

# How to Generate and Exchange Secrets (Yao86)

Hong Jiang

Apr. 16, 2004

## Introduction

- Purpose of this talk

- Terminology

## Generating a secret

- Problem definition

- Validity

- Fairness

- Main theorem

## Exchange of secrets

- Problem definition

- Validity

- Fairness

- Main theorem

## General computation

- Model I

- Model II - Semi-honest parties

- Model III

## Purpose of presenting this paper

This paper has been cited too many times. It may help us:

- ▶ Learn the original definition of many concepts and properties (privacy, validity, fairness etc.) of protocols.
- ▶ Understand the relation between primitives (OT, SS, ZKP etc.)
- ▶ Learn some general possibility results of two-party computation.
- ▶ Provide context for future talks on crypto protocols.

## Purpose of presenting this paper

This paper has been cited too many times. It may help us:

- ▶ Learn the original definition of many concepts and properties (privacy, validity, fairness etc.) of protocols.
- ▶ Understand the relation between primitives (OT, SS, ZKP etc.)
- ▶ Learn some general possibility results of two-party computation.
- ▶ Provide context for future talks on crypto protocols.

But ...

- ▶ Proofs for the theorems were not published.

## Protocol

- ▶ A protocol  $\mathcal{M} = (M_a, M_b)$  is a pair of communicating probabilistic Turing machines each with a special “send-receive” tape.
- ▶ Given inputs  $(n, i_A)$  and  $(n, i_B)$ , the two machines will alternately send and receive strings using the send-receive tapes.
- ▶ Each machine performs computation as a standard Turing machine after receiving a message string.
- ▶ Both machines halt within a number of steps bounded by some polynomial of  $n$ , leaving output  $u_A$  and  $u_B$ .
- ▶ For any run  $\sigma$  of the protocol, let  $\Delta_A(\sigma)$  denote the history of the run from A's view.

## Polynomial ensemble

A sequence of distributions  $(h_1, h_2, \dots, h_n, \dots)$  is a *polynomial ensemble*, if there's a probabilistic Turing machine which, given input  $n$ , generates in time polynomial in  $n$  a random string  $x$  with a distribution computationally indistinguishable from  $h_n$

## The general problem

- ▶ Assuming  $\{h_n\}$  and  $\{w_{n,i_A,i_B}\}$  are polynomial ensembles.
- ▶ Given  $(n, i_A)$  and  $(n, i_B)$ , where  $(i_A, i_B)$  is distributed according to  $h_n$  over  $\{0, 1\}^* \times \{0, 1\}^*$ , we want a protocol  $\mathcal{M}$  such that the output  $(u_A, u_B)$  is distributed according to  $w_{n,i_A,i_B}$  over  $\{0, 1\} \times \{0, 1\}$ .
- ▶ Require validity, privacy, and fairness (defined later).
- ▶ Call it a two-party interactive computational problem  $\langle h_n, w_n \rangle$ .

## An example

Minimum/zero knowledge interactive proof

- ▶ Compute function  $f_n(i_A, i_B)$  and  $g_n(i_A, i_B)$ , where  $i_A = (s, \tau)$  and  $i_B = \tau$ .
- ▶  $g_n(i_A, i_B) = 1$  iff  $s$  is a short proof (witness) for  $\tau \in L$
- ▶  $f_n(i_A, i_B) = 1$

## More terminology

- ▶  $D = (d_1, d_2, \dots)$  is a sequence of predicates computable in probabilistic polynomial time.
- ▶ A guessing algorithm  $Q_B$  is a PPT( $n$ ) algorithm which outputs  $\{0, 1\}$  on input  $(n, y)$  where  $y \in \{0, 1\}^*$ .
- ▶ Let  $r(d_n, X_n, Y_n, Q)$  be the probability that  $Q$  makes a correct guess about  $d_n(X_n)$  on input  $(n, Y_n)$ .
- ▶  $\circ(\text{poly} - \text{small})$  denotes any sequence  $(b_1, b_2, \dots)$  that has the property  $b_n = \circ(\frac{1}{n^k})$ .

