STAM Center

CSE/CEN 598
Hardware Security & Trust

Secure Computation Approaches: Security Protocols

Prof. Michel A. Kinsy

1

STAM Center

Foundations of Secure Computing

- Security protocols
  - Multi-party computation, zero-knowledge, oblivious transfer, security models, etc.
- Homomorphic encryption (HE)
  - Hardware and software implementations
- Design and implementation of trusted platform modules (TPMs)
  - TPM-based anonymous authentication, signature, encryption, identity management, etc.
- Trusted execution environments (TEEs)
  - TEE-based security and privacy techniques, vulnerability and countermeasures of TEE, distributed TEE, decentralized TEE, etc.

2

STAM Center

Foundations of Secure Computing

- Security protocols
  - Multi-party computation, zero-knowledge, oblivious transfer, security models, etc.
- Homomorphic encryption (HE)
  - Hardware and software implementations
- Design and implementation of trusted platform modules (TPMs)
  - TPM-based anonymous authentication, signature, encryption, identity management, etc.
- Trusted execution environments (TEEs)
  - TEE-based security and privacy techniques, vulnerability and countermeasures of TEE, distributed TEE, decentralized TEE, etc.

3

Threshold Secret Sharing Scheme

- Select
  - *p* a large prime number and
  - S as the secret value
  - $s_1,\ldots,s_{k-1}$ a set of randomly numbers from [0, p-1]
- A (k, n) threshold polynomial can be written by
  $s(x) \equiv S+s_1x+s_2x^2+\ldots+s_{k-1}x^{k-1} \pmod p$
- Send $(x_i,s(x_i))$ to the *i*-th participant
- Secret sharing in distributed systems provides
  - Fault-tolerant
  - Multi-factor authentication
  - Multi-party authorization

4

Threshold Secret Sharing Scheme

- Secret Reconstruction
  - To reconstruct the secret S, one needs to collect at least k partial secrets
  - The secret can then be reconstructed using Lagrange interpolation

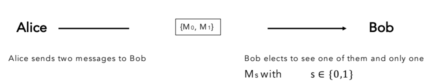
$$s(x) \equiv \sum_{j=1}^{k}\left[s(x_j)\prod_{i=1, i\neq j}^{k}\frac{x-x_i}{x_j-x_i}\right] \bmod p$$

- The scheme can be extended to support share renewal and share recovery

5

Oblivious Transfer

- Oblivious Transfer refers to the technique of transferring a specific piece of data based on the receiver's selection



  - Alice does not know which one of the two Bob has selected
  - Bob is also oblivious to the content of the non-selected message

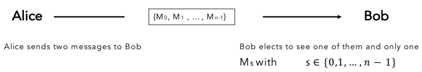
6

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering, Arizona State University

Oblivious Transfer

- Oblivious Transfer refers to the technique of transferring a specific piece of data based on the receiver's selection



  - Alice does not know which one of the *n* Bob has selected
  - Bob is also oblivious to the content of the non-selected message

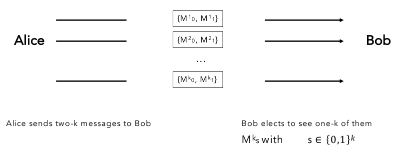
7

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering, Arizona State University

Oblivious Transfer

- Oblivious Transfer refers to the technique of transferring a specific piece of data based on the receiver's selection



  - There are algorithms for optimizing these straightforward implementations

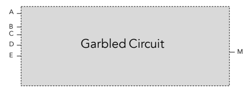
8

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering, Arizona State University

Oblivious Transfer

- Oblivious transfer is the necessary and sufficient condition for multiparty computation
- How can one practically perform this oblivious transfer?
  - For that let us introduce garbled circuits
    - Garbling is a process by means of which the Boolean gate truth table is obfuscated



