Memory Delegation

Kai-Min Chung
Cornell University

Feng-Hao Liu
Brown University

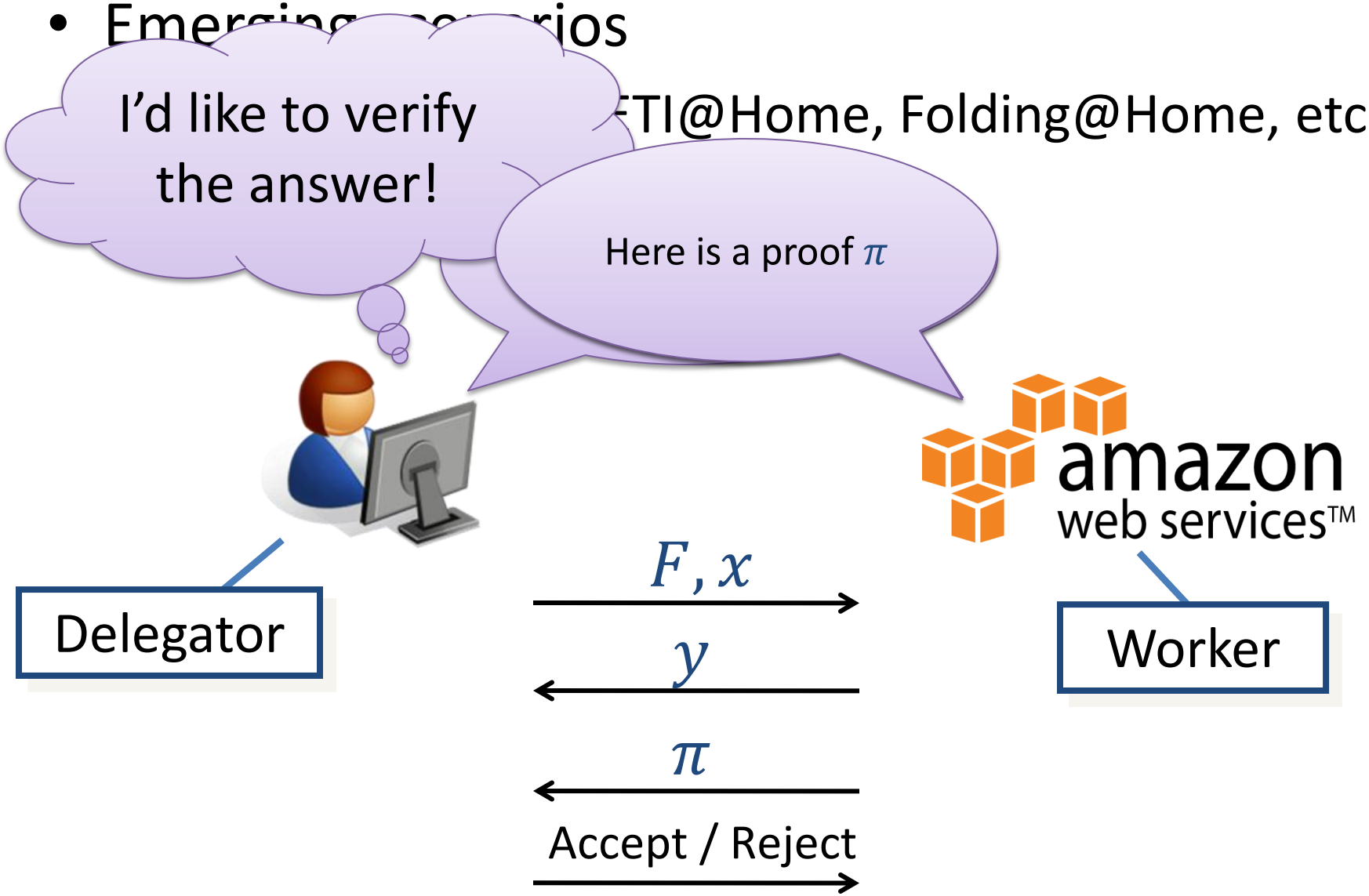
Yael Kalai
Microsoft Research

Ran Raz
Weizmann Inst. of Science

Delegation of Computation

- Emerging scenarios

-TI@Home, Folding@Home, etc…



Important Properties

- Computational Efficiency
  - verification *must be faster* than computation
  - want small overhead for the worker
- Interaction
  - can the proofs be non-interactive ?
- Genera[illegible]
  - can w[illegible]gate all functions?
- Assumptions
  - [illegible]umptions do we need?

