Computing on Encrypted Data

Nigel Paul Smart

Dining Bankers (a.k.a. Millionaire’s Problem)

A set of bankers go to lunch.

They are celebrating their bonuses just being paid.

Each has been given a bonus of x_i dollars.

The one with the biggest bonus should pay.

But they do not want to reveal their bonus values.

Dining Bankers (a.k.a. Millionaire's Problem)

What they want to compute is the function

$$F(x_1,\ldots,x_n) = \{\ i : x_i \geq x_j \ \text{ for all } j\ \}$$

without revealing the x_i values.

This problem (Millionaires Problem) introduced by Andrew Yao in early 1980s.

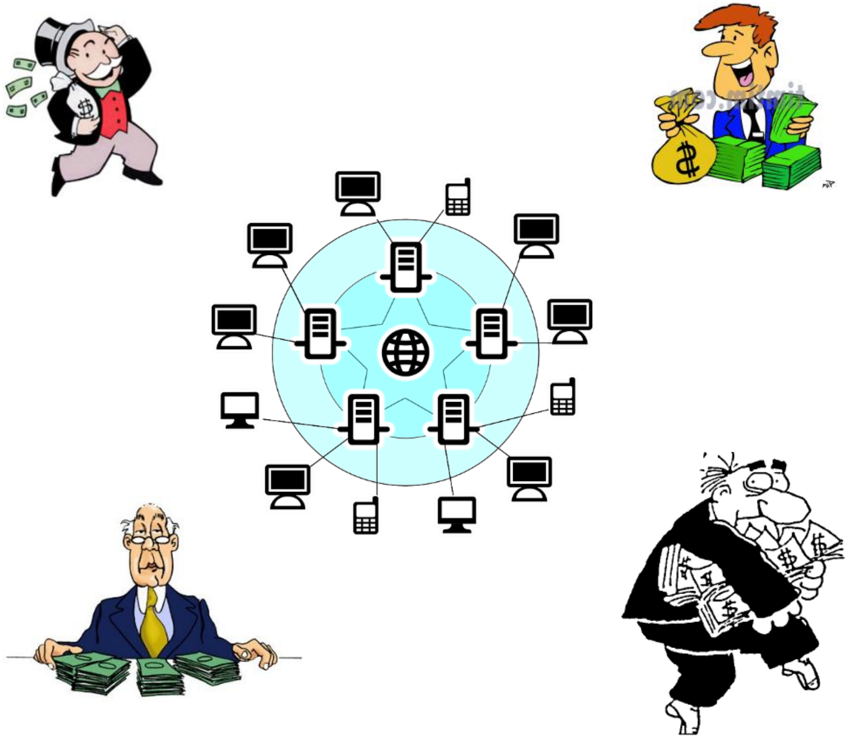
Andrew won the Turing Award for this and other work.

Dining Bankers (a.k.a. Millionaire’s Problem)

If the bankers had a person they trusted they could get this person to compute the answer to their problem for them.

They give the trusted person their bonus values and the trusted person computes who should pay for lunch.

Dining Bankers (a.k.a. Millionaire’s Problem)



In real life such trusted people do not exist, or are hard to come by. So we want a protocol to compute the function securely. This is what MPC does.

It emulates a trusted party, enabling mutually distrusting parties to compute an arbitrary function on their inputs.

All that is revealed is what can be computed from the final output.

Basic Set Up

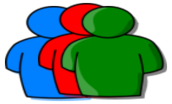
- We assume some data is being processed.
  - Think of genomic data, but it could be anything

- There are three basic groups of actors
  - Input Parties 
  - Processing Parties 
  - Output Parties 

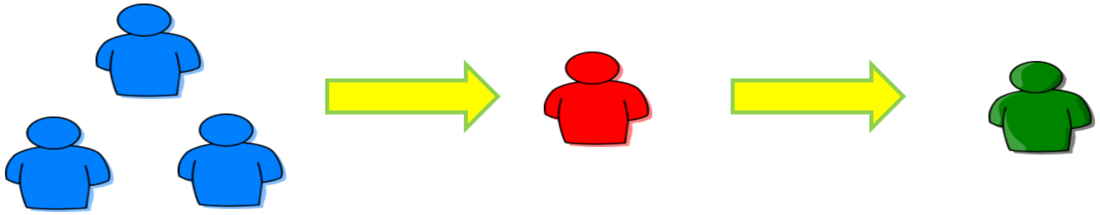
- In a traditional application there is one of each, and they are all the same person.
- We could however have very different scenarios…

