Gadget Note:
Garbled Circuit Lookup Tables

David Heath
University of Illinois Urbana-Champaign

UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

Gadget Note:
Garbled Circuit Lookup Tables

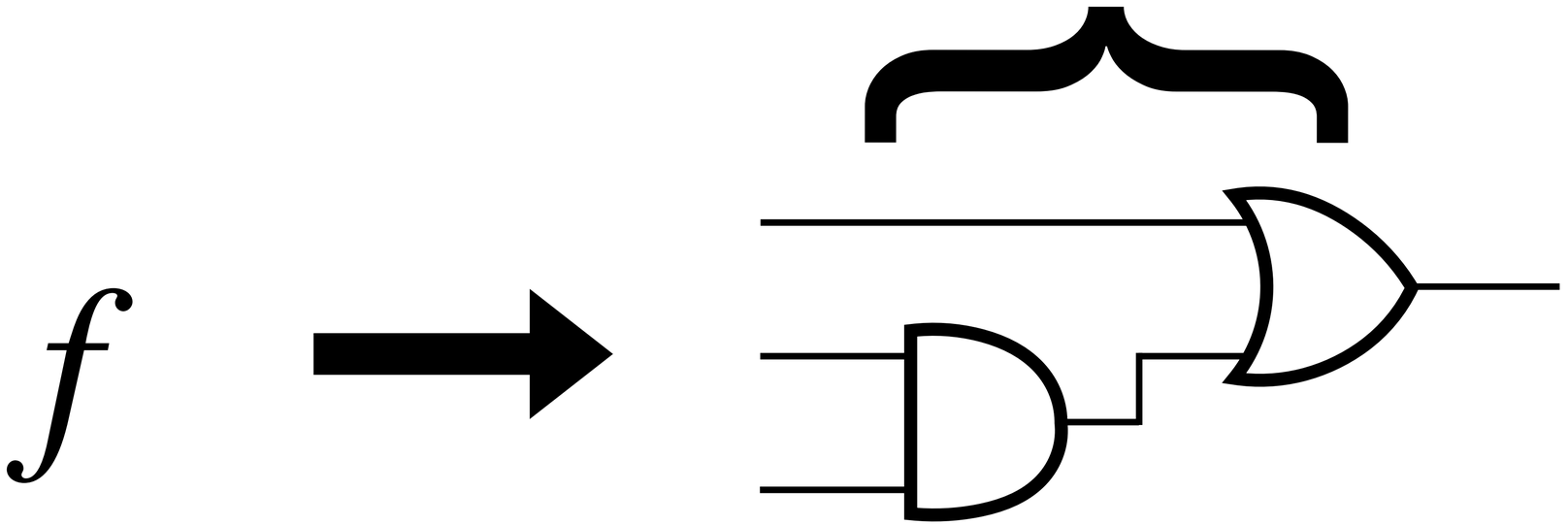
In connection with

“Thresholding symmetric-key primitives based on active general-purpose MPC”

Xiao Wang et al.

