On Building Fine-Grained One-Way Functions from Strong Average-Case Hardness

Chris Brzuska, Geoffroy Couteau

Aalto University

cnrs

irif

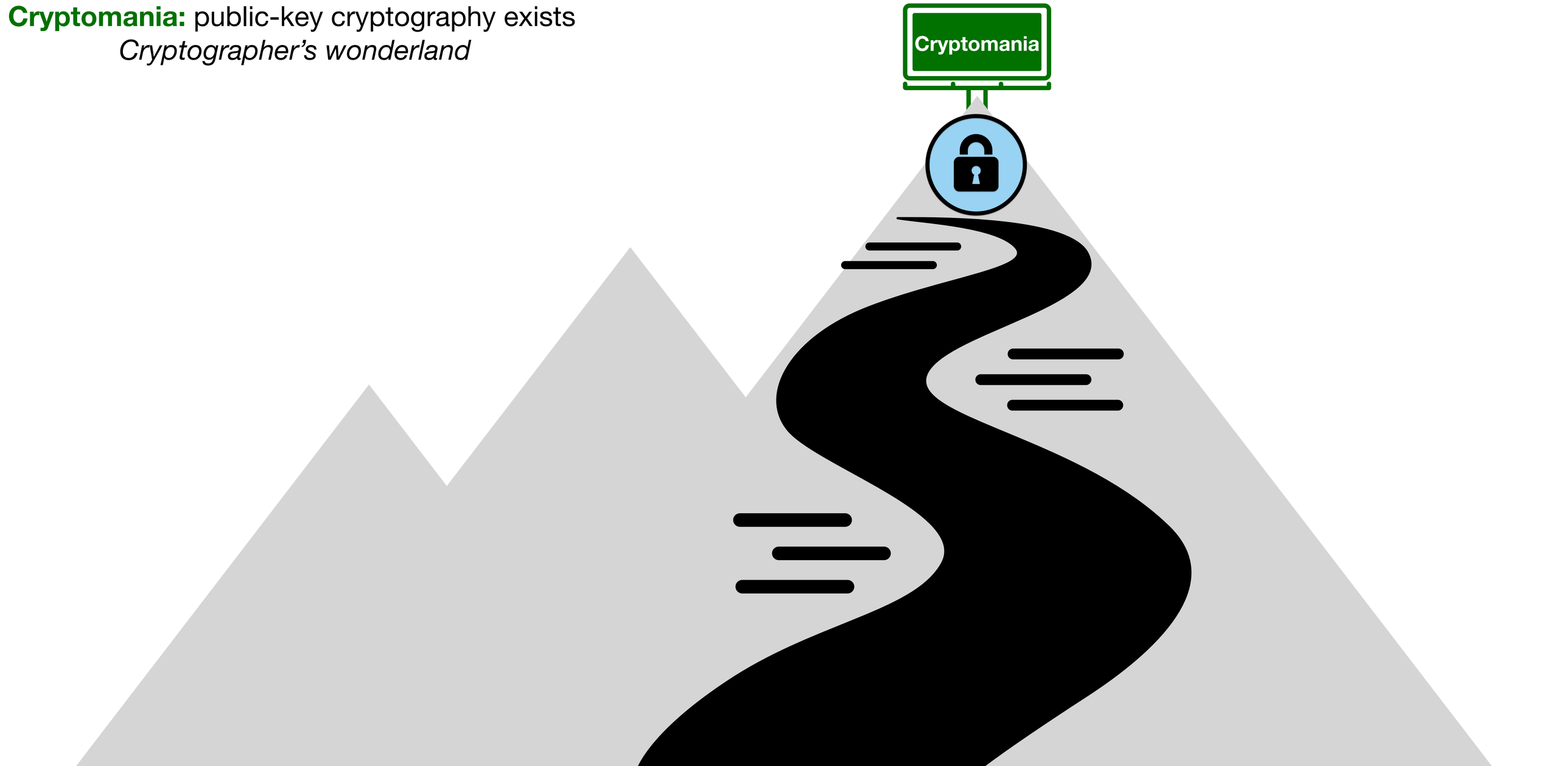
Université de Paris

The Five Worlds of Impagliazzo

Impagliazzo, 1995: what world do we live in?

The Five Worlds of Impagliazzo

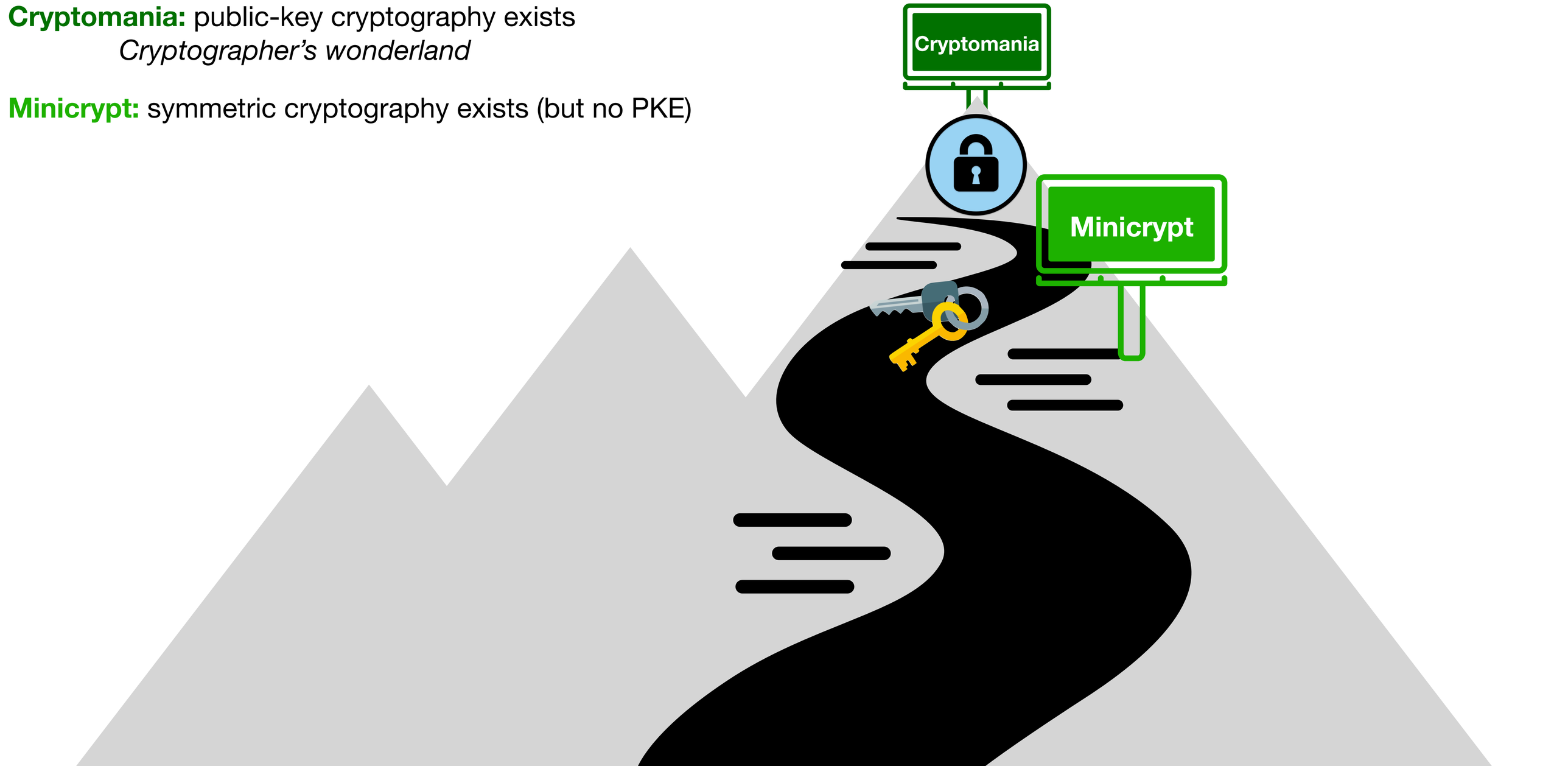
Cryptomania: public-key cryptography exists
Cryptographer's wonderland



