MUSE: Secure Inference Resilient to Malicious Clients

Ryan Lehmkuhl
UC Berkeley

Pratyush Mishra
UC Berkeley

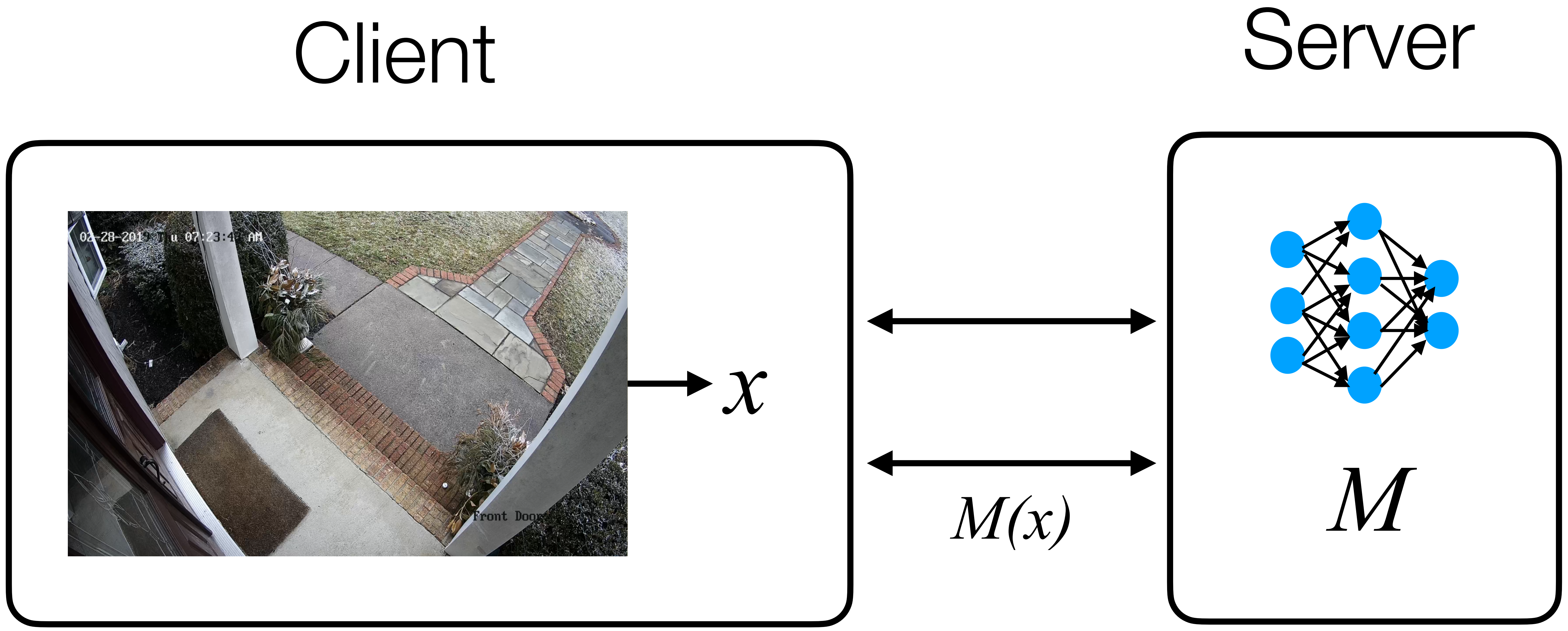
Akshayaram Srinivasan
Tata Institute of Fundamental Research

Raluca Ada Popa
UC Berkeley

Neural Network Inference

A growing number of applications use neural networks in user interactions

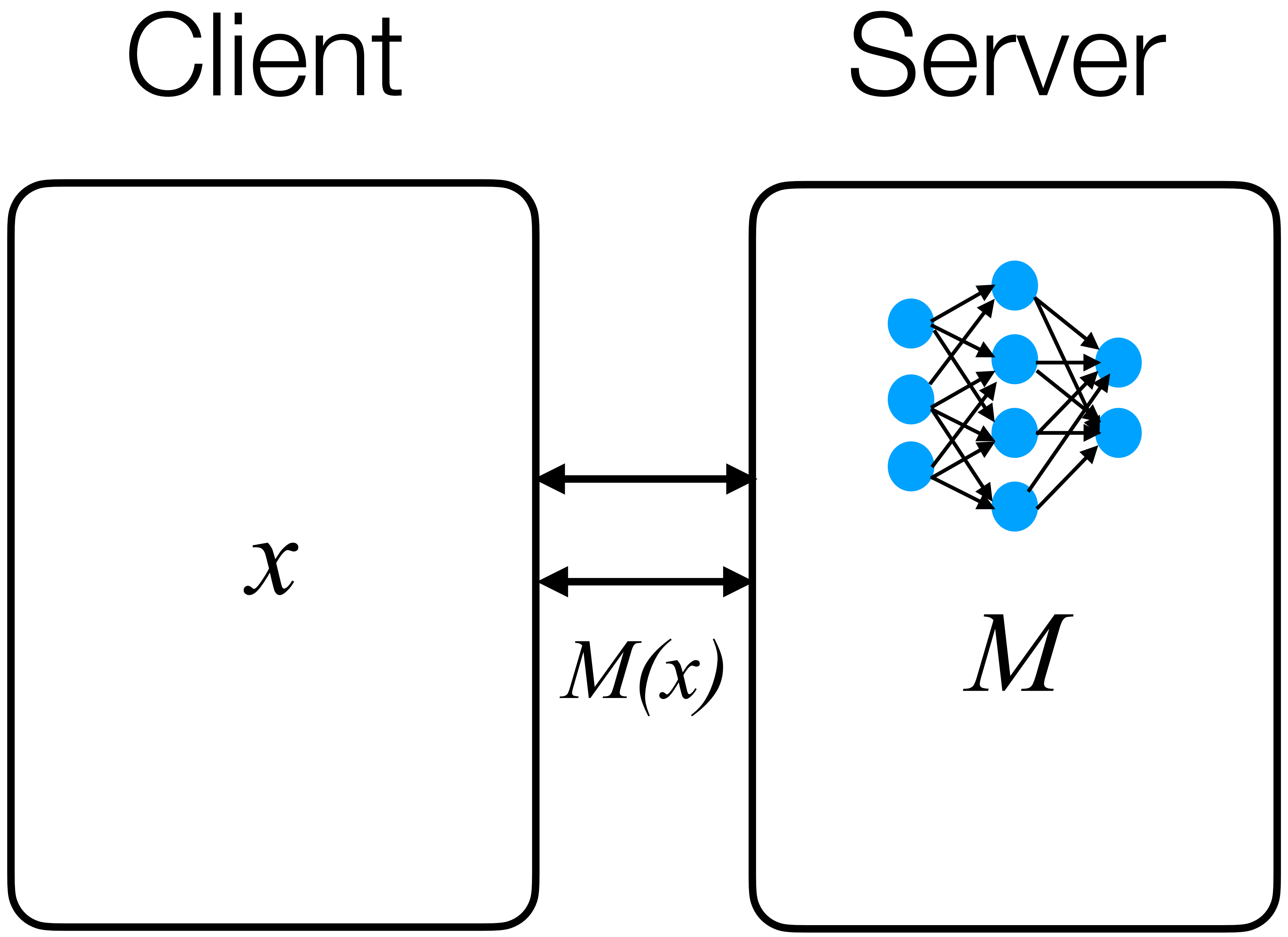
- Baby monitor: motion detection to alert parents
- Home monitoring: detect and recognize visitors



