Cryptographic Complexity & Computational Intractability

Hemanta Maji | Manoj Prabhakaran | Mike Rosulek

ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

The University of Montana

Crypto Means & Goals

One-Way Functions

One-Way Permutations

Trapdoor One-Way Permutations

OT protocol

Collision-Resistant Hash Functions

Zero-Knowledge Proofs

Encryption

Signatures

Homomorphic Encryption

Secret Communication Channels

Authenticated Communication Channels

OT Channel

Digital Cash

Mental Poker

Privacy Preserving Data-Mining

E-Voting

Intractability

Functionalities

Functionalities

Functionalities

- A universe of functionalities: programs for a trusted party

Functionalities

- A universe of functionalities: programs for a trusted party
  - Several constituent ideas: Zero-knowledge/simulatability [GMR85], Ideal/Real paradigm [GMW87], Relative-Resilience [B91], ..., Reactive Simulatability[PW01],UC security [C01]

Functionalities

- A universe of functionalities: programs for a trusted party
  - Several constituent ideas: Zero-knowledge/simulatability [GMR85], Ideal/Real paradigm [GMW87], Relative-Resilience [B91], ..., Reactive Simulatability[PW01],UC security [C01]
- Motivates a Cryptographic Complexity Theory

Functionalities

- A universe of functionalities: programs for a trusted party
  - Several constituent ideas: Zero-knowledge/simulatability [GMR85], Ideal/Real paradigm [GMW87], Relative-Resilience [B91], ..., Reactive Simulatability[PW01],UC security [C01]
- Motivates a Cryptographic Complexity Theory
- Reduction $F \sqsubseteq G$: F can be securely realized given G

Functionalities

- A universe of functionalities: programs for a trusted party
  - Several constituent ideas: Zero-knowledge/simulatability [GMR85], Ideal/Real paradigm [GMW87], Relative-Resilience [B91], ..., Reactive Simulatability[PW01],UC security [C01]
- Motivates a Cryptographic Complexity Theory
- Reduction $F \sqsubseteq G$: F can be securely realized given G
  - Capturing extent of "cryptographic magic" in F, G

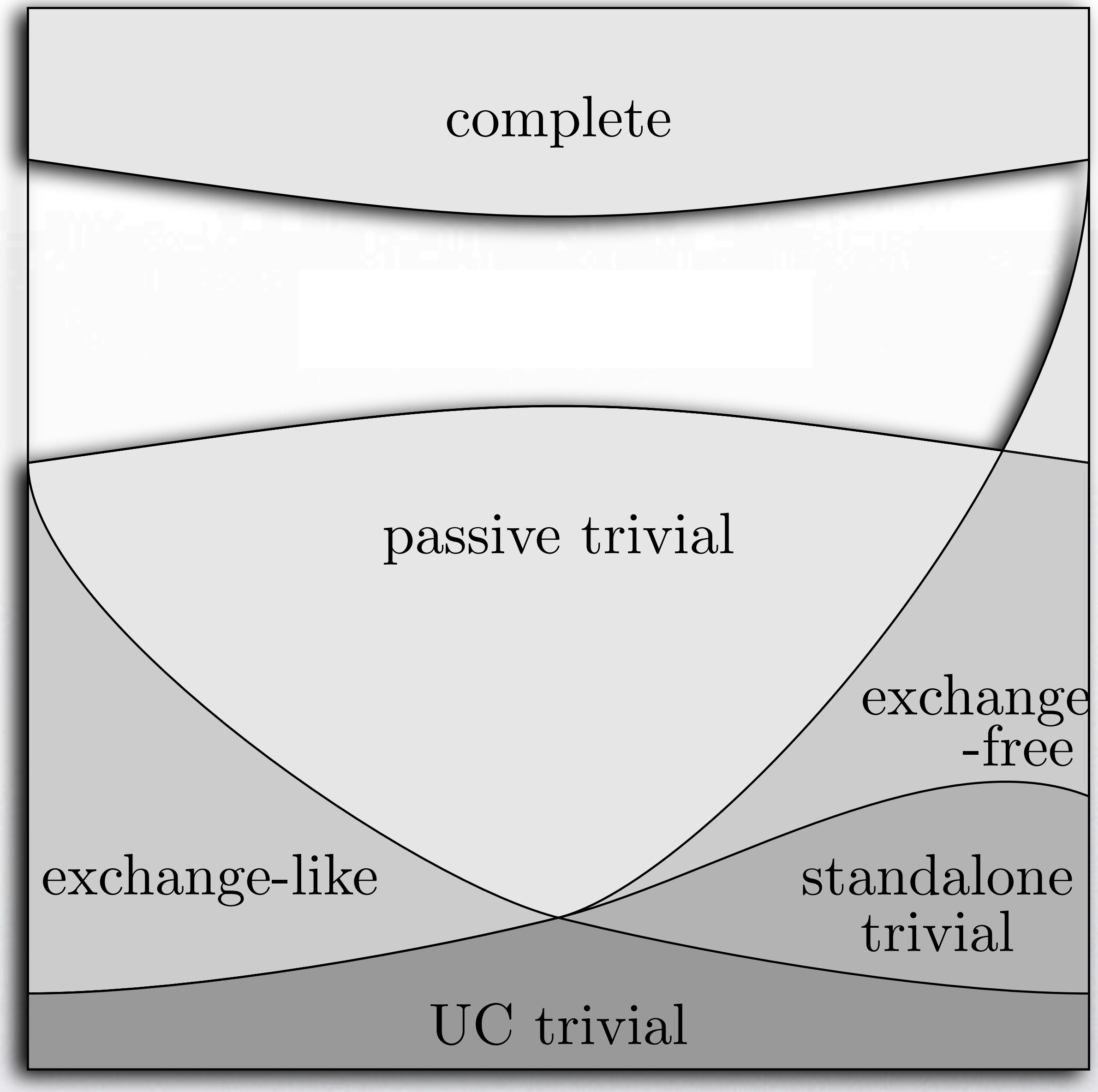
Functionalities

- A universe of functionalities: programs for a trusted party
  - Several constituent ideas: Zero-knowledge/simulatability [GMR85], Ideal/Real paradigm [GMW87], Relative-Resilience [B91], ..., Reactive Simulatability[PW01],UC security [C01]
- Motivates a Cryptographic Complexity Theory
- Reduction $F \sqsubseteq G$: F can be securely realized given G
  - Capturing extent of "cryptographic magic" in F, G
  - Strict (to capture fine distinctions), while remaining useful (to allow protocols): statistical (adaptive) UC security reduction

Functionalities

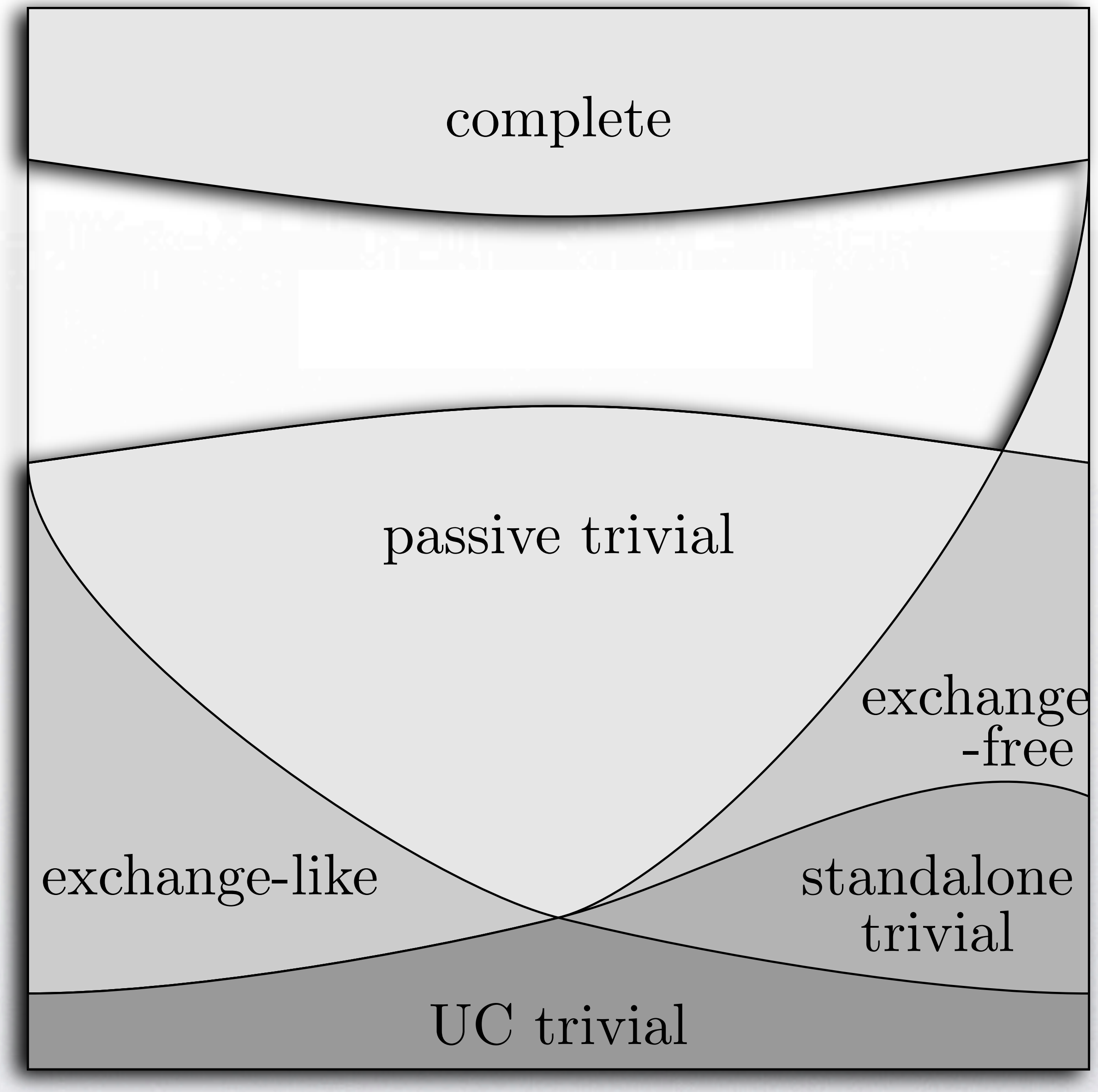
- A universe of functionalities: programs for a trusted party
  - Several constituent ideas: Zero-knowledge/simulatability [GMR85], Ideal/Real paradigm [GMW87], Relative-Resilience [B91], ..., Reactive Simulatability[PW01],UC security [C01]
- Motivates a Cryptographic Complexity Theory
- Reduction $F \sqsubseteq G$: F can be securely realized given G
  - Capturing extent of "cryptographic magic" in F, G
  - Strict (to capture fine distinctions), while remaining useful (to allow protocols): statistical (adaptive) UC security reduction
- Reductions represent cryptographic goals (cf. algorithmic goals)

Cryptographic Complexity



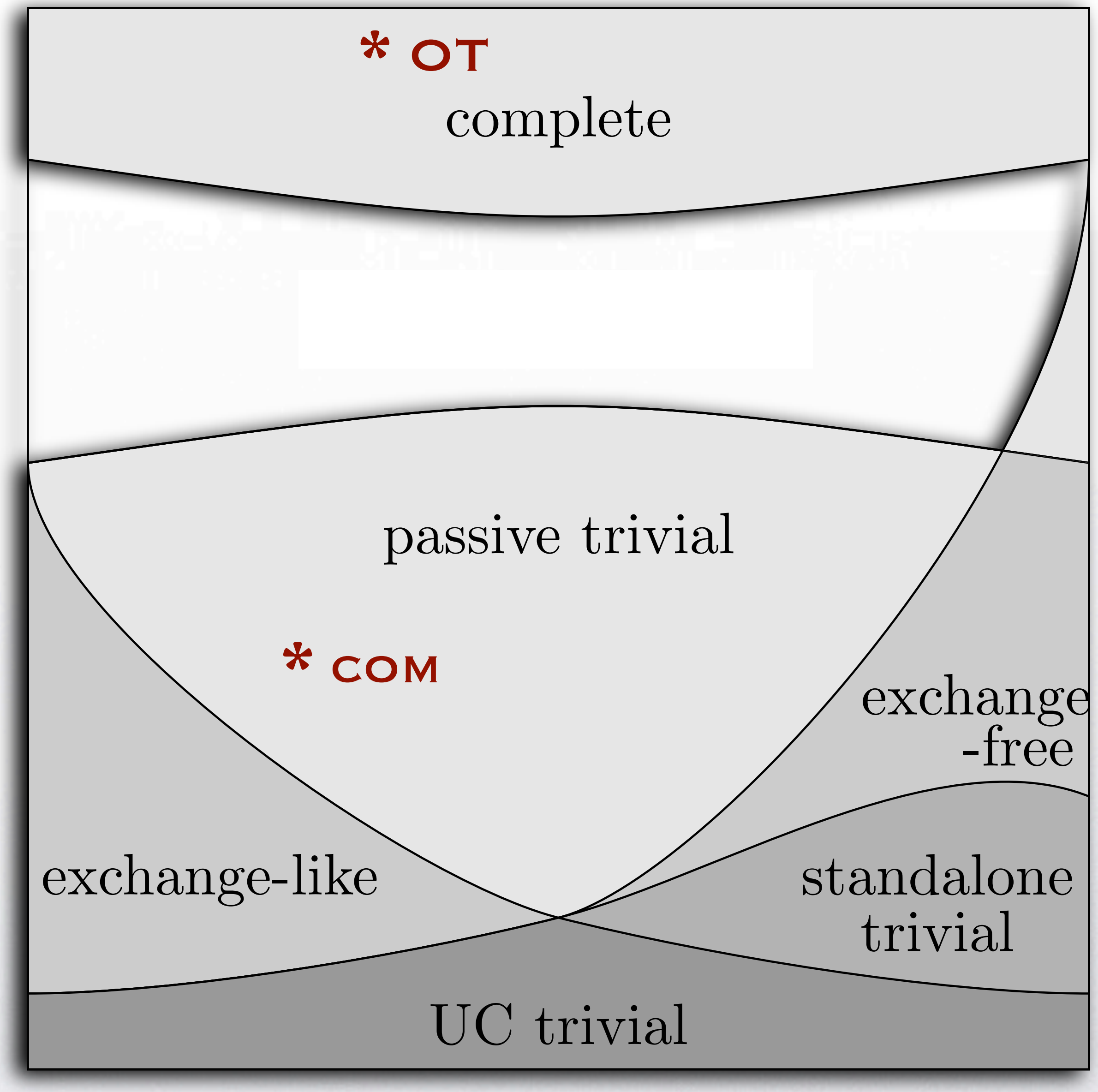
Cryptographic Complexity

- Complexity classes



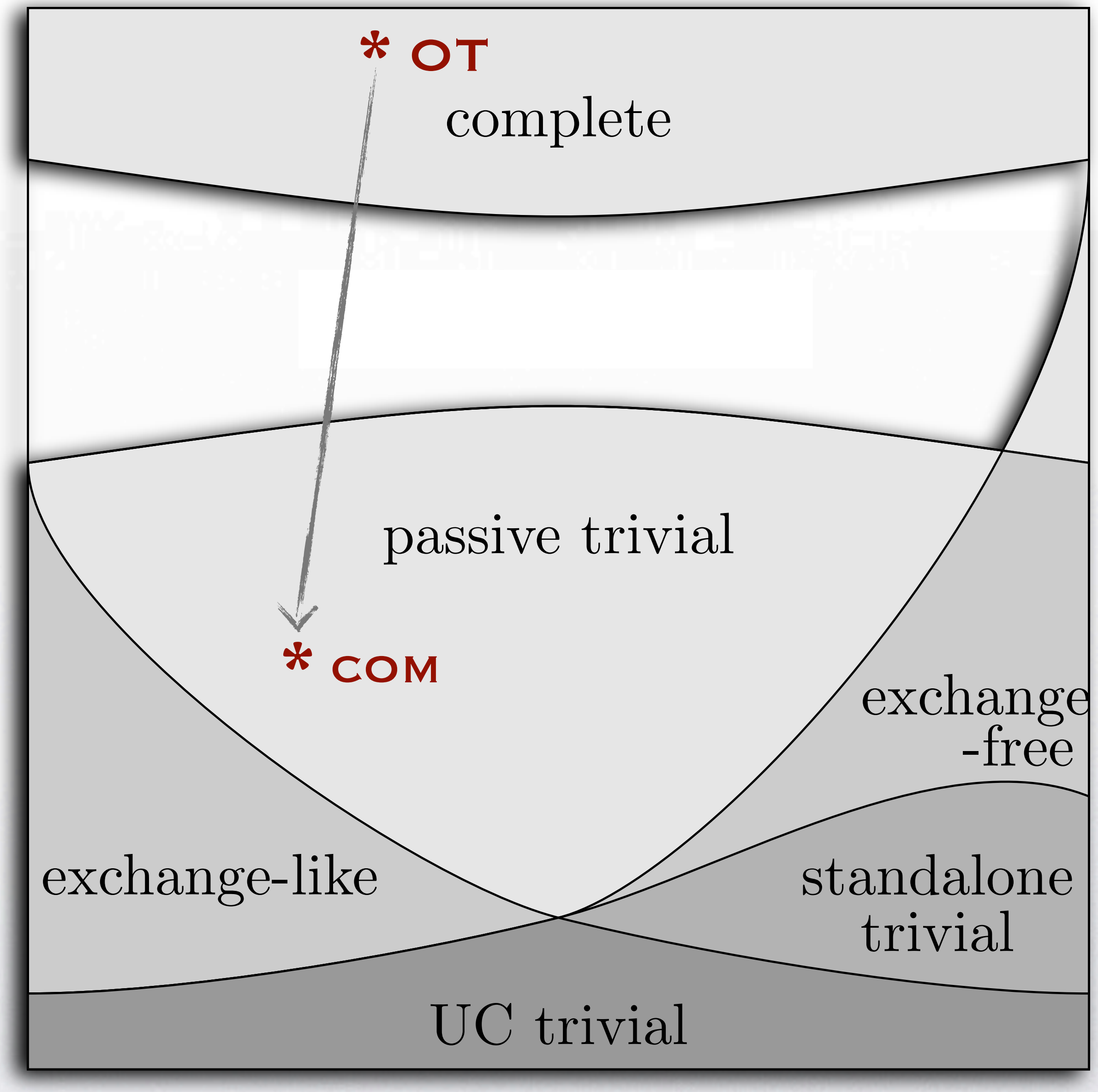
Cryptographic Complexity

- Complexity classes



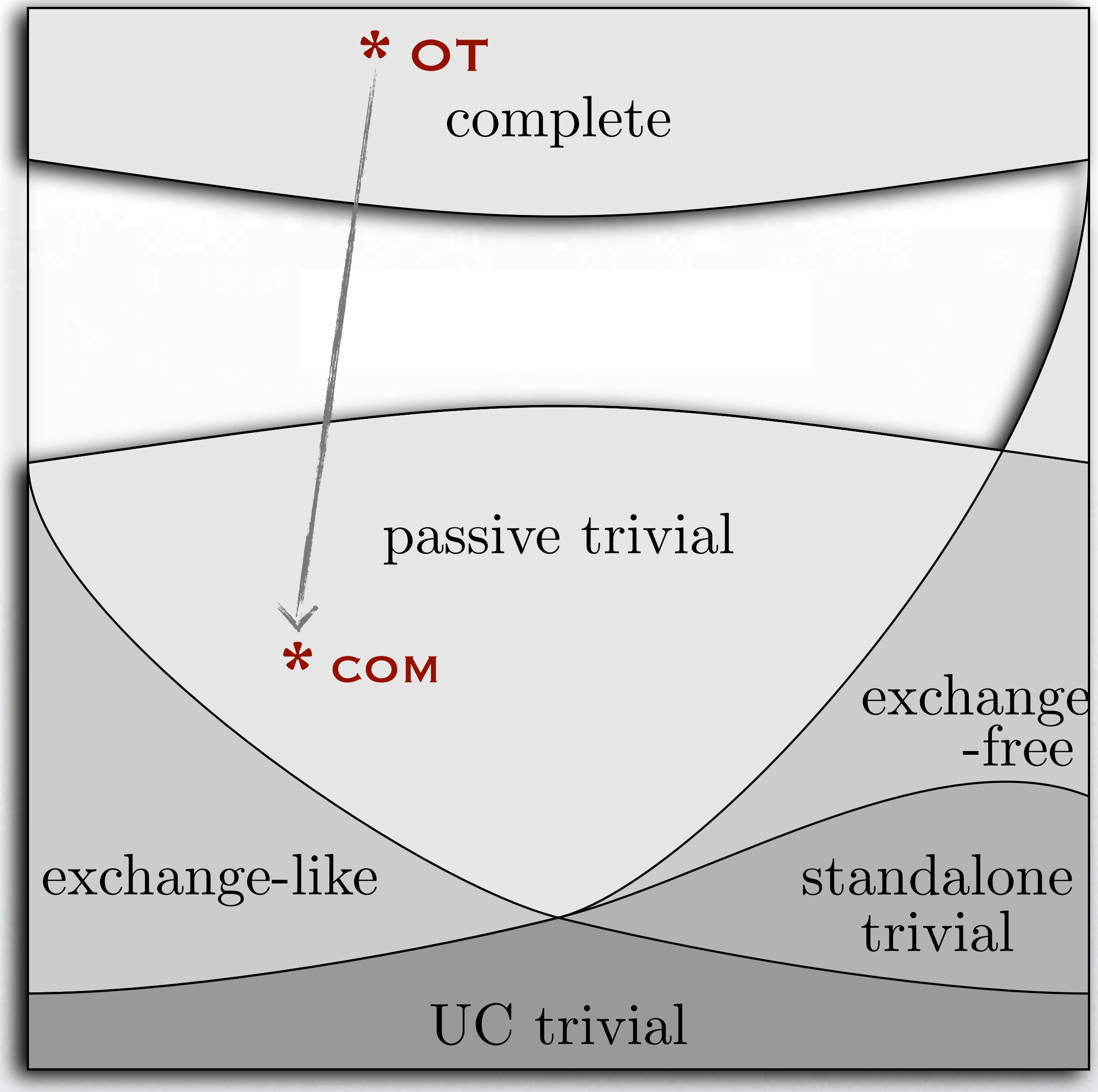
Cryptographic Complexity

- Complexity classes



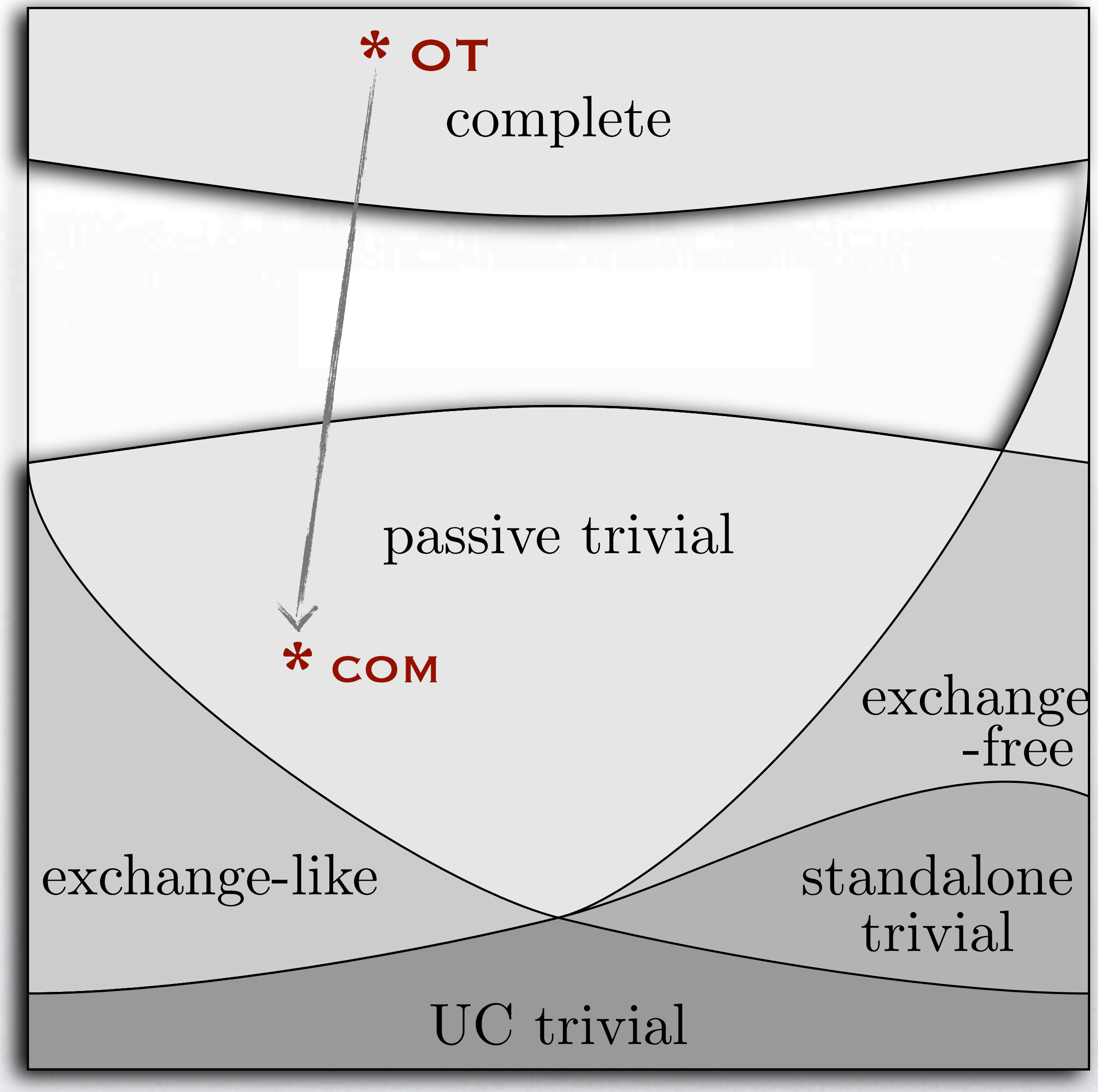
Cryptographic Complexity

- Complexity classes
  - Many results [K88,CK89,K89,K91,K00,KKMO00,..., PR08,KMQ08,KMQR09,MPR09, MPR10b]