9

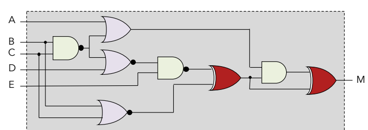
STAM Center

Garbled Circuit

10

STAM Center

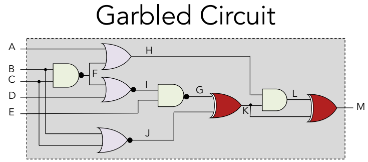
Garbled Circuit



11

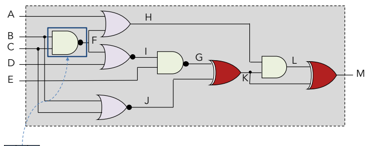
STAM Center

Garbled Circuit



12

Garbled Circuit

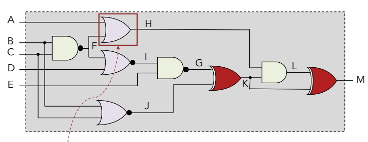


B	C	F
0	0	1
0	1	1
1	0	1
1	1	0

13

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS
ASU Ira A. Fulton Schools of Engineering
Arizona State University

Garbled Circuit



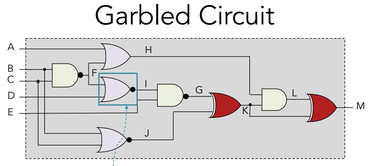
B	C	F
0	0	1
0	1	1
1	0	1
1	1	0

A	F	H
0	0	0
0	1	1
1	0	1
1	1	1

14

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS
ASU Ira A. Fulton Schools of Engineering
Arizona State University

Garbled Circuit

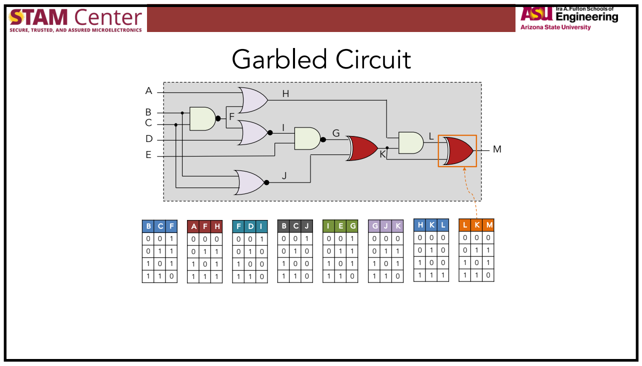


B	C	F
0	0	1
0	1	1
1	0	1
1	1	0

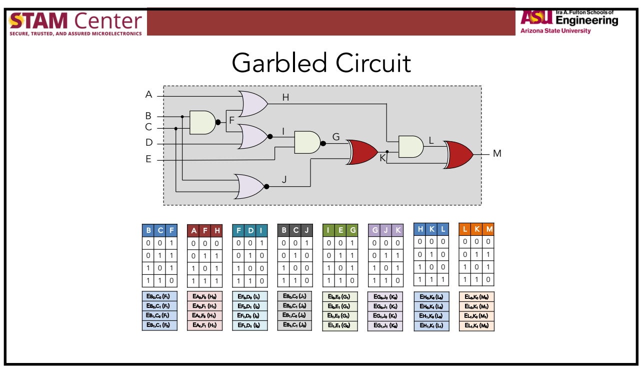
A	F	H
0	0	0
0	1	1
1	0	1
1	1	1

F	D	I
0	0	1
0	1	0
1	0	0
1	1	0

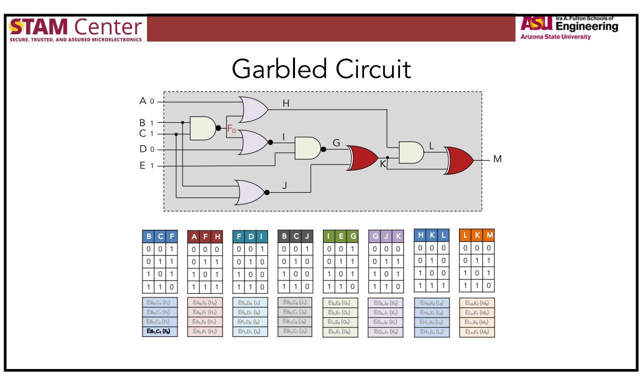
15


STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS
Engineering
Arizona State University
Garbled Circuit
A
B
C
D
E
F
H
I
G
J
K
L
M

