

Cryptography CS 555

Lecture 22



Secure Function Evaluation

Lecture Outline

- The Secure Function Evaluation Problem
- 1-out-2 Oblivious Transfer
- Yao's Scrambled Circuits for 2-party SFE
- Secret Sharing
- n-party SFE



Secure Function Evaluation

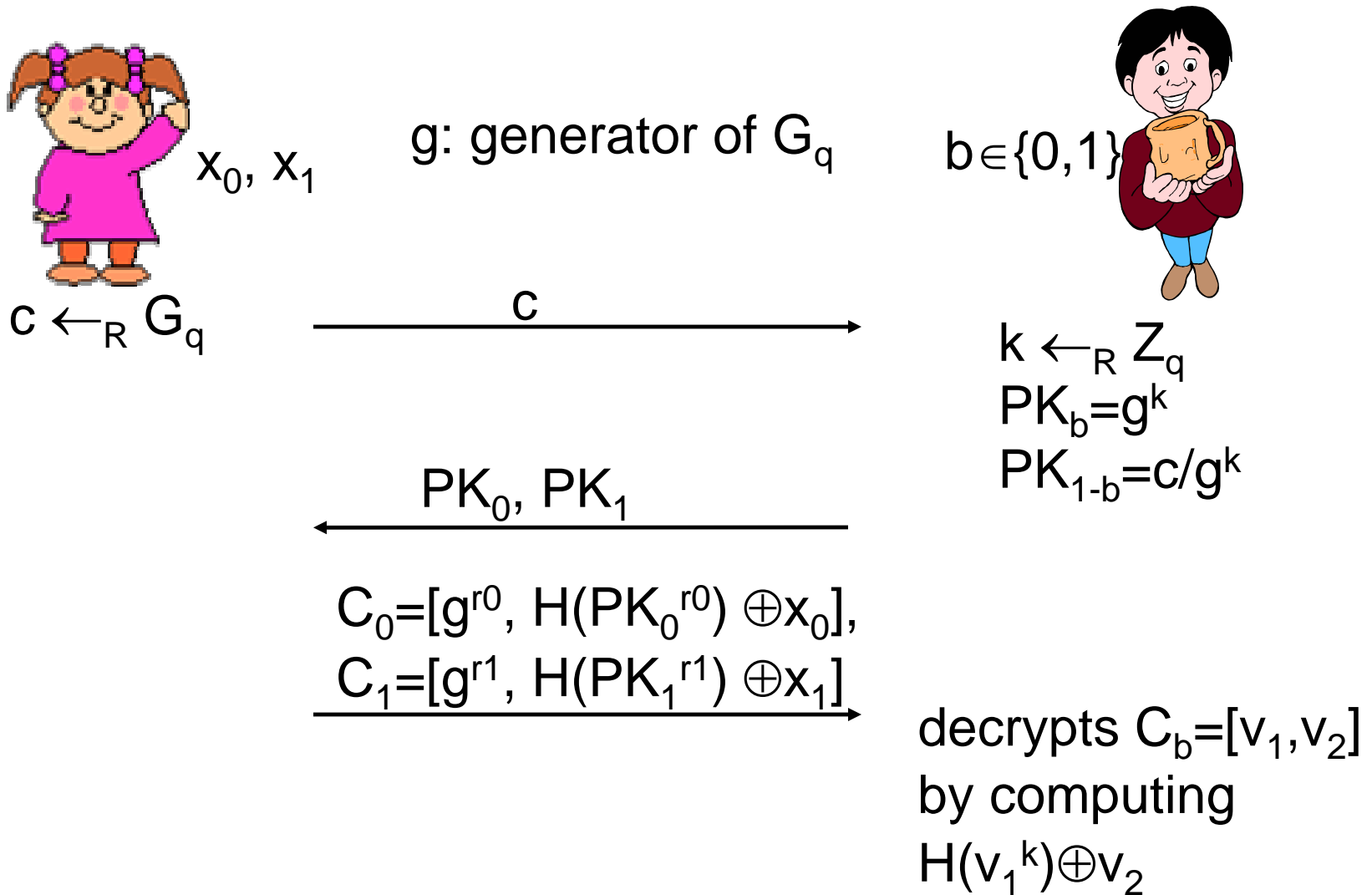
- Also known as Secure Multiparty Computation
- 2-party SFE: Alice has x , Bob has y , and they want to compute two functions $f_A(x,y)$, $f_B(x,y)$. At the end of the protocol
 - Alice learns $f_A(x,y)$ and nothing else
 - Bob learns $f_B(x,y)$ and nothing else
- n-party SFE: n parties each have a private input, and they join compute functions