Delegator

Worker

F, x

y

π

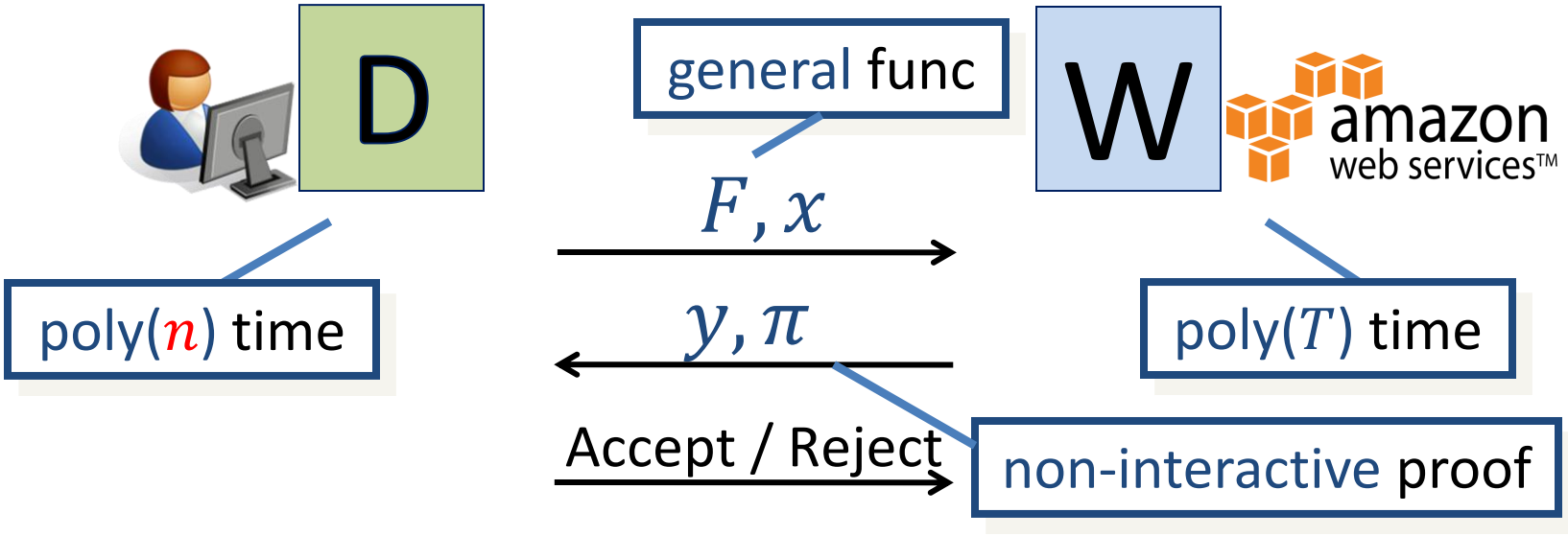
Accept / Reject

n: length of input x

T: time complexity of F

Holy Grail of Comp. Delegation

n: length of input x, T: time complexity of F



- Completeness: D accepts correct y, π w.p. 1
- Soundness: $\forall$ poly(T)-time W^*,

  Pr[D accepts *wrong* answer] $\leq$ ngl

Previous Results on Comp. Del.

Results	Trade-offs
GKR scheme [GKR '08, KR '09]	Non-interactive proofs ☺ For low-depth functions ☹
Universal Arguments [K '92,M '94,BG '02]	4-message interactive proofs ☹ For general functions ☺
Offline/Online [GGP '10, CKV '10, AIK'10]	With (inefficient) offline preprocessing ☹ Non-interactive & for general functions* ☺

All above results are *efficient*, but require *assumptions*

(*) W^* is not allowed to learn the decision bits of D

The Goal of Delegation

- Holy grail of computation delegation:
  - Can we achieve *efficient* and *non-interactive* computation delegation for *general* functions under reasonable assumptions ?

Delegator runs in $O(n)$ time

- We don't know the answer to this question yet. But we want *more*!

Delegator should run in $o(n)$ time !

When data x is large and in the cloud…

Can D delegate the data x as well,

only keep Cert(x) & verify in $o(N)$ time?

D

W

Gmail by Google

$x =$ All e

Ce

e–mails

o(N) t

Yes, we can!

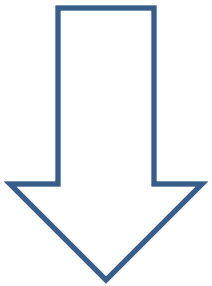
Memory Delegation

Streaming Delegation

N: length of input x, T: time complexity of F

Our Main Results

GKR Scheme & Universal Argument
as Computation Delegation Schemes



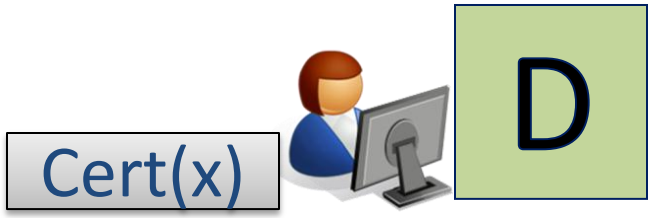
GKR Scheme & Universal Argument
as Memory/Streaming Delegation Schemes

