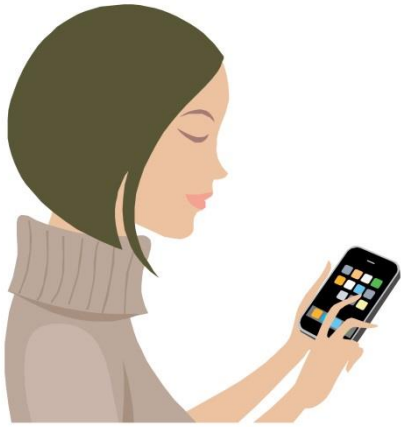
Enabling data sharing with secure computation

Vlad Kolesnikov
Bell Labs

DIMACS/Northeast Big Data Hub Workshop on Privacy and Security for Big Data
Apr 25, 2017

Data sharing: service providers

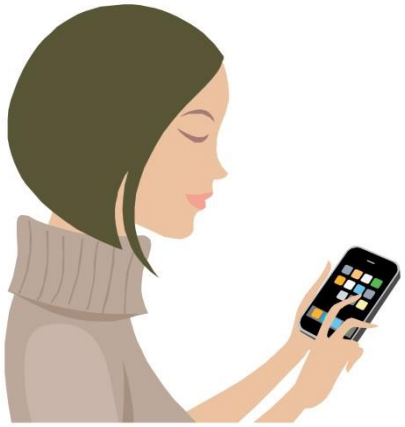


You are near Starbucks; here is a special

Legislation may require user consent ***each time*** for Location-Based Service
(E.g. SK Telecom, Korea)

Data sharing: service providers

Compliant location-based service:



May I use your location now?

OK

Here is a Starbucks coupon

Never mind, I don't want coupons

Data sharing: private DB queries

I want to query patient records

HIPAA protects patient privacy.
Only certain queries are OK.
What is your query?

My queries are private

Data sharing: enterprise

Ad campaign:
I have a list of my customers.
Display an upgrade offer to those who have researched FIOS.

Neither company wishes to share customer lists and histories.
FB protects data by instead exchanging hashes of data.

Utility *and* privacy

Ask a Trusted Third Party for help.

UserList F

UserList C

$F \cap C$

$\perp$

COMCAST

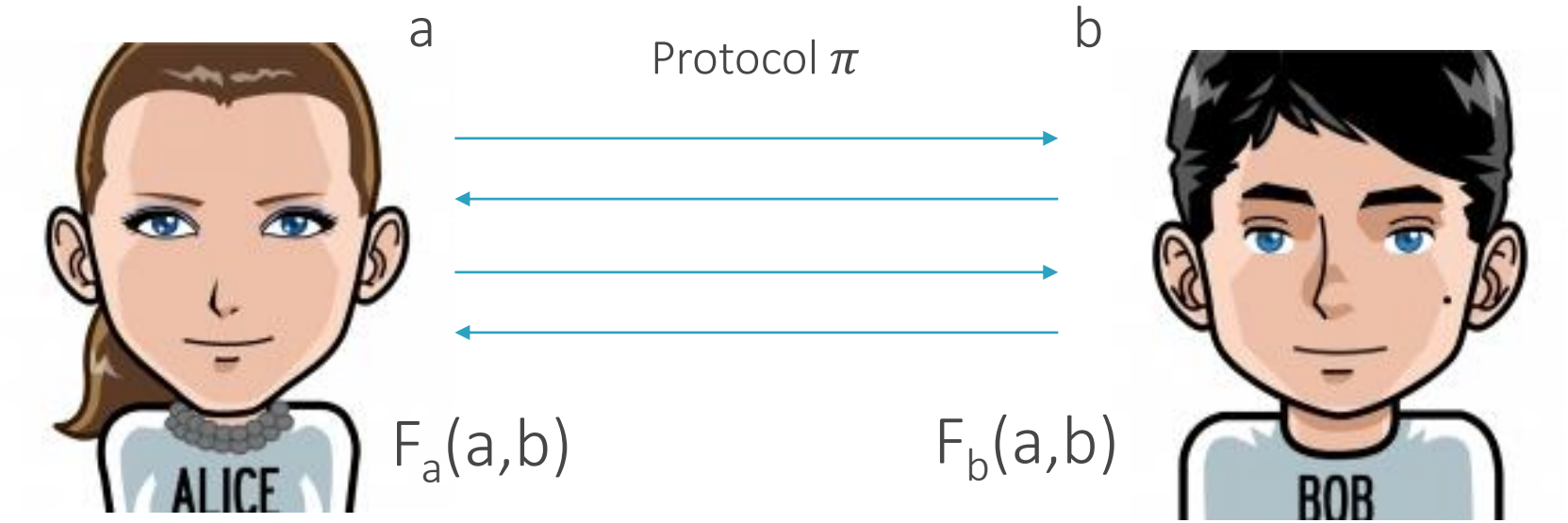
"Any task involving a Trusted Third Party can also be implemented using a cryptographic protocol **without any loss of security**."