## More terminology

- ▶ We write  $I_n(X_n | Y_n) \preceq I_n(X'_n | Y'_n)$  if  $\forall D \forall Q \exists Q'$  s.t.  
 $r(d_n, X'_n, Y'_n, Q) - r(d_n, X_n, Y_n, Q) \geq o(\text{poly} - \text{small})$ .
- ▶ We write  $I_n(X_n | Y_n) \approx I_n(X'_n | Y'_n)$  if  
 $I_n(X_n | Y_n) \preceq I_n(X'_n | Y'_n)$  and  $I_n(X'_n | Y'_n) \preceq I_n(X_n | Y_n)$ .

## Puzzle ensemble

- ▶ A *puzzle ensemble*  $P = (L, \mathcal{F})$  in which  $L \in BPP$  and  $\mathcal{F} = (F_1, F_2, \dots)$  is a poly-time ensemble where each  $F_n$  is a distribution over  $\{0, 1\}^* \times \{0, 1\}^*$ .
- ▶ Require a random  $(s, \tau)$  drawn from  $F_n$  satisfies  $(n, s, \tau) \in L$  with probability  $1 - o(\text{poly} - \text{small})$ .
- ▶ Call  $s$  a secret of the text  $\tau$ .

## Puzzle ensemble (cont.)

- ▶  $T_{P,n}$  denotes the distribution of a random  $\tau$  taken from the second component of a random  $(s, \tau)$  distributed as  $F_n$ .
- ▶ A puzzle ensemble  $P$  is intractable if, for every PPT algorithm  $S$ , when given  $(n, \tau)$  (was  $(s, \tau)$  in Yao86) where  $\tau$  is distributed according to  $T_{P,n}$ ,  $S$  will fail with probability  $1 - o(\text{poly} - \text{small})$  to produce an  $s$  satisfying  $(n, s, r) \in L$

## Problem definition

Let  $P = (L, \mathcal{F})$ , where  $\mathcal{F} = (F_1, F_2, \dots)$ , be an intractable puzzle ensemble. We want a protocol  $\mathcal{M} = (M_A, M_B)$  with the following properties for any  $n$ :

- ▶  $\mathcal{M}$  generates implicitly a pair  $(s, \tau)$  distributed according to  $F_n$ .
- ▶  $\tau$  is in the outputs of  $M_A$  and  $M_B$ .
- ▶  $s$  is computable by  $A$  and  $B$  jointly based on the information they have at the end of the protocol execution.
- ▶  $s$  is completely hidden from each party by itself, even if one of them cheats.

## Problem definition (cont.)

This paper restrict the discussion to puzzles with unique secrets:

- ▶  $P = (L, \mathcal{F})$  is uniquely decipherable if,  $\forall n, \tau$ , there is at most one  $s$  satisfying  $(n, s, \tau) \in L$ .
- ▶ We can write such a puzzle ensemble as  $P = (\alpha, \mathcal{D})$ , where  $\alpha = (a_1, a_2, \dots)$  and  $\mathcal{D} = (D_1, D_2, \dots)$  are given by  $a_n(\tau) = s$  and  $D_n = T_{P,n}$ .

## Validity

If both A and B follow the protocol:

- ▶  $Pr[u_A = u_B = \tau] = 1 - o(\text{poly} - \text{small})$ , and  $\tau$  is distributed according to a distribution poly-time indistinguishable from  $D_n$ .
- ▶  $I_n^{(J)}(a_n(\tau) \mid \tau, \Delta_j) \approx I_n^{(L)}(a_n(\tau) \mid \tau)$  for  $j \in \{A, B\}$ , where  $J$  is the stochastic process induced by the execution of the protocol  $\mathcal{M} = (M_A, M_B)$ , and  $L$  is the stochastic process of fetching  $\tau$  according to  $D_n$ .
- ▶  $\exists$  protocol  $\mathcal{N} = (N_A, N_B)$  which, given input  $(\Delta_A, \Delta_B)$ , computes outputs  $v_a$  and  $v_b$ , and  $Pr[v_A = v_B = a_n(\tau)] = o(\text{poly} - \text{small})$ .

