HOW TO SOLVE ANY PROTOCOL PROBLEM

(OR HOW TO PLAY ANY MENTAL GAME)

by

ODED GOLDREICH
COMPUTER SCIENCE DEPT.
TECHNION, ISRAEL

BASED ON WORKS WITH

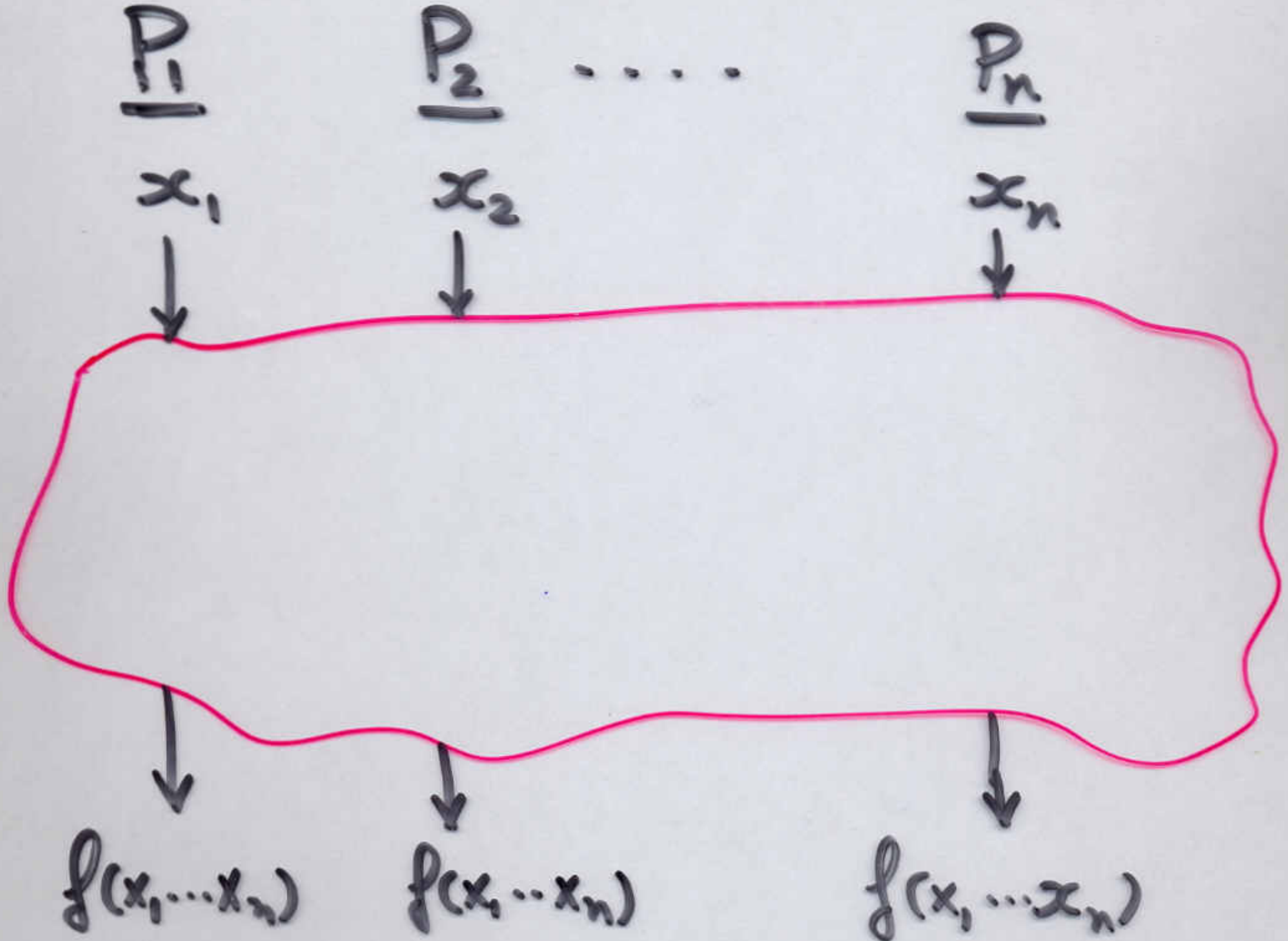
(1) MICALI AND WIGDERSON

(2) VAINISH

MOTIVATION

$$f : D \times D \times \cdots \times D$$

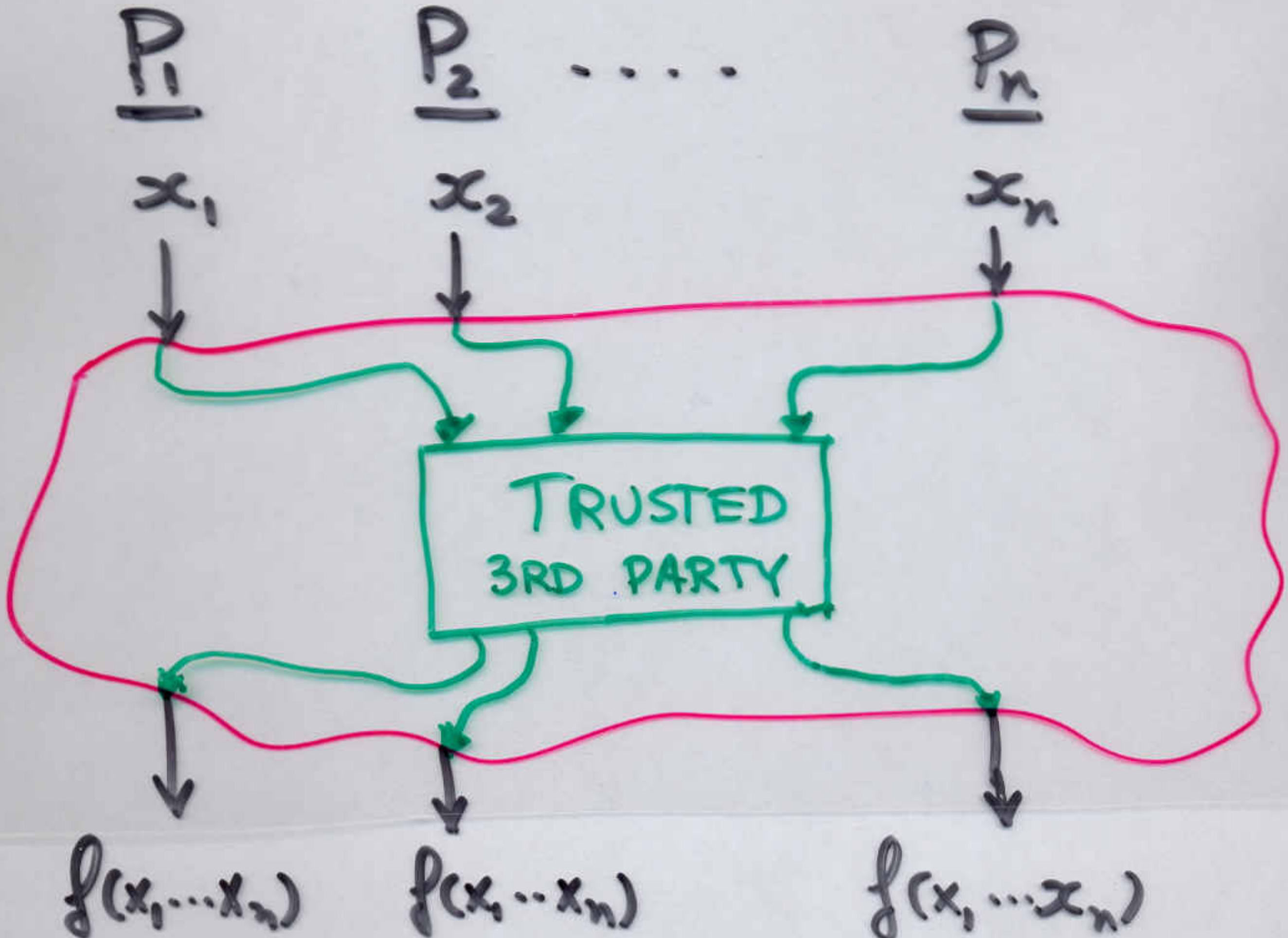
n-ARY FUNCTION



