Breaking Verifiable Delay Functions in the Random Oracle Model

Ziyi Guan

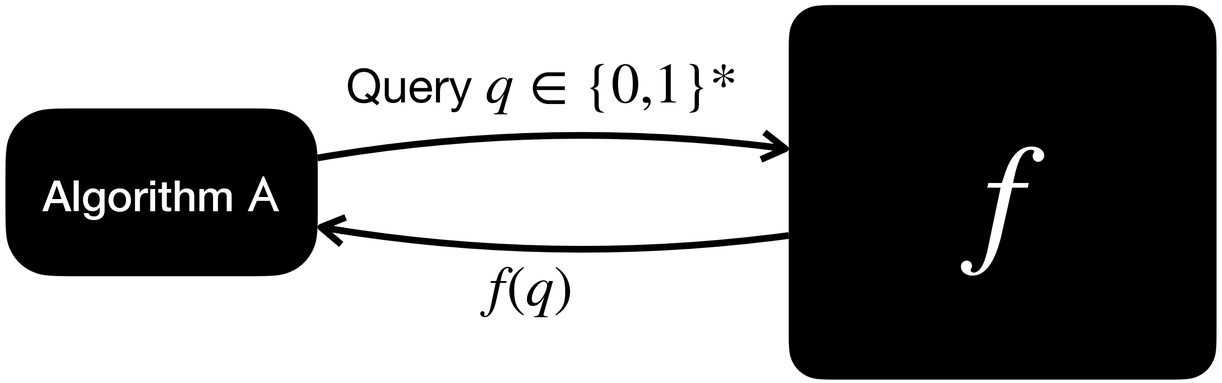
Joint work with Artur Riazanov, Weiqiang Yuan

EPFL

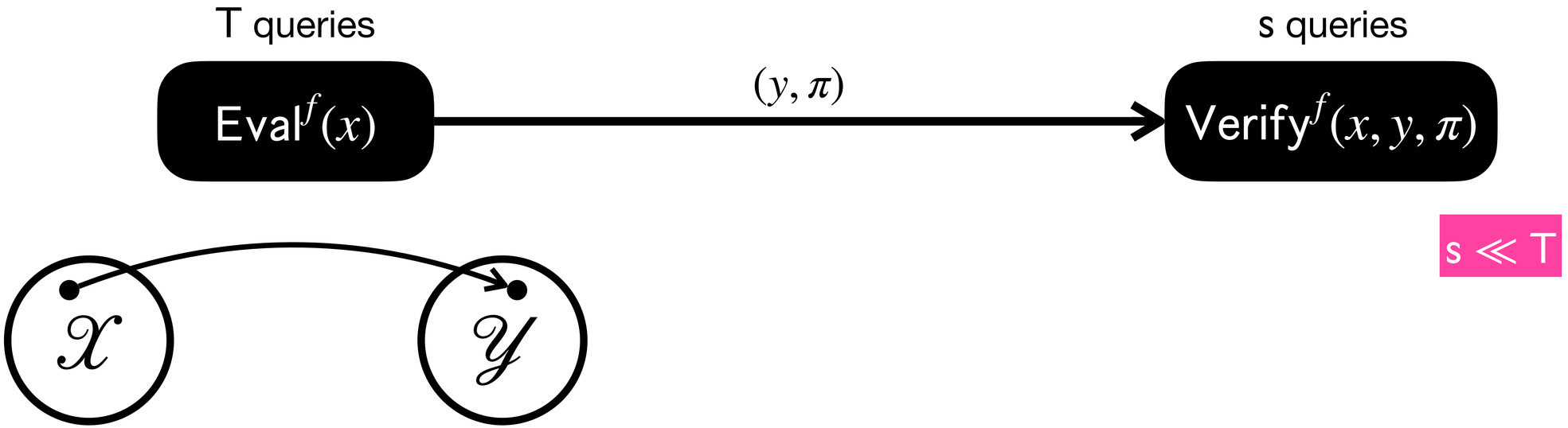
The random oracle model

Random oracle $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$

$\mathcal{O}_\ell$: uniform distribution over $f\colon \{0,1\}^* \to \{0,1\}^\ell$



Verifiable Delay Function (VDF)



$s \ll T$

VERIFIABLE → correctness of output efficiently publicly verifiable

DELAY

→ Can be evaluated in T queries

→ Cannot be evaluated in $o(T)$ rounds of queries

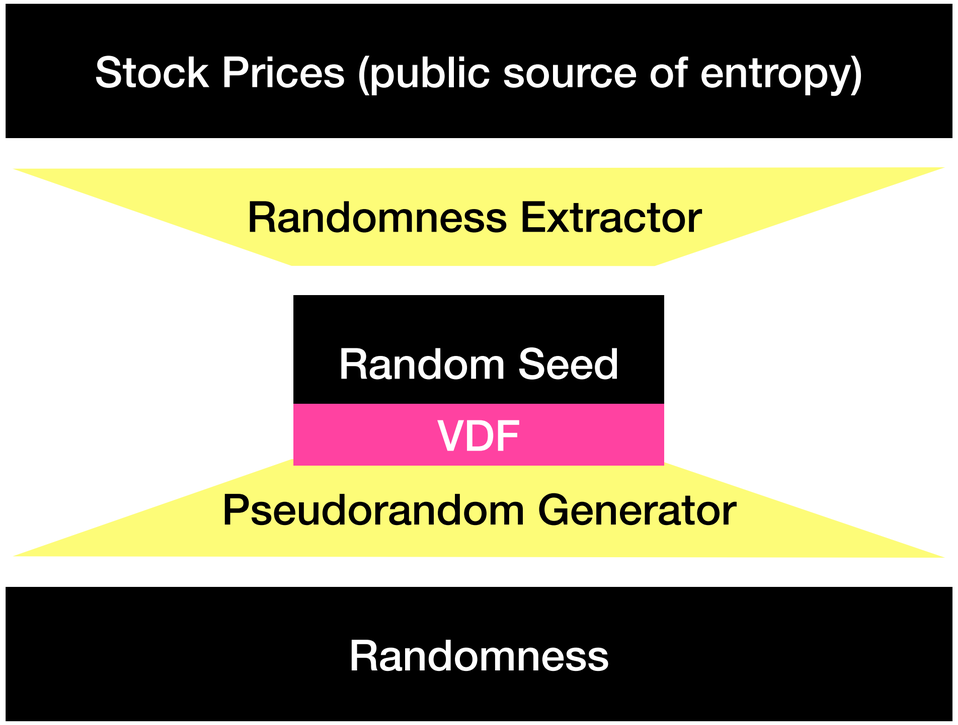
One can make multiple non-adaptive queries in one round

FUNCTION → one **unique** output

Why study VDF?

