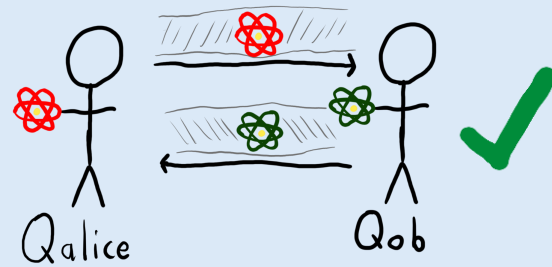


"Strong OWPuzz"

# Are one way puzzles a good central primitive?

## Important properties of central primitives

1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible?

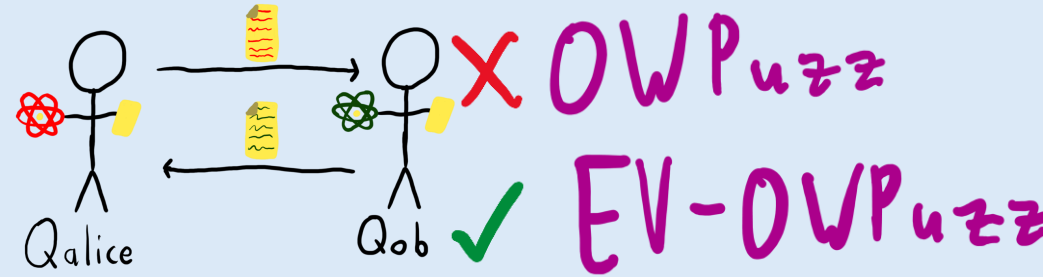
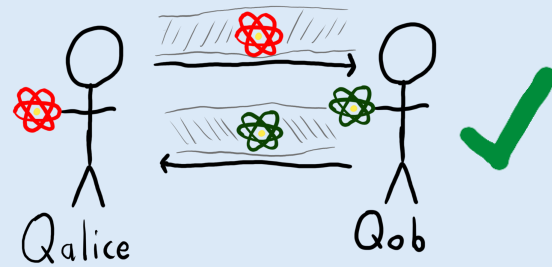




# Are one way puzzles a good central primitive?

## Important properties of central primitives

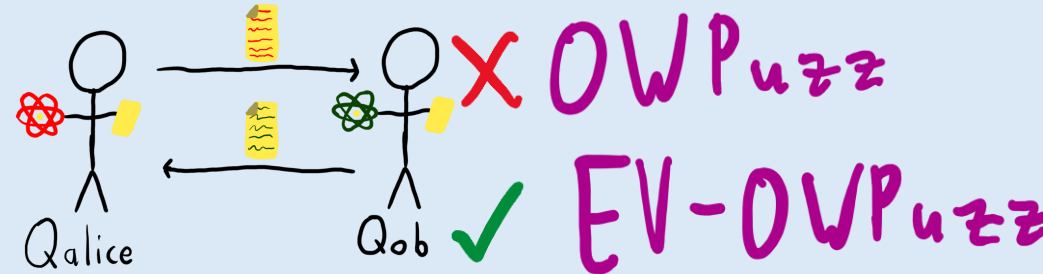
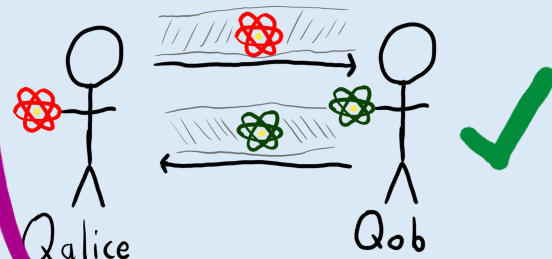
1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible ✓



# Are one way puzzles a good central primitive?

## Important properties of central primitives

1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible ✓



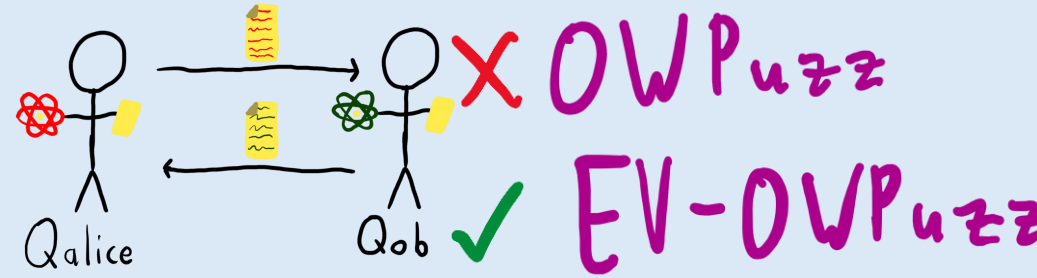
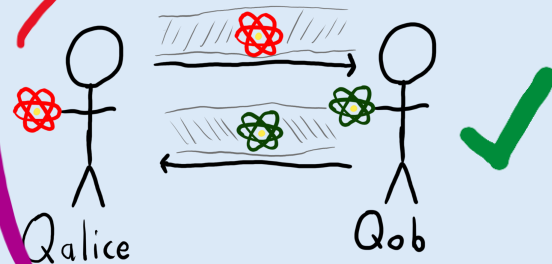
For both OWPuzzle & EV-OWPuzzle

# Are one way puzzles a good central primitive?

Important properties of central primitives

1. Simple ✓
2. Minimal ✓
3. Useful
4. Flexible ✓

exception: commitments



For both OWPuzz & EV-OWPuzz

# QC Topography



Pseudorandom  
State Generator



Post-Quantum  
One Way Function

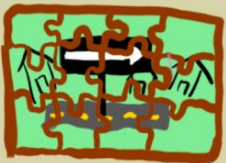
qc Commitment

EFID

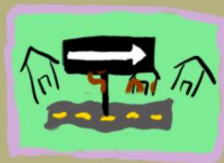
qcSKE

qc Signatures

EFI pair

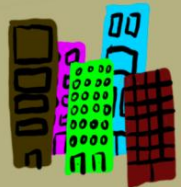


One Way  
Puzzle



Efficiently Verifiable  
One Way Puzzle

# QC Topography



Pseudorandom  
State Generator



Post-Quantum  
One Way Function



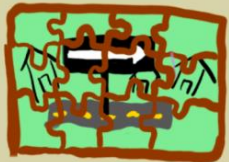
qc Commitment

qc SKE

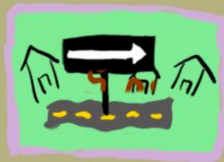
EFID

qc Signatures

EFI pair



One Way  
Puzzle



Efficiently Verifiable  
One Way Puzzle

# One Way Puzzle Amplification

Why is this hard?