[Yao86] [Goldreich Micali Wigderson 87]

Outline

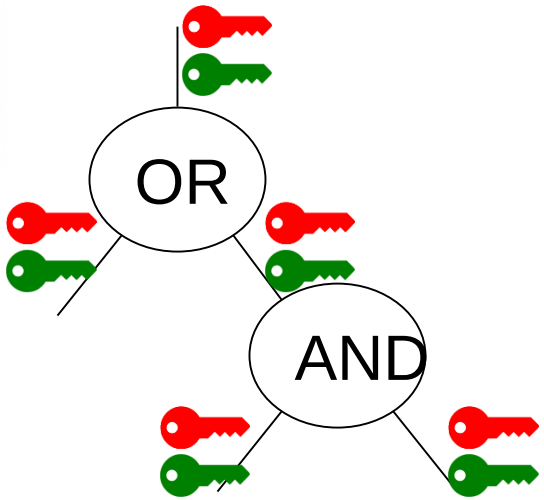
- Privacy and security enables data sharing
- Secure multi-party computation (MPC)
  - Approaches and progress
- MPC for big(ger) data: private DB (if time)

Secure computation



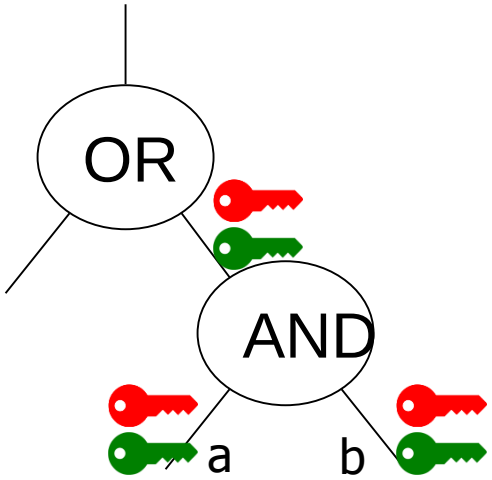
Garbled circuit: computation under encryption [Yao86]

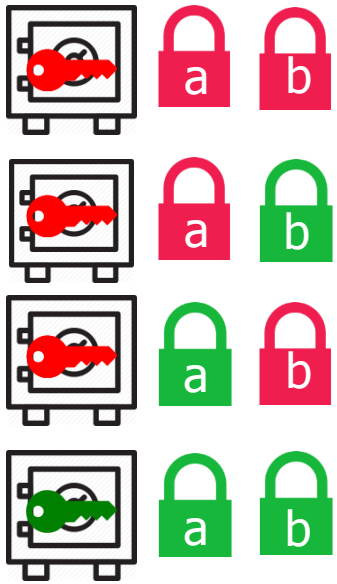
Circuit for F



Alice encrypts Boolean wire signals

Garbled circuit: computation under encryption [Yao86]





a	b	a^b
0	0	0
0	1	0
1	0	0
1	1	1

Alice encrypts Boolean gates (truth tables)

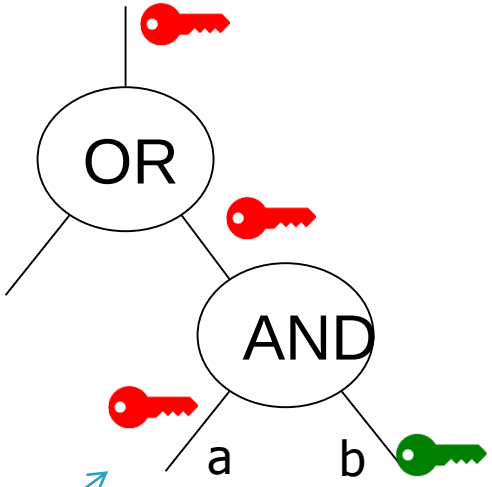
Goal: allow Bob to compute correct gate output key from input keys