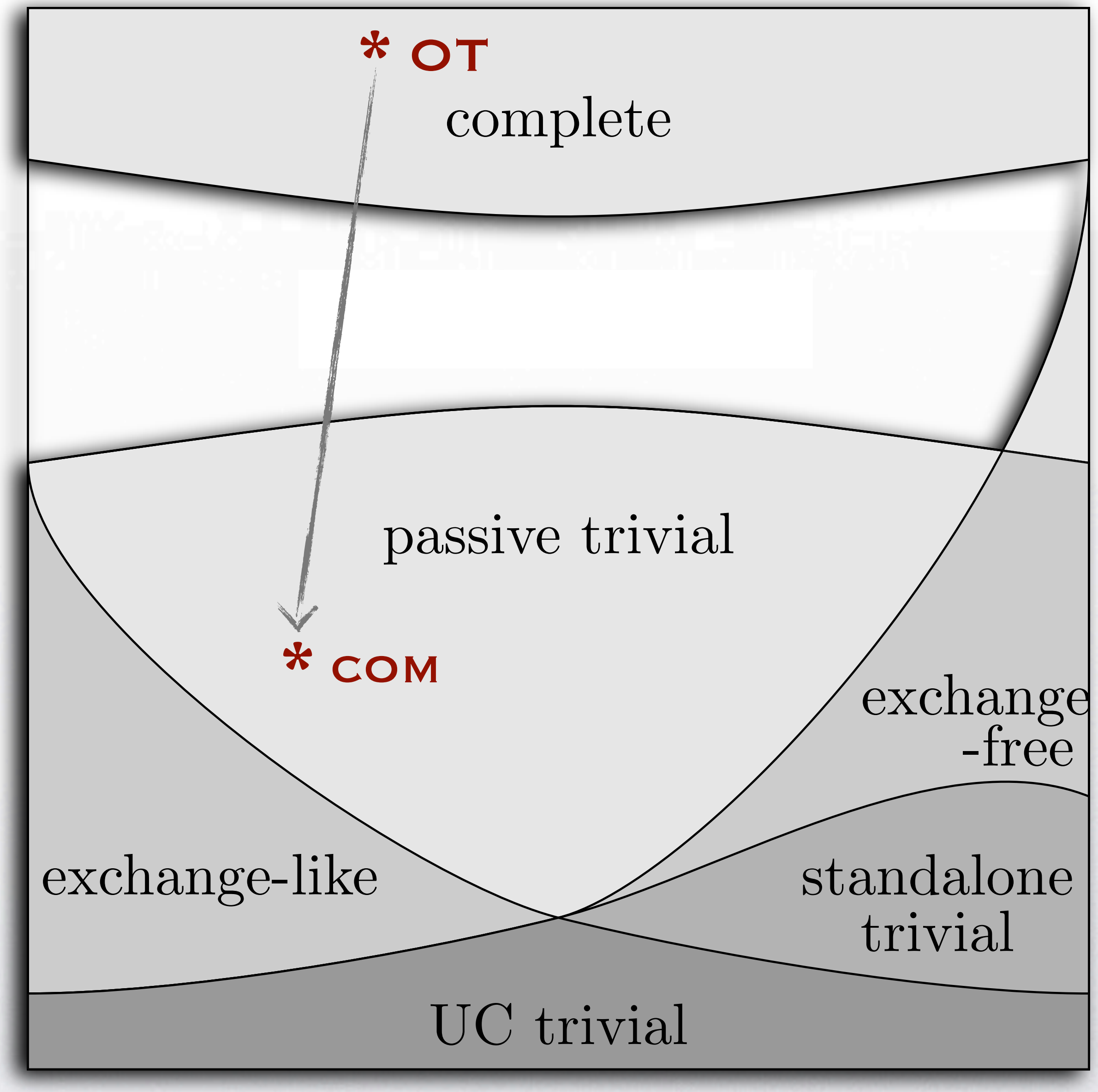
Cryptographic Complexity

- Complexity classes
  - Many results [K88,CK89,K89,K91,K00,KKMO00,..., PR08,KMQ08,KMQR09,MPR09, MPR10b]
  - e.g. "Passive Trivial"



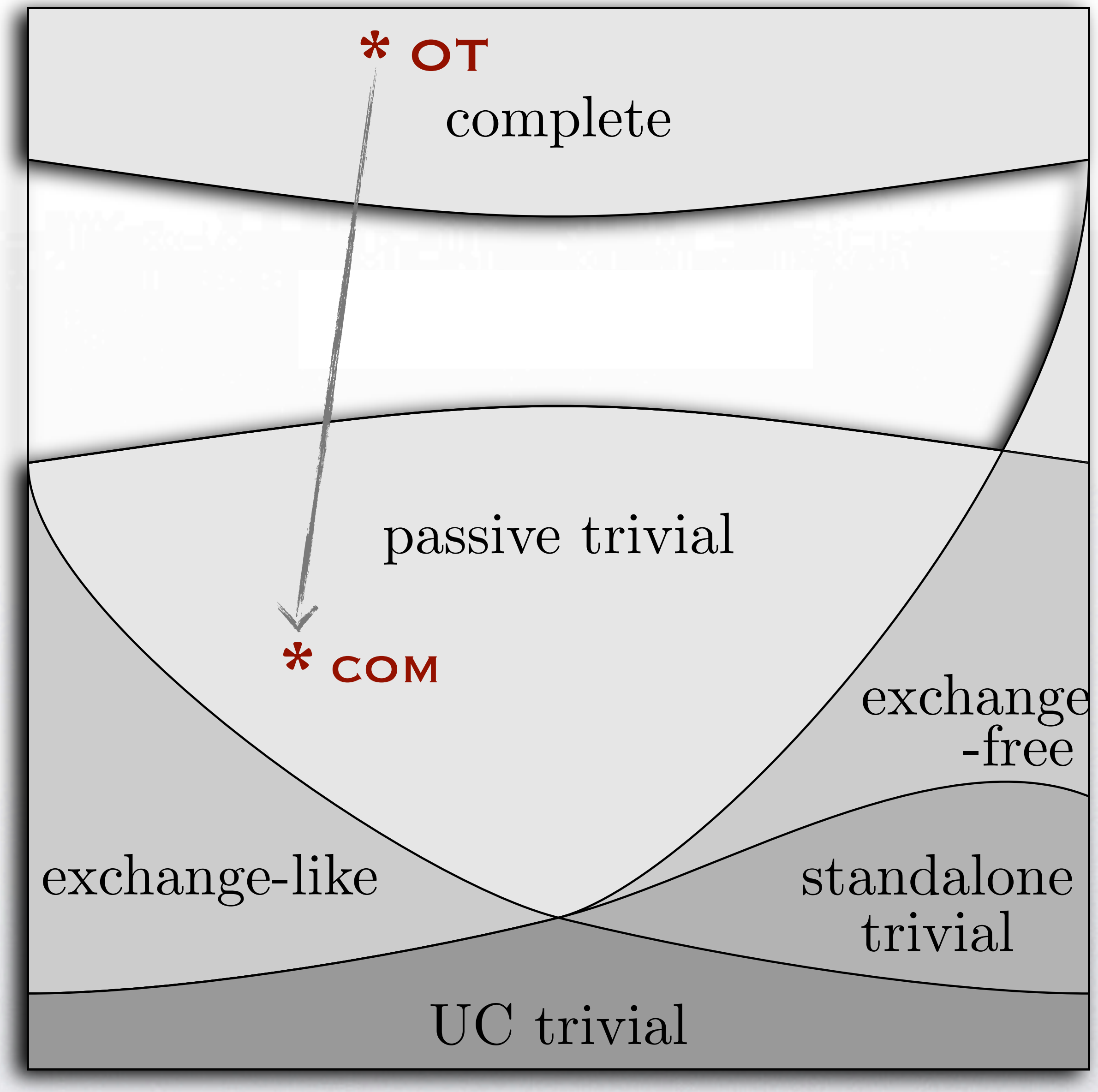
Cryptographic Complexity

- Complexity classes
  - Many results [K88,CK89,K89,K91,K00,KKMO00,..., PR08,KMQ08,KMQR09,MPR09, MPR10b]
  - e.g. "Passive Trivial"
    - F_{COM} is complete for PT, but no non-reactive F is [MPR09]



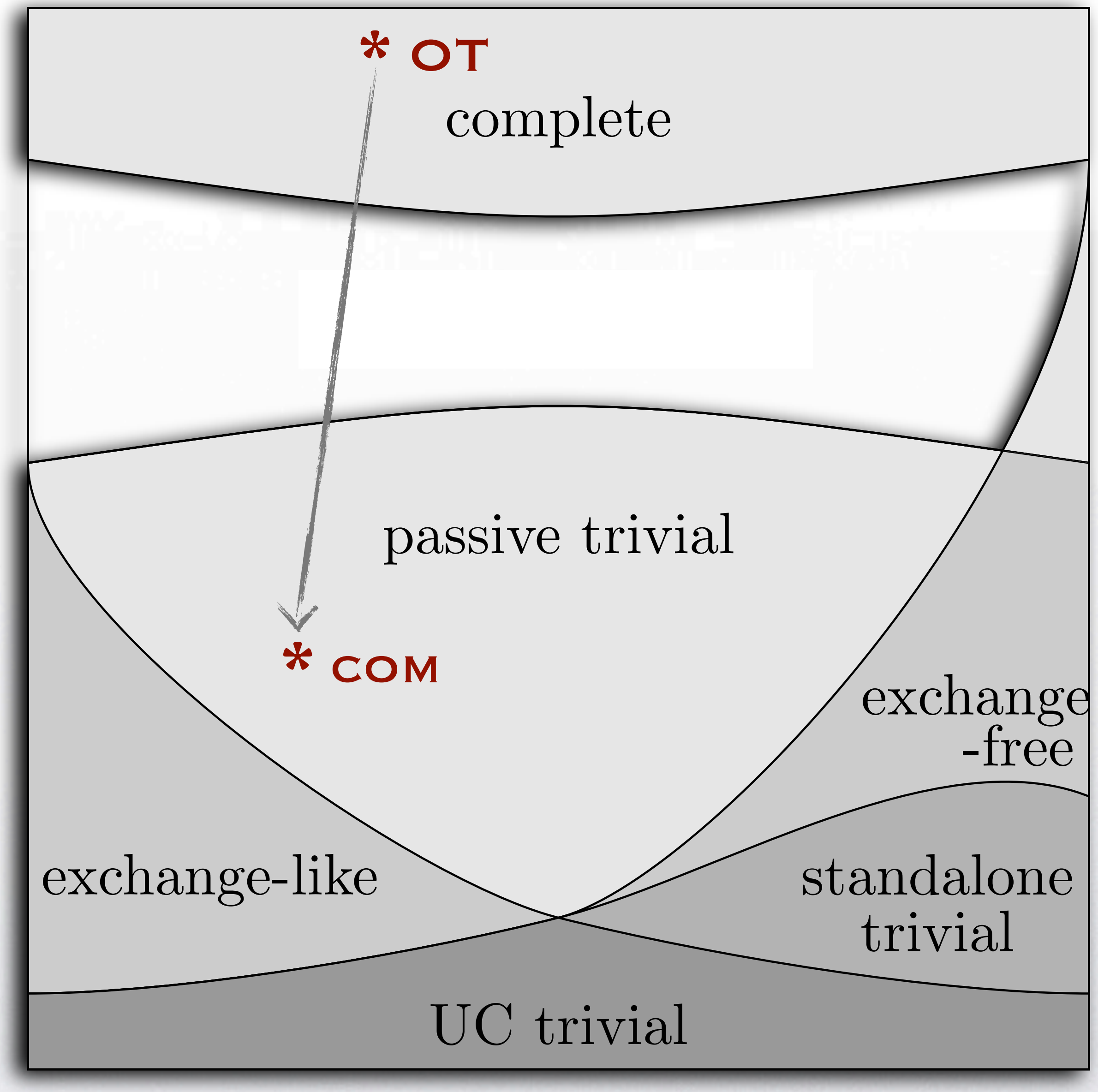
Cryptographic Complexity

- Complexity classes
  - Many results [K88,CK89,K89,K91,K00,KKMO00,..., PR08,KMQ08,KMQR09,MPR09, MPR10b]
  - e.g. "Passive Trivial"
    - F_{COM} is complete for PT, but no non-reactive F is [MPR09]
  - e.g. 3 reasons of non-triviality: hidden influence, commitment, simultaneity



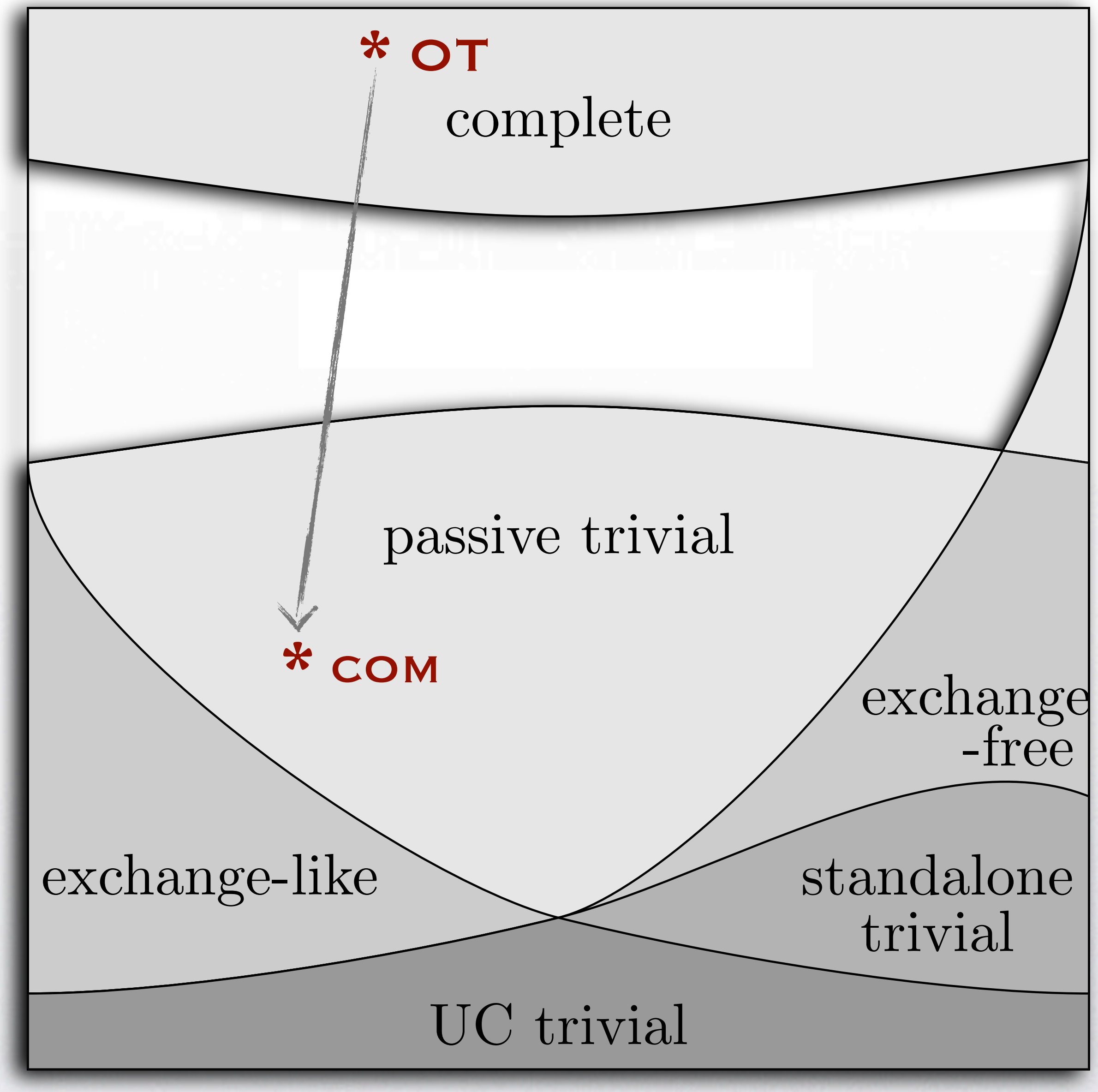
Cryptographic Complexity

- Complexity classes
  - Many results [K88,CK89,K89,K91,K00,KKMO00,..., PR08,KMQ08,KMQR09,MPR09, MPR10b]
  - e.g. "Passive Trivial"
    - F_{COM} is complete for PT, but no non-reactive F is [MPR09]
  - e.g. 3 reasons of non-triviality: hidden influence, commitment, simultaneity
    - Exchange-Like: essentially $F_{Exch}^{m\times n}$ [MPR10b]



Cryptographic Complexity

- Complexity classes
  - Many results [K88,CK89,K89,K91,K00,KKMO00,..., PR08,KMQ08,KMQR09,MPR09, MPR10b]
  - e.g. "Passive Trivial"
    - F_{COM} is complete for PT, but no non-reactive F is [MPR09]
  - e.g. 3 reasons of non-triviality: hidden influence, commitment, simultaneity
    - Exchange-Like: essentially $F_{\text{Exch}}^{m\times n}$ [MPR10b]
- Computationally unbounded setting



Intractability Assumptions

Intractability Assumptions

- No satisfactory framework so far

Intractability Assumptions

- No satisfactory framework so far
- We consider here a subset of assumptions as “inherent” to cryptographic goals

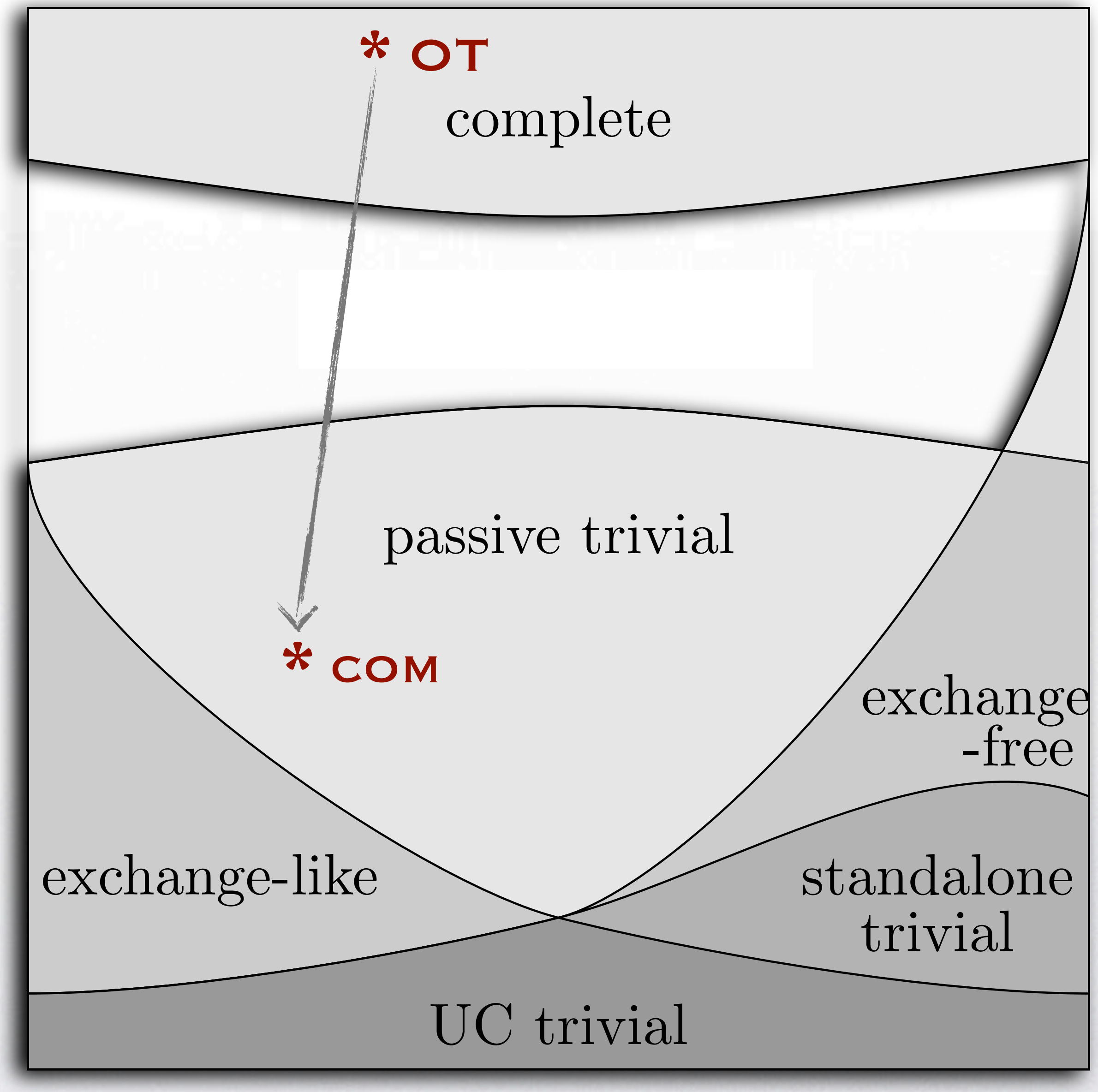
Intractability Assumptions

- No satisfactory framework so far
- We consider here a subset of assumptions as “inherent” to cryptographic goals
  - Plan: Leverage cryptographic complexity of functionalities to chart the landscape of intractability assumptions