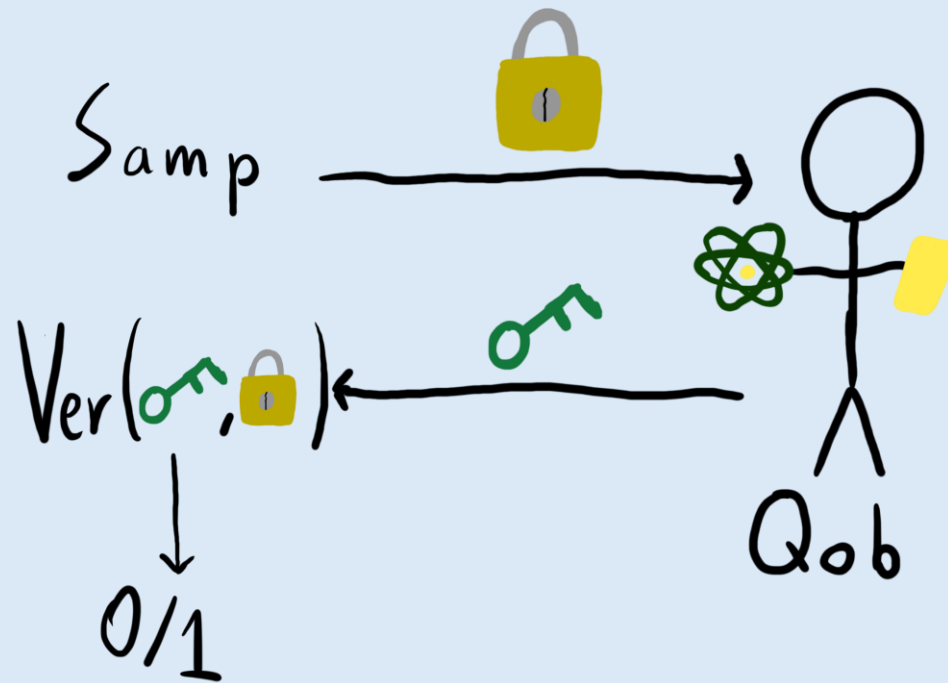
*We can* amplify soundness for efficient 3 round quantum interactive protocols via parallel repetition [BQSY23].

Why is this hard?

We *can* amplify soundness for efficient 3 round quantum interactive protocols via parallel repetition [BQSY23].

Security

Game



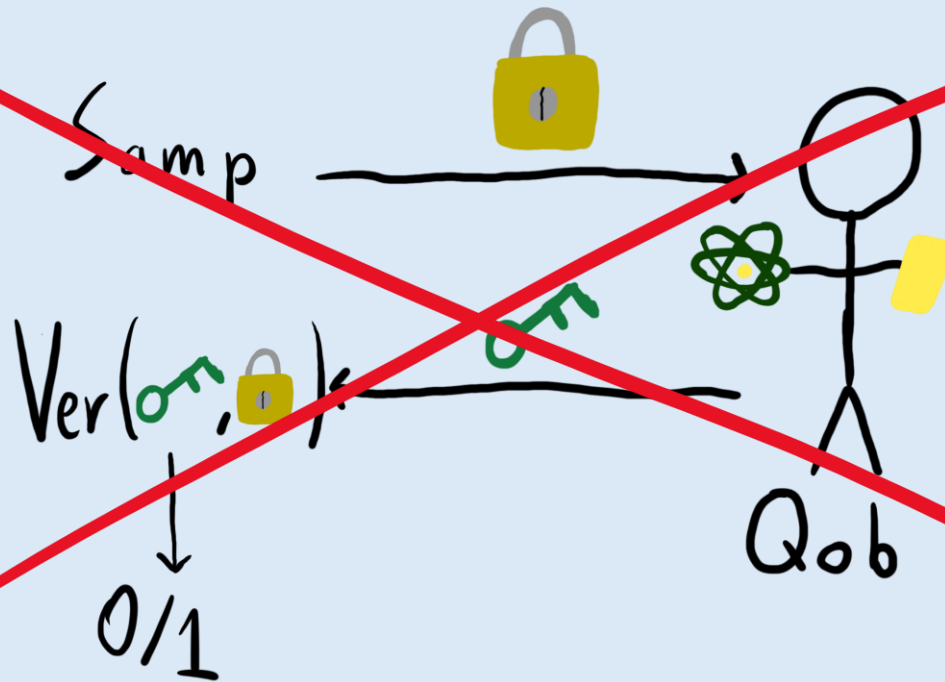


Why is this hard?

We *can* amplify soundness for efficient 3 round quantum interactive protocols via parallel repetition [BQSY23].

Security

Game

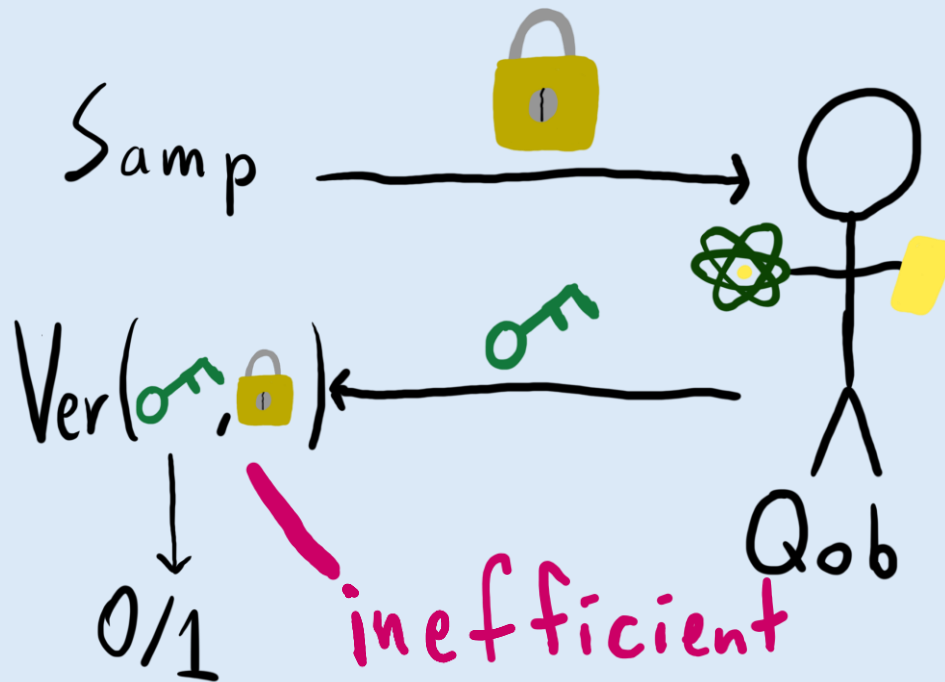


# Why is this hard?

We *can* amplify soundness for efficient 3 round quantum interactive protocols via parallel repetition [BQSY23].