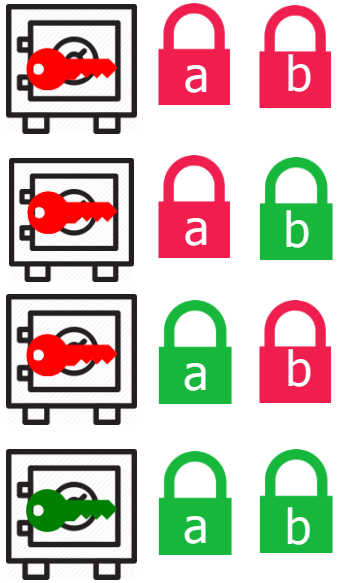
Garbled circuit: computation under encryption [Yao86]

Decoding table for output wire

0

1





a	b	a^b
0	0	0
0	1	0
1	0	0
1	1	1

a is Alice's input

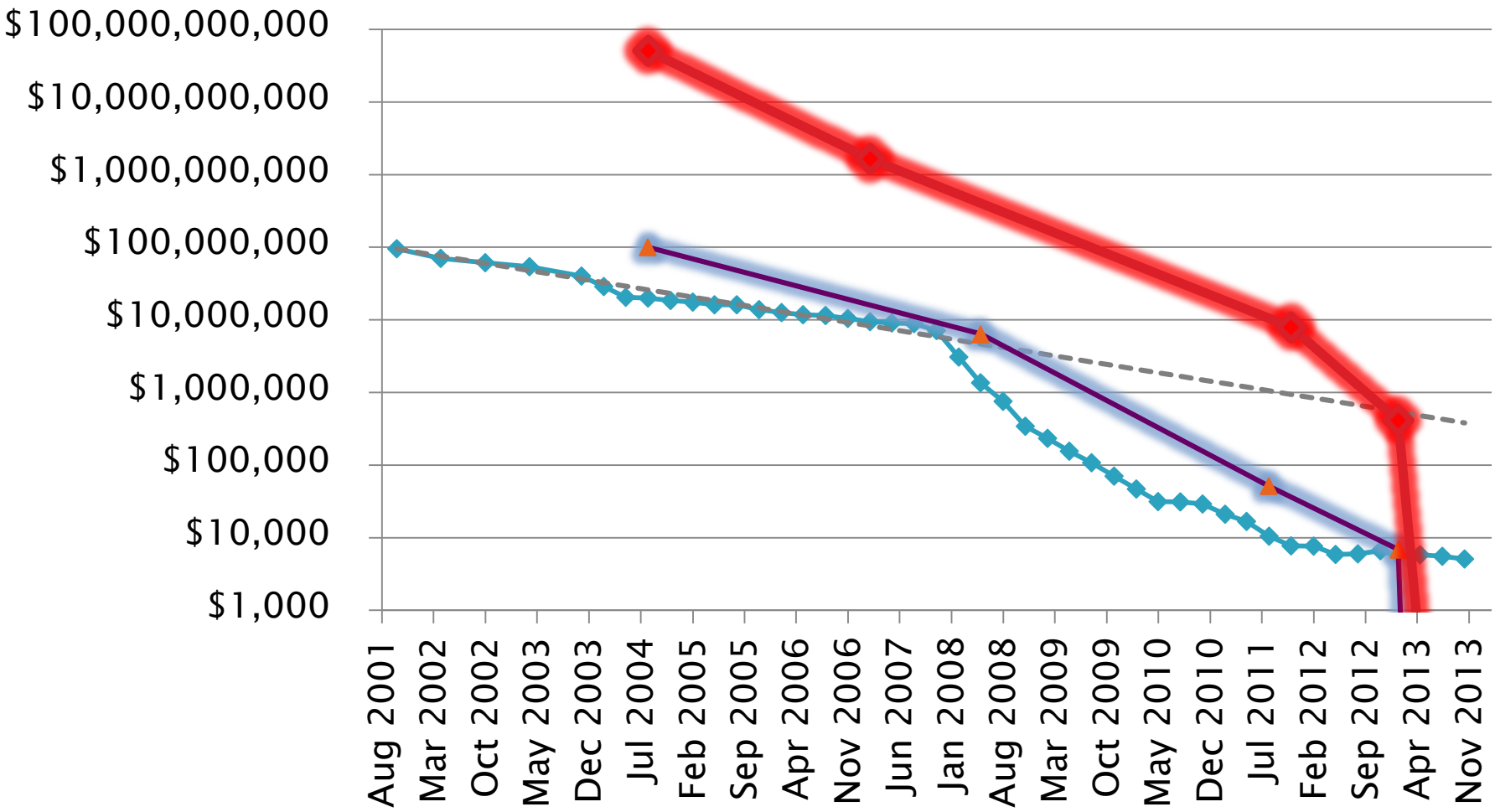
Alice sends this key

b is Bob's input

Alice and Bob run Oblivous Transfer (OT)