Intractability Assumptions

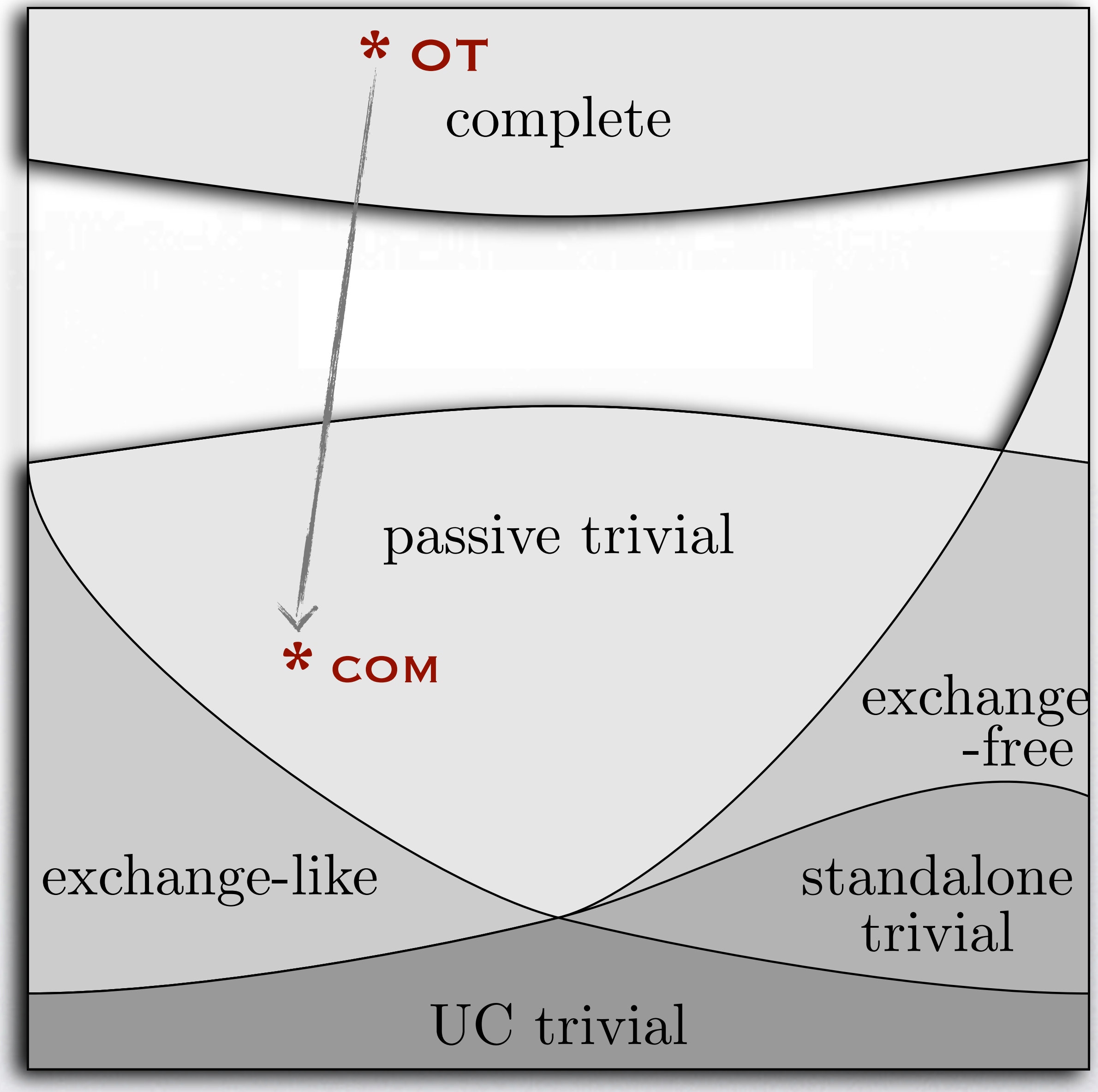
- No satisfactory framework so far
- We consider here a subset of assumptions as “inherent” to cryptographic goals
  - Plan: Leverage cryptographic complexity of functionalities to chart the landscape of intractability assumptions
  - Universe of assumptions: $F \sqsubseteq G$ in the computationally bounded setting

Assumptions: F ⊑ G



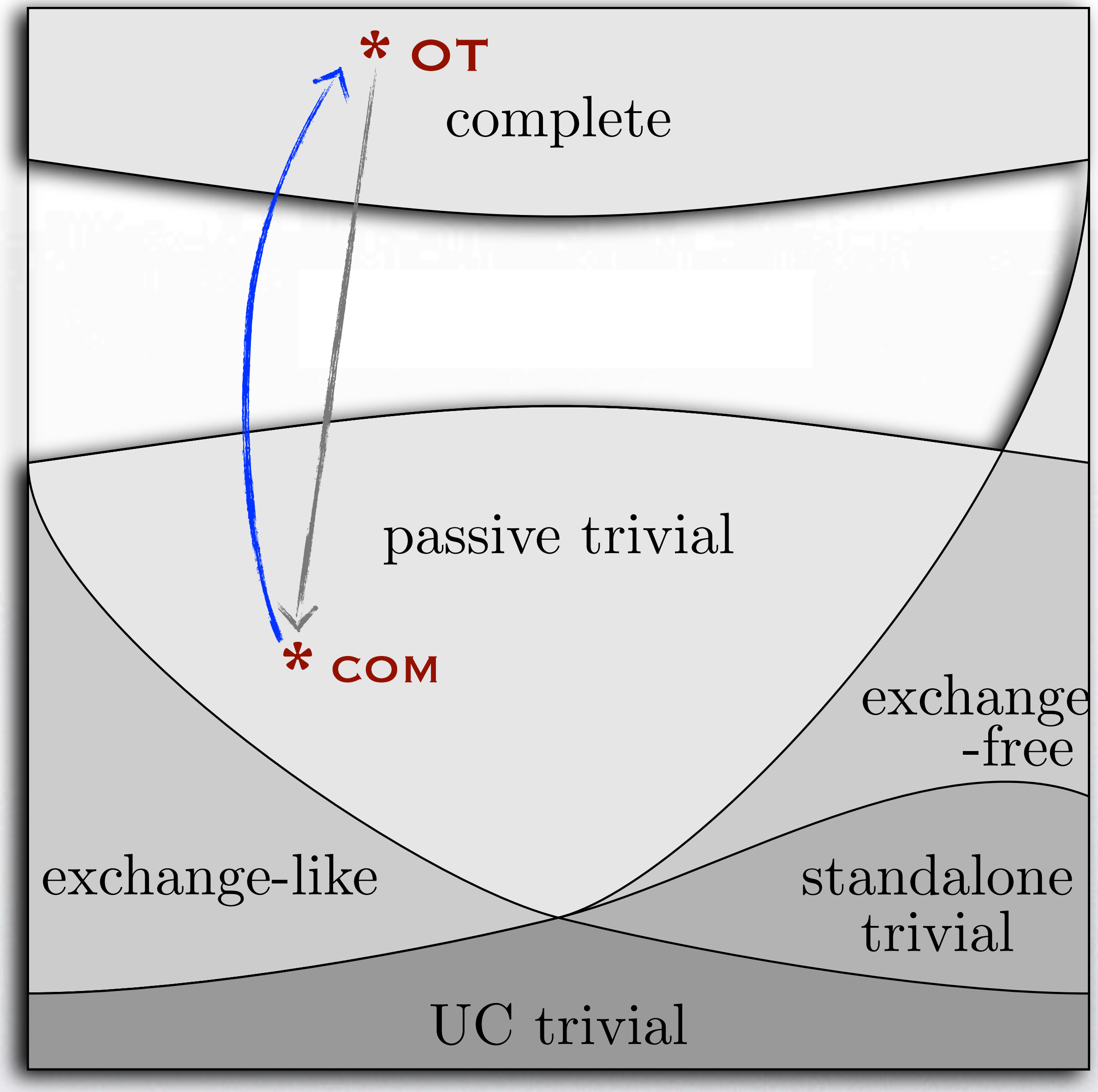
Assumptions: F ⊑ G

- Reductions which are not true in the computationally unbounded setting



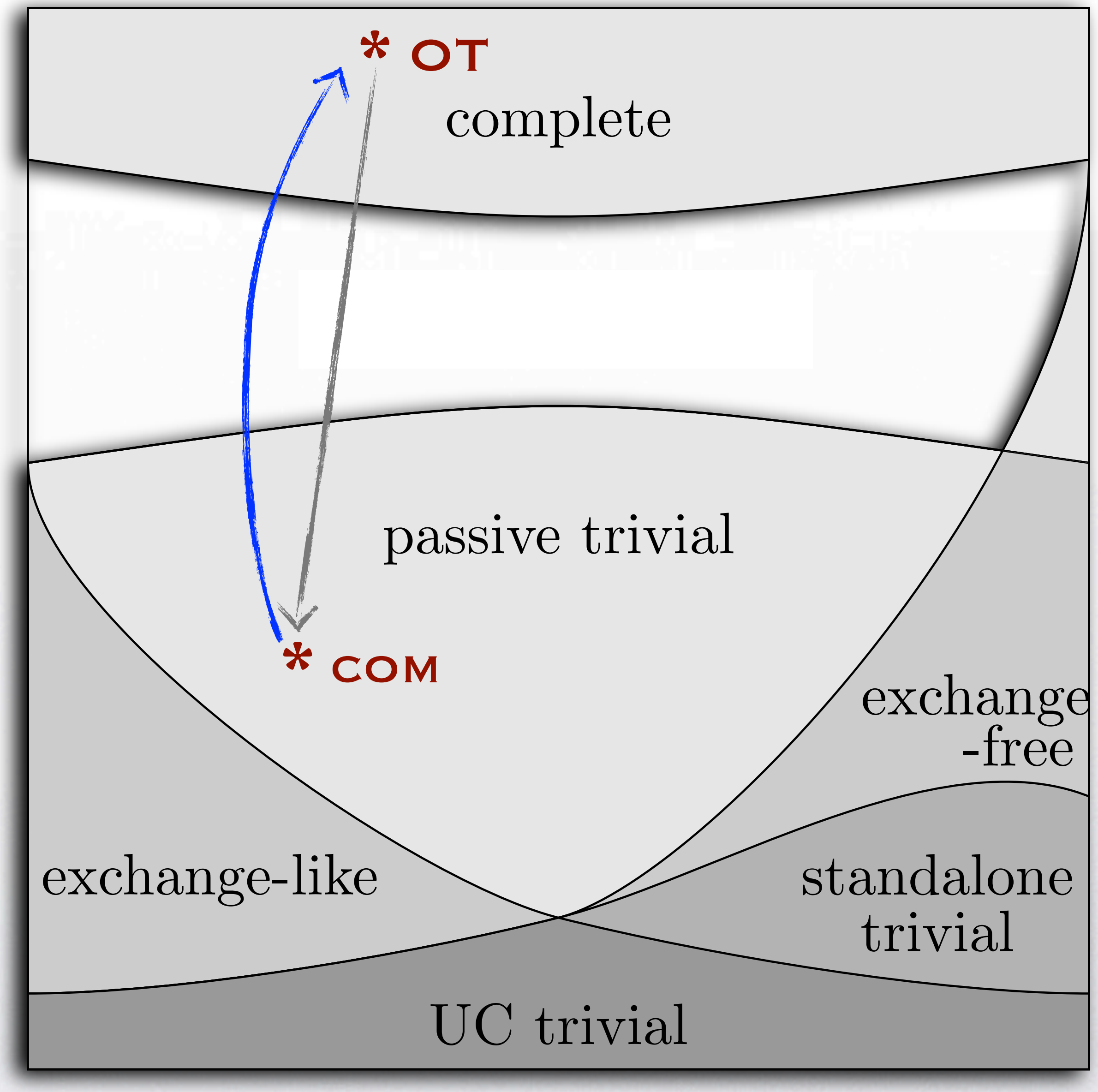
Assumptions: F ⊑ G

- Reductions which are not true in the computationally unbounded setting



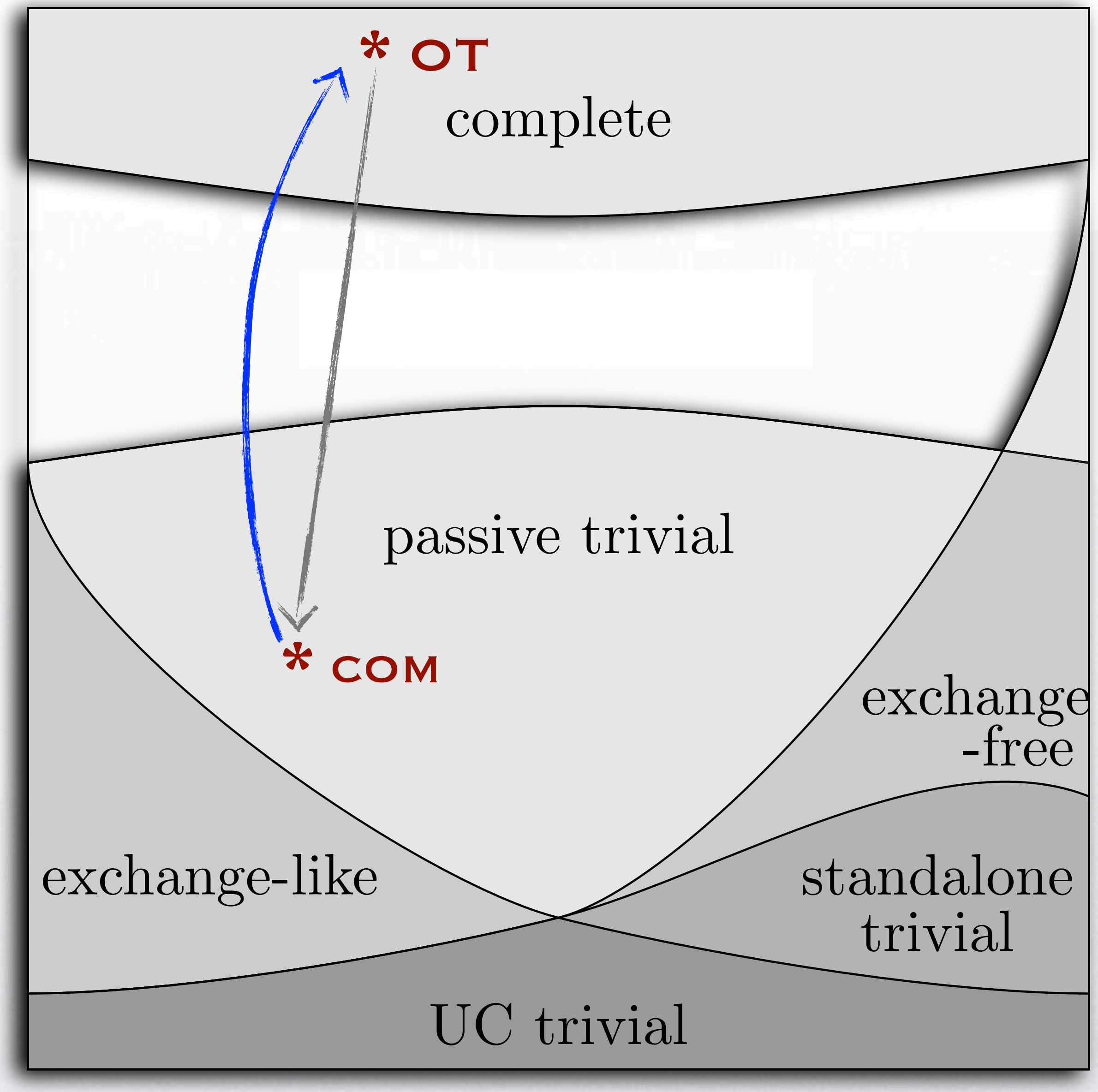
Assumptions: F $\sqsubseteq$ G

- Reductions which are not true in the computationally unbounded setting
- Assumption that it holds in the PPT setting



Assumptions: F ⊑ G

- Reductions which are not true in the computationally unbounded setting
- Assumption that it holds in the PPT setting
- Can consider multiple notions of ⊑. Here, UC security against active (static) adversaries.



Intractability Assumptions

Intractability Assumptions

- Assumptions: $F \sqsubseteq G$

Intractability Assumptions

- Assumptions: F⊑G
    - Maximal assumption(s)?

Intractability Assumptions

- Assumptions: $F \sqsubseteq G$
  - Maximal assumption(s)?
  - Minimal assumption(s)?

Intractability Assumptions

- Assumptions: $F \sqsubseteq G$
  - Maximal assumption(s)?
  - Minimal assumption(s)?
  - How many distinct assumptions?

Intractability Assumptions

- Assumptions: F$\sqsubseteq$G
    - Maximal assumption(s)?
    - Minimal assumption(s)?
    - How many distinct assumptions?
- And identify equivalent "traditional" assumptions like OWF

Intractability Assumptions

- Assumptions: F$\sqsubseteq$G
  - Maximal assumption(s)?
  - Minimal assumption(s)?
  - How many distinct assumptions?
- And identify equivalent "traditional" assumptions like OWF
- Contrast with deriving general assumptions to abstract specific algebraic/number-theoretic assumptions

Intractability Assumptions

- Assumptions: $F \sqsubseteq G$
  - Maximal assumption(s)?
  - Minimal assumption(s)?
  - How many distinct assumptions?
- And identify equivalent "traditional" assumptions like OWF
- Contrast with deriving general assumptions to abstract specific algebraic/number-theoretic assumptions
  - Many standard general assumptions (like OWP) may not appear in our universe

Results

Results

- Every assumption $F \sqsubseteq G$ (for 2-party F, G) that we classify is *equivalent* to existence of one-way functions (OWF) or that of semi-honest OT protocols (shOT)

Results

- Every assumption F⊑G (for 2-party F, G) that we classify is *equivalent* to existence of one-way functions (OWF) or that of semi-honest OT protocols (shOT)

A protocol for OT that is secure against "semi-honest" adversaries (equivalently, against "stand-alone" adversaries)

Results

- Every assumption F⊑G (for 2-party F, G) that we classify is *equivalent* to existence of one-way functions (OWF) or that of semi-honest OT protocols (shOT)
  - Significance of "Minicrypt" and "Cryptomania"

A protocol for OT that is secure against "semi-honest" adversaries (equivalently, against "stand-alone" adversaries)

Results

- Every assumption $F \sqsubseteq G$ (for 2-party F, G) that we classify is *equivalent* to existence of one-way functions (OWF) or that of semi-honest OT protocols (shOT)
  - Significance of “Minicrypt” and “Cryptomania”
  - In this work: $F \sqsubseteq G \Rightarrow$ OWF/shOT

A protocol for OT that is secure against “semi-honest” adversaries (equivalently, against “stand-alone” adversaries)

Results

- Every assumption $F \sqsubseteq G$ (for 2-party F, G) that we classify is *equivalent* to existence of one-way functions (OWF) or that of semi-honest OT protocols (shOT)
  - Significance of "Minicrypt" and "Cryptomania"
  - In this work: $F \sqsubseteq G \Rightarrow$ OWF/shOT
    - Other direction from companion work [MPR10b]

A protocol for OT that is secure against "semi-honest" adversaries (equivalently, against "stand-alone" adversaries)

Results

- Every assumption F⊑G (for 2-party F, G) that we classify is *equivalent* to existence of one-way functions (OWF) or that of semi-honest OT protocols (shOT)
  - Significance of "Minicrypt" and "Cryptomania"
  - In this work: F⊑G ⇒ OWF/shOT
    - Other direction from companion work [MPR10b]
      - In particular shOT is *the maximal assumption*

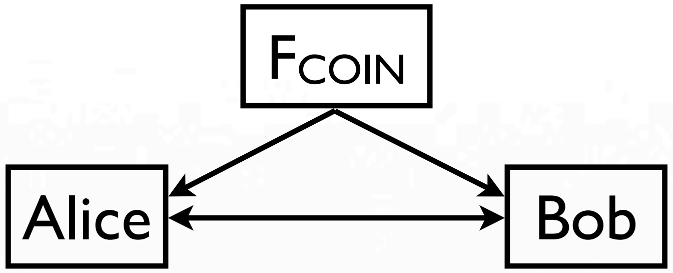
A protocol for OT that is secure against "semi-honest" adversaries (equivalently, against "stand-alone" adversaries)

An Example

- $F_{\text{Exch}} \sqsubseteq F_{\text{Coin}} \Rightarrow$ shOT

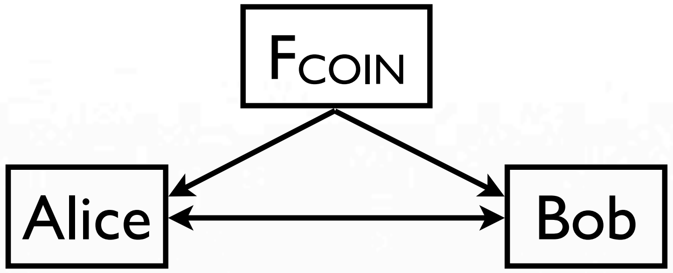
An Example

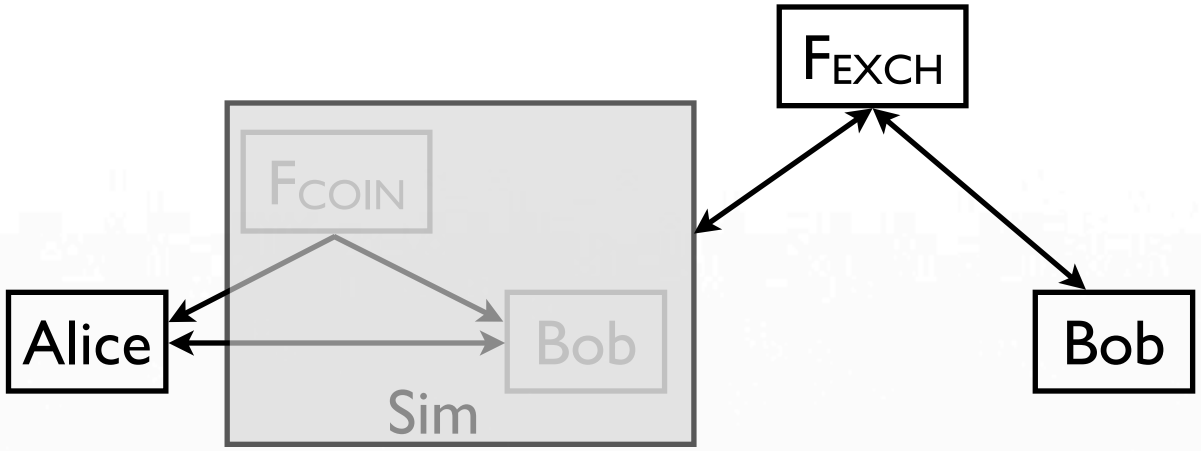
- $F_{\text{Exch}} \sqsubseteq F_{\text{Coin}} \Rightarrow \text{shOT}$



An Example

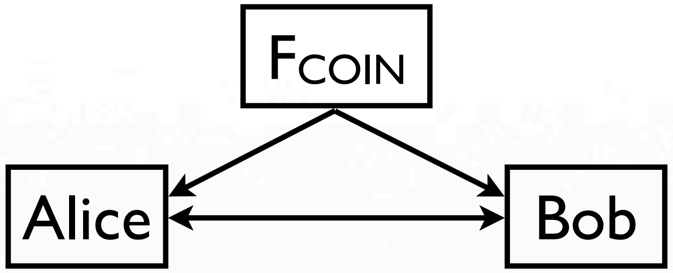
- $F_{\text{Exch}} \sqsubseteq F_{\text{Coin}} \Rightarrow \text{shOT}$