MOTIVATION

$$f : D \times D \times \cdots \times D$$

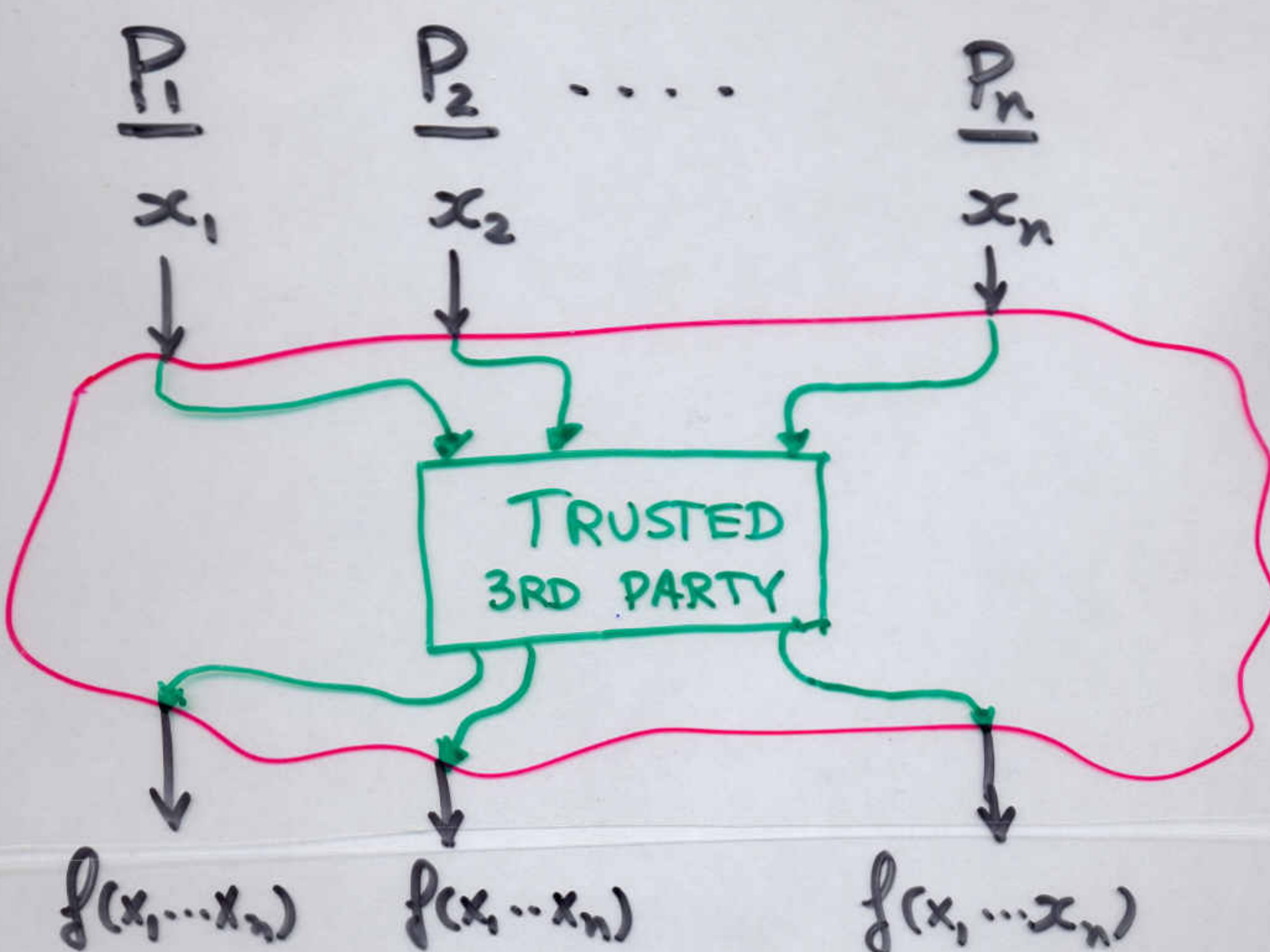
n-ARY FUNCTION



MOTIVATION

$$f : D \times D \times \cdots \times D$$

n-ARY FUNCTION



ISSUES

- (1) CORRECTNESS →
  - SIM. COMMIT.
  - FORCING PROPER EXEC.
- (2) PRIVACY

FORMAL SETTING – PRELIMINARIES

- A PROTOCOL PROBLEM IS
  AN n-ARY FUNCTION, f.

