Communication Complexity, Streaming and Computational Assumptions

Shahar Cohen
Roey Magen
Boaz Menuhin

Moni Naor

Weizmann Institute of Science

Simons Workshop on Minimal Assumptions, May 3rd 2023

What Effect Do Crypto Assumptions have on Algorithms

Choose a setting where **randomness** helps

- Show a good algorithm against an **inactive/static** adversary
- Show what an **active/adaptive** adversary can do
- Discuss whether **crypto** can help
  - **And if it can help, show that the tools are essential**

Repeat

Minimal Assumptions

Can we **automate** the process?

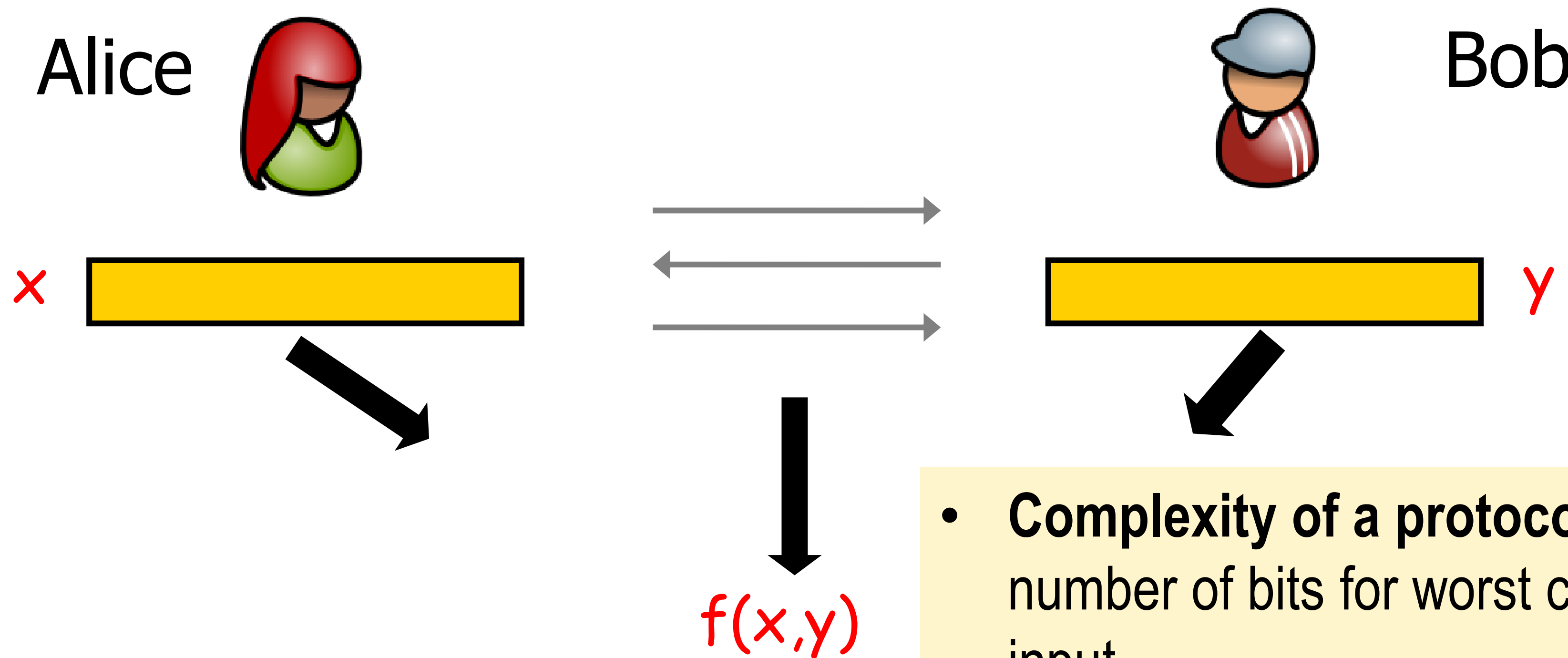
Other Examples

- **Sketching**, Mironov, Naor and Segev 2008
- **Error correction**, Lipton, Micali-Peikert-Sudan-Wilson, Grossman-Holmgren-Yogev
- **Communication vs. Computation,** Harsha, Ishai, Kilian, Nissim and Venkatesh
- **Lower Bound for Checking Correctness of Memories,** Naor and Rothblum 2005
- **Adversarially Robust Bloom Filters,** Naor-Yogev 2015
  - Bet-or-Pass TCC 2022 - Noa Oved
  - Defining the success of an Adversary with adaptive choices
- **Adversarially Robust Property Preserving Hash Functions,** Boyle, LaVigne and Vaikuntanathan

What Will We See (Time Permits…)

- **Communication Complexity**, Crypto 2022 –Shahar Cohen
  - Low Communication Complexity Protocols, Collision Resistant Hash Functions and Secret Key-Agreement Protocols
- Streaming (**card guessing**), ITCS 2022 - Boaz Menuhin
  - **Mirror Games**, FUN 2022 - Roey Magen
  - **WIP: Low Memory Permutation Generation**

Communication Complexity



- **Complexity of a protocol**: number of bits for worst case input
- **Complexity of a function**: complexity of best protocol

