Secure Multi-Party Computation & Cryptographic Complexity

Manoj Prabhakaran
IIT Bombay

MPC: Emulating Trusted Computation

- Encryption/Authentication allow us to emulate a trusted channel

- Secure Multi-Party Computation (MPC): to emulate a source of trusted computation

  - Trusted means it will not "leak" a party's information to others

    - And it will not cheat in the computation

  - Emulate: there is no trusted party!

Cryptographic Complexity

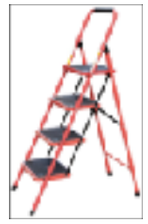
- How hard is to securely compute a (multi-party, finite) function?
  - cf., Computability, Computational Complexity, Communication Complexity
- Lowest level of complexity: "Trivial" functions
  - Those which can be securely computed
  - cf. decidable languages

MPC Dimensions

Time Model

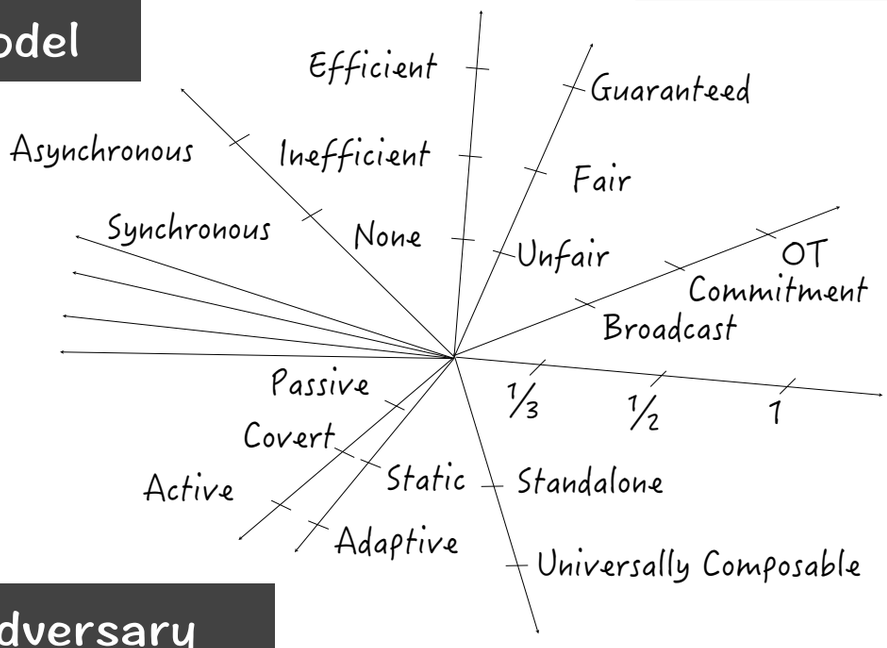
Simulation

Output delivery



Set-up

Complexity Parameters



Corruption Threshold

Adversary

Composition

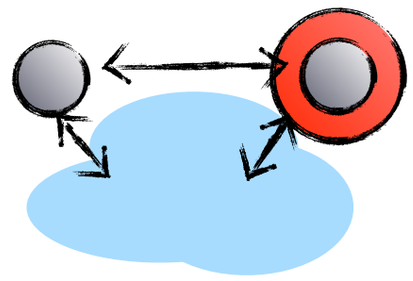
Plan

- Lowest level of complexity for 2-party functions
  - Passive, Active Standalone, UC security
    (only information-theoretic security today)
- Defining higher levels of complexity
- Highest level for 2-party functions
- Looking briefly into intermediate levels
- Some open problems
- Part 2: Quantitative Cryptographic Complexity

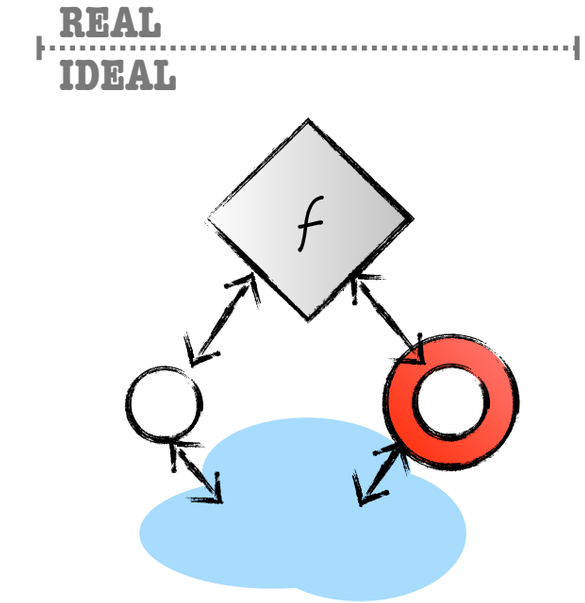
Passive Security

- Information-theoretic condition for security (inputs X,Y; desired outputs A, B). For <u>all</u> distributions of X, Y:
  - $(Output_{Alice}, Output_{Bob} \mid X, Y) \equiv (A, B \mid X, Y)$
  - $View_{Alice} \longleftrightarrow X,A \longleftrightarrow Y,B$

    $X,A \longleftrightarrow Y,B \longleftrightarrow View_{Bob}$
- Statistical security: Allow "negligible" error in these conditions
  - Negligible: as a function of <u>security parameter</u> κ. Think $2^{-\kappa}$.



$P_{X,A,Y,B,View(Bob)}$



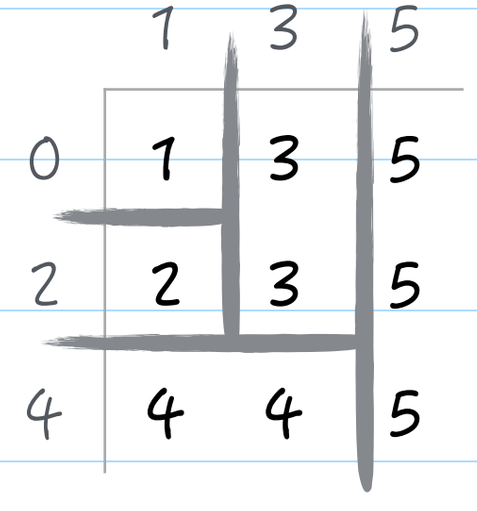
$P_{X,A,Y,B} \times P_{View(Bob) \mid Y,B}$

Quiz

- What's the complexity of the following 3 functions (defined over say [0,100] × [0,100]), w.r.t, passive secure MPC?
  - max(x,y)
  - [x < y]
  - (max(x,y), [x < y])

Passive Trivial Functions

- e.g., max(x,y), where x even, y odd



  - Dutch flower auction

- Iteratively "decompose" the domain to zoom into a monochromatic rectangle

- Only functions with decompositions are passive trivial

[Kus.'89, Bea.'89, MPR'09, KMR'09]

Frontier Analysis

- Only functions with decompositions are passive trivial

  [Kus.'89, Bea.'89, MPR'09, KMR'09]

- Suppose f is not decomposable.

  - Then it has a sub-function g which is not decomposable at the top-level.

  - If f passive trivial, so is g

  - So, enough to prove that g not passive trivial

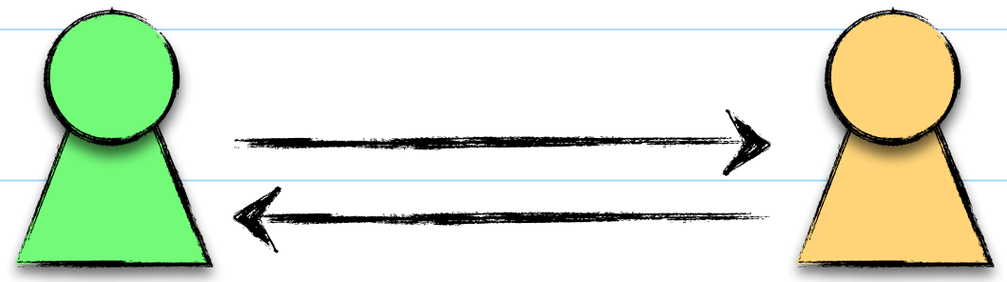
  - Suppose g has a passive secure protocol

Frontier Analysis

- Only functions with decompositions are passive trivial

  [Kus.'89, Bea.'89, MPR'09, KMR'09]

- Protocol



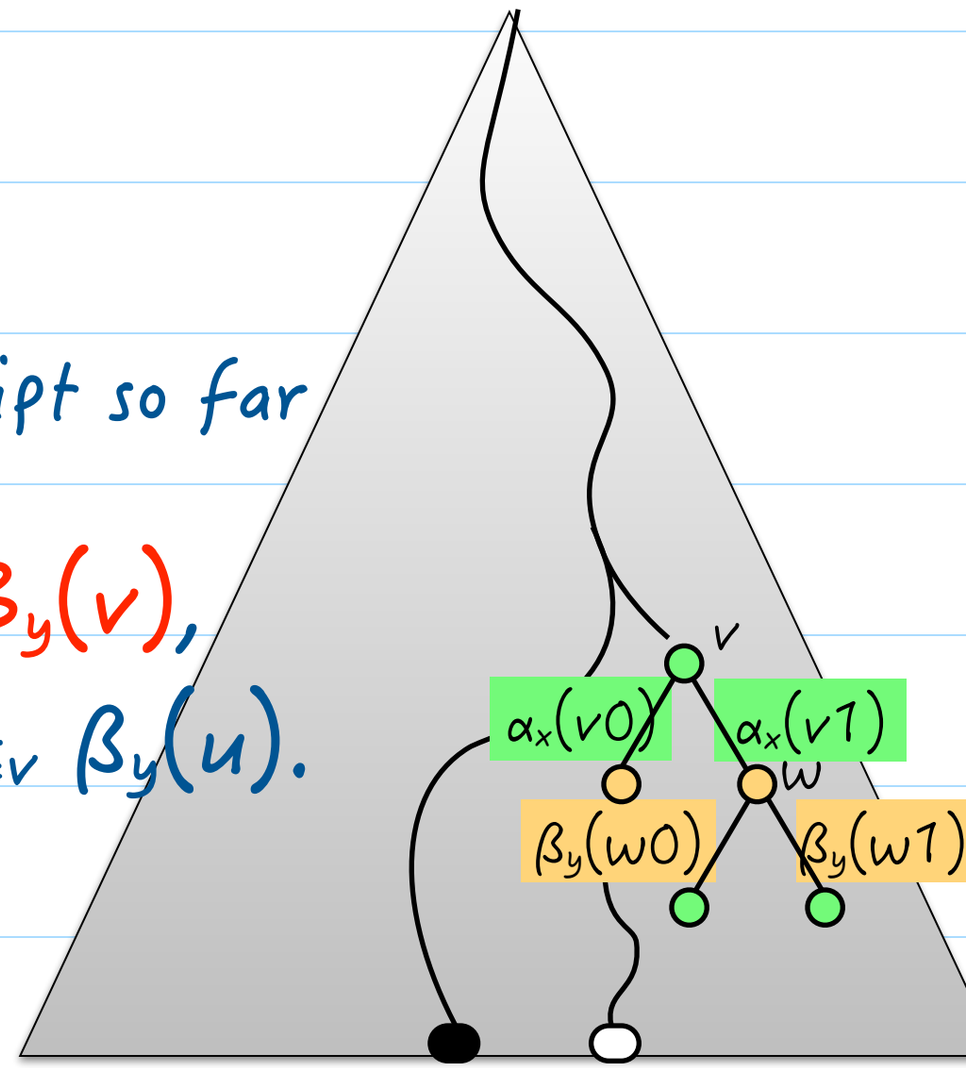
- Normal form:

  Repeat: { Exchange a bit each. }

  Output: part of transcript

  Stateless: Next bit based on input & transcript so far

- For any node v $Pr[v \mid x,y] = A_x(v) \cdot B_y(v)$,
  where $A_x(v) = \Pi_{u \leq v} \alpha_x(u)$, $B_y(v) = \Pi_{u \leq v} \beta_y(u)$.



Frontier Analysis

- Only functions with decompositions are passive trivial

  [Kus.'89, Bea.'89, MPR'09, KMR'09]

- For any node v $Pr[v \mid x,y] = A_x(v) \cdot B_y(v)$,

  where $A_x(v) = \Pi_{u \le v}\, \alpha_x(u)$, $B_y(v) = \Pi_{u \le v}\, \beta_y(u)$.