Randomness beacon

- Publish randomness regularly
- Cannot predict/manipulate



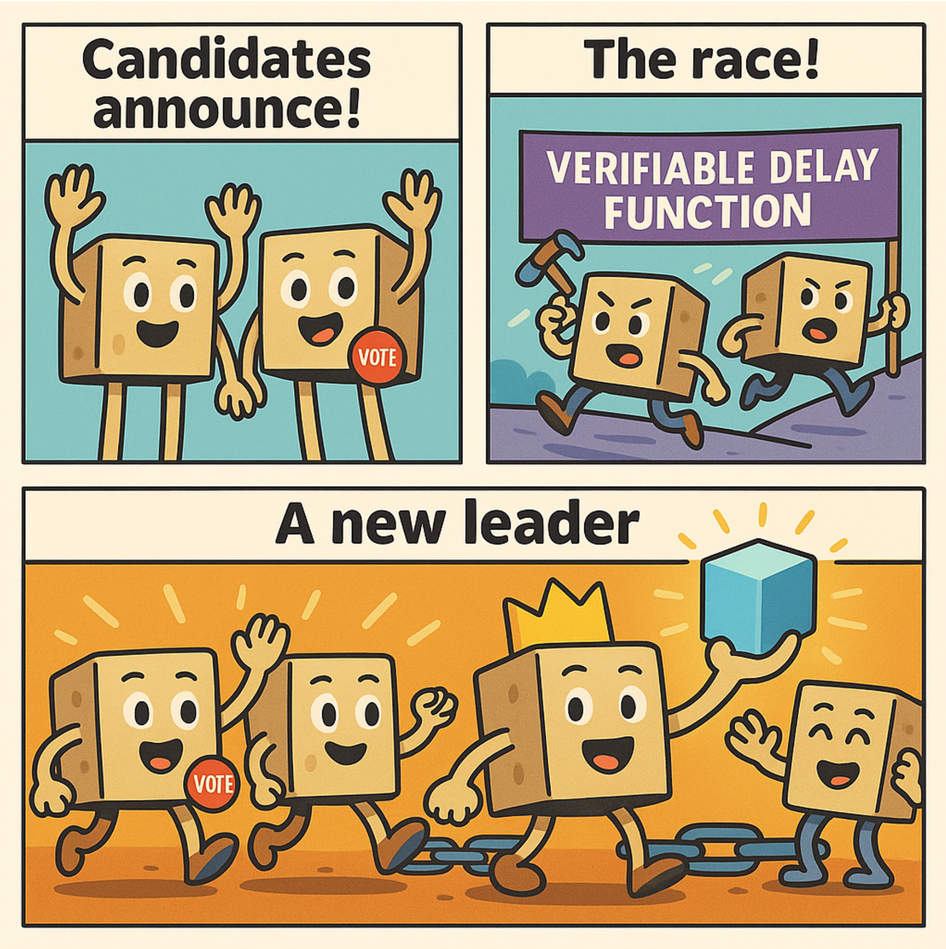
ISSUE: final randomness easy to compute & manipulate
(Stock prices can be biased/manipulated)

DELAY: not computable before market closes

UNIQUENESS: no ambiguity on output

Why study VDF?

Blockchain: leader election



UNIQUENESS → one **unique** leader

DELAY → cannot predict the next leader until shortly before the announcement

SOLANA Tezos chia IOTA

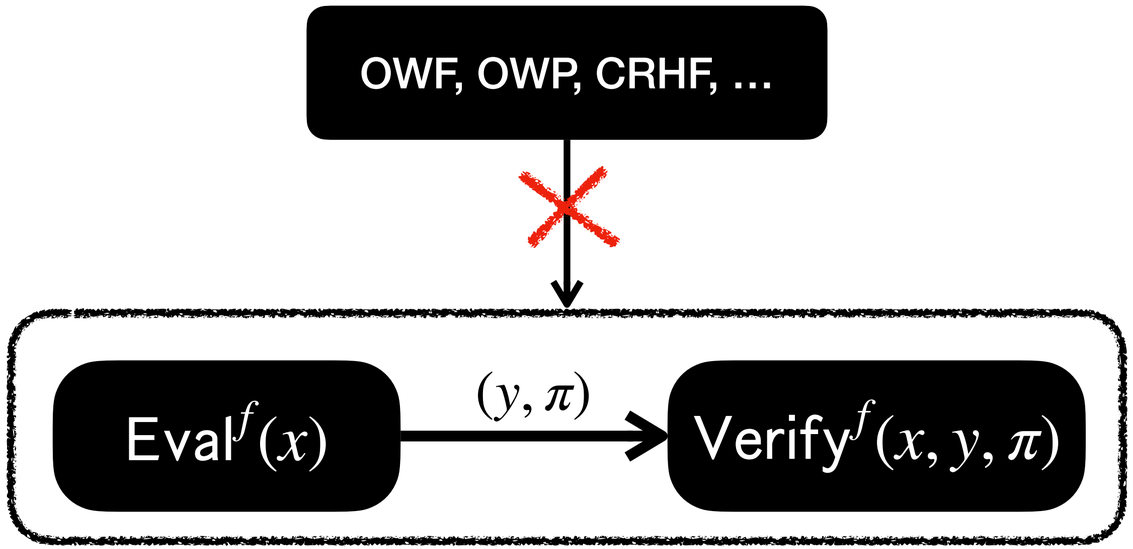
Verifiable Delay Functions Do Not Exist in the Random Oracle Model!!!!!!