Garbled Circuits

AND/XOR gates



f

Garbler

Evaluator

Garbled Circuits

Garbler

C, x

C, y

Evaluator

Garbled Circuits

$\tilde{C}$

Garble

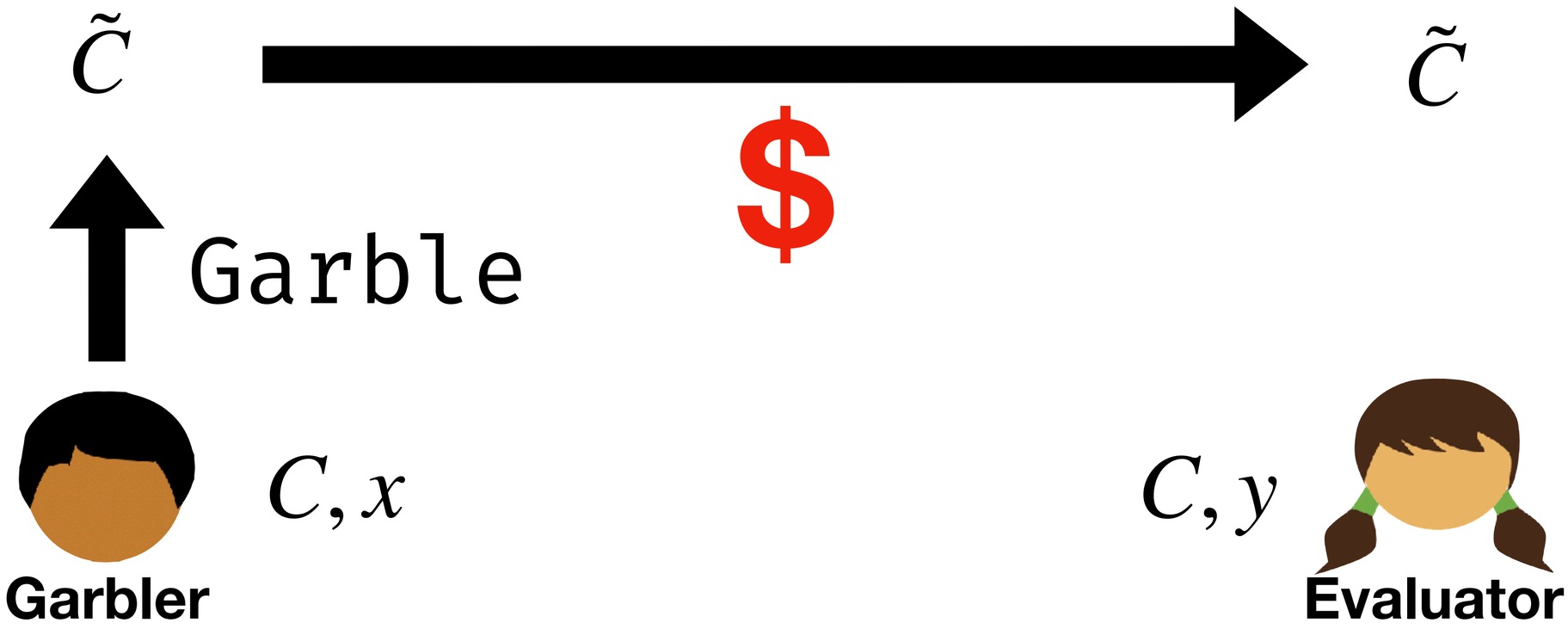
C, x

Garbler

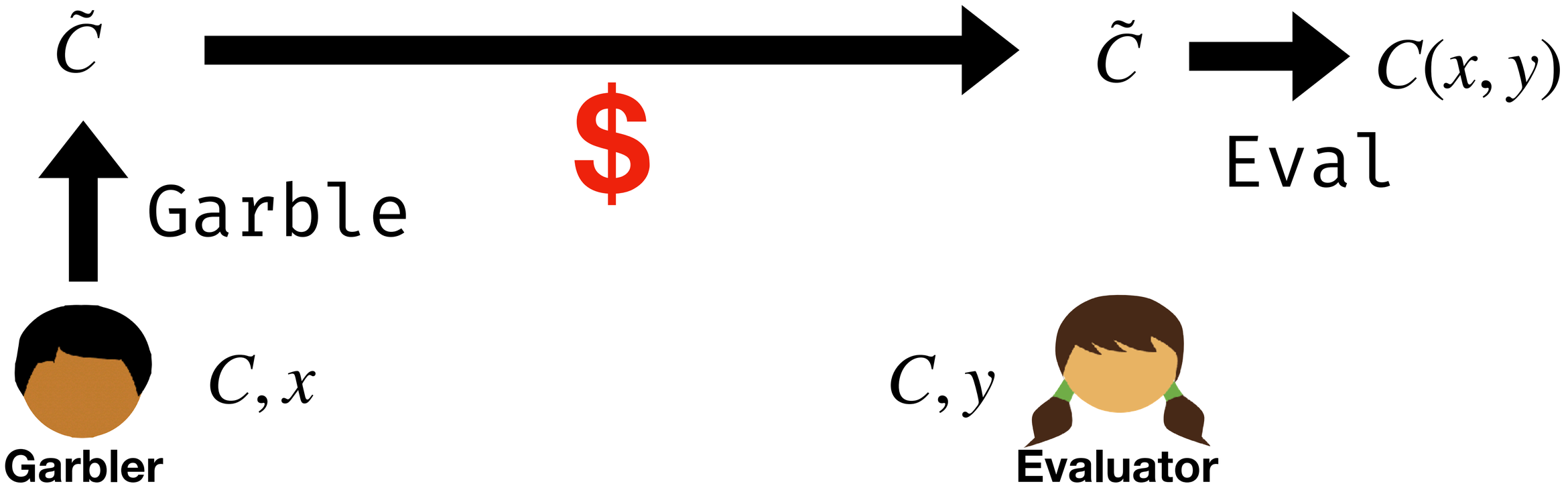
C, y

Evaluator