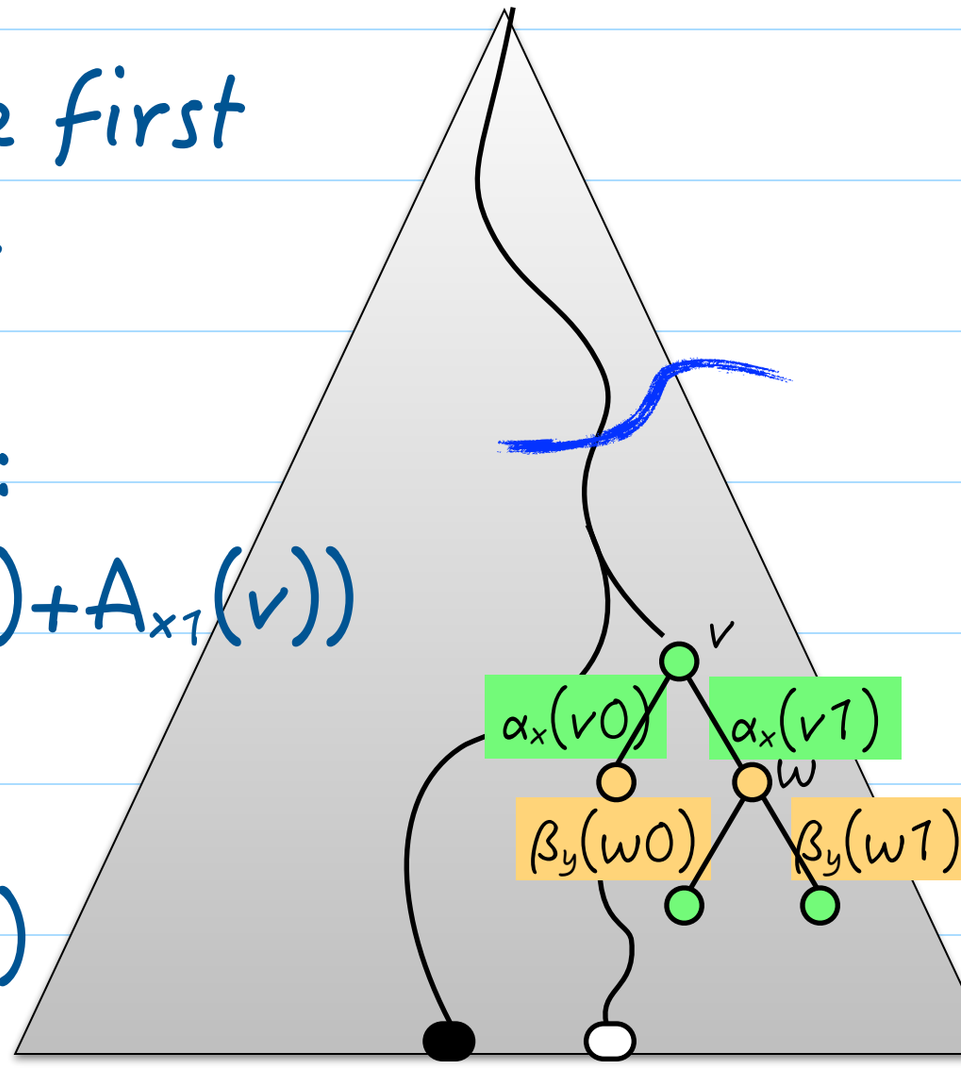
- Consider the <u>frontier</u> F where Alice first reveals information about her input

- <u>Distinction between x_0, x_1 at node v:</u>

  $D_A(x_0, x_1 \mid v) = |A_{x_0}(v) - A_{x_1}(v)| \,/\, (A_{x_0}(v) + A_{x_1}(v))$

- F = set of nodes v where $\exists x_0, x_1$

  s.t. $D_A(x_0, x_1 \mid v) > \varepsilon$ (for a suitable $\varepsilon > 0$)



Frontier Analysis

- Only functions with decompositions are passive trivial

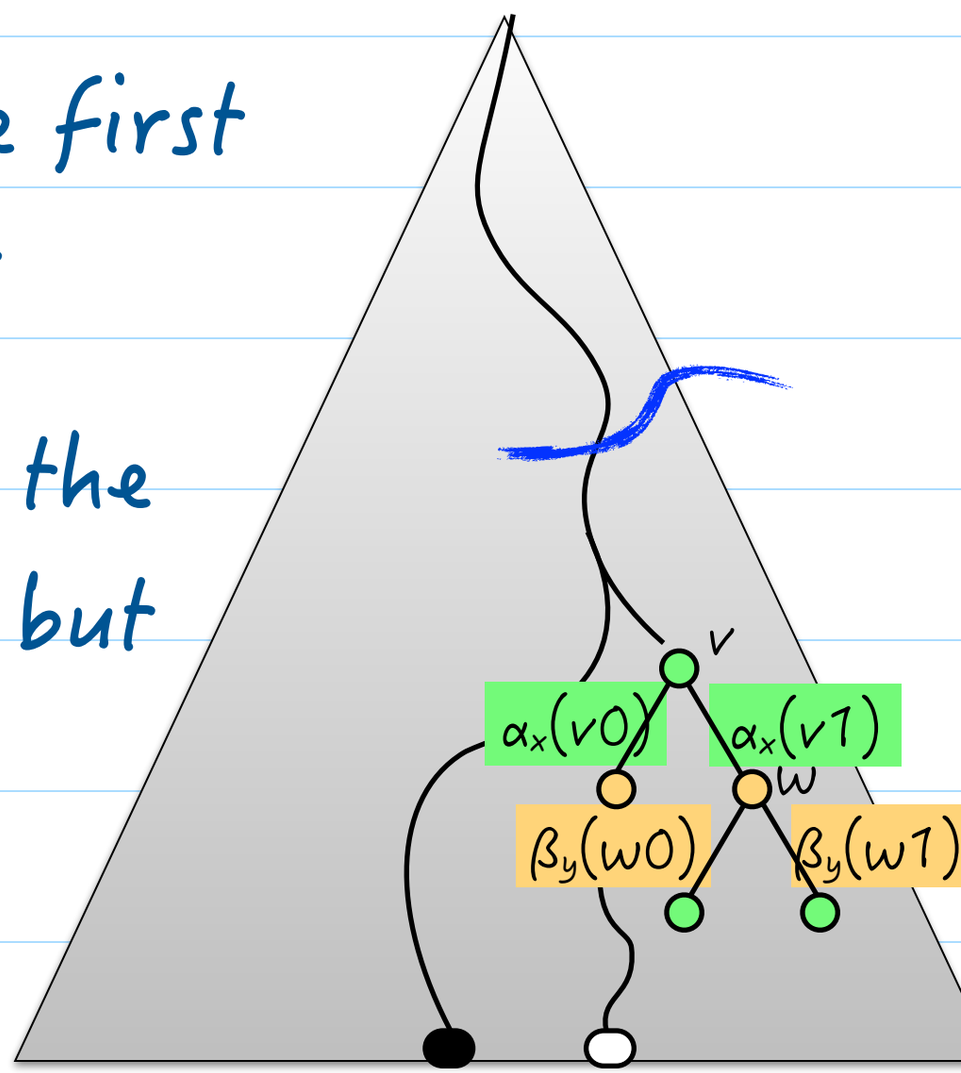
  [Kus.'89, Bea.'89, MPR'09, KMR'09]

- For any node v $Pr[v \mid x,y] = A_x(v) \cdot B_y(v)$,

  where $A_x(v) = \Pi_{u \le v}\, \alpha_x(u)$, $B_y(v) = \Pi_{u \le v}\, \beta_y(u)$.

- Consider the <u>frontier</u> F where Alice first reveals information about her input

- F has significant weight because at the leaves some x's are differentiated, but at root none are



Frontier Analysis

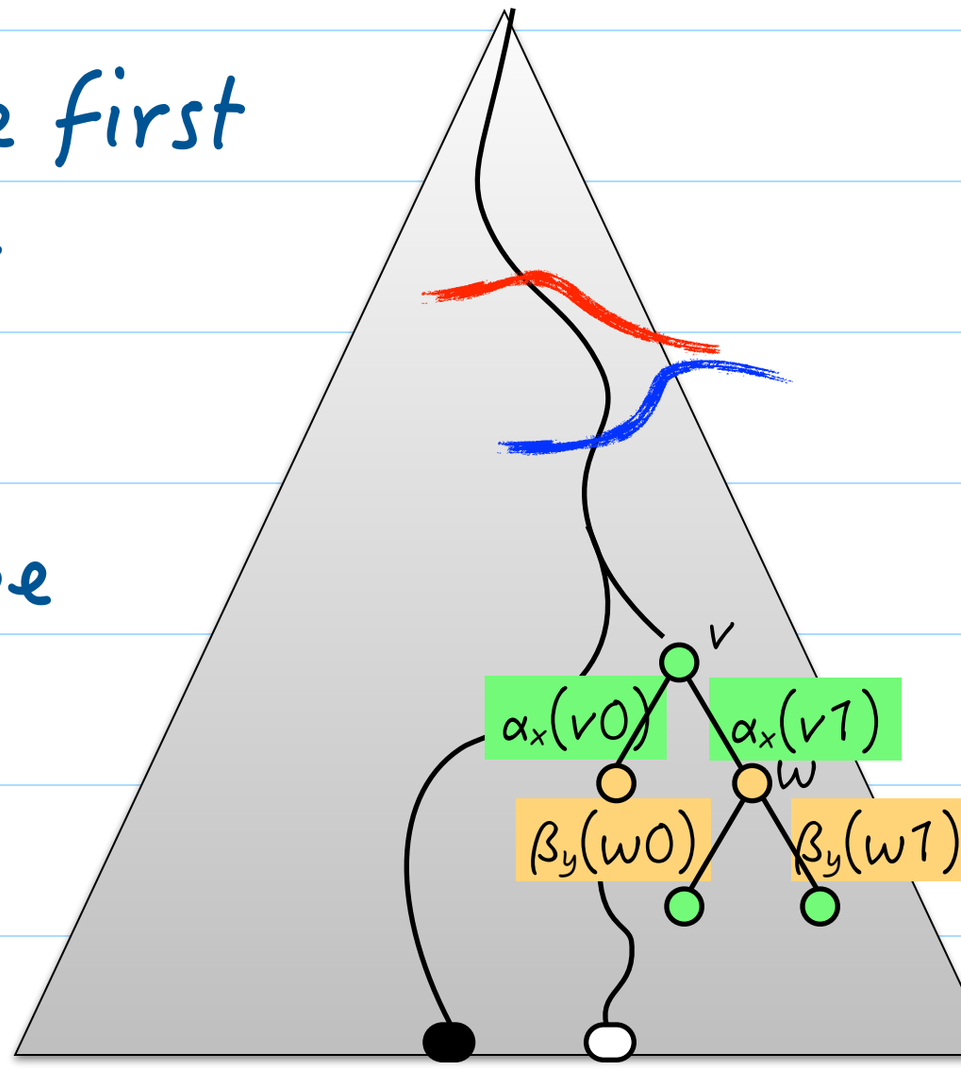
- Only functions with decompositions are passive trivial

  [Kus.'89, Bea.'89, MPR'09, KMR'09]

- For any node v $Pr[v \mid x,y] = A_x(v) \cdot B_y(v)$,

  where $A_x(v) = \Pi_{u \le v}\, \alpha_x(u)$, $B_y(v) = \Pi_{u \le v}\, \beta_y(u)$.