But, VDFs exist in the standard model…?

Why do we care about ROM? It's not real anyway

What cryptography is needed for VDF constructions?

Existing VDFs rely on algebraic assumptions — not post-quantum secure



Complex assumptions (e.g. lattice) necessary for post-quantum VDF

But, VDFs exist in the standard model…?

Why do we care about ROM? It's not real anyway

What security do VDF constructions have?

Standard model construction do not give concrete security analysis…
How to set security parameters in practice?

Incrementally Verifiable Computation
(Believed to not exist in the ROM)

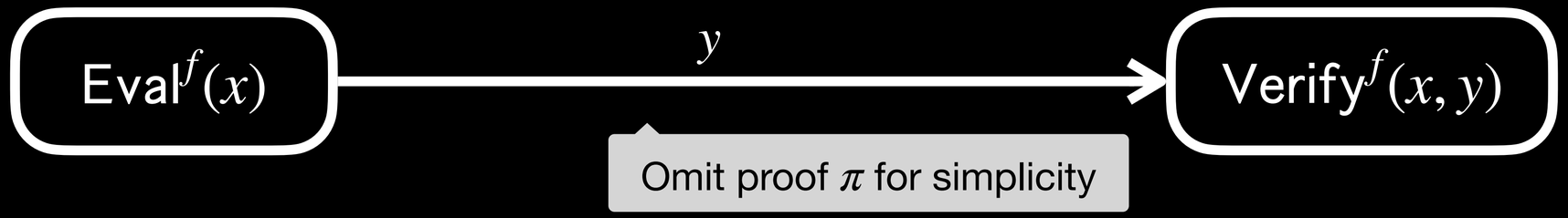
Alternative idealized model (LDROM, AROM)

Non-succinct IVC in the ROM

…

Similar approach for VDFs?

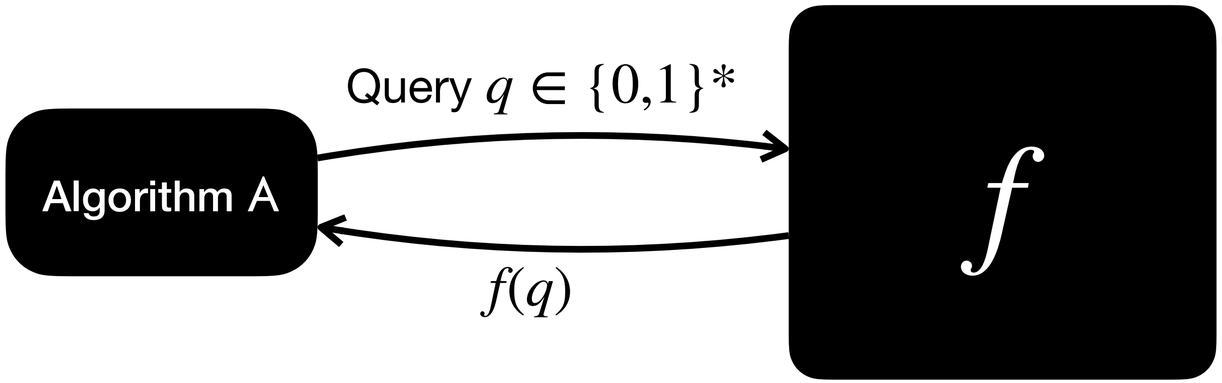
Warm-up: perfect VDF

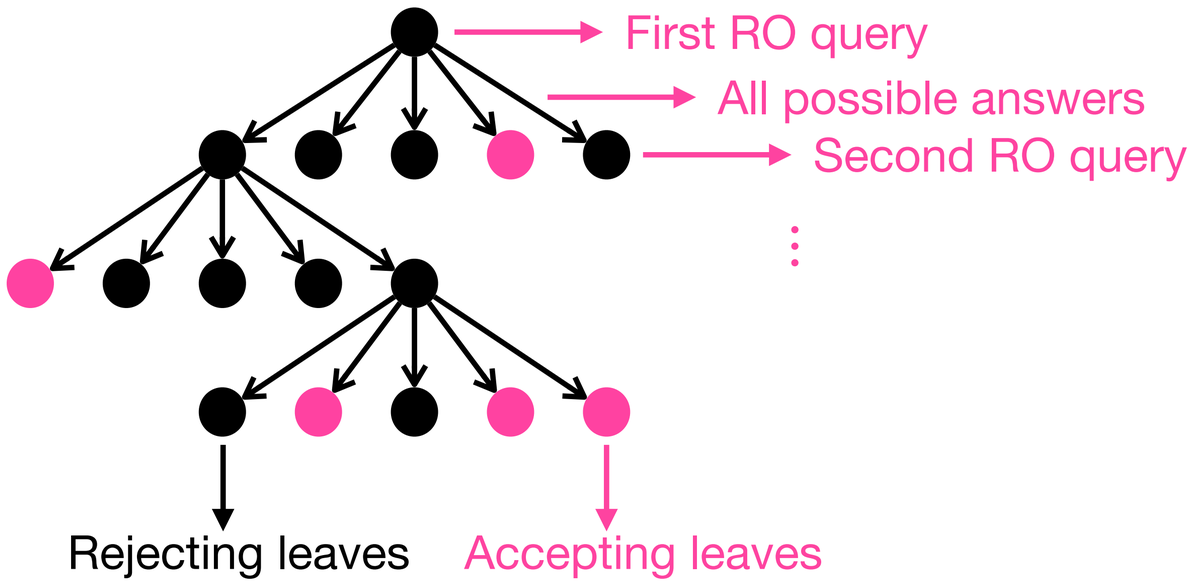


Perfect correctness:
$\rightarrow \forall f, \forall x, \mathsf{Verify}^f(x, \mathsf{Eval}^f(x)) = 1$