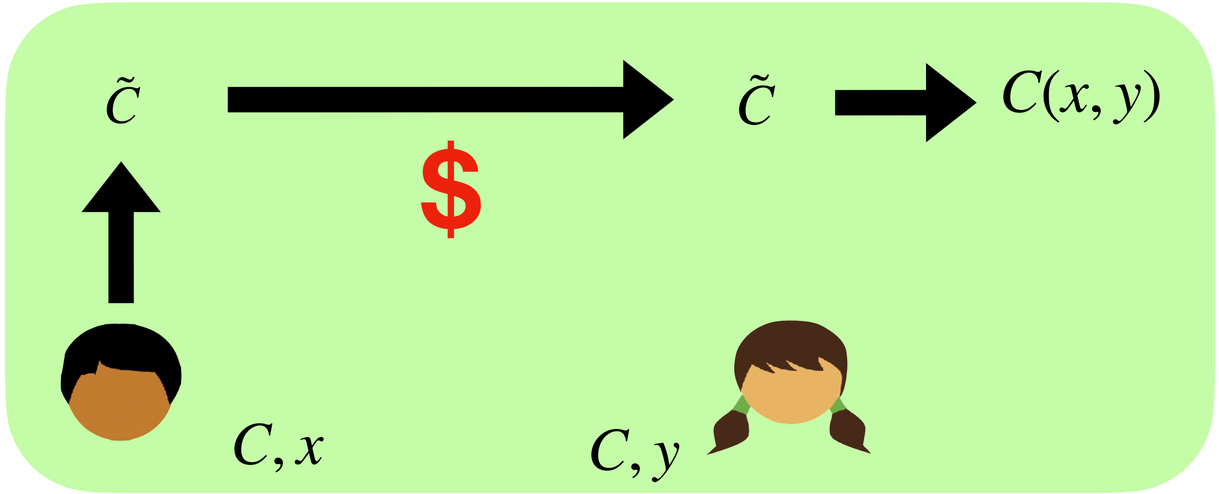
Garbled Circuits



Garbled Circuits



Garbled Circuits



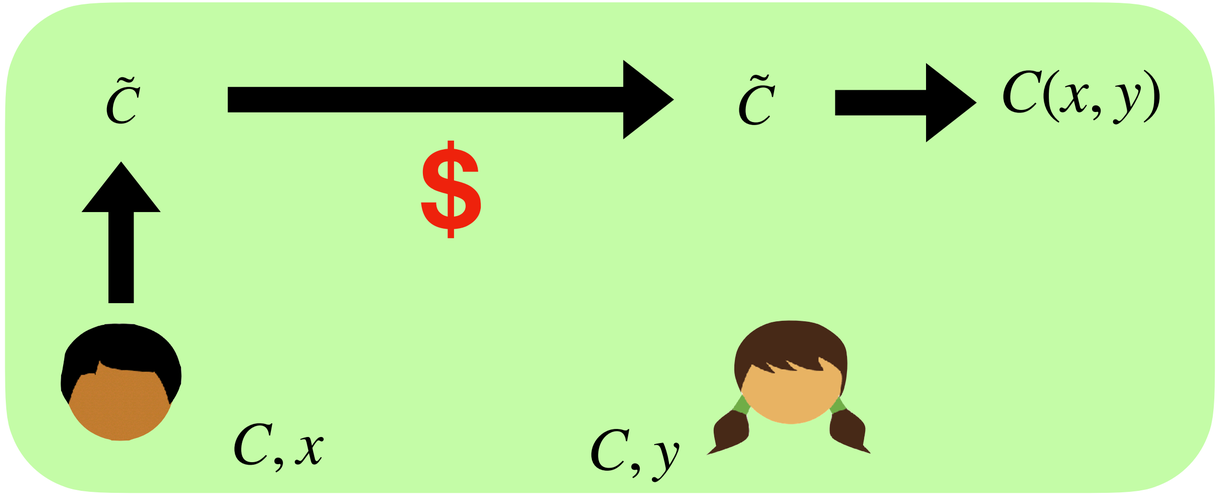
General Purpose

Stable, well understood

Simple, flexible (standalone primitive)

Fast, symmetric-key

Garbled Circuits



General Purpose

Stable, well understood