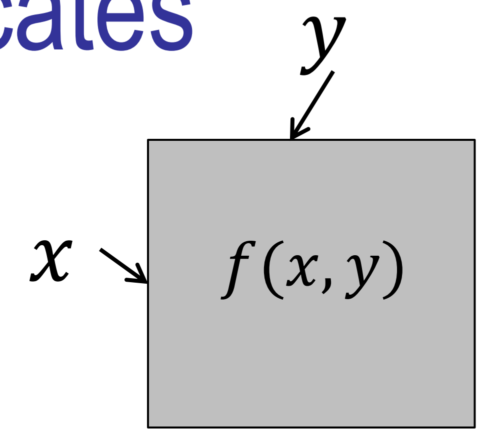
Let $f: X \times Y \mapsto Z$

Input is split between two participants

Want to compute: $z=f(x,y)$

while exchanging as few bits as possible

Equality and Other Predicates



- Our canonical example – **equality**.
  - $f(x,y) = 1$ iff $x = y$
- A non-trivial predicate: with no redundant rows and columns
  - No two rows or two columns are **identical**

Efficiently Separable Predicate:

- There is an efficient algorithm that given

$$x_1, x_2 \in X$$

finds y s.t. $f(x_1, y) \neq f(x_2, y)$

Communication Complexity Protocol Variants

Deterministic complexity is often n
- Example: equality

Protocols differ by
- Network layout
  - Who talk to who and number of rounds
    - Interactive Model
    - Simultaneous Message Model
- Use of Randomness
  - **Shared** public randomness
    - Independent of the inputs
  - Private Randomness

Newman: largest possible gap

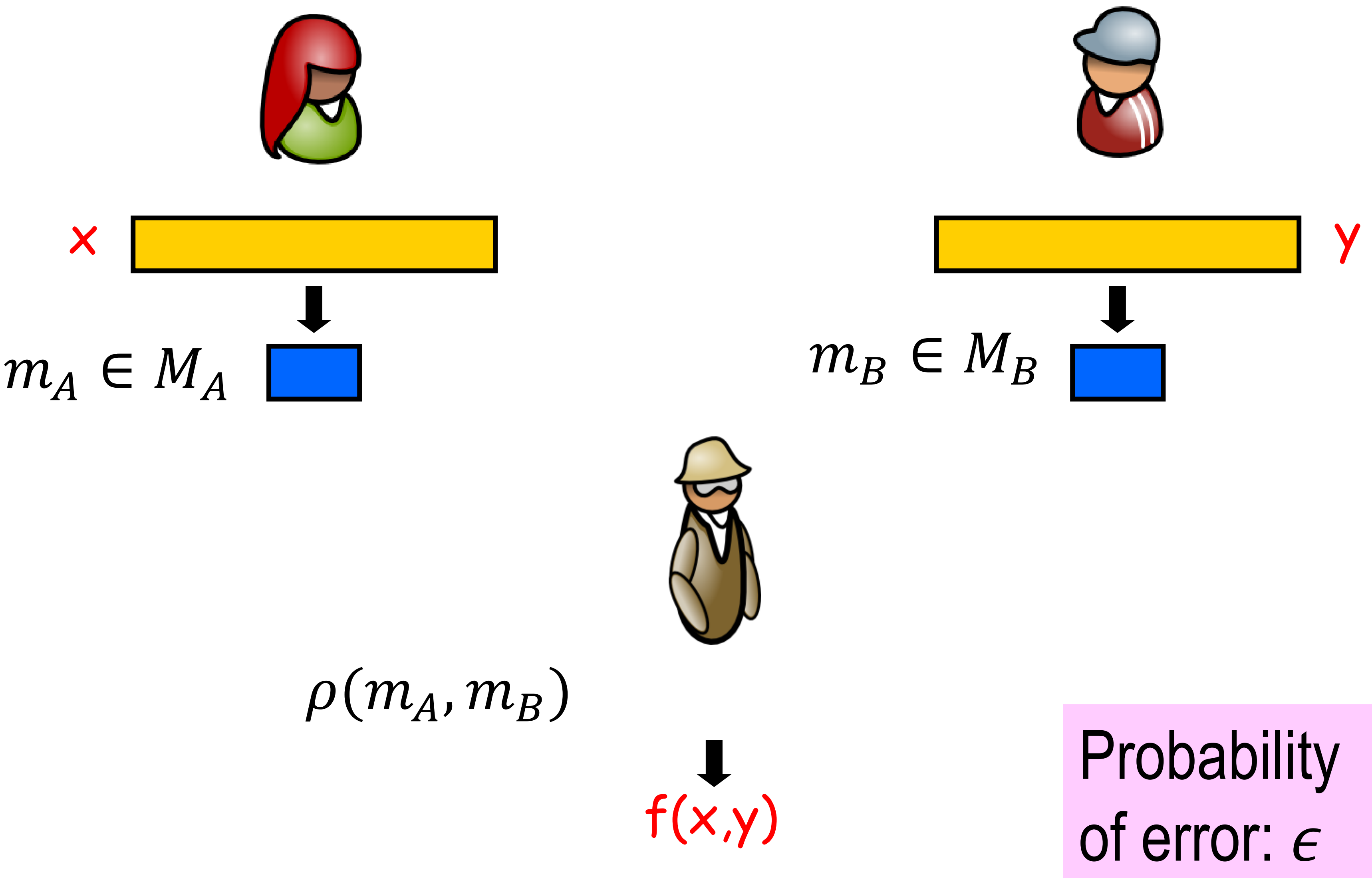
•Orthogonal!