- MODEL (FOR SOLUTIONS)
  A COMPUTATION IS CALLED EFFICIENT
  IF IT IS COMPLETED IN TIME
  POLYNOMIAL IN THE COMPLEXITY OF f.

- A SOLUTION TO THE PROTOCOL PROBLEM f
  IS AN EFFICIENT PROTOCOL GUARANTEEING
  "SIMULTANOUS COMMITMENT", "CORRECTNESS",
  AND "PRIVACY" IN PRESENCE OF $< n/2$
  FAULTY BUT FEASIBLE PROCESSORS.

FORMAL SETTING – ESSENCE

A SOLUTION TO THE PROTOCOL PROBLEM f IS AN EFFICIENT FAULT-TOLERANT PROTOCOL GUARANTEEING (W.R.T $< n/2$ FEASIBLE FAULTS):

(1) SIMULTANOUS COMMITMENT TO INITIAL VALUES.

(2) (CORRESPONDINGLY) CORRECT OUTPUT VALUES.

(3) MAXIMUM PRIVACY OF THE INITIAL VALUES.

EXAMPLE

$$\sum_{i=1}^{n} x_i$$

[Coh.]

OUR RESULT

ASSUMING EXISTENCE OF PUBLIC-KEY CRYPTOSYSTEMS, EVERY PROTOCOL PROBLEM HAS A SOLUTION.

FURTHERMORE, WE PRESENT AN EFFICIENT PROTOCOL-GENERATOR THAT ON INPUT A (TM)-DESCRIPTION OF THE PROBLEM OUTPUTS A SOLUTION.

OUR RESULT

ASSUMING EXISTENCE OF PUBLIC-KEY CRYPTOSYSTEMS, EVERY PROTOCOL PROBLEM HAS A SOLUTION.

FURTHERMORE, WE PRESENT AN EFFICIENT PROTOCOL-GENERATOR THAT ON INPUT A (TM)-DESCRIPTION OF THE PROBLEM OUTPUTS A SOLUTION.

E.G.

IF FACTORING INTEGERS IS INFEASIBLE THEN OUR ASSUMPTION HOLDS.

OUR PROTOCOL-GENERATOR

- (INDEPENDENT OF THE PROBLEM,) OUTPUTS A FAULT-TOLERANT PROTOCOL FOR SIMULTANOUS COMMITMENT.

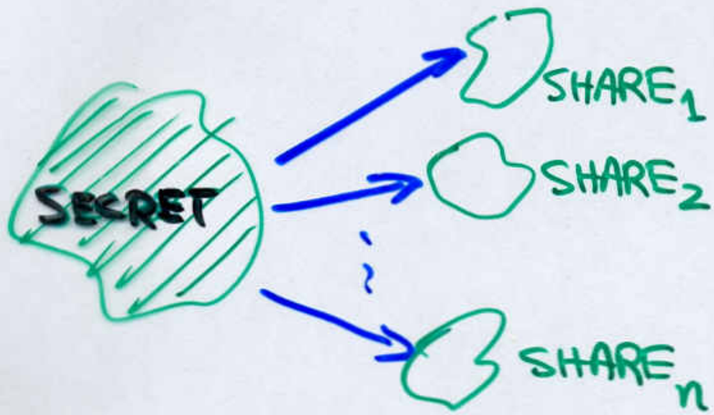
- (1) CONSTRUCT A PROTOCOL, FOR "SEMI-HONEST" PLAYERS, ACHEIVING MAX. PRIVACY.

  (2) COMPILE THIS PROTOCOL TO MAKE IT FAULT-TOLERANT, PRESERVING CORRECTNESS & PRIVACY.

1ST – SIMULTANOUS COMMITMENT

• A KEY NOTION:

SECRET SHARING (SS)



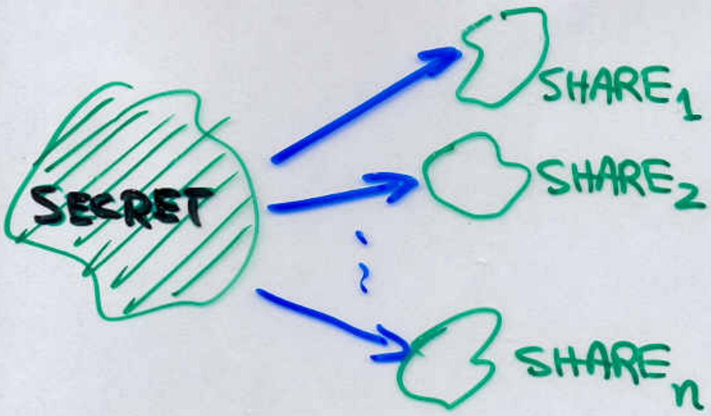
(1) MINORITY OF SHARES YIELDS NOTHING.

(2) MAJORITY OF SHARES YIELDS THE SECRET

1ST — SIMULTANOUS COMMITMENT

- A KEY NOTION: [CGMA]

  VERIFIABLE SECRET SHARING (VSS)



(1) MINORITY OF SHARES YIELDS NOTHING.

(2) MAJORITY OF SHARES YIELDS THE SECRET

(3) VERIFIABILITY OF SHARES!

