Cryptographic Protocols

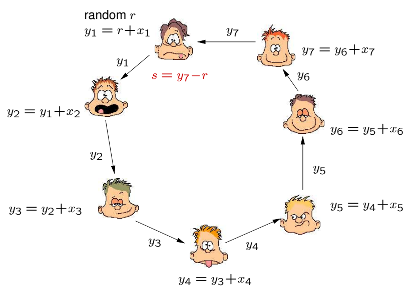
Spring 2021

MPC Part 1

Sum Protocol

Goal: Compute sum of inputs

Protocol

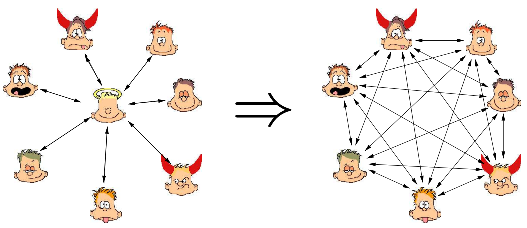


Specification

0. $\forall P_i$: input x_i
1. $\forall P_i$: send x_i to TTP
2. TTP: $y = \sum x_i$
3. TTP: send y to $\forall P_i$

Analysis: 1 passive cheater? 2 passive? 1 active? 2 active?

Multi-Party Computation: Goal



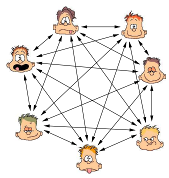
Specification **Protocol**

A protocol is secure if the adversary cannot achieve anything in the protocol that he could not achieve in the specification.

Intuition: $\forall$Adv $\exists$Sim : $\text{Prot}^{\text{Adv}} \sim \text{Spec}^{\text{Sim}}$

Model

Parties and Channels



- n parties $P_1, \ldots, P_n$
- Secure channels among parties
- Broadcast channels

Adversary

- Central adversary (collaborating parties)
- Corrupts t parties
- Passive vs active

Security

- Information-theoretic vs. Cryptographic

Sum Protocol II

Protocol:

	x_1	x_{11}	x_{12}	x_{13}	x_{14}	$\cdots$	x_{1n}
	x_2	x_{21}	x_{22}	x_{23}	x_{24}	$\cdots$	x_{2n}
	x_3	x_{31}	x_{32}	x_{33}	x_{34}	$\cdots$	x_{3n}
	x_4	x_{41}	x_{42}	x_{43}	x_{44}	$\cdots$	x_{4n}
	$\vdots$			$\vdots$			
	x_n	x_{n1}	x_{n2}	x_{n3}	x_{n4}	$\cdots$	x_{nn}
		y_1	y_2	y_3	y_4	$\cdots$	y_n

$y = \sum_{i=1}^{n} y_i$

Analysis: 1 passive cheater? 2 passive? 1 active? 2 active?

More Examples

Examples

- Statistics (first sex, tax evading, etc.)
- Elections / Votes / Auctions
- Millionaires problem
- Loans (several banks, same guarantee)
- ZK-proofs (Peggy sends witness to TTP, who checks & sends 0/1 to Vic)

Secure Function Evalutation (evaluate function f on all inputs)

1. $\forall P_i$: send input x_i to TTP
2. TTP: compute $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$
3. TTP: send output y_j to $\forall P_j$

Limitations

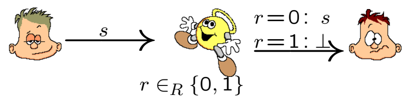
- Poker, etc (not realizable with TTP)

Known Results

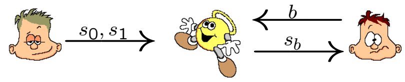
Setting	Condition	Literature
Cryptographic, passive	$t < n$	[GMW87]
Cryptographic, active	$t < n/2$	[GMW87]
Information-theoretic, passive	$t < n/2$	[BGW88,CCD88]
Information-theoretic, active	$t < n/3$	[BGW88,CCD88]
Information-theoretic, active assuming broadcast	$t < n/2$	[RB89,Bea91]