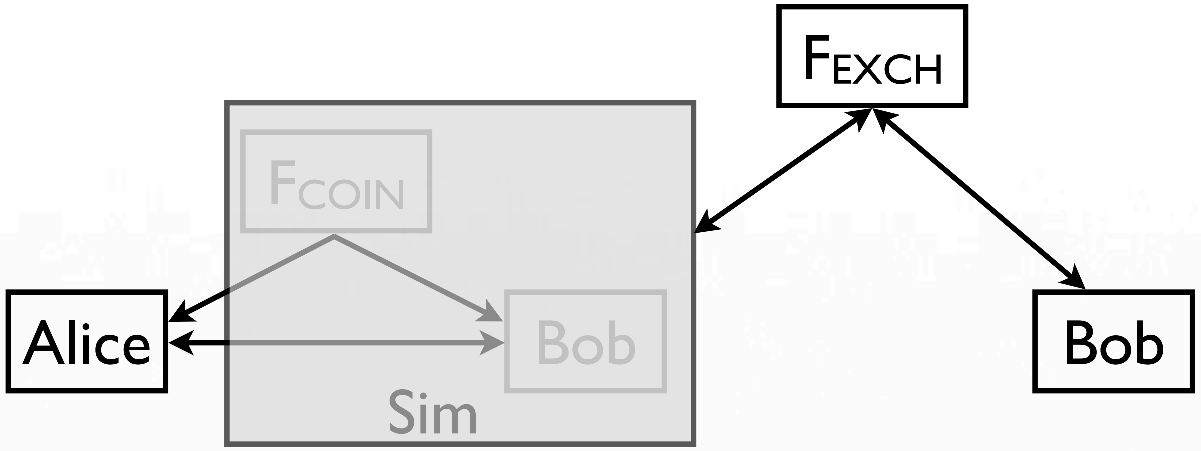




An Example

- $F_{\text{Exch}} \sqsubseteq F_{\text{Coin}} \Rightarrow$ shOT

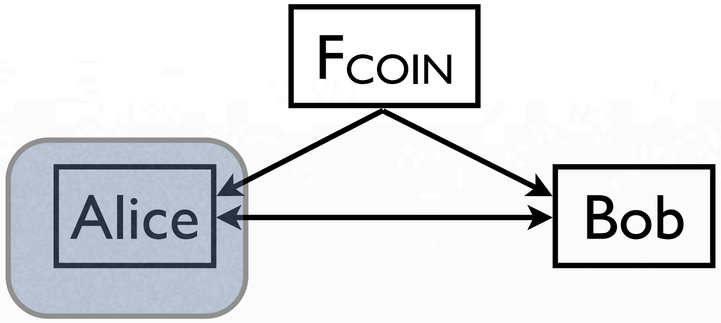


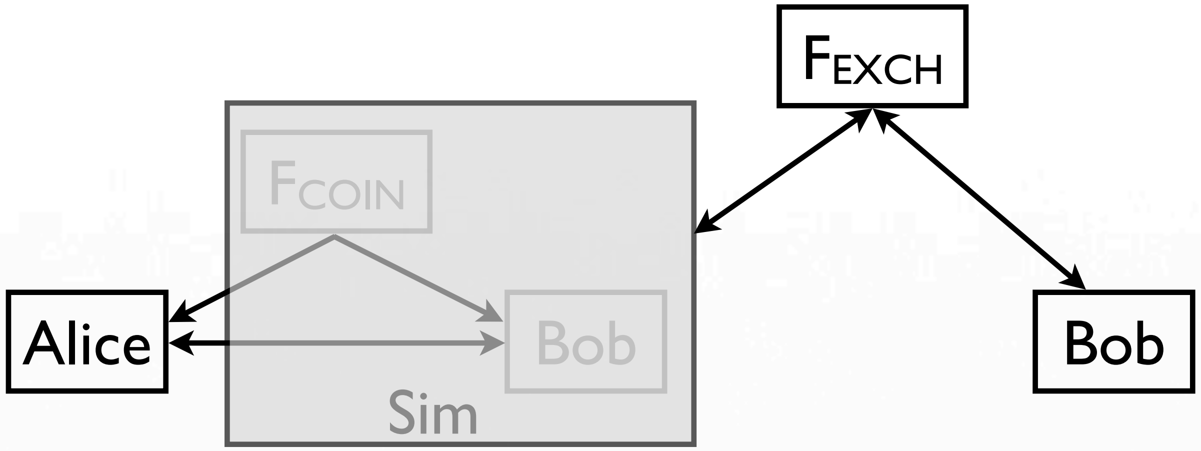


- Basic idea for an shOT protocol:

An Example

- $F_{Exch} \sqsubseteq F_{Coin} \Rightarrow$ shOT

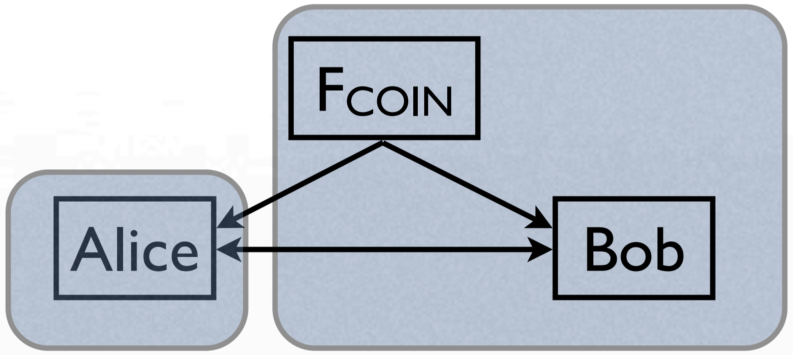


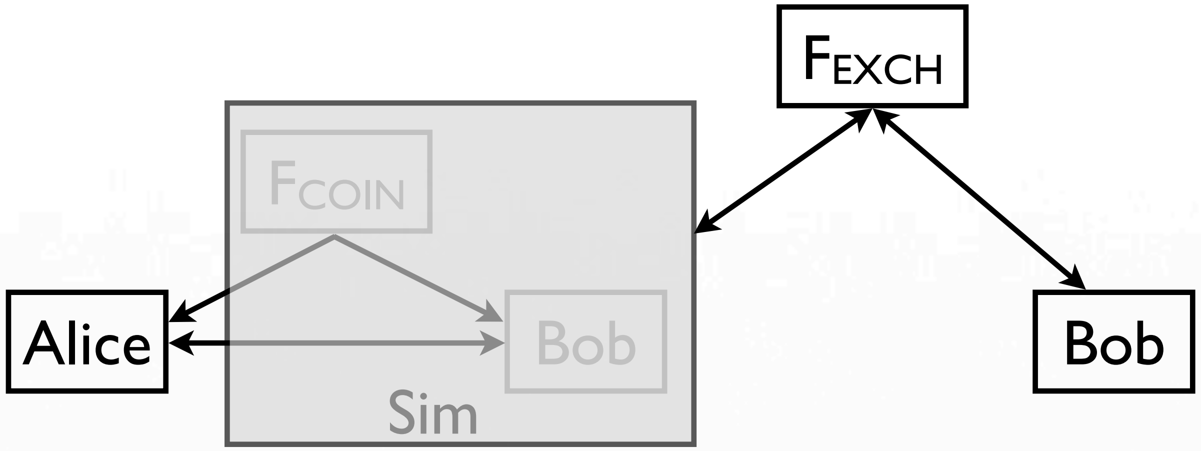


- Basic idea for an shOT protocol:
  - Sender runs F_{Exch} protocol (say, as Alice)

An Example

- $F_{Exch} \sqsubseteq F_{Coin} \Rightarrow$ shOT

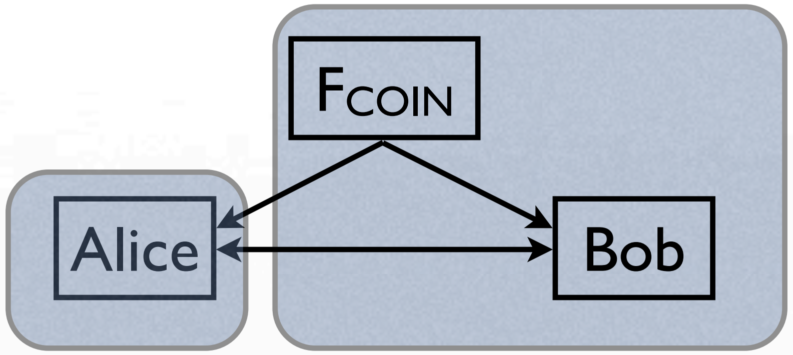


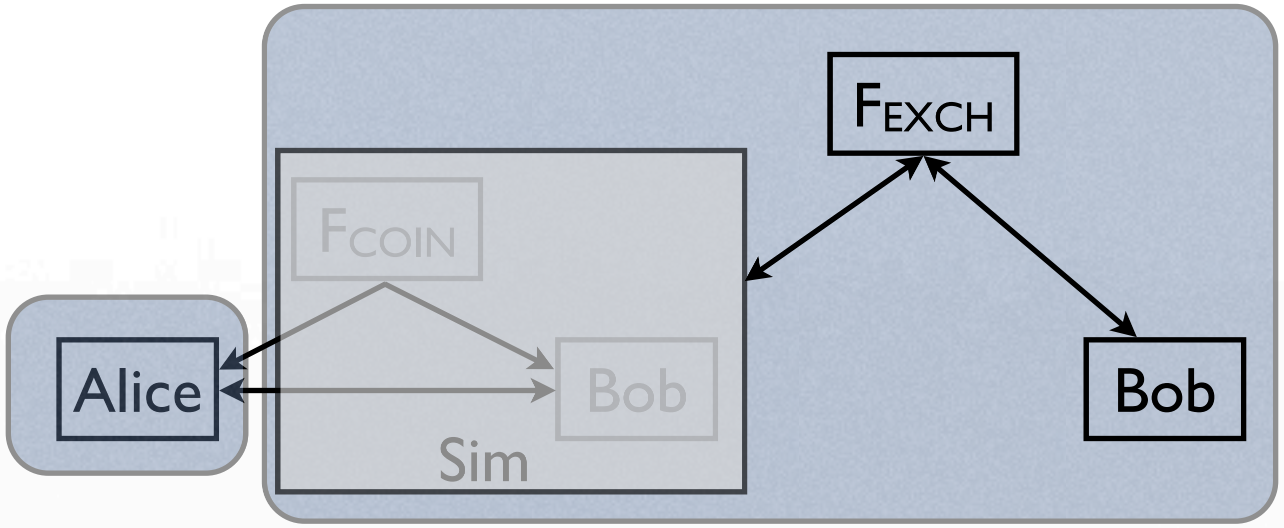


- Basic idea for an shOT protocol:
  - Sender runs F_{Exch} protocol (say, as Alice)
  - Receiver will run either the F_{Exch} protocol (playing F_{Coin} itself), or the simulator for that protocol. Sender cannot distinguish between the two.

An Example

- $F_{Exch} \sqsubseteq F_{Coin} \Rightarrow$ shOT

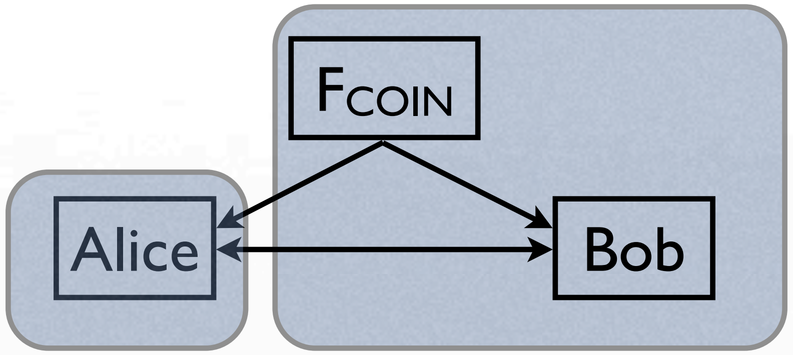


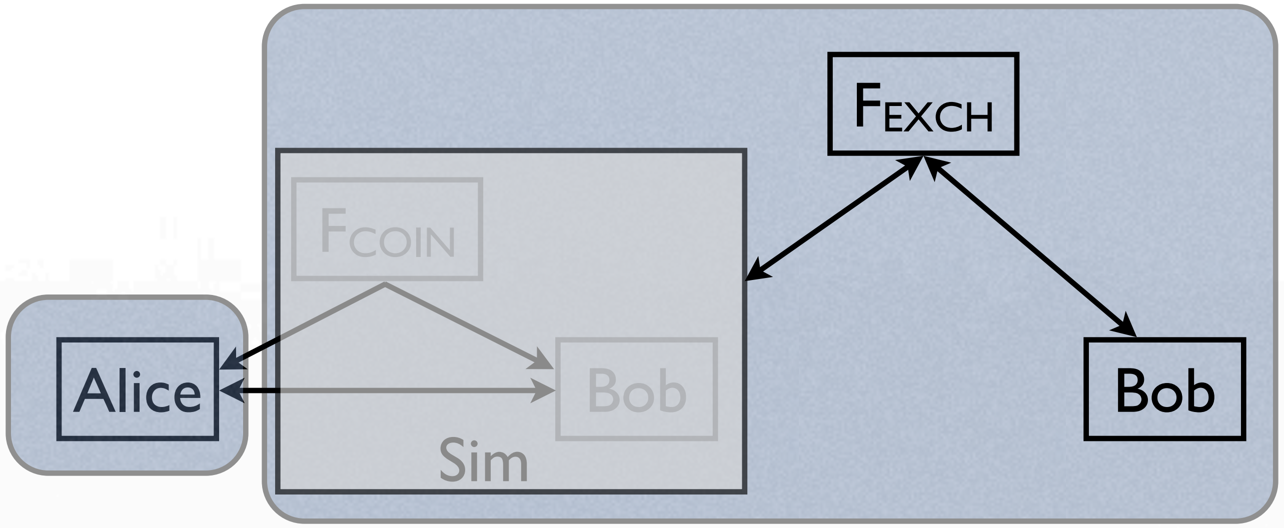


- Basic idea for an shOT protocol:
  - Sender runs F_{Exch} protocol (say, as Alice)
  - Receiver will run either the F_{Exch} protocol (playing F_{Coin} itself), or the simulator for that protocol. Sender cannot distinguish between the two.

An Example

- $F_{Exch} \sqsubseteq F_{Coin} \Rightarrow$ shOT

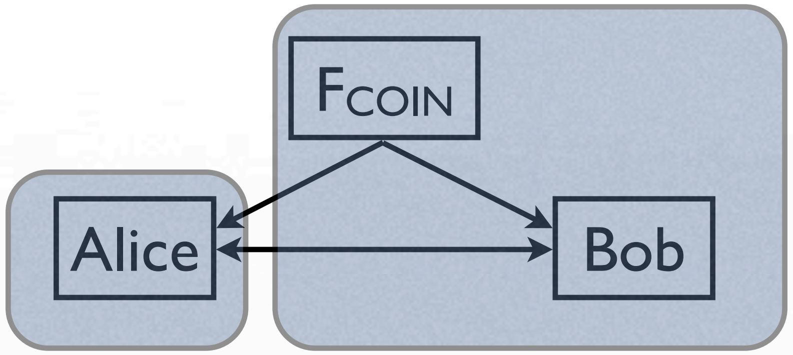


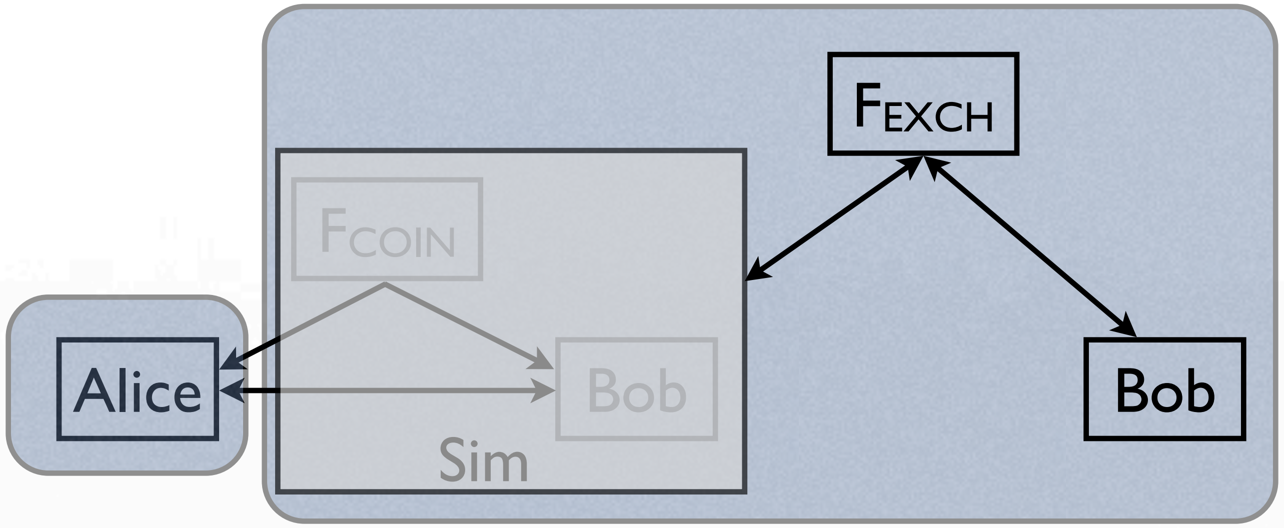


- Basic idea for an shOT protocol:
  - Sender runs F_{Exch} protocol (say, as Alice)
  - Receiver will run either the F_{Exch} protocol (playing F_{Coin} itself), or the simulator for that protocol. Sender cannot distinguish between the two.
  - Truncate the execution at a random round

An Example

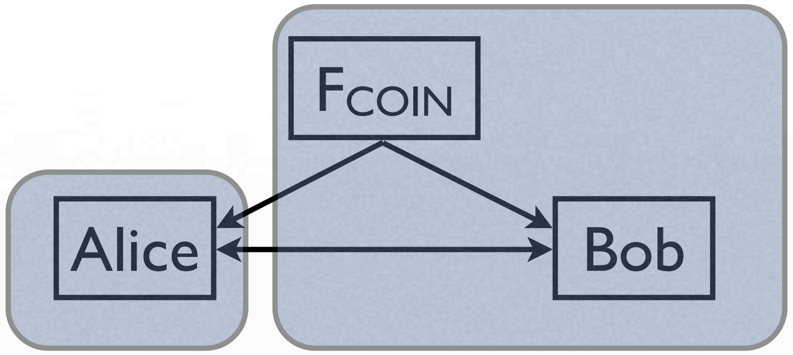
- $F_{Exch} \sqsubseteq F_{Coin} \Rightarrow$ shOT

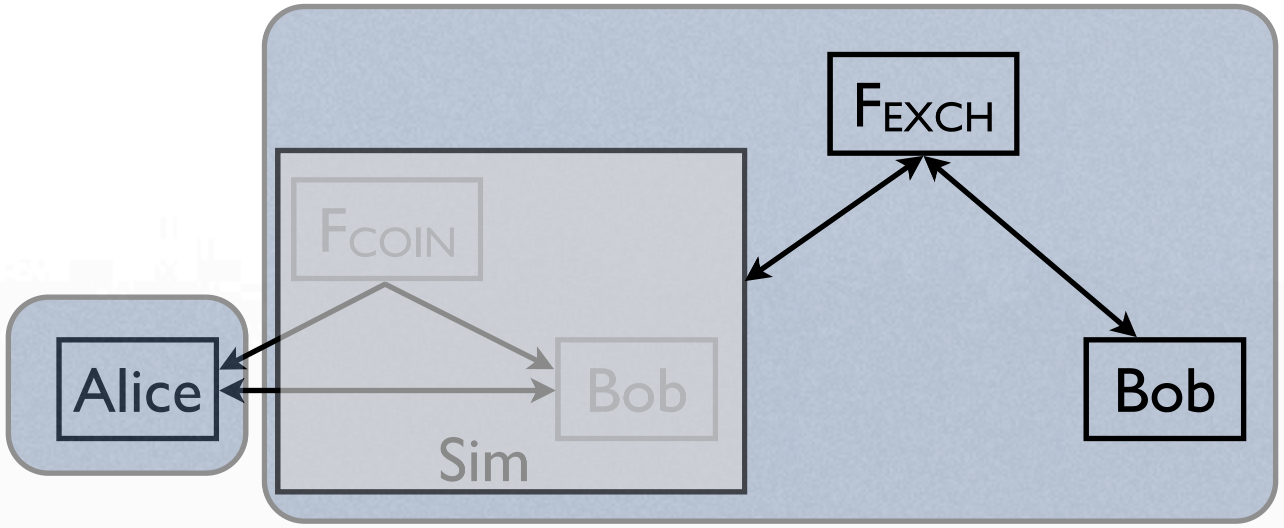




An Example

- $F_{Exch} \sqsubseteq F_{Coin} \Rightarrow$ shOT

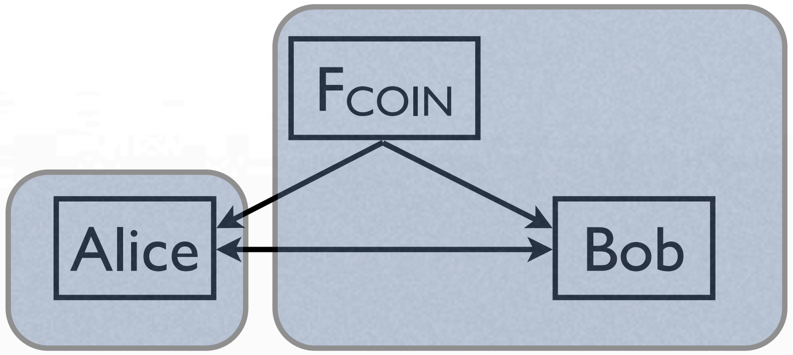


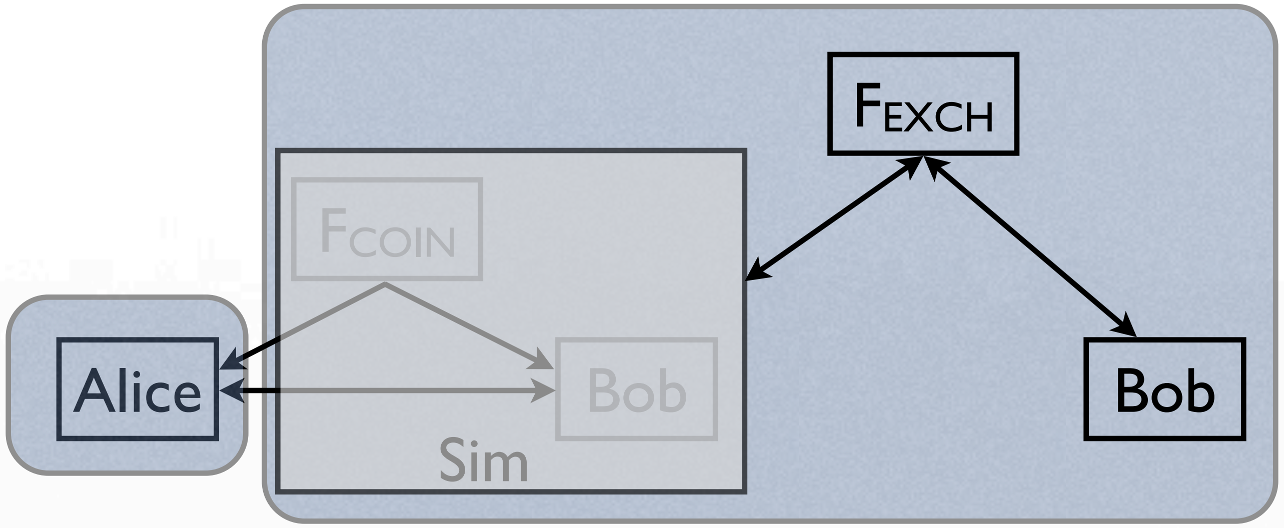


- Can argue: in the F_{Exch} protocol, the expected round in the simulation at which simulator for corrupt Alice extracts her input is before Bob learns it in the real execution (or with Alice/Bob reversed). (Uses the fact that F_{Coin} cannot be used to communicate.)