- Consider the <u>frontier</u> F where Alice first reveals information about her input
- Now, $\exists y$ s.t. $g(x_0,y) = g(x_1,y)$
  - Probability of reaching F should be negligible if Bob's input = y
- So Bob must have revealed information (strictly) above F



Frontier Analysis

- Only functions with decompositions are passive trivial

  [Kus.'89, Bea.'89, MPR'09, KMR'09]

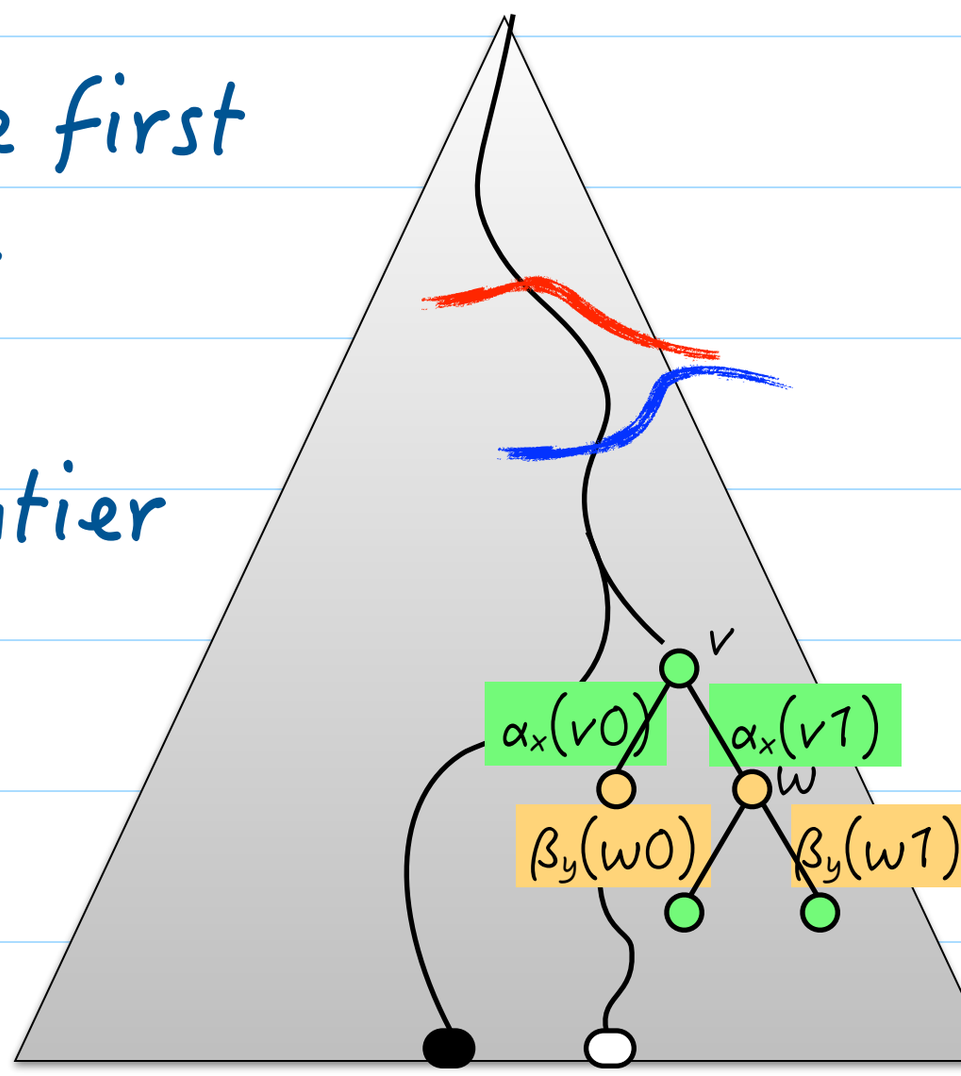
- For any node v $Pr[v \mid x,y] = A_x(v) \cdot B_y(v)$,

  where $A_x(v) = \Pi_{u \le v}\, \alpha_x(u)$, $B_y(v) = \Pi_{u \le v}\, \beta_y(u)$.

- Consider the <u>frontier</u> F where Alice first reveals information about her input

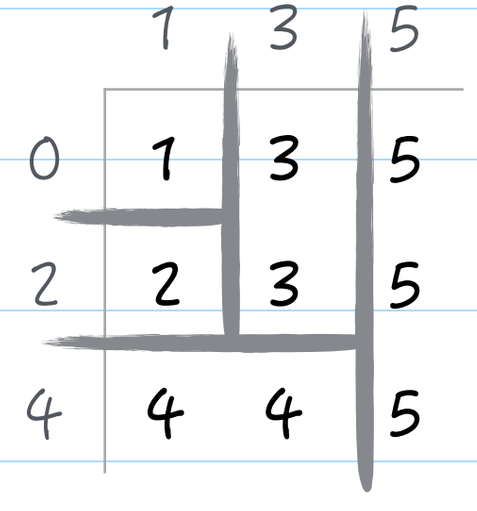
- F must be strictly below Bob's frontier

- Contradiction by repeating the argument for Bob's frontier



Passive Trivial Functions

- e.g., max(x,y), where x even, y odd



  - Dutch flower auction

- Iteratively "decompose" the domain to zoom into a monochromatic rectangle

- Only functions with decompositions are passive trivial

[Kus.'89, Bea.'89, MPR'09, KMR'09]

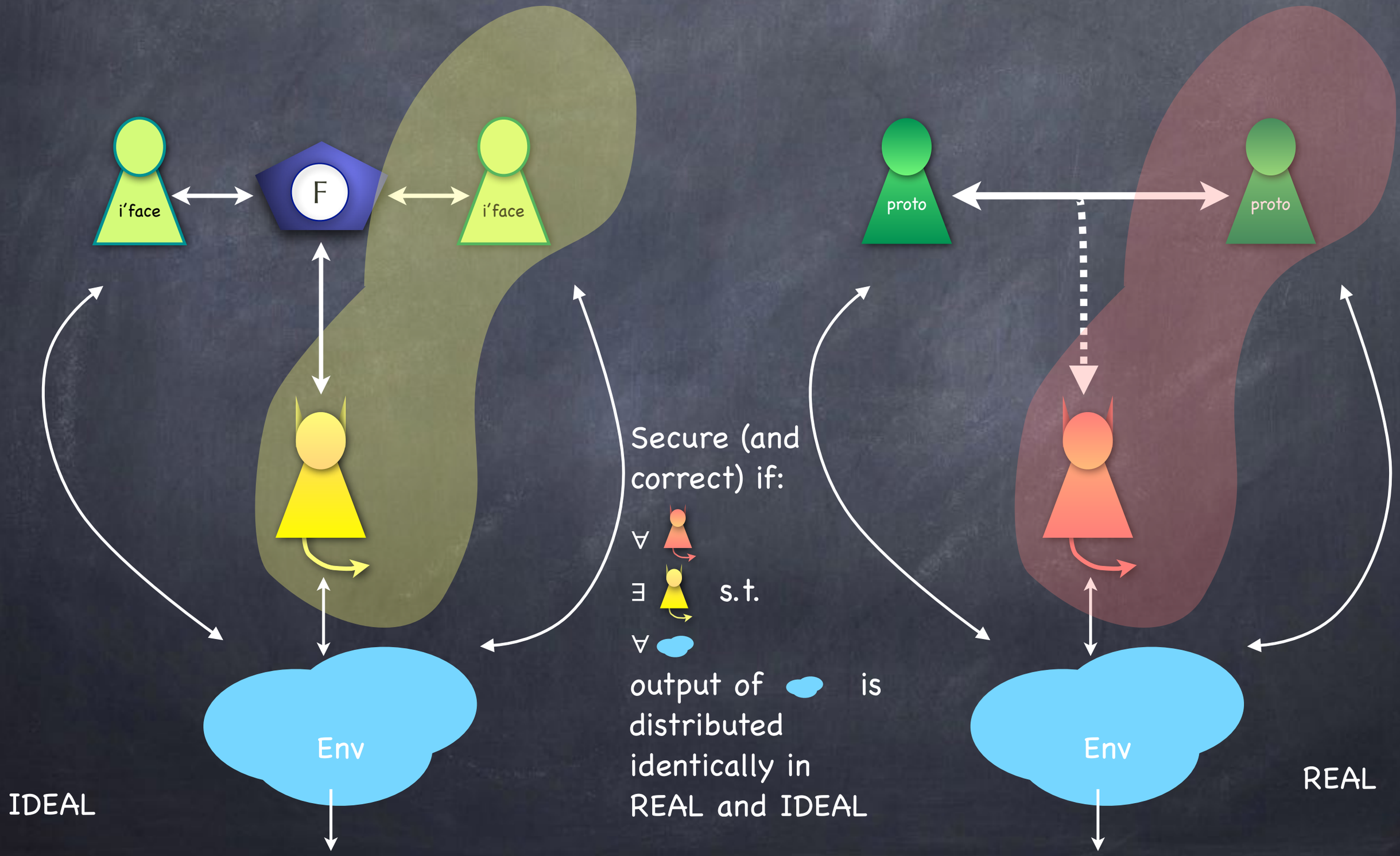
Open Problem: What about randomised functions?

Active Security

- Security against active adversary
  - Corrupt party may deviate from the protocol
- Same security definition?
  - $\text{View}_{\text{Alice}} \longleftrightarrow X, A \longleftrightarrow Y, B$

    $X, A \longleftrightarrow Y, B \longleftrightarrow \text{View}_{\text{Bob}}$
- But no well-defined input!

Simulation-Based Security



Standalone (Active) Security

- Simulation based security definition where:
- Environment interacts with the parties and the adversary only before and after (not during) the protocol execution

Standalone Trivial Functions

- Standalone trivial ⇔ <u>uniquely</u> decomposable & <u>saturated</u>

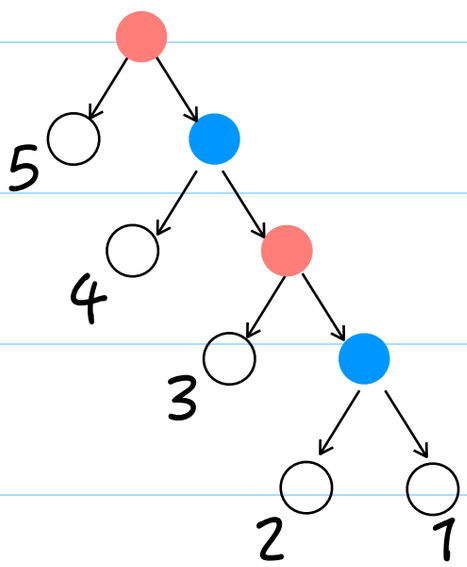
	0	1
0	0	1
1	1	0

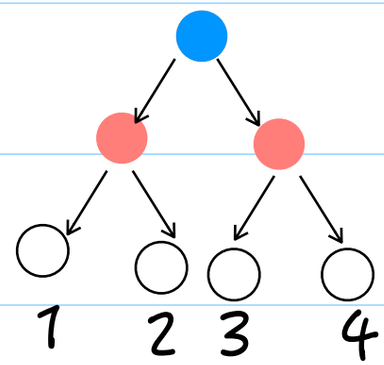
	1	3	5
0	1	3	5
2	2	3	5
4	4	4	5

	00	01	11	10
0	1	1	2	2
1	3	4	4	3

Not uniquely decomposable:

No active-secure protocol!





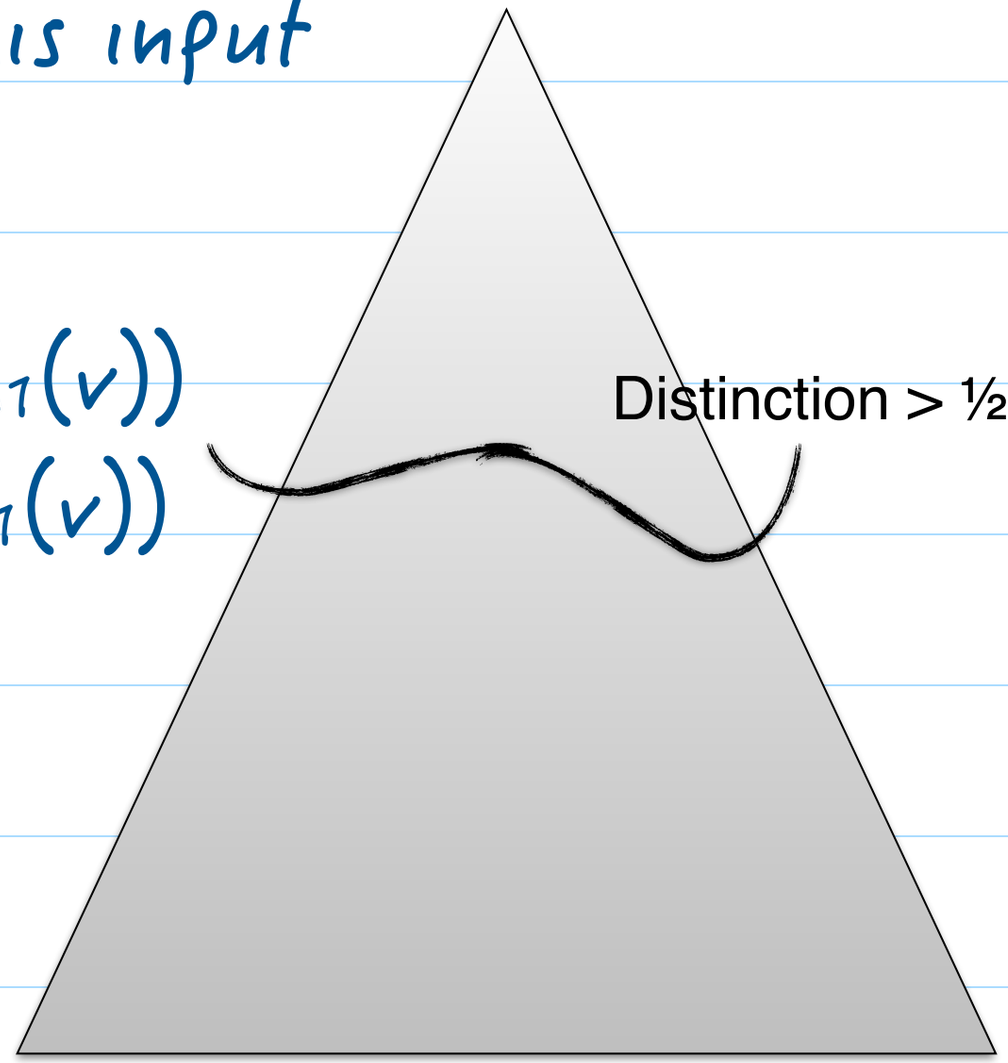
Frontier for XOR Protocol

- Frontier F : the first point where either Alice or Bob allows a "½-distinction" of her/his input