## Validity (cont.)

If one party (B) may misbehave. Let  $d_n = Pr[v_A \neq \text{CHEATING}]$ , and  $D'_n$  be the probability distribution for  $\tau$  when restricted to such runs.

- ▶  $P' = (\alpha, D')$  is a uniquely decipherable intractable puzzle ensemble, where  $D' = (D'_1, D'_2, \dots)$ .
- ▶  $I_n^{(J)}(a_n(\tau) \mid \tau, \Delta_B) \approx I_n^{(L)}(a_n(\tau) \mid \tau)$  for  $j \in \{A, B\}$ , where  $J$  is the stochastic process induced by the execution of the protocol  $\mathcal{M} = (M_A, M'_B)$ , and  $L$  is the stochastic process of fetching  $\tau$  according to  $D_n$ .



## Fairness

- ▶ Suppose  $\mathcal{M} = (M_A, M_B)$  and  $\mathcal{N} = (N_A, N_B)$ .  $\exists$  PPT algorithm  $Y$  which takes a history pair for  $\mathcal{M}$  and  $\mathcal{N}$  and outputs a string  $w$ . IF  $A$  follows protocols  $\mathcal{M}$  and  $\mathcal{N}$  and then runs  $Y$ ,  
$$\Pr[u_A = \tau \wedge v_B = a_n(\tau) \wedge w \neq a_n(\tau)] = o(\text{poly} - \text{small})$$
- ▶ Interchange the roles of  $A$  and  $B$  above.

## Theorem 1.

Let  $P$  be an intractable puzzle ensemble that is uniquely decipherable. There exists a protocol  $\mathcal{M} = (M_A, M_B)$  for generating a secret from  $P$  that achieves validity and fairness. (Proof is left to the audience as exercise.)

## problem definition

- ▶  $P_A = (L_A, F_A)$  and  $P_B = (L_B, F_B)$  be two intractable puzzle ensembles.
- ▶  $(s_A, \tau_A)$  and  $(s_B, \tau_B)$  be random puzzles distributed according to  $F_{A,n}$  and  $F_{B,n}$ .
- ▶ Give  $(n, s_A, \tau_A, \tau_B)$  as input to  $A$  and  $(n, s_B, \tau_A, \tau_B)$  to  $B$ .
- ▶ We want a protocol  $\mathcal{M} = (M_A, M_B)$  that will enable  $A$  and  $B$  to exchange their secrets  $s_A$  and  $s_B$  so that neither will be swindled.

## Validity

If both parties follow the protocol, then  
 $Pr[u_B = s_A \wedge u_A = s_B] = 1 - o(\text{poly} - \text{small}).$

## Fairness

▶ If  $B$  gets  $s_A$ , then  $A$  can compute  $s_B$ .

▶ If  $B$  gets  $s_A$ , then  $A$  can compute  $s_B$ .

## Fairness

- ▶ If  $A$  follows the protocol and  $B$  does not (i.e.,  $M'_B \neq M_B$ ), for any fixed  $k$ , there exists a PPT algorithm  $S$  which takes  $(n, \Delta_A)$  as input and output a string  $v$  such that the following is true:

$$\Pr[(u_B = s_A) \wedge (v \neq s_B)] = O\left(\frac{1}{n^k}\right)$$

- ▶ Interchange  $A$  and  $B$  in the above definition.

## Theorem 2.

Let  $P_A$  and  $P_B$  be any two intractable ensembles. There exists a protocol  $\mathcal{M} = (M_A, M_B)$  for exchanging secrets between  $P_A$  and  $P_B$  that achieves validity and fairness.

