

## CSE 199, Projects/Research

- Individual enrollment
- Projects / research, individual or small group
- Implementation or theoretical
- Weekly one-on-one meetings, no lectures
- Course grade based on project report and presentation

# More crypto and security

## CSE 207

- Graduate introduction to cryptography
- Focus on provable-security
- Material overlaps with 107 but changes in focus as above

## CSE 127, Savage / Shacham

- Undergraduate introduction to security
- Systems issues

## CSE 227, Savage/Shacham

- Graduate introduction to security

CSE 208 is a “Topics” course whose content varies from year to year.  
Some past topics:

- Program obfuscation
- Pairing-based cryptography
- Lattices in cryptography
- Zero-knowledge
- Electronic payments

# Grad school?

- MS: 2 years
- PhD: 5+ years, research-orientated

You can consider a PhD if you enjoy research: solving new problems, exploring the unknown.

An MS is to gain expertise.

# PhD application process

- Applications for Fall due in previous winter
- PhD students are usually fully funded through fellowships, RA-ships and TA-ships.
- Admission to top schools is very competitive but there are niche schools that are excellent in specific areas
- To get admitted to a good program you need a high GPA, strong recommendation letters and a convincing statement of purpose.
- Best of all: accomplished research!

# APPLICATIONS AND PROTOCOLS

# Some applications and protocols

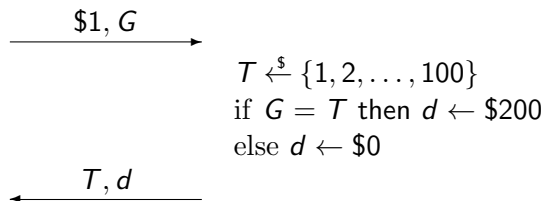
- Internet Casino
  - Commitment
  - Shared coin flips
  - Threshold cryptography
  - Forward security
  - Program obfuscation
  - Zero-knowledge
  - Certified e-mail
  - Electronic voting
  - Auctions
- Identity-based encryption
  - Functional encryption
  - Fully-homomorphic encryption
  - Searchable encryption
  - Oblivious transfer
  - Garbling schemes
  - Secure computation
  - Group signatures
  - Aggregate signatures



# Internet Casino: Protocol G1

Player

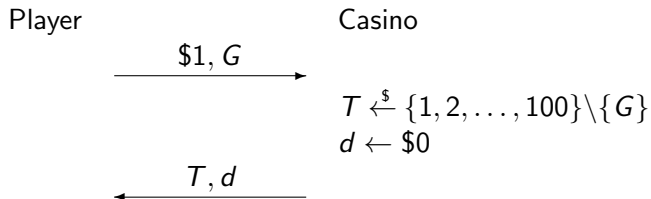
Casino



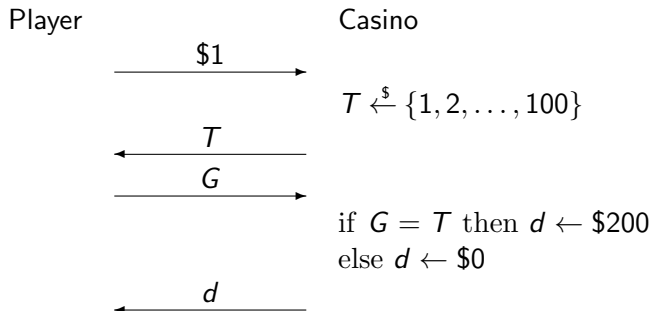
Would you play?

Expected value of  $d$  is  $\$200(\frac{1}{100}) = \$2 > \$1$  so probability theory says that the player will earn money by playing.

# Problem: Casino can cheat



# Internet Casino: Protocol G2



But now player can always win by setting  $G = T$ . No casino would do this!

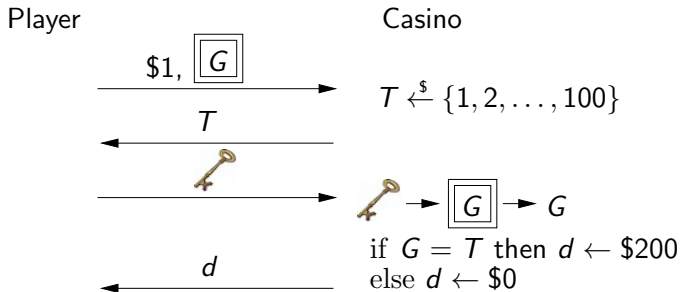
# Internet Casino problem

Player and Casino need to exchange  $G$ ,  $T$  so that

- Casino cannot choose  $T$  as a function of  $G$ .
- Player cannot choose  $G$  as a function of  $T$ .