An Example

- $F_{\text{Exch}} \sqsubseteq F_{\text{Coin}} \Rightarrow$ shOT

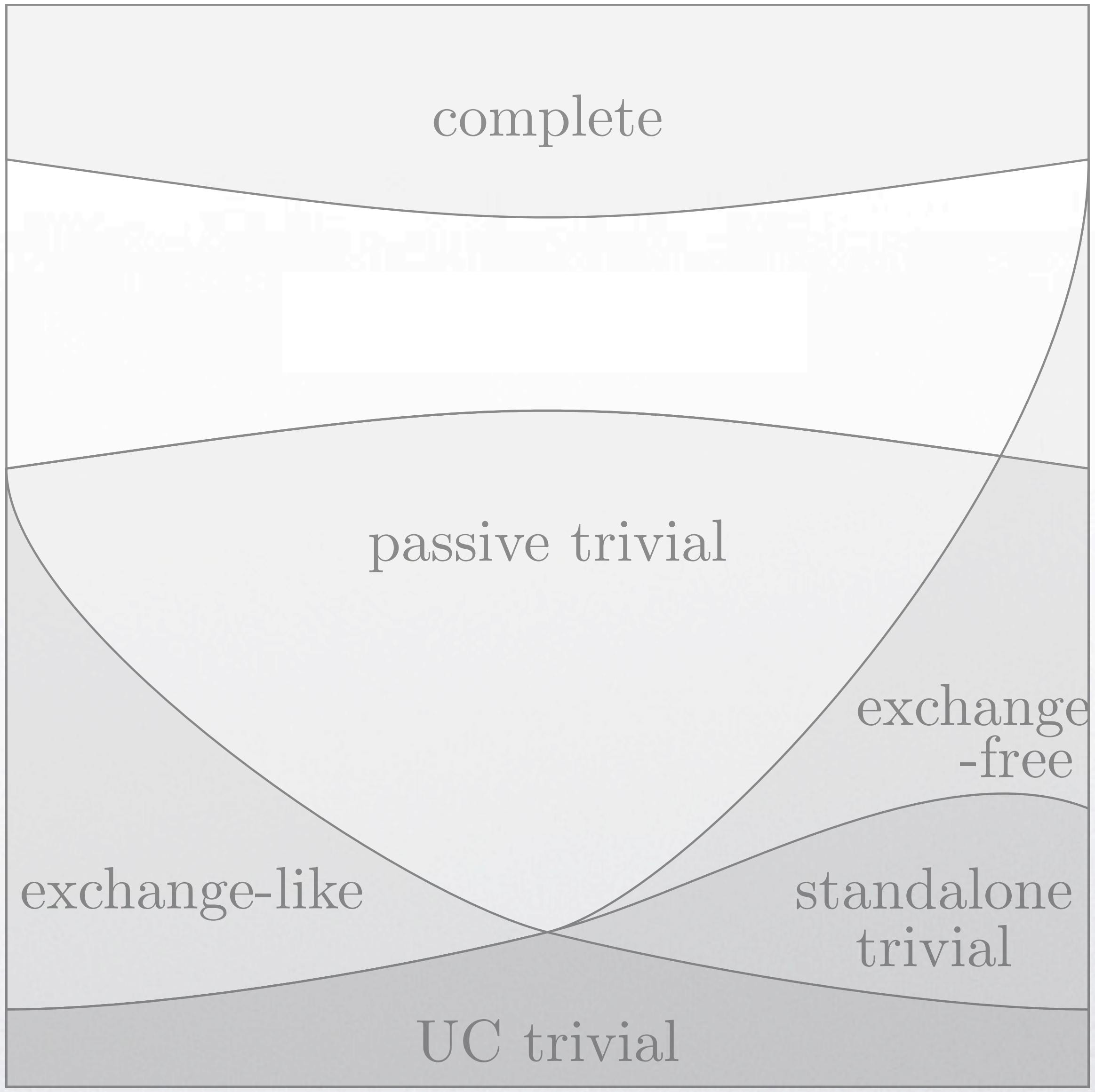




- Can argue: in the F_{Exch} protocol, the expected round in the simulation at which simulator for corrupt Alice extracts her input is before Bob learns it in the real execution (or with Alice/Bob reversed). (Uses the fact that F_{Coin} cannot be used to communicate.)

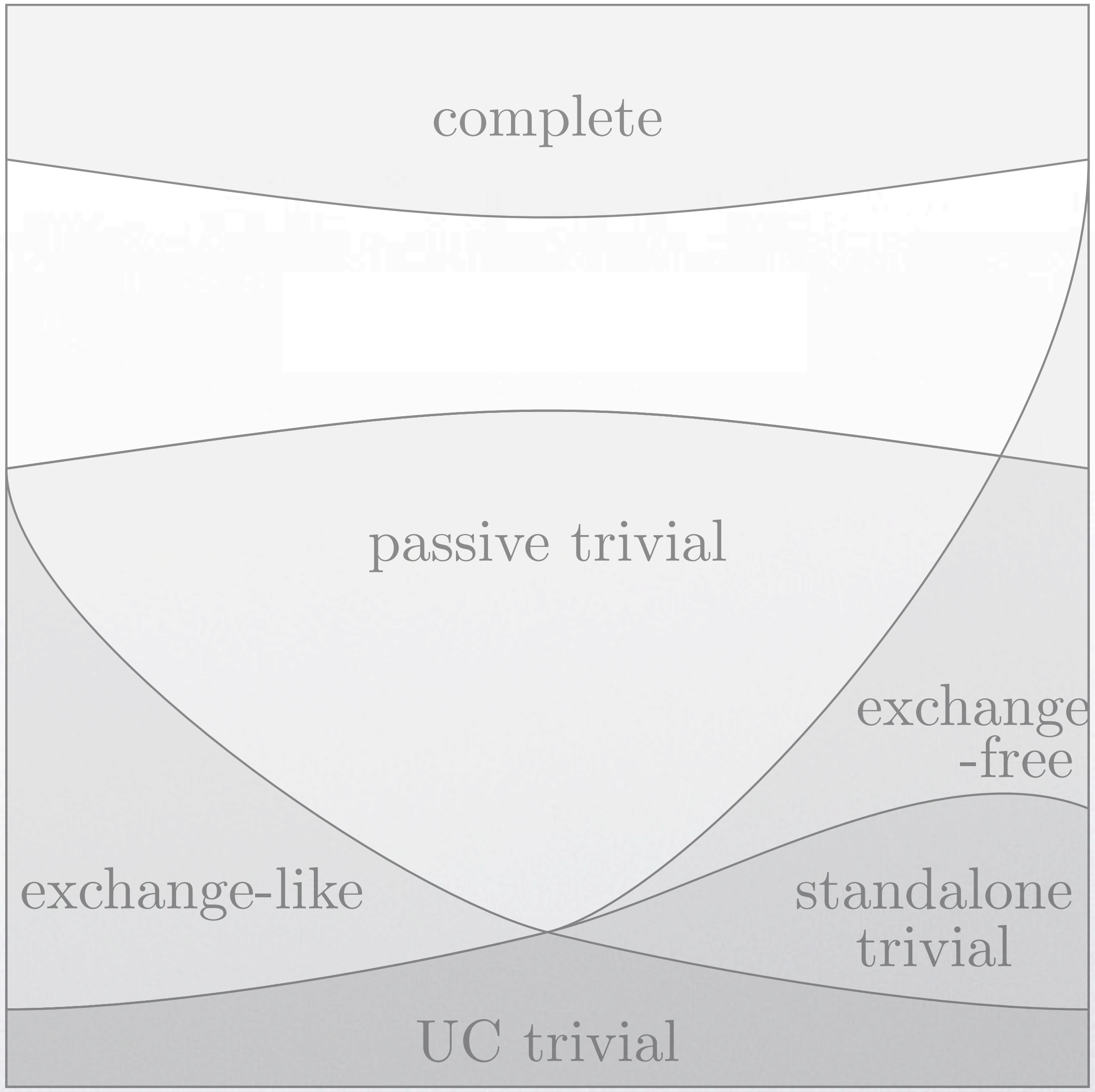
- So stopping the protocol at a random point gives the simulation an advantage over the honest strategy. Provides a “weak OT” that can then be amplified [DKS99]

shOT


complete
passive trivial
exchange
-free
exchange-like
standalone
trivial
UC trivial

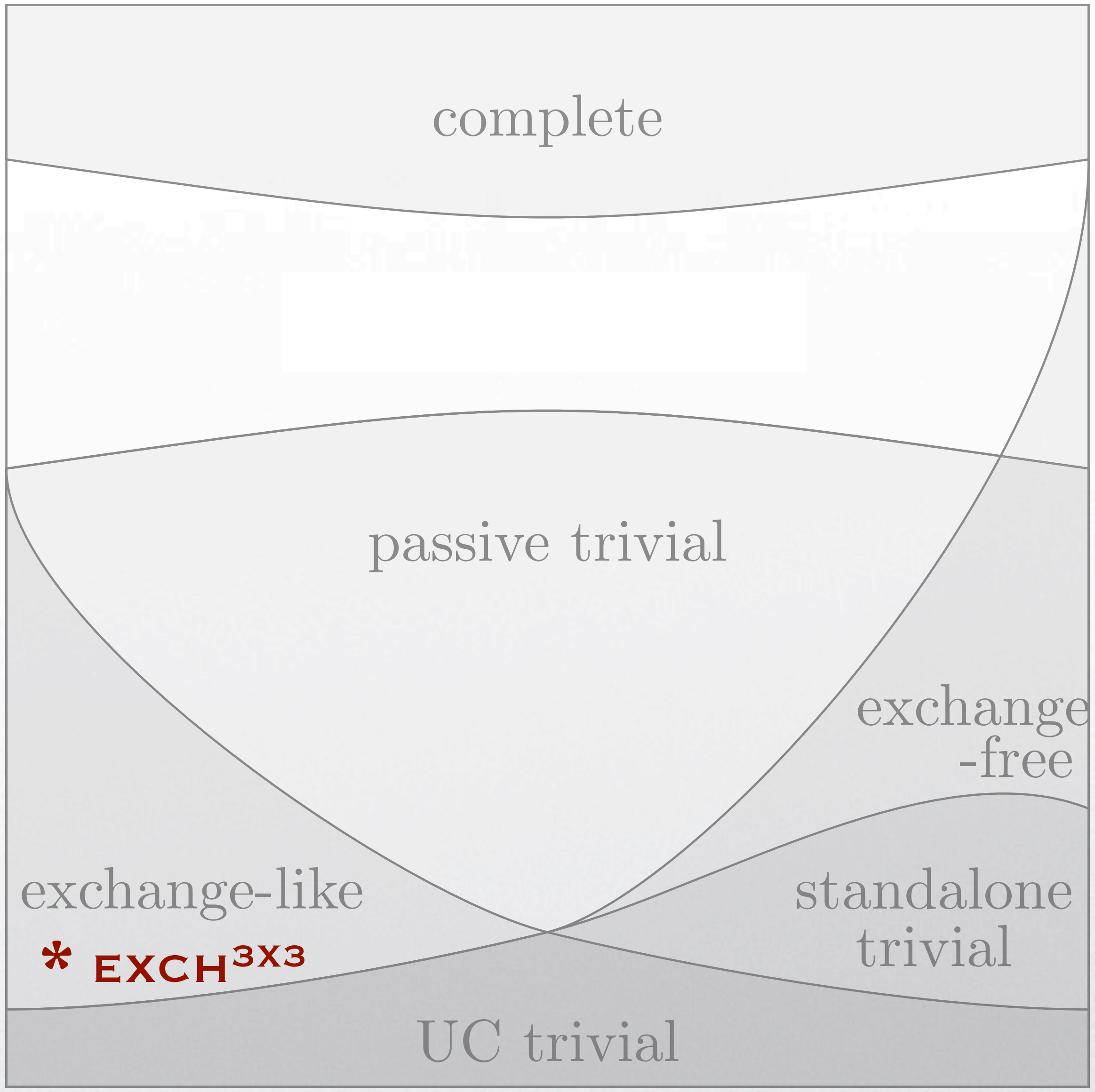
shOT

- For any “exchange-like” functionality G (not trivial), and for any F s.t F⊑G doesn’t hold statistically,



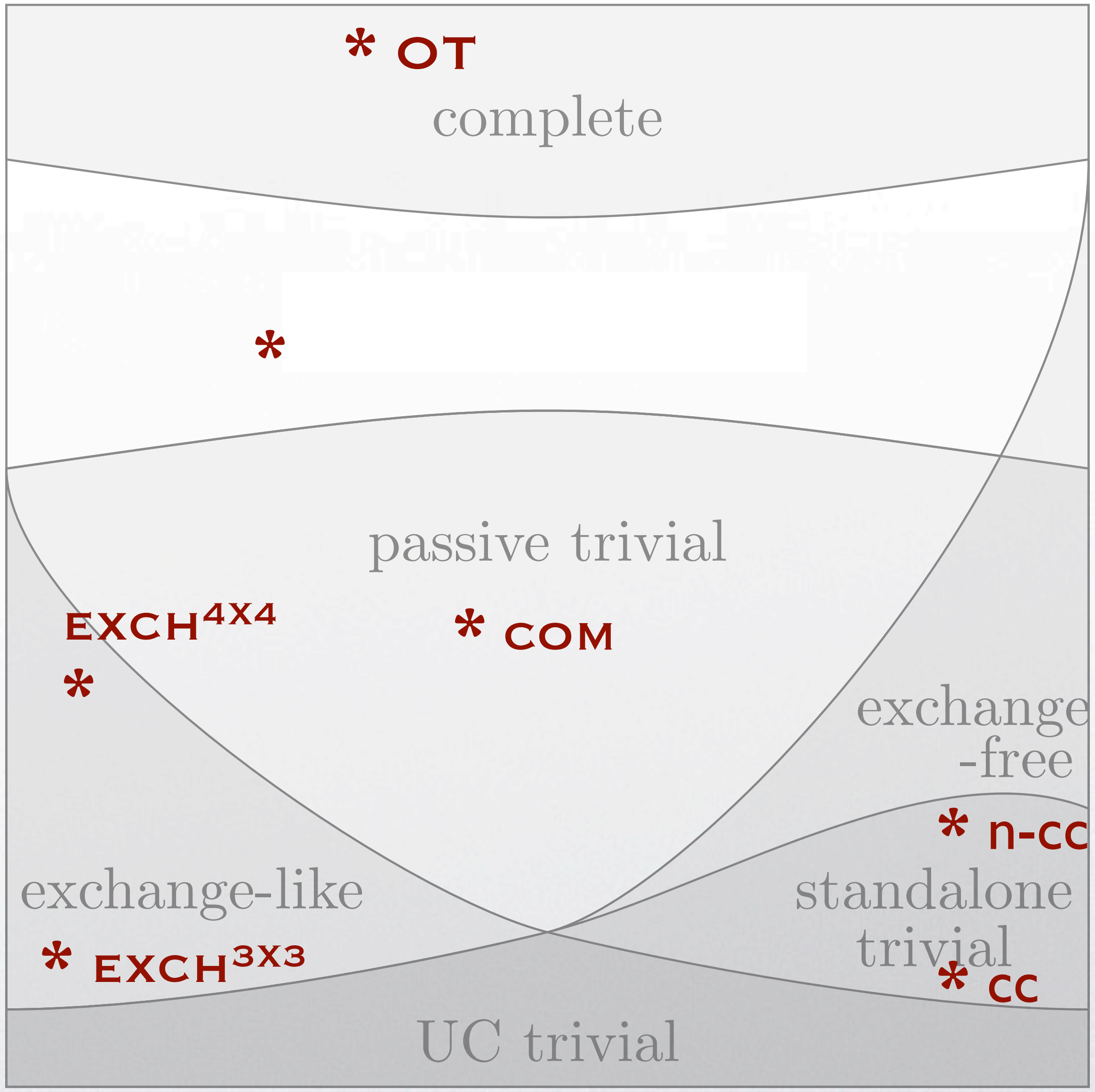
shOT

- For any "exchange-like" functionality G (not trivial), and for any F s.t F⊑G doesn't hold statistically,



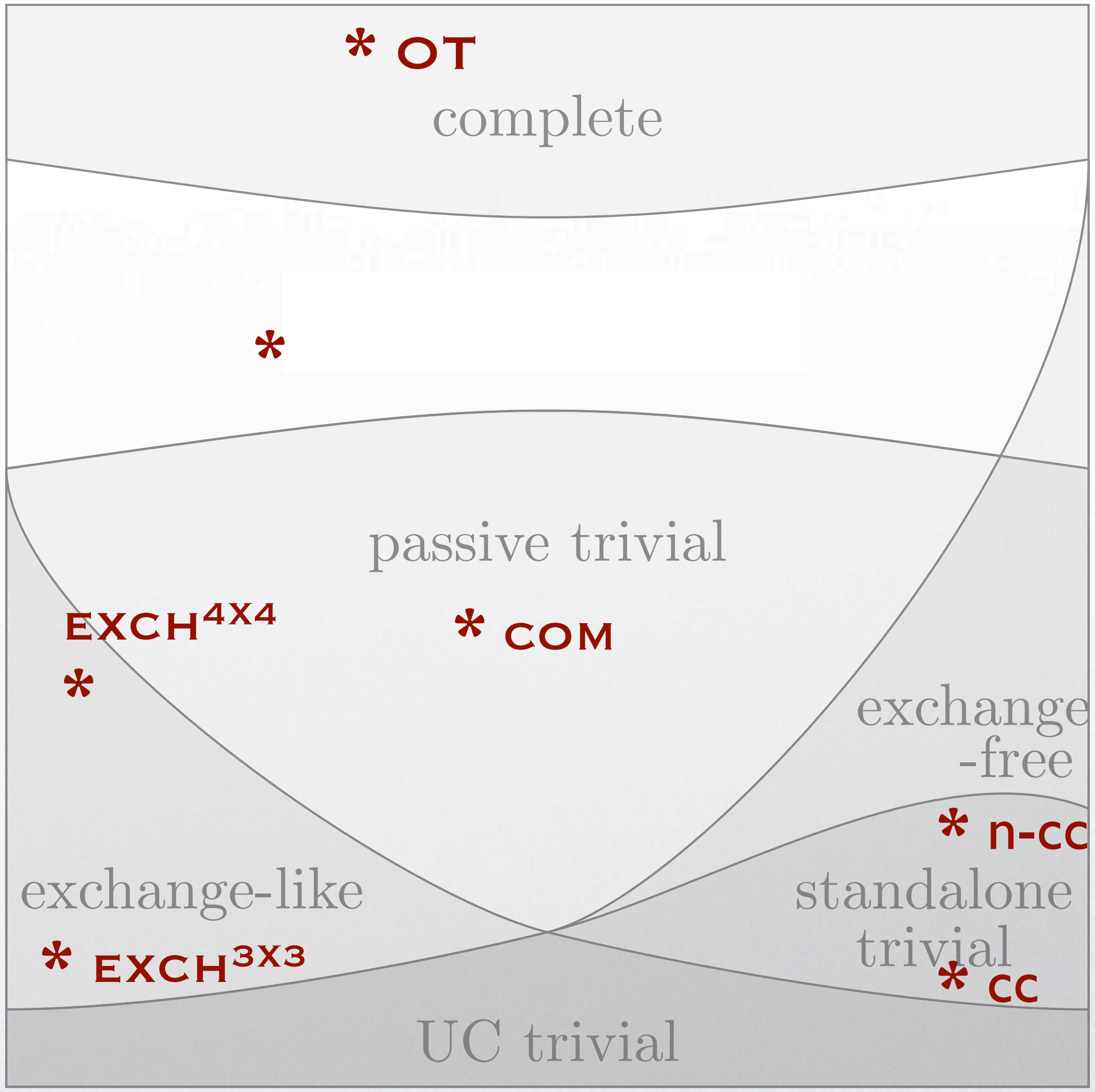
shOT

- For any “exchange-like” functionality G (not trivial), and for any F s.t F⊑G doesn’t hold statistically,



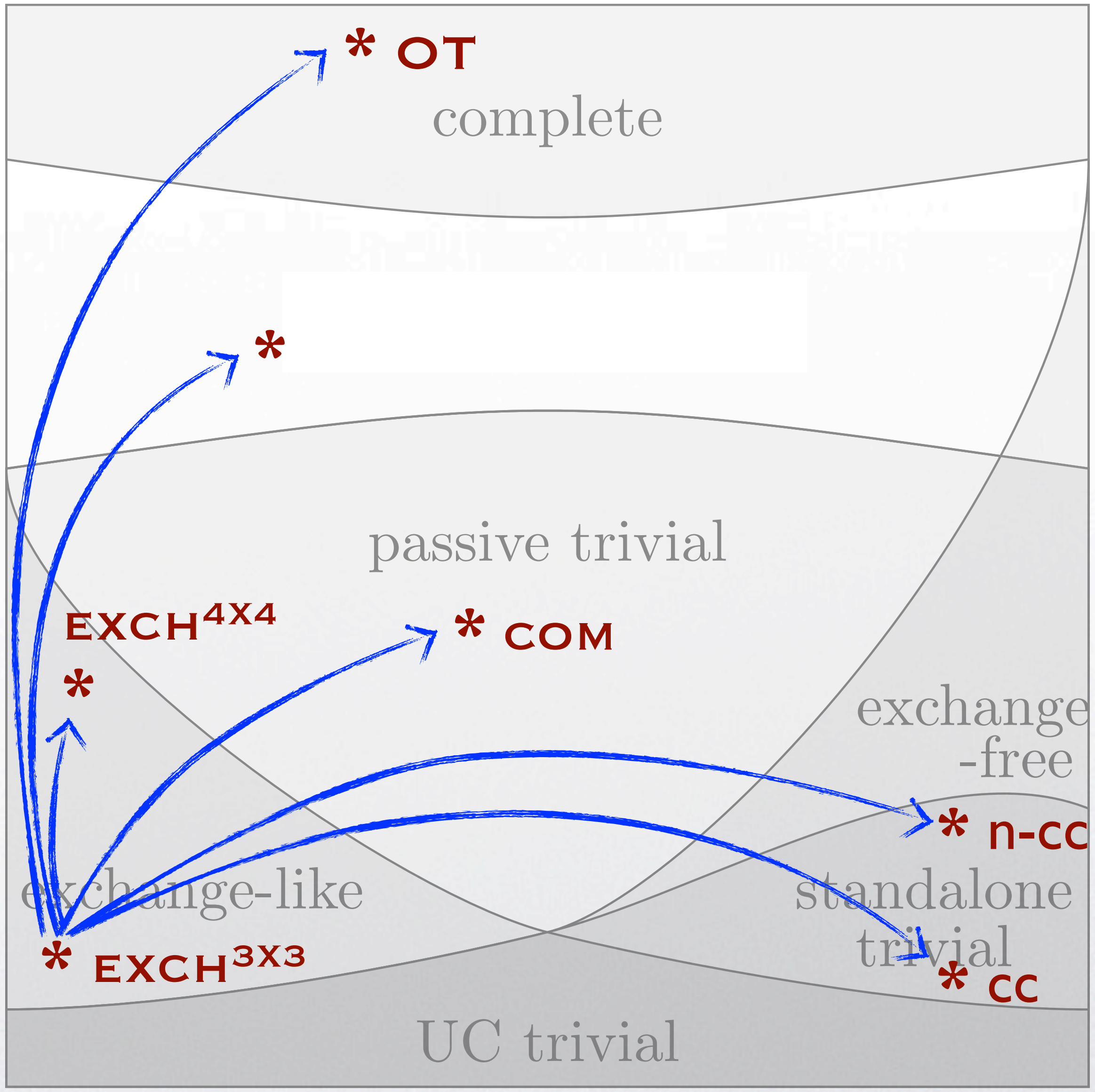
shOT

- For any “exchange-like” functionality G (not trivial), and for any F s.t $F \sqsubseteq G$ doesn’t hold statistically,
  - $F \sqsubseteq G$ is equivalent to shOT