Outline

- Computation Delegation
- Memory Delegation
- Streaming Delegation
- Conclusion

Memory Delegation

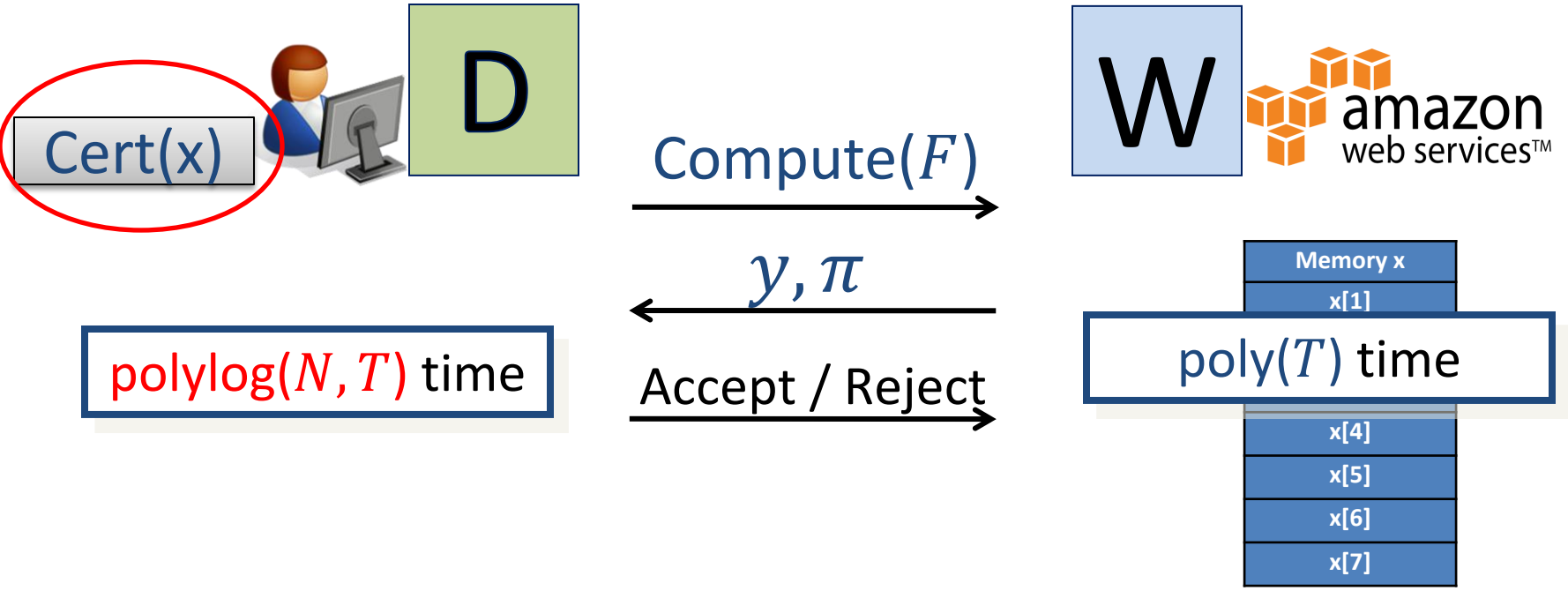
- Initial memory x holds by delegator D
- D computes a certificate Cert(x)
- D sends x to worker W



Memory x
x[1]
x[2]
x[3]
x[4]
x[5]
x[6]
x[7]

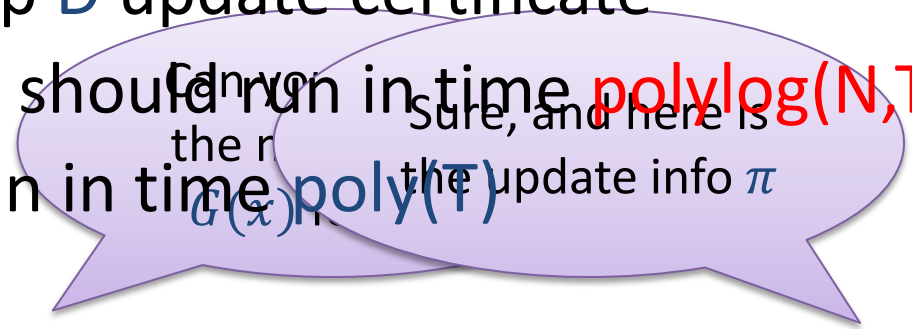
Compute Operation

- D can verify π using certificate Cert(x)
- Efficiency: D should run in time polylog(N,T)
- W should run in time poly(T)



