Algorand Centres of Excellence

MEGA-ACE

SCALES

MPC with Small Clients and Larger Ephemeral Servers

Anasuya Acharya — Bar Ilan University
Carmit Hazay — Bar Ilan University
Vladimir Kolesnikov — Georgia Institute of Technology
Manoj Prabhakaran — Indian Institute of Technology Bombay

MPC with Specialized Communication Patterns

[BGG+20,GHK+21,CGG+21,GMPS21,GHM+21,KRY22]

- Large pool of parties
- Short term workers
- Motivated by blockchain platforms

The YOSO Model [GHK+21]

- You Only Speak Once
  - Parties compute a message, erase state, send message to a receiver with unknown ID
- Avoid adaptive corruption
  - by not revealing identity until server sends a message

Existing YOSO protocols require

- **Target anonymous channels** [BGG+20, GHM+21]
- n-party **committees**, each with **honest majority**
- **Number of committees** proportional to **size of computation**

Our Contributions

- Define **SCALES**: an Ephemeral Servers model (clients speak twice)

- Construct **a SCALES protocol** in the **semi-honest model** using
  - Constant number of servers
  - All-but-one corruption (dishonest majority)
  - Without target anonymous channels (no PKI)

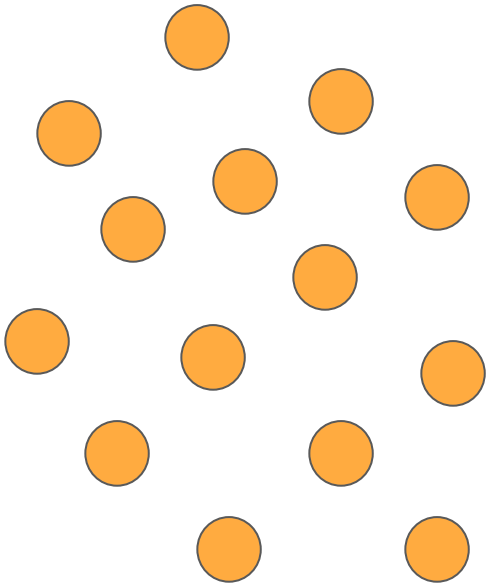
- Define and construct its **building-blocks**
  - Strong Key-and-Message Homomorphic Encryption
  - Rerandomizable Garbling Schemes
  - Incremental Decomposable Randomized Encodings

Also used to fix a gap in the proof of multi-hop FHE [GHV10]

Outline

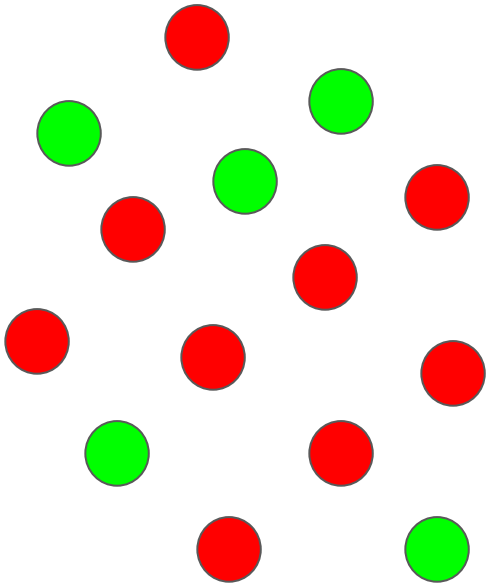
- **SCALES**
- Rerandomizable Garbling Schemes
- Construction - RGCs
- A SCALES protocol