16

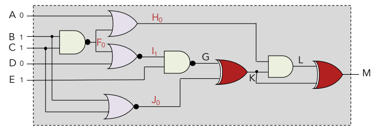

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS
Engineering
Arizona State University
Garbled Circuit
A
B
C
D
E
F
H
I
G
J
K
L
M

17


STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS
Engineering
Arizona State University
Garbled Circuit
A 0
B 1
C 1
D 0
E 1
H
I
G
J
K
L
M

18

Garbled Circuit

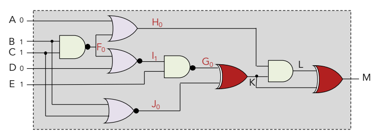


B	C	F	A	F	H	F	D	I	B	C	J	I	E	G	G	J	K	H	K	L	L	K	M
0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	1	0	0	1	0	0	1	1	0	1	1	0	1	0	0	1	1
1	0	1	1	0	1	1	0	0	1	0	0	1	0	1	1	0	1	1	0	0	1	0	1
1	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0

$E_{B_0,C_0}(F_1)$	$E_{A_0,F_0}(H_0)$	$E_{F_0,D_0}(I_1)$	$E_{B_0,C_0}(J_1)$	$E_{I_0,E_0}(G_1)$	$E_{G_0,J_0}(K_0)$	$E_{H_0,K_0}(L_0)$	$E_{L_0,K_0}(M_0)$
$E_{B_0,C_1}(F_1)$	$E_{A_0,F_1}(H_1)$	$E_{F_0,D_1}(I_0)$	$E_{B_0,C_1}(J_0)$	$E_{I_0,E_1}(G_1)$	$E_{G_0,J_1}(K_1)$	$E_{H_0,K_1}(L_0)$	$E_{L_0,K_1}(M_1)$
$E_{B_1,C_0}(F_1)$	$E_{A_1,F_0}(H_1)$	$E_{F_1,D_0}(I_0)$	$E_{B_1,C_0}(J_0)$	$E_{I_1,E_0}(G_1)$	$E_{G_1,J_0}(K_1)$	$E_{H_1,K_0}(L_0)$	$E_{L_1,K_0}(M_1)$
$E_{B_1,C_1}(F_0)$	$E_{A_1,F_1}(H_1)$	$E_{F_1,D_1}(I_0)$	$E_{B_1,C_1}(J_0)$	$E_{I_1,E_1}(G_0)$	$E_{G_1,J_1}(K_0)$	$E_{H_1,K_1}(L_1)$	$E_{L_1,K_1}(M_0)$

19

Garbled Circuit

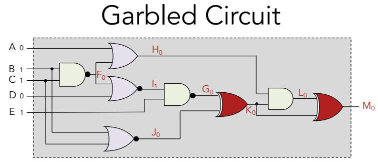


B	C	F	A	F	H	F	D	I	B	C	J	I	E	G	G	J	K	H	K	L	L	K	M
0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	1	0	0	1	0	0	1	1	0	1	1	0	1	0	0	1	1
1	0	1	1	0	1	1	0	0	1	0	0	1	0	1	1	0	1	1	0	0	1	0	1
1	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0