How do we resolve this Catch-22 situation?

# "Internet" Casino: Protocol G3

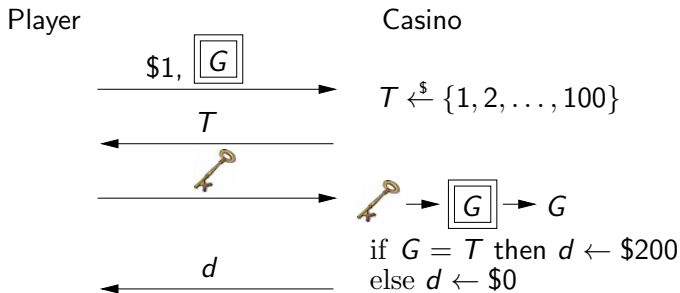


is a locked safe containing a piece of paper with  $G$  written on it.



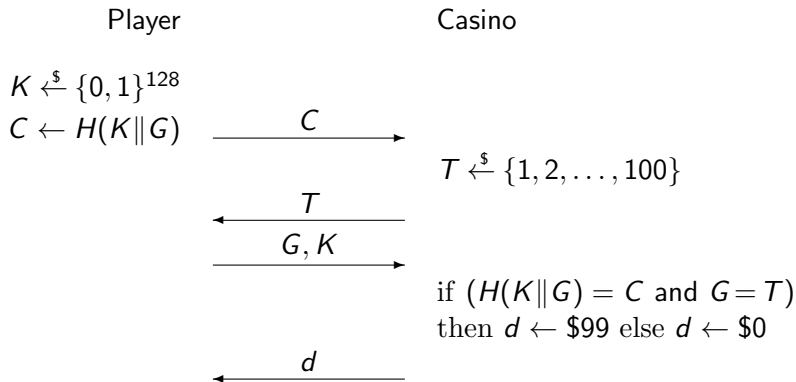
is a key to open the safe.

# "Internet" Casino: Protocol G3



- Casino cannot choose  $T$  as a function of  $G$  because, without the key, it cannot see  $G$ .
- Player cannot choose  $G$  as a function of  $T$  because, by putting it in the safe, she is committed to it in the first move.

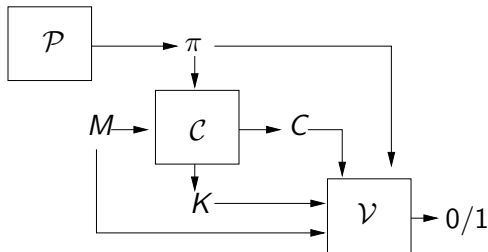
# Internet Casino Protocol using cryptography





Here  $H$  is a cryptographic hash function. More generally one can use a primitive called a *commitment scheme*.

# Commitment Schemes

A commitment scheme  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  is a triple of algorithms



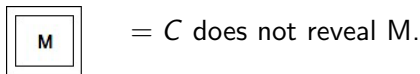
Parameter generation algorithm  $\mathcal{P}$  is run once by a trusted party to produce public parameters  $\pi$ .