Server Pool



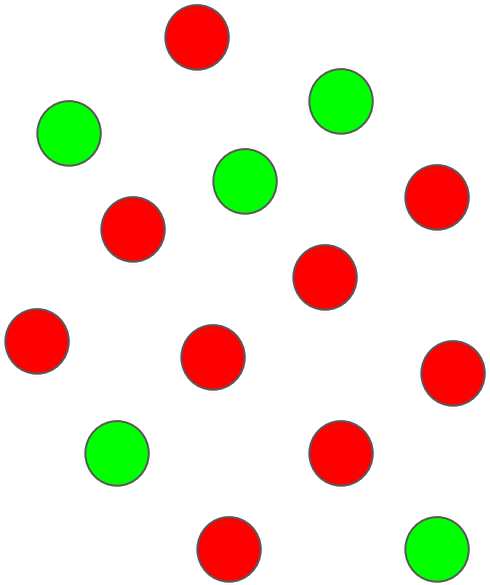
SCALES - the model

Server Pool



SCALES - the model

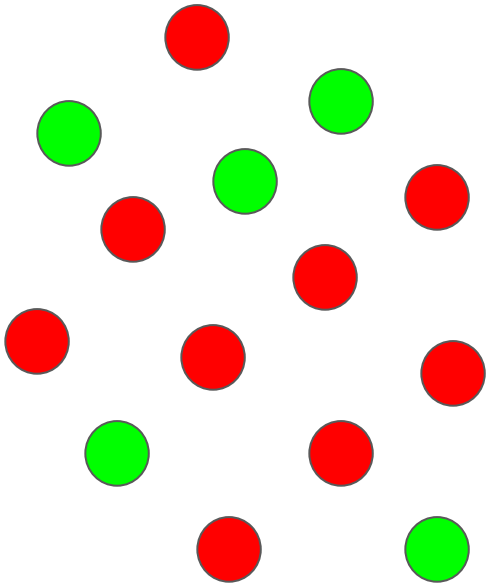
Server Pool



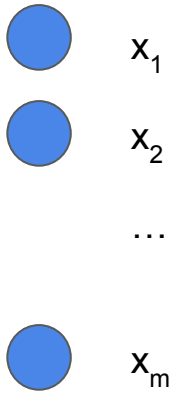
Clients

SCALES - the model

Server Pool

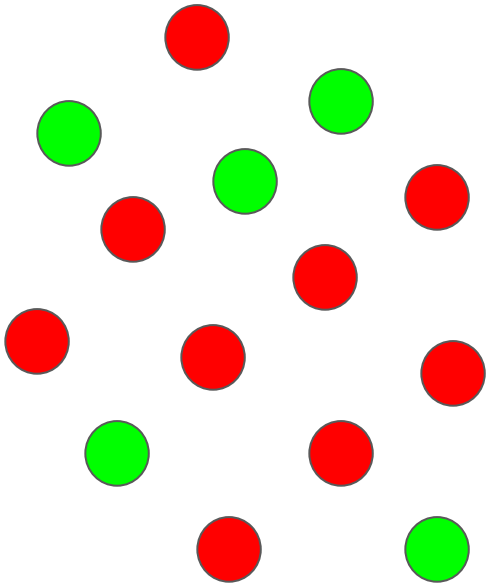


Clients

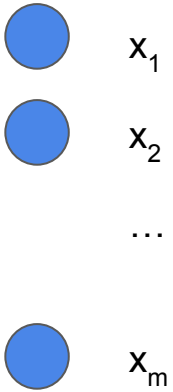


SCALES - the model

Server Pool



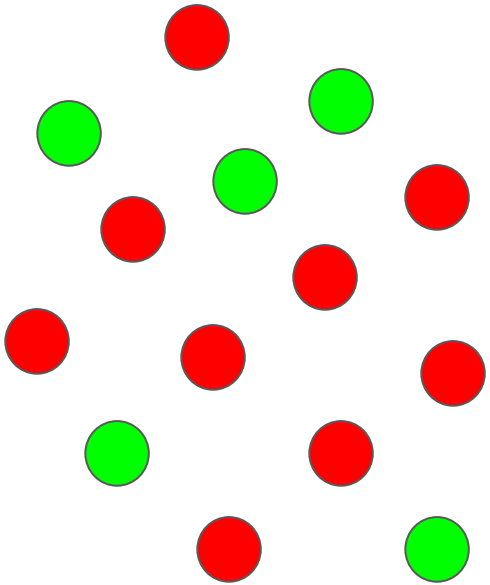
Clients



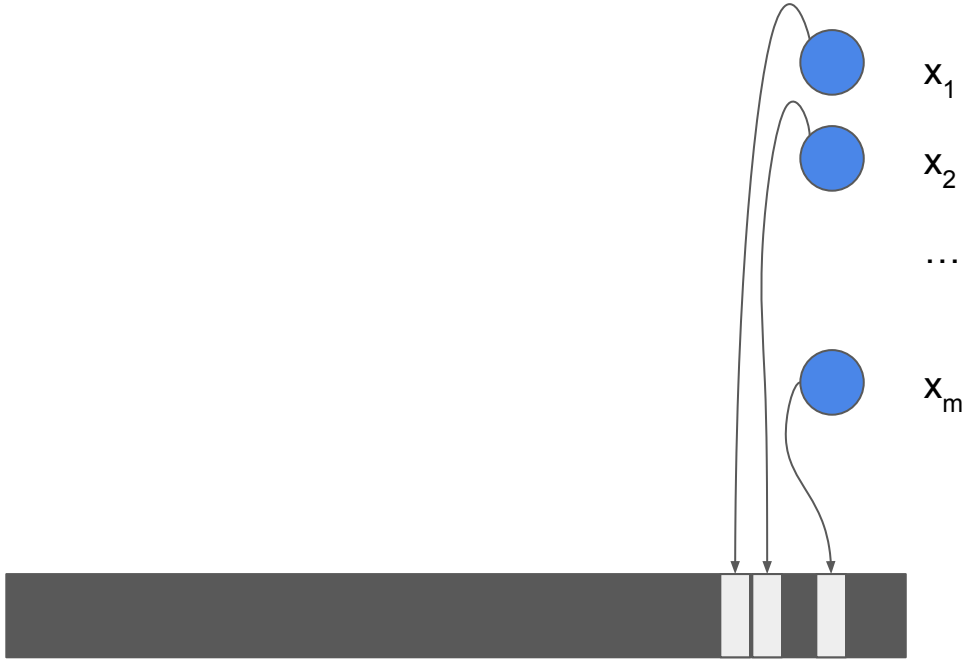
Append-only Bulletin Board

SCALES - the model

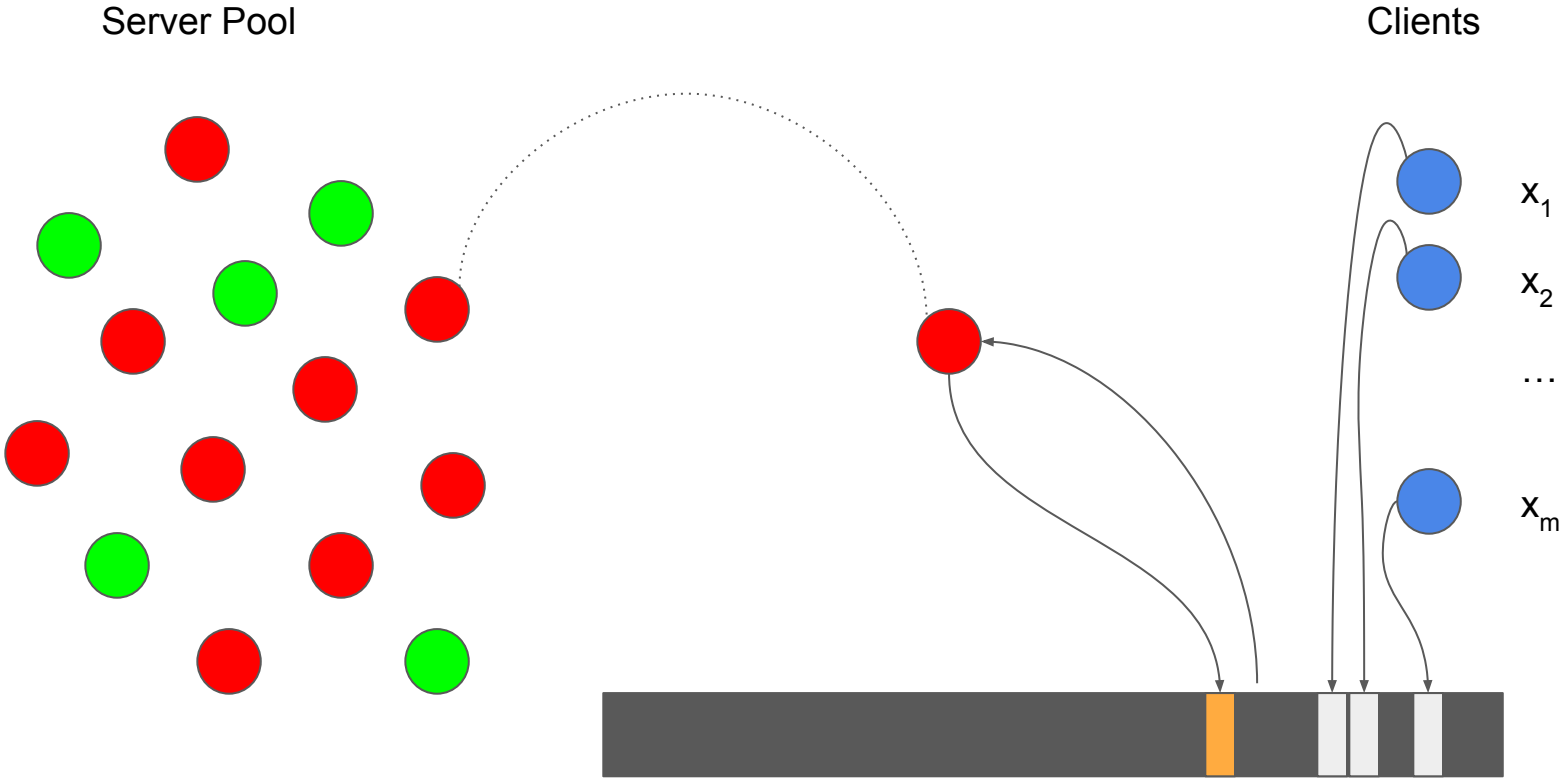
Server Pool



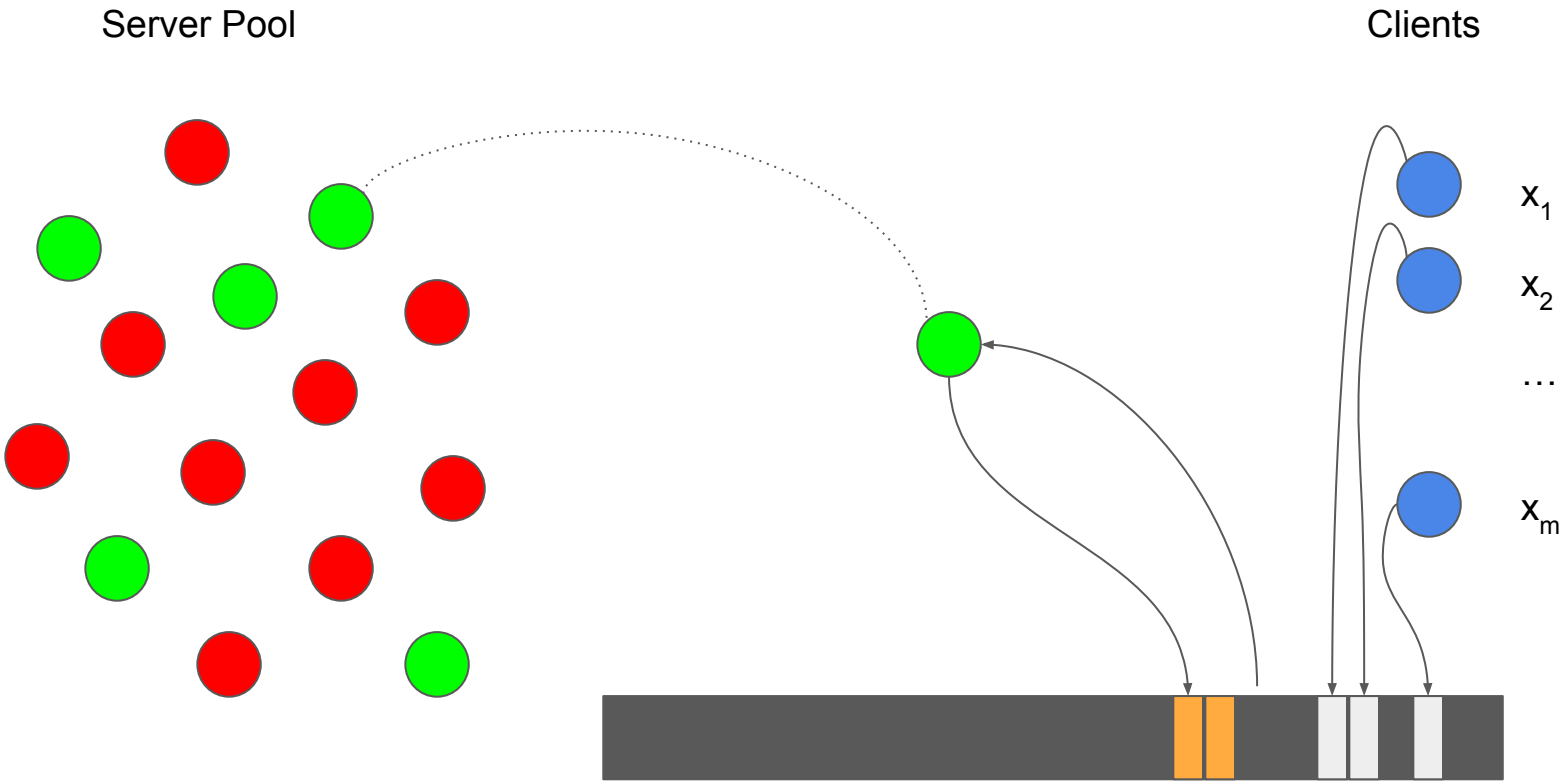
Clients



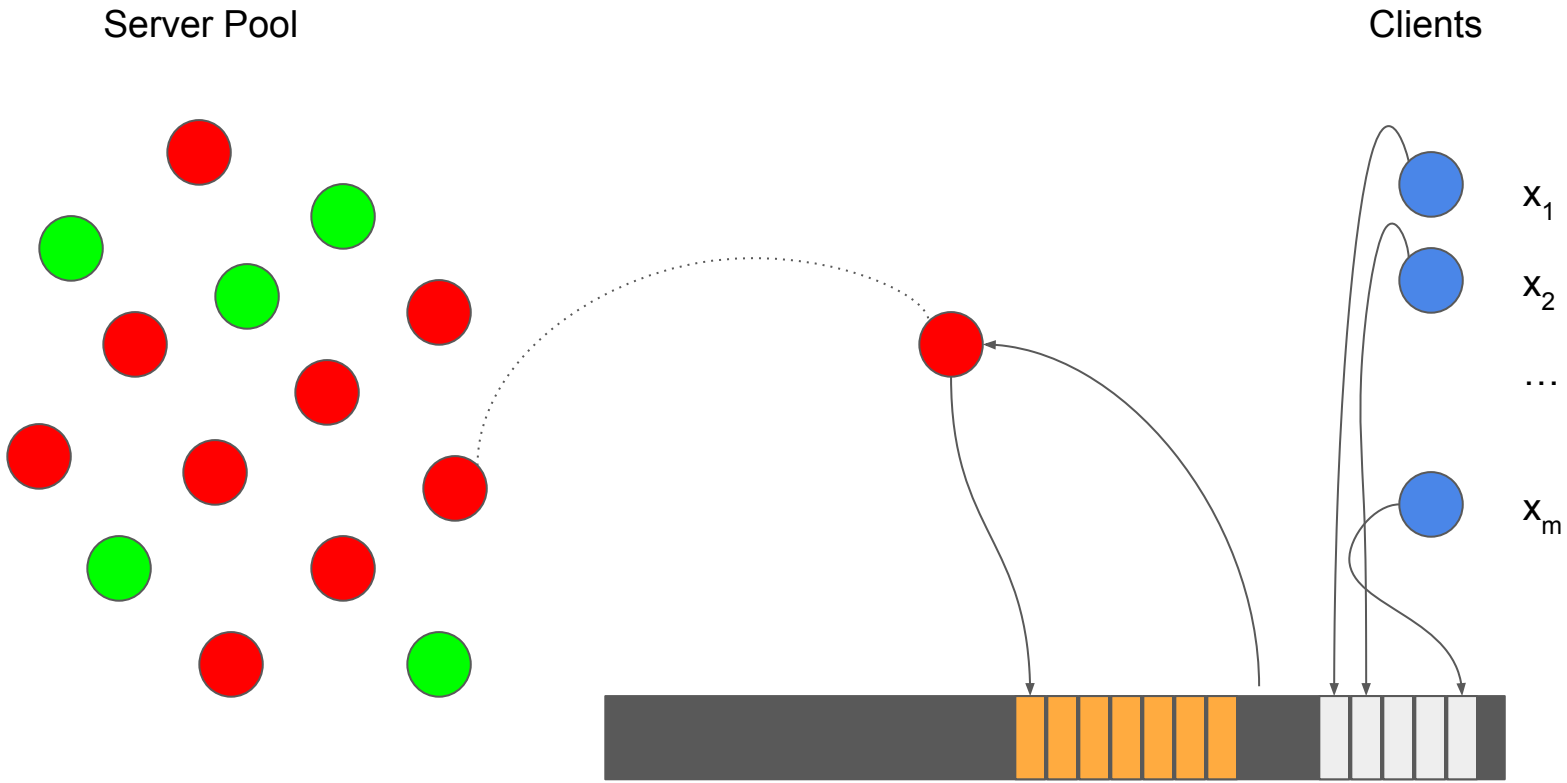
SCALES - the model



SCALES - the model



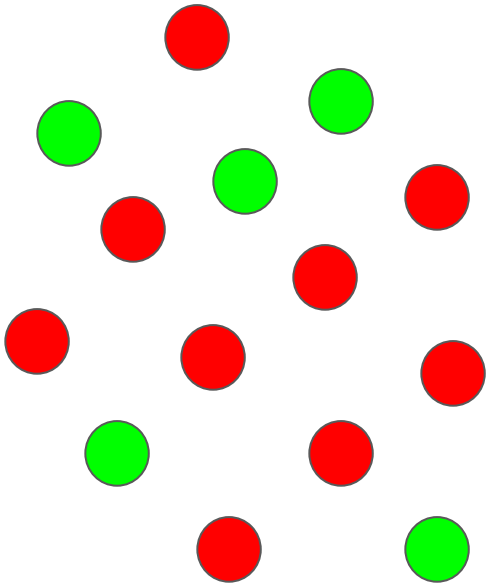
SCALES - the model

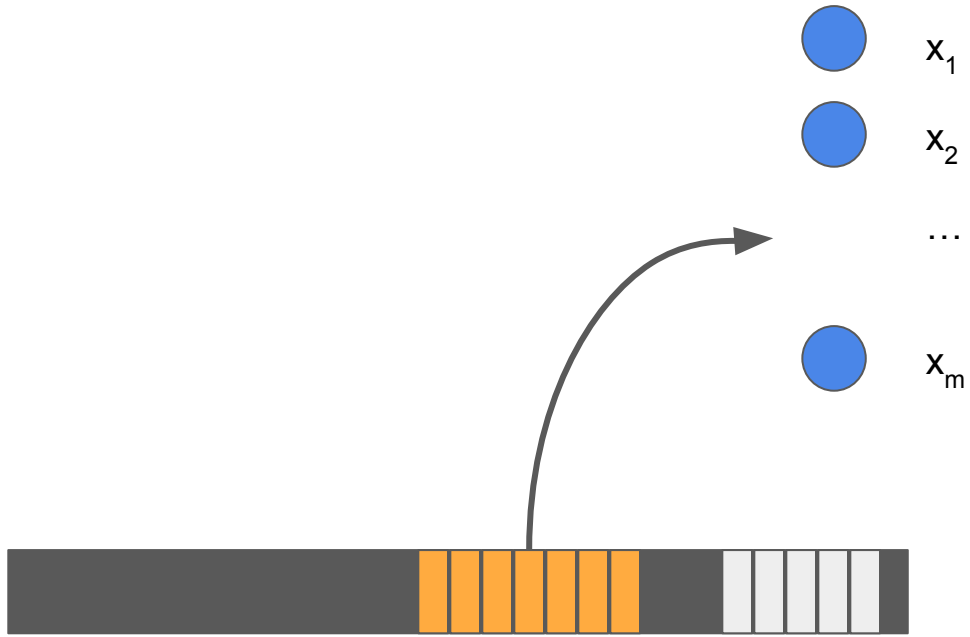


SCALES - the model

Server Pool

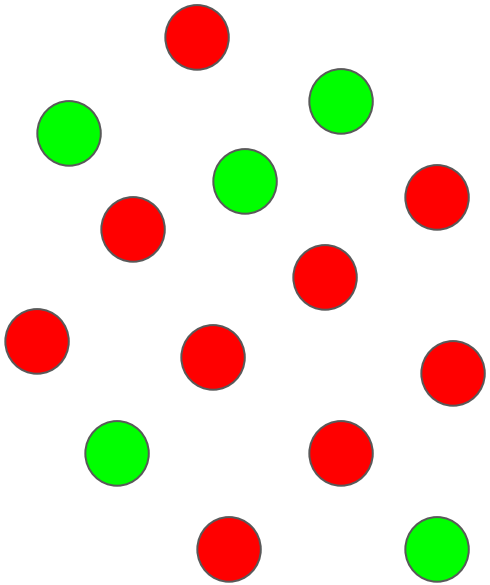
Clients



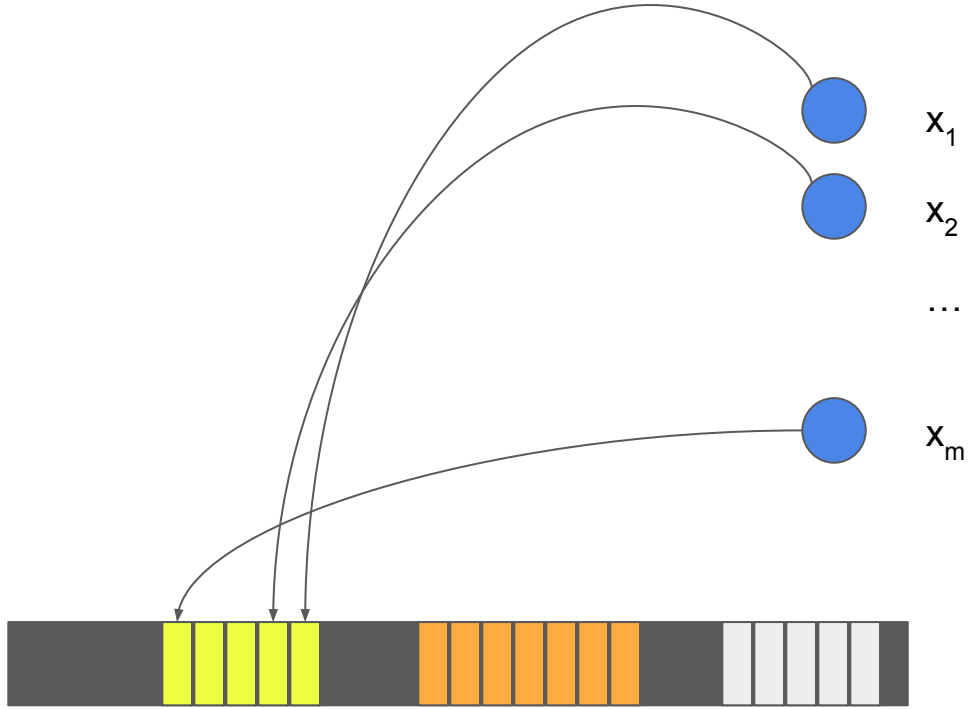