Security

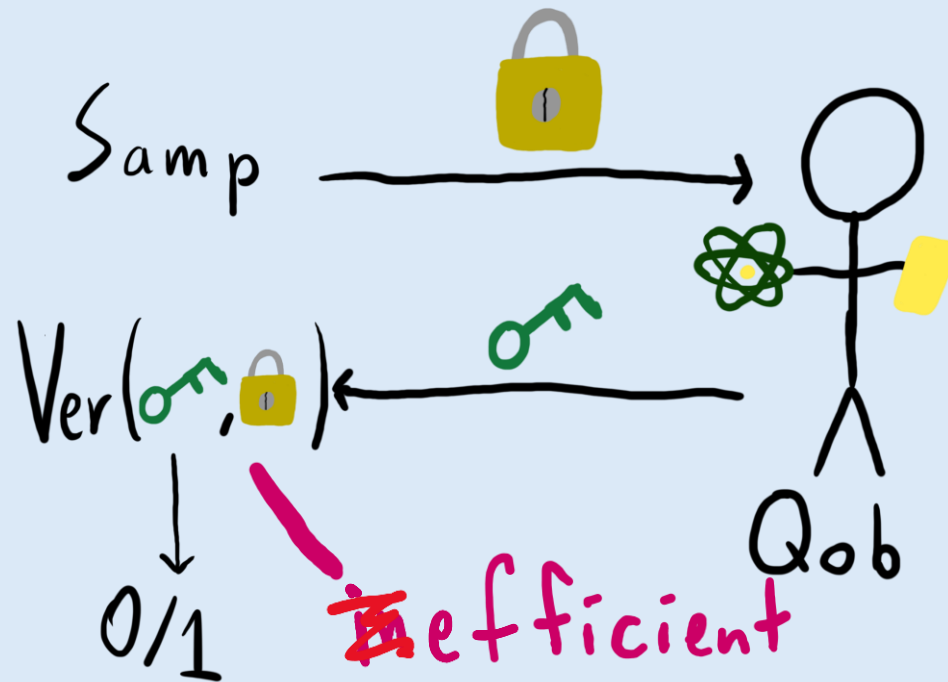
Game



Why is this hard?

We *can* amplify soundness for efficient 3 round quantum interactive protocols via parallel repetition [BQSY23].

Security  
Game



Works for  
EV-DWPuzz

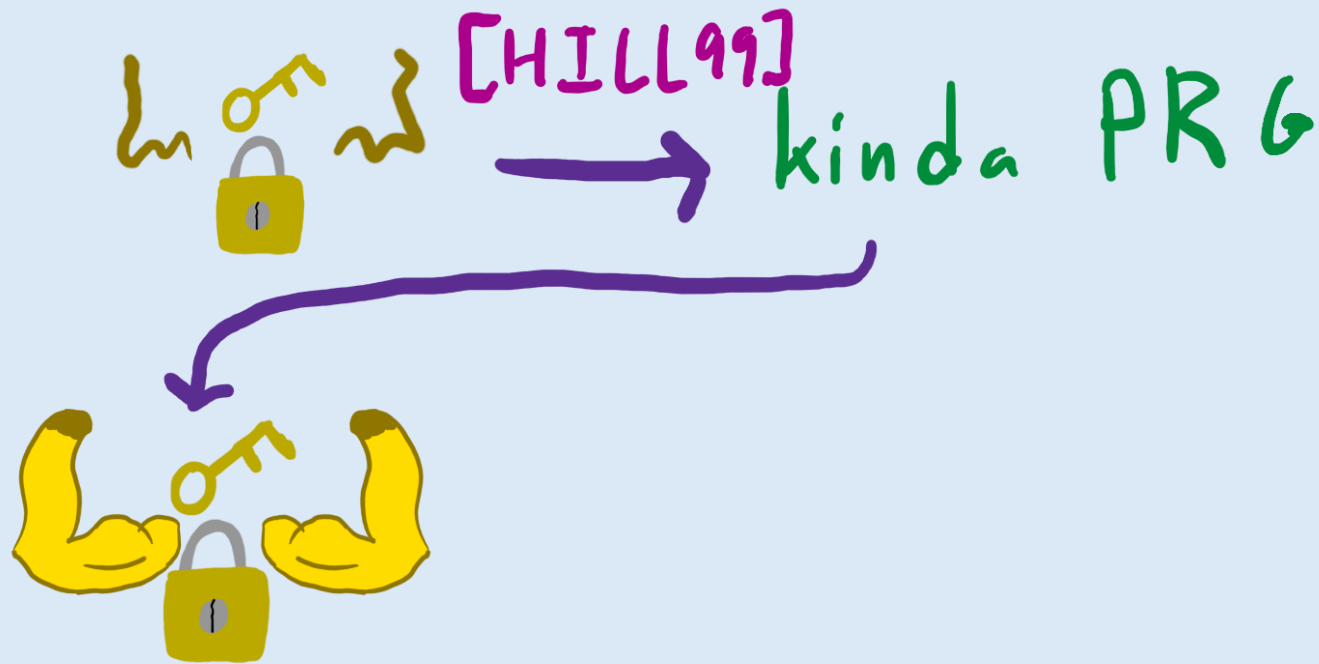
# Our approach

- [KT24] builds “kinda PRGs” (EFID) from OWPuzz using techniques of [HILL99]



# Our approach

- [KT24] builds “kinda PRGs” (EFID) from OWPuzz using techniques of [HILL99]