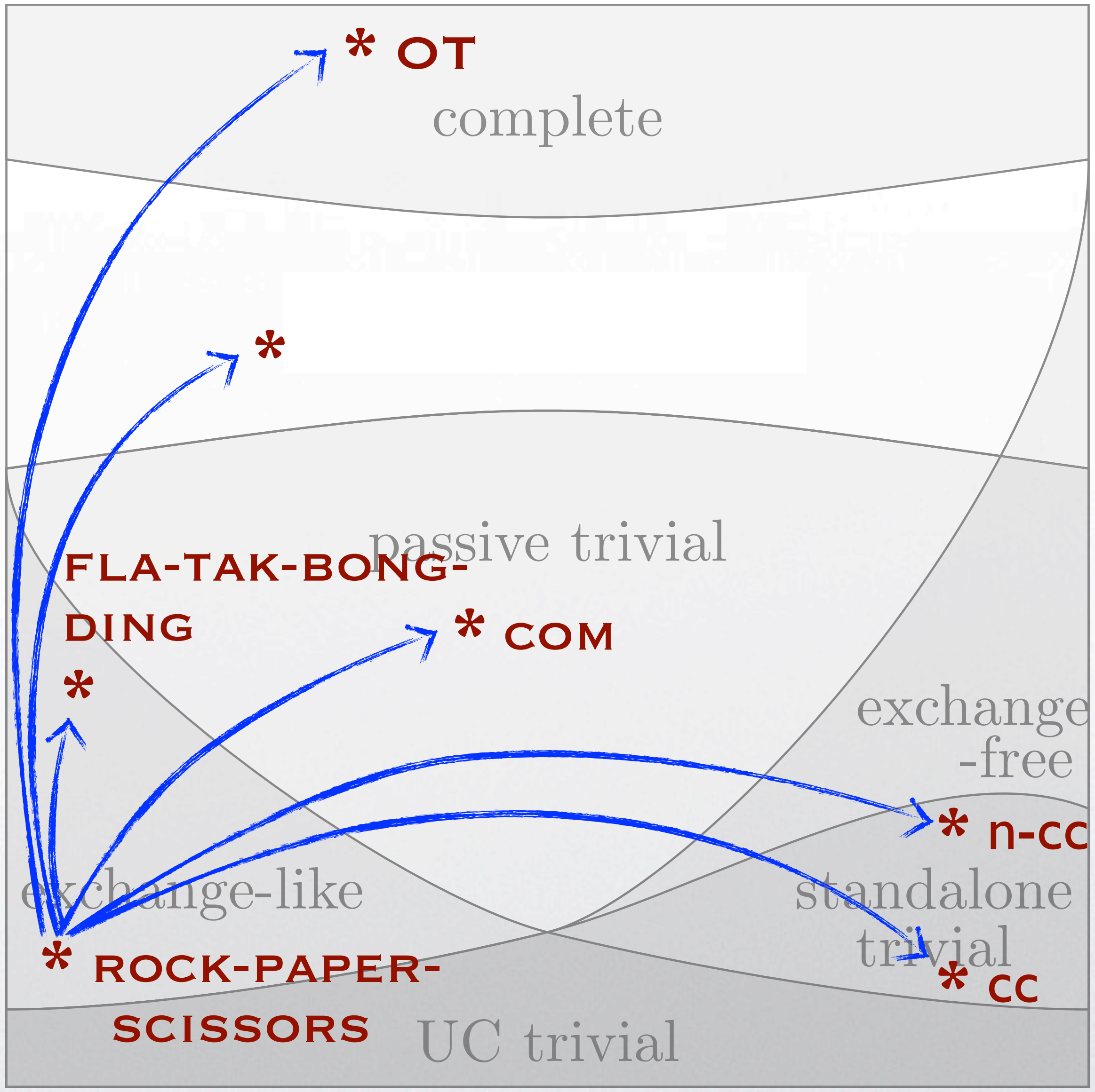
shOT

- For any “exchange-like” functionality G (not trivial), and for any F s.t F⊑G doesn’t hold statistically,
    - F⊑G is equivalent to shOT



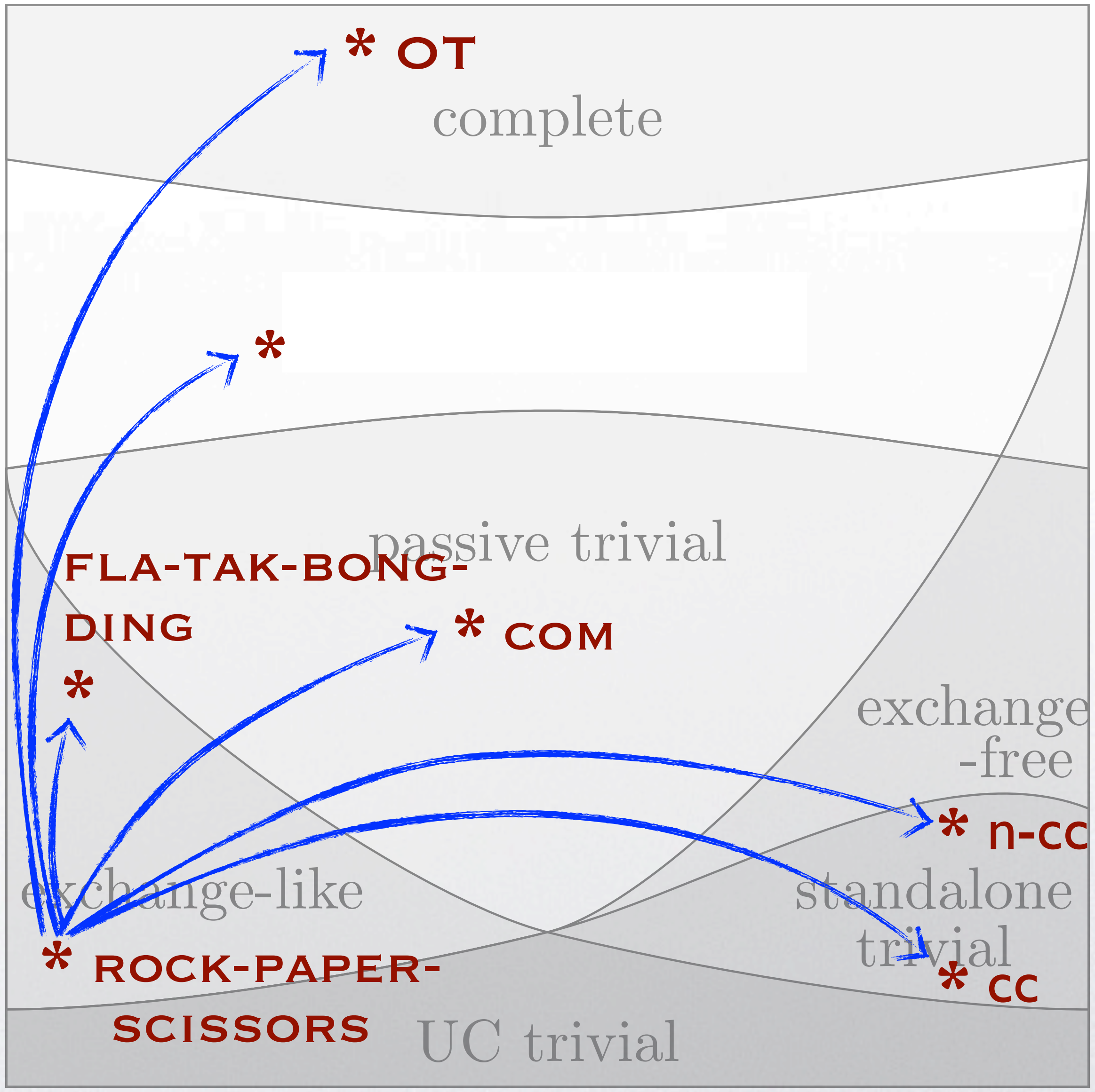
shOT

- For any “exchange-like” functionality G (not trivial), and for any F s.t F⊑G doesn’t hold statistically,
  - F⊑G is equivalent to shOT



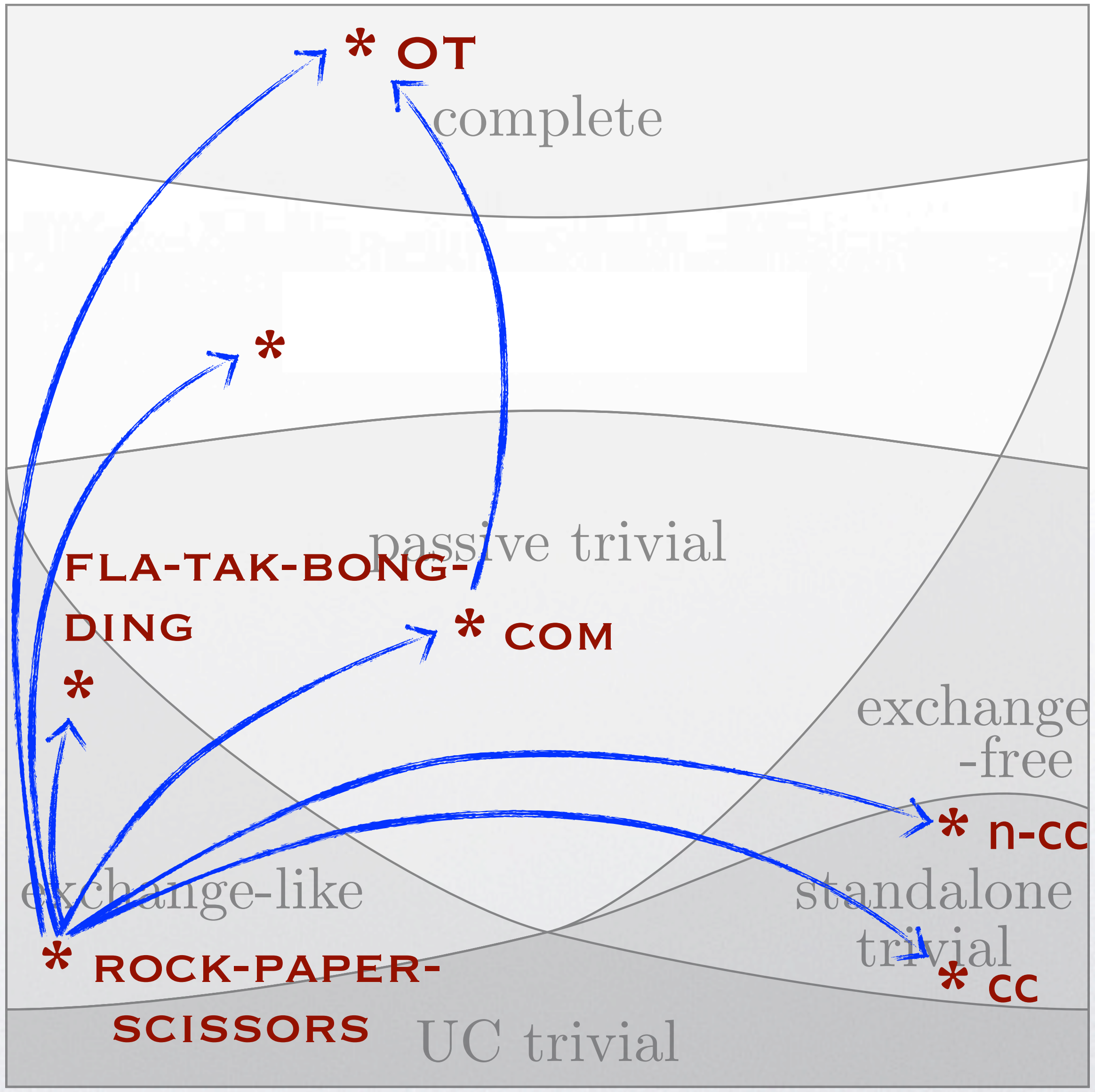
shOT

- For any "exchange-like" functionality G (not trivial), and for any F s.t F⊑G doesn't hold statistically,
  - F⊑G is equivalent to shOT
- Also, if F complete and G passive trivial (not trivial), F⊑G is equivalent to shOT



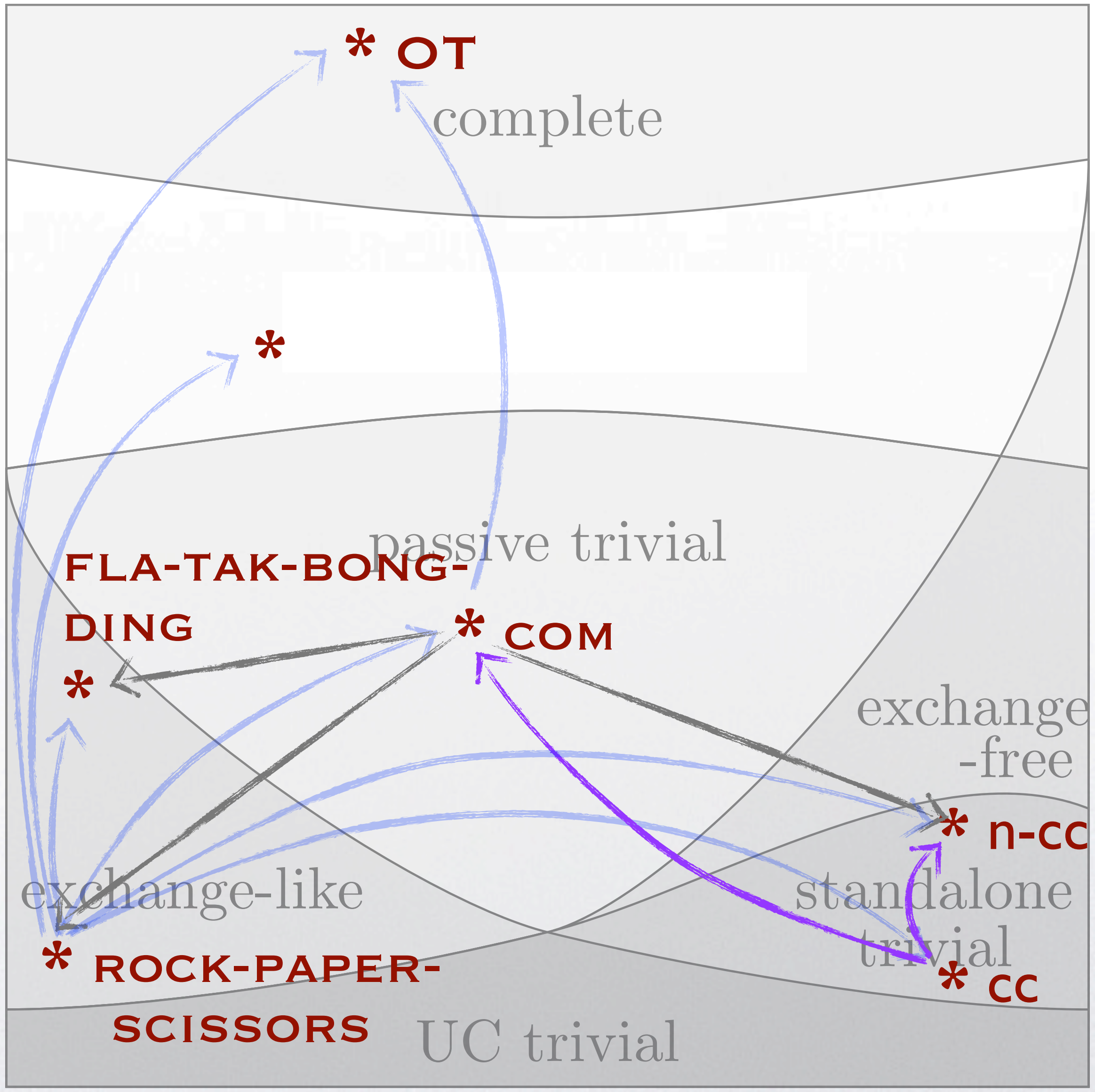
shOT

- For any "exchange-like" functionality G (not trivial), and for any F s.t F⊑G doesn't hold statistically,
  - F⊑G is equivalent to shOT
- Also, if F complete and G passive trivial (not trivial), F⊑G is equivalent to shOT

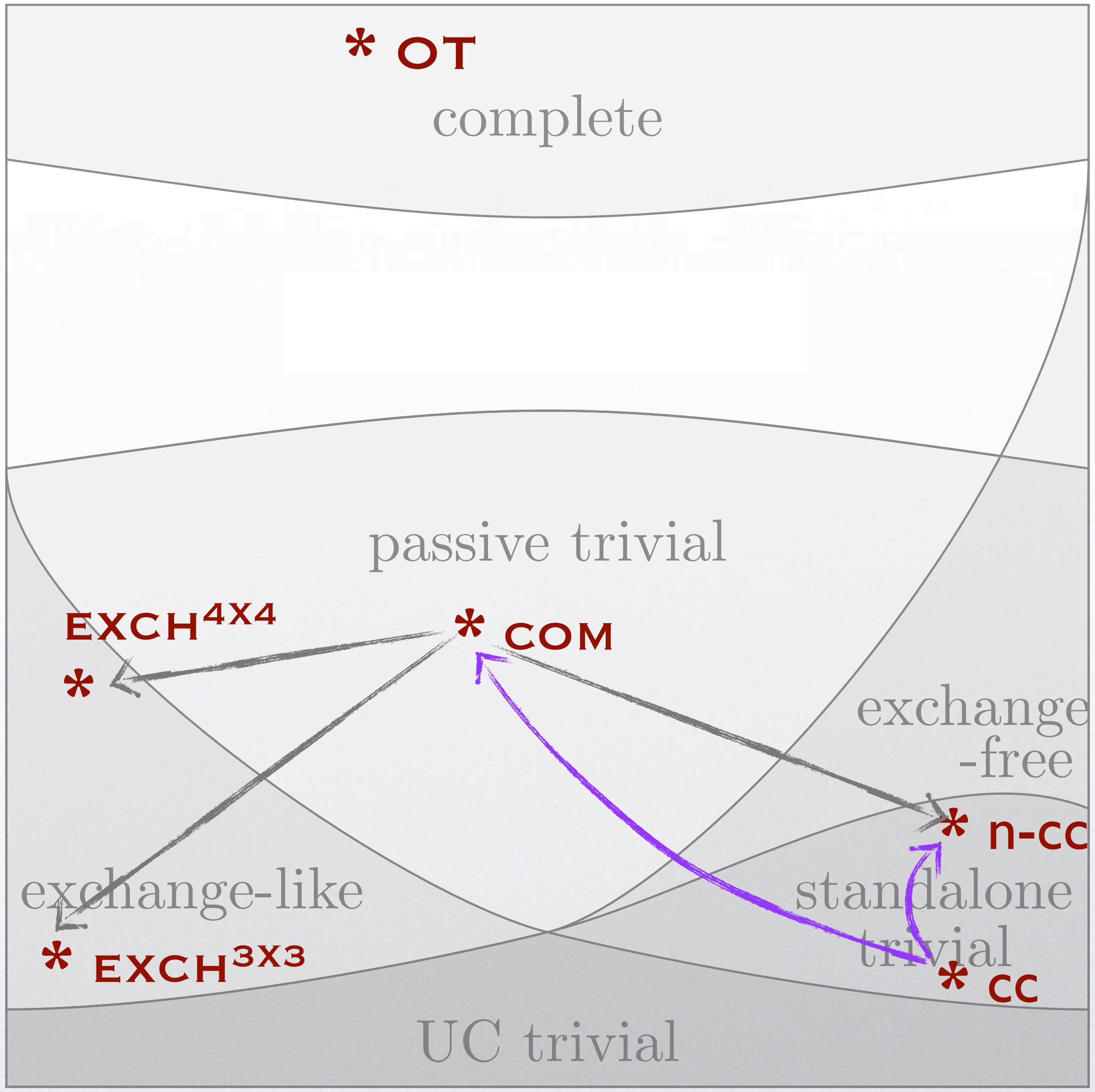


shOT

- For any "exchange-like" functionality G (not trivial), and for any F s.t F⊑G doesn't hold statistically,
  - F⊑G is equivalent to shOT
- Also, if F complete and G passive trivial (not trivial), F⊑G is equivalent to shOT
- All other reductions among "classified" F, G are implied by OWF (by results in [MPR09,MPR10b])

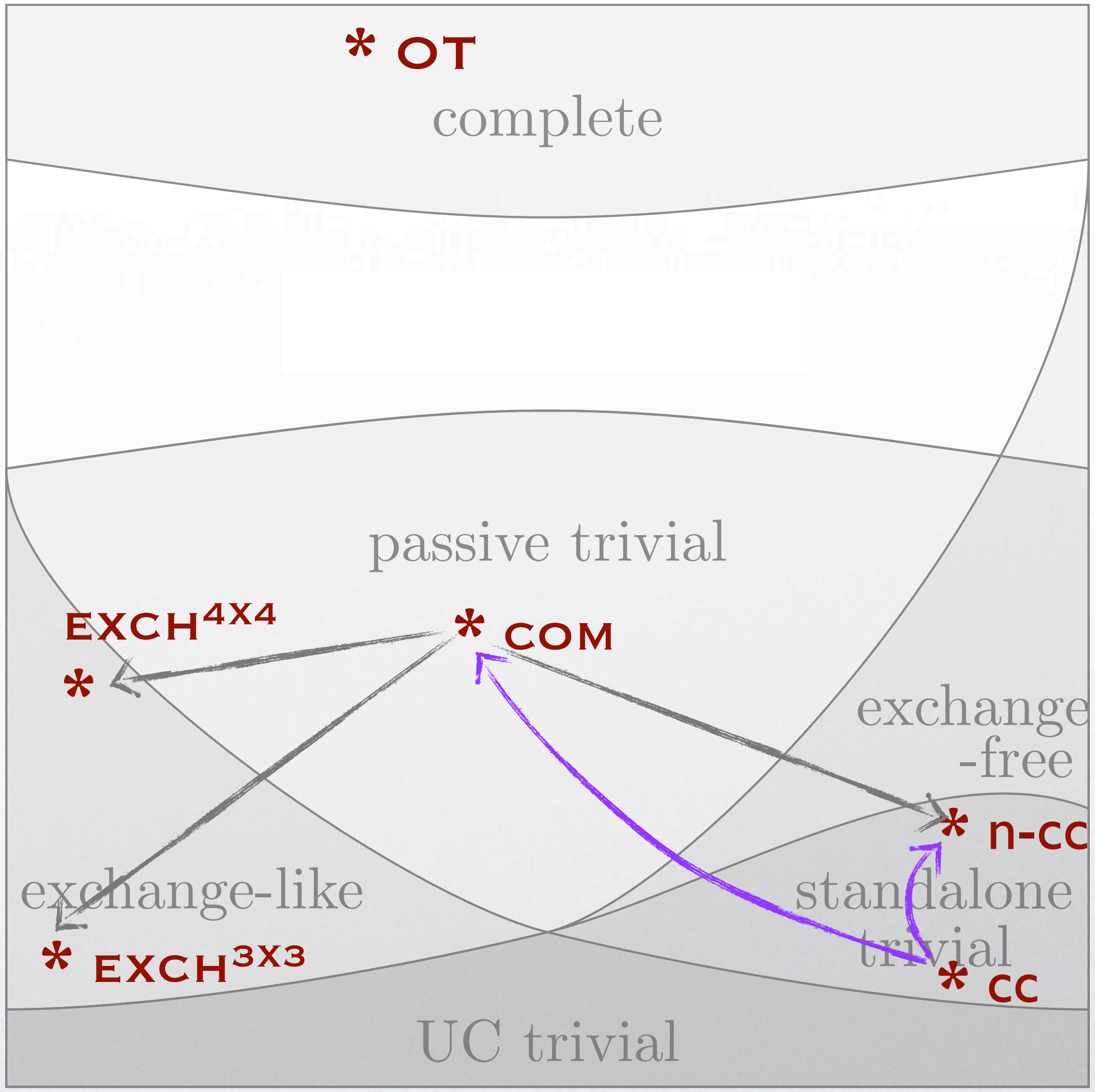


OWF


* OT
complete
passive trivial
EXCH^4X4
*
* COM
exchange
-free
* n-CC
exchange-like
* EXCH^3X3
standalone
trivial
* CC
UC trivial