Equality function Interactive
- Shared Randomness $O(1)$
- Private Randomness $\Theta(\log n)$

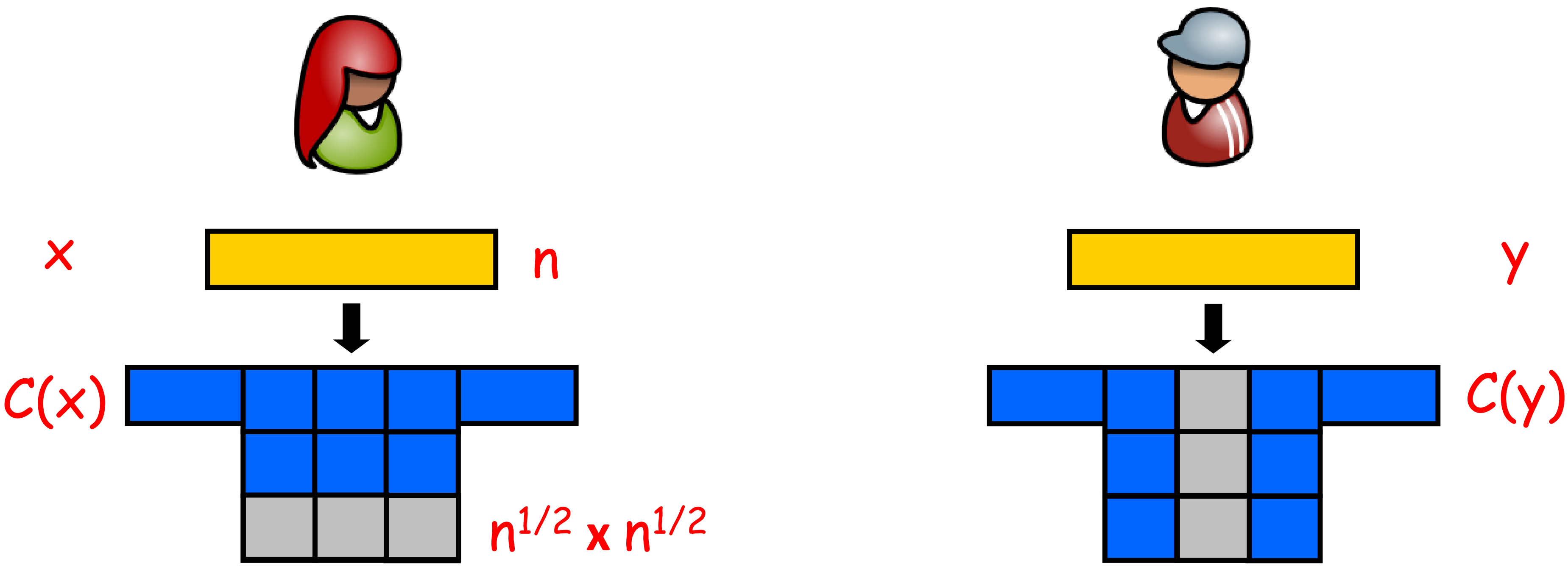
No function is $o(\log n)$ with private randomness

First proof: Ben-Sasson-Maor

Simultaneous Messages Model



Simultaneous Equality Testing



C should be a good error correcting code

Communication $O(n^{1/2})$

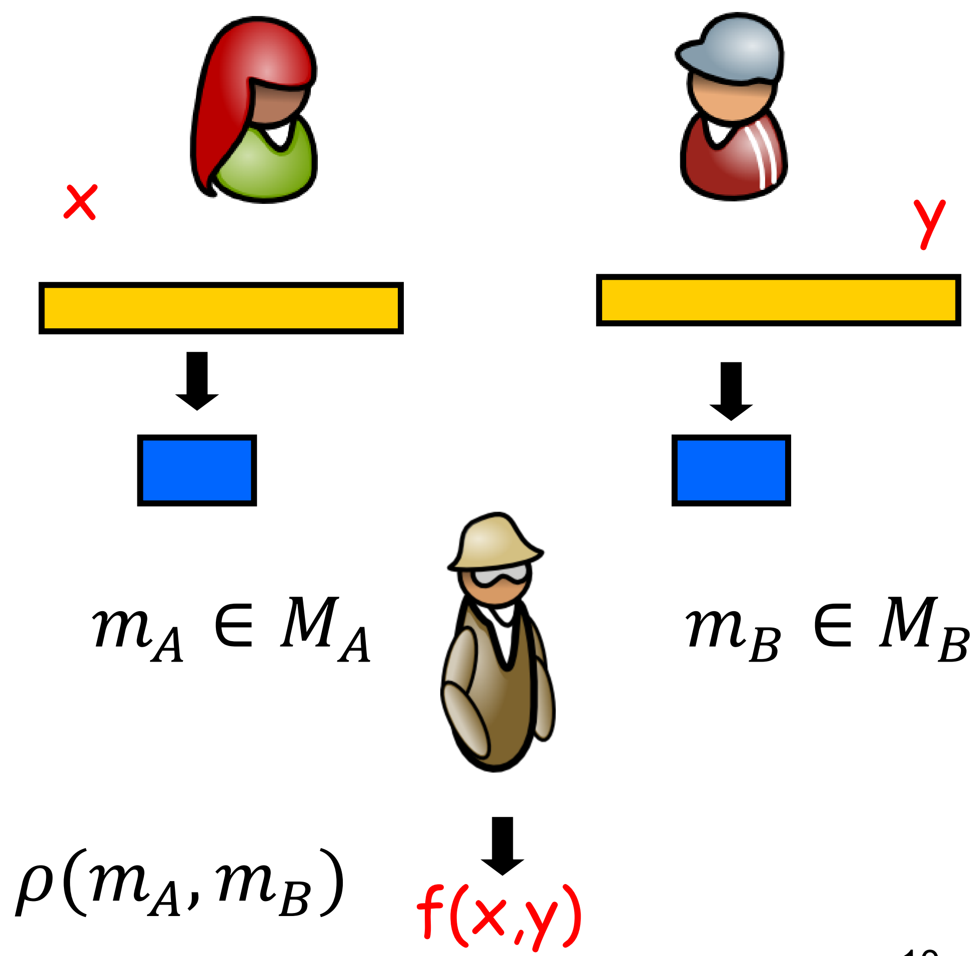
Simultaneous Messages Model Lower Bound

- Newman-Segedy 96
  $|m_A| + |m_B| = \sqrt{n}$
- Babai-Kimmel 97
  $|m_A| \cdot |m_B| = n$

In general:

Deterministic complexity

- Bottesch, Gavinsky, and Klauck 2015



Central Question

- Can we reduce communication complexity by **assuming certain hardness assumptions**
  - What assumptions do we need?
- What changes to the model do we need to make?