Bob receives key, while Alice learns nothing.

MPC progress



Cost to sequence genome
Estimates and chart by Dave Evans (UVA)

Cheating opportunities

F(a,b)

Alice can send a GC implementing wrong F
Bob cannot tell!

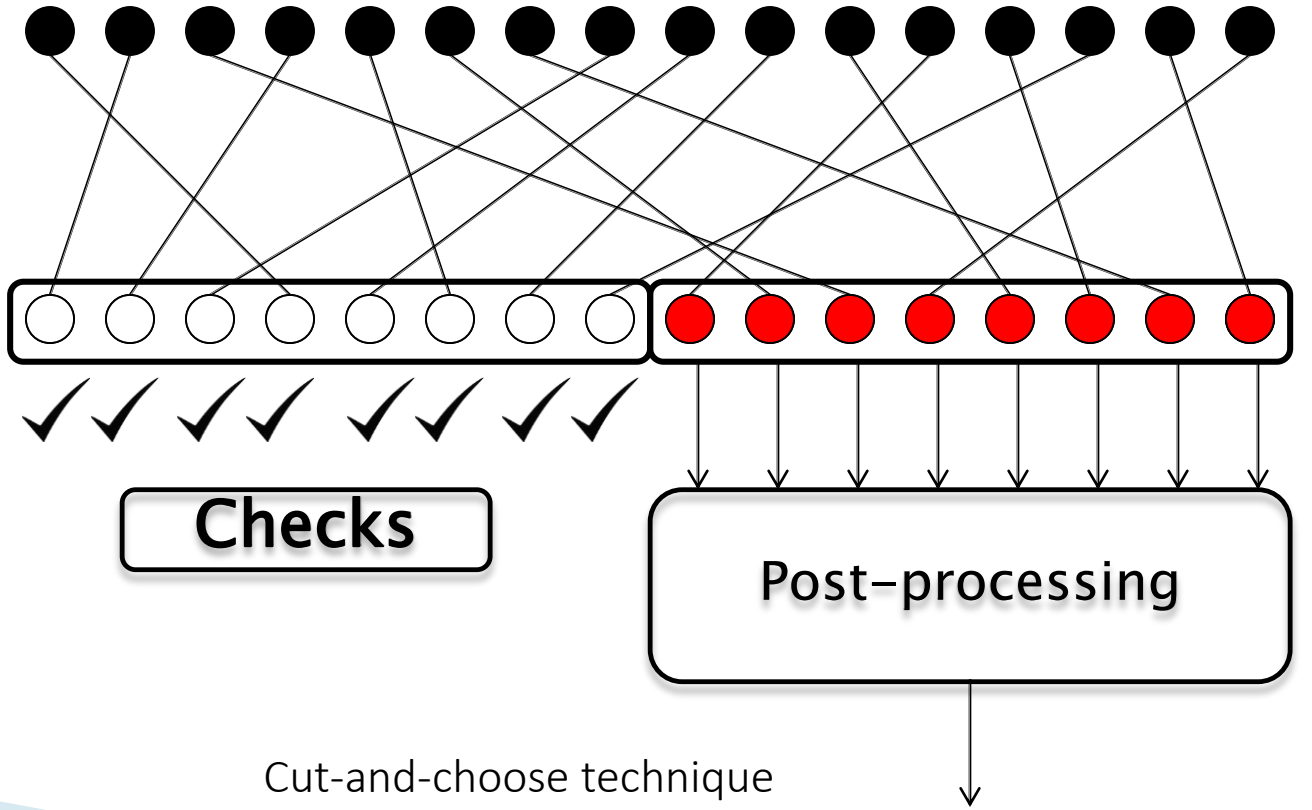
Bob only decrypts
- cheating not possible
- only abort

Catch me if you can!