Server's model is proprietary and sensitive

Client data is sensitive

Secure inference



Client (& server) should learn *only* prediction $M(x)$

Server should not learn private client input x

Client should not learn private model weights M

Prior work on 2-party secure inference

Semi-honest Security

Malicious Security

Slow (Generic Protocols)

ABY^3

CrypTFlow2

DeepSecure

MiniONN

TAPAS

Overdrive

Ponytail

CryptoNets

LoLa

Fast (Specialized protocols)

Marbled Circuits

FHE-DiNN

XONN

?

CHET

CryptoDL

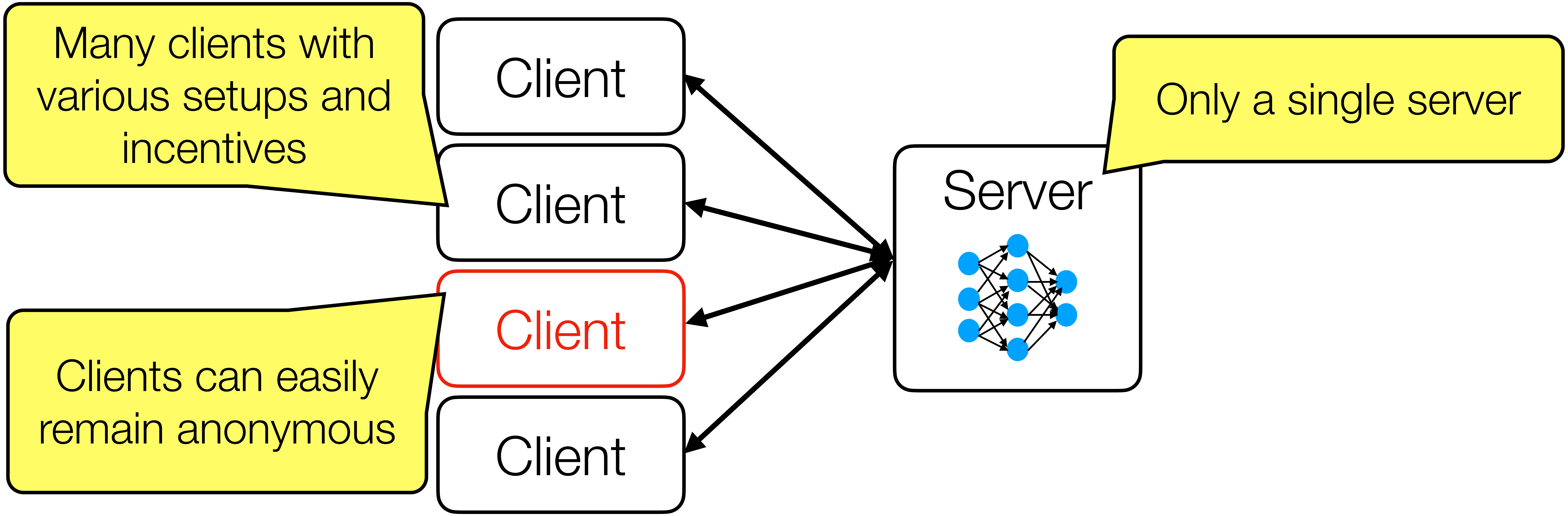