The Five Worlds of Impagliazzo

Cryptomania: public-key cryptography exists
Cryptographer's wonderland

Minicrypt: symmetric cryptography exists (but no PKE)

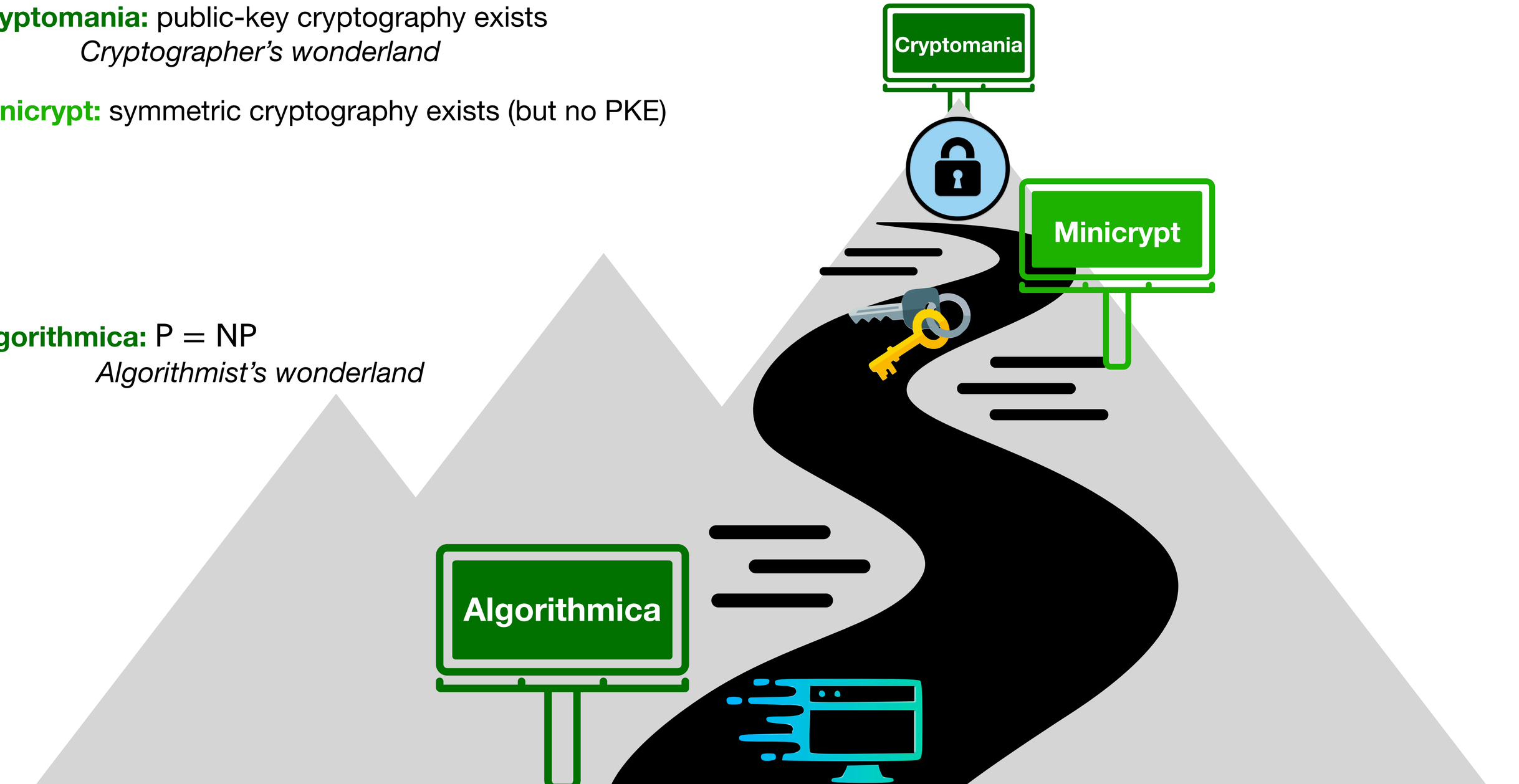


The Five Worlds of Impagliazzo

Cryptomania: public-key cryptography exists
Cryptographer's wonderland

Minicrypt: symmetric cryptography exists (but no PKE)