## General computation

- ▶ Consider an interactive computational problem  $\langle h_n, w_{n,i_A,i_B} \rangle$  as defined in slide 3.
- ▶ This paper only discusses the case when  $w_{n,i_A,i_B}$  represents a pair of functions  $(f_n, g_n)$ .
  - ▶ i.e., given  $i_A$  and  $i_B$ ,  $A$  and  $B$  want to compute  $f_n(i_A, i_B)$  and  $g_n(i_A, i_B)$ .

## Two variant models

We consider two variants of this problem:

Model I.  $A$  and  $B$  are given  $i_A$  and  $i_B$  as inputs respectively.

Model II.  $A$  is given as input  $(i_A, p_A, q_A, N_A, N_B, E_{N_B}(i_B))$ , and  $B$  is given input  $(i_B, p_B, q_B, N_B, N_A, E_{N_A}(i_A))$ .

- ▶  $N_A = p_A \cdot q_A$  and  $N_B = p_B \cdot q_B$ .
- ▶  $p_A, q_A, p_B,$  and  $q_B$  are large primes.
- ▶  $E_N$  is any probabilistic encryption scheme probably secure under the intractability assumption of factoring.

## Model I (Semi-honest parties)

The constraints when both parties follow the protocol.

Validity.  $Pr[u_A = f_n(i_A, i_B) \wedge u_B = g_n(i_A, i_B)] = 1 - o(\text{poly} - \text{small})$

Privacy. The following hold:

1.  $I_n^{(J)}(i_A, i_B, u_A \mid \Delta_B, i_B, u_B) \approx I_n^{(L)}(i_A, i_B, u_A \mid i_B, u_B)$   
where  $J$  is the stochastic process of running  $M$  with input from  $h$ , and  $L$  is the stochastic process of having  $(i_A, i_B)$  distributed according to  $h$  and  $u_A = f_n(i_A, i_B)$ ,  $u_B = g_n(i_A, i_B)$ .
2. Interchange  $A$  and  $B$  in (i).

## Model I - One cheating party

Suppose  $A$  follows the protocol while  $B$  may cheat.

- ▶ let  $M'_B$  be any communicating Turing machine.
- ▶ let  $U_n$  be the distribution corresponding to  $u_A$  when  $(M_A, M'_B)$  are run.
- ▶  $Z = \{0, 1\}^* \cup \{CHEATING\}$ , and extend the function  $f_n$  by  $f_n(i_A, y) = CHEATING$  if  $y = CHEATING$ .

## Model I - One cheating party (cont.)

Validity:

- ▶ With high probability,  $A$  either detects cheating or learns the value of  $f_n(i_A, y)$  where  $y$  is the value  $B$  supplies as his input.

## Model I - One cheating party (cont.)

Validity:

- ▶  $\exists S$  (PPT Algorithm) that takes  $(n, i_B)$  and produces random  $y \in Z$  s.t.  $U_n$  is indistinguishable from the distribution corresponding to  $f_n(i_A, y)$ .

## Model I - One cheating party (cont.)

- ▶ Let  $J$  be the stochastic process of running  $(M_A, M'_B)$  with input  $(i_A, i_B)$  distributed according to  $h_n$ .
- ▶ Let  $\mathcal{S}$  be the set of all probabilistic polynomial-time algorithms  $S$  that take  $(n, i_B)$  to produce random  $y \in Z$ .
- ▶  $\forall S \in \mathcal{S}$ , let  $L(s)$  be the stochastic process of generating  $(i_A, i_B)$  according to  $h_n$  and running  $S$  on input  $(n, i_B)$  to produce a random  $y$ . (*Different from Yao86*) And define  $u_A = f_n(i_A, y)$  and  $u_B = g_n(i_A, y)$ .

## Model I - One cheating party (cont.)

Privacy:

- ▶ For any  $M'_B$ ,  $\exists S \in \mathcal{S}$  s.t.

$$I_n^{(J)}(i_A, i_B, u_A \mid \Delta_B, i_B, u_B) \preceq I_n^{(L(S))}(i_A, i_B, u_A \mid i_B, u_B).$$

## Model I - One cheating party (cont.)

- ▶  $L_n$  denotes the set of all possible values of  $f_n(i_A, i_B)$  when  $(i_A, i_B)$  is distributed according to  $h_n$ .
- ▶ A *recovery algorithm*  $R$  for  $A$  is a PPT algorithm which takes  $(n, \Delta_A)$  and outputs a string  $v$ .
- ▶  $G$  denotes a PPT Turing machine that takes  $i_B$  and output  $z$ , and  $\beta(G) = Pr[z = g_n(i_A, i_B)]$ .