- Distinction at node v:

  $D_A(v) = |A_0(v)-A_1(v)| / (A_0(v)+A_1(v))$

  $D_B(v) = |B_0(v)-B_1(v)| / (B_0(v)+B_1(v))$



- Full frontier must exist:

  eventually Alice & Bob divulge their inputs completely

  (Distinction = 1, ignoring error probability)

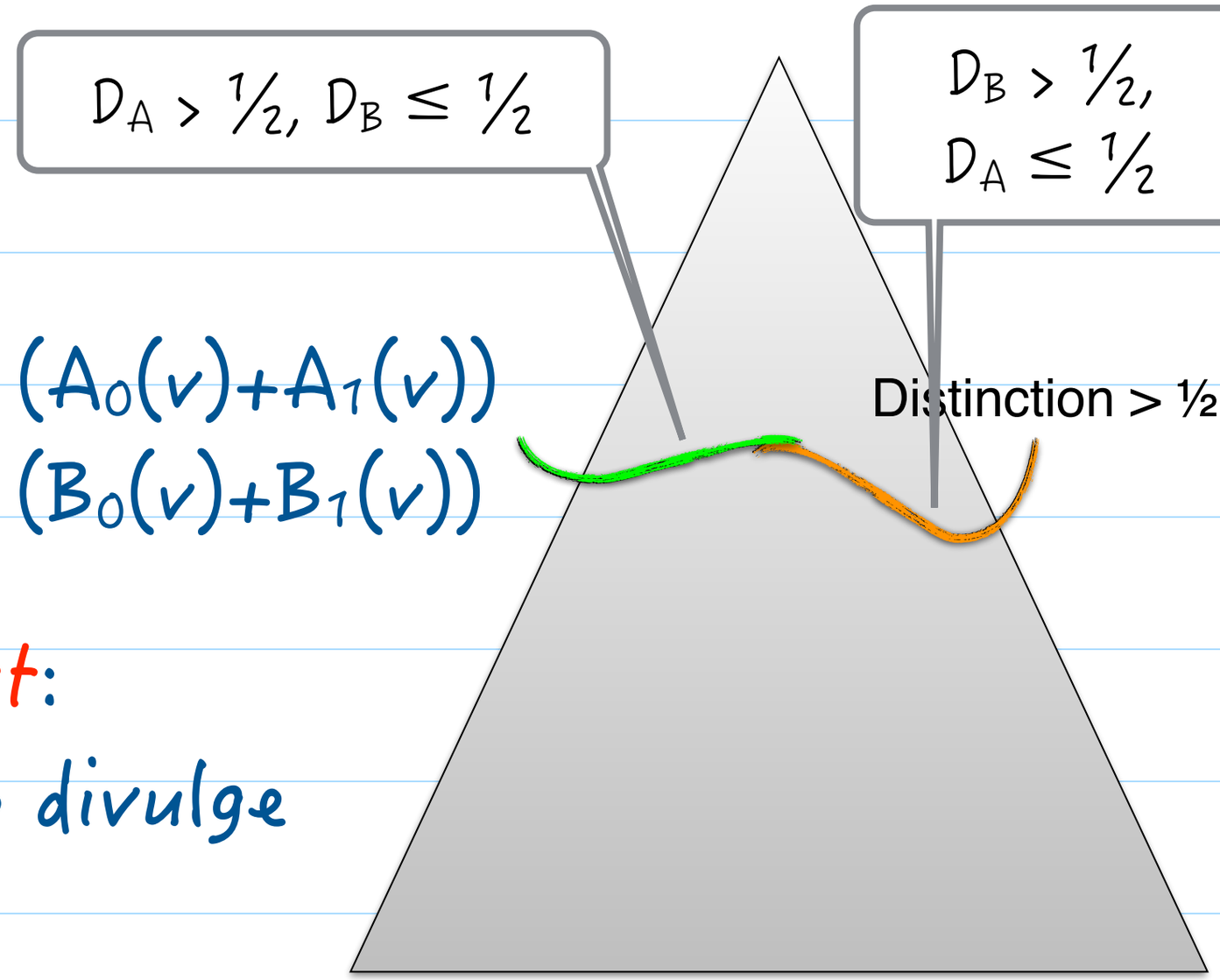
Frontier for XOR Protocol

- Frontier F : the first point where either Alice or Bob allows a "½-distinction"

- Distinction at node v:

  $D_A(v) = |A_0(v)-A_1(v)| / (A_0(v)+A_1(v))$

  $D_B(v) = |B_0(v)-B_1(v)| / (B_0(v)+B_1(v))$



- Full frontier must exist:

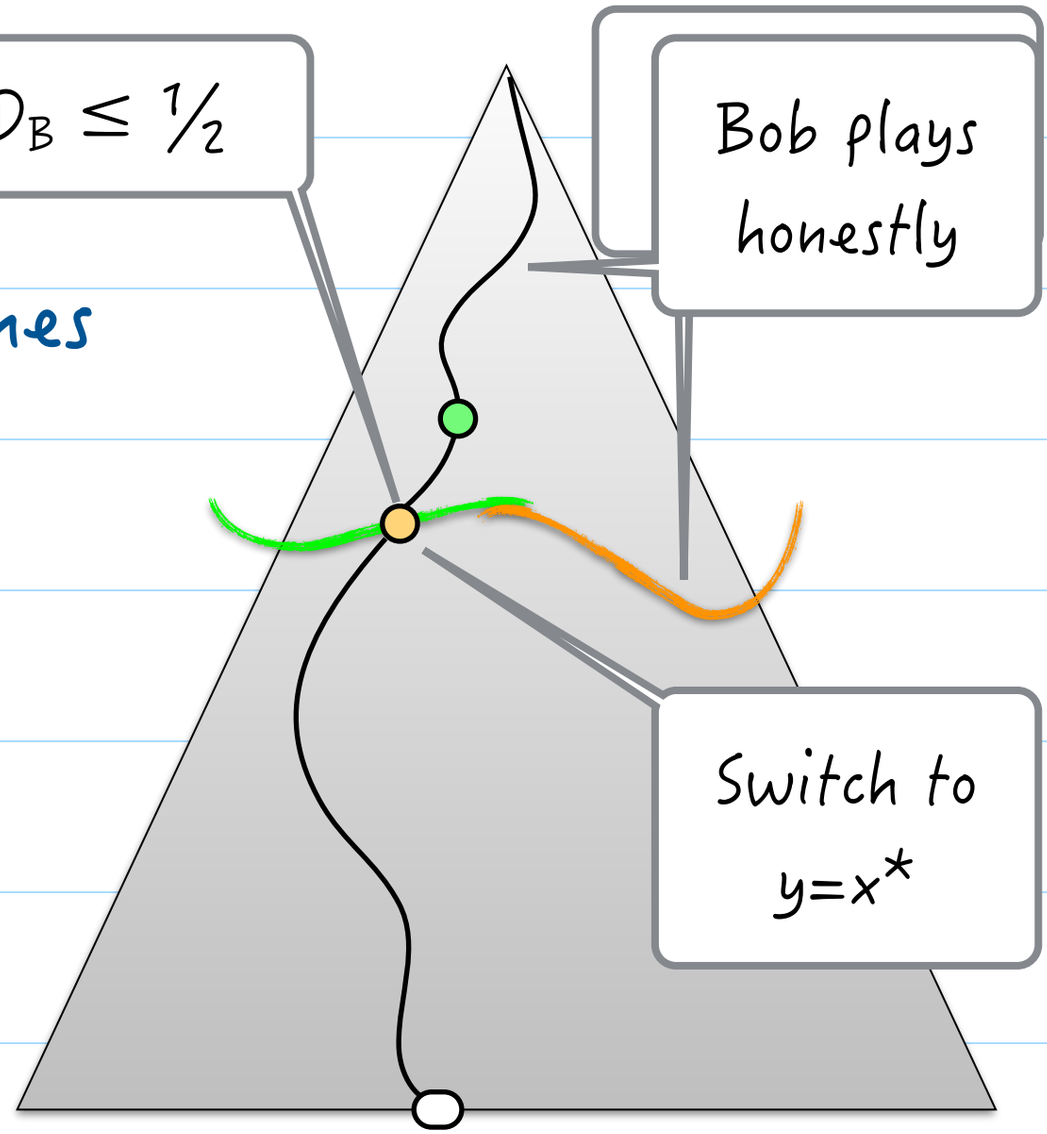
  eventually Alice & Bob divulge their inputs completely

  (Distinction = 1, ignoring error probability)

- $F = F_A \cup F_B$

Active Attack on XOR

- Suppose more weight on F_A than F_B.
  - Then Bob attacks:
  - If transcript hits F_A, then Bob switches y to x^*, the more likely value for Alice's input
    - $D_B \ll 1 \Rightarrow$ Bob hasn't fully revealed his input. So switching "legitimate"
    - $D_A \gg 0 \Rightarrow$ Outcome biased towards 0



- Valid attack: Cannot force this in the ideal setting if Alice's input is random

Standalone Trivial Functions

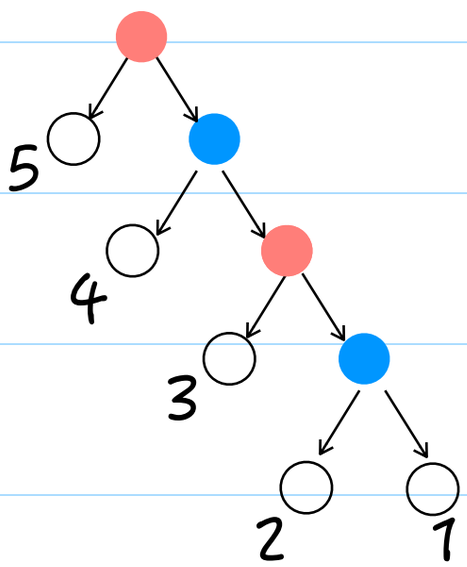
- Standalone trivial ⇔ <u>uniquely</u> decomposable & <u>saturated</u>

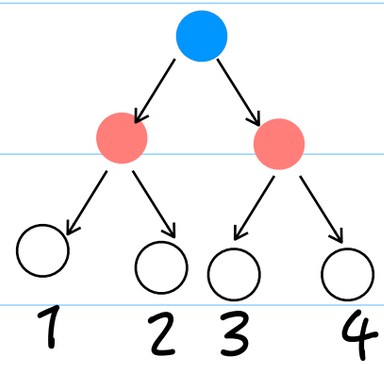
	0	1
0	0	1
1	1	0

	1	3	5
0	1	3	5
2	2	3	5
4	4	4	5

	00	01	11	10
0	1	1	2	2
1	3	4	4	3

Not uniquely decomposable:
No active-secure protocol!





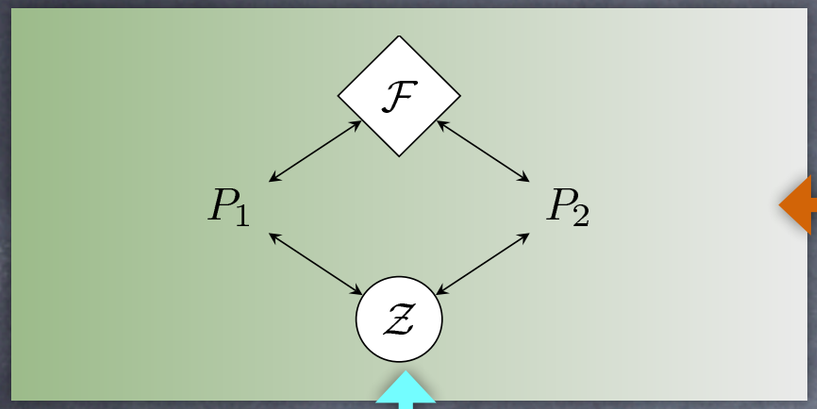
UC (Active) Security

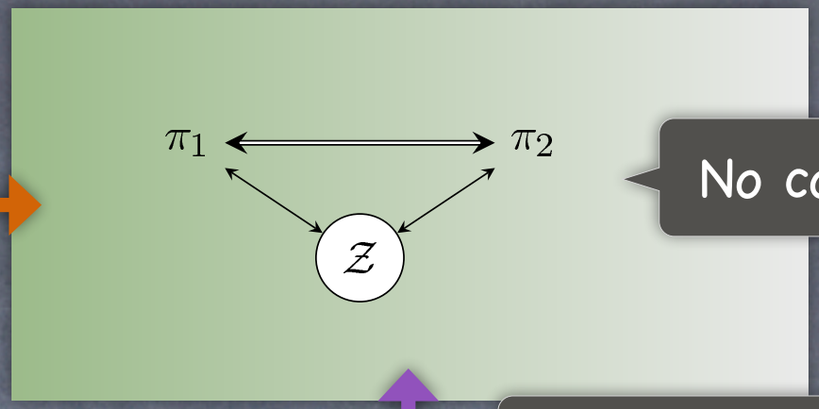
- Active security with a "live" environment
  - Adversary can interact with the environment throughout the protocol
  - In particular, adversary can be under the control of the environment