Authenticated Garbling

Gazelle

Delphi

SecureML

The case for client-malicious security



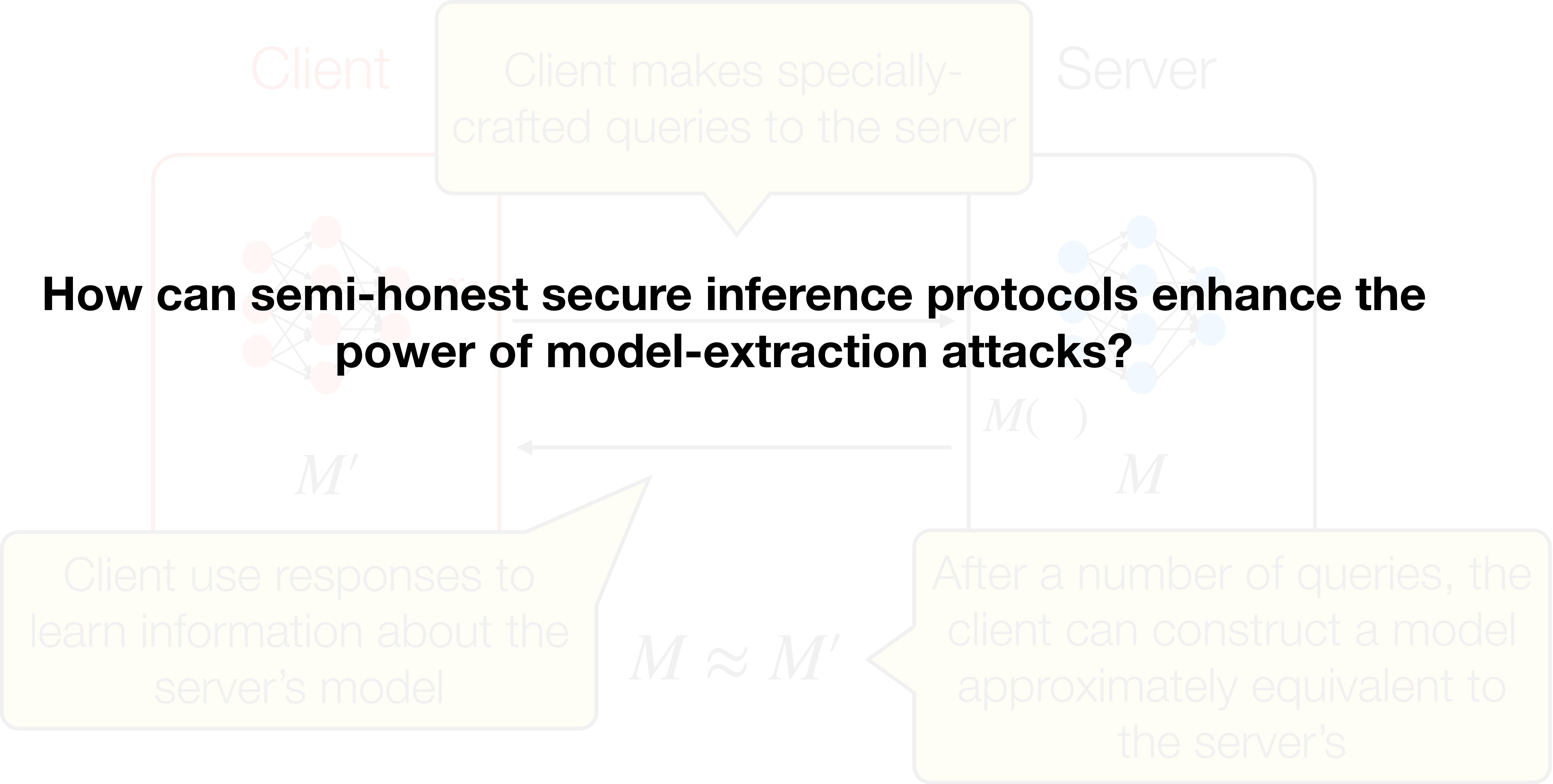
Client-malicious security => semi-honest server, malicious client

Contributions

1) A ***model-extraction attack*** against semi-honest secure inference protocols

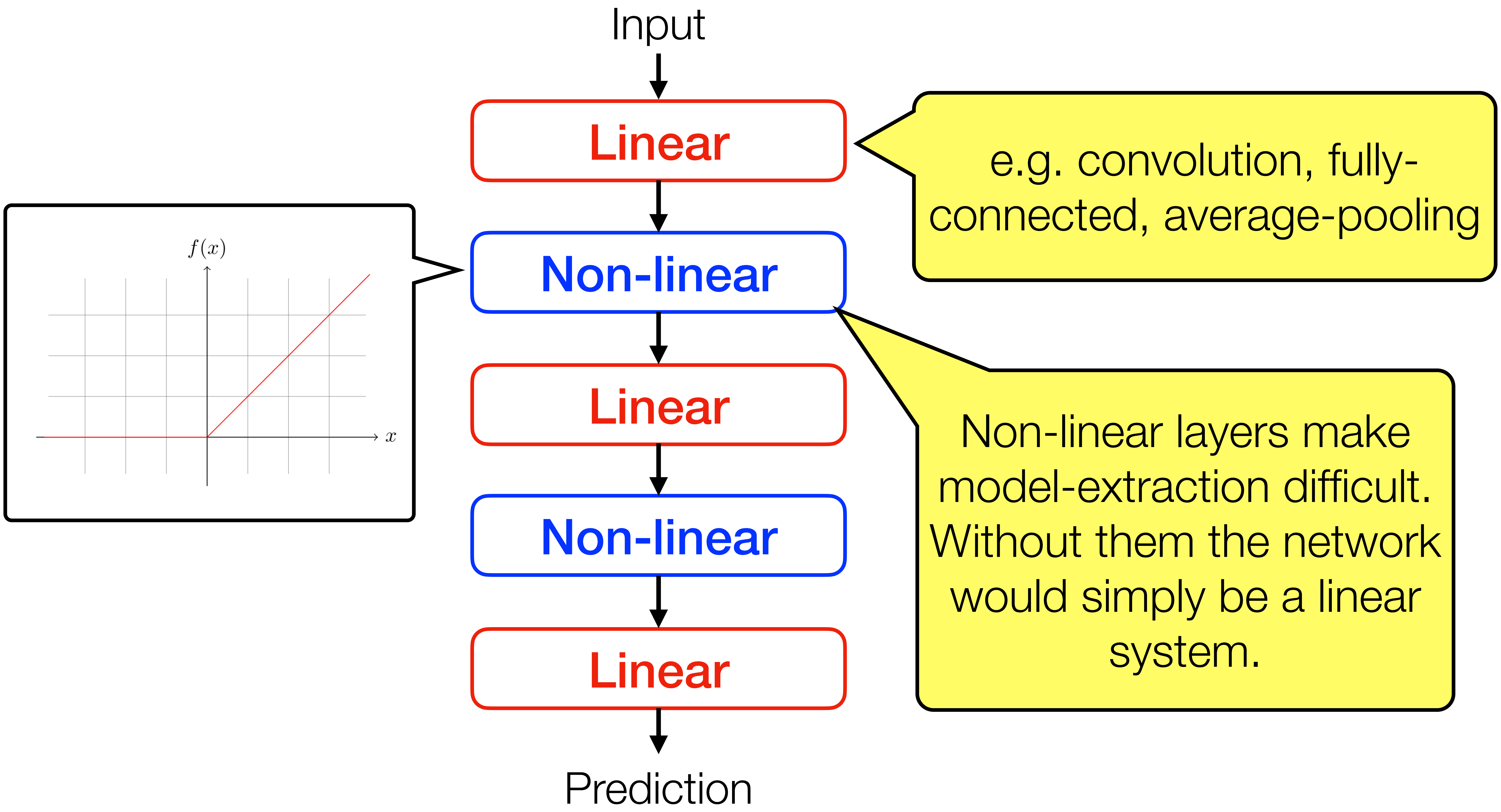
2) **Muse**: An efficient ***client-malicious*** secure inference protocol

Model-extraction attacks



How can semi-honest secure inference protocols enhance the power of model-extraction attacks?