Simple, flexible (standalone primitive)

Fast, symmetric-key

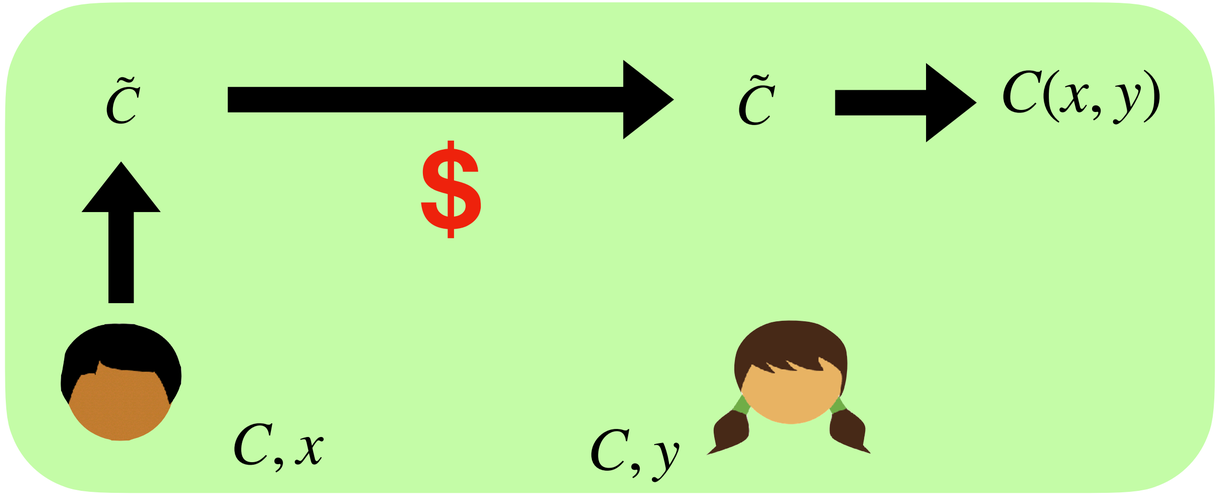
Cost:

Communication

Garbler Computation

Evaluator Computation

Garbled Circuits



General Purpose

Stable, well understood

Simple, flexible (standalone primitive)