SCALES - the model

Server Pool

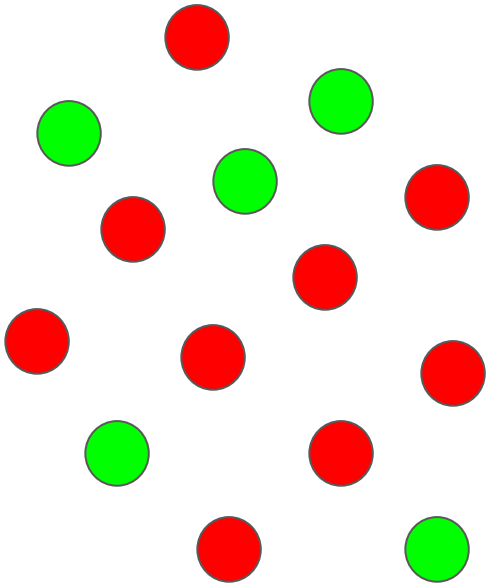


Clients

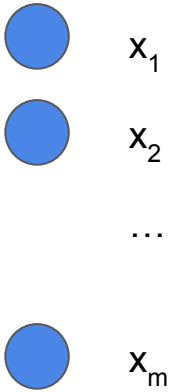


SCALES - the model

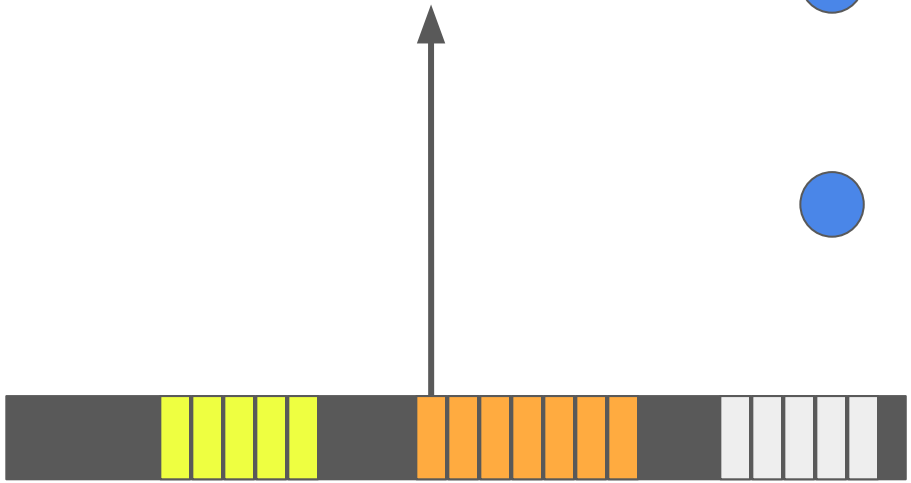
Server Pool



Clients



public decoding: f(x)



SCALES - the model

Semi-Honest Security

Servers	Clients
YOSO-style Adaptive Corruption All-but-one Corruption (dishonest majority)	Adaptive Corruption No restrictions (can collude with the servers)

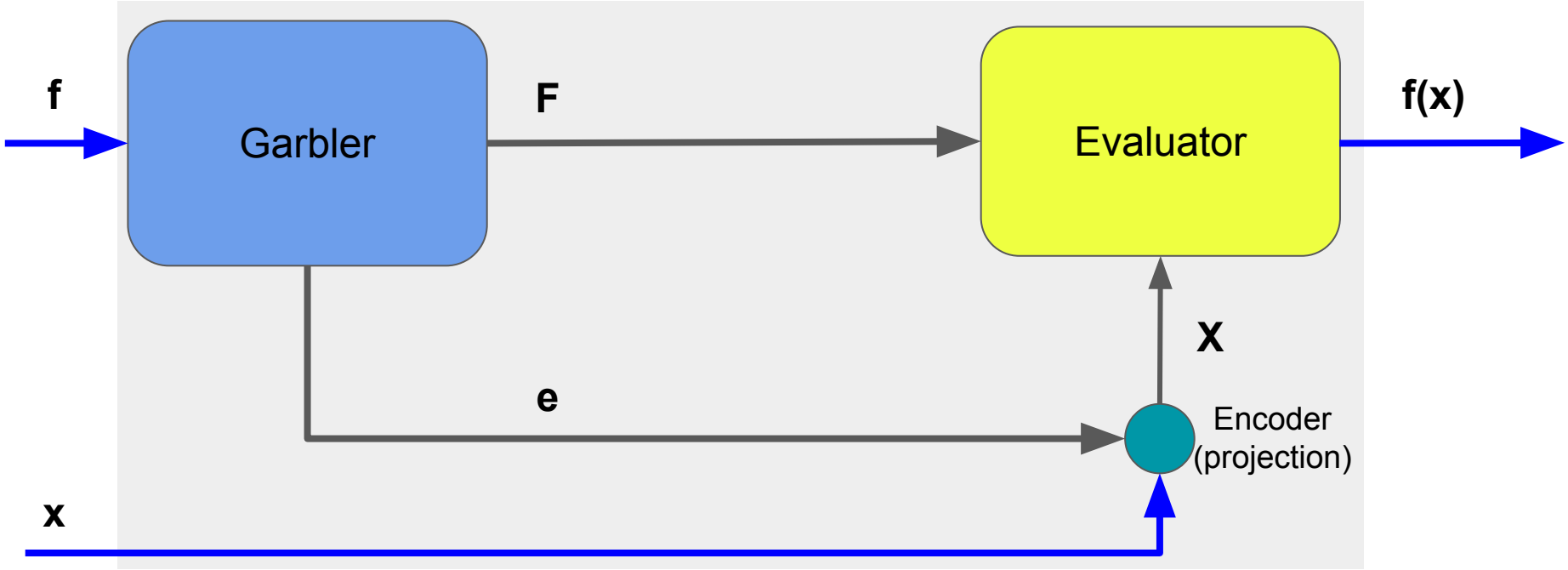
Features of a SCALES Protocol

Ephemeral Servers	Small Clients
One honest server overall Constant number of servers One message per server Compatible with just-in-time random self-selection	Computation proportional to its own input size and number of servers – independent of the full circuit or number of clients Can dynamically control when the protocol ends (by choosing when to post the second message)
Public Output Decoding	