$E_{B_0,C_0}(F_1)$	$E_{A_0,F_0}(H_0)$	$E_{F_0,D_0}(I_1)$	$E_{B_0,C_0}(J_1)$	$E_{I_0,E_0}(G_1)$	$E_{G_0,J_0}(K_0)$	$E_{H_0,K_0}(L_0)$	$E_{L_0,K_0}(M_0)$
$E_{B_0,C_1}(F_1)$	$E_{A_0,F_1}(H_1)$	$E_{F_0,D_1}(I_0)$	$E_{B_0,C_1}(J_0)$	$E_{I_0,E_1}(G_1)$	$E_{G_0,J_1}(K_1)$	$E_{H_0,K_1}(L_0)$	$E_{L_0,K_1}(M_1)$
$E_{B_1,C_0}(F_1)$	$E_{A_1,F_0}(H_1)$	$E_{F_1,D_0}(I_0)$	$E_{B_1,C_0}(J_0)$	$E_{I_1,E_0}(G_1)$	$E_{G_1,J_0}(K_1)$	$E_{H_1,K_0}(L_0)$	$E_{L_1,K_0}(M_1)$
$E_{B_1,C_1}(F_0)$	$E_{A_1,F_1}(H_1)$	$E_{F_1,D_1}(I_0)$	$E_{B_1,C_1}(J_0)$	$E_{I_1,E_1}(G_0)$	$E_{G_1,J_1}(K_0)$	$E_{H_1,K_1}(L_1)$	$E_{L_1,K_1}(M_0)$

20

Garbled Circuit



B	C	F	A	F	H	F	D	I	B	C	J	I	E	G	G	J	K	H	K	L	L	K	M
0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	1	0	0	1	0	0	1	1	0	1	1	0	1	0	0	1	1
1	0	1	1	0	1	1	0	0	1	0	0	1	0	1	1	0	1	1	0	0	1	0	1
1	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0

$E_{B_0,C_0}(F_1)$	$E_{A_0,F_0}(H_0)$	$E_{F_0,D_0}(I_1)$	$E_{B_0,C_0}(J_1)$	$E_{I_0,E_0}(G_1)$	$E_{G_0,J_0}(K_0)$	$E_{H_0,K_0}(L_0)$	$E_{L_0,K_0}(M_0)$
$E_{B_0,C_1}(F_1)$	$E_{A_0,F_1}(H_1)$	$E_{F_0,D_1}(I_0)$	$E_{B_0,C_1}(J_0)$	$E_{I_0,E_1}(G_1)$	$E_{G_0,J_1}(K_1)$	$E_{H_0,K_1}(L_0)$	$E_{L_0,K_1}(M_1)$
$E_{B_1,C_0}(F_1)$	$E_{A_1,F_0}(H_1)$	$E_{F_1,D_0}(I_0)$	$E_{B_1,C_0}(J_0)$	$E_{I_1,E_0}(G_1)$	$E_{G_1,J_0}(K_1)$	$E_{H_1,K_0}(L_0)$	$E_{L_1,K_0}(M_1)$
$E_{B_1,C_1}(F_0)$	$E_{A_1,F_1}(H_1)$	$E_{F_1,D_1}(I_0)$	$E_{B_1,C_1}(J_0)$	$E_{I_1,E_1}(G_0)$	$E_{G_1,J_1}(K_0)$	$E_{H_1,K_1}(L_1)$	$E_{L_1,K_1}(M_0)$

21

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Computation Approaches

Multi-Party Computation (MPC)

Pros
- Low compute requirements
- Easy to accelerate
- Provably secure
- Supports multiple threat models
- Easy to map existing algorithms

Cons
- High communication costs
- High latency
- Information theoretic proofs are weaker than PKE ones

Fully Homomorphic Encryption (FHE)

Pros
- Very low communication costs
- Requires a single round of communications, i.e., "fire and forget"
- Useful when one side is limited in compute / memory / storage
- Provably secure – relies on strength of PKE

Cons
- Very high computational requirements
- Harder to accelerate
- Mapping existing algorithms to FHE may be difficult

Trusted Execution Environments (TEE)

Pros
- No communication required
- Trivial to accelerate
- Great support for existing software

Cons
- Weaker security guarantees
- Cannot stop determined adversaries
- Historically plagued by vulnerabilities and breaches
- Long term deployment is difficult – TEE's can 'run out' of entropy / CRP's, etc.