Fast, symmetric-key

Cost:

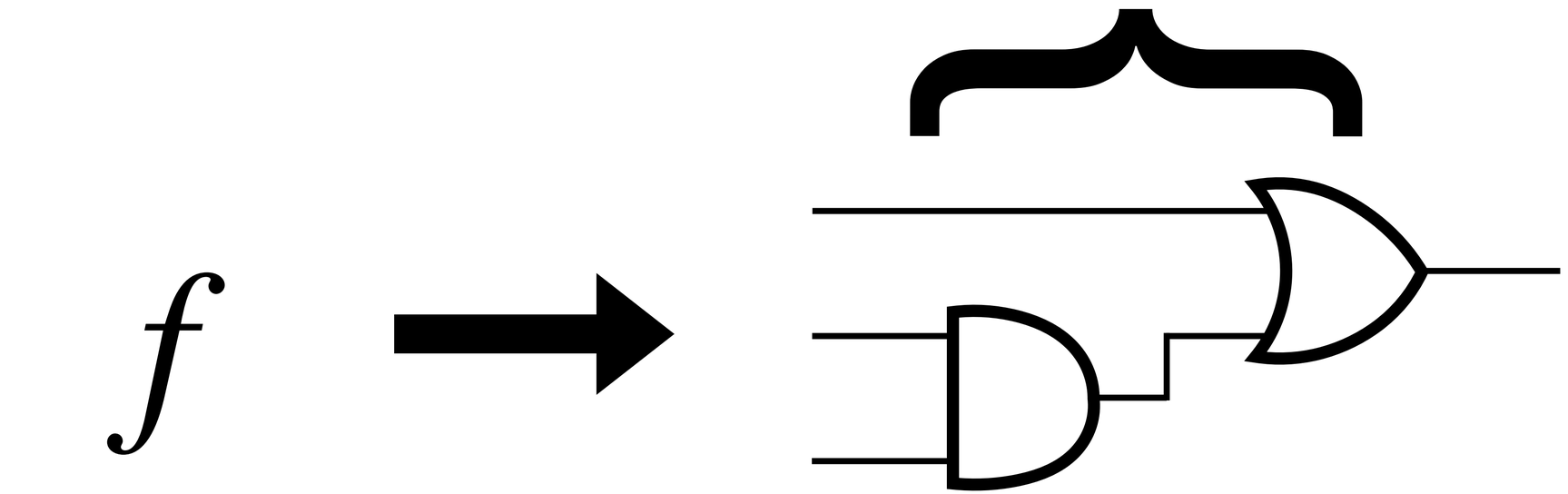
Communication!

Garbler Computation

Evaluator Computation

Garbled Circuits

AND/XOR gates



f

Garbler

Evaluator

Garbled Circuits

f → ?

Garbler

Evaluator

```
76    // The lookup-tables are marked const so they can be placed in read-only storage instead of RAM
77    // The numbers below can be computed dynamically trading ROM for RAM -
78    // This can be useful in (embedded) bootloader applications, where ROM is often limited.
79    static const uint8_t sbox[256] = {
80      //0     1    2      3     4    5     6     7      8    9     A      B    C     D     E     F
81      0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe, 0xd7, 0xab, 0x76,
82      0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72, 0xc0,
83      0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5, 0xf1, 0x71, 0xd8, 0x31, 0x15,
84      0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75,
85      0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f, 0x84,
86      0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b, 0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf,
87      0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f, 0xa8,
88      0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2,
89      0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19, 0x73,
90      0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb,
91      0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79,
92      0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4, 0xea, 0x65, 0x7a, 0xae, 0x08,
93      0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a,
94      0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d, 0x9e,
95      0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf,
96      0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb, 0x16 };
```

https://github.com/kokke/tiny-AES-c

In everyday computing, often useful to use **lookup tables**

Can we garble lookup tables?