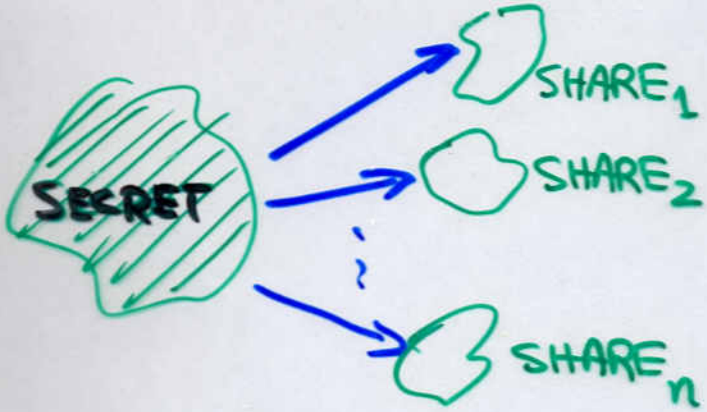
- IMPLEMENTING VSS = [SHAMIR] + [GMW]

- SIMULTANOUS COMMITMENT IS LINEARLY REDUCIBLE TO VSS. [CGMA]

1ST — SIMULTANOUS COMMITMENT

- A KEY NOTION: [CGMA]

  VERIFIABLE SECRET SHARING (VSS)



(1) MINORITY OF SHARES YIELDS NOTHING.

(2) MAJORITY OF SHARES YIELDS THE SECRET

(3) VERIFIABILITY OF SHARES!

- IMPLEMENTING VSS = [SHAMIR] + [GMW]

- SIMULTANOUS COMMITMENT IS LOGARITHMICALLY ~~LINEARLY~~ REDUCIBLE TO VSS. [CR] ~~[CGMA]~~

2ND — MAX. PRIVACY FOR "SEMI-HONEST"

- WHAT IS A SEMI-HONEST PARTY ?

  EXECUTES PROTOCOL PROPERLY,

  BUT RECORDS <u>ALL</u>

  <u>INTERMIDIATE RESULTS</u>.

- A MAX. PRIVACY PROTOCOL FOR f

  WHATEVER CAN BE EFFICIENTLY COMPUTED AFTER PARTICIPATING (SEMI-HONESTLY) IN THE PROTOCOL

  $\overset{?}{=}$

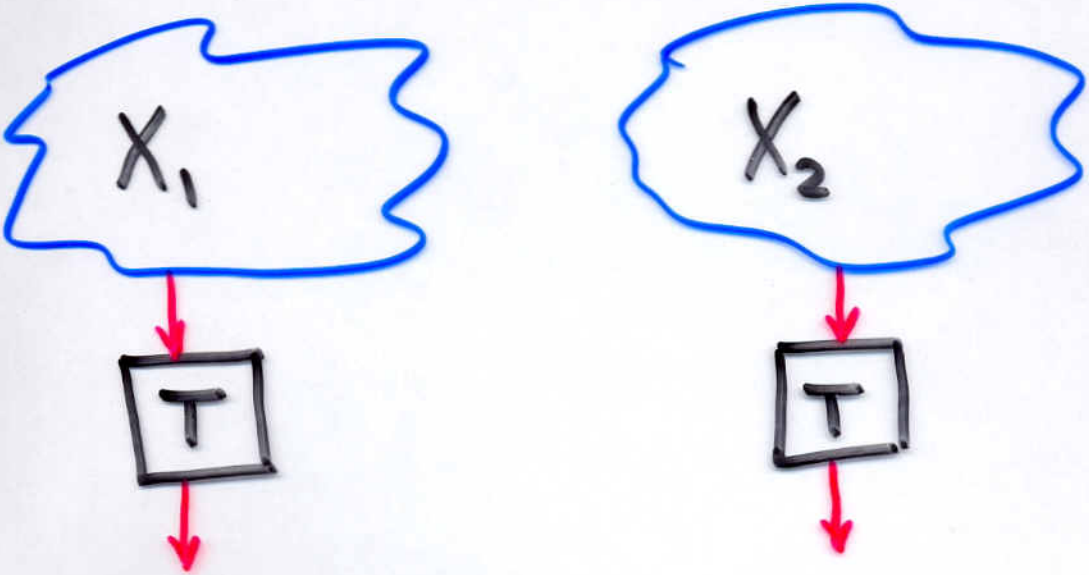
  WHATEVER CAN BE COMPUTED FROM THE PRIVATE INPUTS & OUTPUTS.

- HOW TO CONSTRUCT A MAX. PRIVACY PROTOCOL?

  essence of this work

Polynomial Indistinguishability

[Yao] ([GM])

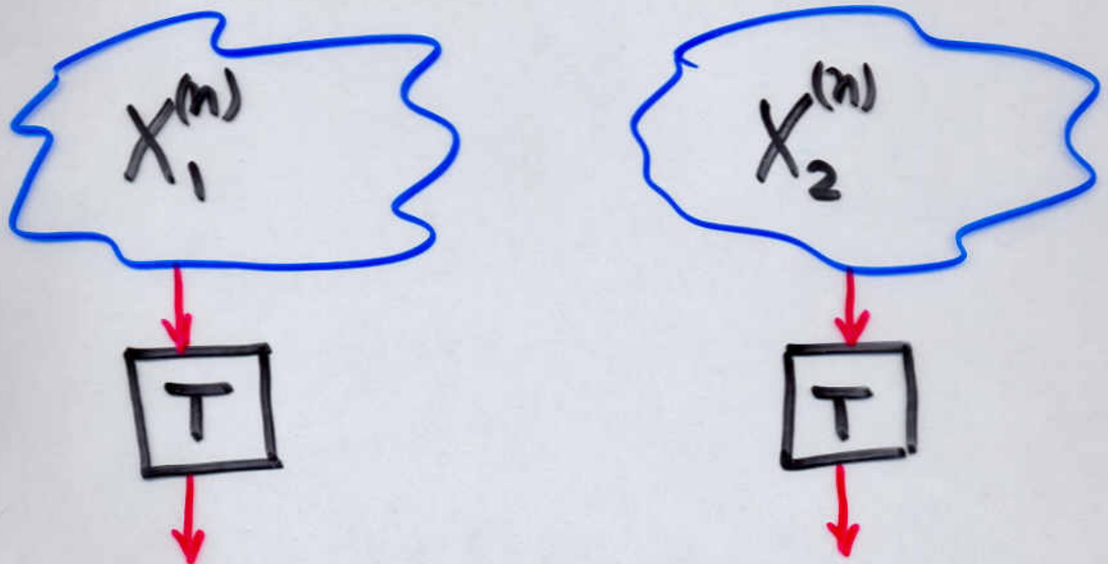


$P_1 \triangleq \mathrm{Prob}(T(X_1)=1)$ $\quad$ $P_2 \triangleq \mathrm{Prob}(T(X_2)=1)$

