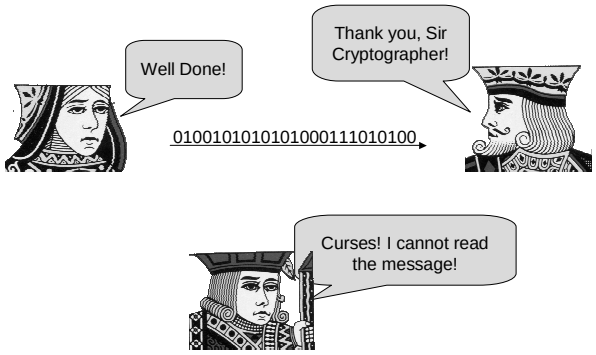
Secure Computation
(a tutorial)

Joe Kilian
NEC Laboratories, America

Aladdin Workshop on
Privacy in DATA
March 27, 2003

Cryptology – The First Few Millennia



Goal of cryptology – protect messages from prying eyes.

Lockboxes for data: data safe as long as it is locked up.

The Last Twenty Years

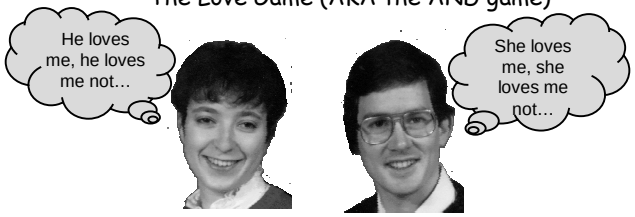
Then: data protected, but not used.

Now: Use data, but still protect it as much as possible.

Secure Computation:

Can we combine information while protecting it as much as possible?

The Love Game (AKA the AND game)



Want to know if both parties are interested in each other.

But… Do not want to reveal unrequited love.

Input = 1 : I love you
Input = 0: I love you … as a friend

Must compute $F(X,Y)=X \wedge Y$, giving $F(X,Y)$ to both players.

Can we reveal the answer without revealing the inputs?

The Spoiled Children Problem

(AKA The Millionaires Problem [Yao])

Pearl wants to know whether she has more toys than Gersh,
Doesn't want to tell Gersh anything.

Gersh is willing for Pearl to find out who has more toys,
Doesn't want Pearl to know how many toys he has.

Can we give Pearl the information she wants, and nothing else, without giving Gersh any information at all?

Auctions with Private Bids

Auction with private bids:

Normal auction: Players reveal bids – high bid is identified along with high bidders.

Bids are made to the system, but kept private

Only the winning bid, bidders are revealed.

Drawback: Revealing the losing bids gives away strategic information that bidders and auctioneers might exploit in later auctions.

Can we have private bids where no one, not even the auctioneer, knows the losing bids?

Electronic Voting

Final Tally: War: 2
Peace: 2
Nader: 1

The winner is: War

Secure Computation

(Yao, Goldreich-Micali-Wigderson)

X_1 X_2 X_3 X_4 X_5

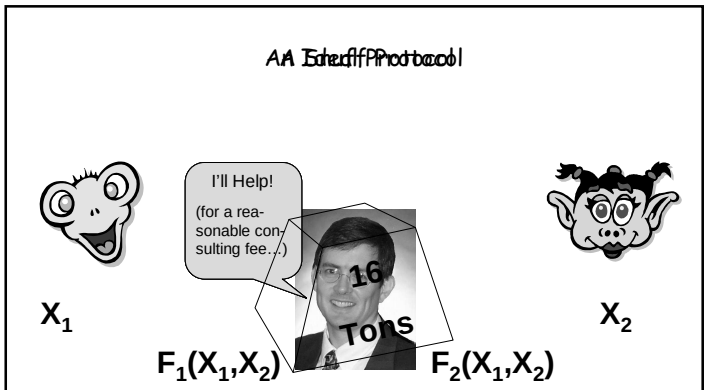
$F_1(X_1,\ldots,X_5)$ $F_2(X_1,\ldots,X_5)$ $F_3(X_1,\ldots,X_5)$ $F_4(X_1,\ldots,X_5)$ $F_5(X_1,\ldots,X_5)$