## Model I - One cheating party (cont.)

Fairness For any fixed  $k$ , there exists a recovery algorithm  $R$  whose output  $v$  satisfied the following condition:

$$Pr[(u_B = g_n(i_A, i_B)) \wedge (R(n, \Delta_A) \notin L_n)] \leq \beta_n(G) + O\left(\frac{1}{n^k}\right)$$

for some  $G$ .

## Model II - Semi-honest parties

Validity:

- ▶  $Pr[u_A = f_n(i_A, i_B) \wedge v_A = (p'_A, q'_A, N'_A) \wedge w_A = E_{N'_B}(g_n(i_A, i_B))] = d_n - o(\text{poly} - \text{small})$ , where  $N'_B$  is the product of  $n$ -bit primes  $p'_A$  and  $q'_A$ .

Privacy:

1.  $I_n^{(J)}(i_A, i_B, u_A \mid \Delta_B, i_B, u_B, w_B) \approx I_n^{(L)}(i_A, i_B, u_A \mid i_B, u_B)$   
where  $J$  is the stochastic process of running  $\mathcal{M}$  with input from  $h$ , and  $L$  is the stochastic process of having  $(i_A, i_B)$  distributed according to  $h$  and  $u_A = f_n(i_A, i_B), u_B = g_n(i_A, i_B)$ .
2. Interchange A and B in (1).

## Model II - One cheating party

Suppose  $A$  follows the protocol while  $B$  may cheat.

- ▶ Let  $M'_B$  be any communicating Turing machine.
- ▶ Let  $d_n$  be the probability of runs in which  $v_A \neq \text{CHEATING}$ .  
And let  $U_{n,i_B}$  be the distribution corresponding to  $u_B$  when restricted to such runs and with  $i_B$  being an input for  $B$ .

## Model II - One cheating party

▶ Validity.

- ▶  $Pr[u_A = f_n(i_A, i_B) \wedge v_A = (p'_A, q'_A, N'_A) \wedge w'_A = E_{N'_B}(g_n(i_A, i_B))] = d_n - o(\text{poly} - \text{small})$ , where  $N'_B$  is the product of  $n$ -bit primes  $p'_A$  and  $q'_A$ .

▶ Privacy

- ▶  $I_n^{(J)}(i_A, i_B, u_A \mid \Delta_B, i_B, u_B, u_B, w_B) \approx I_n^{(L)}(i_A, i_B, u_A \mid i_B, u_B)$  where  $J$  is the stochastic process of running  $\mathcal{M} = (N_A, M'_B)$ , and  $L$  is the stochastic process of being given the value of  $u_B$  distributed according to  $U_{n, i_B}$

▶ Fairness

- ▶ For any fixed  $k$ , there exists a recovery algorithm  $R$  whose output  $v$  satisfied the following condition:

$$Pr[(u_B = g_n(i_A, i_B)) \wedge (s \neq f_n(i_A, i_B))] \leq \beta_n(G) + O\left(\frac{1}{n^k}\right)$$

for some  $G$ .

## Theorem 3

For any interactive computational problem  $\langle h_n, (f_n, g_n) \rangle$ , there exists a protocol  $\mathcal{M}$  that achieves validity, privacy and fairness.

## Conclusion and questions

- ▶ The definitions were too complicated.
- ▶ The proofs were probably more complicated.
- ▶ How to justify these definitions? Do they really capture the requirements of *all* two-party computation problem?
- ▶ Are there simpler and more elegant definitions of security?
  - ▶ There are other definitions in the literature, which have their own problems.
  - ▶ We may talk about some more recent works in future.