22

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Computation Approaches

Multi-Party Computation (MPC)

Pros
- Low compute requirements
- Easy to accelerate
- Provably secure
- Supports multiple threat models
- Easy to map existing algorithms

Cons
- High communication costs
- High latency
- Information theoretic proofs are weaker than PKE ones

Fully Homomorphic Encryption (FHE)

Pros
- Very low communication costs
- Requires a single round of communications, i.e., "fire and forget"
- Useful when one side is limited in compute / memory / storage
- Provably secure – relies on strength of PKE

Cons
- Very high computational requirements
- Harder to accelerate
- Mapping existing algorithms to FHE may be difficult

Trusted Execution Environments (TEE)

Pros
- No communication required
- Trivial to accelerate
- Great support for existing software

Cons
- Weaker security guarantees
- Cannot stop determined adversaries
- Historically plagued by vulnerabilities and breaches
- Long term deployment is difficult – TEE's can 'run out' of entropy / CRP's, etc.

23

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- For the Two-party secure multiparty computation
- Assume
  - Alice has x, Bob has y, and they want to compute two functions $f_A(x,y)$ and $f_B(x,y)$
    - It could be the same function f(x,y)
  - The desired outcome is that at the end of the protocol
    - Alice learns the result of her function $f_A(x,y)$ and not Bob's input y
    - Bob learns the result of his function $f_B(x,y)$ and not Alice's input x

24

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- For the Two-party secure multiparty computation
- Assume
  - Alice has x, Bob has y, and they want to compute two functions $f_A(x,y)$ and $f_B(x,y)$
    - It could be the same function f(x,y)
- Illustration
  - Alice represents the function f(x,y) as a garbled circuit
  - She then sends the circuit and values corresponding to her input bits to Bob
  - Bob evaluates the circuits using the sent Alice's bits and his own input bits
  - He then transfers the result to Alice

25

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- For the Two-party secure multiparty computation
- Assume
  - Alice has x, Bob has y, and they want to compute two functions $f_A(x,y)$ and $f_B(x,y)$
    - It could be the same function
- The set up for the n-party secure multiparty computation makes the same assumptions
  - Here instead of just Alice and Bob, there are n parties
  - Each party with a private input
  - And they want to jointly compute the function
    $f_X=(x_1, \ldots, x_n)$

26

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

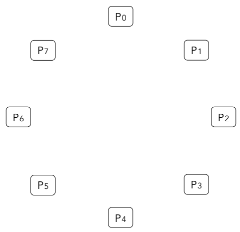
- Validity
  - Secure function evaluation (SFE) system must be able to correctly computed
    - For example, result must be computed with inputs from at least all correct parties
- Privacy
  - P_1 and P_2 cannot know each others input ip_1, ip_2
- Agreement
  - Result must be same for all parties (P_1 and P_2)
- Termination
  - All active parties (P_1 and P_2) eventually receive final result
- Fairness
  - P_1 should not be able to learn the result while denying it to P_2

27

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - Let us have 8 parties $P_1, \dots, P_7$ that want to perform a joint computation

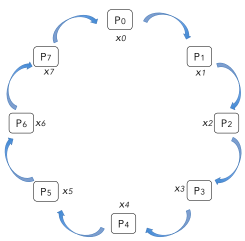


28

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - Let us have 8 parties $P_0, \dots, P_7$ that want to perform a joint computation
  - Each party P_i with i ∈ [0..7], has private input x_i

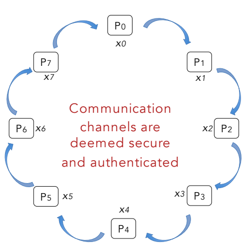


29

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - Let us have 8 parties $P_0, \dots, P_7$ that want to perform a joint computation
  - Each party P_i with i ∈ [0..7], has private input x_i