Outline

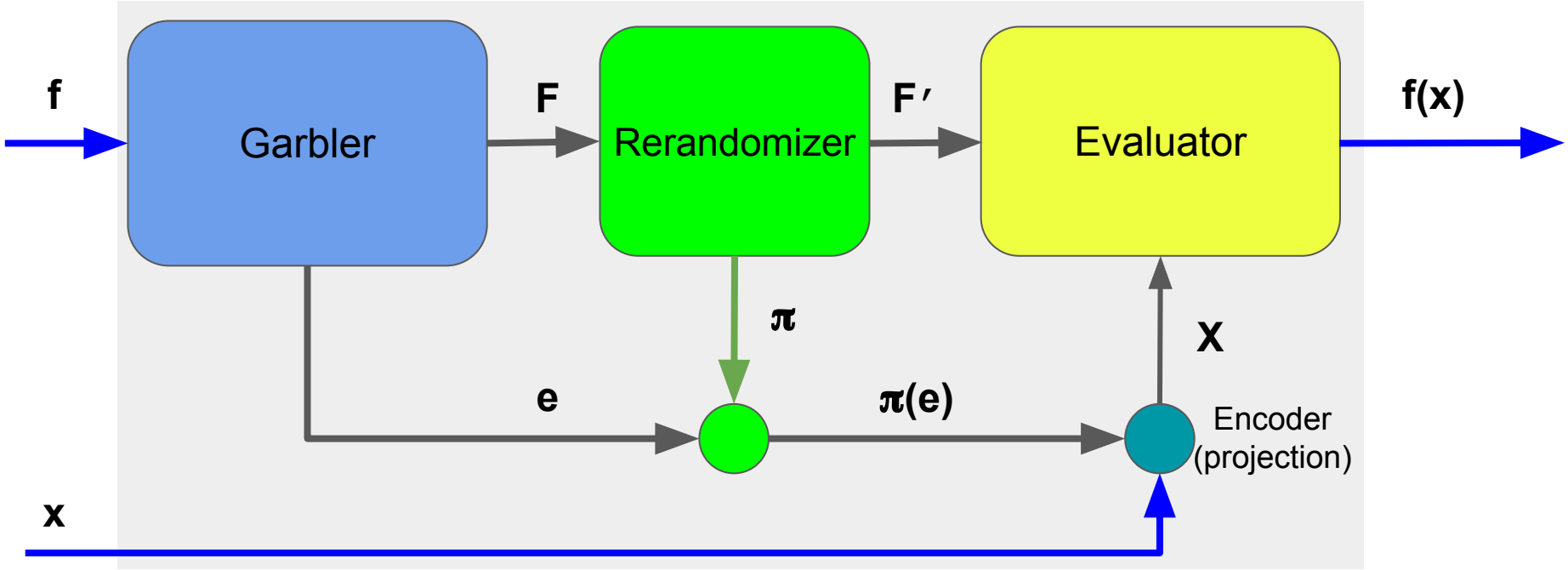
- SCALES
- **Rerandomizable Garbling Schemes**
- Construction - RGCs
- A SCALES protocol

Rerandomizable Garbling Schemes [BHR12]



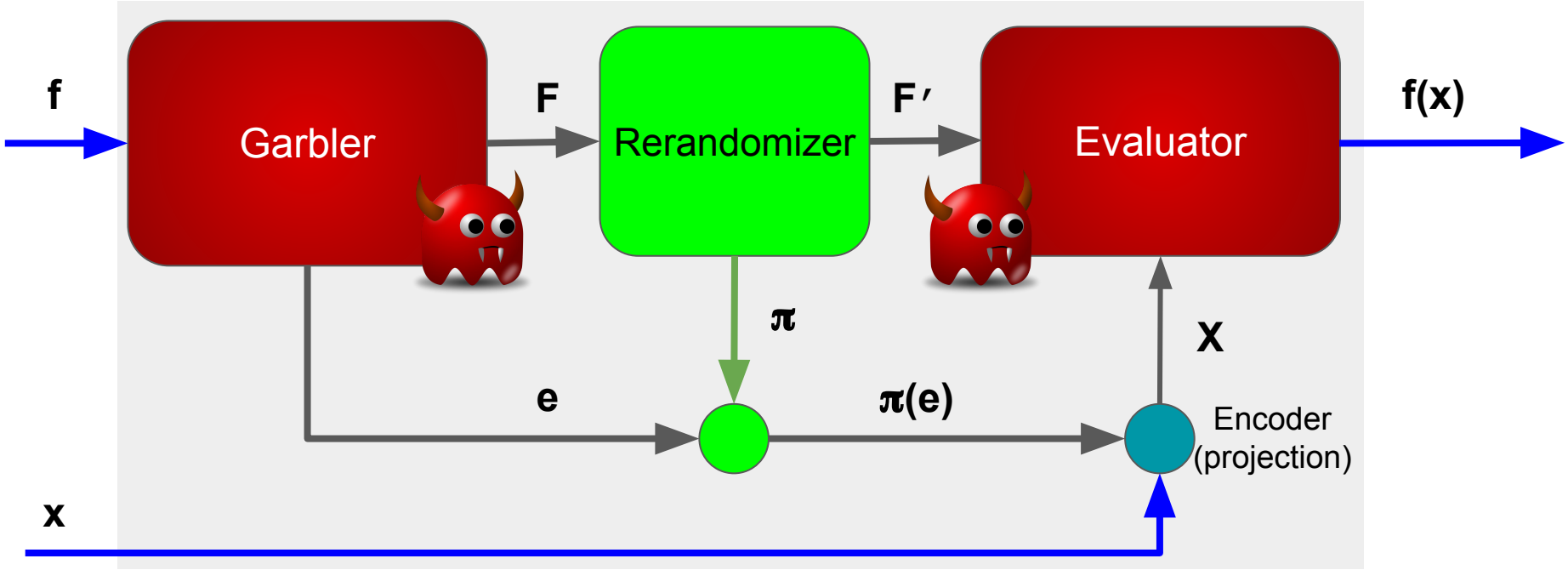
Rerandomizable Garbling Schemes

What we want:



Rerandomizable Garbling Schemes

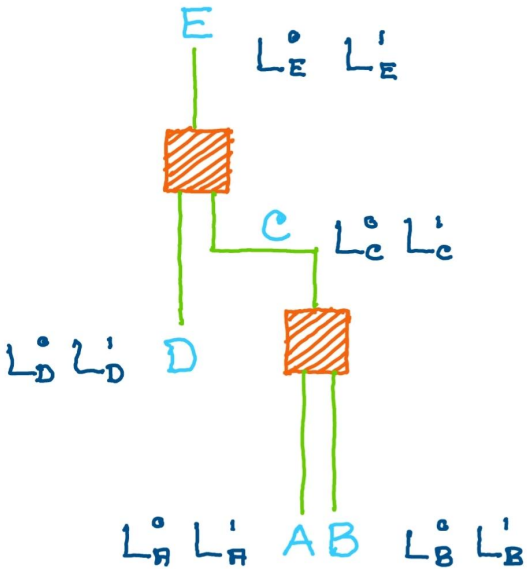
What we want:



Outline

- SCALES
- Rerandomizable Garbling Schemes
- **Construction - RGCs**
- A SCALES protocol

Rerandomizing a GC [GHV10]



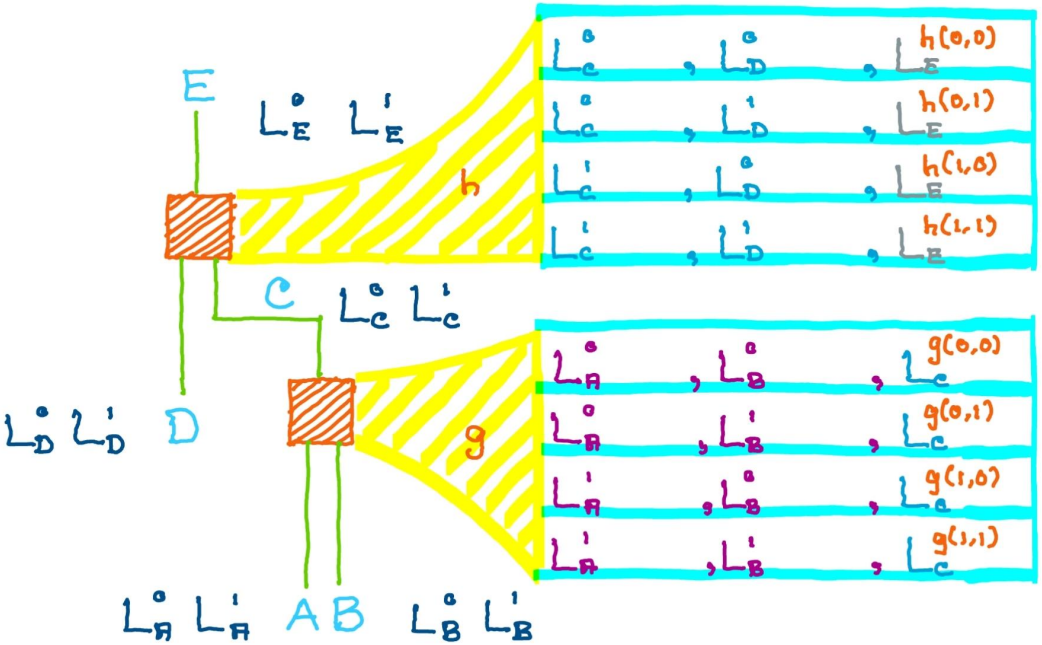
A
L^0_A
L^1_A

B
L^0_B
L^1_B

D
L^0_D
L^1_D

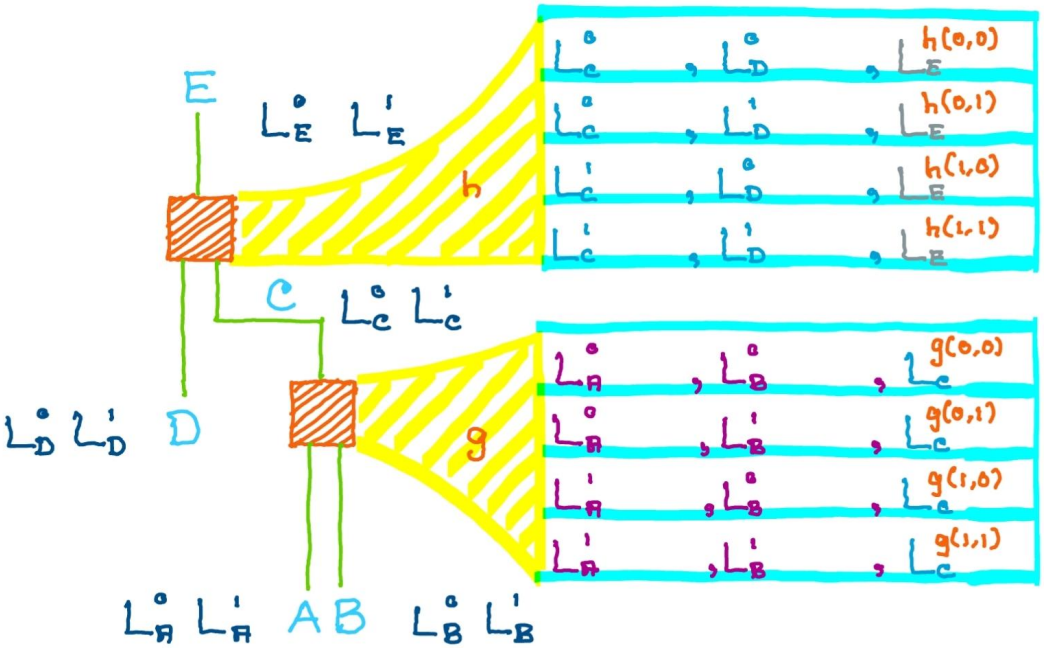
Rerandomizing a GC [GHV10]