Perfect uniqueness:
$\rightarrow \forall f, \forall x, \exists\, !y, \mathsf{Verify}^f(x, y) = 1$

Decision tree algorithms in the ROM



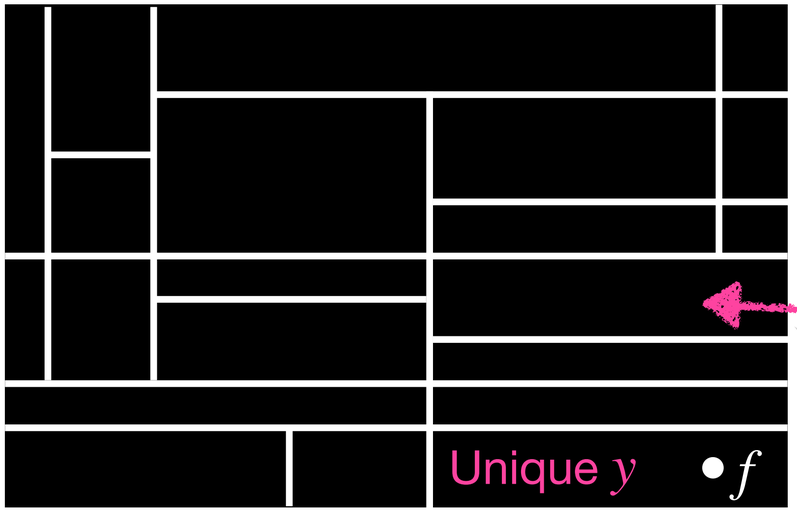


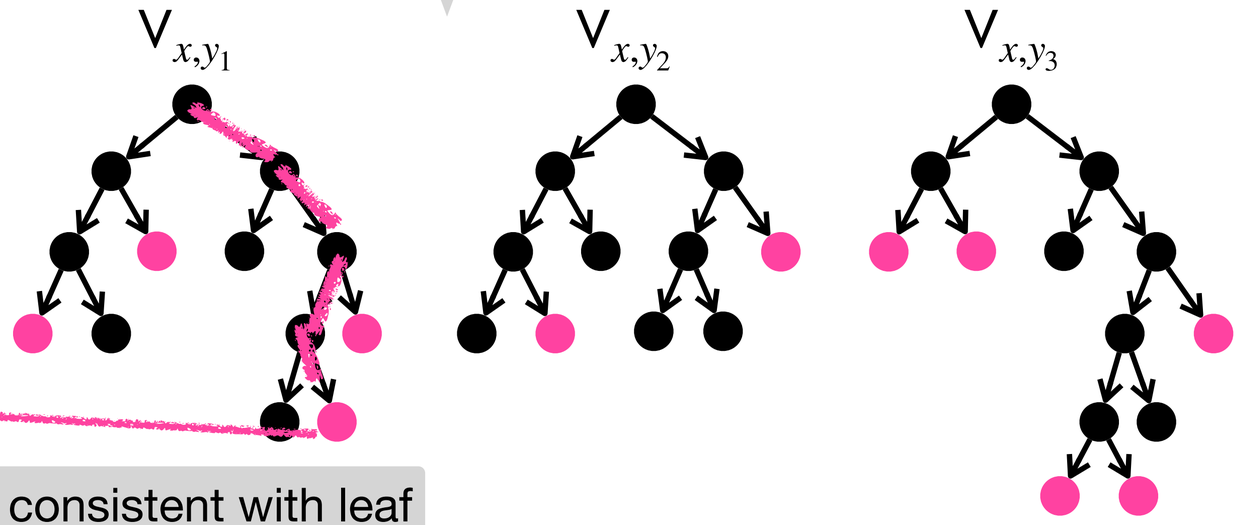
$\mathsf{V}_{x,y}(f) := \mathsf{Verify}^f(x, y)$

Accepting leaves of $\mathsf{V}_{x,y}$ partitions random oracles

Leaf: a partial RO with at most s locations fixed

Set of all RO functions $\{0,1\}^*$





All RO consistent with leaf

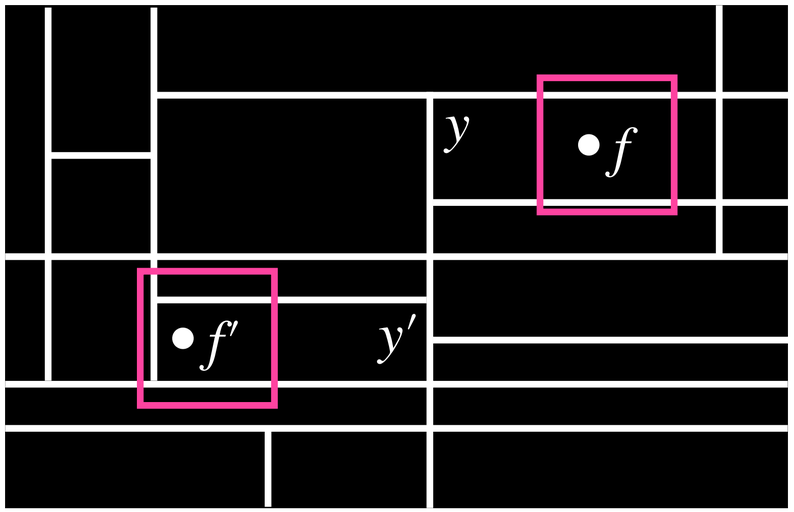
Perfect correctness $\rightarrow \forall f, \forall x, \mathsf{Verify}^f(x, \mathsf{Eval}^f(x)) = 1$

Perfect uniqueness $\rightarrow \forall f, \forall x, \exists !y, \mathsf{Verify}^f(x, y) = 1$

i.e. the boxes are disjoint!