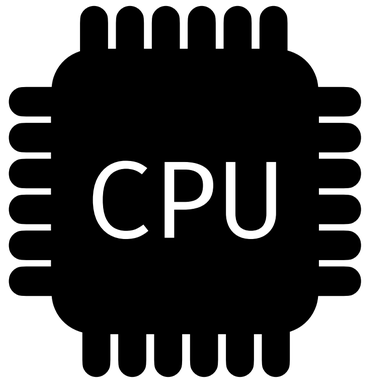
One-Hot Garbling

[HK21]

$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

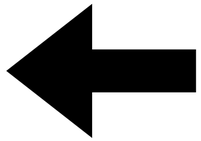
Garbled RAM

[LO13,…,HKO22,PLS23,HKO23]



One-Hot Garbling

$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

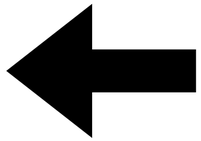


One-hot vector

where evaluator knows the hot position

One-Hot Garbling

$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$



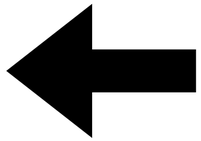
One-hot vector
where evaluator knows
the hot position

Possible to evaluate **any** function f(x),
where evaluator knows x

Somewhat similar to a distributed point function

One-Hot Garbling

$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$



One-hot vector
where evaluator knows
the hot position

Possible to evaluate **any** function f(x),
where evaluator knows x

Somewhat similar to a distributed point function

<u>One-time-use</u> lookup tables

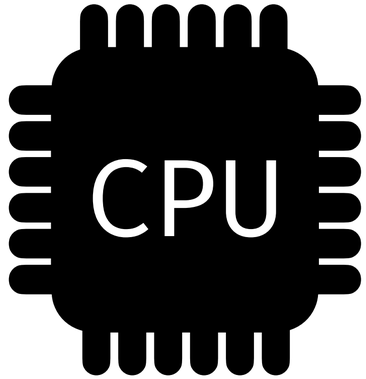
Trades *computation* for improved **communication**

"Outsources certain functions to the evaluator"

One-Hot Garbling

$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Garbled RAM



One-time-use lookup tables

Trades *computation* for improved **communication**

"Outsources certain functions to the evaluator"

One-Hot Garbling

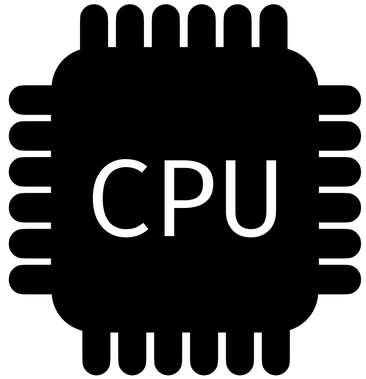
$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

<u>One-time-use</u> lookup tables

Trades *computation* for improved **communication**

"Outsources certain functions to the evaluator"

Garbled RAM



Basis for general-purpose garbling of programs

<u>Amortized</u> read/write array

Balances *computation* and **communication**

"Oblivious RAM protocol between the Garbled Circuit and the evaluator"

One-Hot Garbling

$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

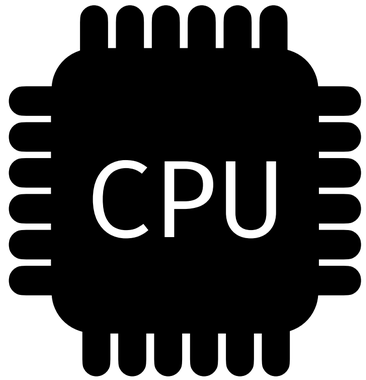
$O(\log n)$ comm.

$O(n \log n)$ comp.

Useful for functions with algebraic structure

n row table
Rows size log n

Garbled RAM



less than

$O(\log^4 n)$ comm./comp.