T DOES <u>NOT DISTINGUISH</u> IF

POLYNOMIAL INDISTINGUISHABILITY

[YAO] ([GM])

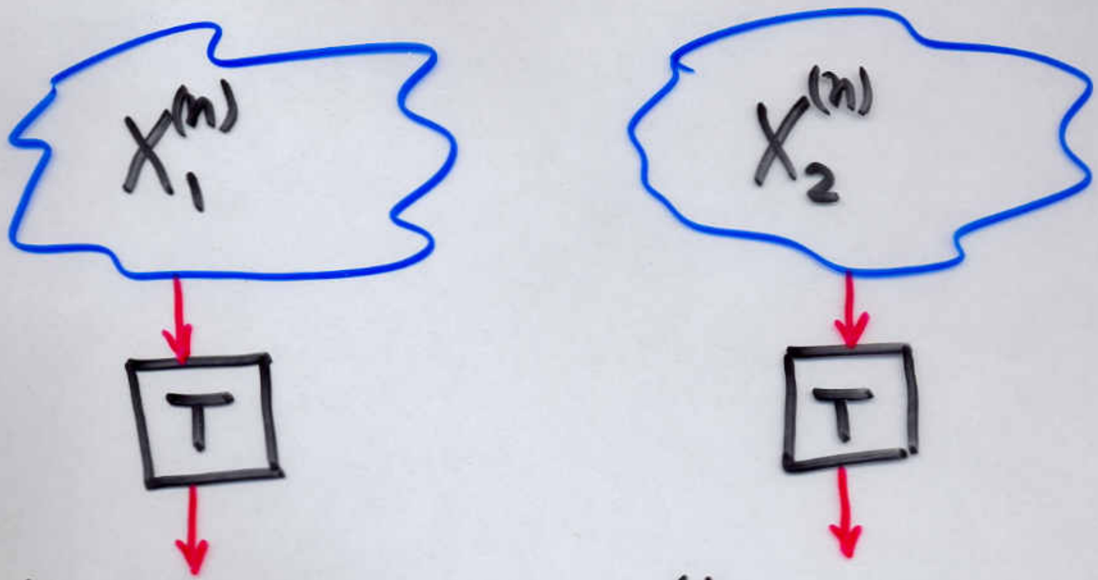


$P_1^{(n)} \triangleq \text{PROB}(T(X_1^{(n)})=1)$ $\quad$ $P_2^{(n)} \triangleq \text{PROB}(T(X_2^{(n)})=1)$

T DOES <u>NOT DISTINGUISH</u> IF

Polynomial Indistinguishability

[Yao] ([GM])



$p_1^{(n)} \triangleq \text{Prob}(T(X_1^{(n)})=1)$ $\qquad$ $p_2^{(n)} \triangleq \text{Prob}(T(X_2^{(n)})=1)$

T does <u>not distinguish</u> if

$\forall c>0,\ |p_1^{(n)} - p_2^{(n)}| < n^{-c}$ ($\forall$ n large enough)

$X_1 = \{X_1^{(n)}\}$ & $X_2 = \{X_2^{(n)}\}$ are

<u>polynomially indistinguishable</u>

if $\forall$ prob. poly-time T

cannot distinguish them.

3RD – FORCING SEMI-HONEST BEHAVIOUR

- IDEA: "APPEND" TO EACH MESSAGE A <u>ZERO-KNOWLEDGE PROOF</u> THAT IT IS COMPUTED PROPERLY.

- A <u>ZERO-KNOWLEDGE PROOF</u> IS A "CONVINCING ARGUMENT" THAT YIELDS <u>NOTHING</u> BUTS THE VALIDITY OF STATMENT.

- ZERO-KNOWLEDGE PROOFS EXIST FOR <u>EVERY</u> NP-STATEMENT [GMW], AND THAT'S ALL WE NEED!

3RD – FORCING SEMI-HONEST BEHAVIOUR

- IDEA : "APPEND" TO EACH MESSAGE
  A ZERO-KNOWLEDGE PROOF
  THAT IT IS COMPUTED PROPERLY.

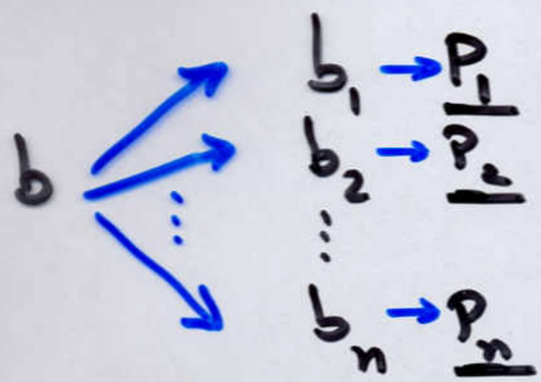
- DETAILS (FOR EXPERTS ONLY):
  (1) TO DEAL WITH RANDOMIZED PROTOCOLS
  WE USE DISTRIBUTED COIN-FLIPPING,
  WHICH IS IMPLEMENTED USING SIM. COMMIT.

  (2) WE NEED AND HAVE
  AUXILIARY-INPUT ZK PROOFS
  FOR EVERY NP-STATEMENT.