Scenarios

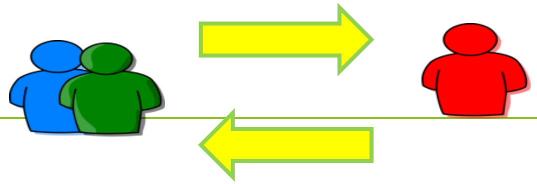
- Traditional



- Many Different Input Parties

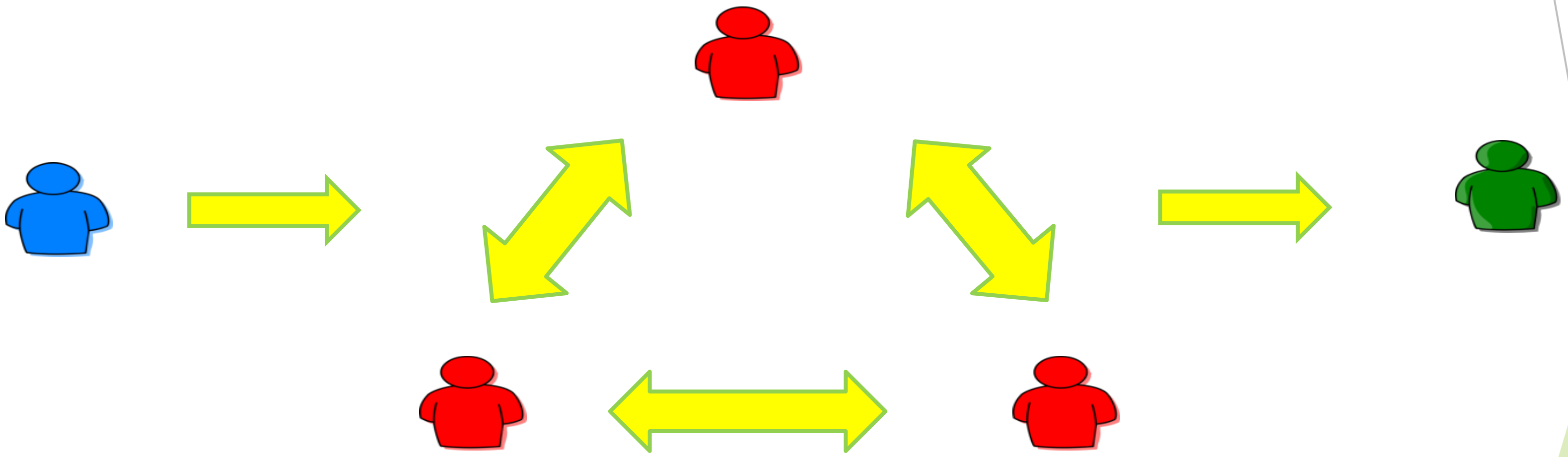


- Input Parties=Output Parties
  - Think of this as the usual paradigm for Cloud Computing