Impossibility of UC Security

Indist. by security

Identical systems

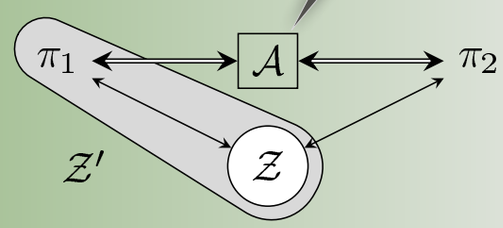


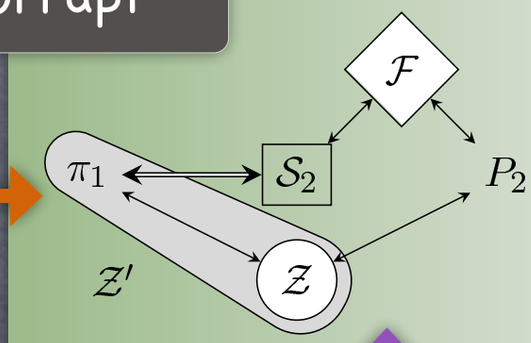


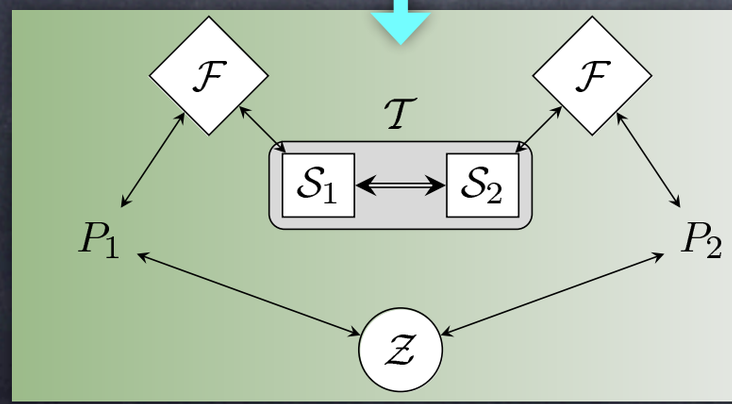
No corruption

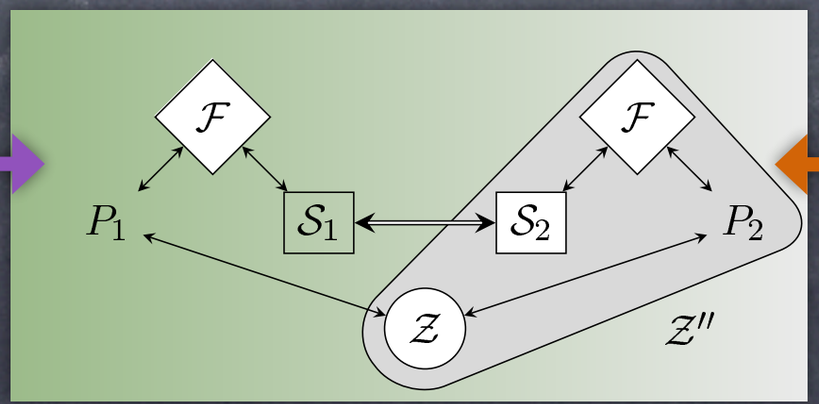
F has a UC-secure protocol only if F is "splittable"
Very few are splittable!

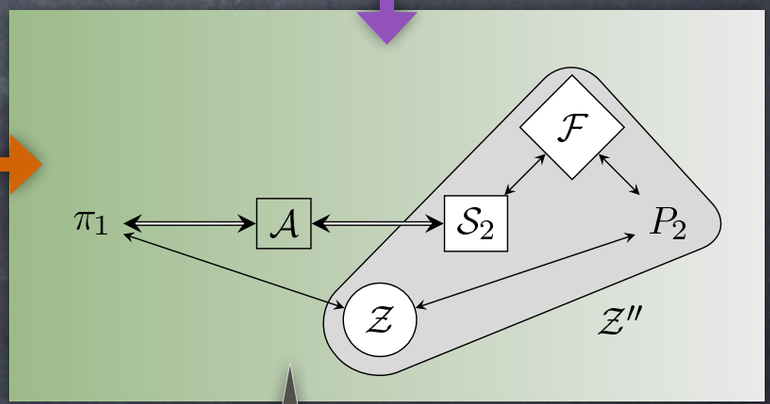
Party 1 corrupt







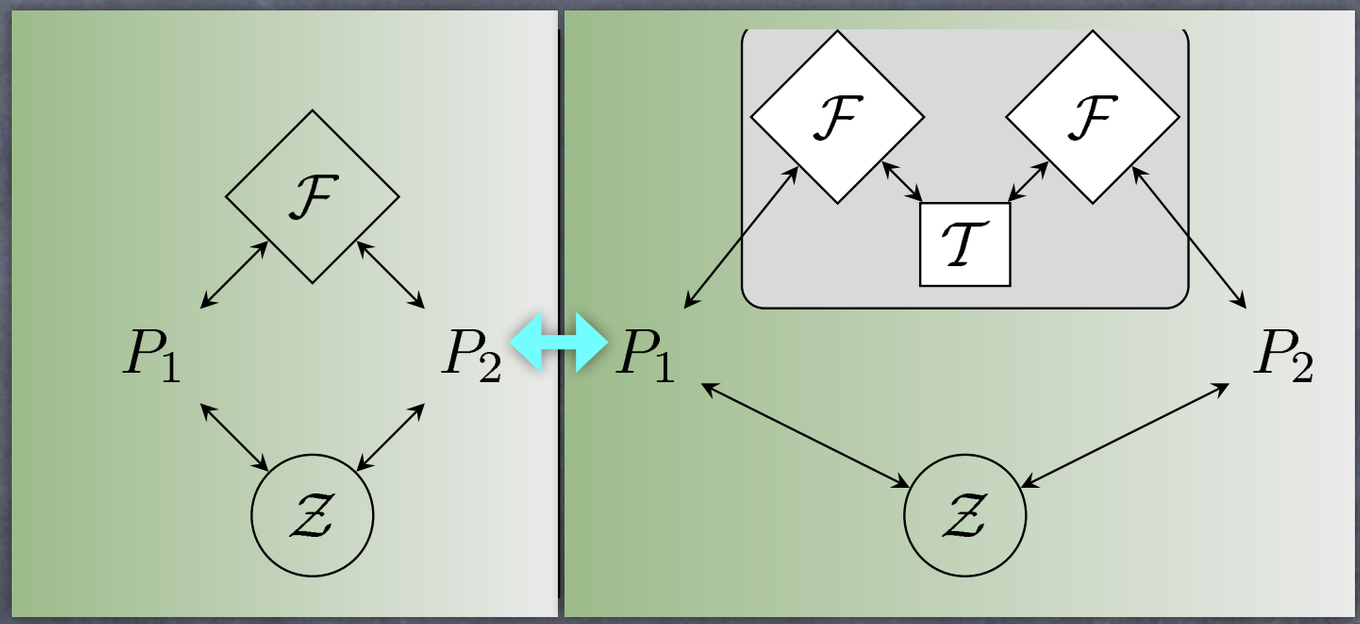




Party 2 corrupt

Splittable Functionalities

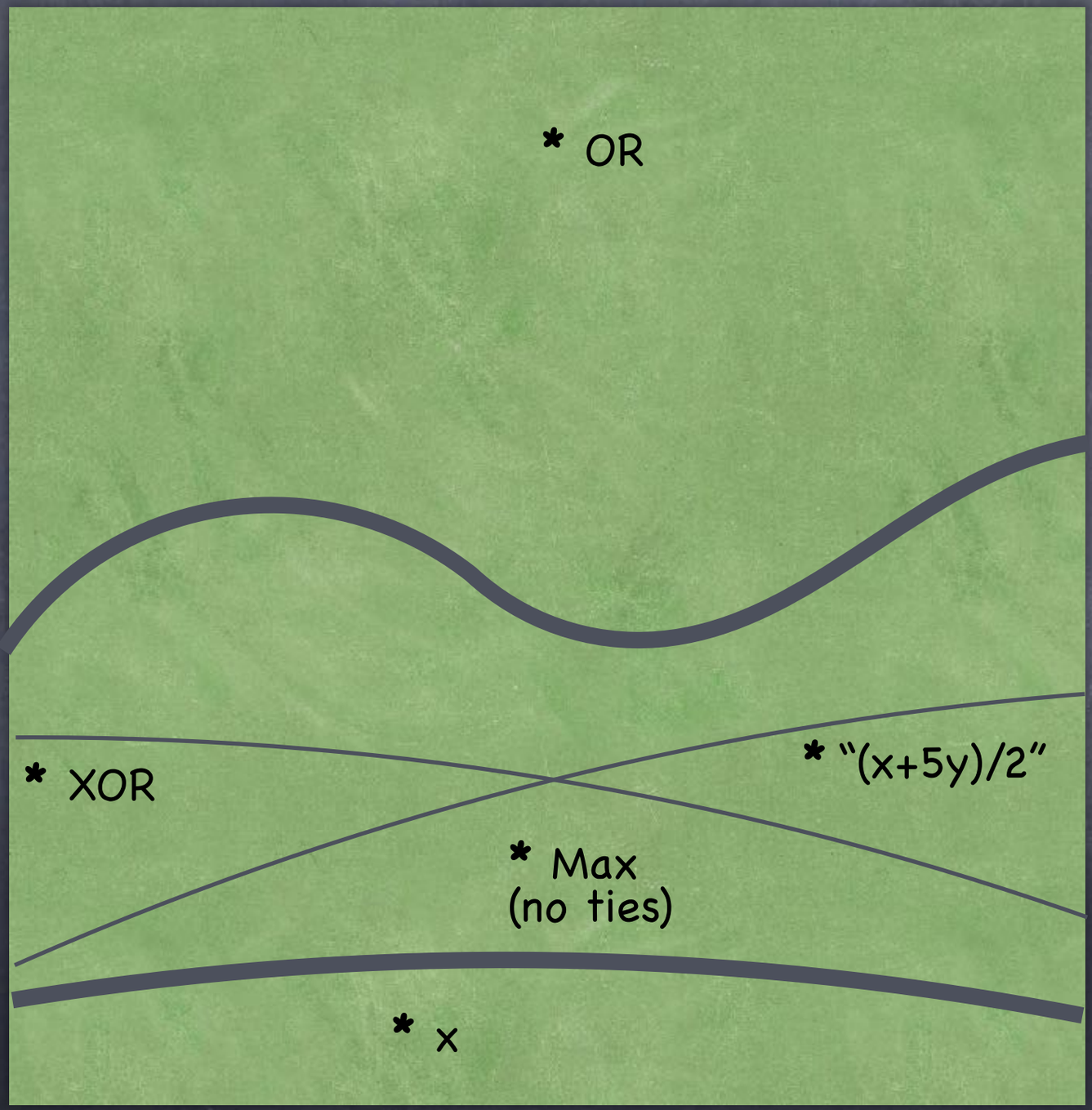
- F splittable if ∃T ∀Z the outputs of Z in the following two experiments are negligibly far from each other:



- Splittable functionality essentially involve only communication and local computation. All splittable functionalities have UC-secure protocols.

- Most interesting functionalities are unsplittable. E.g., coin-tossing, commitment, XOR, OT, decomposable functions with depth > 1, ...

A Map of 2-Party Functions



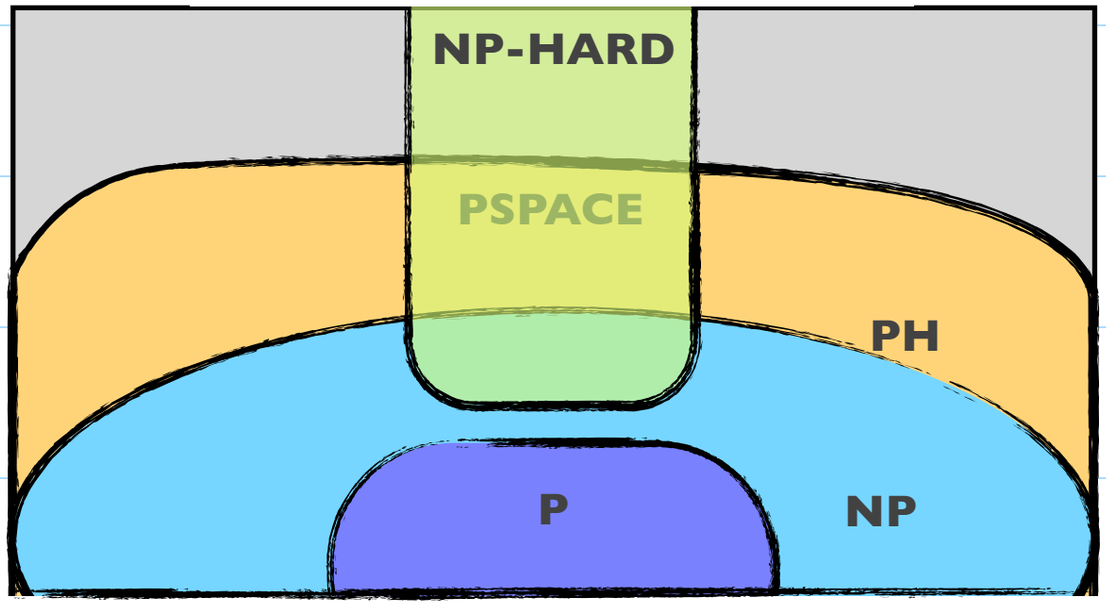
Cryptographic Complexity

- How hard is to <u>securely compute</u> a (multi-party) function?