MAX. PRIVACY PROTOCOLS FOR SEMI-HONEST

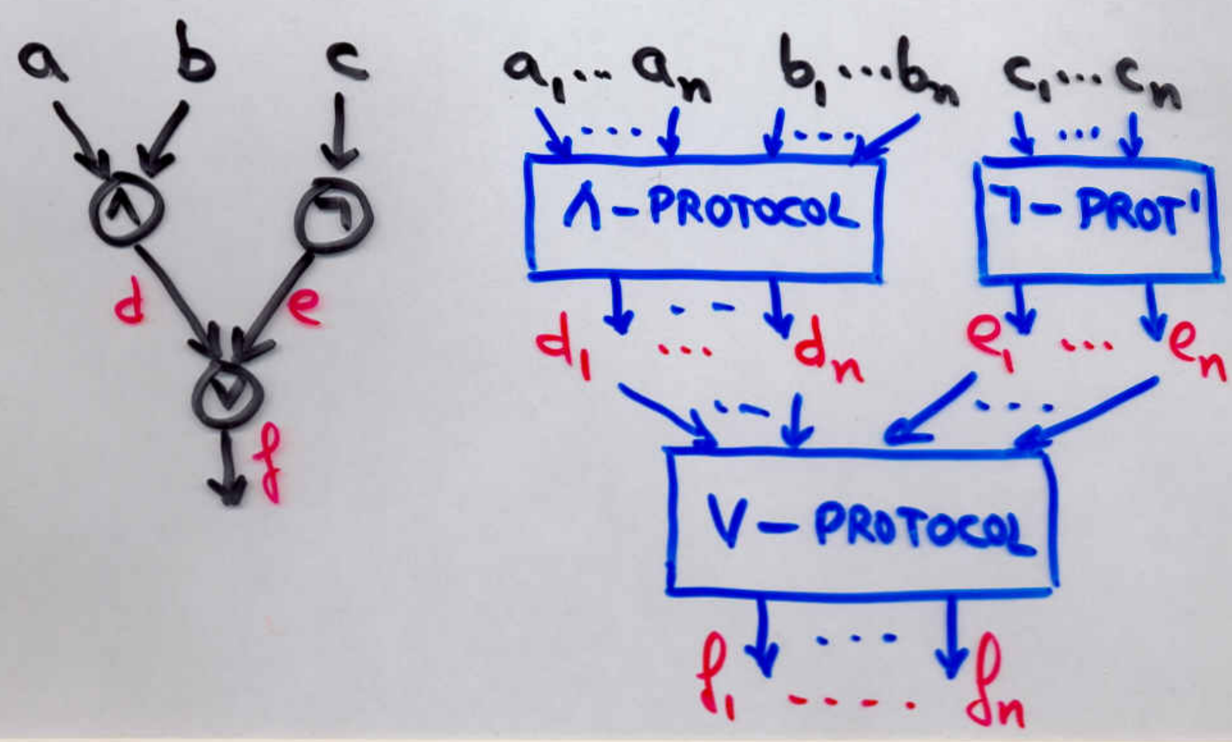
- IDEA: "DISTRIBUTED SIMULATION" OF BOOLEAN CIRCUIT EVALUATION.

- "SHARING" A PRIVATE INPUT

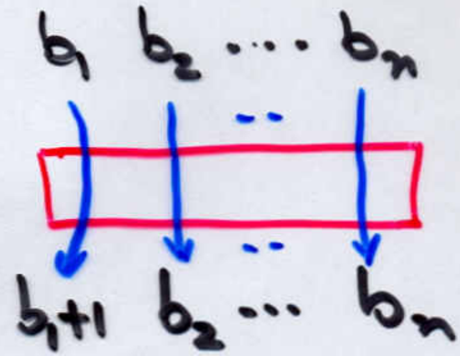


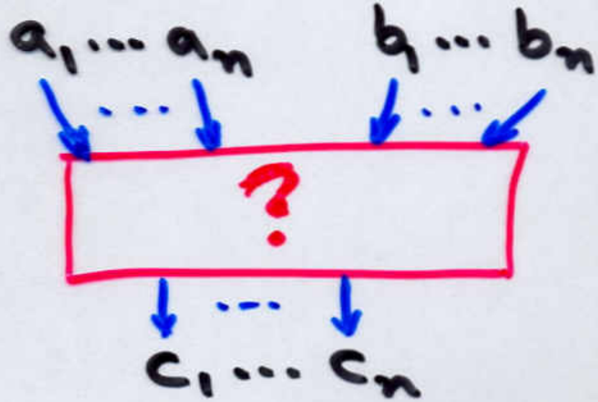
$$b = \bigoplus_i b_i = \sum_i b_i \quad \underline{GF(2)}$$

- THE "SIMULATION"