Alice generates many copies of garbled circuits

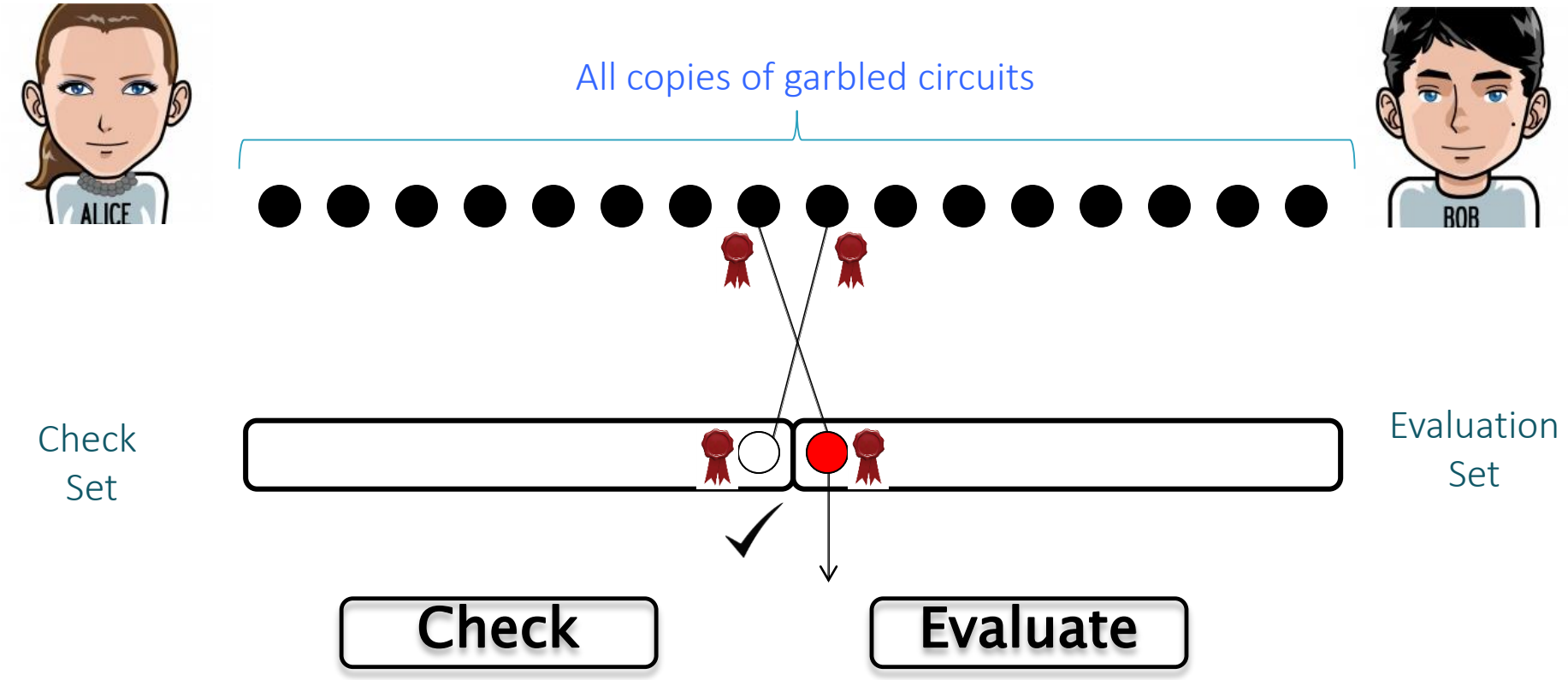
Check Set

Evaluation Set



Cut-and-choose technique
40 Circuits need to be sent to prevent cheating by Alice

Publicly verifiable covert (PVC) MPC [K Malozemoff15]