Oblivious Transfer

- 1 out of 2 OT
 - Alice has two messages x_0 and x_1
 - At the end of the protocol
 - Bob gets exactly one of x_0 and x_1
 - Alice does not know which one Bob gets

- 1 out of n OT
 - Alice has n messages
 - Bob gets exactly one message, Alice does not know which one Bob gets

Bellare-Micali 1-out-2-OT protocol



Yao's Scrambled Circuit Protocol for 2-party SFE

- For simplicity, assume that Alice has x , Bob has y , Alice learns $f(x,y)$, and Bob learns nothing
 - represent $f(x,y)$ using a boolean circuit
 - Alice encrypts the circuit and sends it to Bob
 - in the circuit each wire is associated with two random values
 - Alice sends the values corresponding to her input bits
 - Bob uses OT to obtain values for his bits
 - Bob evaluates the circuits and send the result to Alice

Secret Sharing

- t-out-of-n secret sharing
 - divides a secret s into n pieces so that any t pieces together can recover s
- How to do n-out-of-n secret sharing?
- Shamir's secret sharing scheme
 - secret $s \in \mathbb{Z}_p$
 - pick a random degree $t-1$ polynomial $f \in \mathbb{F}_p[x]$ s.t. $f(0)=s$
 - user i gets $s_i=f(i)$
 - t users can interpolate f and find out s
 - $t-1$ shares reveal no information about s

Proactive Secret Sharing

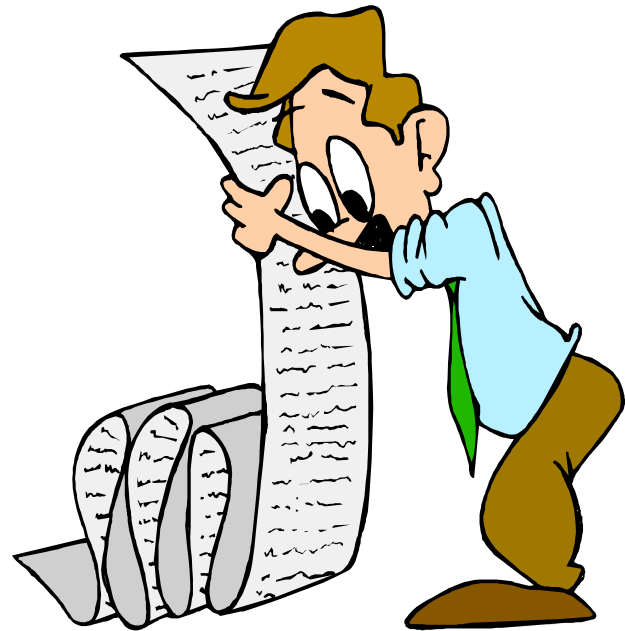
- Suppose that s is shared in t -out-of- n
- User i has $s_i=f(i)$
- Proactive updates:
 - user 1 picks random degree $t-1$ polynomial s.t. $g(t)=0$
 - user 1 sends $y_j=g(j)$ to user j
 - user j does $s_j^{\text{new}}=s_j^{\text{old}}+y_j$

BGW n-party SFE

- Use algorithmic circuits where operations are $+$ and \times
- Each private input is shared among all participants
- Do computation with the shared value
 - e.g., given x and y both are shared by n parties, compute the shares of $x+y$ and $x\times y$
- Secure when the majority of the parties are honest

Summary

- Yao's 2-part SFE uses OT and block cipher
- BGW's n-party SFE uses secret sharing



Next Lecture...

- Quantum Cryptography