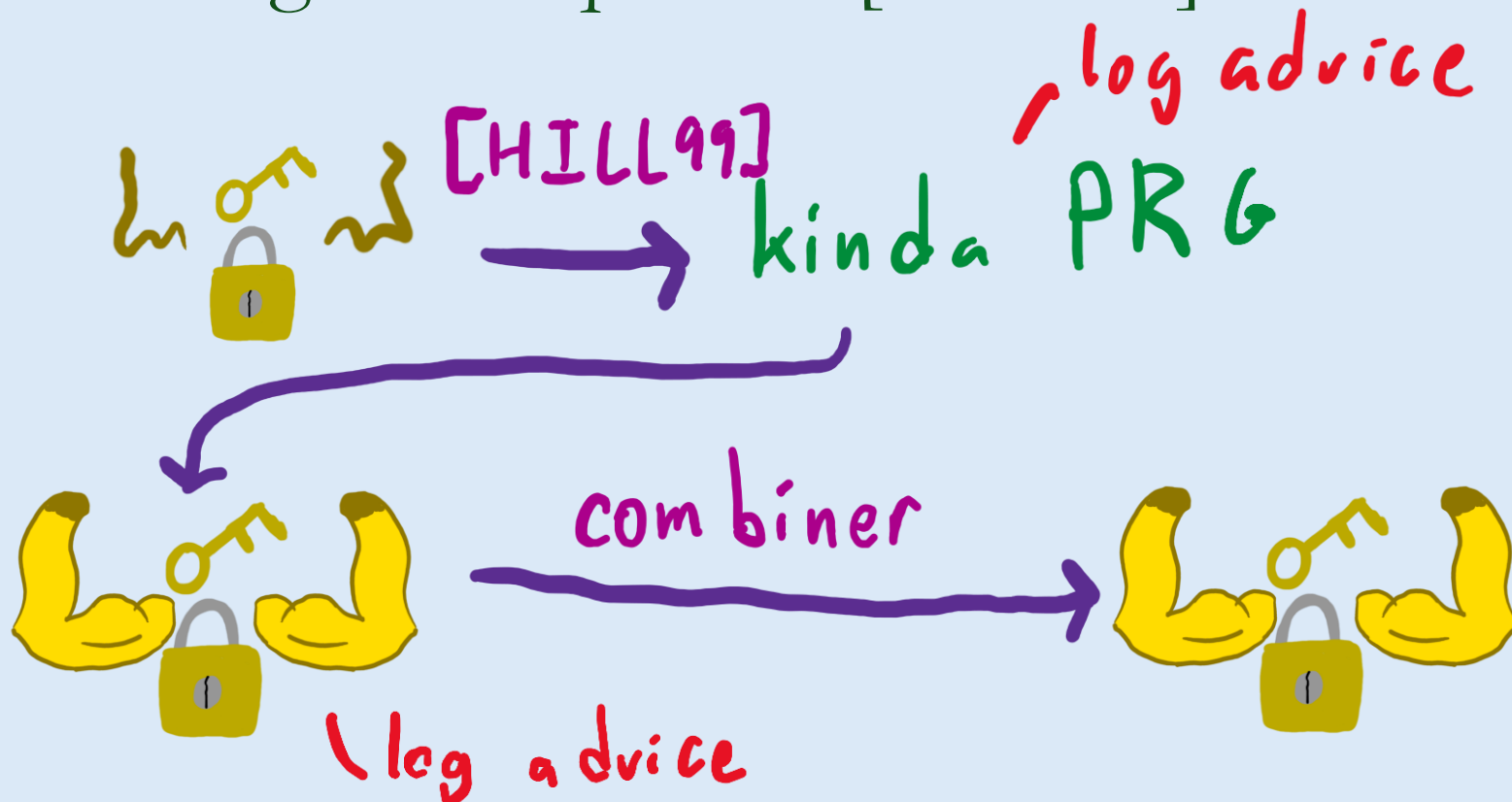
# Our approach

- [KT24] builds “kinda PRGs” (EFID) from OWPuzz using techniques of [HILL99]



# Our approach

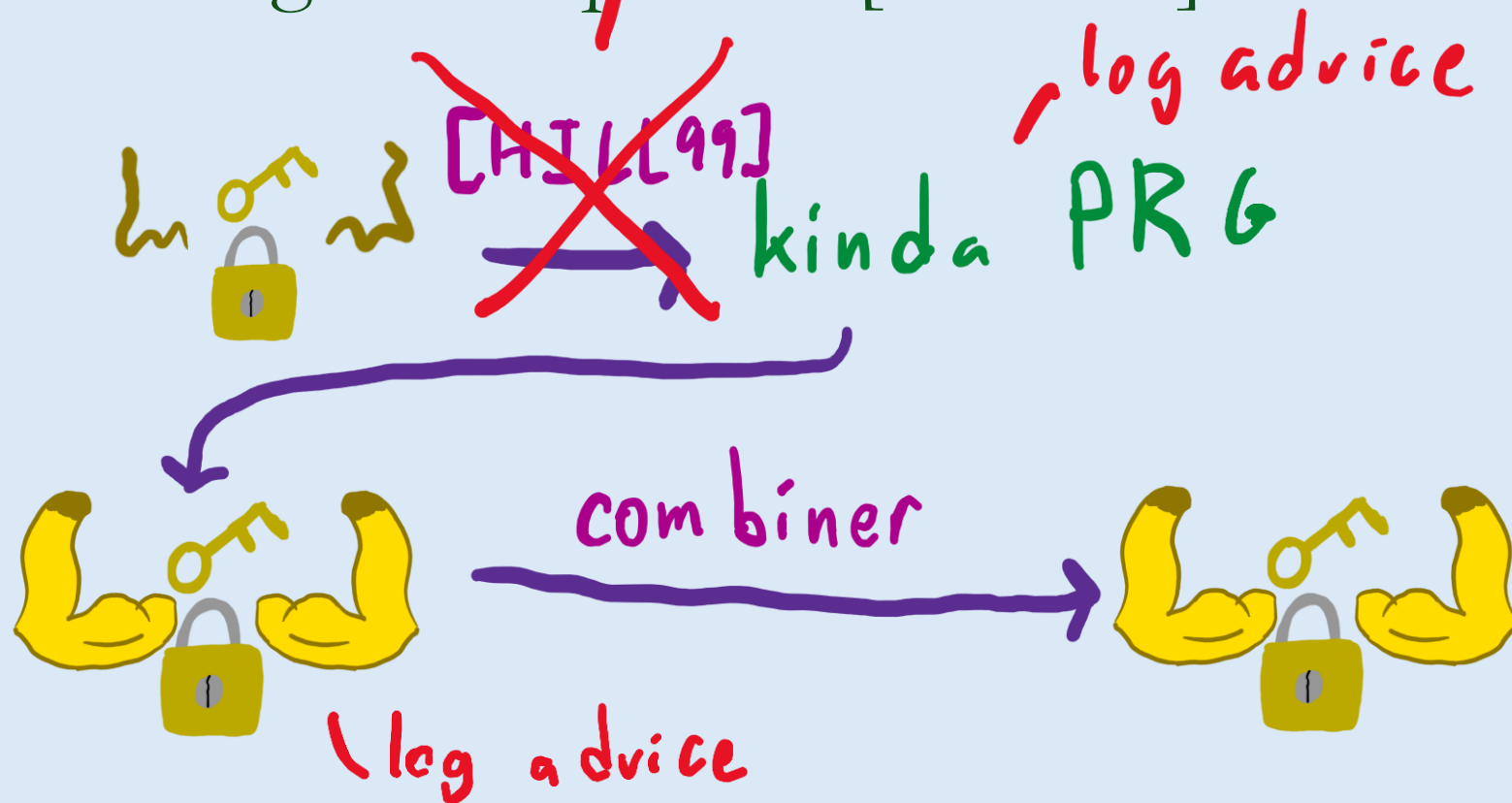
- [KT24] builds “kinda PRGs” (EFID) from OWPuzz using techniques of [HILL99]