$\Longrightarrow$ learning all queries in a box can uniquely determine y

Adversary breaking “DELAY”



$\mathsf{Adv}^{f}(x)$:

1. Initialize an empty evaluation table of f.
2. Repeat for $2s + 1$ times:
   a. Pick $f' \in \{0,1\}^*$ consistent with current table of f.
   b. Compute $y' := \mathsf{Eval}^{f'}(x)$.
   c. Let $\mathsf{Q}_{\mathsf{Eval}}(f', x)$ be the query set of $\mathsf{Eval}^{f'}(x)$.
   d. Query f with $\mathsf{Q}_{\mathsf{Eval}}(f', x)$ in one round.
3. Output the majority of y'.

$2s + 1$ rounds of queries
$\leq$ T queries each round

Queries by $\mathsf{Verify}^{f}(x, y)$

- $\forall i \in [2s + 1]$ such that $y' \neq y$, Adv queries at least one new position in $\mathsf{Q}_{\mathsf{V}}(f, x, y)$
  - Otherwise:
    - $\mathsf{Verify}^{f}(x, y') = \mathsf{Verify}^{f'}(x, y') = 1$
    - $\mathsf{Verify}^{f}(x, y) = 1$ by perfect correctness

Contradicting perfect uniqueness!!!

- $|\mathsf{Q}_{\mathsf{Verify}}(f, x, y)| \leq s \implies$ At most s iterations such that $y' \neq y \implies$ majority would output y

How about computational uniqueness?

Perfect correctness:
$\rightarrow \forall f, \forall x, \mathsf{Verify}^f(x, \mathsf{Eval}^f(x)) = 1$

The proof works for imperfect correctness…
For simplicity, we consider only perfect correctness

~~**Perfect uniqueness:**~~
~~$\rightarrow \forall f, \forall x, \exists! y, \mathsf{Verify}^f(x, y) = 1$~~

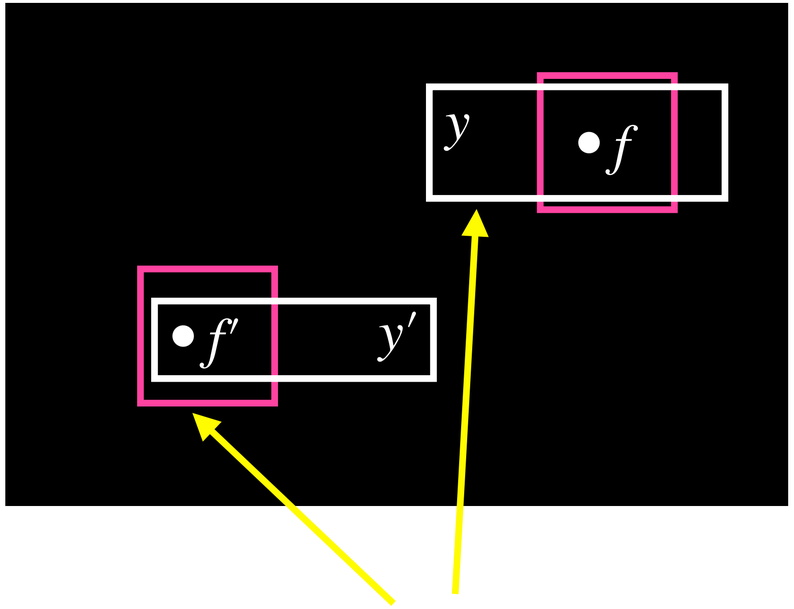
Every efficient Adv can find non-unique solutions for at most $\mathsf{negl}(\lambda)$ fraction of $f \in \mathcal{O}_\lambda$

Computational uniqueness:

$\rightarrow$ for every $\mathsf{poly}(\lambda, T)$-query Adv, $\Pr\left[\begin{array}{l} y' \neq \mathsf{Eval}^f(x) \\ \wedge\ \mathsf{Verify}^f(x, y') = 1 \end{array} \middle| \begin{array}{r} f \leftarrow \mathcal{O}_\lambda \\ x \leftarrow \mathcal{X}_f \\ y' \leftarrow \mathsf{Adv}^f(x) \end{array}\right] \leq \mathsf{negl}(\lambda)$

How does previous adversary fail?

Perfect uniqueness



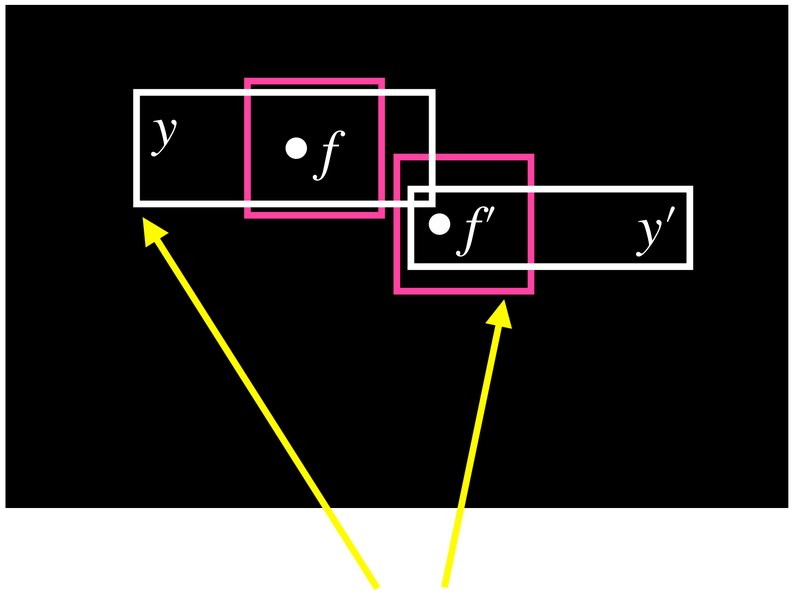
Disjoint: can learn new location

$y \neq y'$

$\Longrightarrow$ $\mathsf{Q}_{\mathsf{Eval}}(f', x)$ contain a new query in $\mathsf{Q}_{\mathsf{V}}(f, x, y)$