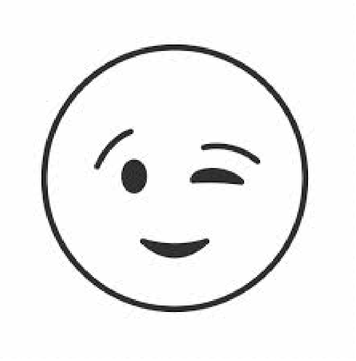
- **When** is the randomness chosen
- **Who** maintains state
- The **exact power** of the adversary

Models
•Preset Randomness
•Free talk stateful

Results

Almost Tight bounds on communication complexity, assumptions and models

When you close one eye



Results: preset randomness

- Breaking the $\sqrt{n}$ lower bound for equality in the **simultaneous message** model implies the existence of **distributional Collision Resistant Hash** (dCRH) functions in a constructive manner
- Dito for the $\log n$ bound in **interactive communication**
- There are **no protocols** of constant communication

Techniques employ the Babai-Kimmel Proof

- Assuming existence of CRH: can break the bounds

Collision Resistance Hash

Results: stateful ``free talk"

- Parties Alice and Bob talk freely **before the inputs are chosen by adversary**
  - May maintain secret states τ_A and τ_B ***respectively***
  - The communication is measured only after the preprocessing

Very efficient protocols for equality against a **rushing** adversary imply the existence of **secret-key agreement protocols**

- Assuming that for a c bit protocol the probability of error is at most $2^{-0.7c}$

Assuming SKA exist: there is a c bit protocol with error probability 2^{-c}

Assumptions in cryptography

Minicrypt

Oracle Separation

- One-way functions
  - Existentially equivalent to a whole host applications such a private key encryption
- Collision resistance Hash Function
- Secret-key Agreement.
  - Implied by Public-key encryption

- Separating OWFs from CRHs: consider a collision finder: Given a **collision finder**, OWFs do exist but CRHs do not exist
- Separating SKAs from CRHs: In the random oracle model CRHs do exist but SKAs do not exist

Collision Resistance Hash Functions

CRH

A family of hash functions H where it is **hard to find any collision**

- All functions $h \in H$ are compressing
- Efficiently computable
  - Given $h \in H$ and x

easy to evaluate $h(x)$

- Hard to find collisions: for every PPT Adv, and large enough λ, for a random $h \in_R H$

Probability $Adv(h)$ finds $x \neq x'$ s.t. $h(x) = h(x')$ is negligible in security parameter λ

Simon 98….:
- Black box separation from one-way functions

Random Collision finder

If can compress by a little –
Can compress by a lot

Distributional Collision Resistance Hash

dCRH

Dubrov and Ishai 06. Bitansky, Haitner, Komargodski and Yogev 19

Constant-round statistically hiding commitment schemes

A family of hash functions H where it is hard to find a **random** collision

Simon 98….:
- Black box separation from one-way functions

Random Collision finder

Random Collision finder **COL**

- **COL** gets $h \in H$ and outputs (x, x') s.t. x is uniformly random and x' is uniformly random from $h^{-1}(x)$
- H is a family of **distributional CRHs** if there exists poly $p(\cdot)$ s.t. for every PPT Adv, and large enough λ, for a random $h \in_R H$
$$\Delta(COL(h), Adv(h)) \geq 1/p(\lambda).$$