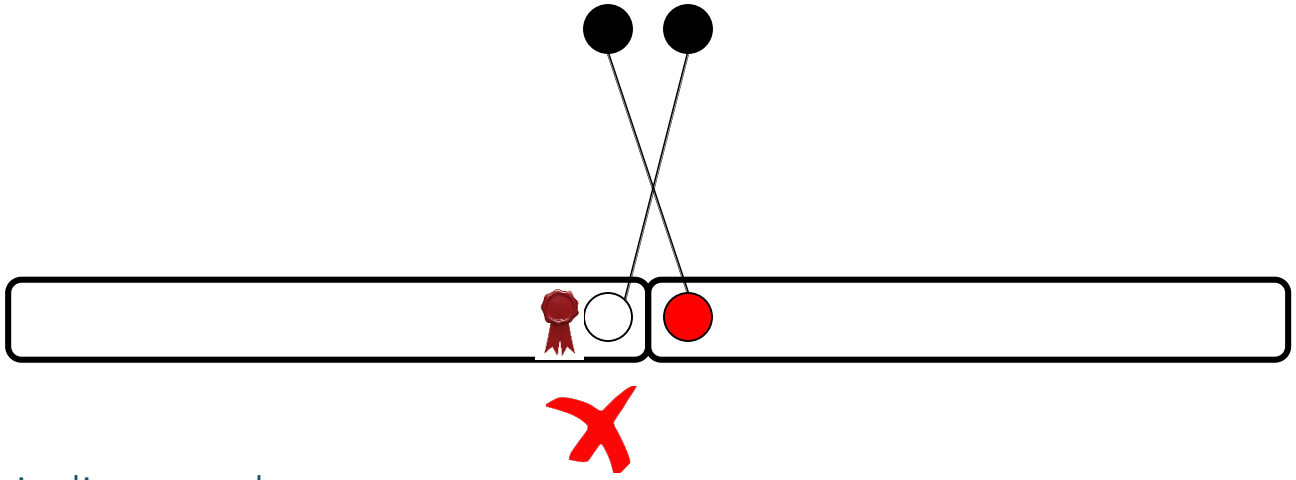


Idea: Alice can cheat, but caught w prob 50%
If caught, Bob gets irrefutable *publicly verifiable* proof of cheating.

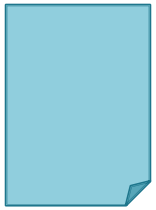
Publicly verifiable covert (PVC) MPC [KM15]

All copies of garbled circuits

Check Set



Evaluation Set



If cheating is discovered
irrefutable ***publicly verifiable*** **proof of cheating** can be produced

Informal Theorem [KM15]: P is a secure protocol where:

- Aborting will not help cheating Alice
- Bob cannot defame honest Alice
- Proof does not reveal Bob's input
- Very high efficiency (no public key operations)

Publicly verifiable covert (PVC) MPC [KM15]

Before	After
Nobody can cheat	Alice can cheat. Caught with prob ½. If caught, proof of cheating is published. Sufficient deterrent in most scenarios.
	~~20X~~ speed improvement ~30X, Free Hash [FGK17]

Free Hash [Fan Ganesh K17]

Idea [GMS08]: don't send circuits.
Instead:

1) choose seed s
2) generate GC(PRG(s))
3) compute ~~h=SHA(GC)~~
4) send h. A cannot later send a wrong GC

5) A send s to open circuits
6) A send GC to evaluate

Free Hash:
$h = \bigoplus$ {GC labels}

Free GC hash definition

- GC hash definition weaker than standard collision resistance
- Take advantage of the input to hash being a Garbled Circuit
- Given a correctly generated garbled circuit and hash (GC; h)
  - If A finds $\widehat{GC}$ such that $H(\widehat{GC}) = H(GC)$
  - Then, w.h.p, the garbled circuit property of $\widehat{GC}$ is broken
  - $\widehat{GC}$ will fail to evaluate
- Verification of hash involves GC evaluation

GC hash definition

GC, GC, e, e, d, h

C

Ve(C, GC, d, e) = accept

H(GC) = H(GC) = h

Same decoding information d

De(Eval(GC, En(e, x), d) = ⊥ for all x , w.h.p

Main idea for GC hash construction

- Garbled rows are encryptions of output labels
- Garbling of a gate relates garbled rows and input and output labels as preimage/image of a crypto function
- Change in a garbled row or input label creates unpredictable change in computed output label
- Hard to change *active* garbled rows and still get output label that you want
- During GC evaluation, once label is wrong, hard to make it right
- Idea: ensure all rows are active, i.e. GC evaluation involves *all* GC rows
  - *Not quite enough, but close. Not hard to work out precise requirements.

Thank you!