Recall: Each ciphertext in a garbled gate encrypts an output wire label under two input wire labels

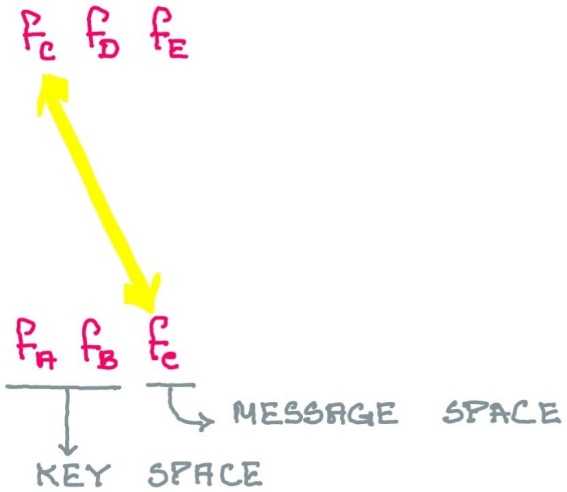


To randomize: use Homomorphic Encryption

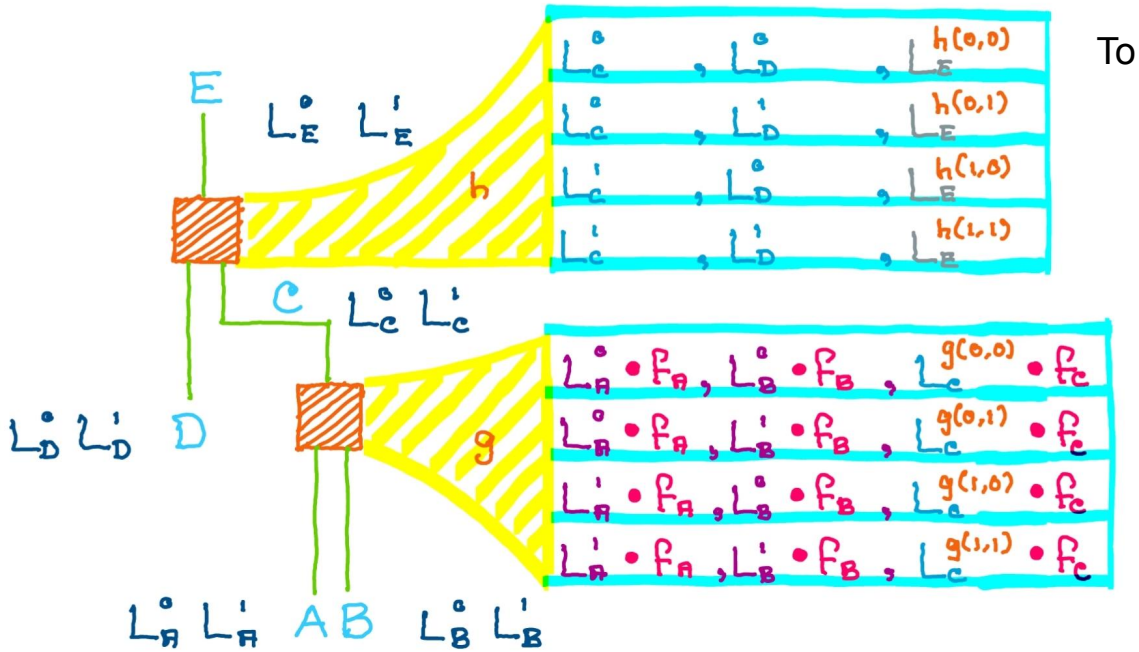
Rerandomizing a GC [GHV10]



To randomize: use Homomorphic Encryption



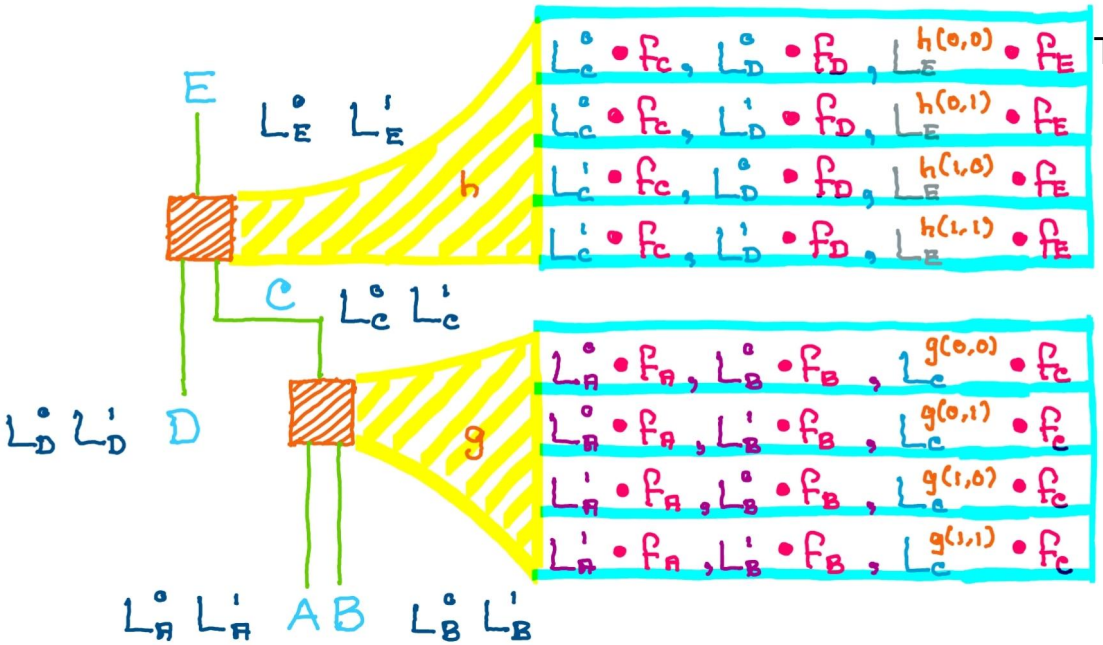
Rerandomizing a GC [GHV10]



To randomize: use Homomorphic Encryption

Transform key and message:
Key & Message Homom. Enc.

Rerandomizing a GC [GHV10]

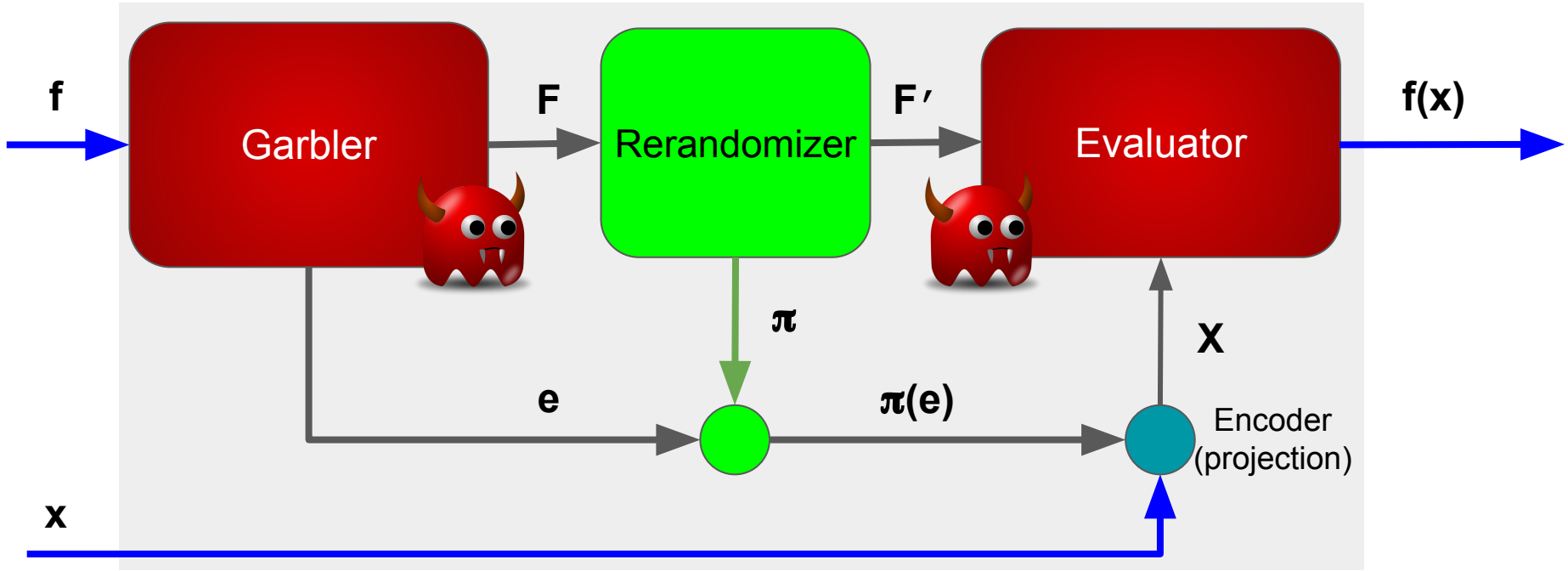


To randomize: use Homomorphic Encryption

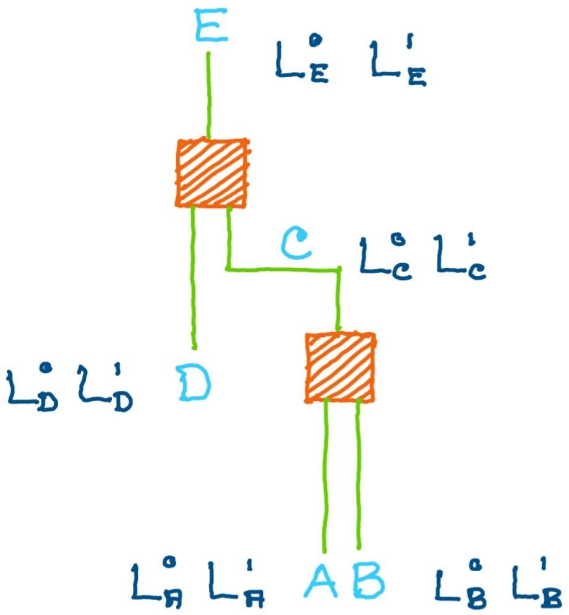
Transform key and message:
Key & Message Homom. Enc.