CRHs imply succinct protocols

Theorem: If CRHs exist, then given a family of CRHs

$$\{h\colon \{0,1\}^n \rightarrow \{0,1\}^\lambda\}$$

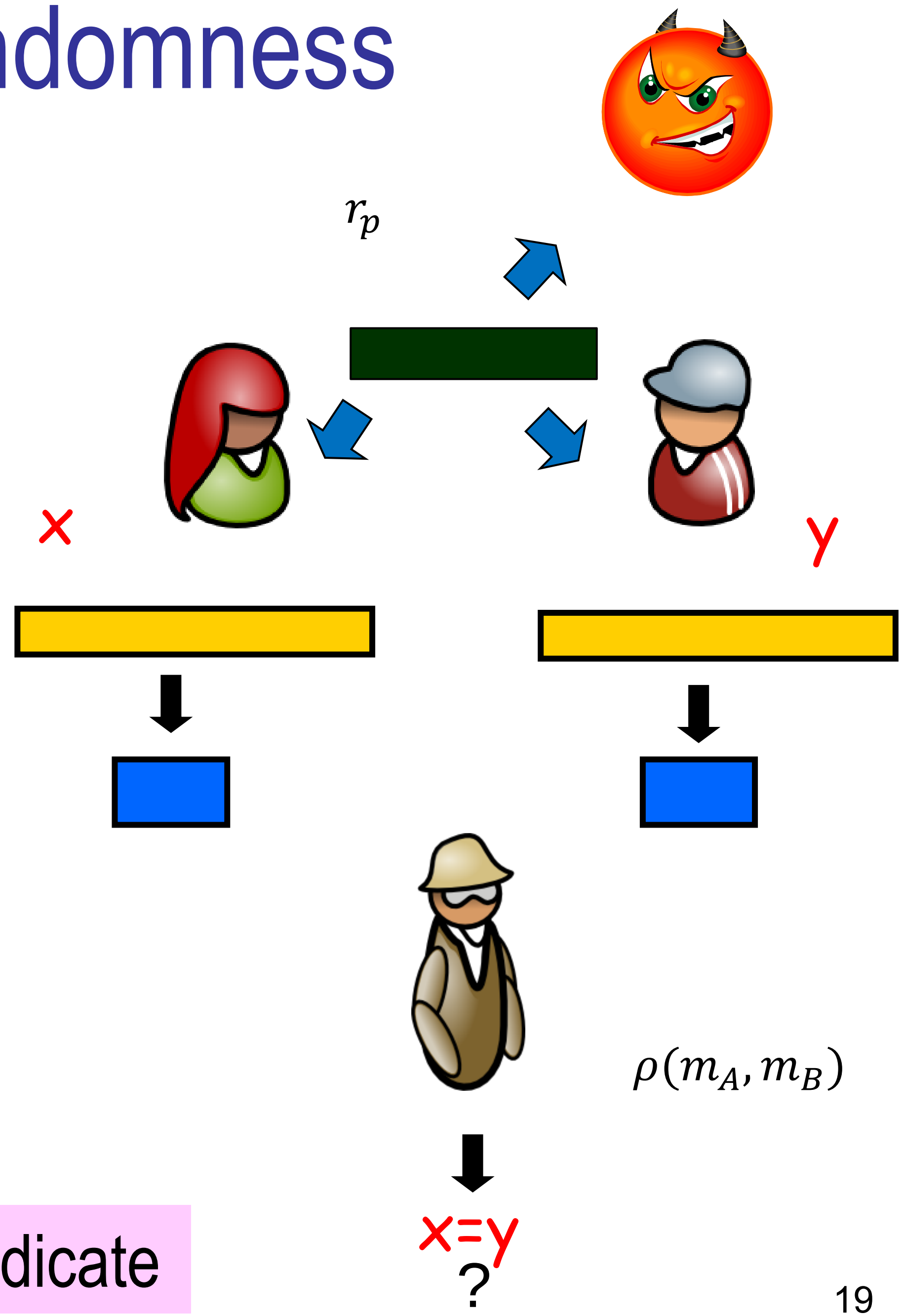
- In the preset public coins SM model: there is a protocol of complexity $O(\sqrt{\lambda})$ for the Equality predicate.
- In the preset public coins interactive model: there is a protocol of complexity $O(\log \lambda)$ for the Equality predicate.

- Public string: the hash function h
- Replace x with $h(x)$

Preset randomness

Need to show how to construct from a **succinct** protocol a **hash function**

- Inputs are chosen by the adversary depending on the public random string
- Idea: use a **characterizing multi-set** of responses as a hash function



Works for every non redundant predicate

SM Protocol Π for Equality

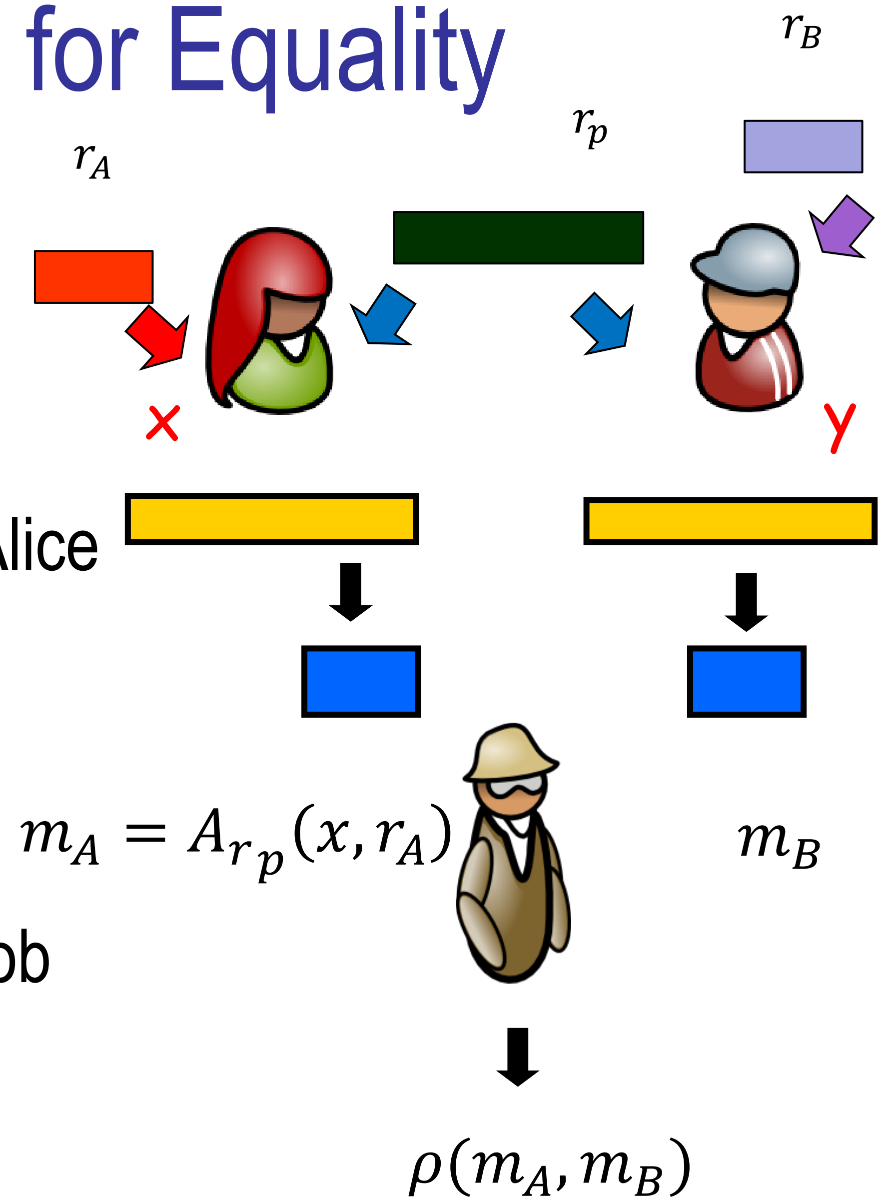
- Preset Public random string r_p
- Input space for X and Y
- Alice gets $x \in X$ and Bob $y \in Y$
- M_A and M_B message space for Alice and Bob
- Private randomness:

$$r_A \in R_A \; and \; r_B \in R_B$$

  - Random tapes for Alice and Bob
- Message Alice sends:

$$m_A = A_{r_p}(x, r_A) \in M_A$$

- Referee's Decision $\rho(m_A, m_B)$



Characterizing Multisets

input of Alice

- For every $x \in X$ there exists a multiset **characterizing** the behavior of Alice on x.
  - Instead of running Alice, can approximate the protocol's result (referee's output) by a uniform sample from the multiset.
  - Such a multiset can be found (w.h.p.) by relatively few independent samples from the distribution defined by Alice on x and r_p.

Characterizing Multisets

input of Alice

For public string r_P and input $x \in X$ a multiset of messages $T_x \subset M_A$ **characterizes** $\boldsymbol{x}$

- if $\forall\, m_B \in M_B$,

$$|Q(T_x, m_B) - Prob\left[\rho\left(A_{\mathrm{r_p}}(x, r_A), m_B\right) = 1\right]| \leq 0.1$$

over r_A

- where $Q(T_x, m_B)$ is the referee's **expected value** for the multiset T_x and Bob's message m_B.

Sampling yields characterizing multisets

Theorem:

- For any public string r_p and for and $x \in X$
- Let $r' = (r_A^1, \ldots, r_A^t)$ be t independent uniform samples from R_A where $t = \Theta(\log |M_B|)$.
- Then, for the multiset $T_x = \{A_{\mathrm{r_p}}(x, r_A^i) : i \in [t]\}$ it holds that $\boldsymbol{T_x}$ **characterizes Alice for** $\boldsymbol{x}$ with constant probability

Constructing Hash Functions From Characterizing Multisets

The function h is defined by

- The public random string r_p and
- t random tapes for Alice $r_A^1, \dots, r_A^t \in R_A$.

Output: For x ∈ X, the value of the function is the multiset

$$h(x) = \{A_{r_p}(x, r_A^i : i \in [t]\}$$

where the multiset is encoded as a sequence

$$A_{r_p}(x, r_A^1), \dots, A_{r_p}(x, r_A^t)$$

- Every message of Alice encoded using $\log|M_A| = c$ bits

The constructed function is good

- The function h is compressing

Should be characterizing to both

- Any x and x' which **share a characterizing multiset**, induce **bad inputs** for the protocol:
  Let $x, x' \in X$ and $y \in Y$ that separates them.
  If there is a multiset T that is characterizing for both x and x', then
  - the sum of the failure probability of $\pi(x, y)$ and $\pi(x', y)$ is at least 0.8.
  - At least one of them fails.

From $Adv_{collision}$ breaking h as a dCRH to Adv_{π} breaking Π

- Given an efficient adversary $Adv_{collision}$ that breaks the security of h as a **distributional CRH** for some $p \in poly(\lambda)$:

$$\Delta(Adv_{collision}(h), COL(h)) \leq \ 1/p(\lambda)$$

- Then, we can construct an adversary Adv_{π}
  - with running time of the same order as $Adv_{collision}$

that succeeds in making Π fail with probability 0.4(1-$1/p(\lambda)$)

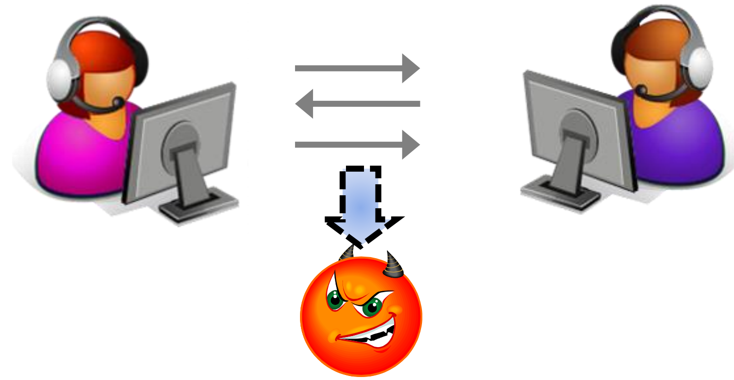
Using Collision Finder for h to Find Bad Inputs for Protocol Π

- Construct $h(x)$ using the public random string of π
- $x, x' \leftarrow Adv_{collision}(h)$.
- Find $y \in Y$ which separates x and x'
- Set Bob's input to be y and Alice input to be
  - x w.p. ½ or
  - x' w.p. ½.