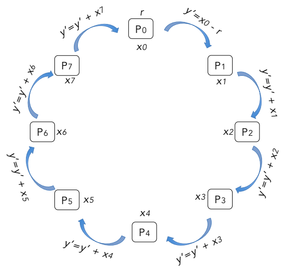
30

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - r is a random number



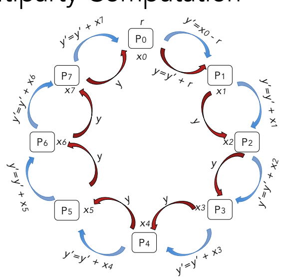
31

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - r is a random number



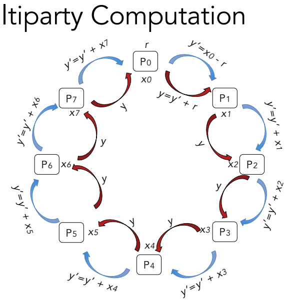
32

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - r is a random number
  - If any P_i is semi-honest or malicious, then these messages may not be passed along properly or be modified in a way that break the protocol

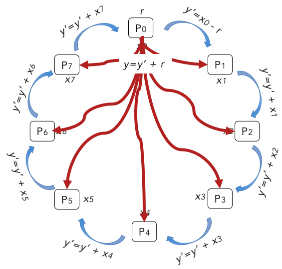


33

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - Result distribution could be faster

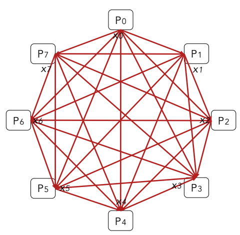


34

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - Even fast compute



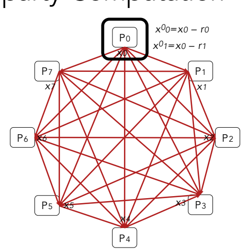
35

STAM Center — SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - *The parties can use a linear secret sharing scheme to create a distributed state of their inputs*
  - *For each party, the random variables r_i are different*

$x^0_2 = x_0 - r_2$ $x^0_3 = x_0 - r_3$
$x^0_4 = x_0 - r_4$ $x^0_5 = x_0 - r_5$
$x^0_6 = x_0 - r_6$ $x^0_7 = x_0 - r_7$



36

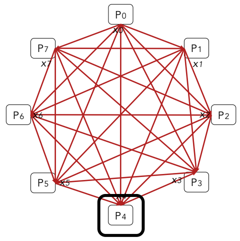
STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - *The parties can use a linear secret sharing scheme to create a distributed state of their inputs*
  - *For each party, the random variables r_i are different*

$x^4_0 = x_4 - r_0$ $x^4_3 = x_4 - r_3$
$x^4_1 = x_4 - r_1$ $x^4_4 = x_4 - r_4$
$x^4_2 = x_4 - r_2$ $x^4_5 = x_4 - r_5$
$x^4_6 = x_4 - r_6$ $x^4_7 = x_4 - r_7$



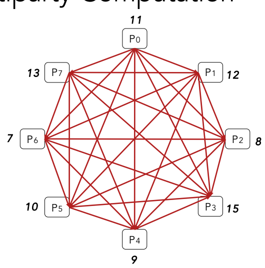
37

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Construction of the computation
  - Let us have 8 parties P_1, . . . , P_7 that want to perform a joint computation
  - Let us do summation



38

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

Private Inputs		P0	P1	P2	P3	P4	P5	P6	
11	P0								
12	P1								
8	P2								
15	P3								
9	P4								
10	P5								
7	P6								
13	P7								
				Local Total					

39

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering
Arizona State University

Secure Multiparty Computation

Private Inputs		P_0	P_1	P_2	P_3	P_4	P_5	P_6	
11	P_0	-1	1	4	3	1	0	3	0
12	P_1								
8	P_2								
15	P_3								
9	P_4								
10	P_5								
7	P_6								
13	P_7								

Local Total

40

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering
Arizona State University

Secure Multiparty Computation

Private Inputs		P_0	P_1	P_2	P_3	P_4	P_5	P_6	
11	P_0	-1	1	4	3	1	0	3	0
12	P_1	3	-5	1	2	4	0	2	5
8	P_2	1	0	0	0	2	3	1	1
15	P_3	4	3	1	-4	3	2	2	4
9	P_4	1	1	3	0	2	0	1	1
10	P_5	2	4	0	1	2	-2	3	0
7	P_6	1	0	5	2	0	1	-5	3
13	P_7	1	2	3	2	1	1	3	0