Algorithmica: P = NP
Algorithmist's wonderland



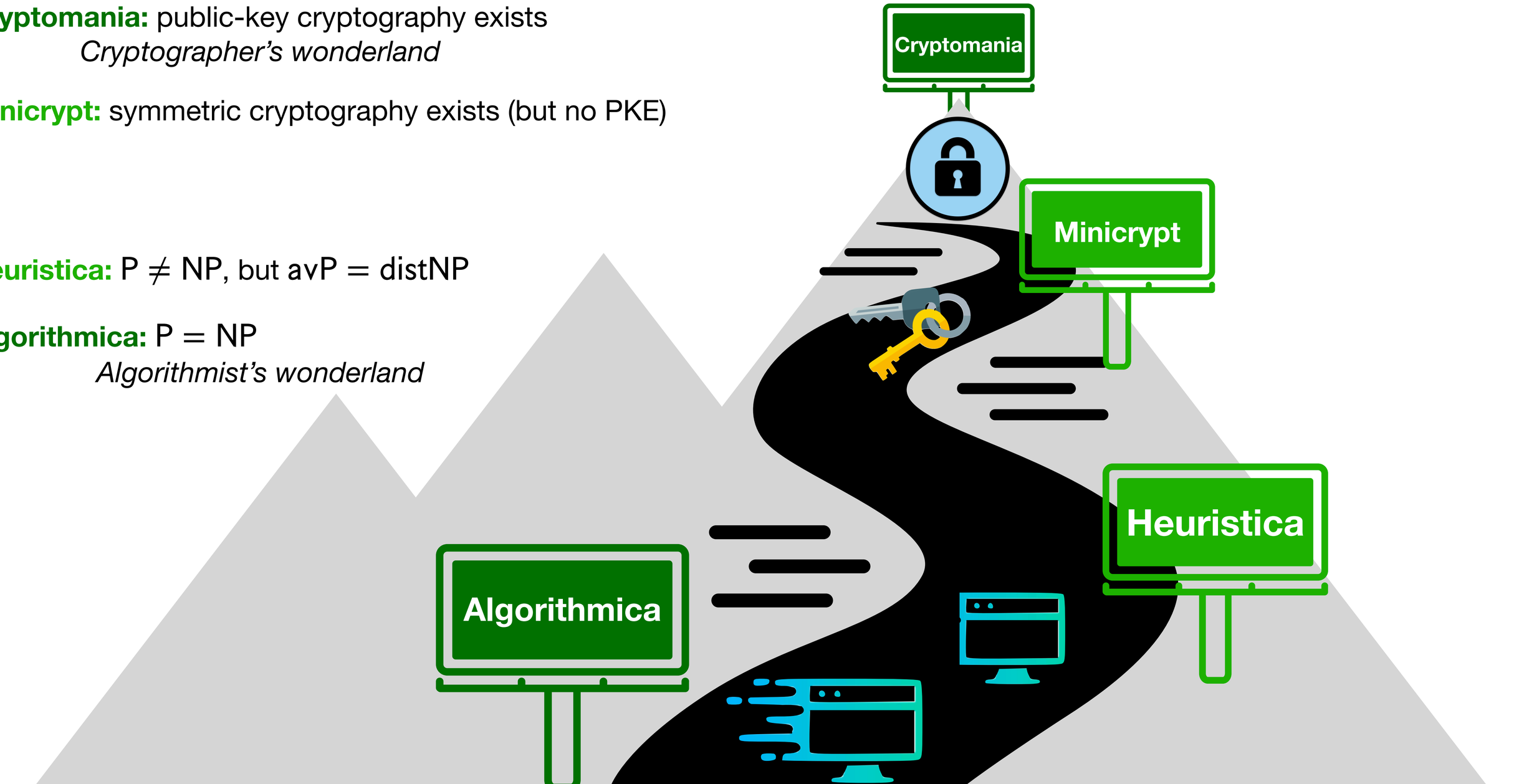
The Five Worlds of Impagliazzo

Cryptomania: public-key cryptography exists
Cryptographer's wonderland

Minicrypt: symmetric cryptography exists (but no PKE)

Heuristica: P ≠ NP, but avP = distNP

Algorithmica: P = NP
Algorithmist's wonderland



The Five Worlds of Impagliazzo

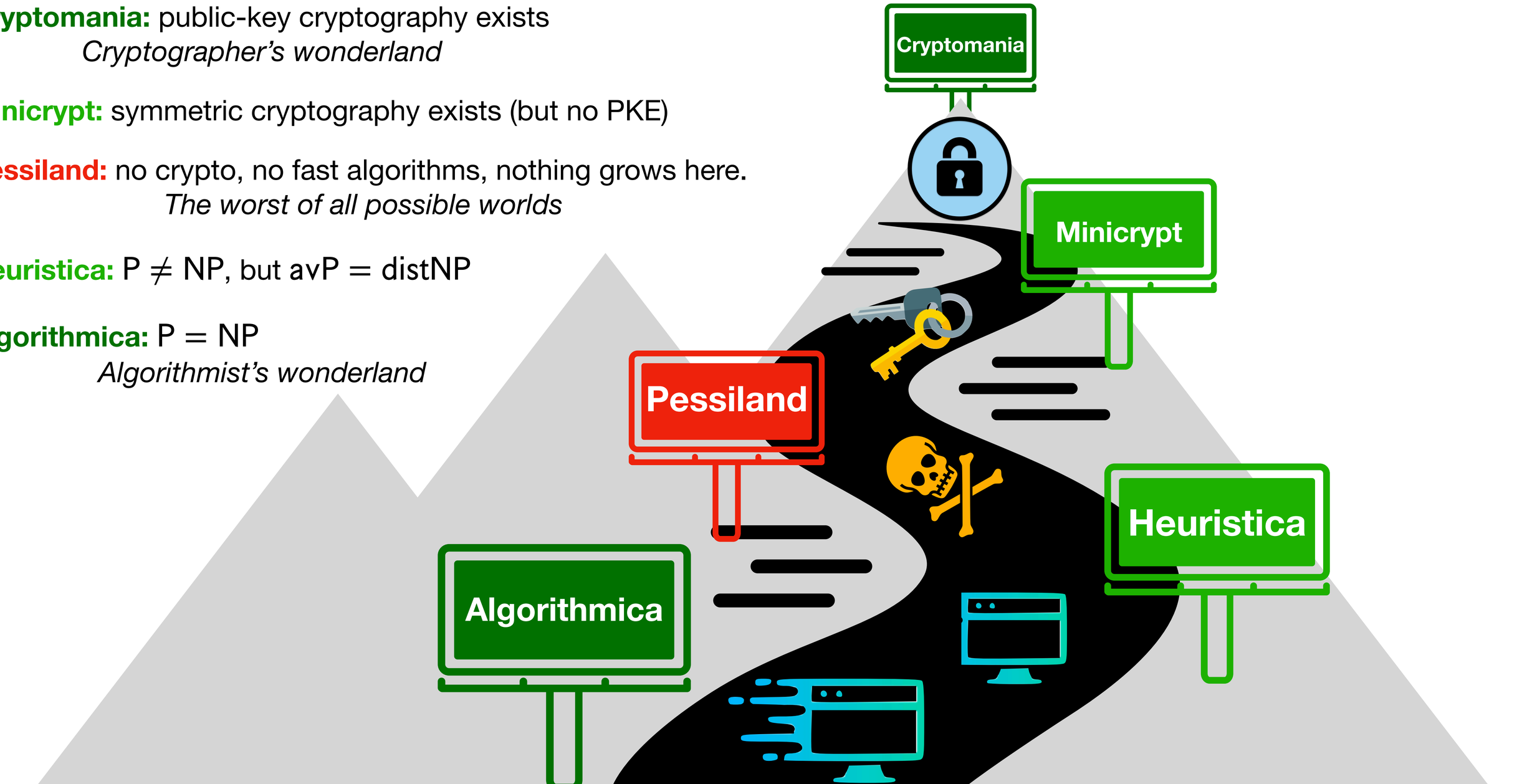
Cryptomania: public-key cryptography exists
Cryptographer's wonderland

Minicrypt: symmetric cryptography exists (but no PKE)

Pessiland: no crypto, no fast algorithms, nothing grows here.
The worst of all possible worlds

