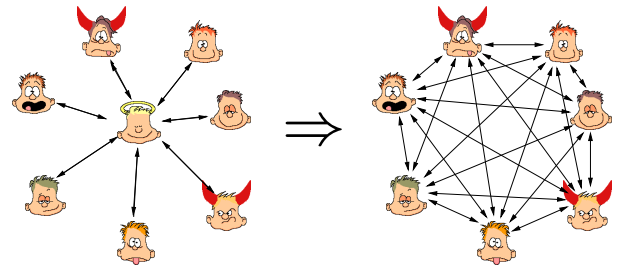


Cryptographic Protocols

Spring 2017

Part 6

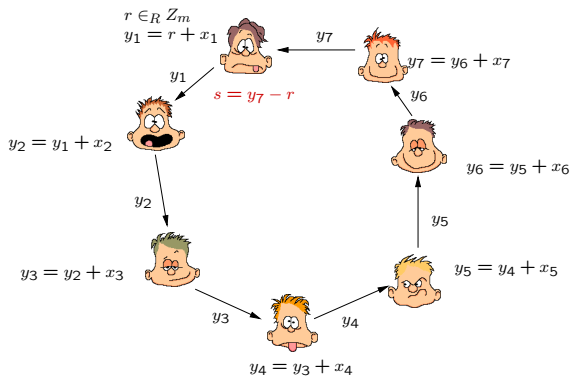
Multi-Party Computation: Goal



Specification

Protocol

Sum Protocol



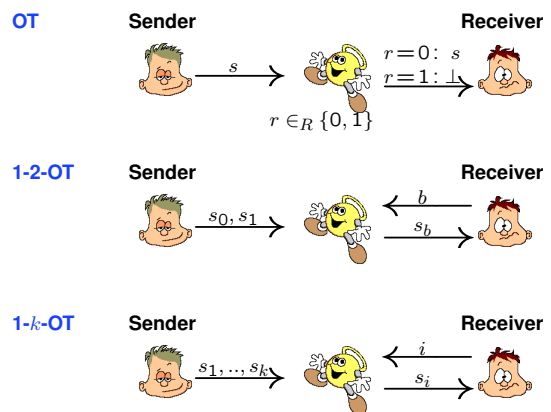
Sum Protocol II

			...		
	x_1	x_{11}	x_{12}	...	x_{1n}
	x_2	x_{21}	x_{22}	...	x_{2n}
	\vdots	\vdots	\vdots	\vdots	\vdots
	x_n	x_{n1}	x_{n2}	...	x_{nn}
	y_1	y_2	...	y_n	$y = \sum_{i=1}^n y_i$

Known Results

Setting	Adv. Type	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]
i.-t. with broadcast	active	$t < n/2$	[RB89, Bea91]

Oblivious Transfer



1-2-OST based on RSA and DES

Sender

messages s_0, s_1

generate RSA-Keys

n_0, e_0, d_0 and n_1, e_1, d_1

with $n_0 \approx n_1$

n_0, e_0, n_1, e_1

Receiver

selector $b \in \{0, 1\}$

k at random,

$u = k^{e_b} \pmod{n_b}$

u

$k_0 = u^{d_0} \pmod{n_0}$

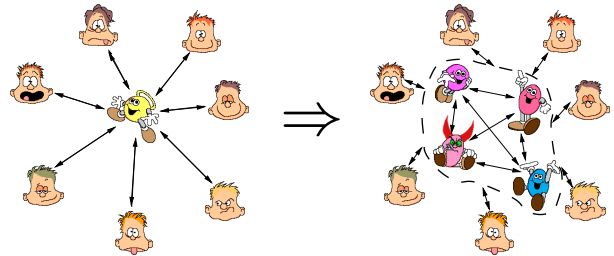
$k_1 = u^{d_1} \pmod{n_1}$

$y_0 = \text{DES}_{k_0}(s_0)$

$y_1 = \text{DES}_{k_1}(s_1)$

$y_0, y_1 \rightarrow s_b = \text{DES}_k^{-1}(y_b)$

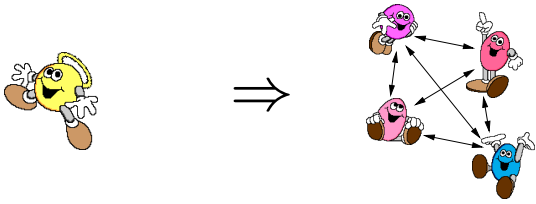
Multi-Party Computation: Goal III



Specification

Protocol

Multi-Party Computation: Goal IV



Trusted party

- receive input
- \oplus and \otimes over finite field \mathbb{F}
- give output

Simulating players ...

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

Operations

	Specification \Rightarrow	Protocol	
Input:			<input type="checkbox"/>
Compute:			<input type="checkbox"/> <input type="checkbox"/>
Output:			<input type="checkbox"/>

Sum Protocol III

			...	
x_1	x_{11}	x_{12}	...	x_{1n}
x_2	x_{21}	x_{22}	...	x_{2n}
\vdots			\vdots	
x_n	x_{n1}	x_{n2}	...	x_{nn}
	y_1	y_2	...	y_n
	$y = \sum_{i=1}^n y_i$			

Passive Protocol

Share input

- P_i has input s .
- P_i selects r_1, \dots, r_t at random.
- P_i comp. $\begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} = A \begin{pmatrix} r_1^s \\ \vdots \\ r_t^s \end{pmatrix}$.
- P_i sends s_j to every P_j .

Reconstruct Output

- a is shared by a_1, \dots, a_n .
- every P_j sends a_j to P_i .
- P_i comp. $a = \mathcal{L}(a_1, \dots, a_n)$.

Addition and linear functions \mathcal{L}

- a, b, \dots shared by $a_1, \dots, a_n, b_1, \dots, b_n$, etc.
- every P_i computes $c_i = \mathcal{L}(a_i, b_i, \dots)$.

Multiplication

- a, b are shared by $a_1, \dots, a_n, b_1, \dots, b_n$.
- every P_i computes $d_i = a_i b_i$.
- every P_i shares $d_i \rightarrow d_{i1}, \dots, d_{in}$.
- every P_j computes $c_j = \mathcal{L}(d_{1j}, \dots, d_{nj})$.