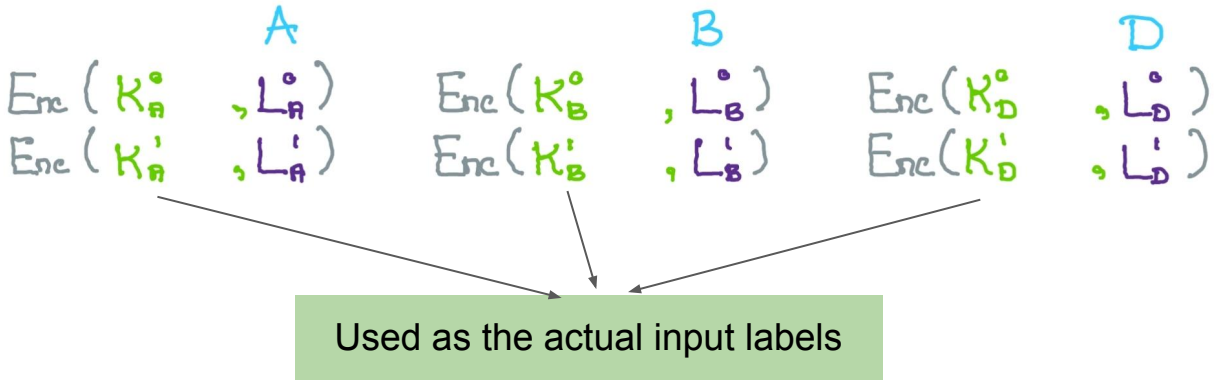
Rerandomizable Garbling Schemes

Also need to specify how input labels are transformed (π)

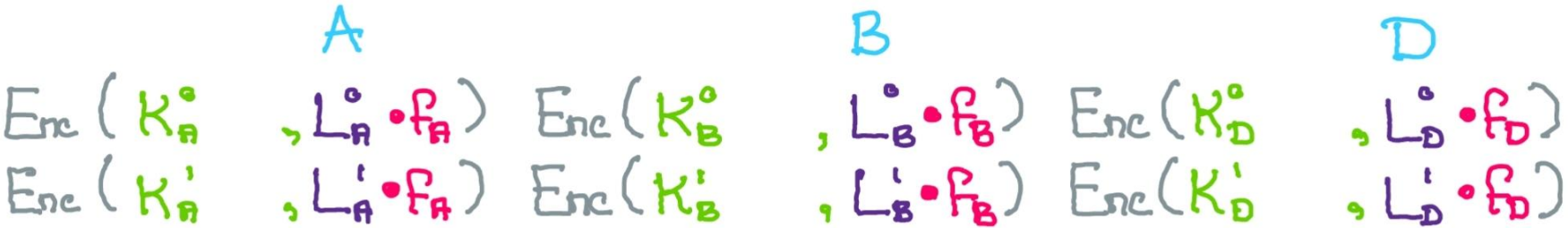


Input Label Transformation

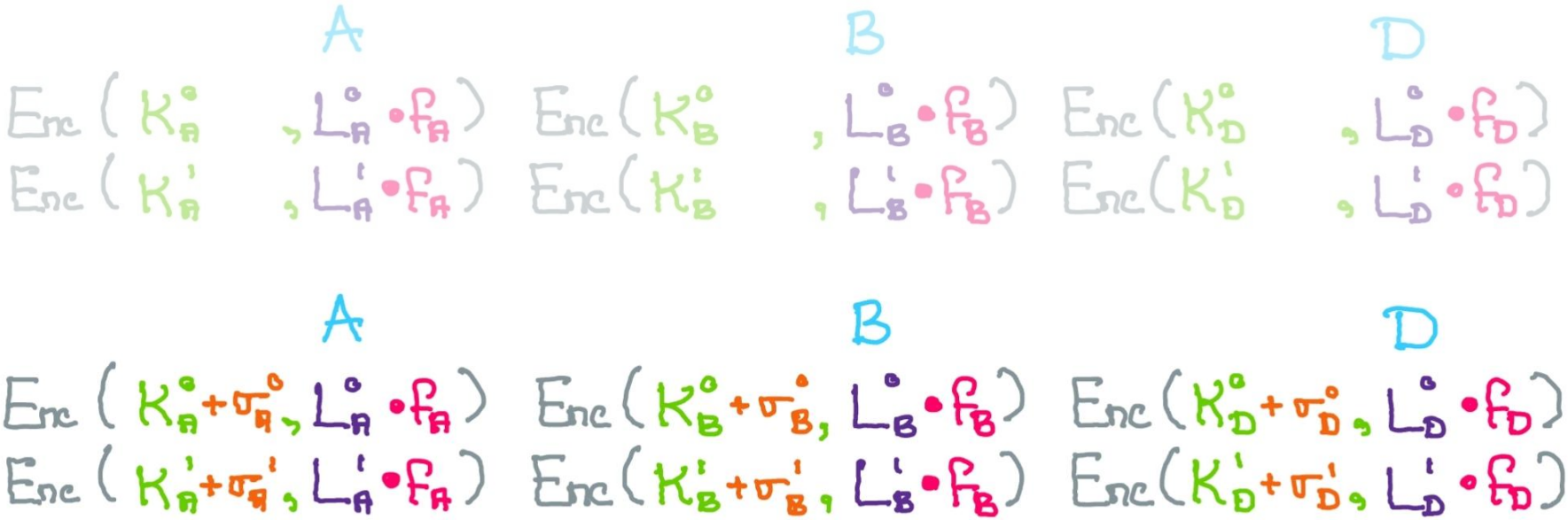




Input Label Transformation



Input Label Transformation

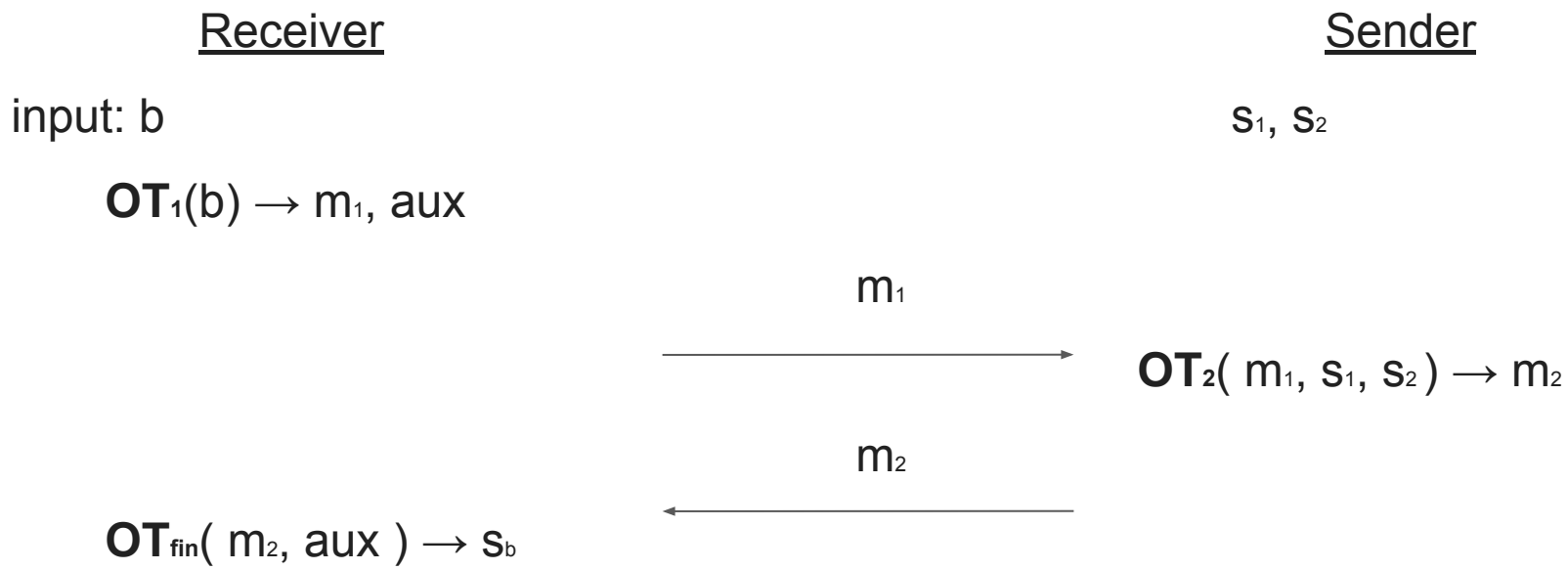


Outline

- MPC with Ephemeral Servers
- Rerandomizable Garbling Schemes
- Construction - RGCs
- **A SCALES protocol**