Players: $1,\ldots,N$

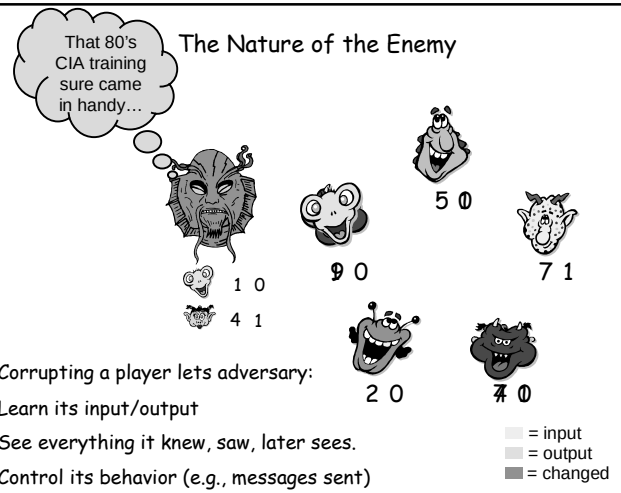
Inputs: $X_1,\ldots,X_N$

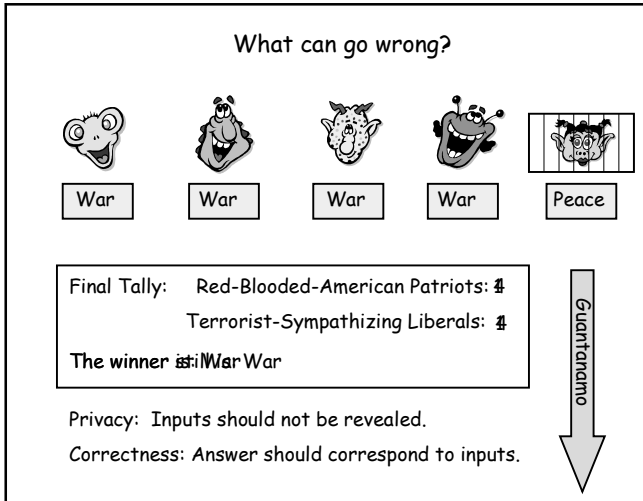
Outputs: $F_1(X_1,\ldots,X_N),\ldots,F_N(X_1,\ldots,X_N)$

Players should learn correct outputs and nothing else.



Goal: Implement something that "looks like" ideal protocol.





What We Can/Can't Hope For

Corrupted players have no privacy on inputs/outputs.

Outputs may reveal inputs:

If candidate received 100% of the votes,

we know how you voted.

Cannot complain about adversary learning what it can by (independently) selecting its inputs and looking at its outputs.

Cannot complain about adversary altering outcome solely by (independently) altering its inputs.

Goal is to not allow the adversary to do anything else.

Definitions very subtle: Beaver, Micali-Rogaway, Canetti…

Can We Do It?

Yao (GMW,GV,K,…):

Yes (for two party case)!*

Cryptographic solutions require "reasonable assumptions"

e.g., hardness of factoring

*Slight issues about both players getting answer at same time.

Goldreich-Micali-Wigderson (BGW,CCD,RB,Bea,…):

Yes, if number of parties corrupted is less than some constant fraction of the total number of players (e.g., <n/2, <n/3).

No hardness assumptions necessary.

As long as functions are computable in polynomial time, solutions require polynomial computation, communication.

Can We **Really** Do It?

General solutions as impractical as they are beautiful.

Step 1:

Break computations to be performed into itsy-bitsy steps. (additions, multiplications, bitwise operations)

Step 2:

For each operation...

Step 3:

Despair at how many itsy-bitsy steps your computation takes.

Is there any hope?

Signs of Hope

Sometimes, don't need too many itsy-bitsy operations.

Naor-Pinkas-Sumner

Functions computed when running auctions are simple.

Highly optimize Yao-like constructions.

Testing if two strings are equal is very practical.

Can exploit algebraic structure to minimize work.

Rabin: Can compute sums very efficiently

Electronic Voting

Most extensively researched subarea of secure computation.

Protocols are now very practical.

100,000 voters a piece of cake,

1,000,000 voters doable.

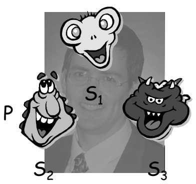
Several commercial efforts

Chaum, Neff, NEC,…

Many interesting issues, both human and technical:

What should our definitions be?

Distributed Cryptographic Entities



Trusted public servant cheerfully encrypts, decrypts, signs messages, when appropriate.

Blakley, Shamir, Desmedt-Frankel…
[illegible]

- Can break secret key up among several entities,
- Can still encrypt, decrypt, sign,
- Remains secure even if a few parties are corrupted.

And Sometimes There's Magic

Chor-Goldreich-Kushilevitz-Sudan,…,Kushilevitz-Ostrovsky,…

Private information retrieval:

Can you download a data entry from a repository without letting the repository know what you're interested in?

Solution: Download everything.

Much more efficient solutions possible!

Data Repository

- The Empire Strikes
- Rabid Liberalism for Dummies
- Cooking with Ricin
- Applied Cryptology
- Flaming 101
- How I Stole the Election

Conclusions

Secure computation is an extremely powerful framework.

Very rich general theory.

A few applications now ready for prime time.

Keep watching this space!