# Our approach

GL can only extract  
1 bit of randomness

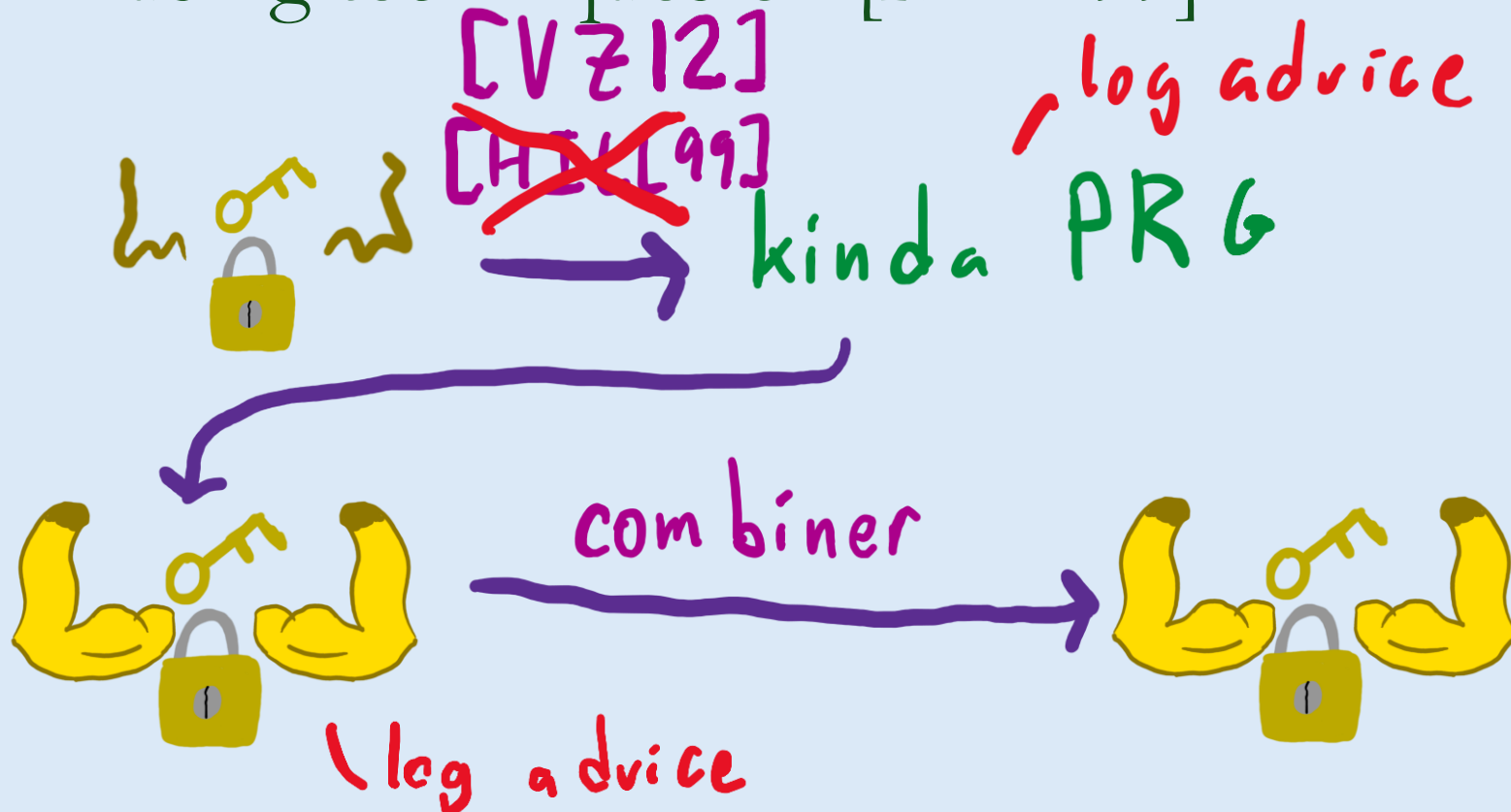
- [KT24] builds “kinda PRGs” (EFID) from OWPuzz using techniques of [HILL99]