Heuristica: P ≠ NP, but avP = distNP

Algorithmica: P = NP
Algorithmist's wonderland



The Five Worlds of Impagliazzo

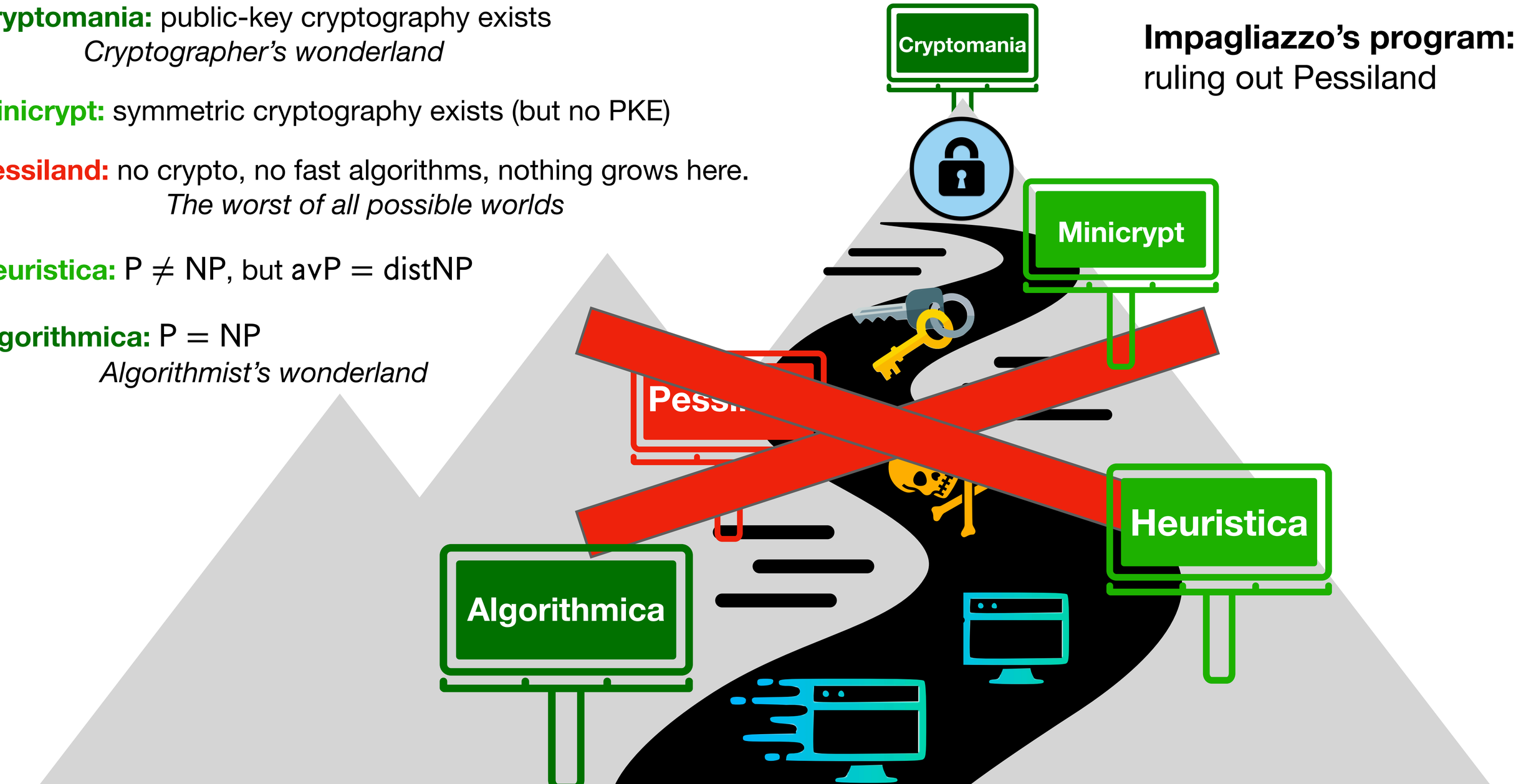
Cryptomania: public-key cryptography exists
Cryptographer's wonderland

Minicrypt: symmetric cryptography exists (but no PKE)

Pessiland: no crypto, no fast algorithms, nothing grows here.
The worst of all possible worlds

Heuristica: P ≠ NP, but avP = distNP

Algorithmica: P = NP
Algorithmist's wonderland

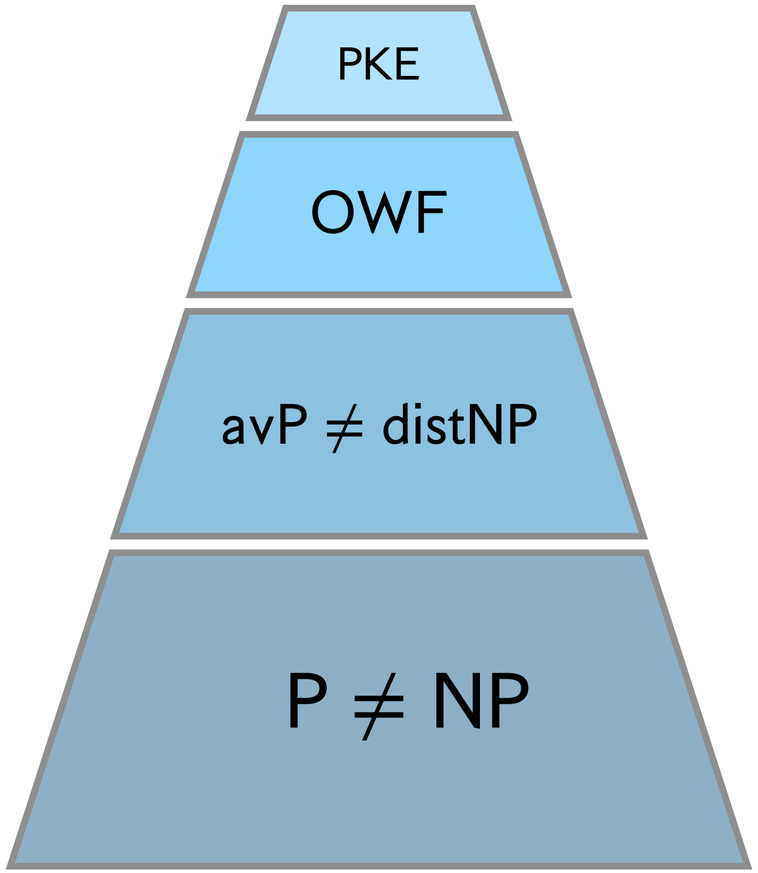


Impagliazzo's program:
ruling out Pessiland

The Five Worlds of Impagliazzo

Ruling out Pessiland would be a win-win result for humanity: either (some form of) cryptography exists, or there exists fast algorithms (on average) for all NP problems.

Alas… seems to be a very hard problem.