Oblivious Transfer

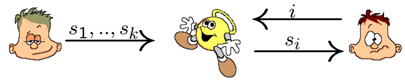
Rabin-OT Sender Receiver



1-2-OT Sender Receiver



1-k-OT Sender Receiver



1-2-OST based on RSA and AES

Sender		Receiver
Messages s_0, s_1		Selector $b \in \{0,1\}$
Generate RSA-Keys		
n_0, e_0, d_0 and n_1, e_1, d_1 with $n_0 \approx n_1$	$\xrightarrow{n_0, e_0, n_1, e_1}$	
		k at random,
	$\xleftarrow{u}$	$u = k^{e_b} \pmod{n_b}$
$k_0 = u^{d_0} \pmod{n_0}$		
$k_1 = u^{d_1} \pmod{n_1}$		
$y_0 = \mathrm{AES}_{k_0}(s_0)$		
$y_1 = \mathrm{AES}_{k_1}(s_1)$	$\xrightarrow{y_0, y_1}$	$s_b = \mathrm{AES}_k^{-1}(y_b)$

MPC from OT

Starting Point

- 2 parties Alice and Bob
- Inputs $a \in A$ and $b \in B$
- Fixed function $f : A \times B \to C$

Truth table:

a	b	c
0	0	17
0	1	23
0	2	8
1	0	17
1	1	10
1	2	−4
2	0	...
2	1	...

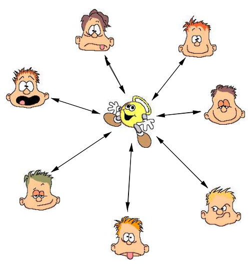
Protocol

1. Alice sends $[f(a, b_1) \mid f(a, b_2) \mid \ldots \mid f(a, b_\ell)]$ via OT
2. Bob selects b-th value
3. Bob sends output to Alice

Analysis:
- Security
- Efficiency

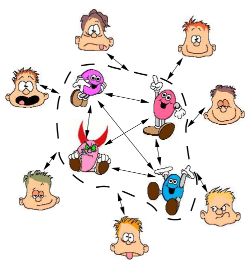
Extension: 3 parties ...

Multi-Party Computation: Goal II



Specification

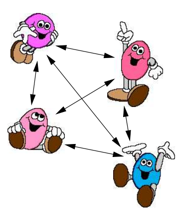
$\Longrightarrow$



Protocol

Multi-Party Computation: Goal II

$\Longrightarrow$



Trusted party

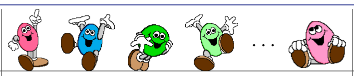
- Receive input
- $\oplus$ and $\otimes$ over finite field $\mathbb{F}$
- Give output

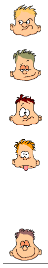
Simulating players ...

- n players: $\mathcal{P} = \{P_1, \ldots, P_n\}$
- Players can $\oplus$ and $\otimes$ in $\mathbb{F}$
- Players can communicate

Sum Protocol III

Protocol:





x_1	x_{11}	x_{12}	x_{13}	x_{14}	$\cdots$	x_{1n}	
x_2	x_{21}	x_{22}	x_{23}	x_{24}	$\cdots$	x_{2n}	
x_3	x_{31}	x_{32}	x_{33}	x_{34}	$\cdots$	x_{3n}	
x_4	x_{41}	x_{42}	x_{43}	x_{44}	$\cdots$	x_{4n}	
$\vdots$			$\vdots$				
x_n	x_{n1}	x_{n2}	x_{n3}	x_{n4}	$\cdots$	x_{nn}	
	y_1	y_2	y_3	y_4	$\cdots$	y_n	$y = \sum_{i=1}^{n} y_i$

Analysis: 1 passive cheater? 2 passive? 1 active? 2 active?

Secret-Sharing Schemes – Definition

Intuition

- Dealer D can share a secret s among parties $\mathcal{P}$
- Qualified subsets of $\mathcal{P}$ can reconstruct s (w/o D)
- Access structure $\Gamma \subseteq 2^{\mathcal{P}}$

Definition

A secret-sharing scheme for parties $\mathcal{P}$ and access structure Γ is a pair of protocols (SHARE, RECONSTRUCT), s.t.