Recap: Neural Networks



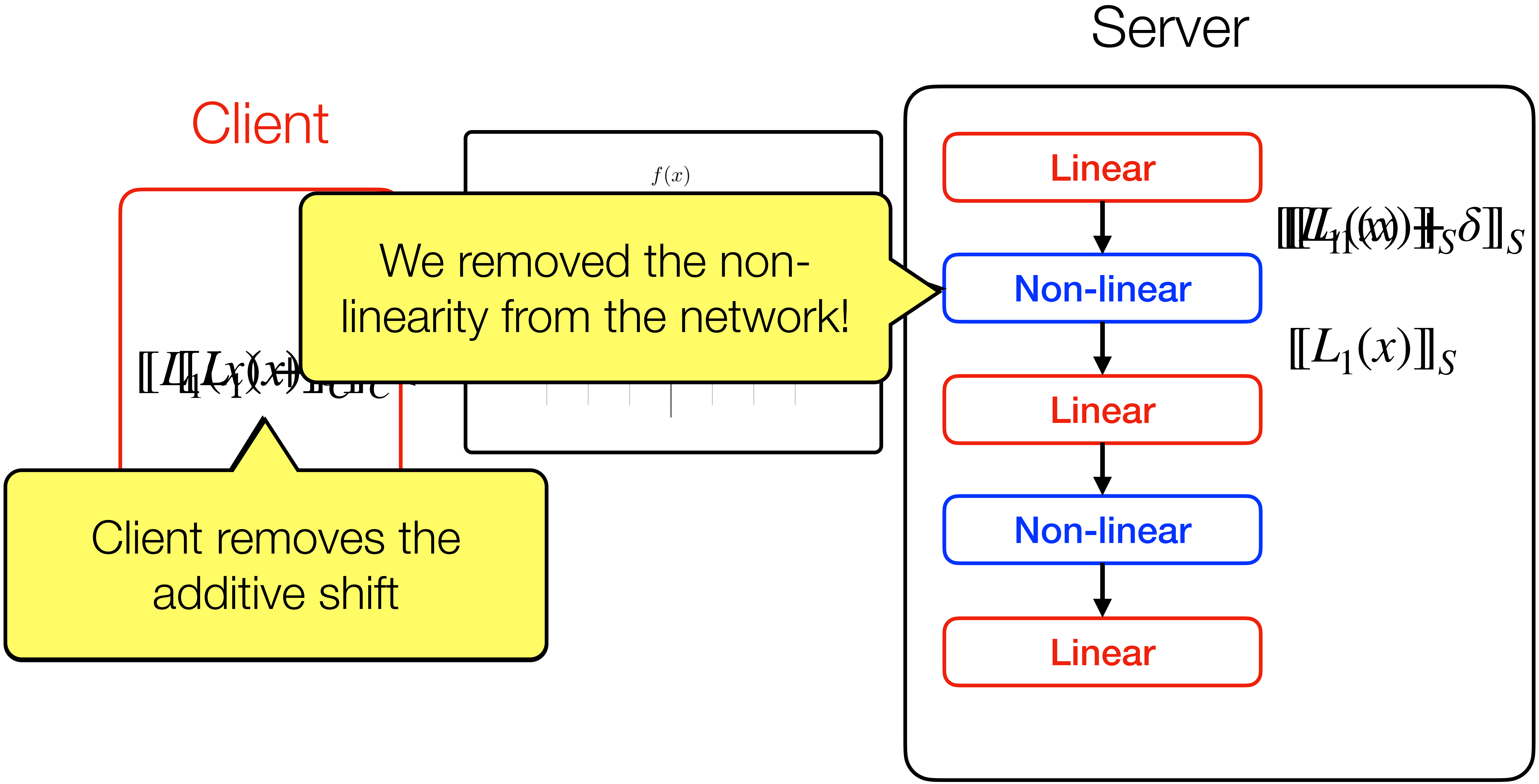
Semi-honest secure inference protocols based on additive secret-sharing

1) Compared to standard inference, secure inference has $O(\ell)$ additional rounds of interaction

2) A malicious client can shift intermediate values in the network evaluation

How can a malicious client leverage these two properties?

Model-extraction attack intuition



Evaluating our attack

Compared to the state-of-the-art black-box model extraction attack [Car+20], our attack:

- **Uses 24x-312x fewer queries**
- ***Perfectly* extracts model weights** rather than approximating them
- **Scales on the number of parameters**, *not* the depth of the network
- Evaluated on networks **100x deeper** and with **60x the parameters**