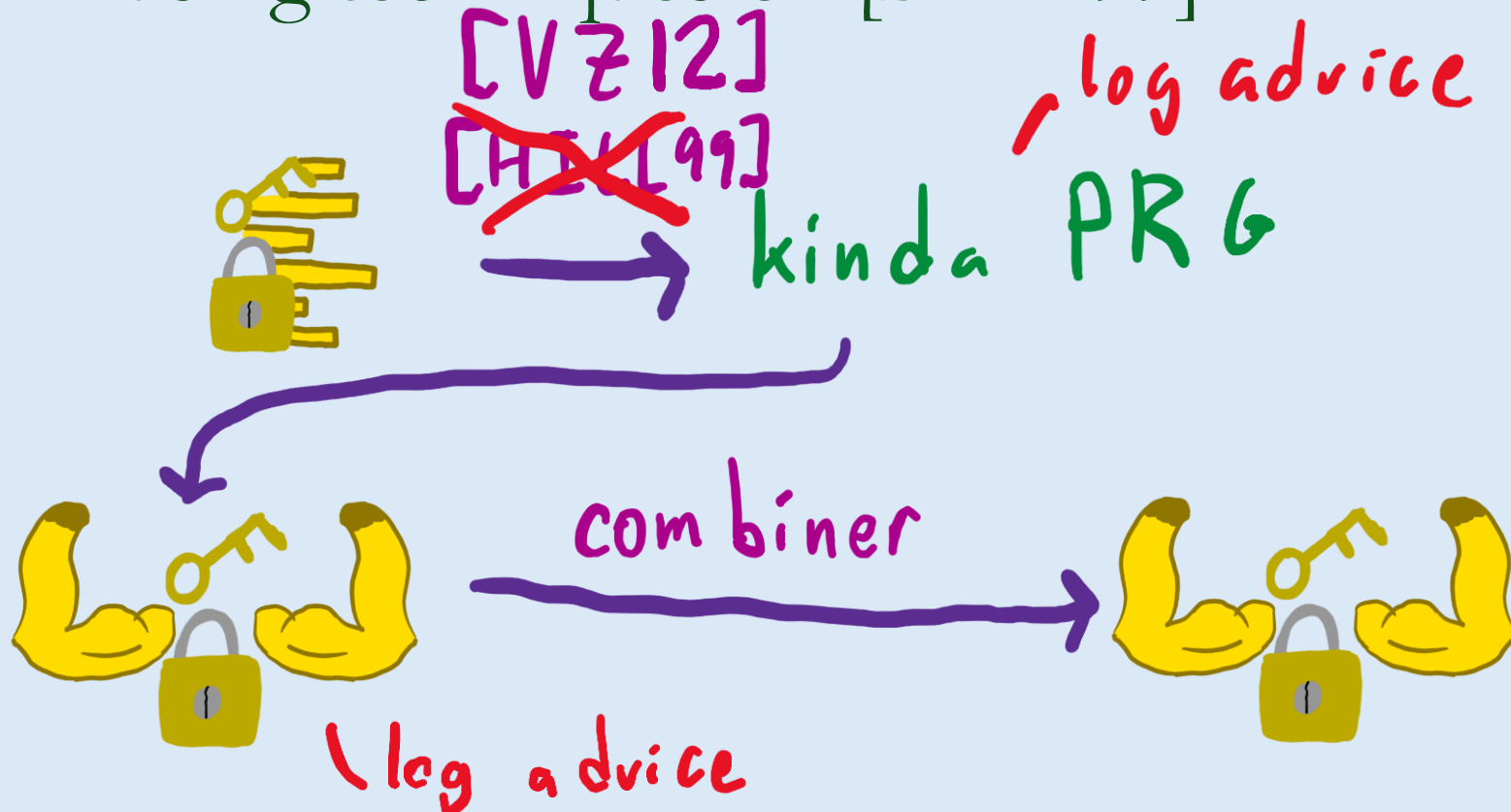
# Our approach

- [KT24] builds “kinda PRGs” (EFID) from OWPuzz using techniques of [HILL99]



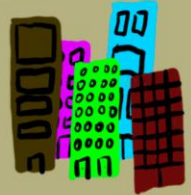
# Our approach

- [KT24] builds “kinda PRGs” (EFID) from OWPuzz using techniques of [HILL99]