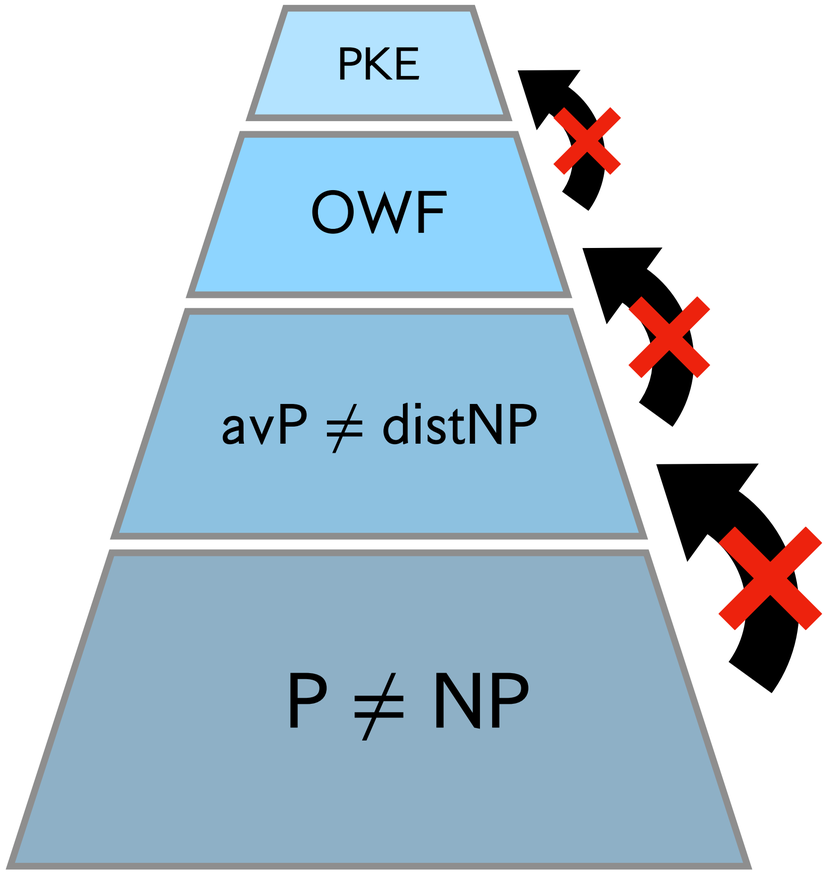


The Five Worlds of Impagliazzo

Ruling out Pessiland would be a win-win result for humanity: either (some form of) cryptography exists, or there exists fast algorithms (on average) for all NP problems.

Alas… seems to be a very hard problem.

Black-box separations between all these assumptions!



: black-box impossibility result

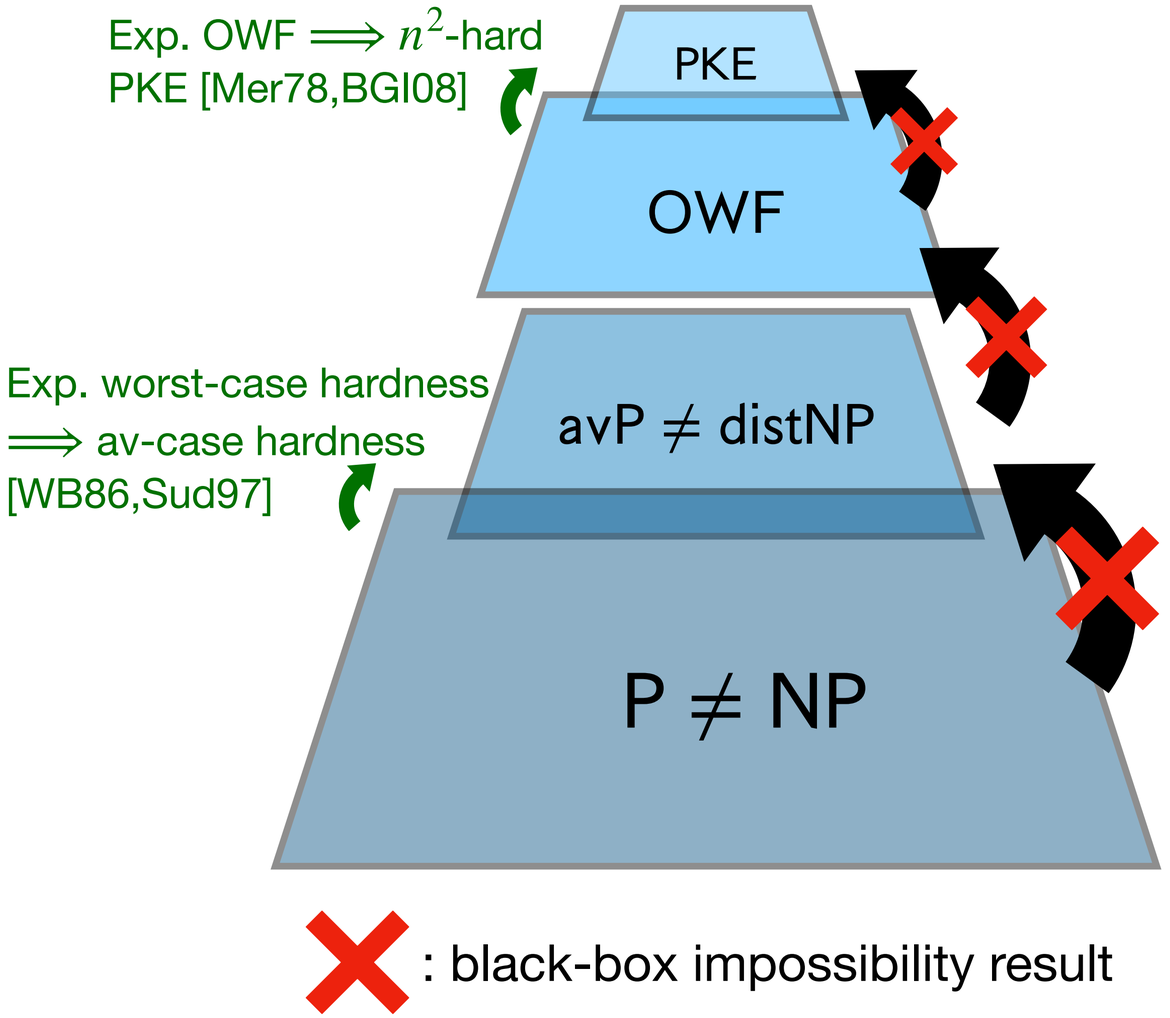
The Five Worlds of Impagliazzo

Ruling out Pessiland would be a win-win result for humanity: either (some form of) cryptography exists, or there exists fast algorithms (on average) for all NP problems.

Alas… seems to be a very hard problem.

Black-box separations between all these assumptions!

The real landscape is more subtle :)



: black-box impossibility result

The Five Worlds of Impagliazzo

Ruling out Pessiland would be a win-win result for humanity: either (some form of) cryptography exists, or there exists fast algorithms (on average) for all NP problems.