Scenarios

- Many computing parties



- And all other combinations of the above

Security Problems

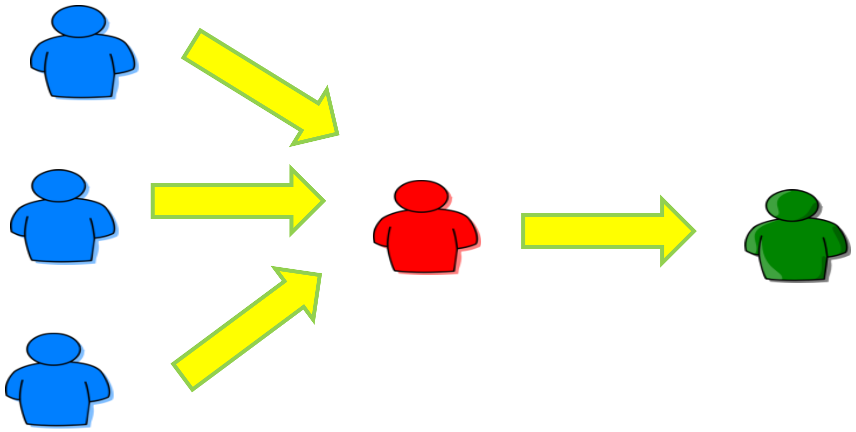
- As soon as we separate input, processing and output parties we have various security problems
    - What if the input data is private, how can we allow the computing parties to compute on it? (FHE/MPC)
    - Maybe the output of the function reveals the input data to some degree. How do we protect against this? (Differential Privacy)
    - How do we know the computing parties are doing the correct operation? (Verifiable Computation)

Security Models

- In some sense we always assume input parties are “honest”.
- Other parties can be dishonest.
- How dishonest is dishonest though?
  - Semi-honest (a.k.a passive or honest-but-curious)
  - Covert (does not want to be caught)
  - Fully malicious (does not care if caught)
    - Honest parties still want the output though, or be able to catch the bad guys
  - Monolithic (all bad guys act together)

Fully Homomorphic Encryption

- One computing party
- One or many input parties
- One output party (could be more)



Fully Homomorphic Encryption

- Input parties encrypt their data
- Computing party evaluates the function on the encrypted data (without seeing the data)
- Output party performs the decryption

- First scheme 2008
- In theory can compute any function, with only a small overhead in cost
- In practice much more difficult

Fully Homomorphic Encryption

- We are protecting against a passive/semi-honest dishonest computing party
- To protect against malicious computing party need to add a verifiable computation protocol on top
- To protect against passive output party need to add differential privacy on top.

Fully Homomorphic Encryption

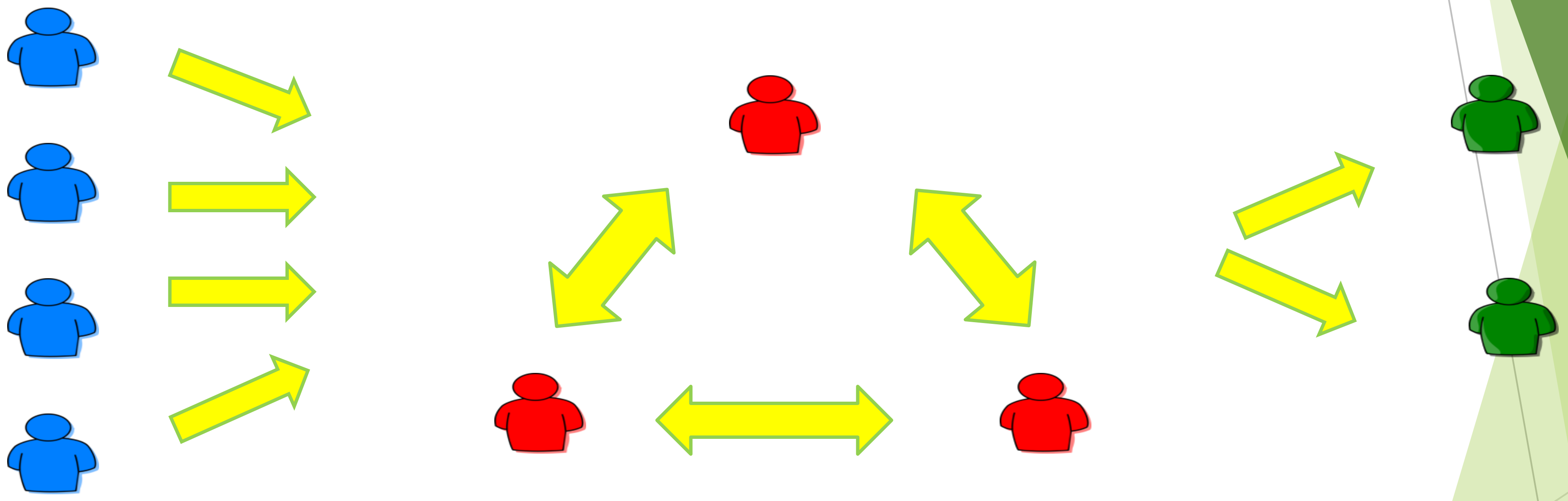
- Today this is practical for functions of low multiplicative depth
    - Think basic statistics, machine learning algorithms

- Been “applied” to various types of applications in the biological space (including genome data) by Microsoft Research (Redmond).