Muse

Cryptographic system for secure inference
on convolutional neural networks

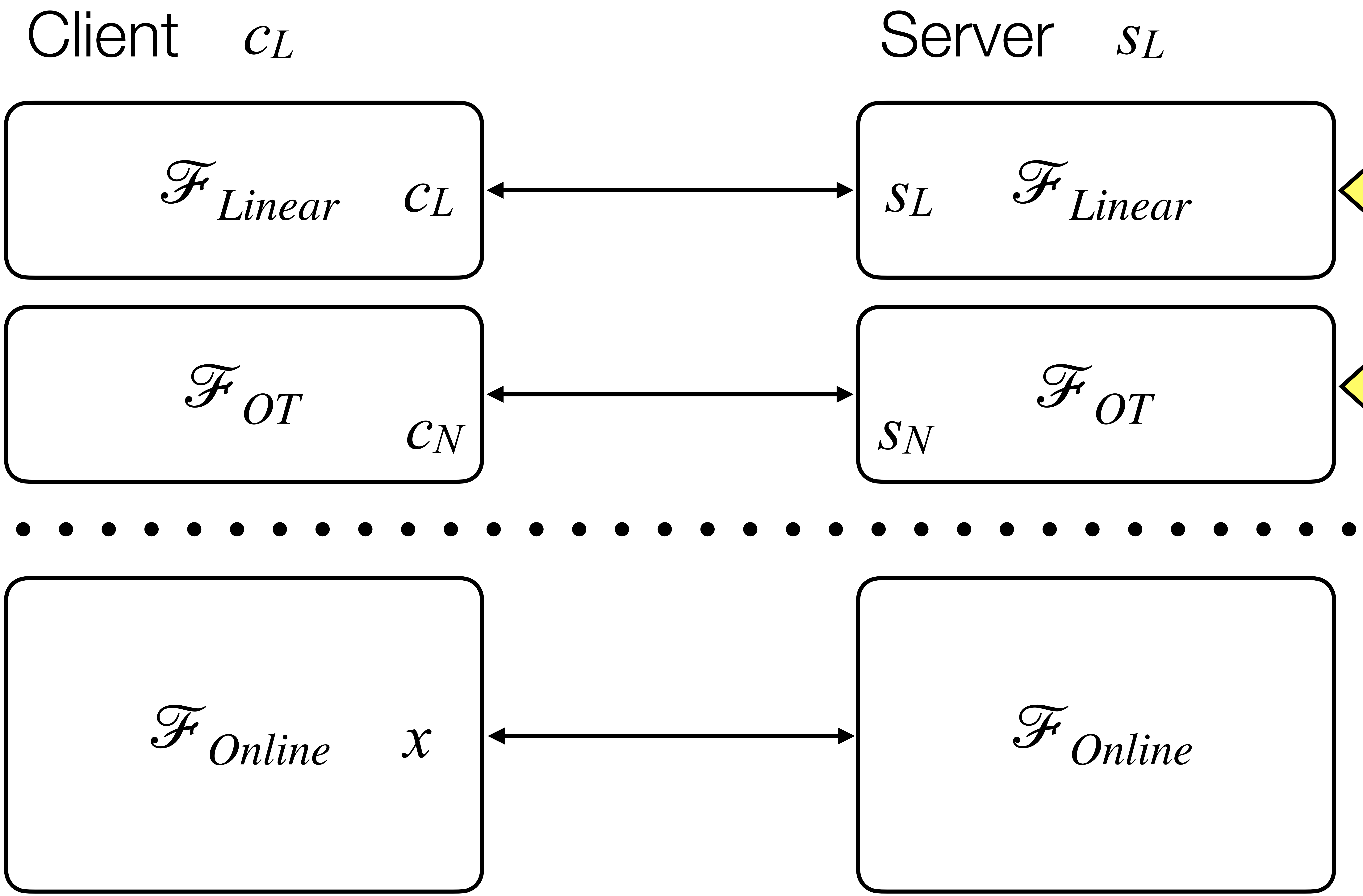
Security: achieves ***client-malicious* simulation-based security**

Functionality: supports **arbitrary ReLU-based CNNs**

Efficiency:

- reduces **bandwidth** (4.6x) and **inference latency** (21x) compared to existing alternatives
- online phase **similar to semi-honest protocols**

Starting point: Delphi [Mis+20]