- Correctness:
  1. After SHARE, there is a unique value s', where $s' = s$ (the dealer's input) if the dealer is honest
  2. After RECONSTRUCT(M), if $M \in \Gamma$, all players in M know s'
- Privacy: After SHARE, non-qualified sets have no information about s

Secret-Sharing Schemes – Examples

Example 1

- Parties $\mathcal{P}$
- $\Gamma = \{\mathcal{P}\}$ (only all parties jointly can reconstruct)
- SHARE: select random $x_1, \ldots, x_n$ with $\sum x_i = s$, send x_i to P_i
- RECONSTRUCT: Obvious

Example 2

- Parties $\mathcal{P}$, arbitrary access structure Γ
- SHARE: $\forall M_i \in \Gamma$:
  - select random $\{x_{ij}\}_{P_j \in M_i}$ with $\sum_j x_{ij} = s$,
  - send x_{ij} to $P_j \in M_i$
- RECONSTRUCT: Obvious

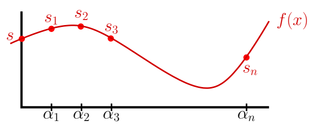
Shamir's Secret-Sharing Scheme (1/3)

Goal

- n parties, k needed for reconstruction
- Threshold access structure $\Gamma = \{M \subseteq \mathcal{P} \;:\; |M| \geq k\}$

Idea

- Random polynomial f of degree d is defined by $d+1$ points
- $s = f(0)$ = secret, party P_i gets share $s_i = f(\alpha_i)$ for fixed α_i
- Degree $d = k-1 \;\Rightarrow\; k$ parties can reconstruct, $k-1$ cannot



Shamir's Secret-Sharing Scheme (2/3)

Starting Point: To each party P_i, some unique $\alpha_i \in \mathbb{F} \setminus \{0\}$ is assigned.

SHARE

1. D: choose random f with $f(0) = s$ and $\deg(f) \leq d$ (i.e., choose random $r_1, \ldots, r_d$, let $f(x) = s + r_1 x + \ldots + r_d x^d$)
2. D: send $s_i = f(\alpha_i)$ to $\forall P_i$

RECONSTRUCT

1. $\forall P_i$: send s_i to P
2. P: compute s with Lagrange interpolation:

$$f(x) = \sum_{i=1}^{n} \lambda_i(x)\, s_i, \text{ where } \lambda_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - \alpha_j}{\alpha_i - \alpha_j}.$$

$$\text{hence } s = \sum_{i=1}^{n} w_i s_i, \text{ where } w_i = \lambda_i(0) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{-\alpha_j}{\alpha_i - \alpha_j}.$$

Shamir's Secret-Sharing Scheme (3/3)

Analysis for passive adversary:

Correctness

- 1: by inspection, $s' = f(0)$
- 2: due to Lagrange interpolation (given $|M| \geq k = d+1$)

Privacy

- For $\leq d = k-1$ shares, every secret s is "compatible" (same #polys)
- $\Rightarrow$ adversary with $< k$ shares obtains no information about s.

Note

- Degree is at most d, not exactly d
- Otherwise privacy violation

Linear Secret-Sharing Schemes

19

Definition: Secret-Sharing is linear, if each share $s_i = \mathcal{L}_i(s, r_1, \ldots, r_\ell)$:

$$\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} = \begin{bmatrix} A_{10} & A_{11} \cdots A_{1\ell} \\ A_{20} & A_{21} \cdots A_{2\ell} \\ \vdots & \vdots \quad\quad \vdots \\ A_{n0} & A_{n1} \cdots A_{n\ell} \end{bmatrix} \cdot \begin{bmatrix} s \\ r_1 \\ \vdots \\ r_\ell \end{bmatrix}$$

Addition

$$\begin{array}{rcl} [\ s_1, \ \ldots, \ s_n \ ] & = & A \cdot [\ s, \ r_0, \ \ldots, \ r_\ell \ ] \\ [\ s_1', \ \ldots, \ s_n' \ ] & = & A \cdot [\ s', \ r_0', \ \ldots, \ r_\ell' \ ] \\ \hline [s_1 + s_1' \ , \ldots, \ s_n + s_n'] & = & A \cdot [s + s', \ r_0 + r_0', \ \ldots, \ r_\ell + r_\ell'] \end{array}$$