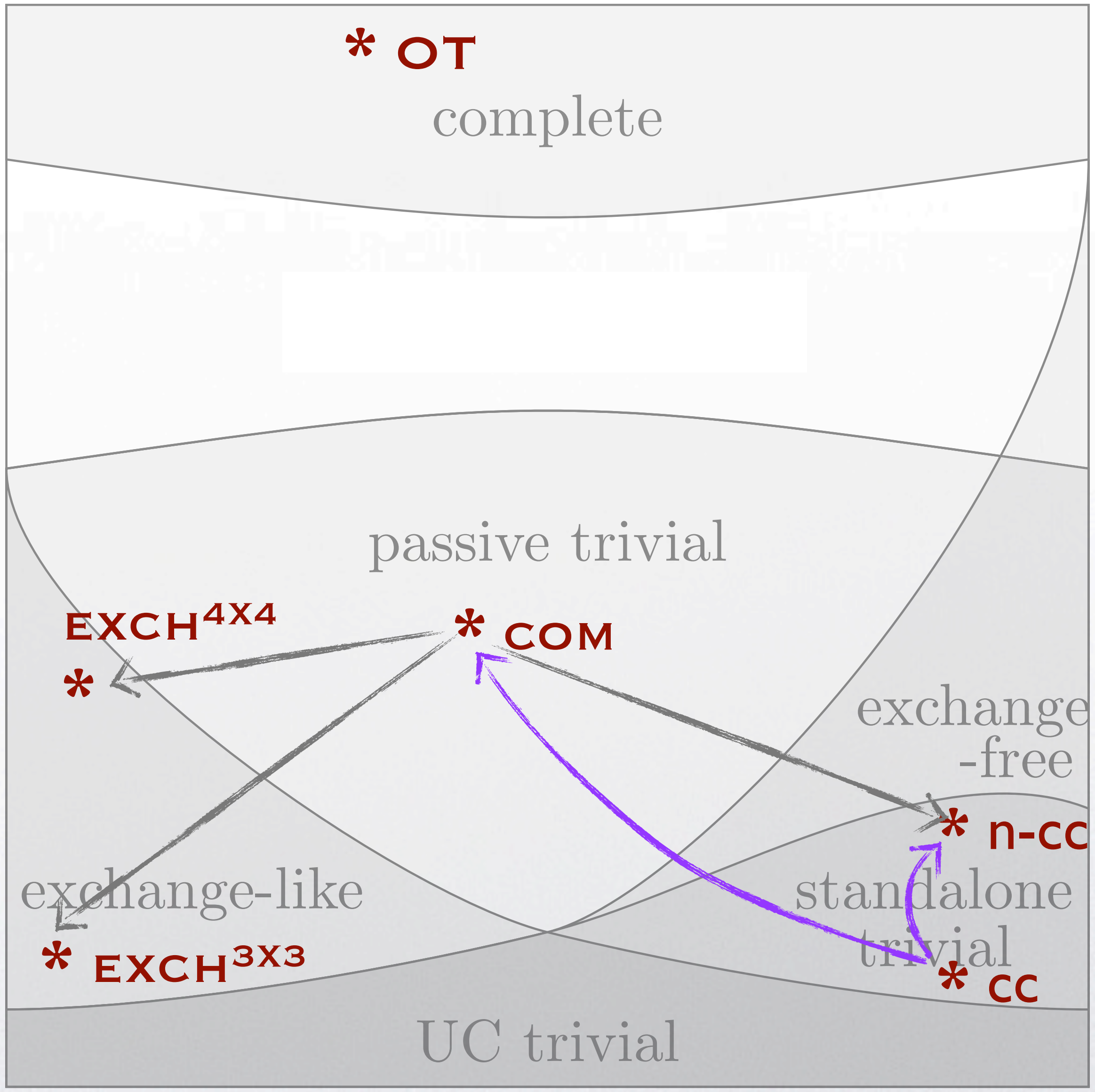
OWF

- Conjecture: *all* these reductions *imply* OWF (except those that hold statistically)



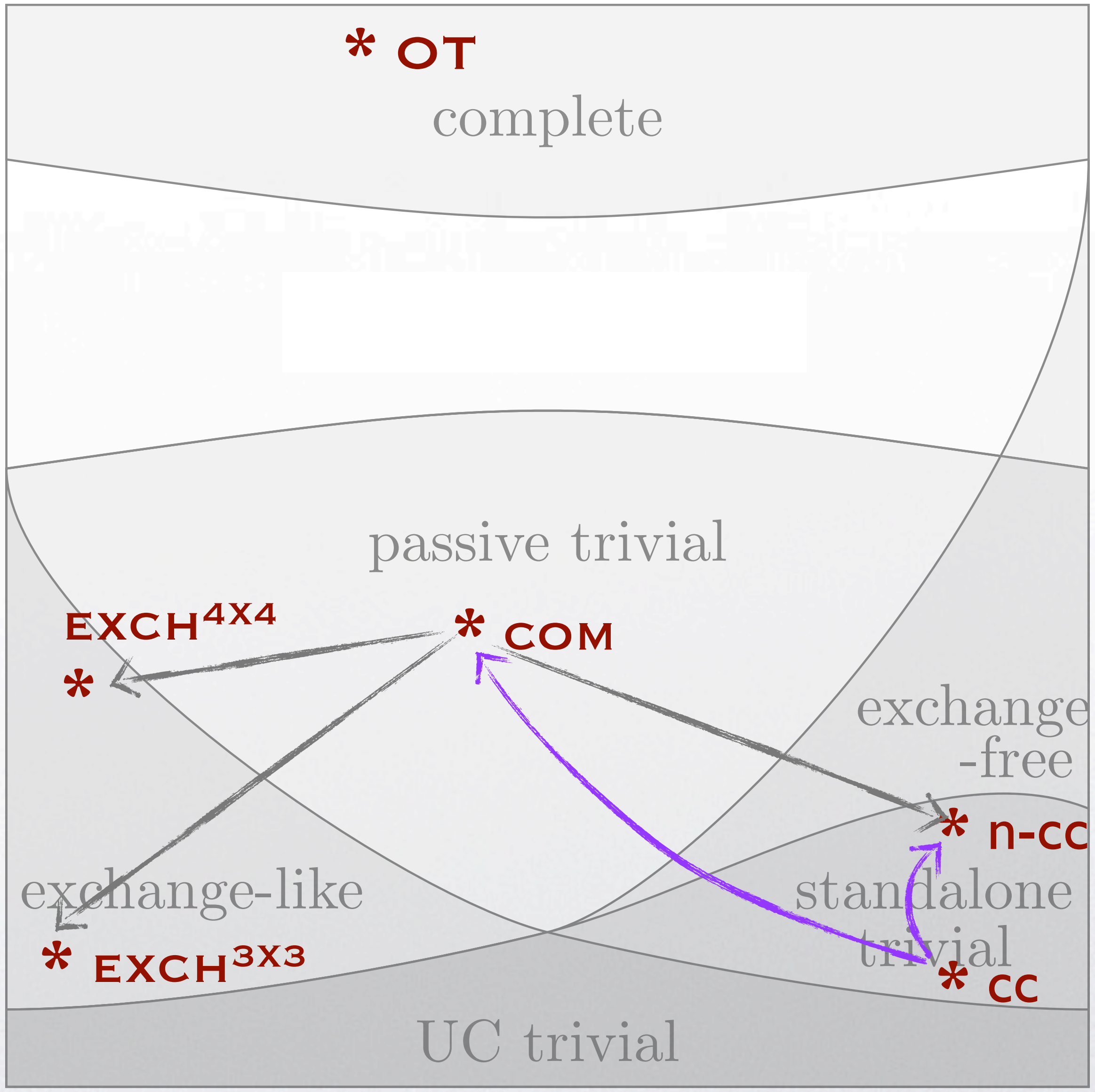
OWF

- Conjecture: *all* these reductions *imply* OWF (except those that hold statistically)
- We validate the conjecture for a large set, using "frontier analysis"

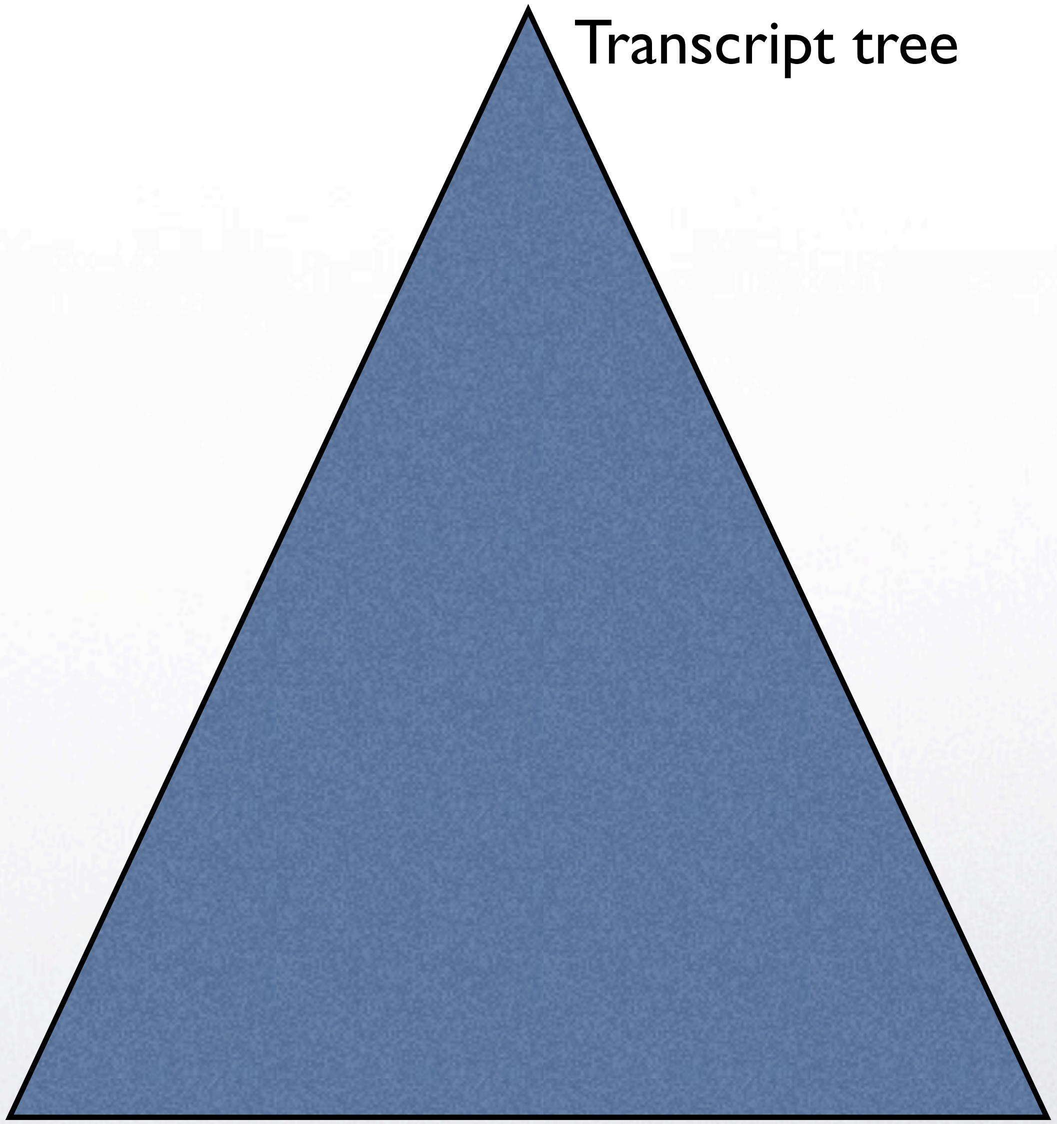


OWF

- Conjecture: *all* these reductions *imply* OWF (except those that hold statistically)
- We validate the conjecture for a large set, using "frontier analysis"
    - Frontier analysis: appears in [CI'93]. Reinvented (for other uses) in [MPR09], and used extensively in [MMOPR,MPS]

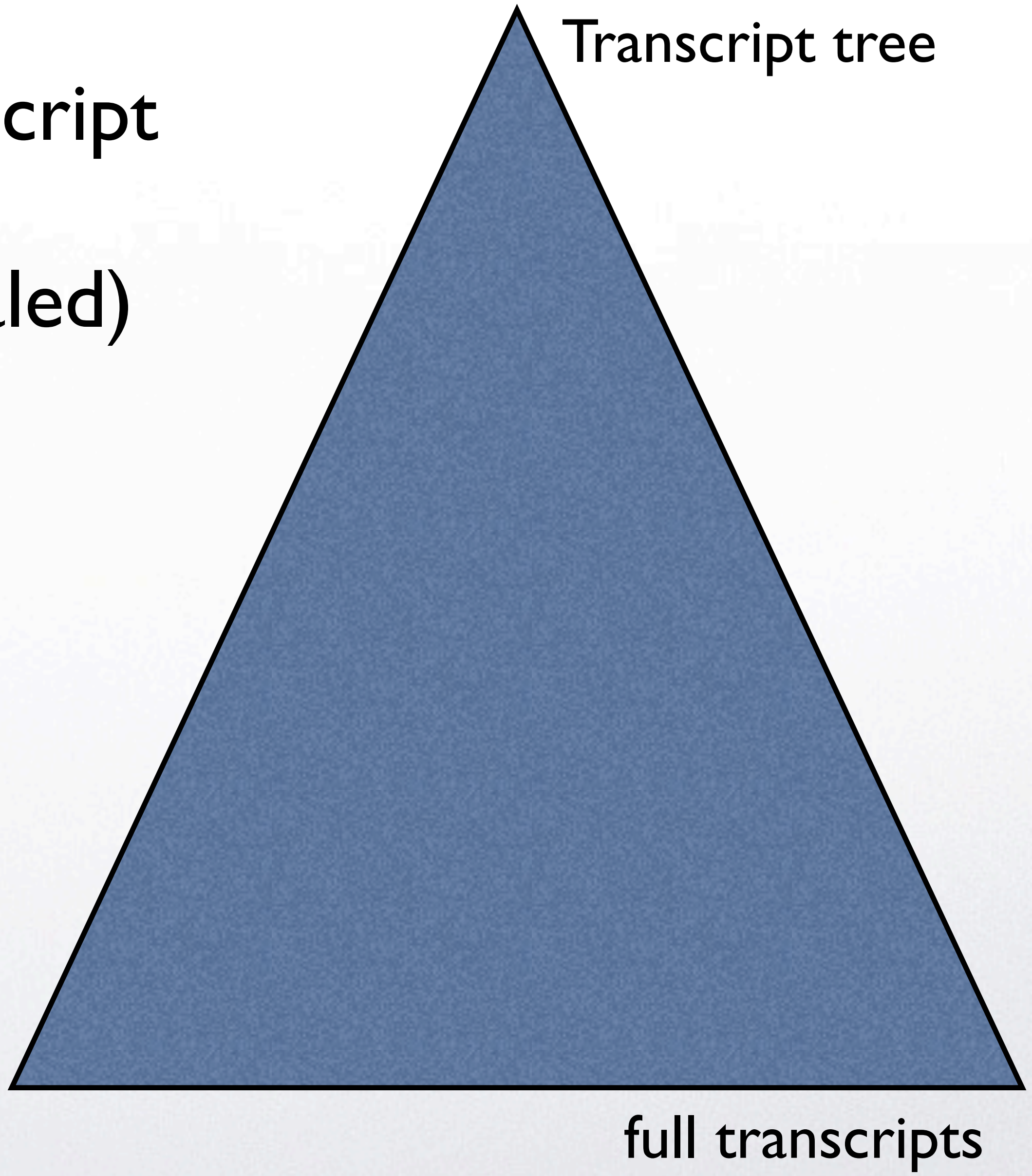


Frontier Analysis & OWF



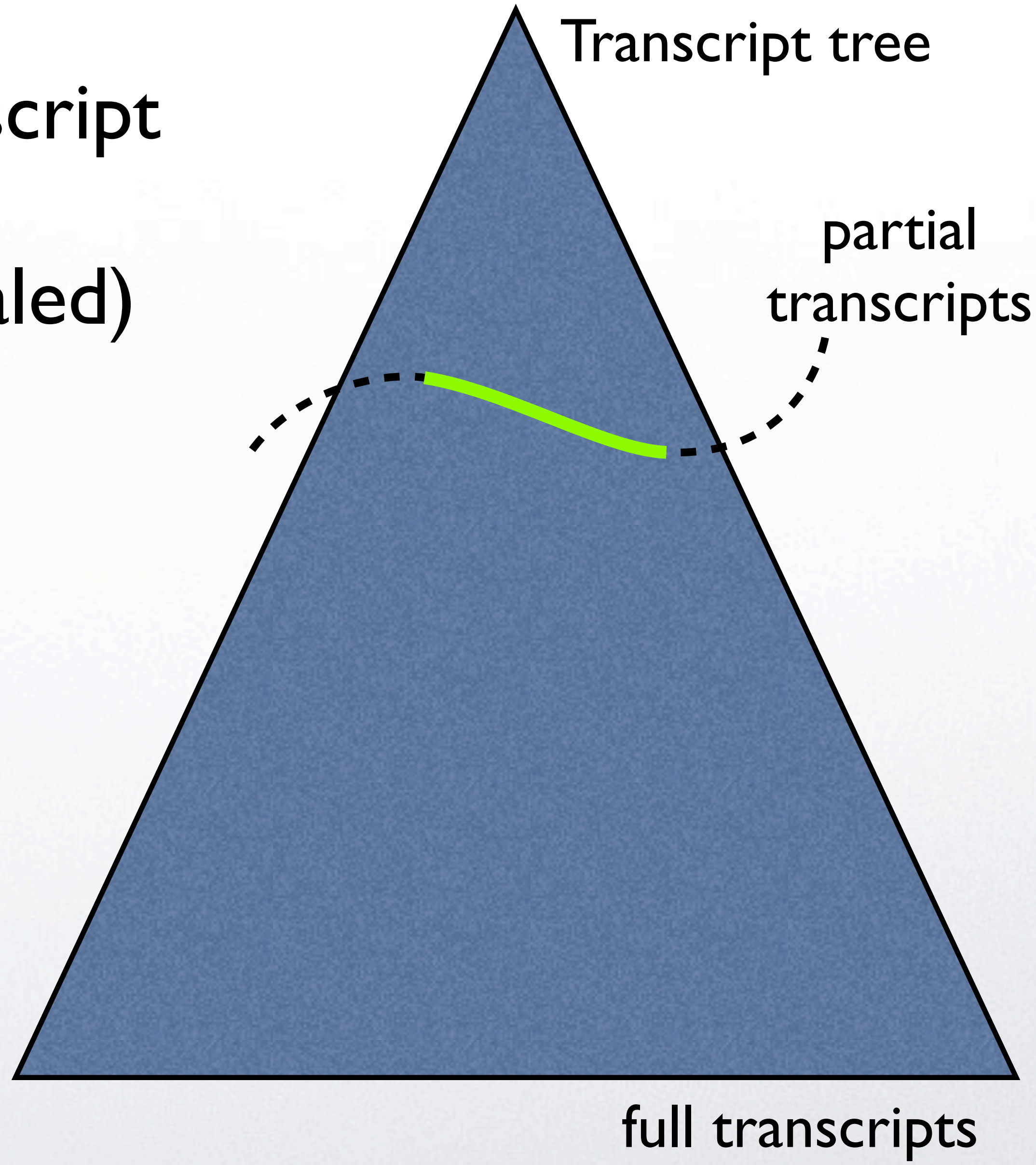
Frontier Analysis & OWF

- Considers frontiers in a protocol's "transcript tree" where certain properties hold (e.g. some information about an input is revealed)



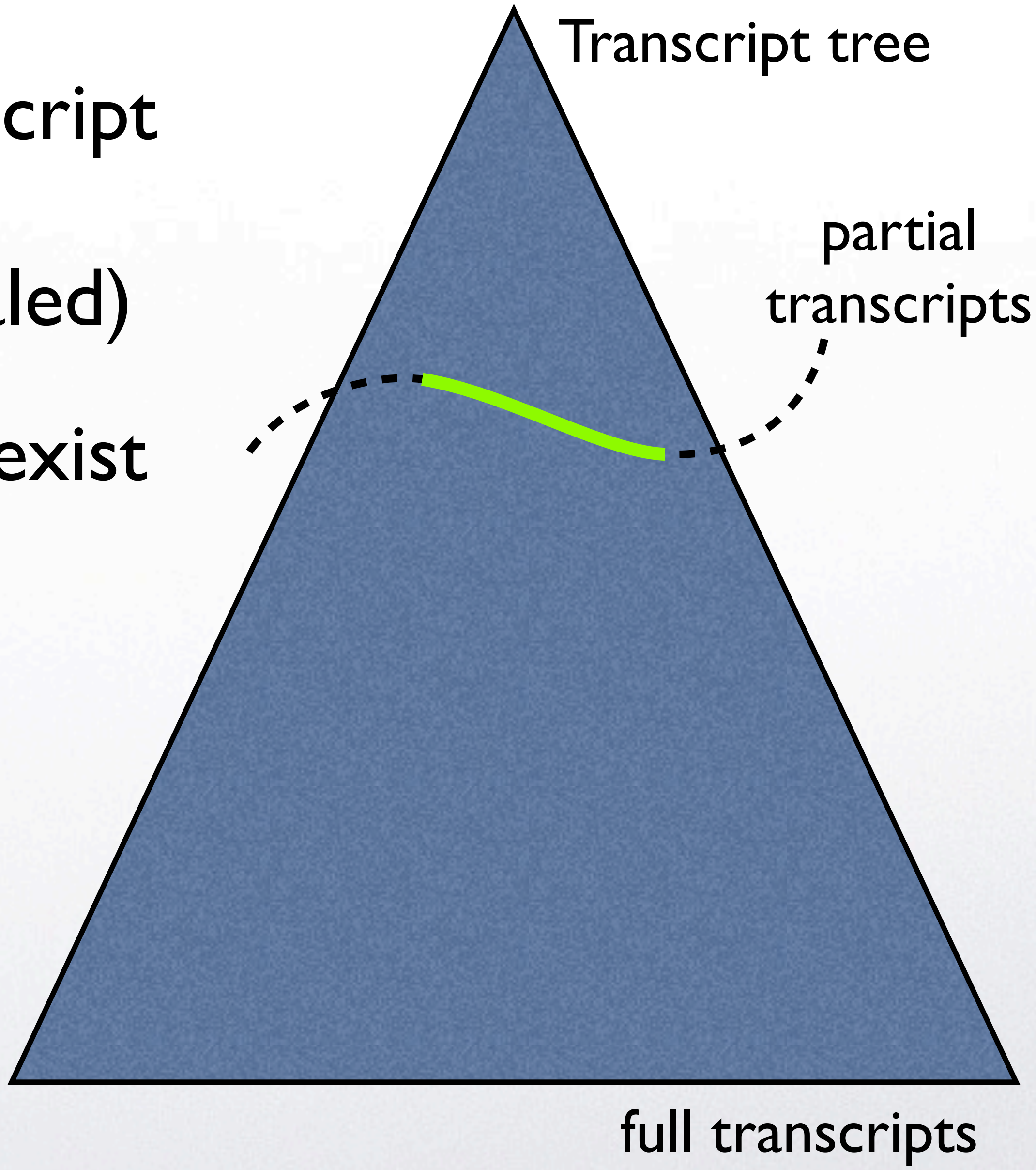
Frontier Analysis & OWF

- Considers frontiers in a protocol's "transcript tree" where certain properties hold (e.g. some information about an input is revealed)