Why dCRH and not CRH?
- **Not all are characterizing**

Characterize the properties of h

Stateful Free Talk



- Alice and Bob talk freely

before **the inputs are chosen by adversary**

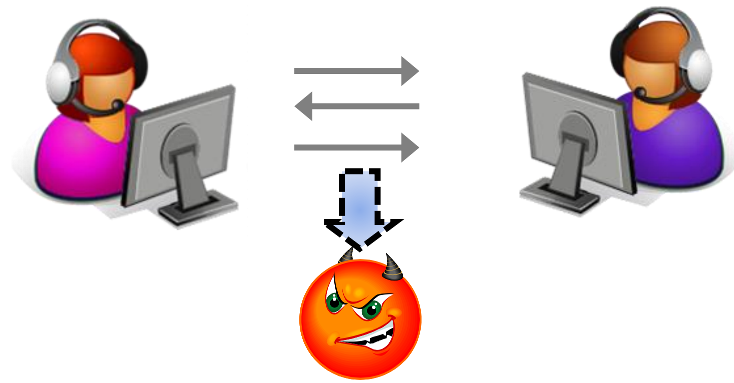
  - Maintain a secret state τ_A and τ_B
  - Adversary eavesdrops to the free talk phase and then selects inputs
- Communication is measured only **after** the free talk preprocessing phase
  - Mostly interested in SM pattern

Free Talk: Rushing Adversary

computationally bounded

- The inputs are chosen by an adversary, depending on the public discussion it witnesses in preprocessing phase.
- A **rushing adversary** can choose Bob's input at the `last moment’:
  - The adversary first chooses the input x of Alice *depending on the public random string*
  - **After** Alice sends her message m_A to the referee, the adversary chooses the input y of Bob
    - Depending on **both** the preprocessing transcript and on m_A
- **Patient adversary**: there are multiple sessions between Alice and Bob and the adversary can choose **one session to attack among them**, after seeing the message Alice sends.

Secret-Key Agreement



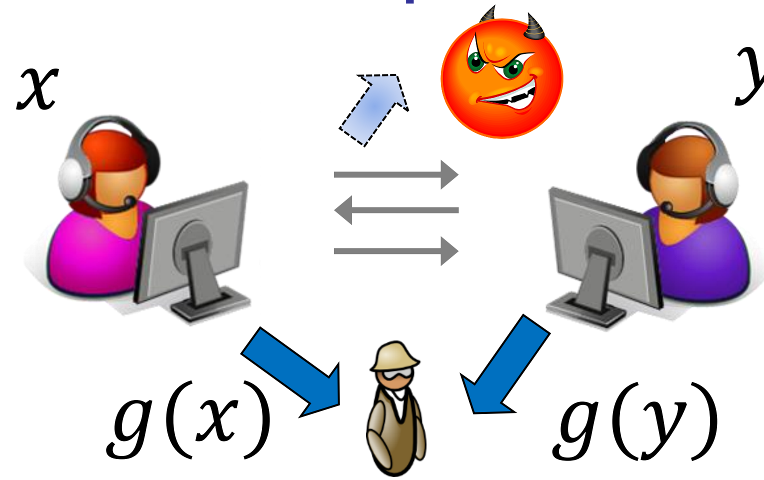
Secret key agreement (SKA)

- A protocol where two parties with **no prior common information** agree on a secret key.
- The key should be secret
  - No PPT adversary, given the transcript of the communication between Alice and Bob, can compute the key with non-negligible advantage
  - Public-key encryption implies SKA

Can “distinguish it from random”

SKA implies succinct protocol with optimal error

Execute an SKA
    Secret state is the key
Given the input use the **key** as a **pairwise ind**. hash function $g \in G$
    Send $g(x)$



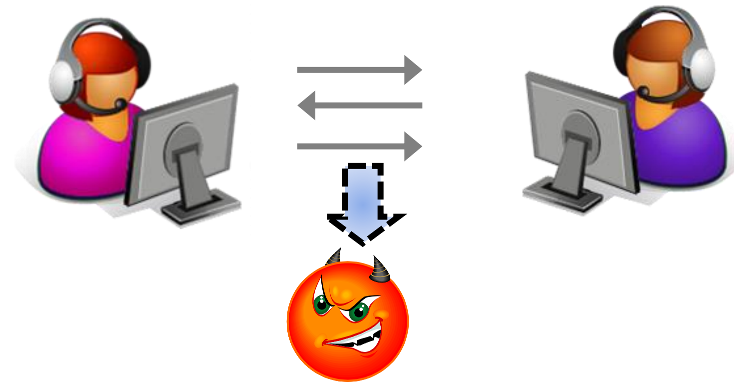
Theorem: Given a secret key agreement protocol there is in the

- Stateful preset public coins
- SM with free talk model:
- For any c(n),

a protocol for equality of complexity c(n), **where any adversary can cause an incorrect answer** with prob. at most $2^{-c} + negl(n)$

- **Even a rushing one**
- **Even a patient one**

Secret-Bit Agreement - Quantification



(α, β)-Secret bit agreement (SBA)