		In Internet Casino
M	Data being committed	G
C	Comital	
K	Decomital key	

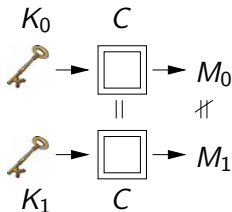


# Security properties

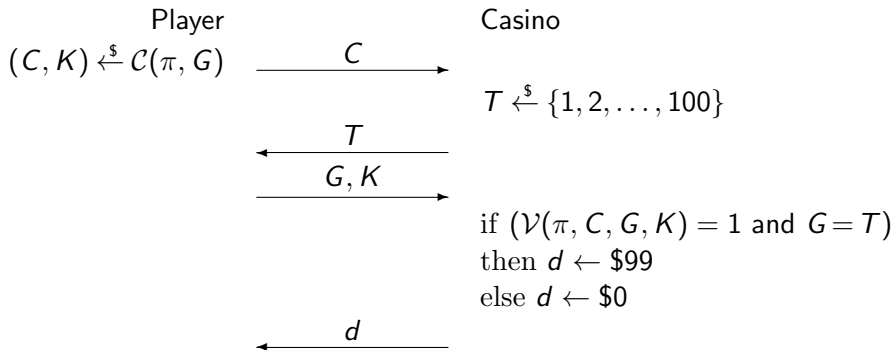
- Hiding: A commital  $C$  generated via  $(C, K) \xleftarrow{\$} \mathcal{C}(\pi, M)$  should not reveal information about  $M$ .



- Binding: It should be hard to find  $C, M_0, M_1, K_0, K_1$  such that  $M_0 \neq M_1$  but  $\mathcal{V}(\pi, C, M_0, K_0) = \mathcal{V}(\pi, C, M_1, K_1) = 1$ .



# Internet Casino Protocol using a commitment scheme



# Hiding Formally

Let  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  be a commitment scheme and  $A$  an adversary.

Game  $\text{HIDE}_{\mathcal{CS}}$

**procedure Initialize**

$\pi \xleftarrow{\$} \mathcal{P}; b \xleftarrow{\$} \{0, 1\}$

return  $\pi$

**procedure LR**( $M_0, M_1$ )

$(C, K) \xleftarrow{\$} \mathcal{C}(\pi, M_b)$

return  $C$

**procedure Finalize**( $b'$ )

return  $(b = b')$

The hiding-advantage of  $A$  is

$$\mathbf{Adv}_{\mathcal{CS}}^{\text{hide}}(A) = 2 \cdot \Pr \left[ \text{HIDE}_{\mathcal{CS}}^A \Rightarrow \text{true} \right] - 1.$$

# Binding Formally

Let  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  be a commitment scheme and  $A$  an adversary.

**Game**  $\text{BIND}_{\mathcal{CS}}$

**procedure Initialize**

$\pi \xleftarrow{\$} \mathcal{P}$

return  $\pi$

**procedure Finalize**( $C, M_0, M_1, K_0, K_1$ )

$v_0 \leftarrow \mathcal{V}(\pi, C, M_0, K_0)$

$v_1 \leftarrow \mathcal{V}(\pi, C, M_1, K_1)$

return ( $v_0 = v_1 = 1$  and  $M_0 \neq M_1$ )

The binding-advantage of  $A$  is

$$\mathbf{Adv}_{\mathcal{CS}}^{\text{bind}}(A) = \Pr \left[ \text{BIND}_{\mathcal{CS}}^A \Rightarrow \text{true} \right].$$

# Commitment from symmetric encryption

Let  $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$  be an IND-CPA-secure symmetric encryption scheme and let  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  be the commitment scheme where  $\mathcal{P}$  returns  $\pi = \varepsilon$  and

<u><b>Alg</b> <math>\mathcal{C}(\pi, M)</math></u>	<u><b>Alg</b> <math>\mathcal{V}(\pi, C, M, K)</math></u>
$K \xleftarrow{\$} \mathcal{K}; C \xleftarrow{\$} \mathcal{E}_K(M)$	if $\mathcal{D}_K(C) = M$ then return 1
return $(C, K)$	else return 0

Is this secure?

# Commitment from symmetric encryption

Let  $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$  be an IND-CPA-secure symmetric encryption scheme and let  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  be the commitment scheme where  $\mathcal{P}$  returns  $\pi = \varepsilon$  and

<u><b>Alg</b> <math>\mathcal{C}(\pi, M)</math></u>	<u><b>Alg</b> <math>\mathcal{V}(\pi, C, M, K)</math></u>
$K \xleftarrow{\$} \mathcal{K}; C \xleftarrow{\$} \mathcal{E}_K(M)$	if $\mathcal{D}_K(C) = M$ then return 1
return $(C, K)$	else return 0

Is this secure?

- It is certainly hiding.
- But need not be binding: it may be possible to find  $C, M_0, M_1, K_0, K_1$  such that

$$\mathcal{D}_{K_0}(C) = M_0 \text{ and } \mathcal{D}_{K_1}(C) = M_1$$

For example this is easy when  $\mathcal{SE}$  is CBC\$ encryption.

