Fairplay

A Secure Two-Party Computation System

Peter Story

Motivation

- The Millionaire’s Problem
    - Who is the wealthier businessman?
- Ecommerce applications

Paper's Contribution

- FairPlay is an implementation of a Secure Function Evaluation (SFE) protocol

- SFE makes it possible for 2 people to answer questions that would ordinarily require a trusted 3rd party

Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. 2004. Fairplay—a secure two-party computation system. In *Proceedings of the 13th conference on USENIX Security Symposium - Volume 13* (SSYM'04), Vol. 13. USENIX Association, Berkeley, CA, USA, 20-20.

Computer Theory

- Fairplay uses encryption
  - Encryption relies on the concept of computational hardness
- Fairplay uses SHA-1 cryptographic hash function
  - One-way function

Overview of Protocol

Step 1: Boolean Circuits

- The function must be defined as a boolean function
- Billionaire's Problem:
  64 inputs, 254 gates, 2 outputs

2: Garbling the Circuits

- Bob garbles (encrypts) the circuit
- Provides a way to hide what 1s and 0s are moving through a circuit
- Inputs and outputs of gates are encryption keys

3: Bob sends Circuit and his Inputs

- Bob sends Alice the circuit and his inputs
- To Alice, Bob’s inputs mean nothing

5: Alice's Inputs

- Only Bob knows which encryption keys Alice needs to represent any specific input
- If Alice asks for the encryption keys correspond to input 11010100, that defeats the point!
- Using Oblivious Transfer:
    - Bob supplies two inputs to the protocol
    - Alice receives exactly one output, and Bob doesn't know which output it is

6: Circuit Evaluation

- Alice has her inputs and Bob’s inputs
- She evaluates the Boolean circuit
  - Her output: plaintext
  - Bob’s output: encrypted

7: Bob Receives Outputs

- If Alice is nice, she will send Bob the output of the function
- He can verify its authenticity, because he encrypted it

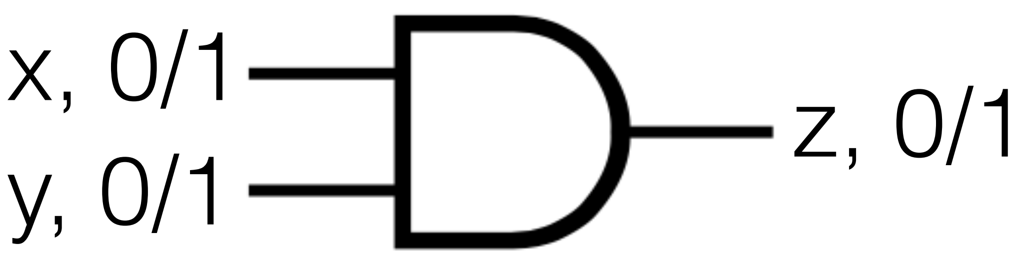
Missing Pieces

- Oblivious Transfer
- Garbled Circuits

The Art of Garbling

- To review, garbling:
  - Hides the 1s and 0s moving through the circuit
  - Inputs and outputs of gates are encryption keys
- Originally presenting in 1986, Yao's Protocol

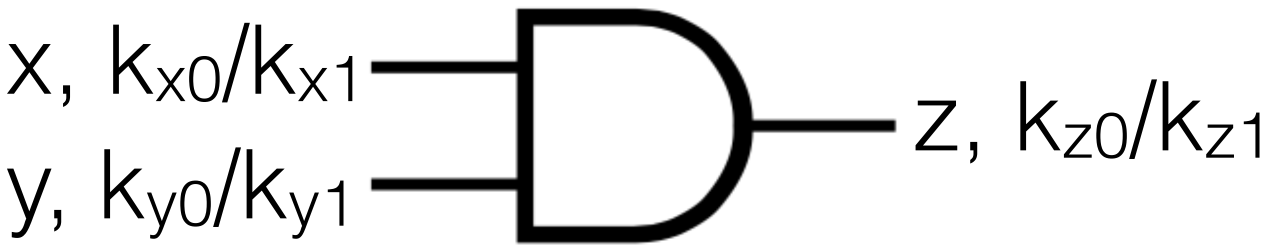
Regular AND-gate



x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

Thanks to Prof. Yehuda Lindell of Bar-Ilan University
“Secure Two-Party Computation in Practice” Lecture

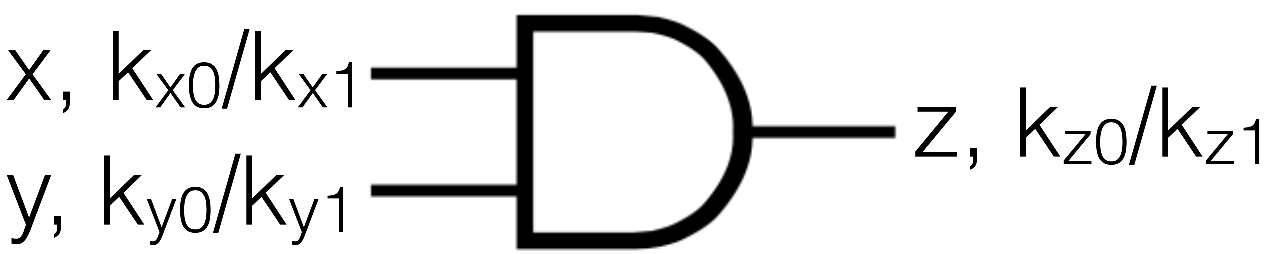
Modified AND-gate



x	y	z
$0 \rightarrow k_{x0}$	$0 \rightarrow k_{y0}$	$0 \rightarrow k_{z0}$
$0 \rightarrow k_{x0}$	$1 \rightarrow k_{y1}$	$0 \rightarrow k_{z0}$
$1 \rightarrow k_{x1}$	$0 \rightarrow k_{y0}$	$0 \rightarrow k_{z0}$
$1 \rightarrow k_{x1}$	$1 \rightarrow k_{y1}$	$1 \rightarrow k_{z1}$

Still a problem:
Frequency of k_{z0} tells you it represents 0.