Local Total

41

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering
Arizona State University

Secure Multiparty Computation

Private Inputs		P_0	P_1	P_2	P_3	P_4	P_5	P_6	
11	P_0	-1	1	4	3	1	0	3	0
12	P_1	3	-5	1	2	4	0	2	5
8	P_2	1	0	0	0	2	3	1	1
15	P_3	4	3	1	-4	3	2	2	4
9	P_4	1	1	3	0	2	0	1	1
10	P_5	2	4	0	1	2	-2	3	0
7	P_6	1	0	5	2	0	1	-5	3
13	P_7	1	2	3	2	1	1	3	0
85		**12**	**5**	**17**	**6**	**15**	**5**	**10**	**14**

Local Total

42

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering
Arizona State University

Secure Multiparty Computation

- There are two major adversary models for secure computation
  - Semi-honest/passive model
    - Follows all required steps
    - Looks for all advantageous information leaked
    - Assumed to be selfish
  - Fully malicious/active model
    - Arbitrarily deviates from the protocol
    - Aborts the protocol at anytime
    - Takes any step that runs counter to the desirable outcome

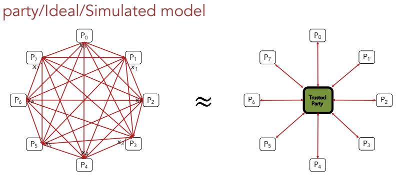
43

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering
Arizona State University

Secure Multiparty Computation

- The multiparty computation is secure if it emulates the trusted central party model to a negligible error range
  - If the two are shown to be indistinguishable
  - Trusted party/Ideal/Simulated model



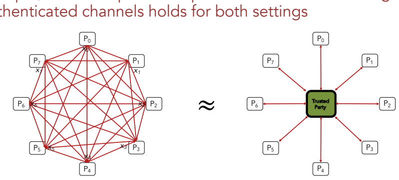
44

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Ira A. Fulton Schools of Engineering
Arizona State University

Secure Multiparty Computation

- The security multiparty computation protocol is also evaluated though the simulated model
  - For example, the assumption that parties communicate through secure and authenticated channels holds for both settings



45

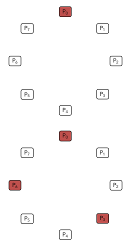
STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

Secure Multiparty Computation

- Dealing with semi-honest and malicious

D. Chaum, C. Crépeau, and I. Damgard. Multiparty unconditionally secure protocols. In Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC '88)

M. Ben-Or, S. Goldwasser, and A. Wigderson Completeness theorems for non-cryptographic fault-tolerant distributed computation. In Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC '88)

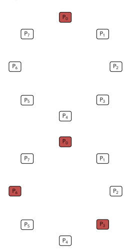


46

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

Secure Multiparty Computation

- Dealing with semi-honest and malicious
  - Any function $f(x_1, \ldots, x_n)$ can be securely computed in a semi-honest setting if the majority is honest
    - The passive adversary controls less than n/2 of the parties
  - Any function $f(x_1, \ldots, x_n)$ can be securely computed if the adversary actively controls less than n/3 of the parties

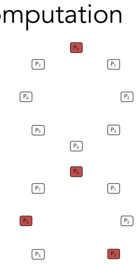


47

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

Secure Multiparty Computation

- It is a rich area of research
  - Secure multiparty computation over groups, fields, rings
  - Authentication of the communication channels
  - Synchronous versus asynchronous messaging
  - And many more sub-topics



48

Secure Multiparty Computation

- Commitment
  - Let p and q be two large prime numbers such that q divides p-1
  - Generator g of the order-q subgroup of Zp*
  - A secret s from Zp such that $y=g^s$ mod p
  - Where the values p,q,g, and y are public
  - There is only one secret s in the system residing with Bob