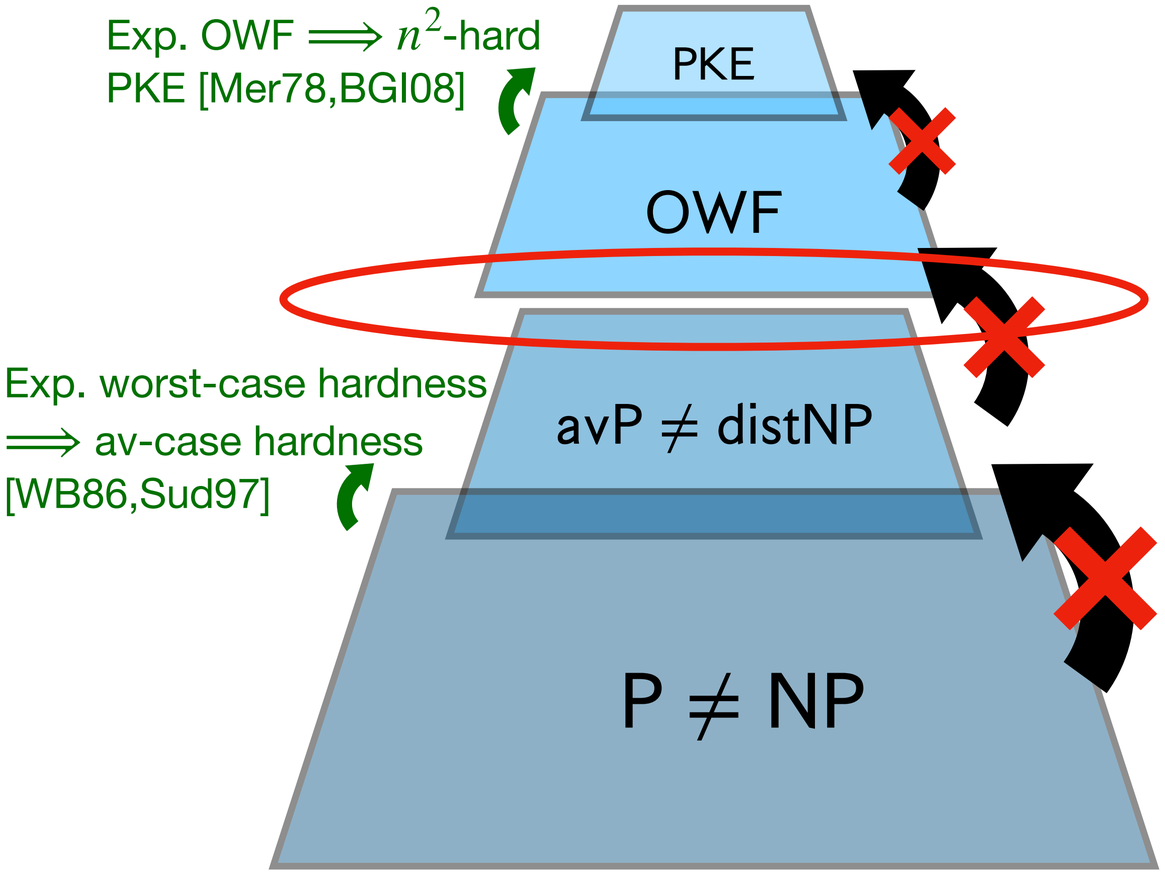
Alas… seems to be a very hard problem.

Black-box separations between all these assumptions!

The real landscape is more subtle :)

We ask:

Could *extreme* average-case hardness imply *very weak* one-way functions?



: black-box impossibility result

The Five Worlds of Impagliazzo

Ruling out Pessiland would be a win-win result for humanity: either (some form of) cryptography exists, or there exists fast algorithms (on average) for all NP problems.

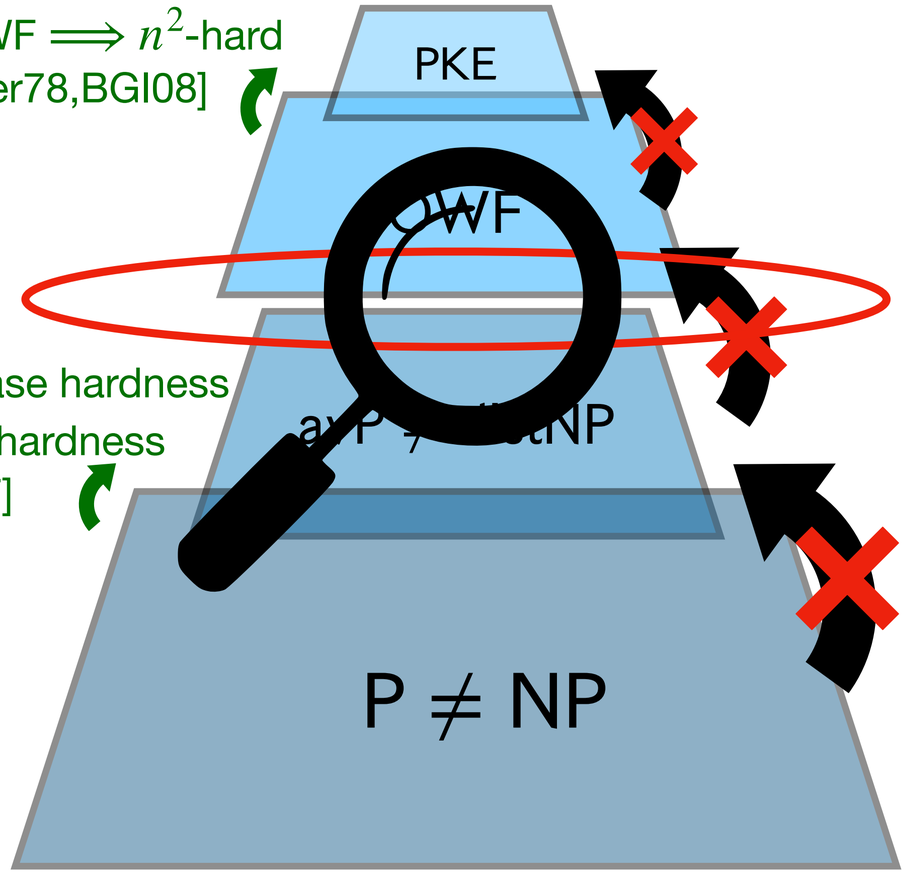
Alas… seems to be a very hard problem.

Black-box separations between all these assumptions!

The real landscape is more subtle :)

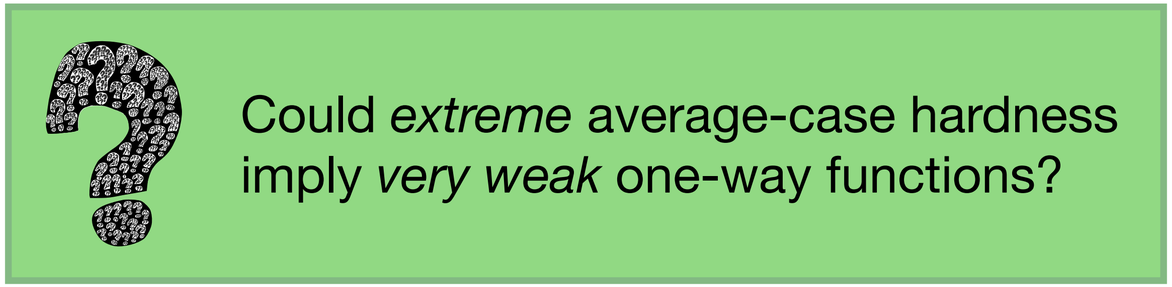
We ask:

Could *extreme* average-case hardness imply *very weak* one-way functions?