Update Operation

- Allow D sends a general update function G to W
- Allow W help D update certificate
- *Efficiency:* D should run in time polylog(N,T)
- W should run in time poly(T)



D

Cert(x)

⇓

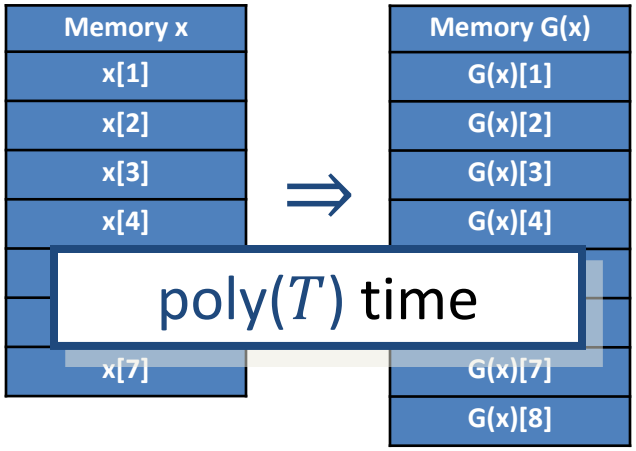
Cert(G(x))

Update(G)

π

Accept / Reject

W



polylog(N, T) time

Desired Properties

- Efficiency
  - D runs in time polylog(N,T)
  - W runs in time poly(T)
- Completeness: D always accepts when W honest
- *Reusable* Soundness: soundness game for D and W^*
  - W^* can chooses inputs of D during interaction
  - W^* learns the decision of D
  - W^* wins if D ever accepts mistakenly
  - ∀ poly(T)-time W^* can win with negligible probability

N: length of memory x, T: time complexity of F, G

Issue of Reusability

- D uses cert(x) to compute his decision

  ⇒ one bit *leakage* info about cert(x) *per input*

- Our memory scheme has *public* cert(x)
  - Simple!
- Our streaming scheme has *secret* cert(x)
  - Challenging! Take ideas from continual-leakage model.
  - New *geometric lemma* "dual" to [BKKV '10]
  - New *entropy lemma* for lower bounding conditional computational entropy

Our Memory Delegation Schemes

Under cryptographic assumptions*, we obtain *efficient* memory delegation schemes with

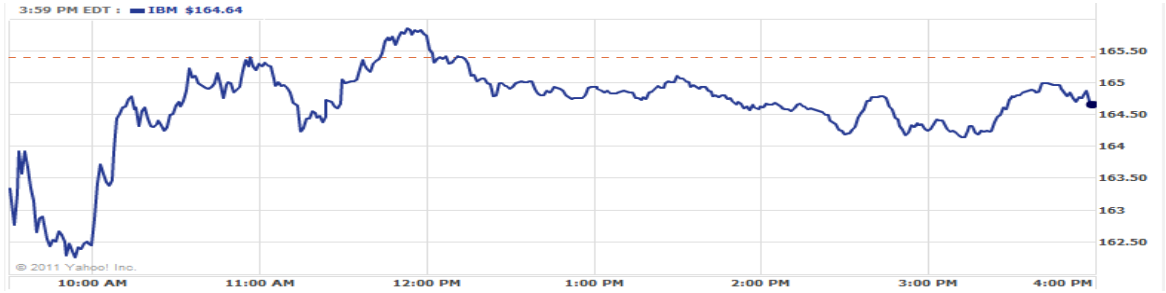
Our Schemes	Property
Based on GKR scheme	Non-interactive proofs ☺ For low-depth functions ☹
Based on Universal Arguments	4-message interactive proofs ☹ For general functions ☺

(*) Based on the same assumptions as the corresponding schemes

Outline

- Computation Delegation
- Memory Delegation
- Streaming Delegation
- Conclusion

Example: Streaming of Stock Ticks



-0.1

Should I buy the stock now?

amazon web services™

...

Comparison to Memory Delegation

- Data stream arrives constantly at a high rate
  ⇒ Ideally, D should update certificate by ***himself***

- Luckily we can!
  - every update simply appends a data item x_t

- Different from memory delegation
  - Recall update for memory delegation is general
  - D gets help from W

Our Streaming Delegation Schemes

Assume the existence of

fully homomorphic encryption schemes [G '09]

Our Schemes	Property
Based on GKR scheme	Non-interactive proofs ☺ For low-depth functions ☹
Based on Universal Arguments	4-message interactive proofs ☹ For general functions ☺

Outline

- Computation Delegation
- Memory Delegation
- Streaming Delegation
- Conclusion

Conclusion

- We construct efficient memory/streaming delegation schemes
  - non-interactive for low depth functions
  - 4-message for general functions

- Can we achieve the holy grail of computation/memory/ streaming delegation?
  - *efficient* and *non-interactive* schemes for *general* functions

Thanks you!