- cf. Computational complexity

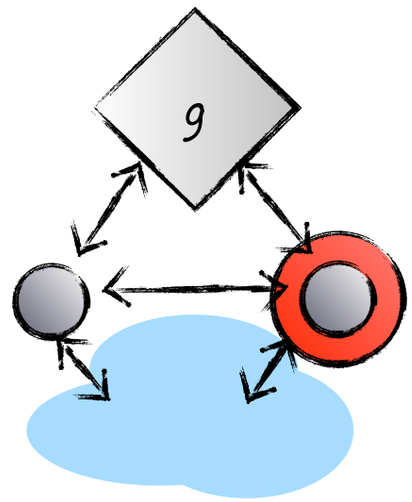
- Hard a la NP-hard?

- In terms of <u>reductions</u>



Secure Reduction

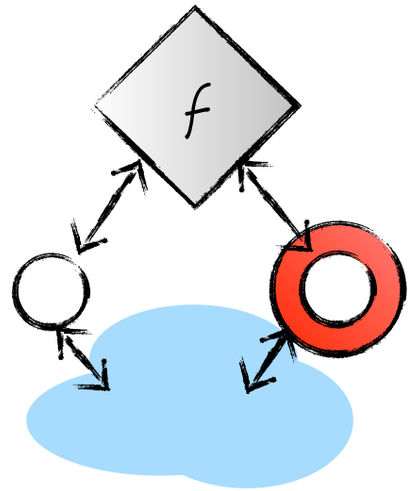
- Reducing functionality f to functionality g
  - A protocol for f, given access to g
- Parties can adaptively decide on what inputs to send to g (and even in which rounds to access g)
- $View_{Alice}$ and $View_{Bob}$ involve their side of the input/output with g



$P_{X,A,Y,B,View(Bob)}$

REAL

IDEAL

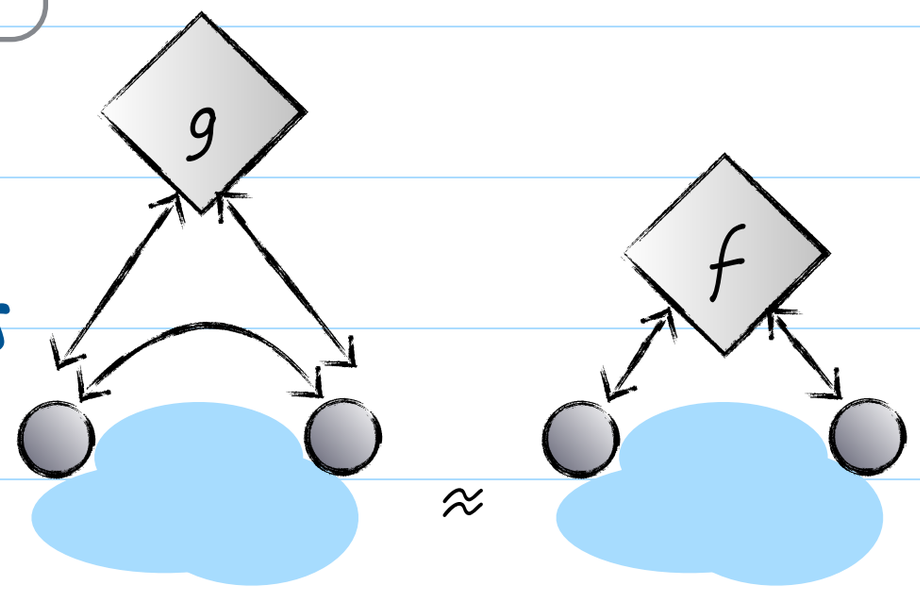


$P_{X,A,Y,B} \times P_{View(Bob) \mid Y,B}$

Secure Reduction

for various notions of security

- f reduces to g if there is a protocol that securely realizes f using g



- g is Complete if everything reduces to g

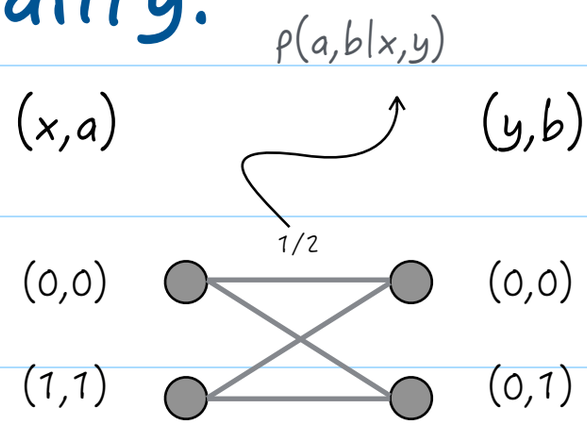
- f is Trivial if f reduces to everything

Complete Functions

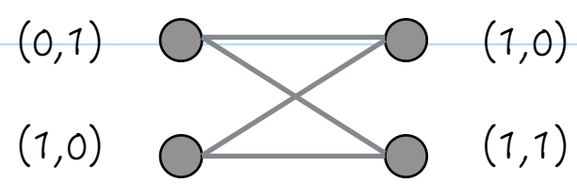
- Is there any complete function at all?
- e.g., Oblivious Transfer [Wie.'70, Rab.'81]
  - n-choose-1 OT: $X = (X_1, \cdots, X_n)$, $Y = i$, $OT_A(X,Y) = \perp$, $OT_B(X,Y) = X_i$
- Passive protocol for arbitrary f: Alice with input x sends $X_i = f(x, y_i)$, and Bob with input y_i sends i to OT.
- Here n = size of Bob's input alphabet
- Can reduce n-choose-1 OT to 2-choose-1 OT easily

Complete Functions

- Characteristic bipartite graph of a functionality:
  - Node sets X×A and Y×B
  - Weight((x,a),(y,b)) = Pr[a,b|x,y]



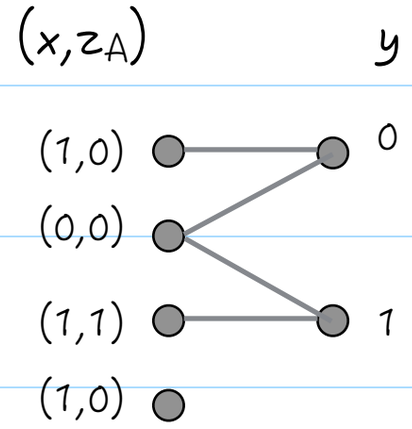
- Simple: Each connected component in the characteristic bipartite graph has edge weight ∝ product of node weights



  - Alternately: "isomorphic" to "common information" functionality
- Passive Complete iff **not simple** [Kil'00,MPR'12]

Complete Functions

- Passive Complete iff not simple [Kil'00,MPR'12]
  - e.g. f gives $x \wedge y$ to Alice only



- Is this complete for active corruption?
  - If Alice is actively corrupt, she can use (even in ideal model) input 1 and learn Bob's input
    - Bob may as well send her his input: secure protocol!
    - Function not complete (in fact, trivial) for active corruption!

Complete Functions

- Passive Complete iff **not simple** [Kil'00,MPR'12]
  - e.g. f gives $x \wedge y$ to Alice only
    - $x=0$ is a redundant input (for active adversaries)
    - First (iteratively) remove all redundant inputs and outputs → **core**
- Active Complete iff **core is not simple** [KMPS'14,KKMPS'16]

Intermediate Levels

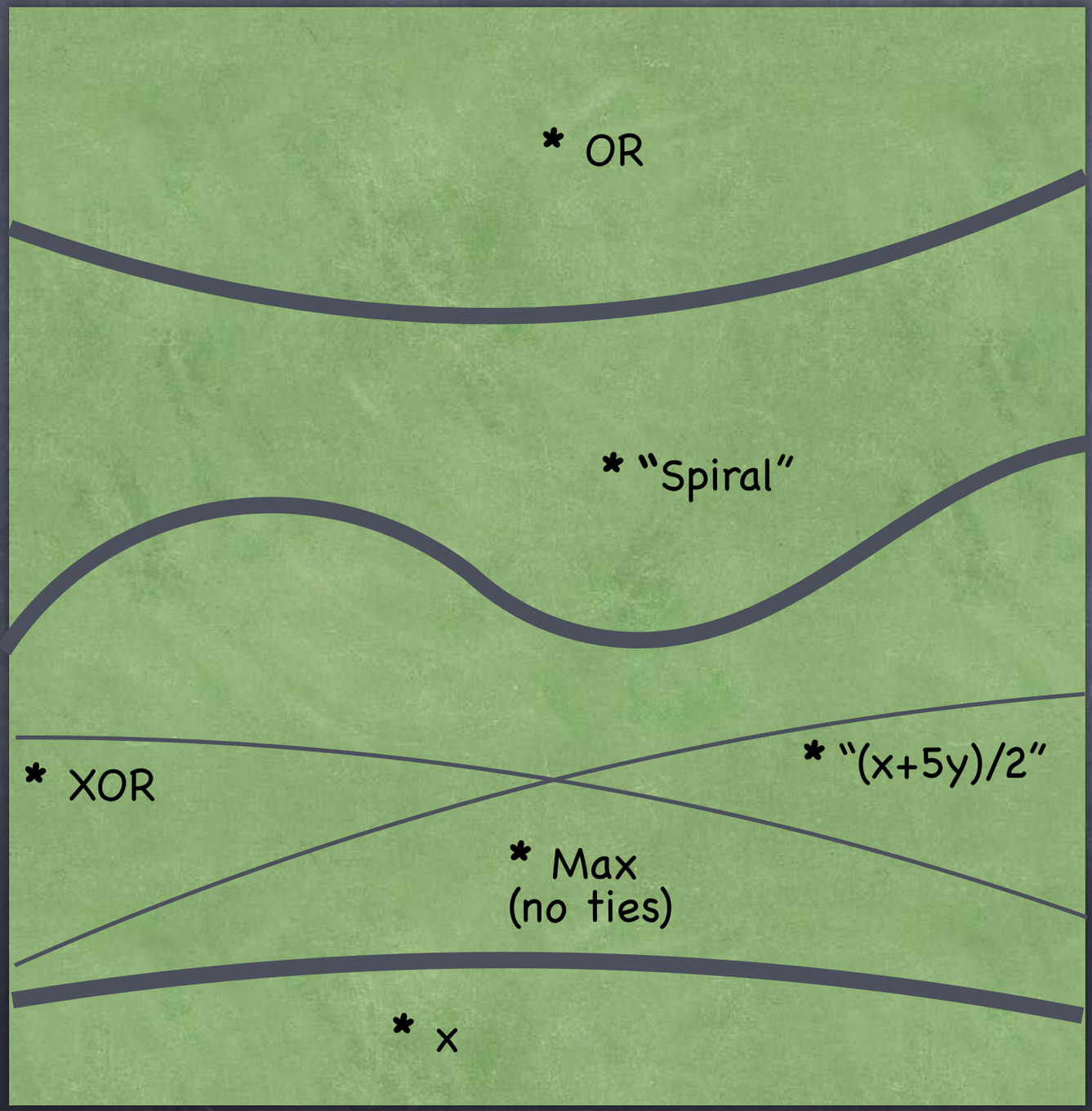
- Between trivial and complete:
  - For passive security:
    - Nothing in between for input-less (sampling) functions:
      Characterizations for trivial/complete are complementary
    - Otherwise, examples known:
  - For active (UC or standalone) security:

1	1	2
4	5	2
4	3	3

1	1	2	3
2	4	4	3
2	3	1	1

    - Infinitely many levels!
    - e.g. n-bit XOR doesn't reduce to (n-1)-bit XOR [MPR'09]
      - Further XOR doesn't reduce to Coin [MOPR'11]