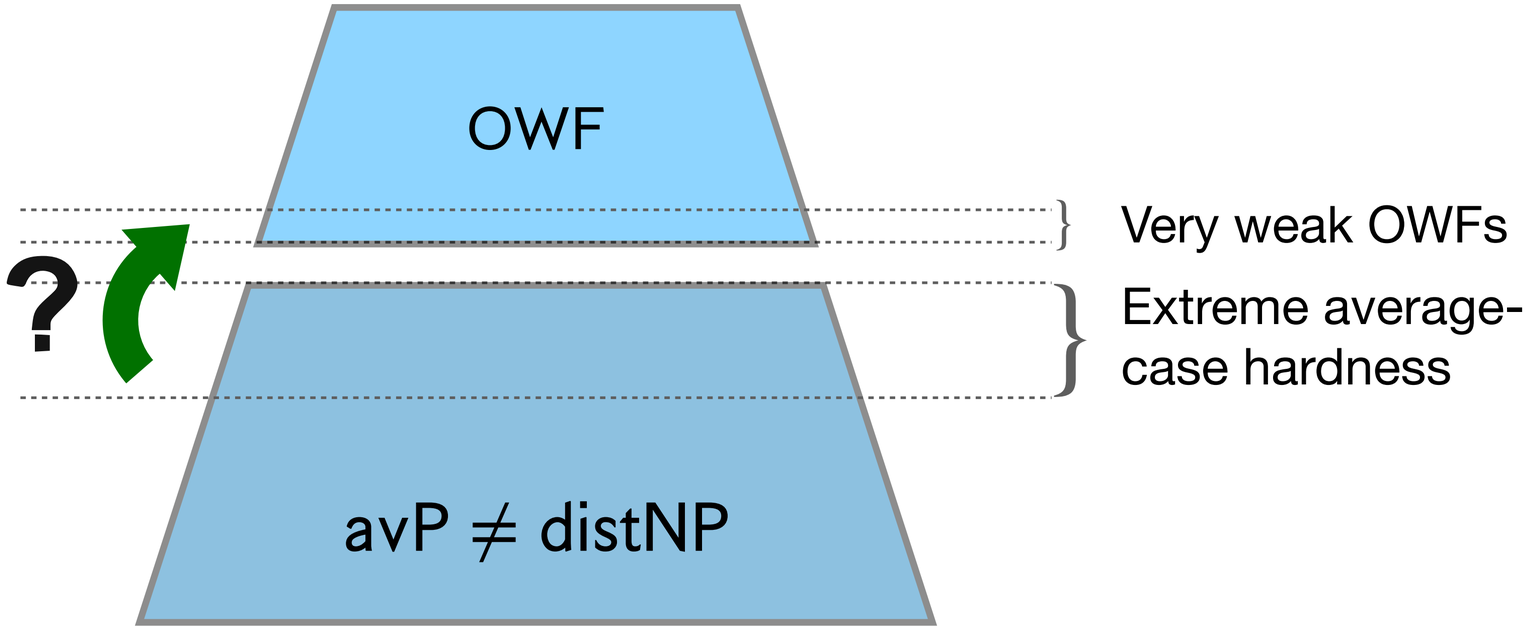


: black-box impossibility result

Can we Rule out Extreme-Pessiland?



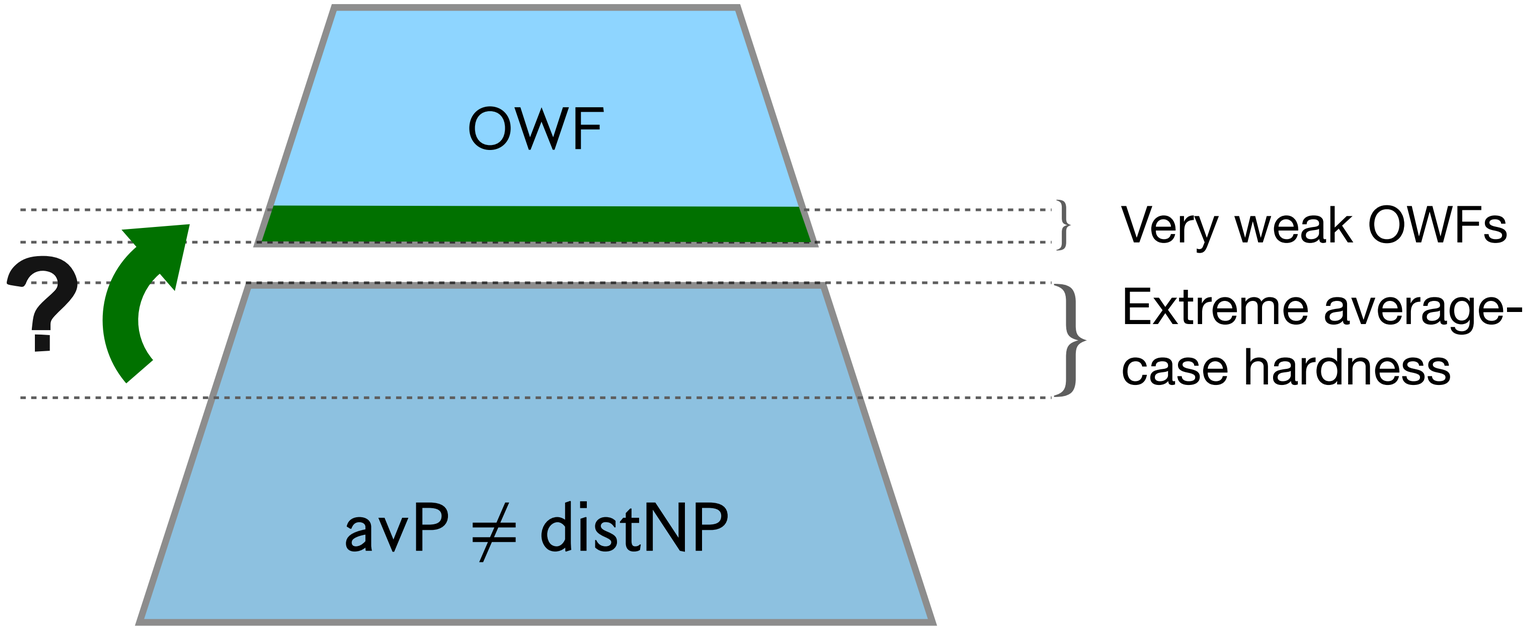
Could *extreme* average-case hardness imply *very weak* one-way functions?



Can we Rule out Extreme-Pessiland?

Could *extreme* average-case hardness imply *very weak* one-way functions?

***Fine-grained* one-way functions**



Can we Rule out Extreme-Pessiland?

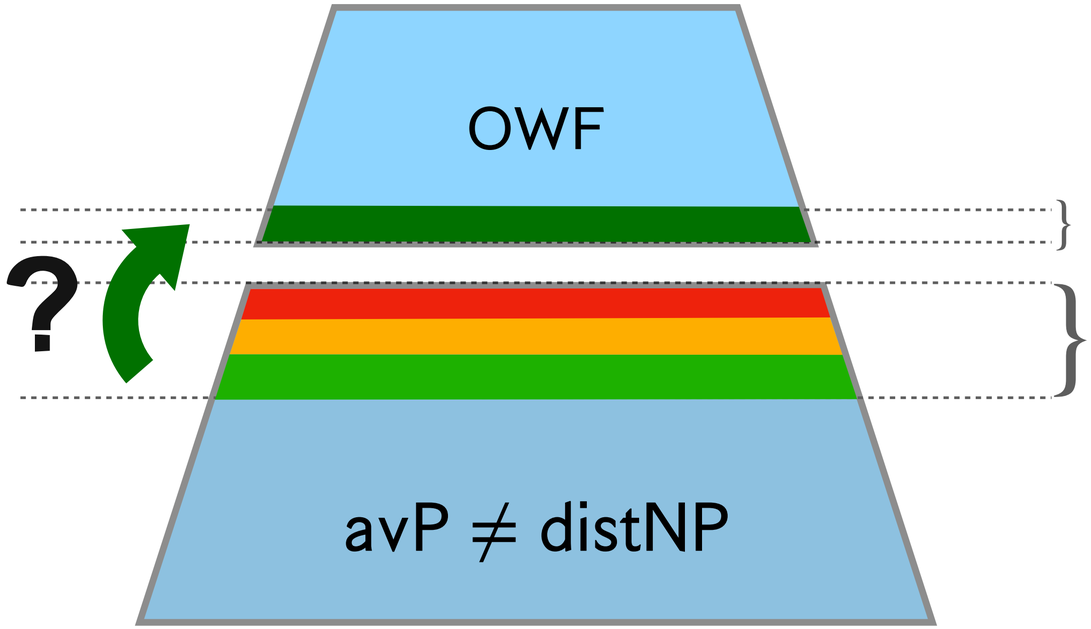
Could *extreme* average-case hardness imply *very weak* one-way functions?