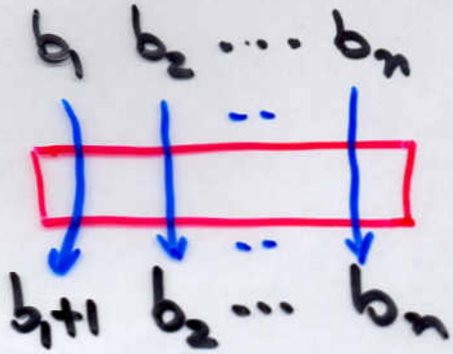
DISTRIBUTED SIMULATION OF AND & NOT

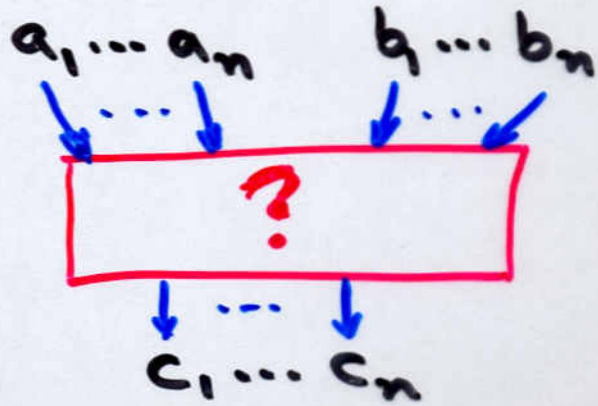




S.T. $\sum c_i = (\sum a_i)\cdot(\sum b_i)$

DISTRIBUTED SIMULATION OF AND & NOT





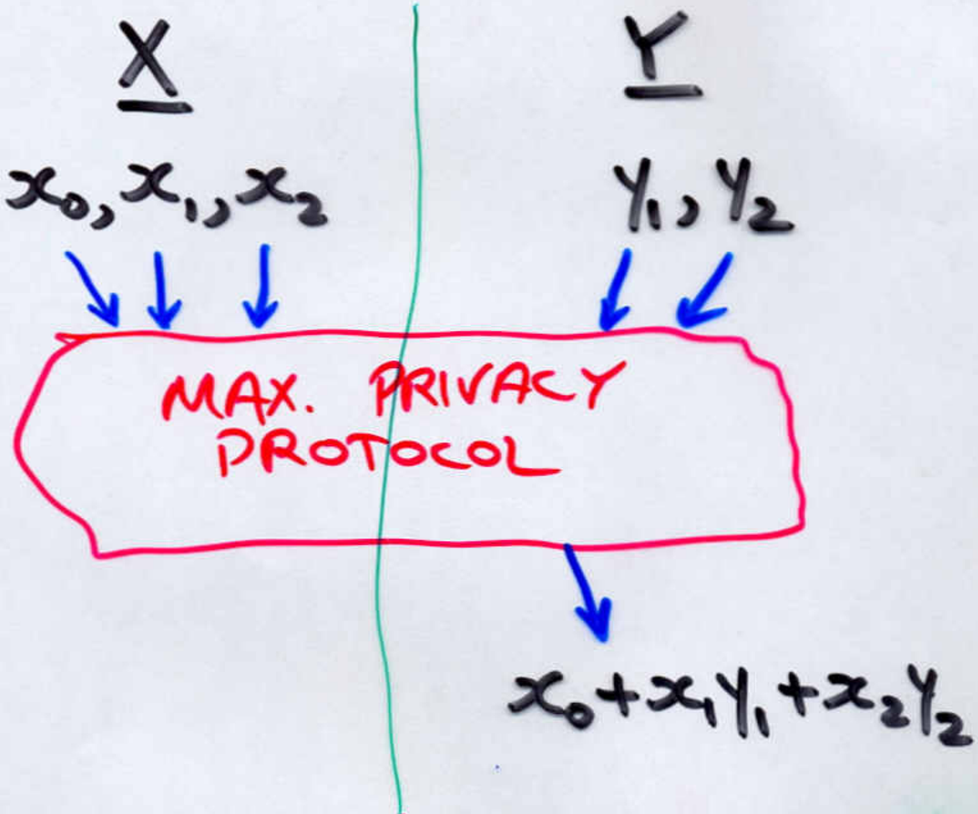
S.T. $\sum c_i = (\sum a_i)\cdot(\sum b_i)$

$$\sum_{i=1}^{n} c_i = \underbrace{\sum_{i=1}^{n} a_i \cdot b_i}_{c_i^{(i)}} + \sum_{i<j} \underbrace{(a_i b_j + a_j b_i)}_{c_i^{(j)} + c_j^{(i)}}$$

easy! how?