# Commitment from public key encryption

Let  $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$  be an IND-CPA-secure asymmetric encryption scheme and let  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  be the commitment scheme where

<b>Alg</b> $\mathcal{P}$	<b>Alg</b> $\mathcal{C}(pk, M)$	<b>Alg</b> $\mathcal{V}(pk, C, M, K)$
$(pk, sk) \xleftarrow{\$} \mathcal{K}$	$K \xleftarrow{\$} \{0, 1\}^k$	if $\mathcal{E}_{pk}(M; K) = C$ then
$\pi \leftarrow pk$	$C \leftarrow \mathcal{E}_{pk}(M; K)$	return 1
return $\pi$	return $(C, K)$	else return 0

$\mathcal{E}_{pk}(M; K)$  means encryption of  $M$  with coins  $K$ .

# Commitment from public key encryption

Let  $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$  be an IND-CPA-secure asymmetric encryption scheme and let  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  be the commitment scheme where

<u>Alg <math>\mathcal{P}</math></u>	<u>Alg <math>\mathcal{C}(pk, M)</math></u>	<u>Alg <math>\mathcal{V}(pk, C, M, K)</math></u>
$(pk, sk) \xleftarrow{\$} \mathcal{K}$	$K \xleftarrow{\$} \{0, 1\}^k$	if $\mathcal{E}_{pk}(M; K) = C$ then
$\pi \leftarrow pk$	$C \leftarrow \mathcal{E}_{pk}(M; K)$	return 1
return $\pi$	return $(C, K)$	else return 0

$\mathcal{E}_{pk}(M; K)$  means encryption of  $M$  with coins  $K$ .

- Certainly hiding.
- Binding too since  $C$  has only one decryption relative to  $pk$ , namely  $M = \mathcal{D}_{sk}(C)$ .



# Commitment from hashing

Let  $H$  be a hash function and  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  the commitment scheme where  $\mathcal{P}$  returns  $\pi = \varepsilon$  and

$$\begin{array}{l|l} \textbf{Alg } \mathcal{C}(\pi, M) & \textbf{Alg } \mathcal{V}(\pi, C, M, K) \\ \hline C \leftarrow H(M); K \leftarrow M & \text{return } (C = H(M) \text{ and } M = K) \\ \text{return } (C, K) & \end{array}$$

This is

# Commitment from hashing

Let  $H$  be a hash function and  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  the commitment scheme where  $\mathcal{P}$  returns  $\pi = \varepsilon$  and

$$\frac{\text{Alg } \mathcal{C}(\pi, M)}{C \leftarrow H(M); K \leftarrow M \quad \text{return } (C, K)} \quad \left| \quad \frac{\text{Alg } \mathcal{V}(\pi, C, M, K)}{\text{return } (C = H(M) \text{ and } M = K)}$$

This is

- Binding if  $H$  is collision-resistant.
- But not hiding. For example in the Internet Casino  $M = G \in \{1, \dots, 100\}$  so given  $C = H(M)$  the casino can recover  $M$  via

for  $i = 1, \dots, 100$  do  
    if  $H(i) = C$  then return  $i$

# Commitment from hashing

A better scheme is  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  where  $\mathcal{P}$  returns  $\pi = \varepsilon$  and

<b>Alg</b> $\mathcal{C}(\pi, M)$	
$K \xleftarrow{\$} \{0, 1\}^{128}$	<b>Alg</b> $\mathcal{V}^H(\pi, C, M, K)$
$C \leftarrow H(K  M)$	return $(H(K  M) = C)$
return $(C, K)$	

# Commitment schemes usage

Commitment schemes are very broadly and widely used across all kinds of protocol design and in particular to construct zero-knowledge proofs.