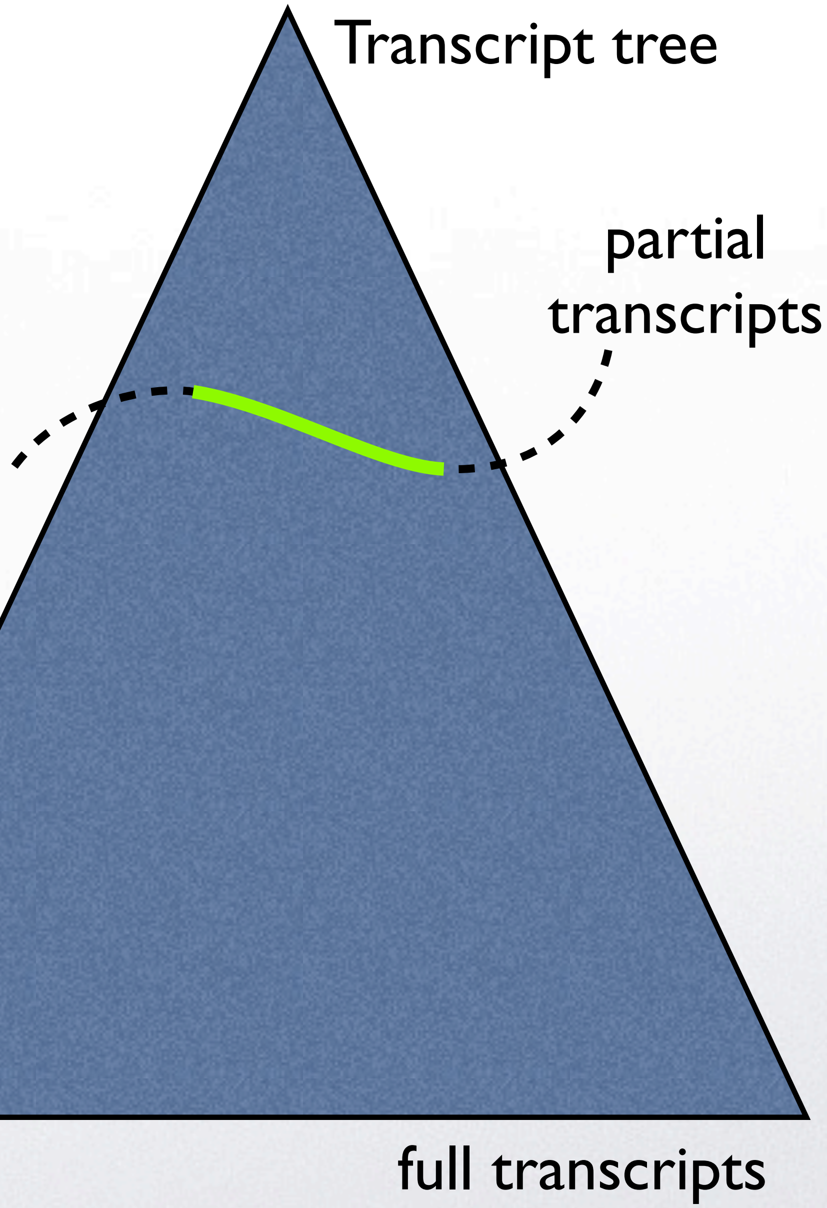
Frontier Analysis & OWF

- Considers frontiers in a protocol's "transcript tree" where certain properties hold (e.g. some information about an input is revealed)
    - Can show that certain frontiers must exist



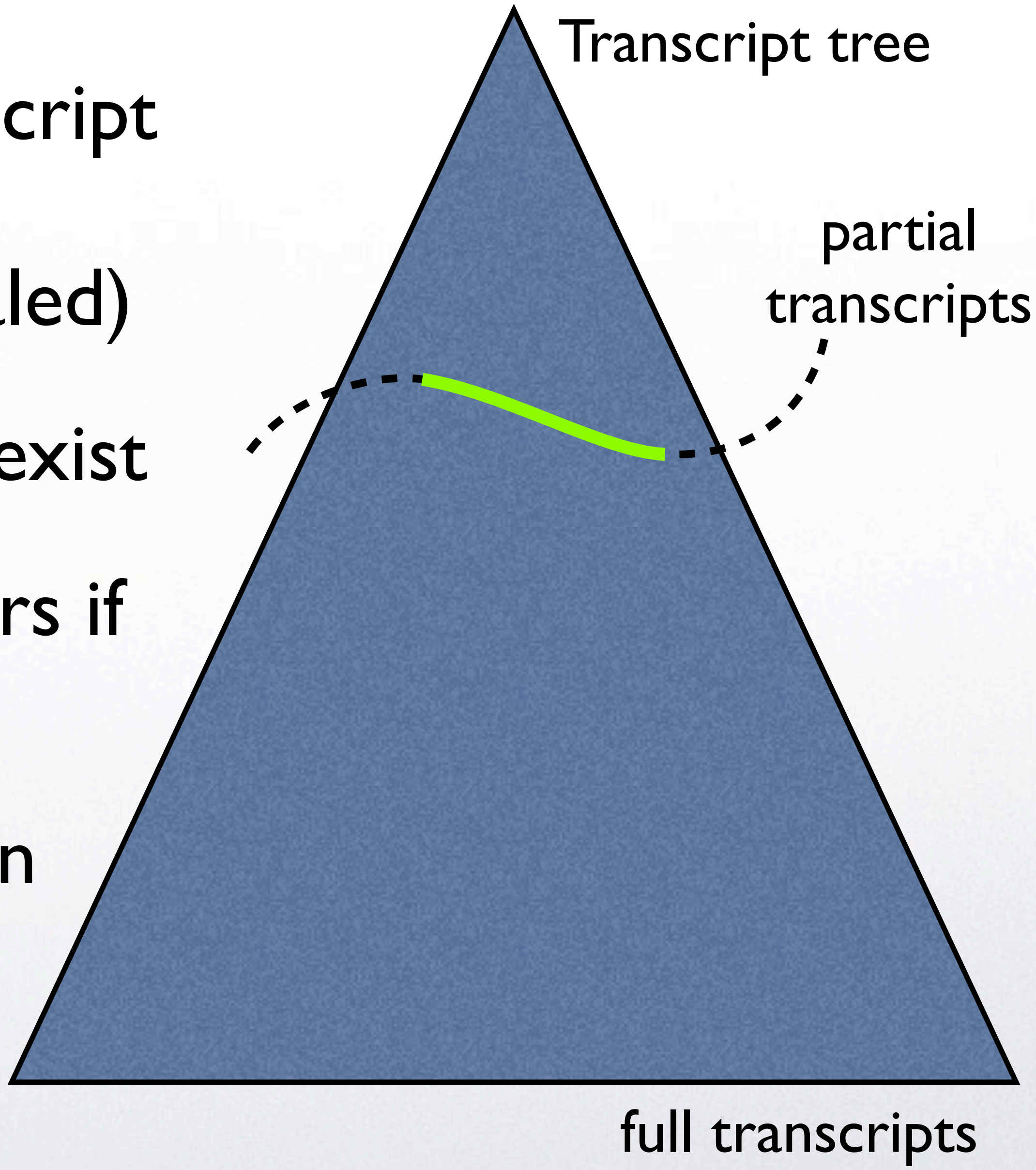
Frontier Analysis & OWF

- Considers frontiers in a protocol's "transcript tree" where certain properties hold (e.g. some information about an input is revealed)
    - Can show that certain frontiers must exist
    - Attacks can be launched at the frontiers if they can be detected



Frontier Analysis & OWF

- Considers frontiers in a protocol's "transcript tree" where certain properties hold (e.g. some information about an input is revealed)
  - Can show that certain frontiers must exist
  - Attacks can be launched at the frontiers if they can be detected
- Turns out, often, if OWFs don't exist, then can efficiently detect the frontiers (using characterization of OWF in [IL89])



Future Work

Future Work

- Conjecture: Among 2-party SFE functionalities F, G, all assumptions $F \sqsubseteq G$ are equivalent to either OWF or shOT

Future Work

- Conjecture: Among 2-party SFE functionalities F, G, all assumptions $F \sqsubseteq G$ are equivalent to either OWF or shOT

- *Key-Agreement* is a “distinct” assumption that emerges on considering 3-party functionalities. Question: Are there more?

Future Work

In progress: "Intractability Abstractions" to formalize distinct assumptions, generalizing the Impagliazzo-Rudich approach

- Conjecture: Among 2-party SFE functionalities F, G, all assumptions F $\sqsubseteq$ G are equivalent to either OWF or shOT
- *Key-Agreement* is a "distinct" assumption that emerges on considering 3-party functionalities. Question: Are there more?

Future Work

In progress: "Intractability Abstractions" to formalize distinct assumptions, generalizing the Impagliazzo-Rudich approach

- Conjecture: Among 2-party SFE functionalities F, G, all assumptions F $\sqsubseteq$ G are equivalent to either OWF or shOT
- *Key-Agreement* is a "distinct" assumption that emerges on considering 3-party functionalities. Question: Are there more?
  - More generally, how about m-party functionalities for m > 2?

Future Work

In progress: "Intractability Abstractions" to formalize distinct assumptions, generalizing the Impagliazzo-Rudich approach

- Conjecture: Among 2-party SFE functionalities F, G, all assumptions F $\sqsubseteq$ G are equivalent to either OWF or shOT
- *Key-Agreement* is a "distinct" assumption that emerges on considering 3-party functionalities. Question: Are there more?
  - More generally, how about m-party functionalities for m > 2?
    - Even (statistical) cryptographic complexity little understood

Future Work

In progress: "Intractability Abstractions" to formalize distinct assumptions, generalizing the Impagliazzo-Rudich approach

- Conjecture: Among 2-party SFE functionalities F, G, all assumptions $F \sqsubseteq G$ are equivalent to either OWF or shOT
- *Key-Agreement* is a "distinct" assumption that emerges on considering 3-party functionalities. Question: Are there more?
  - More generally, how about m-party functionalities for m > 2?
    - Even (statistical) cryptographic complexity little understood
  - Randomized functionalities, fair functionalities, infinite functionalities? (Again, cryptographic complexity little understood)

Crypto Means & Goals

One-Way Functions

One-Way Permutations

Trapdoor One-Way Permutations

OT protocol

Collision-Resistant Hash Functions

Zero-Knowledge Proofs

Encryption

Signatures

Homomorphic Encryption

Secret Communication Channels

Authenticated Communication Channels

OT Channel

Digital Cash

Mental Poker

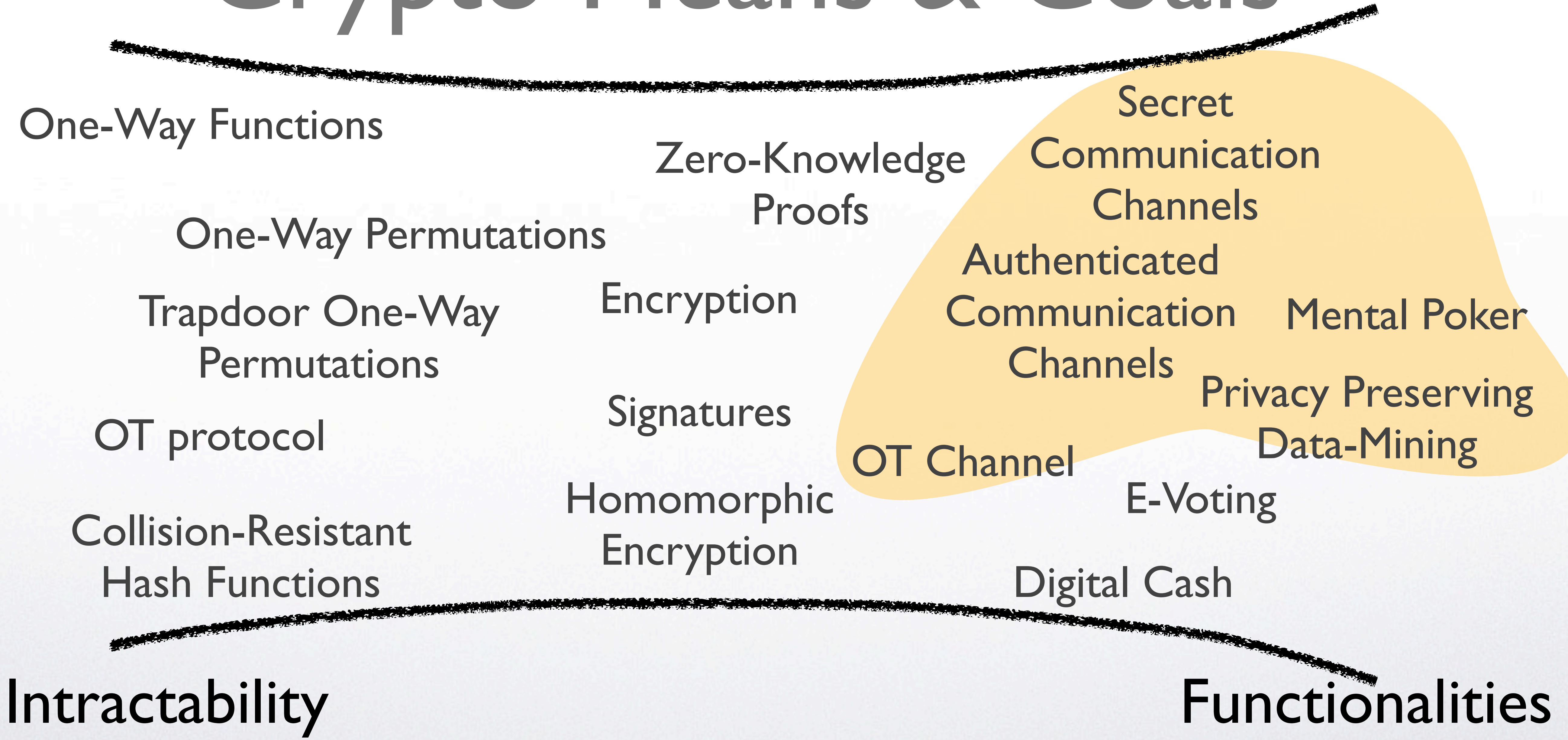
Privacy Preserving Data-Mining

E-Voting

Intractability

Functionalities

Crypto Means & Goals



One-Way Functions

One-Way Permutations

Trapdoor One-Way Permutations

OT protocol

Collision-Resistant Hash Functions

Zero-Knowledge Proofs

Encryption

Signatures

Homomorphic Encryption

Secret Communication Channels

Authenticated Communication Channels

Mental Poker

Privacy Preserving Data-Mining

OT Channel

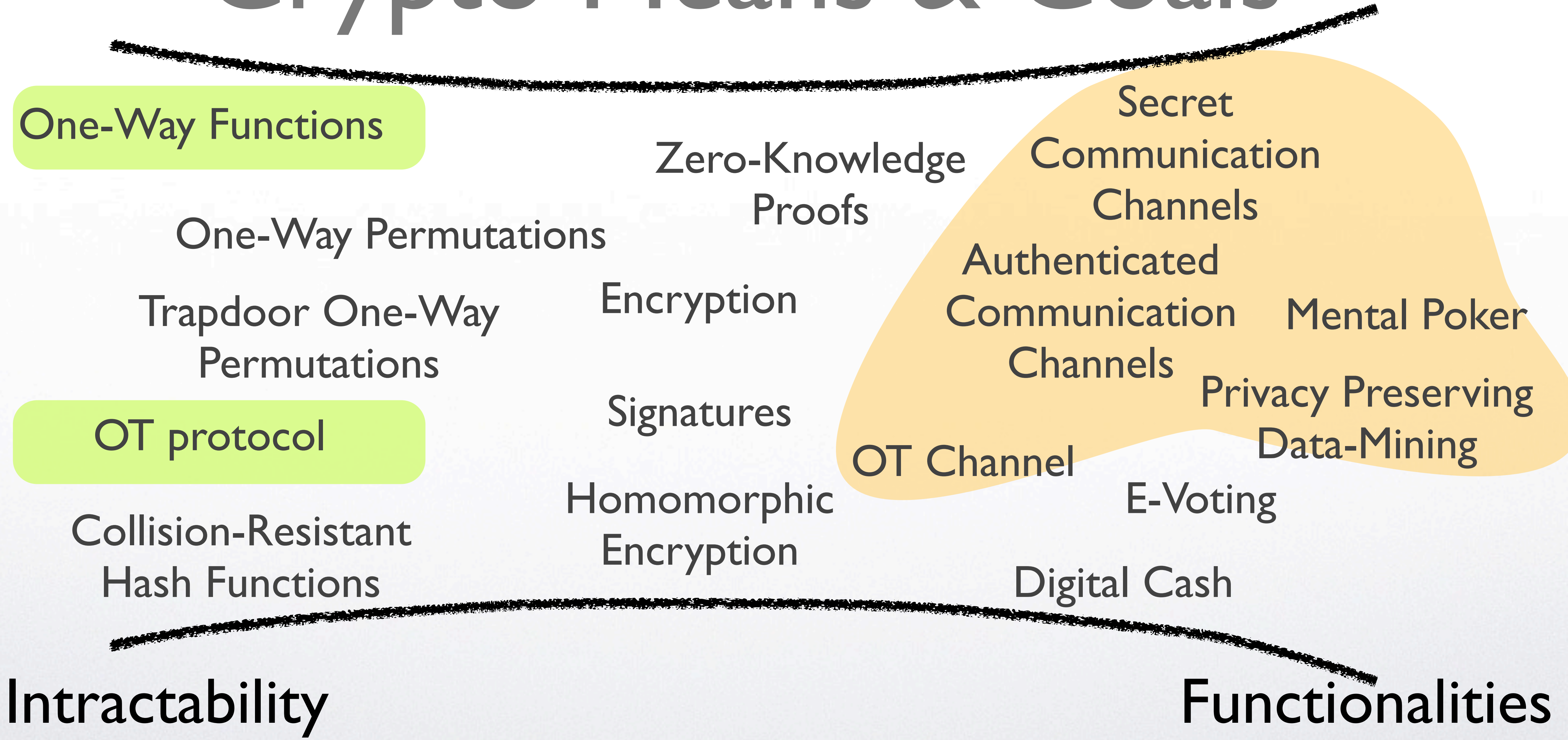
E-Voting

Digital Cash

Intractability

Functionalities

Crypto Means & Goals



One-Way Functions

One-Way Permutations

Trapdoor One-Way Permutations

OT protocol

Collision-Resistant Hash Functions

Zero-Knowledge Proofs

Encryption

Signatures

Homomorphic Encryption

Secret Communication Channels

Authenticated Communication Channels

Mental Poker

Privacy Preserving Data-Mining

OT Channel

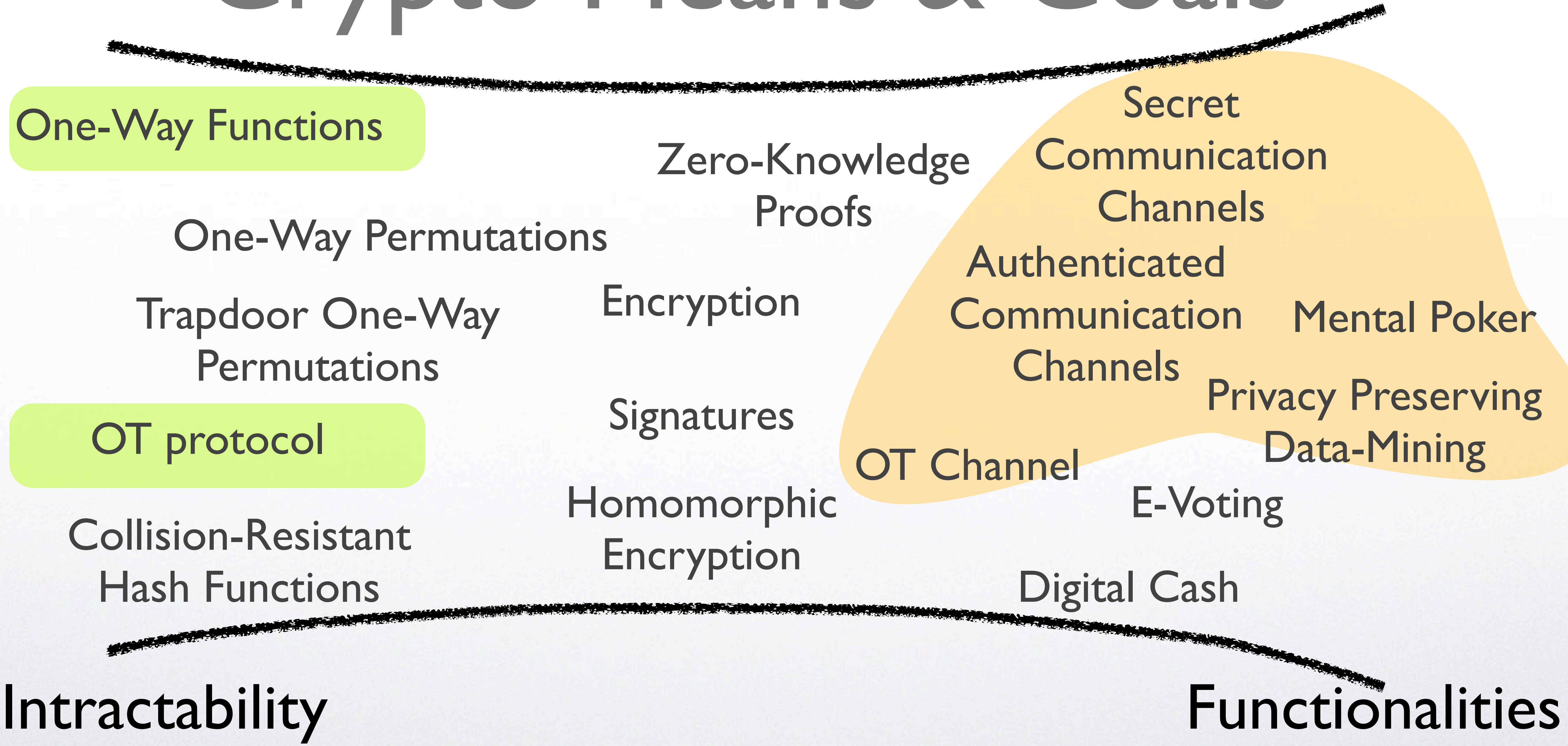
E-Voting

Digital Cash

Intractability

Functionalities

Crypto Means & Goals



One-Way Functions

One-Way Permutations

Trapdoor One-Way Permutations

OT protocol

Collision-Resistant Hash Functions

Zero-Knowledge Proofs

Encryption

Signatures

Homomorphic Encryption

Secret Communication Channels

Authenticated Communication Channels

Mental Poker

Privacy Preserving Data-Mining

OT Channel

E-Voting

Digital Cash

Intractability

Functionalities

- A Theory of Computational Intractability for Cryptography