Multi-Party Computation

- The problem with FHE (i.e. the thing which made it hard to produce) was that we had only one computing party.
- With MPC we can have many input, computing and output parties, and indeed they could all be subsets of each other (or even exactly the same parties)
- Key point is that we have $n \geq 2$ computing parties

Multi-Party Computation



Multi Party Computation

- Most modern MPC protocols protect against
    - Malicious and/or covert adversaries
    - Controlling a majority of the computing parties
        - Need at least one honest computing party

- For efficiency work in a pre-processing model
    - Computation split into two phases
    - Phase 1: Input and (essentially) function independent
    - Phase 2: Process the actual input and function
    - Called the Offline and the Online phase
    - Offline phase is around 10x slower than the online phase.

Multi Party Computation

- To protect against a dishonest output party need to add differential privacy
- No need for verifiable computation, as the MPC takes care of that.
- In theory can compute any function, with only a small overhead in cost
- In practice much more efficient than FHE.
- Can compute relatively complex functions with reasonable efficiency.

Case Study: CYBERNETICA

MPC system based on 2-out-of-3 secret sharing

- Main system is semi-honest secure

Used in DARPA project PROCEED, Estonian government applications

- Satellite conjunction analysis
- Shared tax/education data
- Industrial sector analytics

Distributed secure database system called Sharemind

- Focused on performing operations over different organizations shared data

sharemind

Case Study:

DYADIC

THE KEY TO ENCRYPTION

Distributed “HSM” like functionality, called a Distributed Security Module (DSM)

- Fully actively secure
- 1-out-of-2, or other forms of secret sharing

Distributed credential system

- Doing MPC between a mobile phone and a server

Join NEC/Dyadic product of a distributed MPC enabled database system

- Focused on providing a secure DB in which admin does not have access to the data