Alice —— {M} ——→ **Bob**

M = $g^x y^r$ mod p

Alice commits to some x∈Zq
Then selects a random r ∈Zq

Bob now has M

49

Secure Multiparty Computation

- Commitment
  - Let p and q be two large prime numbers such that q divides p-1
  - Generator g of the order-q subgroup of Zp*
  - A secret s from Zp such that $y=g^s$ mod p
  - Where the values p,q,g, and y are public
  - There is only one secret s in the system residing with Bob

Alice —— {M} ——→ **Bob**

M = $g^x y^r$ mod p

Alice commits to some x∈Zq
Then selects a random r ∈Zq

Bob now has M

—— **{ x, r }** ——→ Bob can verify that $M= g^x(g^s)^r=g^{x+sr}$ mod p

Alice reveals x and r

50

Secure Multiparty Computation

- Zero-Knowledge
  - Let p and q be two large prime numbers such that q divides p-1
  - Generator g of the order-q subgroup of Zp*

Alice —— {U= **g^r mod p** } ——→ **Bob**

Alice knows a number s such that **M = g^s mod p**
and wants to prove it to Bob

r is random number ∈ [1..q]

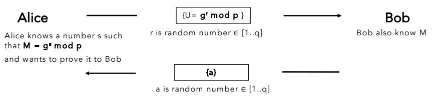
Bob also know M

51

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Zero-Knowledge
  - Let p and q be two large prime numbers such that q divides p-1
  - Generator g of the order-q subgroup of Zp*

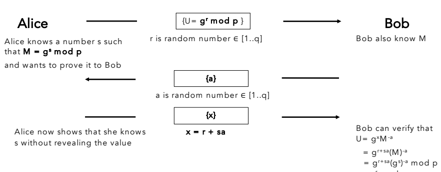


52

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Zero-Knowledge
  - Let p and q be two large prime numbers such that q divides p-1
  - Generator g of the order-q subgroup of Zp*

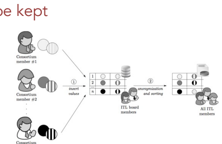


53

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | ASU Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Use Case
  - In order to analyze the economic situation of an industrial sector, a secure system is needed for jointly collecting and analyzing sensitive financial data
  - The financial data should be kept
    - Confidential
    - Anonymous



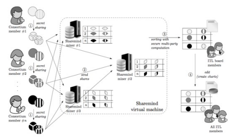
Deploying secure multi-party computation for financial data analysis
D. Bogdanov, R. Talviste and J. Willemson

54

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | Ira A. Fulton Schools of Engineering, Arizona State University

Secure Multiparty Computation

- Use Case
  - Improved version
    - Data stored/sorted separately on three servers
    - No single party has access to original data
    - Anonymous to the board members



Deploying secure multi-party computation for financial data analysis
D. Bogdanov, R. Talviste and J. Willemson

55

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | Ira A. Fulton Schools of Engineering, Arizona State University

Secure Computation Approaches

Multi-Party Computation (MPC)

Pros
- Low compute requirements
- Easy to accelerate
- Provably secure
- Supports multiple threat models
- Easy to map existing algorithms

Cons
- High communication costs
- High latency
- Information theoretic proofs are weaker than PKE ones

Fully Homomorphic Encryption (FHE)

Pros
- Very low communication costs
- Requires a single round of communications, i.e., "fire and forget"
- Useful when one side is limited in compute / memory / storage
- Provably secure – relies on strength of PKE

Cons
- Very high computational requirements
- Harder to accelerate
- Mapping existing algorithms to FHE may be difficult

Trusted Execution Environments (TEE)

Pros
- No communication required
- Trivial to accelerate
- Great support for existing software

Cons
- Weaker security guarantees
- Cannot stop determined adversaries
- Historically plagued by vulnerabilities and breaches
- Long term deployment is difficult – TEE's can 'run out' of entropy / CRP's, etc.

56

STAM Center SECURE, TRUSTED, AND ASSURED MICROELECTRONICS | Ira A. Fulton Schools of Engineering, Arizona State University

Upcoming Lectures

- Secure Computation Approaches
  - Trusted Execution Environment (TEE)
  - Homomorphic Encryption

57