Otherwise, $\mathsf{Verify}^{f}(x, y') = \mathsf{Verify}^{f'}(x, y') = 1$, contradicting perfect uniqueness

Computational uniqueness



Might intersect: cannot learn new location

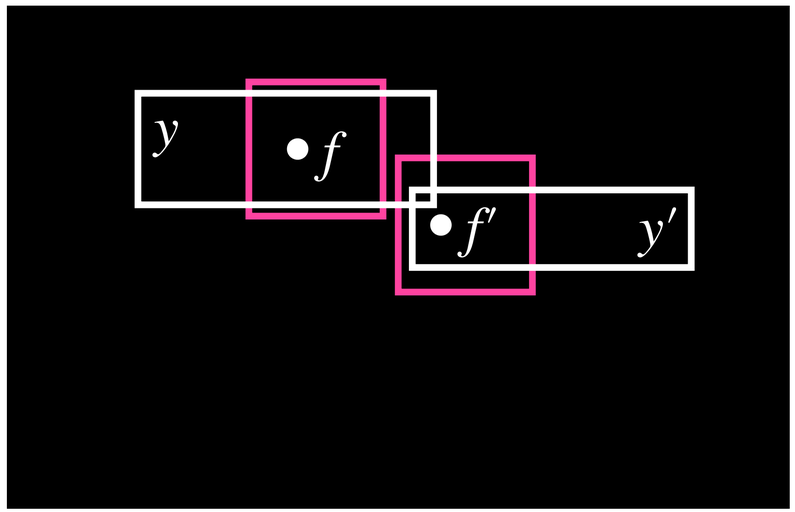
$y \neq y'$

$\not\Longrightarrow$ $\mathsf{Q}_{\mathsf{Eval}}(f', x)$ contain a new query in $\mathsf{Q}_{\mathsf{V}}(f, x, y)$

Since $\mathsf{Verify}^{f}(x, y') = \mathsf{Verify}^{f'}(x, y') = 1$ **doesn't** contradict computational uniqueness

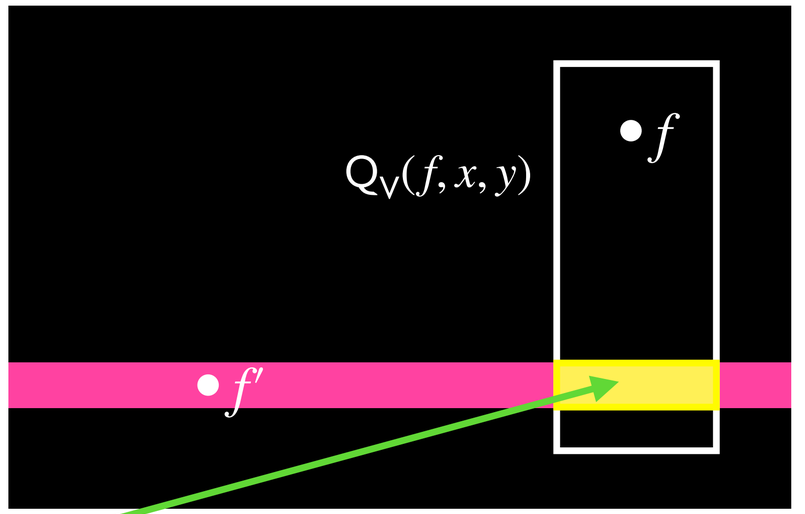
Intersection $\Longrightarrow$ uniqueness breaker

Set of all RO functions $\{0,1\}^*$



A different view

$\{0,1\}^*$ sorted by positions in $\mathsf{Q}_{\mathsf{Eval}}(f', x)$



$\mathsf{Q}_{\mathsf{Eval}}(f', x)$

Oracles consistent with f' on $\mathsf{Q}_{\mathsf{Eval}}(f', x)$

$\{0,1\}^* \setminus \mathsf{Q}_{\mathsf{Eval}}(f', x)$

Oracles consistent with
- f' on $\mathsf{Q}_{\mathsf{Eval}}(f', x)$
- f on $\mathsf{Q}_{\mathsf{V}}(f, x, y)$

i.e., Oracles that break uniqueness!

Row: an answer set for $\mathsf{Q}_{\mathsf{Eval}}(f', x)$

Column: an answer set for $\{0,1\}^* \setminus \mathsf{Q}_{\mathsf{Eval}}(f', x)$