Encrypted AND-gate

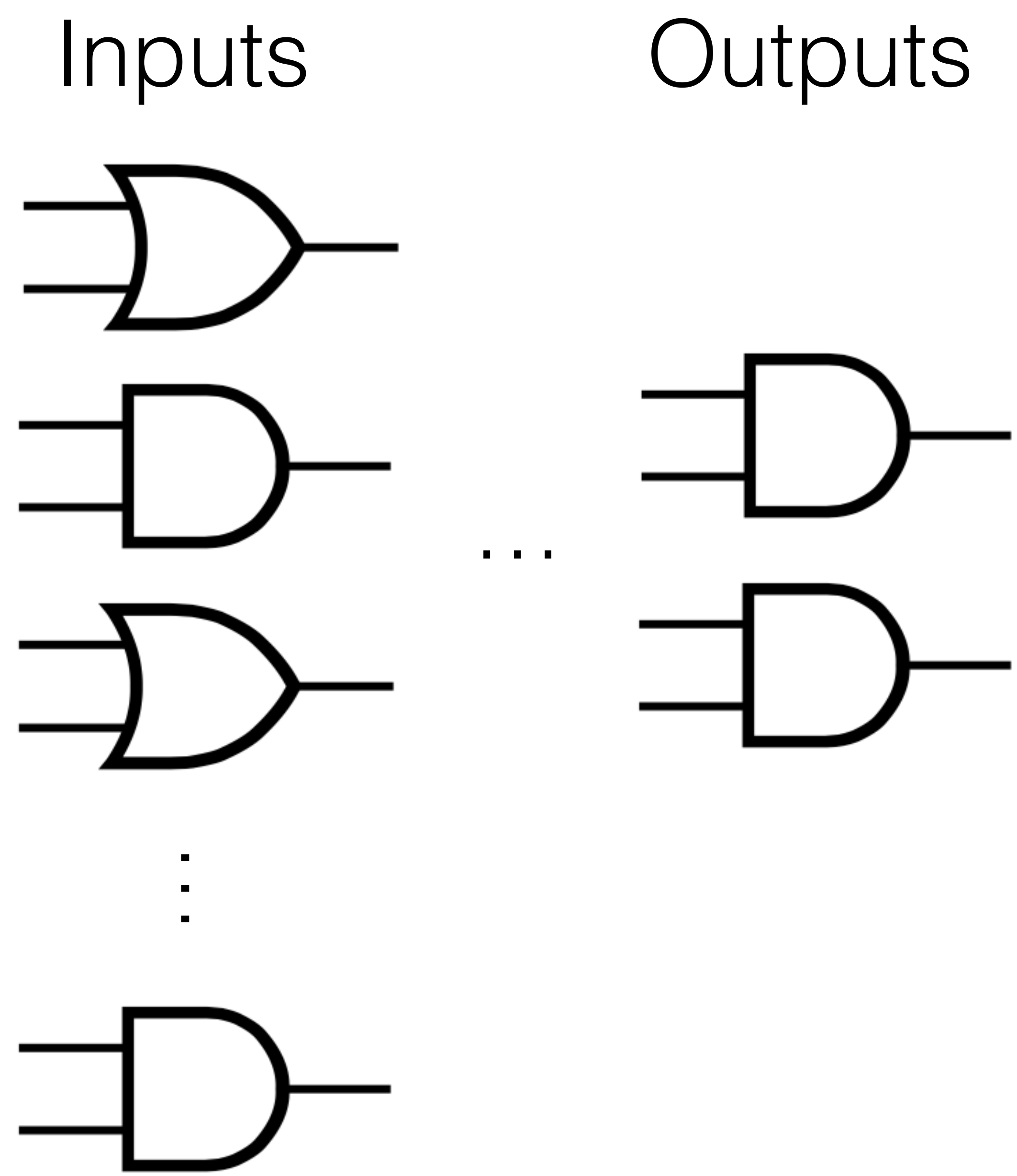


x	y	z
k_{x0}	k_{y0}	$E_{kx0}(E_{ky0}(k_{z0}))$
k_{x0}	k_{y1}	$E_{kx0}(E_{ky1}(k_{z0}))$
k_{x1}	k_{y0}	$E_{kx1}(E_{ky0}(k_{z0}))$
k_{x1}	k_{y1}	$E_{kx1}(E_{ky1}(k_{z1}))$

Solution:
Encrypt the outputs using inputs as encryption keys.

Back to Boolean Function

- Inputs are encrypted
- Intermediate gate I/O is encrypted
- Outputs:
  - Alice's are decrypted
  - Bob's are encrypted



Exercise

- Garbling and evaluating a boolean circuit
- Your exam question will be very similar!

http://peterstory.me/fairplay/