Case Study: PARTISIA

Various auction and other "market" functionalities

Most famous for the Danish Sugar Beet auction

Also does Danish electricity power market, and others

Contract exchange

Recently formed spin-out doing secure key management called Sepior

SEPIOR

Other Companies With Demonstrators

Other Companies With Interest

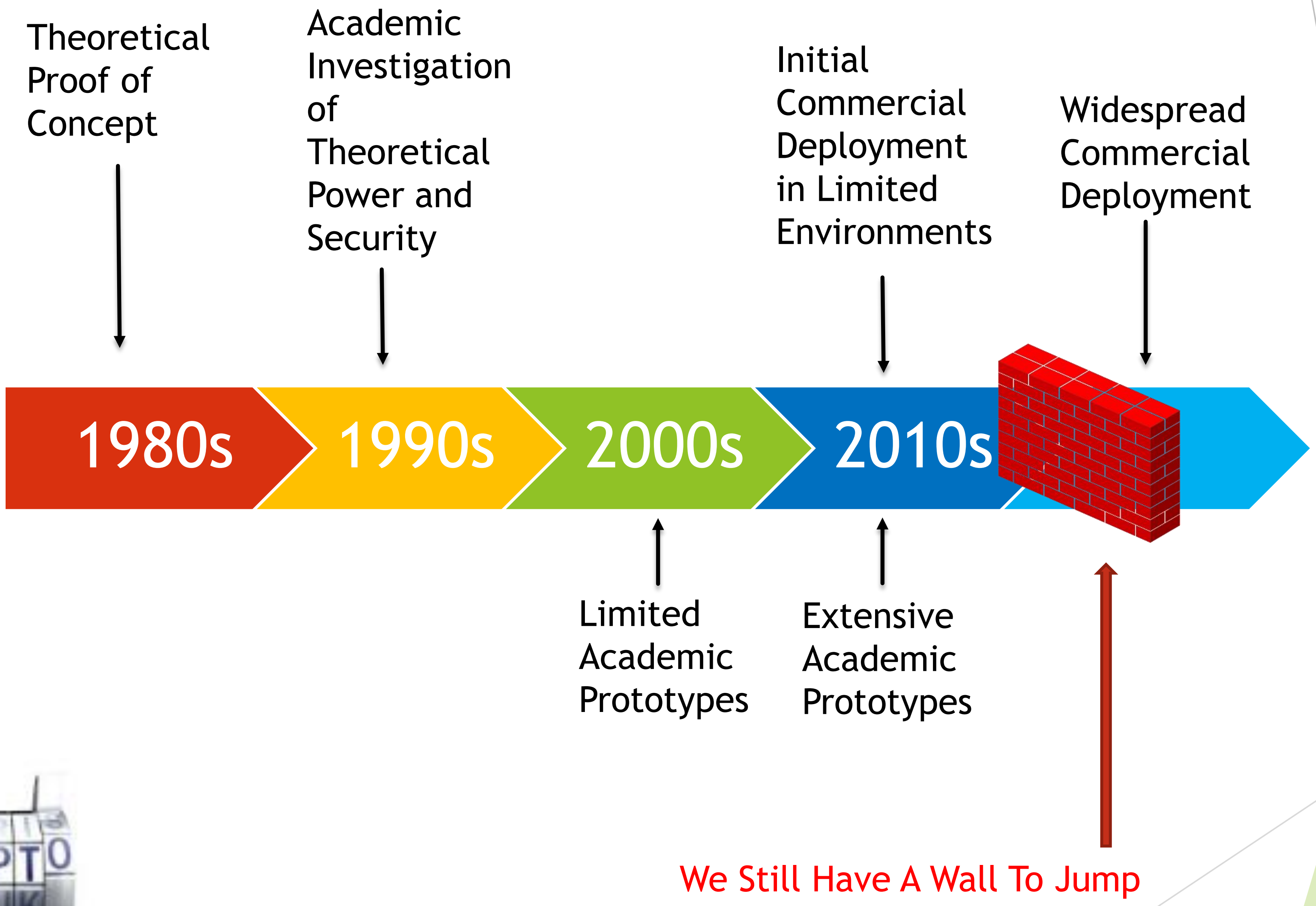
Current Summary

These are all solutions of limited point functionality

Useful in limited applications or limited envionrments.

All use MPC, the current best known approach for MPC

Question is how to take this to the next level....


Theoretical Proof of Concept
Academic Investigation of Theoretical Power and Security
Initial Commercial Deployment in Limited Environments
Widespread Commercial Deployment
1980s
1990s
2000s
2010s
Limited Academic Prototypes
Extensive Academic Prototypes
We Still Have A Wall To Jump

PROJECT: Trust and Security in Numbers

Engineering and Physical Sciences Research Council

Funded by UK's EPSRC : £1,509,000

Aim is to develop demonstrator applications of MPC technology in different application areas.

Focus is on applications and real world engagement, and the underlying engineering.

Teaming up with the above three companies as well as others such as Microsoft, Hewlett-Packard Enterprises.

Working out how to break down the above wall, and what could lie on the other side

PROJECT: IMPaCT

European Research Council
Established by the European Commission

Funded by ERC : € 2,000,000

Looking at the fundamental technological problems impeding MPC.

Focused on the underlying science and basic research behind practical MPC

Providing the theoretical and cryptographic tools to break down the above wall

PROJECT: Brandeis

Funded by DARPA : $ 1,200,000 out of $70,000,000 total programme.

Building a (single) secure DB accessed by an MPC engine.

- Looking at policy, access control, differential privacy and algorithmic aspects
- How to combine the DB and the MPC engine is a major research challenge
  - New symmetric ciphers/modes of operation needed for example

Three demonstrators

- Smart buildings/cities
- Mobile applications
- Pacific fleet (and other fleet) deployment

We are working with Galois and Rutgers University in the Jana subproject

QUESTIONS?