Uses HE to compute correlated randomness

Server garbles circuit and client obtains labels

Online phase

Extending Delphi to client-malicious security

Client c'_L c'_N

Server s_L s_N

$\mathcal{F}_{Linear}$ $\mathcal{F}_{Linear}$

$\mathcal{F}_{OT}$ $\mathcal{F}_{OT}$

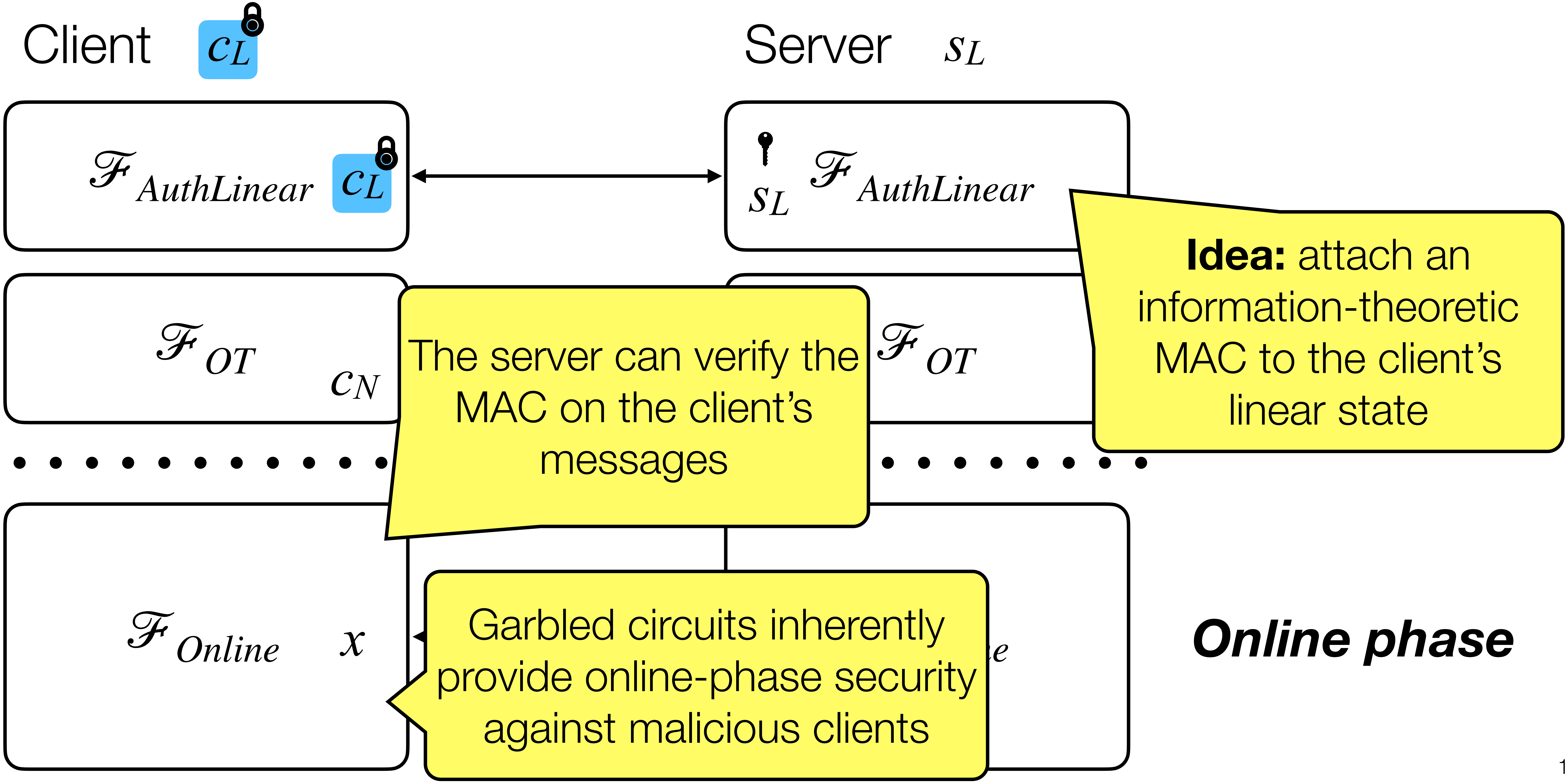
Preprocessing phase

Need to commit the client to the state they receive in the pre-processing phase

$\mathcal{F}_{Online}$ x $\longleftrightarrow$ $\mathcal{F}_{Online}$

Online phase

Extending Delphi to client-malicious security

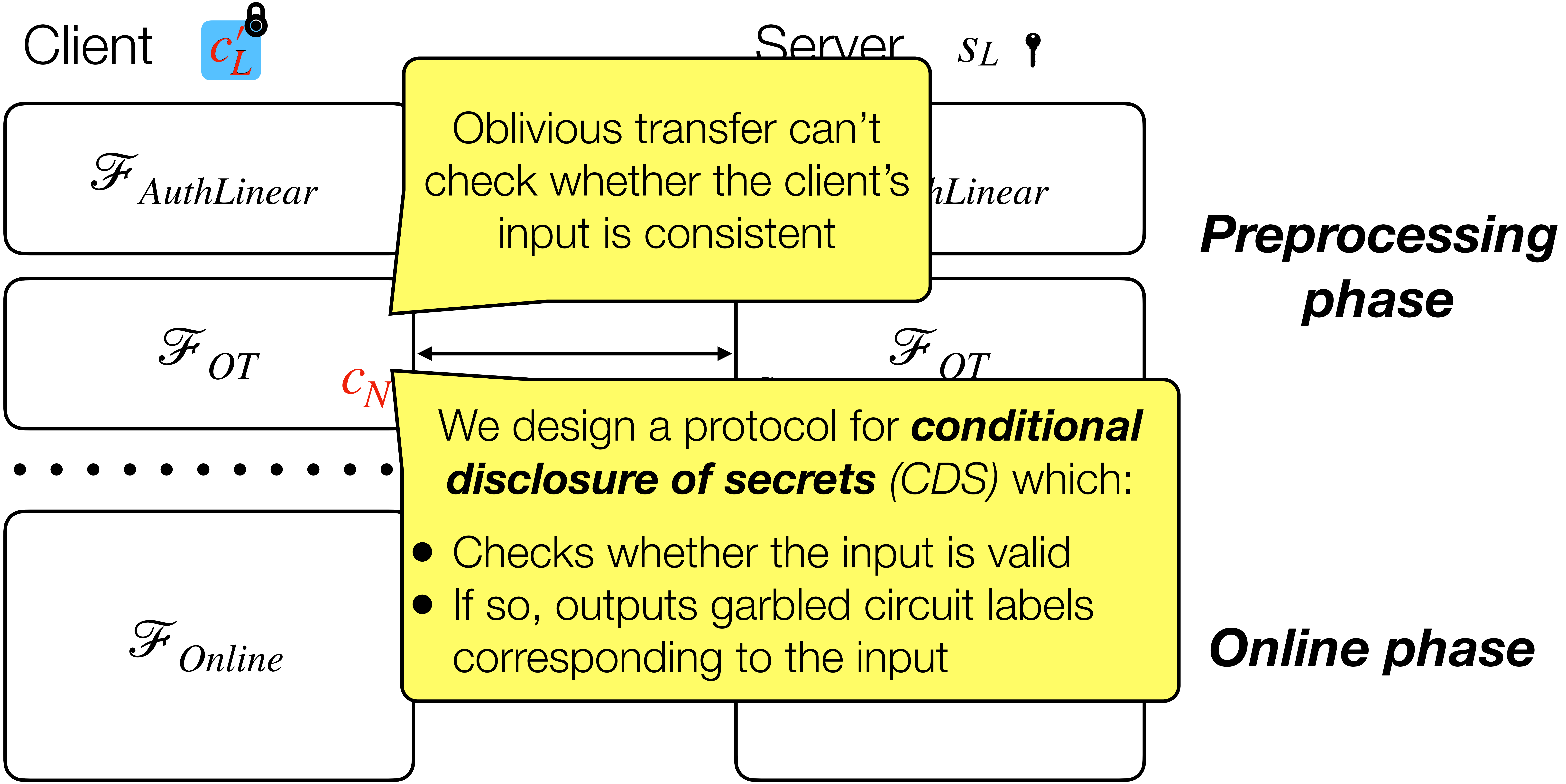


Extending Delphi to client-malicious security

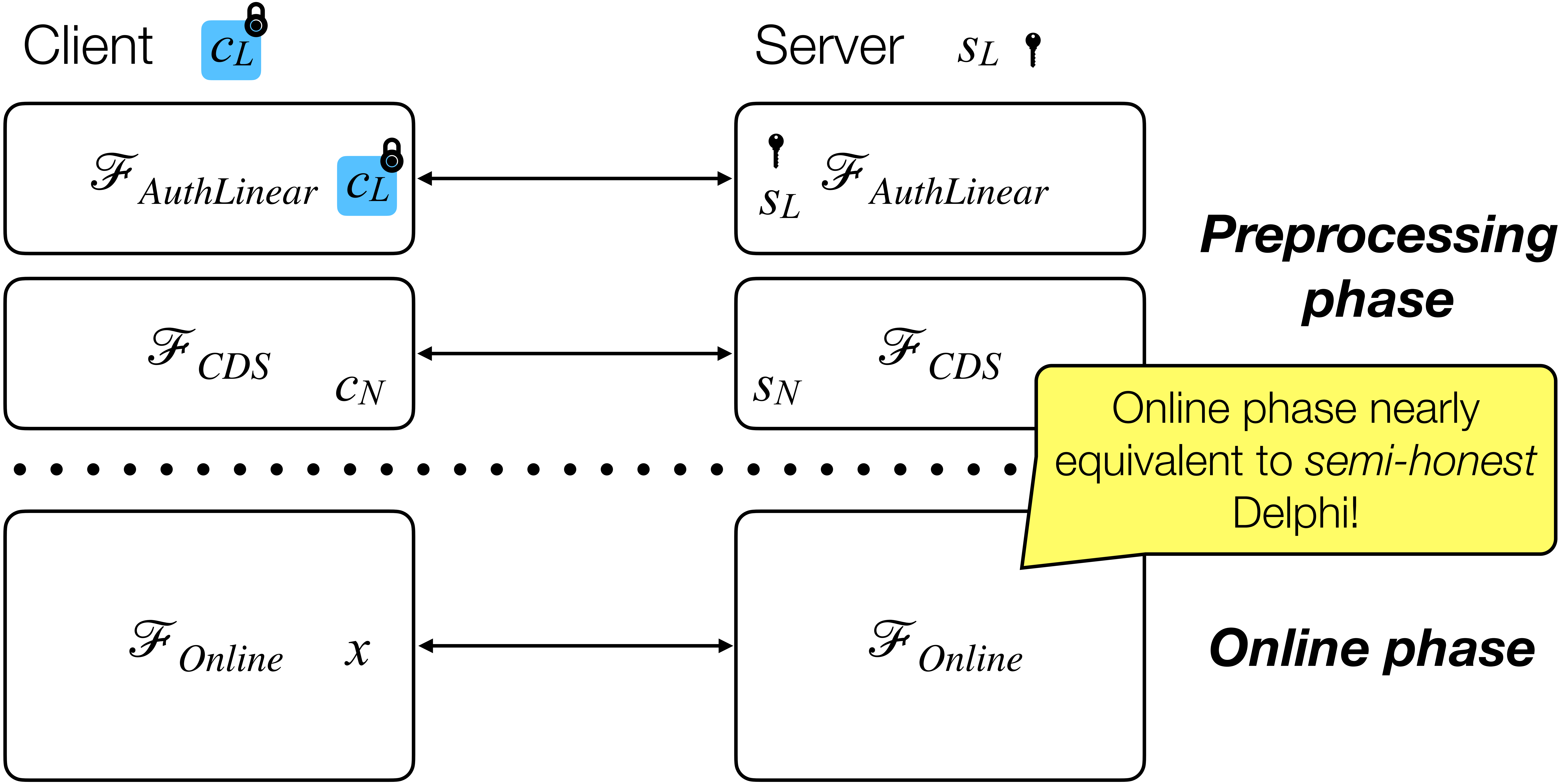
Oblivious transfer can't check whether the client's input is consistent