# Flipping a common coin

- Alice and Bob are getting divorced
- They want to decide who keeps the Lexus
- They agree to flip a coin, but
- Alice is in NY and Bob is in LA

Protocol CF1:



# Flipping a common coin

- Alice and Bob are getting divorced
- They want to decide who keeps the Lexus
- They agree to flip a coin, but
- Alice is in NY and Bob is in LA

Protocol CF1:

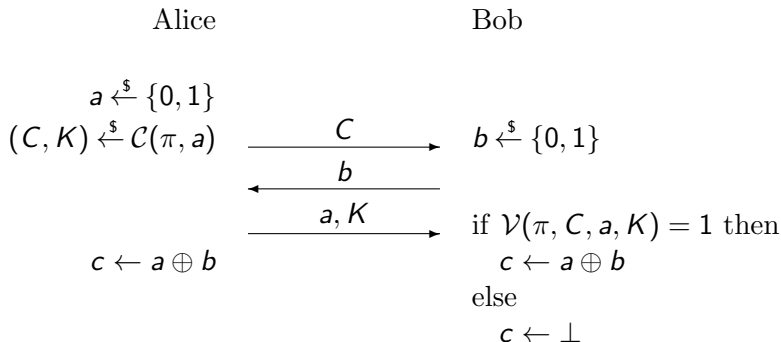


Bob is not too smart but he doesn't like it...

Can you help them out?

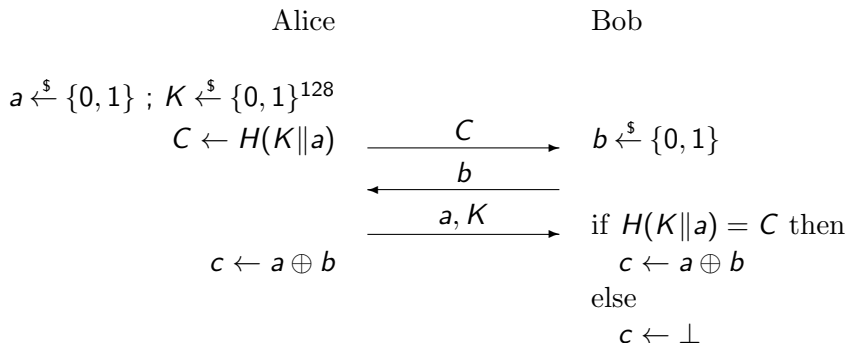
# Protocol CF2

Let  $\mathcal{CS} = (\mathcal{P}, \mathcal{C}, \mathcal{V})$  be a commitment scheme.



$c$  is the common coin. Neither party can control it.

## Protocol CF3: Concrete instantiation of CF2



$c$  is the common coin. Neither party can control it.  $H$  is a cryptographic hash function.



# Secure summation

Suppose we have  $n$  parties  $1, \dots, n$

Party  $i$  has an integer  $x_i$

The parties want to know the value of

$$f(x_1, \dots, x_n) = x_1 + \dots + x_n$$

# Secure summation

Suppose we have  $n$  parties  $1, \dots, n$

Party  $i$  has an integer  $x_i$

The parties want to know the value of

$$f(x_1, \dots, x_n) = x_1 + \dots + x_n$$

Easy: Let

- Party  $i$  send  $x_i$  to party 1 ( $2 \leq i \leq n$ )
- Party 1 computes  $f(x_1, \dots, x_n) = x_1 + \dots + x_n$  and broadcasts it

# Secure summation

Suppose we have  $n$  parties  $1, \dots, n$

Party  $i$  has an integer  $x_i$

The parties want to know the value of

$$f(x_1, \dots, x_n) = x_1 + \dots + x_n$$

Easy: Let

- Party  $i$  send  $x_i$  to party 1 ( $2 \leq i \leq n$ )
- Party 1 computes  $f(x_1, \dots, x_n) = x_1 + \dots + x_n$  and broadcasts it

**What they don't like about this:** Party 1 now knows everyone's values

**Privacy constraint:** Party  $i$  does not wish to reveal  $x_i$

# Secure summation

Party  $i$  has input  $x_i$  ( $1 \leq i \leq n$ ). The parties want to know  $f(x_1, \dots, x_n) = x_1 + \dots + x_n$  but do not want to reveal their inputs in the process.

Scenarios:

- $x_i$  = score of student  $i$  on midterm exam
- $x_i$  = salary of employee  $i$
- $x_i \in \{0, 1\}$  = vote of voter  $i$  on proposition  $X$  on ballot

# The model and goal

Parties  $i, j$  are connected via a secure channel ( $1 \leq i, j \leq n$ ).

Privacy and authenticity of messages sent over channel are guaranteed.

The parties will exchange messages to arrive at  $f(x_1, \dots, x_n)$ .