Coupling with a uniqueness breaker

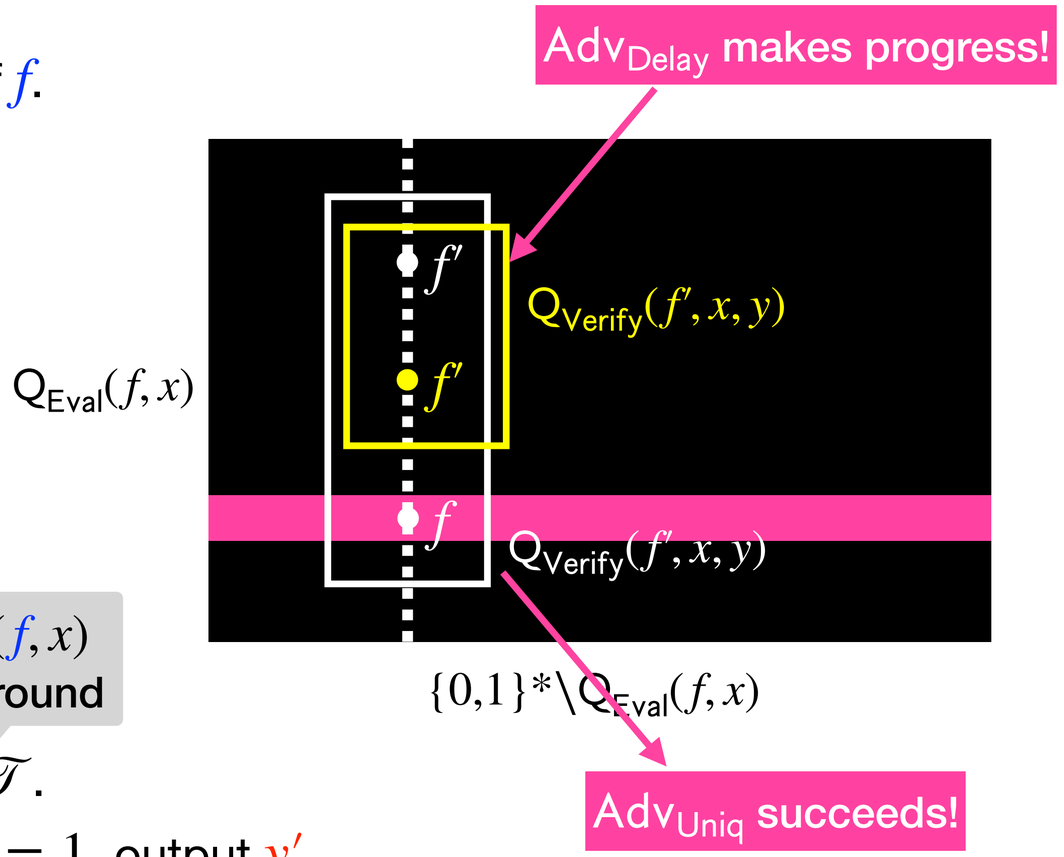
$\mathsf{Adv}^{f}_{\mathsf{Delay}}(x)$:

1. Initialize an empty evaluation table of f.
2. *Repeat for* $3s$ *times*:
   a. *Sample* $f' \leftarrow \{0,1\}^*$ consistent with current table of f.
   b. Compute $y' := \mathsf{Eval}^{f'}(x)$.
   c. Let $\mathsf{Q}_{\mathsf{Eval}}(f', x)$ be the query set of $\mathsf{Eval}^{f'}(x)$.
   d. Query f with $\mathsf{Q}_{\mathsf{Eval}}(f', x)$ in one round.
3. Output the majority of y'.

$\mathsf{Adv}^{f}_{\mathsf{Uniq}}(x)$:

1. Initialize an empty evaluation table $\mathcal{T}$ of f.
2. Compute $y := \mathsf{Eval}^{f}(x)$.
3. *Repeat for* $3s$ *times*
   a. *Sample* $f' \leftarrow \{0,1\}^*$ that agrees with f on $\{0,1\}^* \setminus \mathcal{T}$.
   b. Compute $y' := \mathsf{Eval}^{f'}(x)$. If $y' \neq y$ and $\mathsf{Verify}^{f}(x, y') = 1$, output y'.
   c. Sample $f'' \leftarrow \{0,1\}^*$ consistent with current view of f and query f with $\mathsf{Q}_{\mathsf{Eval}}(f'', x)$ in one round.

$\mathcal{T} = \mathsf{Q}_{\mathsf{Eval}}(f, x)$ for the first round



Stronger impossibility for perfect VDF

Perfect correctness:
$\rightarrow \forall f, \forall x, \mathsf{Verify}^f(x, \mathsf{Eval}^f(x)) = 1$

Perfect uniqueness:
$\rightarrow \forall f, \forall x, \exists\, !y, \mathsf{Verify}^f(x, y) = 1$

Query complexity vs. certificate complexity

Search problem $S \subseteq F \times Y$

- Non-deterministic verifiers $\{\mathsf{V}_y(f)\}_{y \in Y}$: $\forall f \in F, y \in Y, (f, y) \in S$ iff $\mathsf{V}_y(f) = 1$

Query complexity $D(S)$: min # bits in f needed to compute y s.t. $(f, y) \in S$

Certificate complexity

- $C(f, y)$: min # bits in f needed to prove $(f, y) \in S$
- $C(S) = \max_{f \in F} \min_{y \in S(f)} C(f, y)$

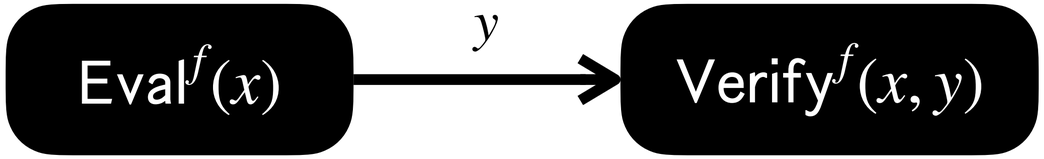
Trivial: $C(S) \leq D(S)$

$D(S) \leq C(S)^2$

e.g. OR $\subseteq \{0,1\}^n \times \{\perp, 0, \ldots, n-1\}$

- For $a \in \{0,1\}^n$:
  - If $a = 0^n$, $(a, \perp) \in$ OR
  - Otherwise, $(a, i) \in$ OR where $a[i] = 1$
- $D(\text{OR}) = n$
- For $i \neq \perp$, $C(a, i) = 1$
- $C(a, \perp) = n$
- $C(\text{OR}) = n$

Verifiable ~~Delay~~ Function



$\text{VDF} \subseteq \{0,1\}^* \times \mathcal{Y}$

$\mathsf{V}_{x,y}(f) := \text{Verify}^f(x, y)$

Search problem $S \subseteq F \times Y$:

- Non-deterministic verifiers $\{\mathsf{V}_y(f)\}_{y \in Y}$: $\forall f \in F, y \in Y, (f, y) \in S$ iff $\mathsf{V}_y(f) = 1$