We design a protocol for ***conditional disclosure of secrets*** *(CDS)* which:

- Checks whether the input is valid
- If so, outputs garbled circuit labels corresponding to the input



Muse



Client c_L
Server s_L
$\mathcal{F}_{AuthLinear}$ c_L
s_L $\mathcal{F}_{AuthLinear}$
Preprocessing phase
$\mathcal{F}_{CDS}$ c_N
s_N $\mathcal{F}_{CDS}$
Online phase nearly equivalent to *semi-honest* Delphi!
$\mathcal{F}_{Online}$ x
$\mathcal{F}_{Online}$
Online phase

Implementation

Open-source Rust, Python, and C++ library with support for GPU acceleration

github.com/mc2-project/muse

ARTIFACT EVALUATED
usenix ASSOCIATION
PASSED

Evaluation

How does Muse compare against the following baselines?

Baselines:

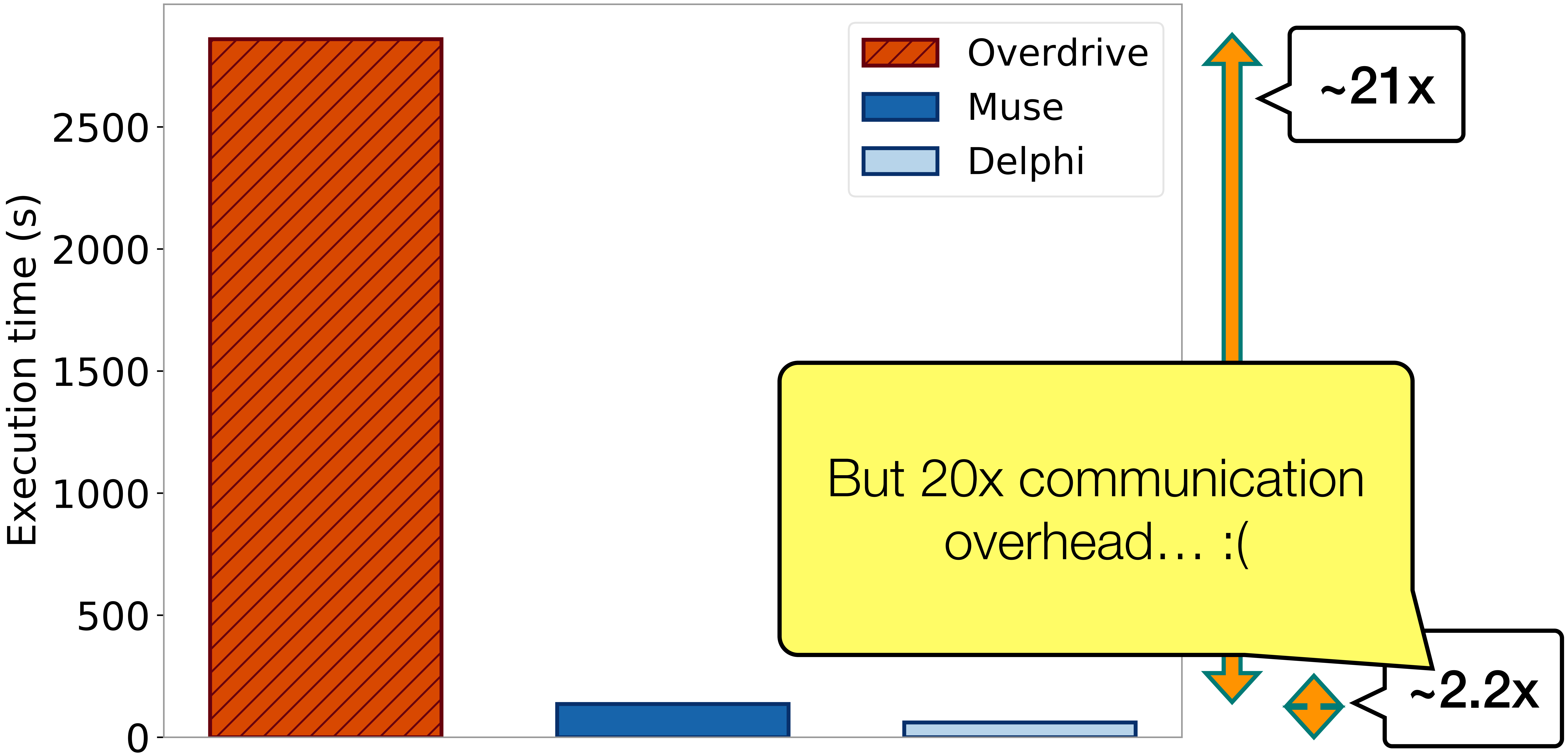
1) Overdrive [Kel+18] *(Generic protocol with **malicious** security)*