If  $i \neq j$  then, at the end of the protocol, party  $i$  should not know  $x_j$ .

For example you, as player  $i$ , enter  $x_i$  into some app on your cellphone which then communicates with the cellphones of the other parties. At the end, the sum shows up on your screen. Take your phone apart and examine all memory contents and you still will not discover  $x_j$  for  $j \neq i$ .

# Setup for secure communication protocol

Let  $N$  be such that  $x_1, \dots, x_n \in Z_N = \{0, \dots, N-1\}$ .

Let  $M = nN$ .

Let  $S$  denote  $x_1 + \dots + x_n$ .

We will compute  $S \bmod M$ , which is just  $S$  since

$$x_1 + \dots + x_n \leq n(N-1) < M$$

# Protocol step 1: secret sharing

For  $i = 1, \dots, n$  party  $i$

- Picks  $x_{i,1}, \dots, x_{i,n} \in Z_M$  at random subject to  $x_{i,1} + \dots + x_{i,n} \equiv x_i \pmod{M}$
- Sends  $x_{i,j}$  to party  $j$  over secure channel ( $1 \leq j \leq n$ )

$$\begin{array}{cccc} \left[ \begin{array}{cccc} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \end{array} \right] & \rightarrow & x_1 \\ & & \rightarrow & x_2 \\ & & \rightarrow & x_3 \\ & & \rightarrow & x_4 \end{array}$$

Observation:  $x_{i,j}$  is a random number unrelated to  $x_i$  so party  $j$  has no information about  $x_i$  ( $i \neq j$ )

## Protocol step 2,3: Column sums and conclusion

$$\begin{array}{ccccccc} \left[ \begin{array}{cccc} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \end{array} \right] & \rightarrow & x_1 \\ & & \rightarrow & x_2 \\ & & \rightarrow & x_3 \\ & & \rightarrow & x_4 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ C_1 & & C_2 & & C_3 & & C_4 \end{array}$$

For  $j = 1, \dots, n$  party  $j$

- Computes  $C_j = (x_{1,j} + x_{2,j} + \dots + x_{n,j}) \bmod M$
- Sends  $C_j$  to party  $i$  ( $1 \leq i \leq n$ )

Observation:  $S \equiv (C_1 + \dots + C_n) \pmod{M}$ .

So each party can compute  $S \leftarrow (C_1 + \dots + C_n) \bmod M$



# Security of the protocol

$$\begin{array}{ccccccc} \left[ \begin{array}{cccc} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \end{array} \right] & \rightarrow & x_1 \\ & & \rightarrow & x_2 \\ & & \rightarrow & x_3 \\ & & \rightarrow & x_4 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ c_1 & & c_2 & & c_3 & & c_4 \end{array}$$

At end of protocol, party 1 knows (1)  $x_1$ , and the first-row entries of the matrix (2) the sum  $S = x_1 + x_2 + x_3 + x_4$  (3)  $c_1, c_2, c_3, c_4$  (4) the first column entries  $x_{1,1}, x_{2,1}, x_{3,1}, x_{4,1}$ .

## Claims:

- Party 1 learn nothing about  $x_4$
- Even if parties 1, 2 pool their information, they learn nothing about  $x_4$
- ...

**Project:** Analyze and prove secure the summation protocol: (1) Give a game based definition of privacy (2) Prove that the protocol meets it.

# Secure Computation

Parties  $1, \dots, n$

Party  $i$  has private input  $x_i$

They want to compute  $f(x_1, \dots, x_n)$

**Fact:** For any function  $f$ , there is a  $n/2$  - private protocol to compute it.

A protocol is  $t$ -private if any  $t$  parties, getting together, cannot figure out anything about the input of the other parties other than implied by the value of  $f(x_1, \dots, x_n)$ .

The protocol views  $f$  as a circuit (program) and computes it gate (instruction) by gate (instruction).

Enormous body of research.

# Zero-Knowledge Proofs [GMR]

A zero-knowledge (ZK) proof allows you to

- Convince Bob your claim is true
- Without revealing anything beyond that

For example:

<b>You claim to have</b>	<b>Bob is</b>	<b>What is not revealed</b>
A solution to the homework problem	Another student	The solution
The password for this account	The server	The password
A proof that $\mathbf{P} \neq \mathbf{NP}$	The Clay Institute	The proof