Shamir Sharing is linear

$$A = \begin{bmatrix} 1 & \alpha_1 & \ldots & \alpha_1^d \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_n & \ldots & \alpha_n^d \end{bmatrix}$$

(Van der Monde Matrix)

MPC Passive: Secret-Sharing and Addition

20

Setting

- n parties, t corrupted (passive), $t < n/2$

Secret Sharing

- Shamir-Sharing with degree t
- $\Rightarrow$ any t (corrupted) parties do not learn anything

Addition and Linear Functions

- Shamir-Sharing is linear $\Rightarrow$ apply linear function on shares
- $a, b, \ldots$ shared by $a_1, \ldots, a_n, \ b_1, \ldots, b_n$, etc.
- Every P_i computes $c_i = \mathcal{L}(a_i, b_i, \ldots)$
- $c_1, \ldots, c_n$ is a sharing of $c = \mathcal{L}(a, b, \ldots)$

MPC Passive: Multiplication

21

Starting Point: a, b shared by $a_1, \ldots, a_n, \ b_1, \ldots, b_n$

Idea

- Every P_i computes $d_i = a_i \cdot b_i$
- Observe: $d_1, \ldots, d_n$ is some-kind-of sharing of $c = a \cdot b$
- Could compute c from $d_1, \ldots, d_n$: $c = \sum_{i=1}^{n} w_i d_i$ (Lagrange)
- Compute c as MPC: Every P_i has input d_i, compute (sharing of) c

Multiplication Protocol

1. $\forall P_i$: compute $d_i = a_i b_i$.
2. $\forall P_i$: share $d_i \ \rightarrow \ d_{i1}, \ldots, d_{in}$.
3. $\forall P_j$: compute $c_j = w_1 d_{1j} + \ldots + w_n d_{nj}$.

Passive Protocol

22

Share input

0. P_i has input s.
1. P_i: select $r_1, \ldots, r_t$ at random.
2. P_i: comp. $\begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} = A \begin{pmatrix} s \\ r_1 \\ \vdots \\ r_t \end{pmatrix}$.
3. P_i: send s_j to every P_j.

Reconstruct Output

0. a is shared by $a_1, \ldots, a_n$.
1. $\forall P_j$: send a_j to P_i.
2. P_i: comp. $a = \mathcal{L}(a_1, \ldots, a_n)$.

Addition and Linear Functions

0. $a, b, \ldots$ are shared by $a_1, \ldots, a_n, \ b_1, \ldots, b_n$, etc.
1. $\forall P_i$: compute $c_i = \mathcal{L}(a_i, b_i, \ldots)$.

Multiplication

0. a, b are shared by $a_1, \ldots, a_n, b_1, \ldots, b_n$.
1. $\forall P_i$: compute $d_i = a_i b_i$.
2. $\forall P_i$: share $d_i \ \rightarrow \ d_{i1}, \ldots, d_{in}$.
3. $\forall P_j$: compute $c_j = \mathcal{L}(d_{1j}, \ldots, d_{nj})$.

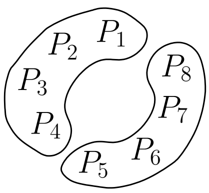
Passive MPC – Impossibility

23

Two Parties ($n = 2, t = 1$)

Alice		**Bob**
Input a		Input b
Random string r		Random string s
	$\xrightarrow{m_1}$	
	$\xleftarrow{m_2}$	
	$\ldots\ldots$	
	$\xrightarrow{m_\ell}$	
Output $c = ab$		Output $c = ab$

n Parties ($n, t \geq n/2$)



Analysis

- $b = 0$: Transcript contains no info about a, i.e. $I(T; a) = 0$.
- $b = 1$: Transcript contains full info about a, i.e. $I(T, a) = H(a)$.
- Alice can check: Is there (a', r') s.t. same transcript is produced?
  (exhaustive search requires unbounded computing power)