- The secret is one bit.
  - The two parties output b and b'.
- With probability at least (1+α)/2

$$b = b'$$

For α and β which are

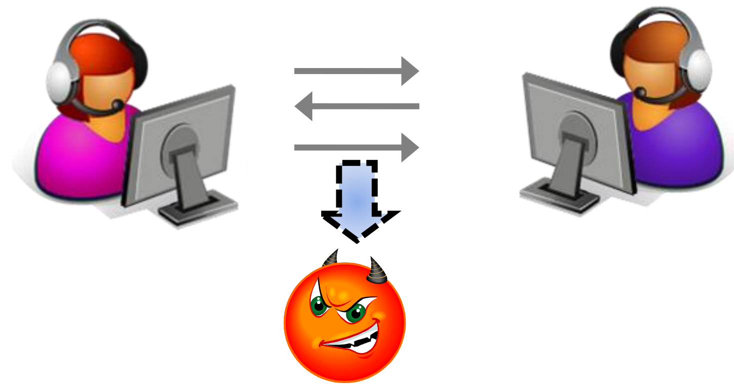
$$\mathbf{1} - \boldsymbol{negl}(\boldsymbol{\lambda})$$

we get SKA

- **Secrecy**: no PPT Adv which gets as input the transcript guesses the agreed bit given $b = b'$ with probability great than $1 - \frac{\beta}{2}$

$$Prob[Adv(\tau) = b | b = b'] \leq 1 - \frac{\beta}{2}$$

Secret-Key Agreement: Amplification



Holenstein 2006

Given an (α, β)-Secret bit agreement (SBA) where

$$\frac{1-\alpha}{1+\alpha} \leq \beta$$

- Can construct a computationally secure SKA
  - where α' **and** β' are $1 - negl(\lambda)$
- The time is $poly(\lambda)$

Succinct stateful free talk implies SKA

- An SM protocol with stateful free talk for equality of complexity $c(n) \in O(\log \log n)$ that is
  - **ε-secure** with $\varepsilon \leq 2^{-0.7c(n)}$
  - Immune to rushing and patient adversaries

  implies the existence of **secret key-agreement** protocols.

- The protocol should be *nearly* optimal in error

Protocol Π for Equality

Structure of Protocol Π :

- Alice and Bob communicate and generate secrets states
  - τ_A for Alice
  - τ_B for Bob
- On inputs x and y respectively
  - Alice sends $m_A = A(x, \tau_A)$
  - Bob sends $m_B = A(y, \tau_B)$
- Result is $\rho(m_A, m_B)$

Weak Bit Agreement from Protocol Π for Equality

- Alice and Bob communicate and toss coins according to the **free talk** phase of protocol π
  - to generate their secret states τ_A and τ_B.
- Alice selects at random a bit $b \in_R \{0,1\}$ and uniformly random inputs $x_0, x_1 \in_R \{0,1\}^n$.
- Alice evaluates $m_A = A(x_b, \tau_A)$
  - A message of the protocol Π for EQ(·, ·).
- Alice sends to Bob the pair (m_A, x_1).
- Bob evaluates $m_B = B(x_1, \tau_B)$.
- Alice outputs b and Bob outputs $b' = \rho(m_A, m_B)$

Referee's response

The SBA protocol is sufficiently good

Theorem:

The Algorithm is an ($\alpha = 1 - 2^{-\frac{c}{2}}, \beta = 2^{-\frac{c}{2}}$)-SBA protocol.

Agreement:

By the fact that the error $\epsilon \le 2^{-0.7c}$

$$\Pr[b = b'] \ \ge \ 1 - 2^{-0.7c}$$

Secrecy: construct an adversary $\boldsymbol{Adv_{eq}}$ from adversary $\boldsymbol{Adv_{sba}}$ breaking the SBA with above parameters

ADV_{Eq} from ADV_{SBA}

Algorithm for Finding Bad Inputs Using $\boldsymbol{Adv_{sba}}$

Repeat at most $6 \cdot 2^{c+1}$ times:

- Select uniformly at random $x \in \{0,1\}^n$ and set it as Alice's input.
  - Let Alice's message be $m_A \in M_A$.
- Select uniformly at random $x' \in \{0,1\}^n$.
- If $Adv_{sba}(x, m_A) = 1$ **and** $Adv_{sba}(x', m_A) = 1$:
  - Pass m_A to the referee and set Bob's input to
    - $y = x$ w.p. ½ or
    - $y = x'$ w.p. ½.
  - Otherwise, continue to the next session

Does not distinguish x and x'

Analysis of Algorithm

Guessing b when it is equal to b'

Given $\boldsymbol{Adv_{sba}}$ with success probability at least $\frac{2^{c/2}-1}{2^{c/2}}$,

we can construct an adversary $\boldsymbol{Adv_{eq}}$ with running time O(2^{c+1})

s.t.

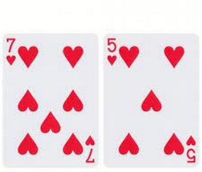
$$\text{Prob}\left[\Pi \text{ fails on inputs chosen by } \boldsymbol{Adv_{eq}}\right] > 2^{-0.7c} \geq \epsilon.$$

Further Research

- Are CRHs equivalent to preset public coins SM protocols of complexity $o(\sqrt{n})$
  - Can we break that bound using a primitive weaker than CRHs. What property do the functions we construct satisfy?
- Multi CRHs (MCRH): For $k \geq 3$, finding a k-collision of size is hard
  - Construct MCRHs from succinct protocols in a black-box manner?
- Free-talk to SKA
  - What about protocols with much worse error probability
    - Constant error probability for c which O(log log λ)
  - Do we need a rushing adversary?
- What about Rushing in the preset model? Do sublinear protocols imply (d)CRH?

Hard to Guess Permutations

- Card Guessing with Limited Memory [Menuhin Naor]
  - The Power of Adaptive Adversaries in Streams



- Mirror Games
  - Garg Schneider
  - Feige
  - Magen Naor

- WIP: Low memory generation of hard to guess permutations.