A MAX. PRIVACY PROTOCOL FOR $x_1 y_1 + x_2 y_2$

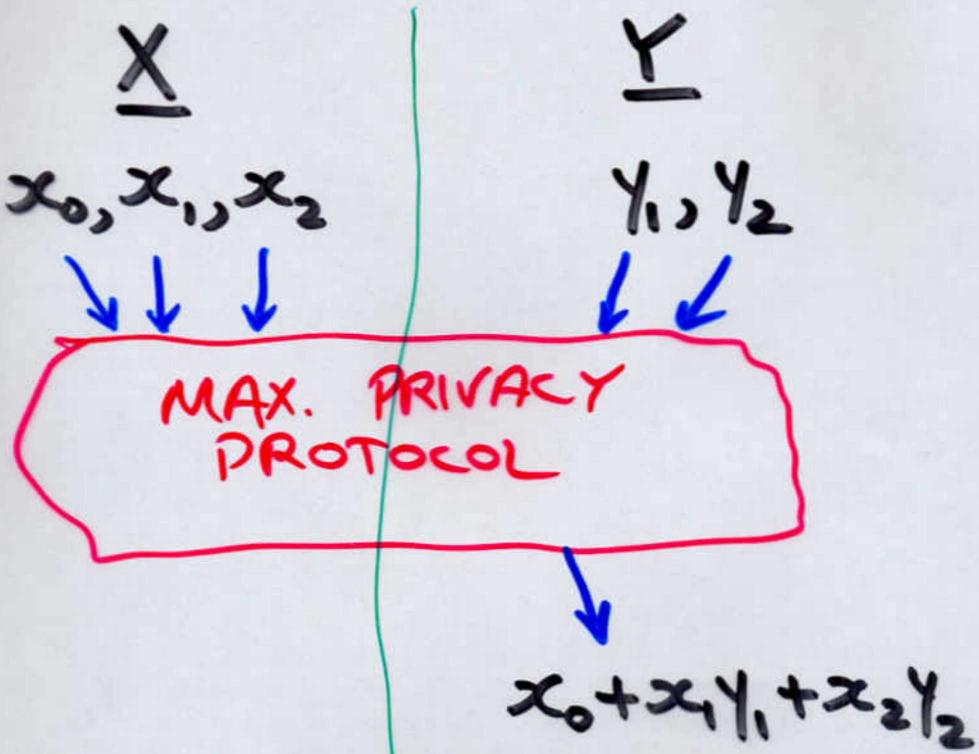
- WHAT WE NEED



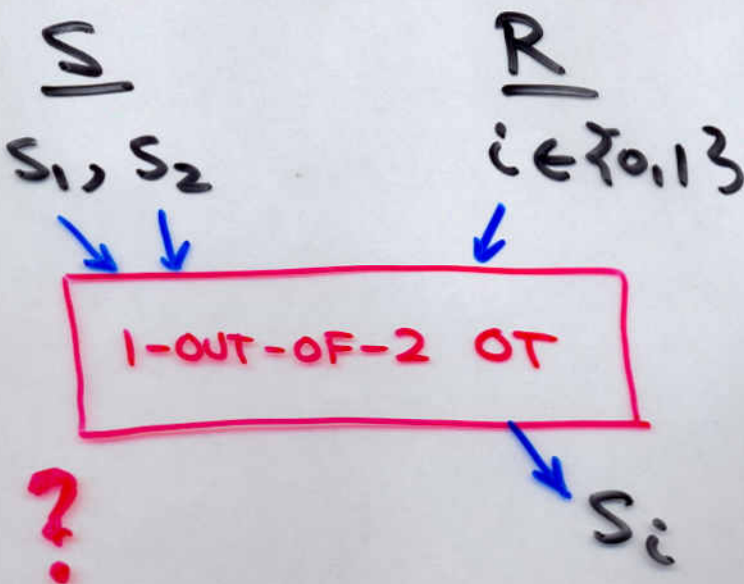
- REDUCTION TO 1-OUT-OF-4 OT

A MAX. PRIVACY PROTOCOL FOR $x_1 y_1 + x_2 y_2$

- WHAT WE NEED



- REDUCTION TO 1-OUT-OF-2 OT [EGL]



A MAX. PRIVACY PROTOCOL FOR $x_1 y_1 + x_2 y_2$

- **WHAT WE NEED**

$\underline{X}$	$\underline{Y}$
x_0, x_1, x_2	y_1, y_2

MAX. PRIVACY PROTOCOL

$x_0 + x_1 y_1 + x_2 y_2$

- **REDUCTION TO 1-OUT-OF-4 OT**

$\underline{X} : x_0, x_1, x_2$ $\qquad$ $\underline{Y} : y_1, y_2$

y_2	y_1	
0	0	x_0
0	1	$x_0 + x_1$
1	0	$x_0 + x_2$
1	1	$x_0 + x_1 + x_2$

$\xrightarrow{\text{1-OUT OF-4 OT}}$

IMPLEMENTING 1-OUT-OF-2 OT

$\underline{S}$ $\underline{R}$

secret bits s_1, s_2 interested in $i \in \{1,2\}$

CHOOSE f_T $(+\, b_T)$ $\xrightarrow{f_T}$

$r_1, r_2 \in_R \{0,1\}^n$

$r'_j \triangleq \begin{cases} f_T(r_j) & j = i \\ r_j & \text{OTHERWISE} \end{cases}$

$\xleftarrow{(r'_1,\, r'_2)}$

$r''_j \leftarrow f_T^{-1}(r'_j)$

$s'_j \leftarrow s_j \oplus b_T(r''_j)$ $\xrightarrow{(s'_1,\, s'_2)}$

$s_i \leftarrow s'_i \oplus b_T(r_i)$

BUT

$s_{i+1} = s'_{i+1} \oplus b_T(f_T^{-1}(r_{i+1}))$

REMAINS UNPREDICTABLE.

SUMMING UP

THM: EVERY PROTOCOL PROBLEM HAS A SOLUTION.

FURTHERMORE,

A SOLUTION CAN BE FOUND EFFICIENTLY!

A GAME (OF INCOMPLETE INFORMATION)

[von NEUMANN & MORGENSTERN]

S	STATES OF THE GAME
$K_i : S \to S_i$	KNOWLEDGE FUNCTIONS
M	POSSIBLE MOVES
$\mu_i : S_i \to M$	STRATEGIES
$\delta : S \times M \to S$	TRANSITION FUNCTION
$\rho : S \to V$	PAYOFF FUNCTION

A GAME (OF INCOMPLETE INFORMATION)
[von NEUMANN & MORGENSTERN]

S STATES OF THE GAME

$K_i : S \to S_i$ KNOWLEDGE FUNCTIONS

M POSSIBLE MOVES

$\mu_i : S_i \to M$ STRATEGIES

$\delta : S \times M \to S$ TRANSITION FUNCTION

$\rho : S \to V$ PAYOFF FUNCTION

- HOW TO SELECT AN <u>OPTIMAL</u> STRATEGY?

A GAME (OF INCOMPLETE INFORMATION)
[von NEUMANN & MORGENSTERN]

S — STATES OF THE GAME

- $K_i : S \to S_i$ <u>COMPUT'</u> KNOWLEDGE FUNCTIONS

M — POSSIBLE MOVES

~~$\mu_i : S_i \to M$ STRATEGIES~~

- $\delta : S \times M \to S$ <u>COMPUT'</u> TRANSITION FUNCTION
- $\rho : S \to V$ <u>COMPUT'</u> PAYOFF FUNCTION

- HOW TO IMPLEMENT A GAME ?

<u>SPECIAL CASES</u> :

(1) A PROTOCOL PROBLEM

(2) A GENERALIZED PROTOCOL PROBLEM
(WITH ON-LINE EXT. INPUTS)

- INIT STATE = $\emptyset$
- MOVES = INPUTS

IMPLEMENTING A COMPUTABLE GAME

- COMMIT TO A MOVE
- DISTRIBUTIVELY COMPUTE THE NEXT STATE.
- DISTRIBUTIVELY COMPUTE PARTIAL INFORMATION (FOR EACH PLAYER!)

repeat (as long as the game continues)

- DISTRIBUTIVELY COMPUTE THE FINAL PAYOFF.

SUMMARY

- GENERAL RESULTS (OBTAINED)
  - HOW TO SOLVE ANY PROTOCOL PROBLEM.
  - HOW TO PLAY ANY GAME.
- THE COST (TO BE REDUCED)
  - PASSING TO CIRCUIT MODEL
    - PRAM → CIRCUIT (~~TM → CIRCUIT~~)
    - WORK WITH BOUNDED DEGREE NETWORK?
  - $\Theta(n^2)$ COMMUNICATION PER EACH ELEMENTARY STEP.
    - ANOTHER MODEL ?
    - AMORTIZE ?
  - ZK PROOFS ON EACH MESSAGE
    - POSTPONE PROOFS TO THE END, GET RID OF OT.