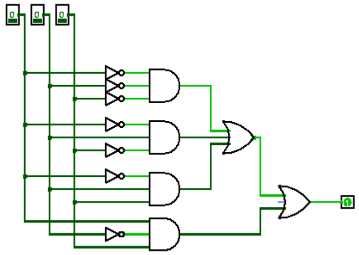
Secure Computation with Sublinear Amortized Work

Dov Gordon, Jonathan Katz, Vladimir Kolesnikov,

Tal Malkin, Mariana Raykova, Yevgeniy Vahlis

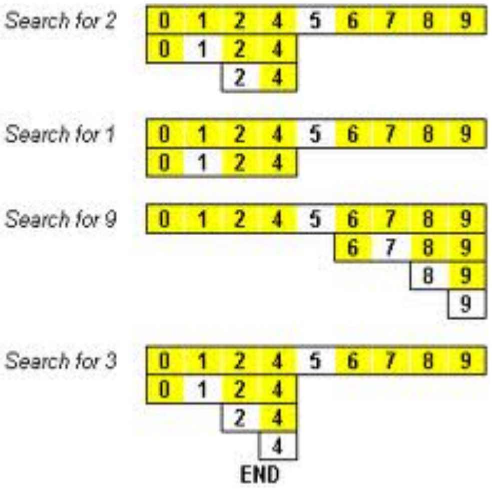
Secure Computation

- We can compute securely every function
- Ω(n) work is necessary for any computation
  - Computation on circuits
  - Inherent inefficiency
    - By definition of privacy, computation must touch every data point
    - The circuit takes as inputs all data



Computation without Security

- Search functionality
  - Given sorted, hashed data
  - Search time sublinear in the number of inputs
  - Examples: database of n inputs
    - Binary search – log n look-ups
    - Hash table – const look-ups



Can we do better?

- Amortized Complexity
- Random Access Machine (RAM) computation model:
  - Sublinear running time in input size
  - Do not need to touch every program branch – better efficiency

Our Contributions

- Generic solution to sublinear 2PC
  - Compiler for any computation in the RAM model into secure computation 2PC in the RAM model.
  - Tools: any oblivious RAM, any 2PC protocol
- Optimized efficiency construction
  - Based on [GO96] and [Yao82]
  - Minimize garbled circuits – multiplication, Boolean comparison and XOR
    - Free XOR gates in garbled evaluation

Efficiency

Any two party computation with input of size N and t instructions on a RAM, can be computed securely:

- $O(\max(\log^3 N, \log^3 t))$ amortized computation overhead
- memory storage: constant for one party and linear in the size of the input data for the other