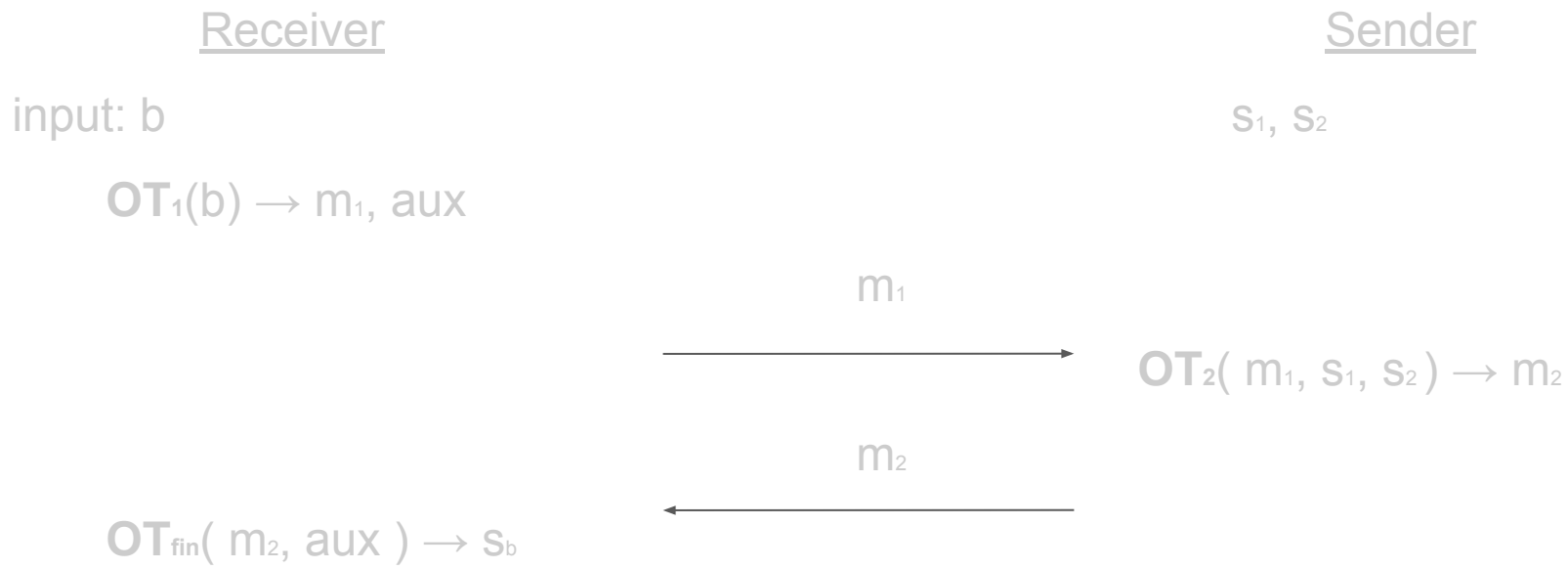
From RGS to SCALES

Building block 1: 2-round OT



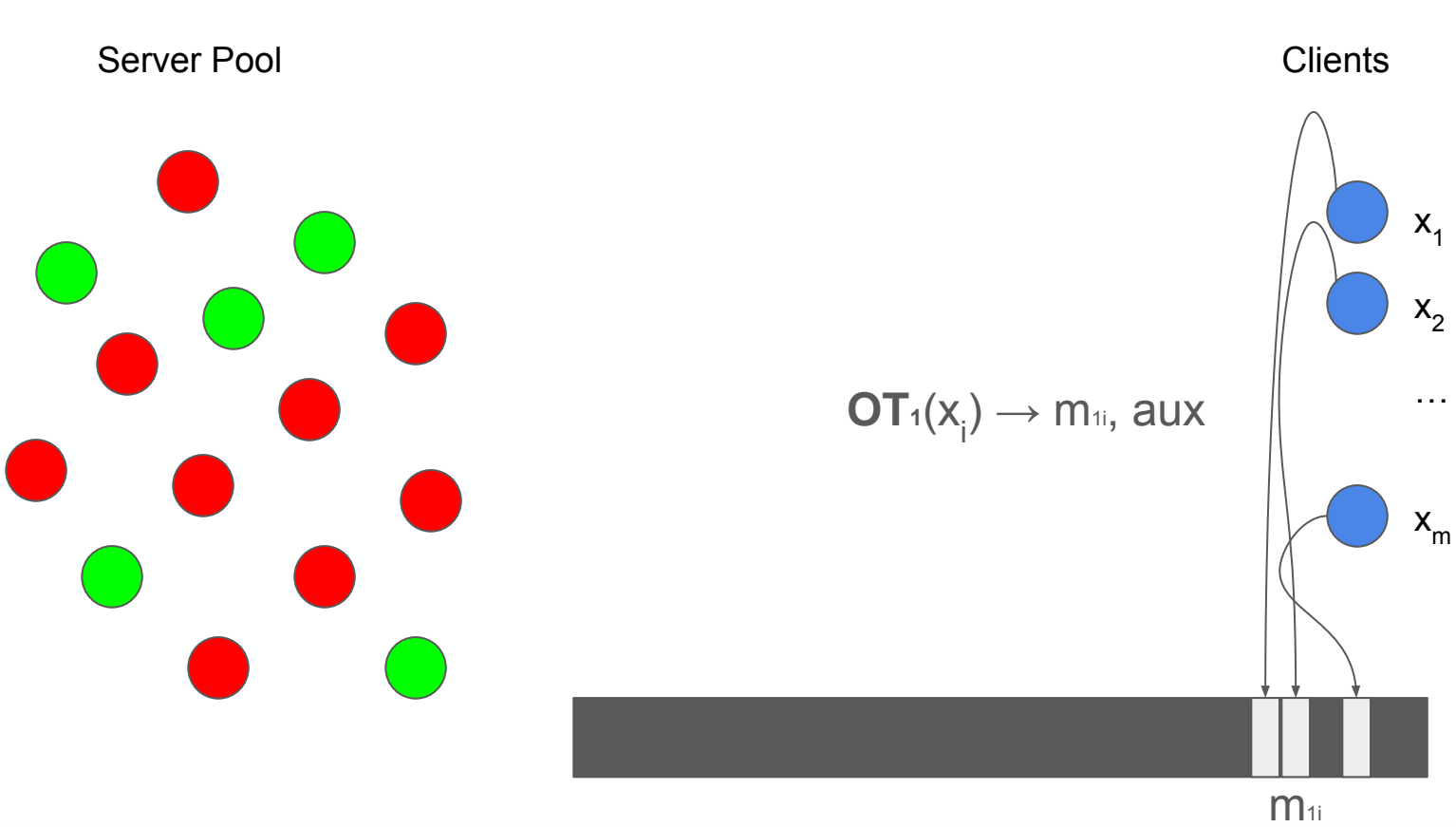
From RGS to SCALES

Building block 1: 2-round OT

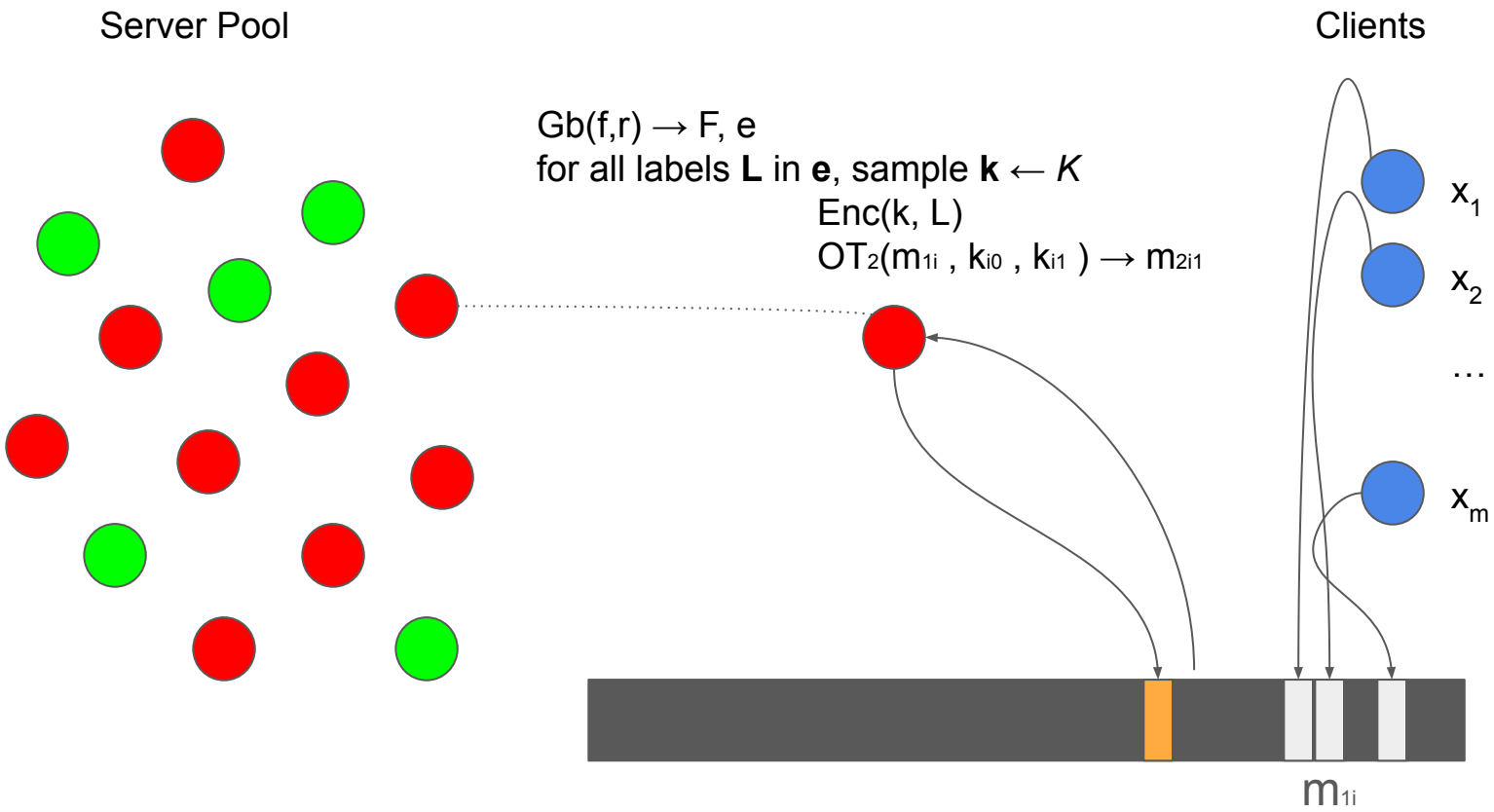


Building block 2: KMHE

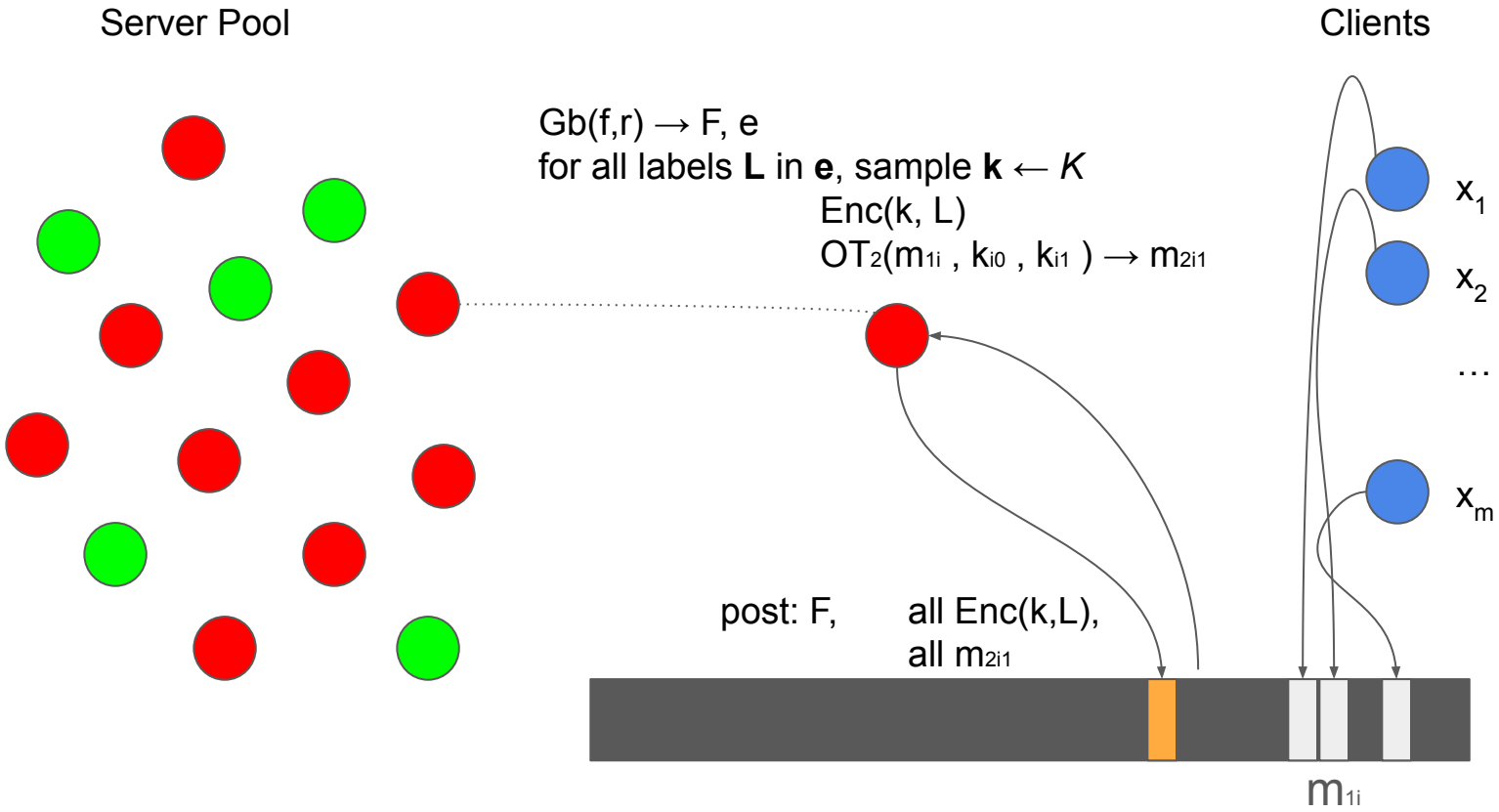
From RGS to SCALES



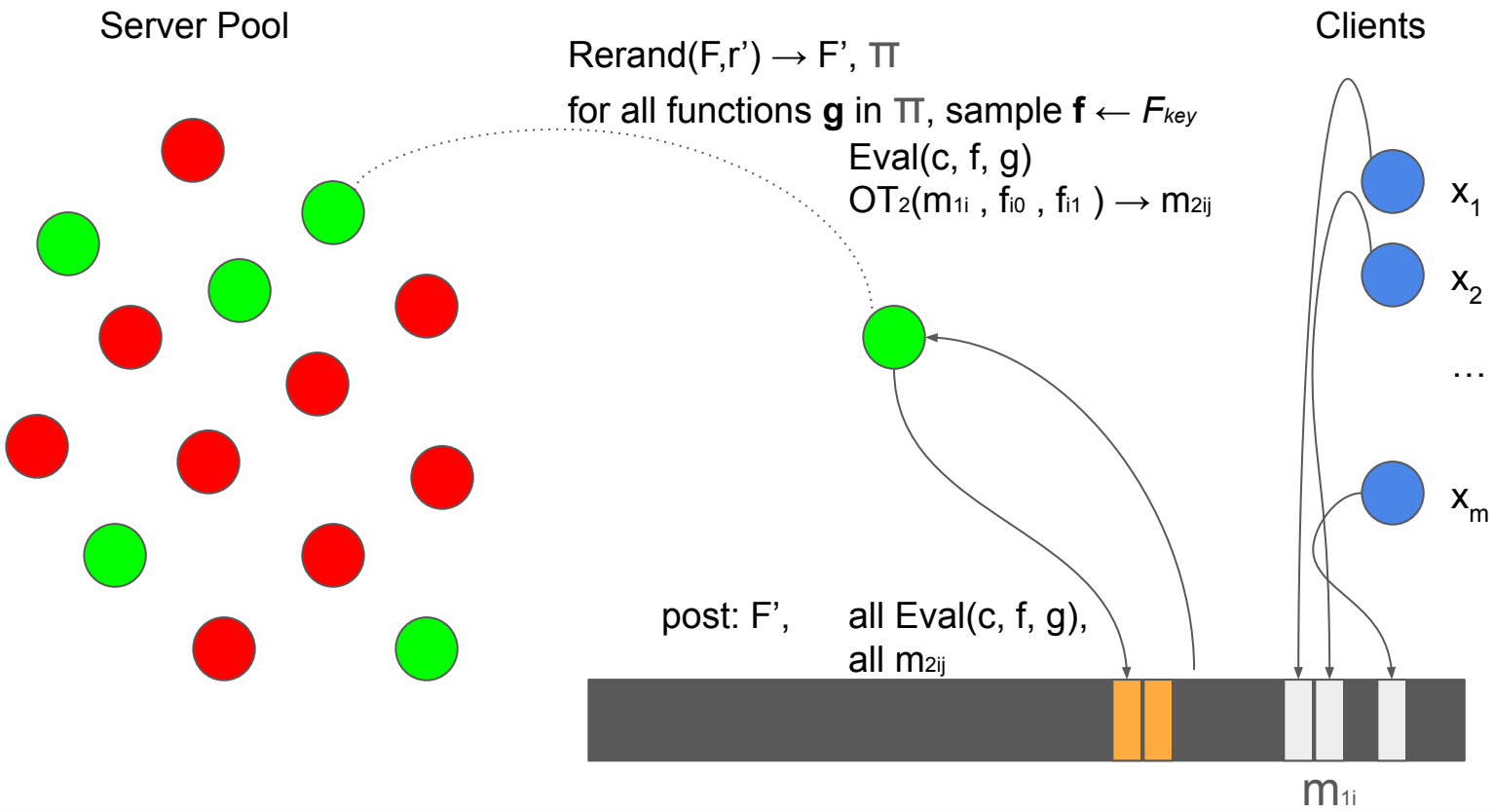
From RGS to SCALES



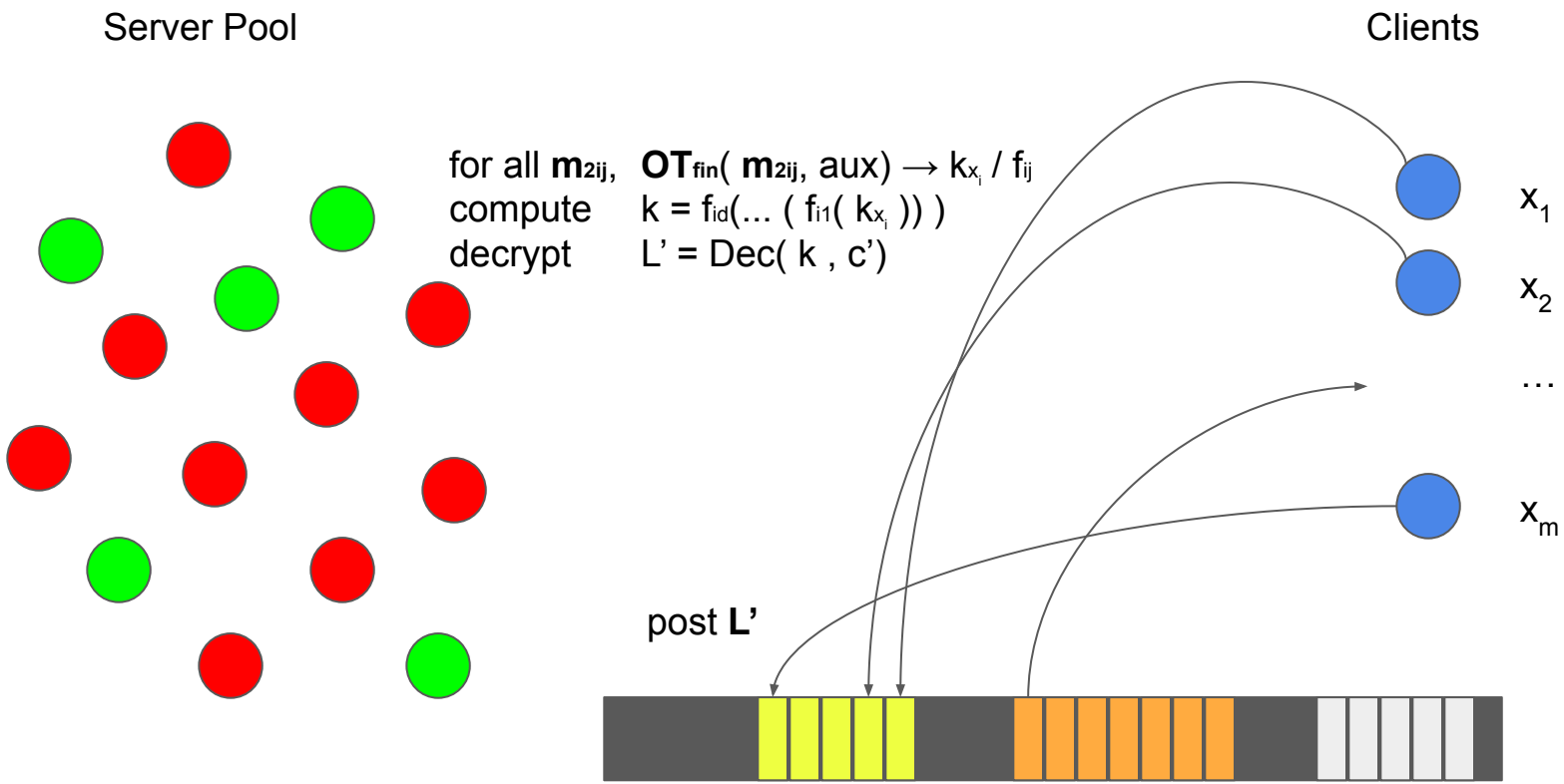
From RGS to SCALES



From RGS to SCALES

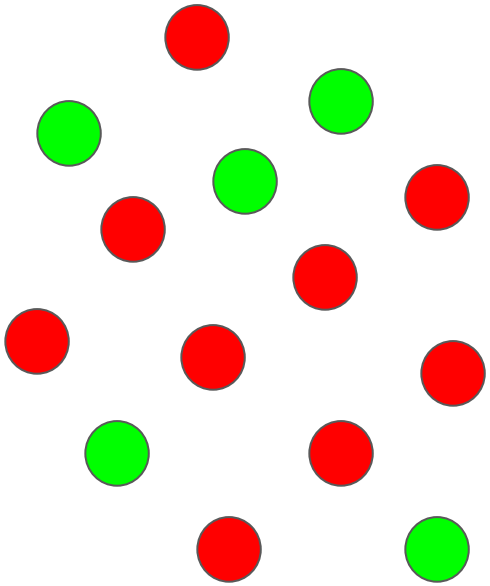


From RGS to SCALES



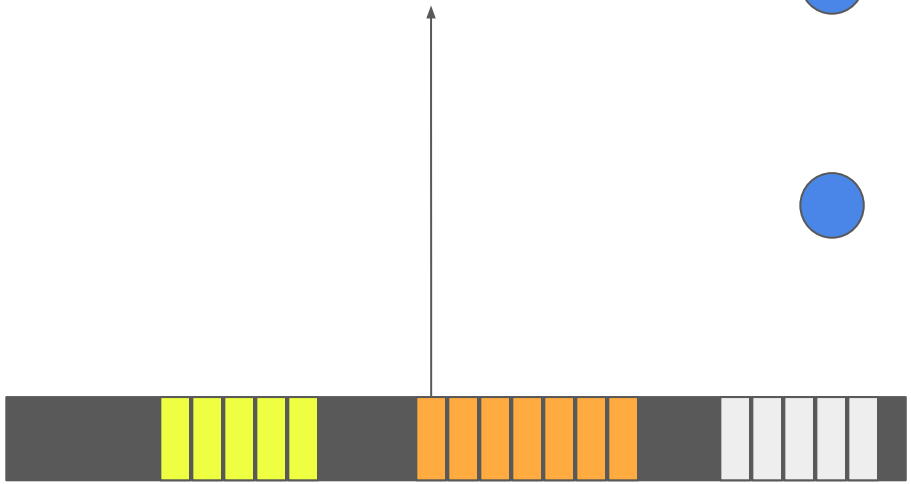
From RGS to SCALES

Server Pool

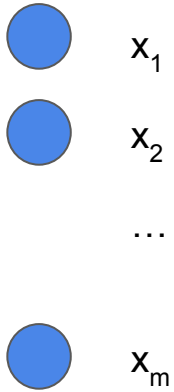


public decoding:

using **X' = { L' }** and final **F'**,
Ev(X',F') = f(x)



Clients



Performance Comparison

	SCALES	YOSO
Setup	Bulletin-Board	Target-Anonymous Channels, PKI
Communication: clients servers number of rounds	 2 1 constant	 1 1 computation size
Computation: Clients servers	 number of servers computation size	 committee size committee size
Corruption (adaptive): servers clients	 all-but-one arbitrary	 honest majority in each committee arbitrary

Open Problems

Better KMHE schemes

Stronger security (malicious, GOD)

Sublinear RCS

Information theoretic iDRE