Useful for large, complex functions with little structure

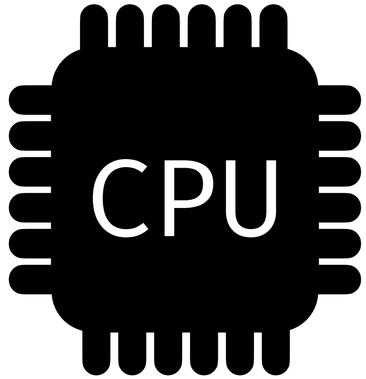
One-Hot Garbling

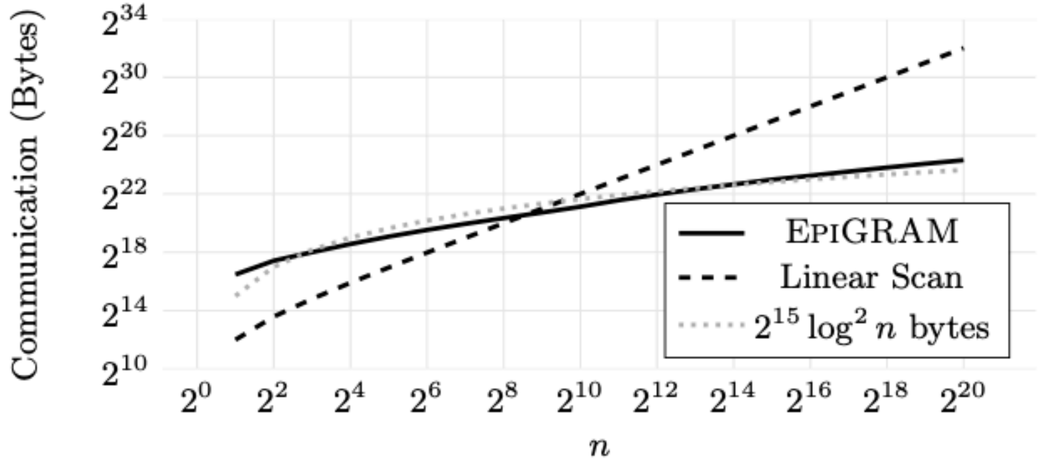
$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

AES S-Box: ~10% communication improvement compared to Boolean circuit evaluation with [ZRE15]