# Additional Results

# Additional Results



Pseudorandom  
State Generator



Post-Quantum  
One Way Function

qc Commitment

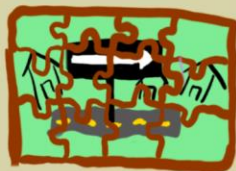
EFID

qcSKE

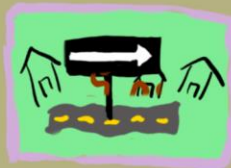
qc Signatures



EFI pair

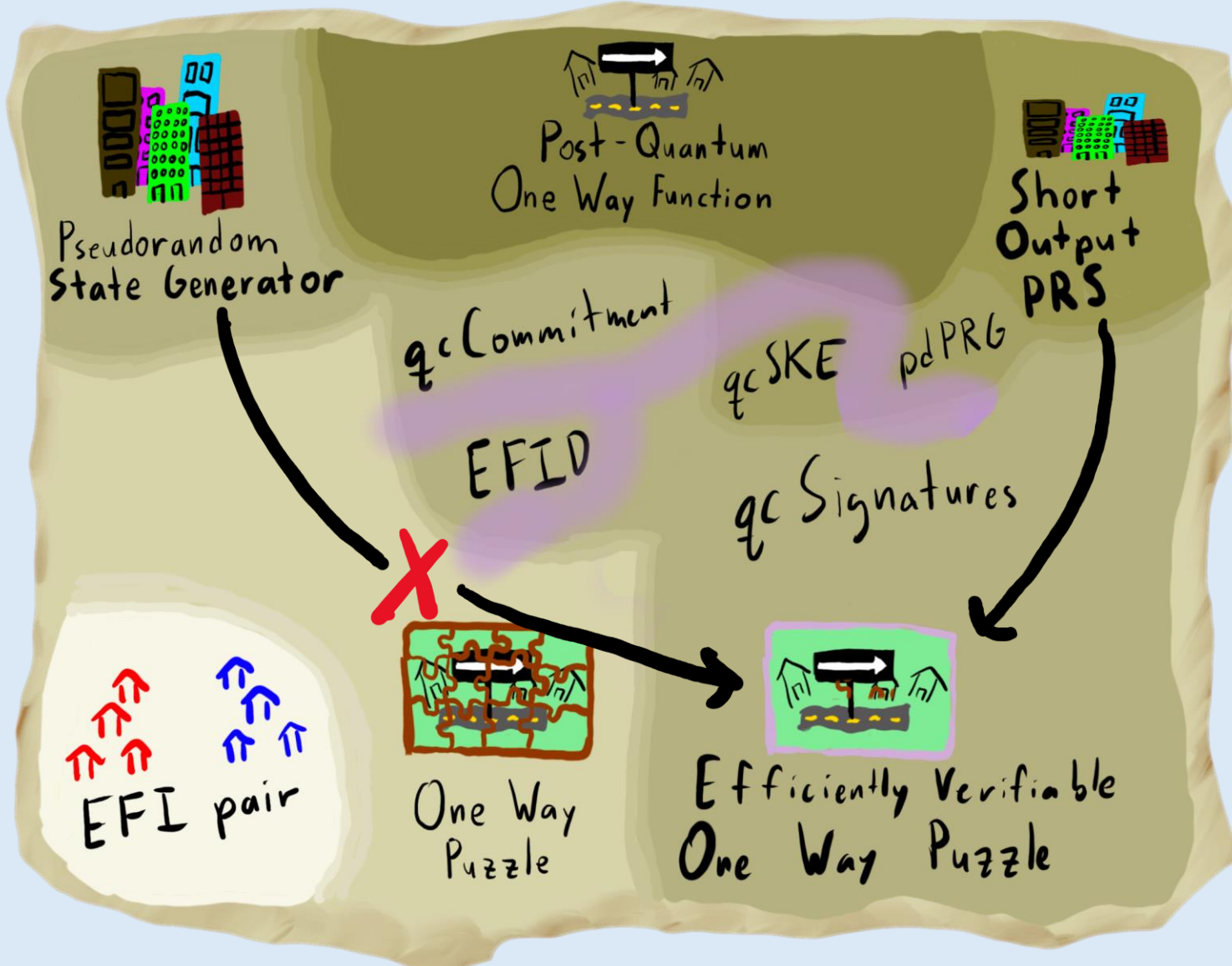


One Way  
Puzzle

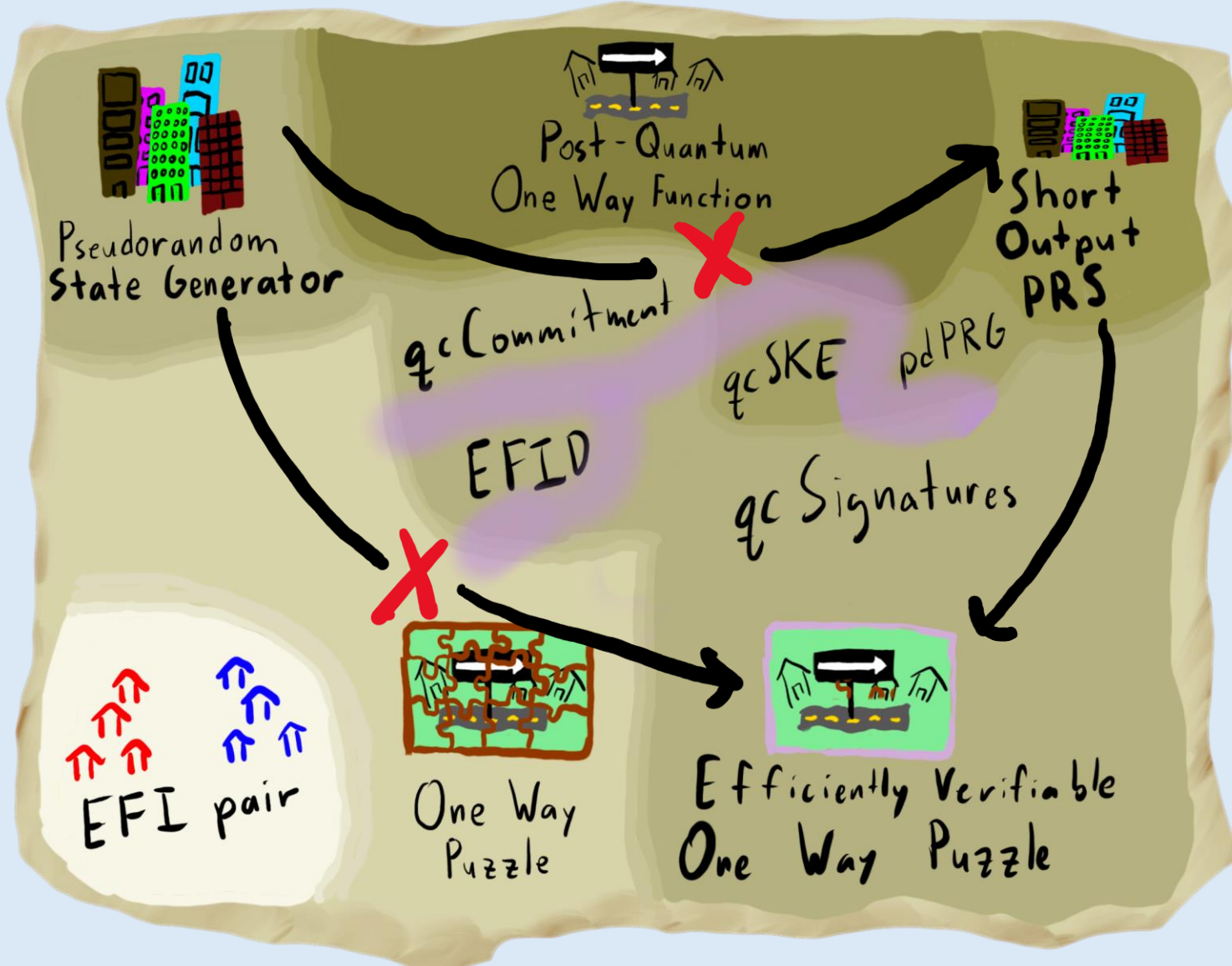


Efficiently Verifiable  
One Way Puzzle

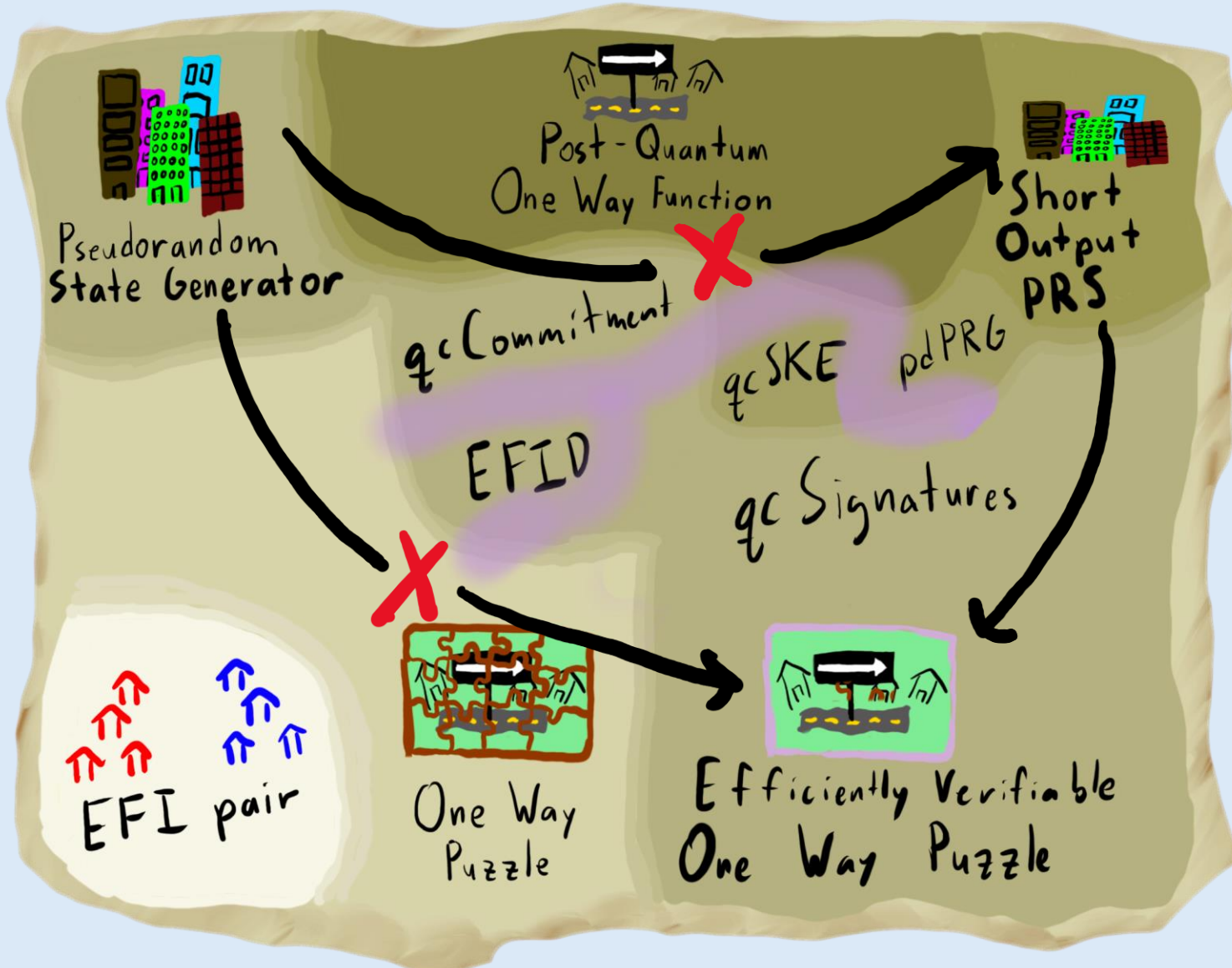
# Additional Results



# Additional Results



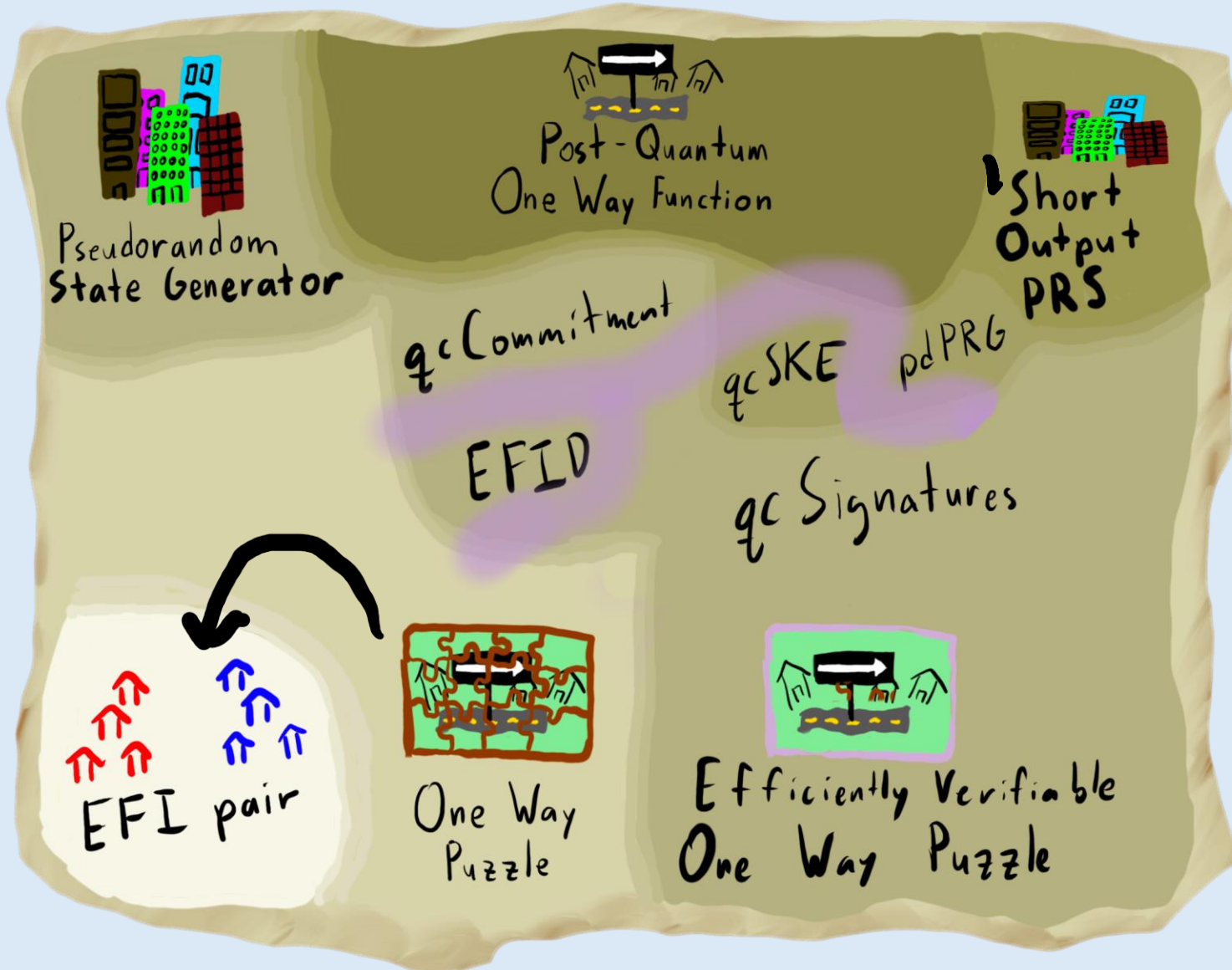
# Additional Results



Separation between pseudorandom states with long output and short output.

Concurrently,  
[CM24][BM24]

# Additional Results



Cleaner construction of EFI from OWPuzz using techniques of [VZ12]



# Open Questions

OWPuzz  $\xrightarrow{?}$  qc Commitments

## Open Questions

OWPuzz  $\xrightarrow{?}$  qc Commitments

OWPuzz  $\xrightarrow{?}$  anything qc

## Open Questions

OWPuzz  $\xrightarrow{?}$  qc Commitments

OWPuzz  $\xrightarrow{?}$  anything qc

EV-OWPuzz  $\xrightarrow{?}$  anything qc  
besides signatures