A Map of 2-Party Functions



Non-Simple

Decomposable

Uniquely Decomposable

Saturated

Splittable

Summary

- For 2-party functions, we have full characterization of:
  - Complete functions under all security notions (UC, standalone, passive) [GV'87,Kilian'88,⋯,MPR'12,KMPS'14,KKMPS'16]
- Trivial functions under:
  - UC security [CKL'03,PR'08]
  - Standalone & passive security, restricted to deterministic functions [Kus.'89,Bea.'89,KMR'09,MPR'09]
    - Open: Randomized functions

Quiz

	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

- What's the complexity of the following 3 functions, w.r.t, passive secure MPC?
  - max(x,y) — Complete
  - [x < y] — Complete
  - (max(x,y), [x < y]) — Trivial (Passive and Standalone/Active)

	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0

	0	1	2	3
0	0	1	2	3
1	1′	1	2	3
2	2′	2′	2	3
3	3′	3′	3′	3

Secure Multi-Party Computation

&

Quantitative Cryptographic Complexity

Manoj Prabhakaran
IIT Bombay

Randomized 2-Party Functions

- Functionality defined by $P_{A,B|X,Y}$
  - e.g., Secure Function Evaluation: $A=f_A(X,Y)$ and $B=f_B(X,Y)$
  - e.g., Secure Sampling: when X, Y are empty
- e.g., Oblivious Transfer [Wie.'70,Rab.'81]
  - SFE variant: $X=(S_0,S_1)$, $Y\in\{0,1\}$, $A=\perp$, $B=S_Y$
  - Sampling variant: $A=(S_0,S_1)$, $B=(C,S_C)$
    - Fact: The two are "isomorphic" to each other

e.g., String-OT or Bit-OT

EXAMPLE

Bit OT

Binary Erasure Channel, a.k.a. Rabin-OT

- Reducing BEC to Bit OT (Alice's input be w, Bob's output z)
  - Alice & Bob invoke Bit OT, with Alice's inputs $(w,0)$ and Bob's input a random bit b. Bob outputs erasure if $b=1$, else outputs w.
- Reducing Bit OT to BEC

amortizes : n bit OTs from ≈ 2n BEC uses

  - Alice sends κ bits $w_1,\cdots,w_\kappa$ to Bob over BEC
  - Bob sends back two random indices i_0, i_1 such that i_b was not erased and i_{1-b} was
  - Alice sends $x_0 \oplus w_{i0}$ and $x_1 \oplus w_{i1}$. Bob decodes and outputs x_b.

f-Capacity of g

- Supremum over all protocols $\Pi_{n,\kappa}$ (optionally "uniform")
  - that implement n copies of f
  - with error a negligible function of κ (for <u>every</u> n)
- $\lim_{n\to\infty}$ n / max # copies of g used by $\Pi_{n,\kappa}$ (as a function of κ)
- cf. Channel capacity for communication: κ identified with n

---

- Capacity* for functions with a size parameter : Considers protocols $\Pi_{n,\kappa}$ that securely implement one copy of f of size n
  - e.g., String-OT Capacity* of BEC

EXAMPLE

Secret/Private-Key Capacity

- Functionality f involving 3 parties (or $m+1$ parties for the m-terminal version)
  - No inputs
  - Outputs are $(K, K, \perp)$, where K is uniform over n-bit strings
- Functionality g: no input, and outputs (X, Y, Z)
- Private-Key Capacity: f-Capacity* of g, for passive security
- Secret-Key Capacity: Restrict to protocols in which the 3rd party can only listen

EXAMPLE

String-OT from BEC

- Passive secure protocol

$m \approx n \cdot \min(p, 1-p)$

Send n uniform bits $x_0, \cdots, x_n$	$\Rightarrow$	Receive $y_0, \cdots, y_n$
Derive one-time pads, $pad_0 = x_{J_0}$ and $pad_1 = x_{J_1}$ respectively	$\leftarrow$	Random $J_0, J_1 \subseteq [n]$, $\lvert J_0 \rvert = \lvert J_1 \rvert$, s.t $y_{J_b} = x_{J_b}$ and $y_{J_{1-b}} = \perp$
$c_0 = s_0 \oplus pad_0$ $c_1 = s_1 \oplus pad_1$	$\rightarrow$	Output $s_b = c_b \oplus y_{J_b}$

- String-OT Capacity* of BEC $\geq \min(p, 1-p)$
- Is this tight? Yes!
  - Intuition: Needs to keep m bits of Sender's input hidden, but potentially revealed

$\Rightarrow$ m erasures & m non-erasures needed

EXAMPLE

OT Capacity of String-OT

- (How) does it depend on the length of the string?
- Answer: it doesn't! Only one bit-OT per string-OT!
  - Intuition: Even in string-OT only one bit of Bob's input is hidden from Alice
- What about Symmetrized-String-OT?
  $X=(S_0,S_1,c)$, $Y=(T_0,T_1,b)$, $A=T_c$, $B=S_b$
- Answer: Still capacity = 1! (i.e., Only 1 pair of bit-OTs per pair of Symm-String-OT)

How do we know?

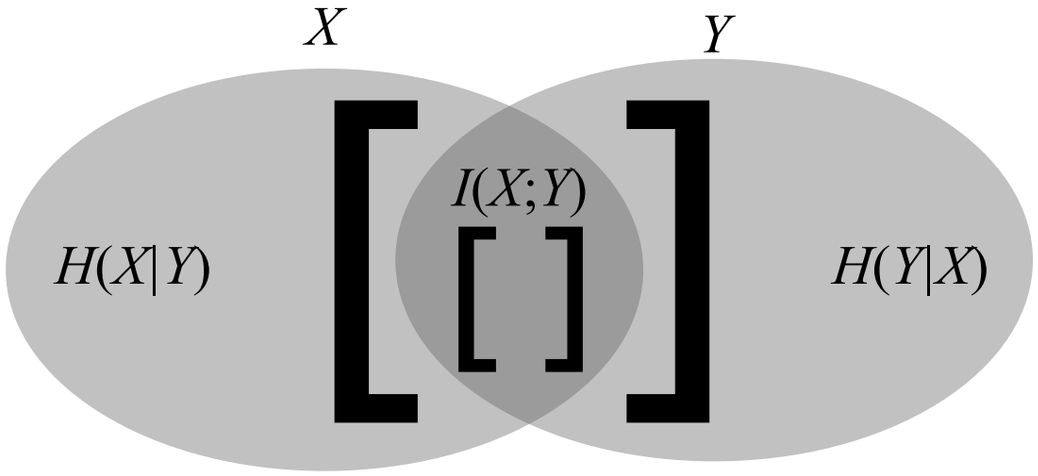
Sources

- Source: input-less functionality
- Source "corresponding to" a functionality:
  - $A'=(X,A)$, $B'=(Y,B)$, where A,B according to functionality and X,Y independent (say uniform)
- Functionality can always be used to passive-securely implement the corresponding source at rate 1
- In many cases, the source can be used to securely implement the functionality too at rate 1
  - e.g., OT, BEC, BSC, …

Monotones

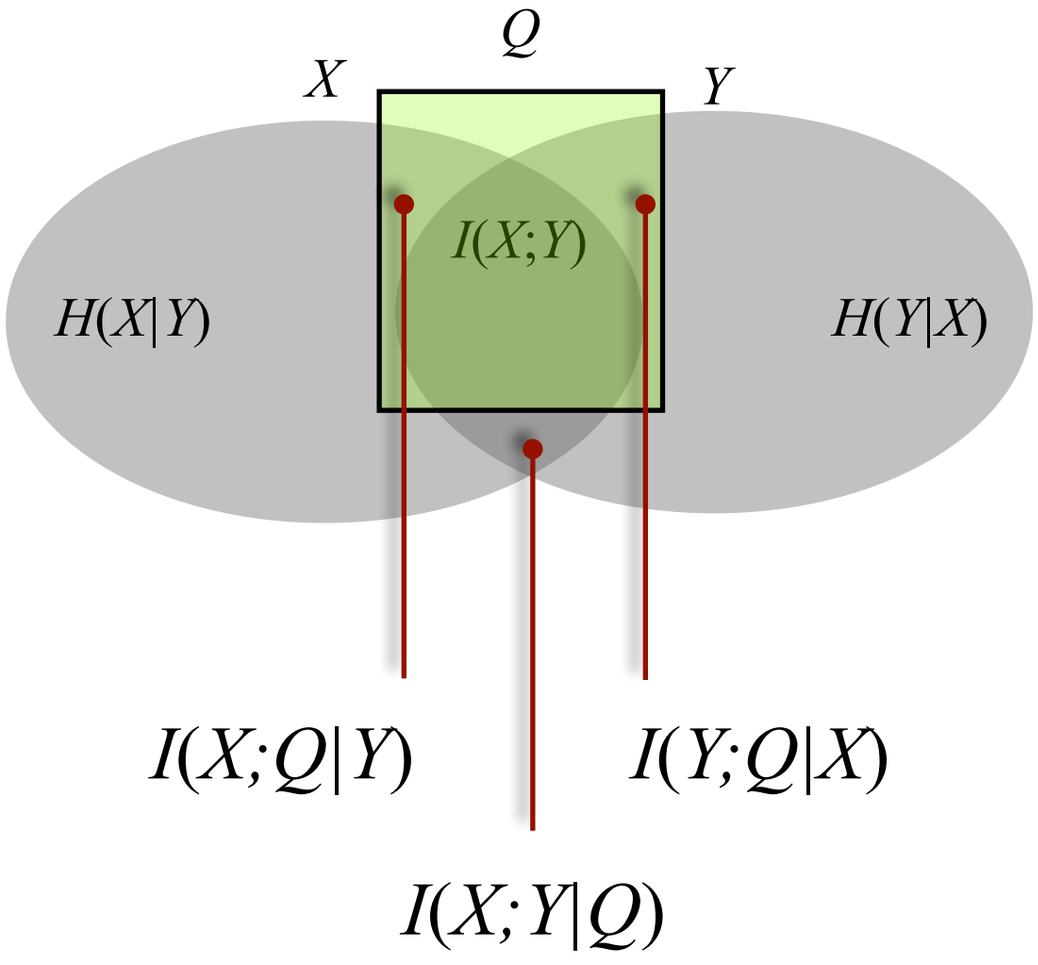
- Goal: Measure non-trivial "cryptographic content" of a source $(X;Y)$
  - Non-trivial: cannot be generated/increased by discussion
  - Comparing the amount at the beginning of a protocol and at the end gives an upper bound on the rate
- Monotone: a quantity that can only decrease during the course of a protocol
  - e.g., Gap between mutual information & common information of the views of the two parties

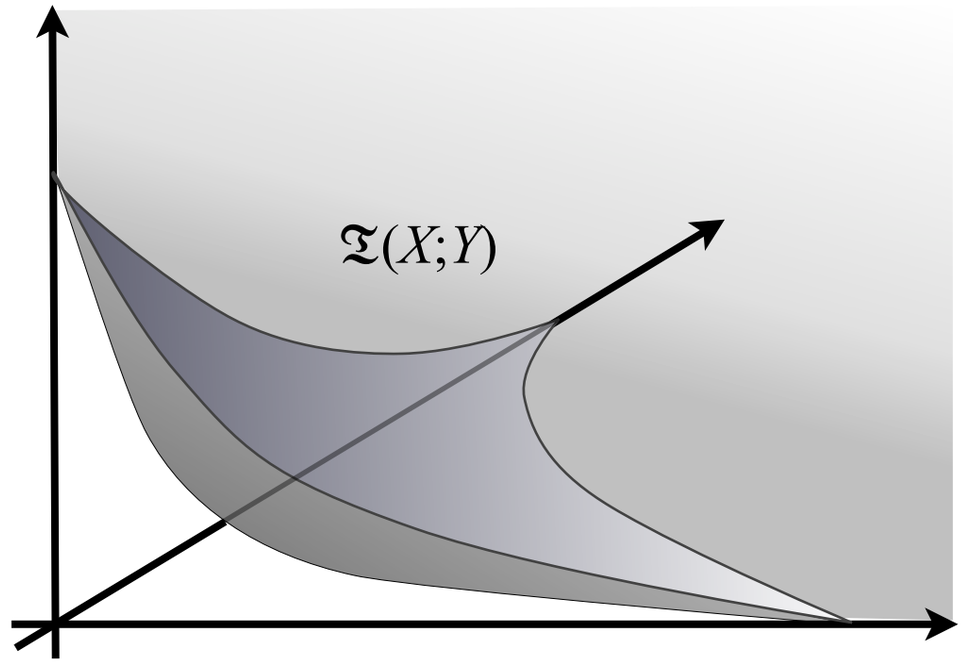
Understanding Correlation



- "Trivial" correlation: independent (P,Q,R) and X=PQ, Y=QR
  - Q exactly captures all the correlation
- In general, there maybe no such random variable. Then, it is a cryptographically "non-trivial" correlation
- Common Information: Random variable that best captures correlation