2) Delphi [Mis+20] *(Specialized protocol with **semi-honest** security)*

Benchmark: MiniONN network on CIFAR-10

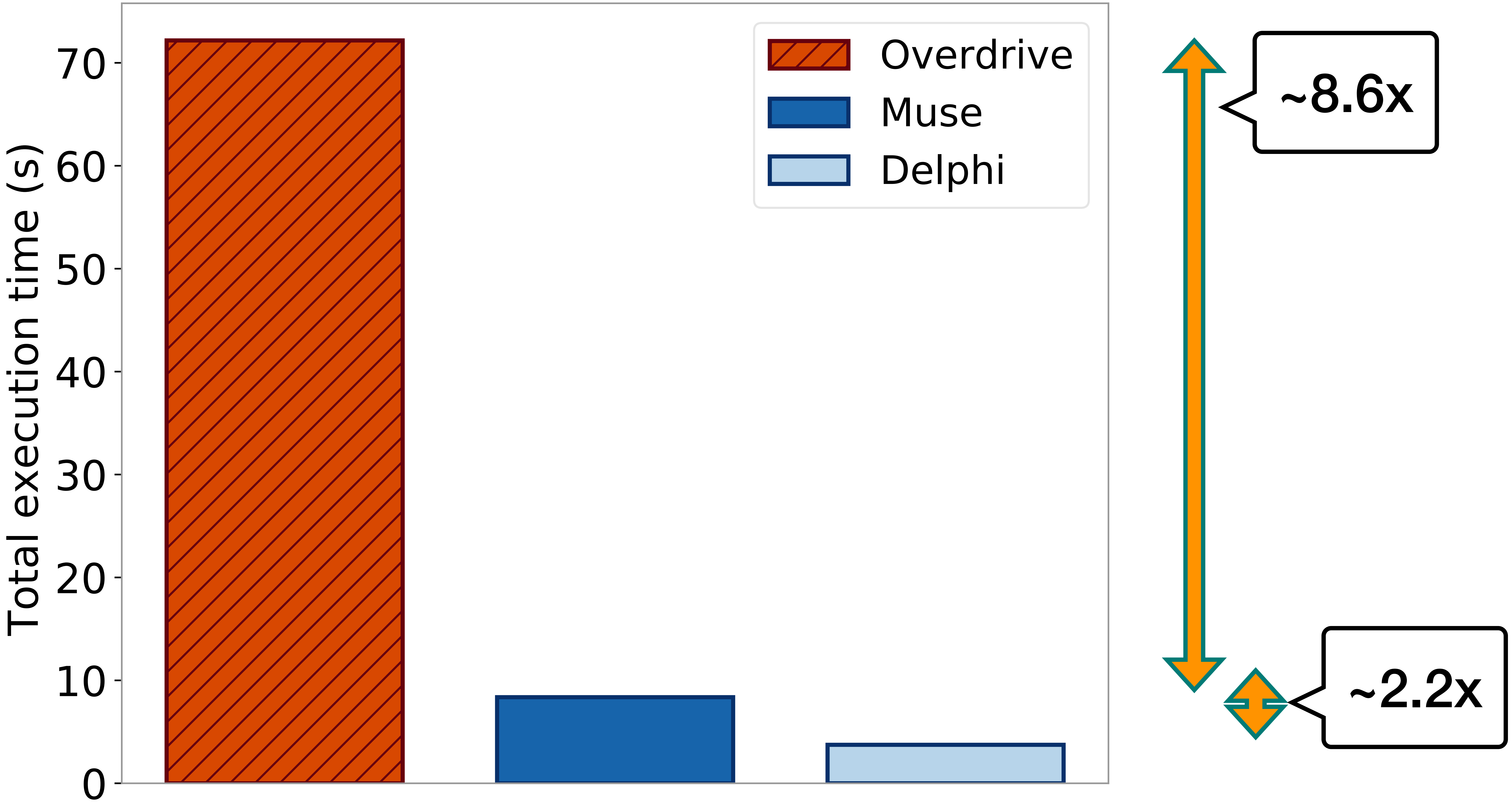
Preprocessing latency

Comparison with malicious Overdrive and semi-honest Delphi



Online latency

Comparison with malicious Overdrive and semi-honest Delphi



Muse

- A novel model-extraction attack against existing semi-honest secure inference protocols 24-312x more efficient than existing attacks
- A client-malicious secure inference protocol 21x more efficient than prior work

Thank you!

Ryan Lehmkuhl

ryanleh@berkeley.edu

github.com/mc2-project/muse