~30% with recent updates

Garbled RAM





One-Hot Garbling

$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

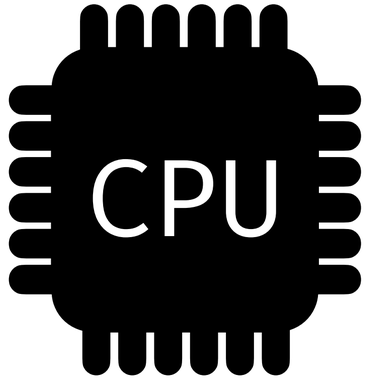
Integer multiplication,

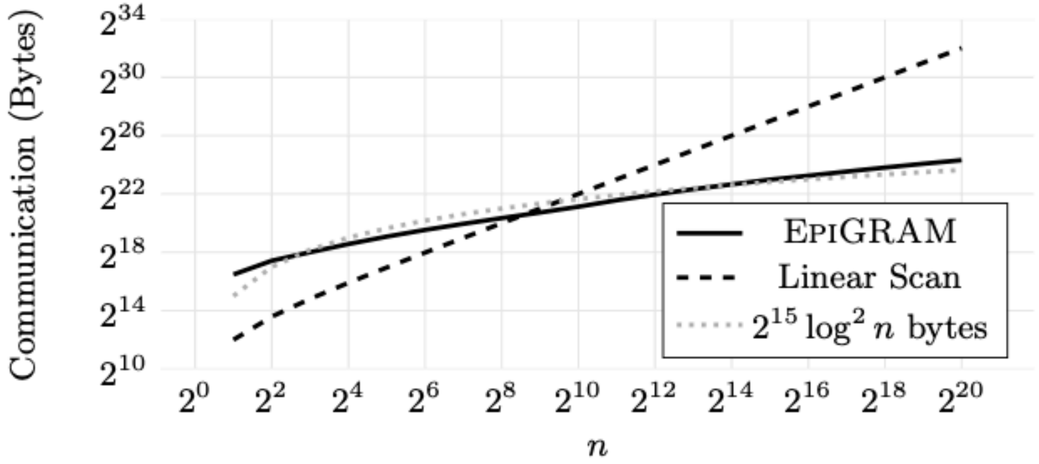
Field arithmetic,

Matrix operations

Opens a design space

Garbled RAM





One-Hot Garbling

$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

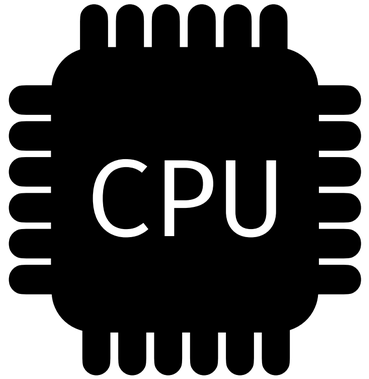
Integer multiplication,

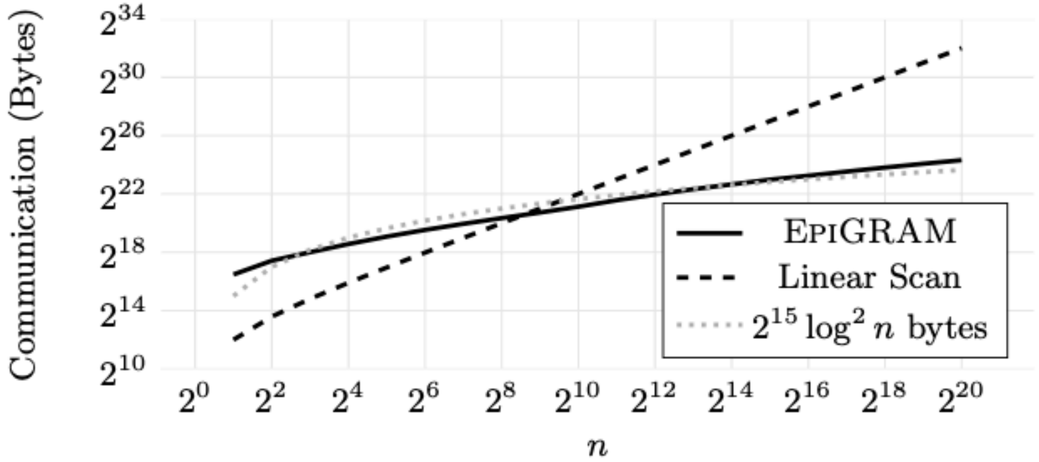
Field arithmetic,

Matrix operations

Opens a design space

Garbled RAM





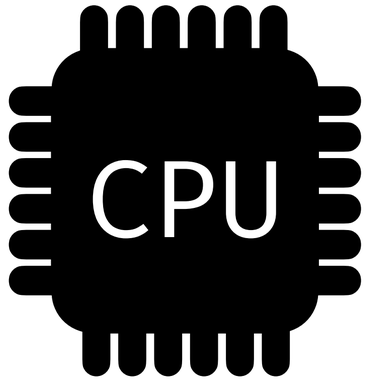
Compatible with authenticated garbling

Garbled Lookup Tables

One-Hot Garbling

$$\mathscr{H}(x) \otimes y = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \\ y_0 & y_1 & \cdots & y_{m-1} \\ 0 & 0 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Garbled RAM



Generalize garbling beyond Boolean circuits

Offer new ways forward for significant improvement

Useful for evaluation of cryptographic primitives