Understanding Correlation





Tension Region:

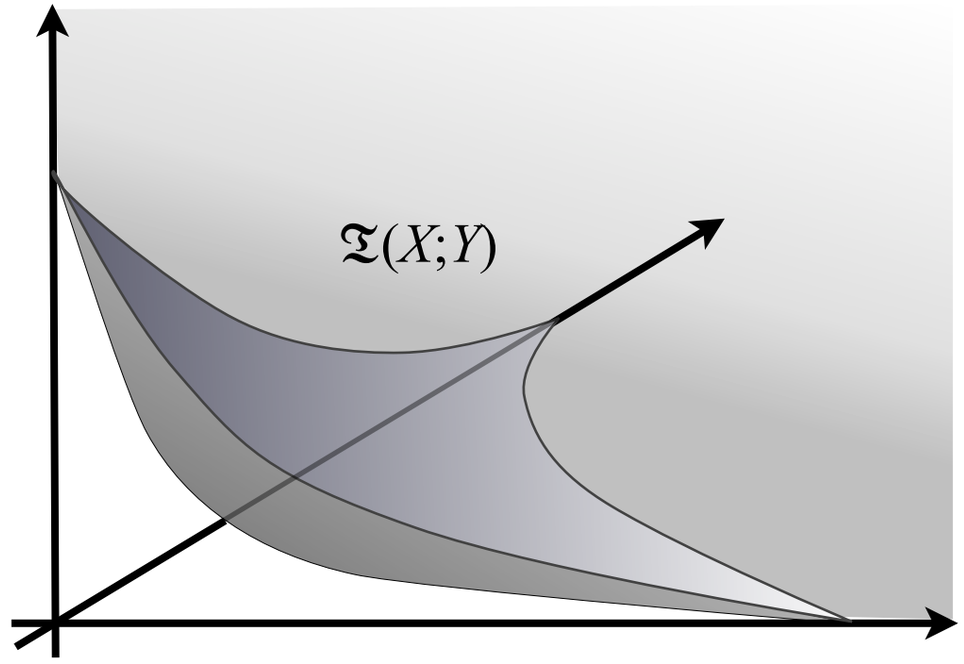
$\mathfrak{T}(X,Y) = \{ (a,b,c) : \exists\ Q$ jointly dist. with X,Y s.t.

$a \geq I(X;Q|Y)$,

$b \geq I(Y;Q|X)$

$c \geq I(X;Y|Q)$) }

- Theorem: cannot increase tension (shrink this region) by a secure protocol that derives $(A^m;B^m)$ from $(U^n;V^n)$:
  - $\mathfrak{T}(U^n;V^n) \subseteq \mathfrak{T}(View_1 ; View_2) \subseteq \mathfrak{T}(A^m ; B^m)$



- $\mathfrak{T}$ for independent copies add up (Minkowski sum). In particular:

$$\mathfrak{T}(X^n;Y^n) = n \cdot \mathfrak{T}(X;Y)$$

- Corollary: If $(A;B)$ can be derived from $(U;V)$ at rate r, then

$$\mathfrak{T}(U;V) \subseteq r \cdot \mathfrak{T}(A;B)$$

$\mathfrak{T}(X;Y) = \{ (a,b,c) : \exists\ Q$ jointly dist. with X,Y s.t.
$a \geq I(X;Q|Y)$,
$b \geq I(Y;Q|X)$
$c \geq I(X;Y|Q)$) }

Tension Region

- Generalizes monotones of [WW'05]

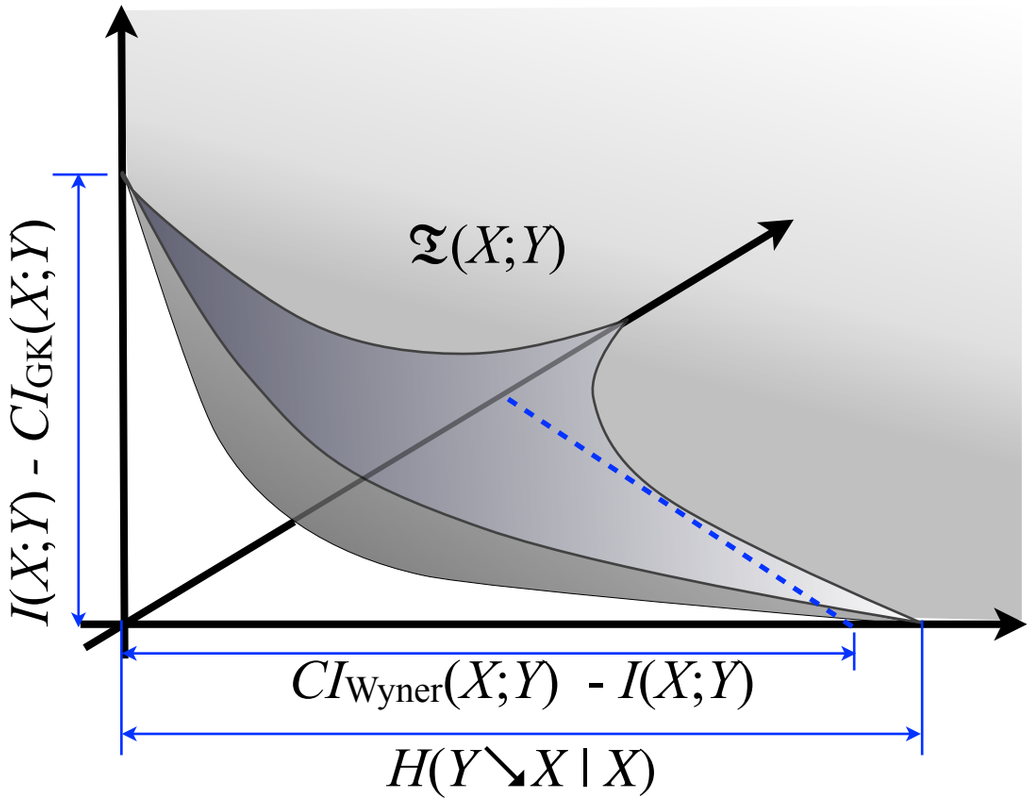
$$I(X;Y|X\wedge Y) = \min_{Q:\, H(Q|X)=H(Q|Y)=0} I(X;Y|Q)$$

$$H(Y\searrow X \mid X) = \min_{Q:\, H(Q|Y)=I(X;Y|Q)=0} H(Q|X)$$

$$H(X\searrow Y \mid Y) = \min_{Q:\, H(Q|X)=I(X;Y|Q)=0} H(Q|Y)$$

- Generalizes bounds used in [AC'07] (take Q=const, Q=Y, Q=X resp.)

- Gives a new monotone of interest: gap between Wyner common information and mutual information



$$\mathfrak{T}(X;Y) = \{\, (a,b,c) : \exists\ Q \text{ jointly dist. with } X,Y \text{ s.t.}$$
$$a \geq I(X;Q|Y),$$
$$b \geq I(Y;Q|X)$$
$$c \geq I(X;Y|Q)\ )\ \}$$

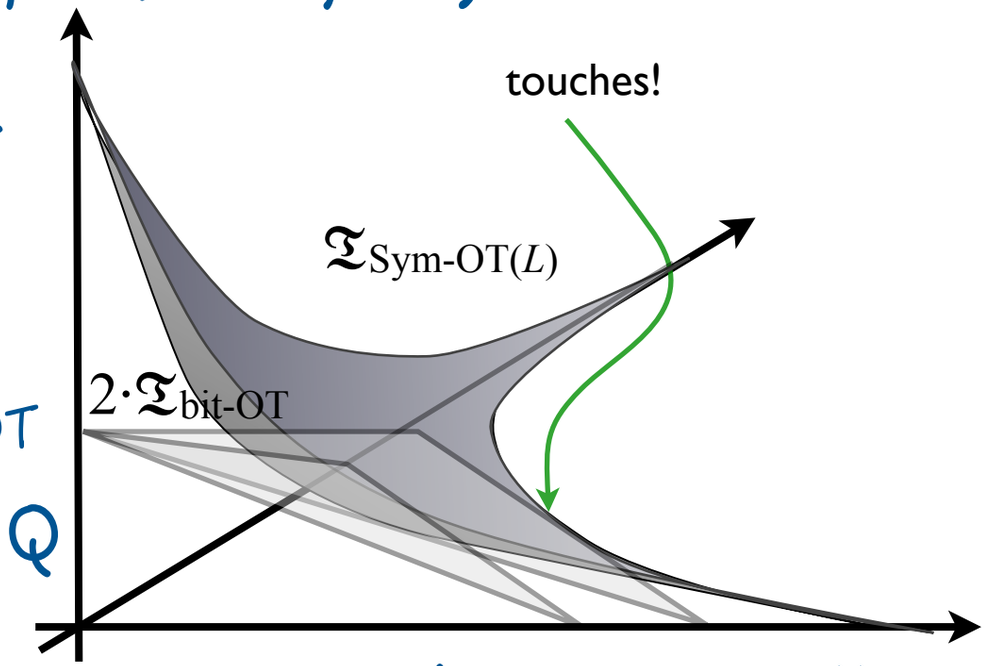
EXAMPLE

OT Capacity of Sym-OT(L)

- We bound $\mathfrak{T}_{\text{Bit-OT}}$ away from the origin, and show $\mathfrak{T}_{\text{Sym-OT}(L)}$ has a point close enough to the origin
- For $\mathfrak{T}_{\text{Sym-OT}(L)}$, $Q=(b,S_b,c,T_c)$ gives
  $I(X;Y|Q)=0$, $I(X;Q|Y)= I(X;Q|Y) = 1$ (independent of L!)

$$I(S_0,S_1,c,T_c\ ;\ b,S_b,c,T_c \mid T_0,T_1,b,S_b) = H(c) = 1$$

- Can get only one pair of bit-OTs from per pair of Sym-OT(L)!
- Main computation involved: to bound $\mathfrak{T}_{\text{bit-OT}}$ away from the origin, need to consider all Q
- We consider the plane $I(X;Y|Q)=0$ so that we can restrict to a small class of joint distributions for (X,Y,Q)



ExTensions

- Behaviour of tension, when using a functionality (not a source)
  - Partial result (i.e., for channel functionalities) in [RP'14]
- Multi-Party Tension
  - Preliminary proposals in [PP'12] (+ unpublished notes)
    - Enough to characterise trivial functionalities in the broadcast channel model
    - Enough to subsume [RW'03] bound for Secret-Key Capacity
      - Can also reproduce [GA'10] bound, but using a different (somewhat ad hoc) monotone
- Applications to communication complexity of MPC [DPP'14]

Active-Security

- Stand-alone security
  - Original definition from the 80's [GMR'85,GMW'87]
  - Assumes a closed system with the parties executing a single instance of the protocol
    - i.e., adversary doesn't communicate with the environment during the protocol execution (only a priori and a posteriori)
- Universally Composable (UC) security [Can.'01]
  - Adversary & environment can interact arbitrarily

String-OT from BEC

- Passive secure protocol

m ≈ n·min(p,1-p)

Send n uniform bits $x_0,\cdots,x_n$	⇒	Receive $y_0,\cdots,y_n$
Derive one-time pads, $pad_0 = x_{J0}$ and $pad_1 = x_{J1}$ respectively	←	Random $J_0, J_1 \subseteq [n]$, $\lvert J_0\rvert=\lvert J_1\rvert$, s.t $y_{Jb} = x_{Jb}$ and $y_{J1-b} = \perp$
$c_0 = s_0 \oplus pad_0$ $c_1 = s_1 \oplus pad_1$	→	Output $s_b = c_b \oplus y_{Jb}$

- Problem with an active adversary: Corrupt Bob may choose J_0 and J_1 to contain unerased positions, and learn parts of both s_0 and s_1

rate = min(p,1-p)/2

- Solution: $pad_i = Extract(x_{Ji})$, of length m/2, since there must be ≥ m/2 erased positions in at least one set!

String-OT from BEC

- Can we get rate $\min(p, 1-p)$?

Using an info-theoretic definition by Crepeau-Wullschleger'08

- For stand-alone security, yes! [CS'06,PDMN'11]
  - By enforcing that Bob knows x_i for (almost) all i in at least one of the two sets J_0, J_1
- Alice will challenge Bob on κ indices in each set
- Indices to challenge in the two sets selected using "Interactive Hashing" [OVY'91]
  - Lets Bob plant one of two values, but the other will be significantly influenced by Alice's choices

More Results

- (UC-secure) OT from BSC at constant rate [IKOPSW'11]
- Constant rate reductions from any (finite) f to any (finite) g, as long as g is "complete" [KMPS'14]
  - Also explicit characterization of complete functions
- Getting constant rate relies on (extensions of) techniques from [IPS'08] which in turn resembles those of [HIKN'07]
- Exact OT-capacities remain open for BSC (even passive security) and even BEC for UC-security