Query complexity $D(S)$: min # bits in f needed to compute y s.t. $(f, y) \in S$

Certificate complexity

- $C(f, y)$: min # bits in f needed to prove $(f, y) \in S$
- $C(S) = \max_{f \in F} \min_{y \in S(f)} C(f, y)$ → s (# queries to f made by $\mathsf{V}_{x,y}(f)$)

$$D(S) \leq C(S)^2$$

$\Longrightarrow$ ∃ algorithm with at most s^2 queries that computes Eval
(Almost as efficient as Verify)

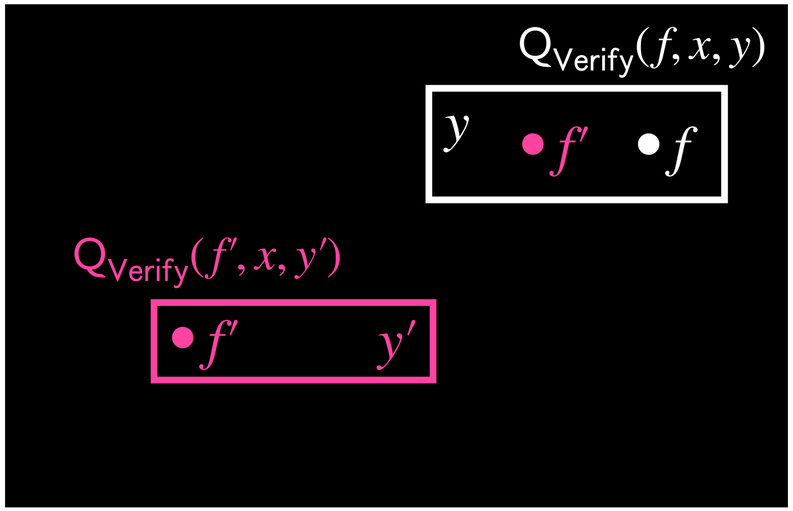
Our adversary computing Eval

$\mathsf{Adv}^{f}(x)$:

1. Initialize an empty evaluation table of f.
2. For $i \in [s]$:
   a. Pick $f' \in \{0,1\}^*$ consistent with current f evaluation table.
   b. Compute $y' := \mathsf{Eval}^{f'}(x)$.
   c. Query f with $\mathsf{Q}_{\mathsf{V}}(f', x, y')$ in one round.
3. Output $y := \mathsf{Eval}^{f^\star}(x)$, where $f^\star$ is the current evaluation table of f.

s rounds of queries
At most s queries each round

No queries here!

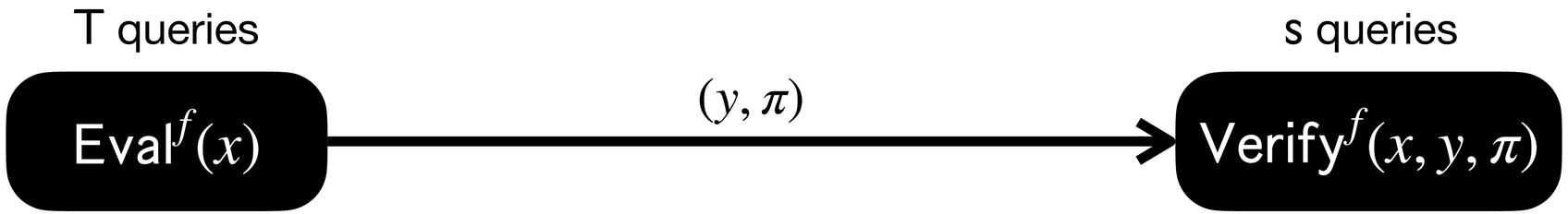


Adv learns at least one query in $\mathsf{Q}_{\mathsf{Verify}}(f, x, y)$:

- $y' = y$: amazing!
- $y' \neq y$: $\exists q \in \mathsf{Q}_{\mathsf{Verify}}(f', x, y') \cap \mathsf{Q}_{\mathsf{Verify}}(f, x, y), f'[q] \neq f[q]$

(Otherwise, $\mathsf{Verify}^{f}(x, y') = \mathsf{Verify}^{f'}(x, y') = 1$, contradicting uniqueness!)

Verifiable functions do not exist in the ROM



$s \ll T$

VERIFIABLE → correctness of output efficiently publicly verifiable

~~**DELAY**~~

~~→ Can be evaluated in T queries~~

~~→ Cannot be evaluated in $o(T)$ rounds of queries~~

FUNCTION

→ Perfect uniqueness: $\forall f, \forall x, \exists\, !y, \mathsf{Verify}^f(x, y) = 1$

Generalization

Verifiable ~~Delay~~ Functions with

Imperfect Correctness and Statistical Uniqueness

Do Not Exist in the Random Oracle Model!!!!!!

$$\Pr\left[\mathsf{Verify}^f(x,y)=1 \;\middle|\; \begin{array}{r} f \leftarrow \mathcal{O}_\lambda \\ x \leftarrow \mathcal{X}_f \\ y \leftarrow \mathsf{Eval}^f(x) \end{array}\right] \geq 1-\mathsf{negl}(\lambda)$$

For every (unbounded query) Adv, $\Pr\left[\begin{array}{l} y' \neq \mathsf{Eval}^f(x) \\ \wedge\, \mathsf{Verify}^f(x,y')=1 \end{array} \;\middle|\; \begin{array}{r} f \leftarrow \mathcal{O}_\lambda \\ x \leftarrow \mathcal{X}_f \\ y' \leftarrow \mathsf{Adv}^f(x) \end{array}\right] \leq \mathsf{negl}(\lambda)$

Recap

Verifiable Delay Functions
Do Not Exist in the Random Oracle Model!!!!!!

Random Oracle Model	Delay	No Delay
Perfectly Unique	Impossible ❌	Impossible ❌
Statistically Unique		
Computationally Unique		???
Non-Unique	Proof of sequential work ✅	Proof of work ✅

Thank you!