***Fine-grained* one-way functions**

Block-finding average-case hardness

Amplifiable average-case hardness

Exponential average-case hardness



Very weak OWFs

Extreme average-case hardness

Can we Rule out Extreme-Pessiland?

Fine-grained one-way functions

1) Can be evaluated in time N
2) Inverting requires time $N^{1+\varepsilon}$ ($1+\varepsilon$ = *hardness gap,* typically $\varepsilon > 0$ is constant).

stronger

Block-finding average-case hardness

Amplifiable average-case hardness

Exponential average-case hardness

weaker

Can we Rule out Extreme-Pessiland?

Fine-grained one-way functions

1) Can be evaluated in time N
2) Inverting requires time $N^{1+\varepsilon}$ ($1+\varepsilon$ = *hardness gap,* typically $\varepsilon > 0$ is constant).

Exponential average-case hardness

$\exists \mathscr{L} \in$ NP, $\forall$ T-time $\mathscr{A}$ and large enough n,

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathscr{A}(x) = \mathscr{L}(x)] \leq \frac{1}{2} + \frac{T}{2^n}$$

Predicate, $\mathscr{L}(x) = 1$ *iff* $x \in \mathscr{L}$

Win-win result? Either $\exists$ non-trivial crypto hardness, or $\exists$ sub-2^n av-time algo for all NP languages

Can we Rule out Extreme-Pessiland?

Fine-grained one-way functions

1) Can be evaluated in time N
2) Inverting requires time $N^{1+\varepsilon}$ ($1+\varepsilon$ = *hardness gap,* typically $\varepsilon > 0$ is constant).

Exponential average-case hardness

$\exists \mathscr{L} \in$ NP, $\forall$ T-time $\mathscr{A}$ and large enough n,

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathscr{A}(x) = \mathscr{L}(x)] \leq \frac{1}{2} + \frac{T}{2^n}$$

▲

Predicate, $\mathscr{L}(x) = 1$ *iff* $x \in \mathscr{L}$

Win-win result? Either ∃ non-trivial crypto hardness, or ∃ sub-2^n av-time algo for all NP languages

Our first result: No black-box construction of FGOWF from exp hard av-case languages

▼ *Not too hard*

Can we Rule out Extreme-Pessiland?

Fine-grained one-way functions

1) Can be evaluated in time N
2) Inverting requires time $N^{1+\varepsilon}$ ($1+\varepsilon$ = *hardness gap,* typically $\varepsilon > 0$ is constant).

Amplifiable average-case hardness

$\exists \mathscr{L} \in$ NP, $\forall$ T-time $\mathscr{A}$, large enough n, $\ell(n)$:

$$\Pr_{\vec{x} \leftarrow \{0,1\}^{\ell \cdot n}} [\mathscr{A}(\vec{x}) = \mathscr{L}(\vec{x})] \leq 2^{\frac{T}{2^n} - \ell(n)}$$

(Simplified definition)

⚠ T can be much larger than 2^n here!

Win-win result? either $\exists$ non-trivial crypto hardness, or $\exists$ sub-2^n av-time algo for all NP languages *when amortising across many instances*.

Further motivation: in the past, *non-amortizability* helped circumventing impossibilities! Examples:

Can we Rule out Extreme-Pessiland?

Fine-grained one-way functions

1) Can be evaluated in time N
2) Inverting requires time $N^{1+\varepsilon}$ ($1+\varepsilon$ = *hardness gap,* typically $\varepsilon > 0$ is constant).

Amplifiable average-case hardness

$\exists \mathscr{L} \in$ NP, $\forall$ T-time $\mathscr{A}$, large enough n, $\ell(n)$:

$$\Pr_{\vec{x} \leftarrow \{0,1\}^{\ell \cdot n}} [\mathscr{A}(\vec{x}) = \mathscr{L}(\vec{x})] \leq 2^{\frac{T}{2^n} - \ell(n)}$$

(Simplified definition)

⚠ T can be much larger than 2^n here!

Win-win result? either $\exists$ non-trivial crypto hardness, or $\exists$ sub-2^n av-time algo for all NP languages *when amortising across many instances*.

Further motivation: in the past, *non-amortizability* helped circumventing impossibilities! Examples:

1. Under *dream XOR lemma*, exp. OWF imply n^2-PKE with negligible security error [BGI08] (BB-impossible w/out it)
   Dream XOR lemma = XORing hard predicates amplifies hardness optimally

Can we Rule out Extreme-Pessiland?

Fine-grained one-way functions

1) Can be evaluated in time N
2) Inverting requires time $N^{1+\varepsilon}$ ($1+\varepsilon$ = *hardness gap,* typically $\varepsilon > 0$ is constant).

Amplifiable average-case hardness

$\exists \mathscr{L} \in$ NP, $\forall$ T-time $\mathscr{A}$, large enough n, $\ell(n)$:

$$\Pr_{\vec{x} \leftarrow \{0,1\}^{\ell \cdot n}} [\mathscr{A}(\vec{x}) = \mathscr{L}(\vec{x})] \leq 2^{\frac{T}{2^n} - \ell(n)}$$

(Simplified definition)

⚠ T can be much larger than 2^n here!

Win-win result? either $\exists$ non-trivial crypto hardness, or $\exists$ sub-2^n av-time algo for all NP languages *when amortising across many instances*.

Further motivation: in the past, *non-amortizability* helped circumventing impossibilities! Examples:

(2) Exp-hard OWFs with *amplifiable hardness* imply CRHFs [HL18]. Without it, it is BB-impossible [Simon98]

Can we Rule out Extreme-Pessiland?

Fine-grained one-way functions

1) Can be evaluated in time N
2) Inverting requires time $N^{1+\varepsilon}$ ($1+\varepsilon$ = *hardness gap,* typically $\varepsilon > 0$ is constant).

Amplifiable average-case hardness

$\exists \mathscr{L} \in$ NP, $\forall$ T-time $\mathscr{A}$, large enough n, $\ell(n)$:

$$\Pr_{\vec{x} \leftarrow \{0,1\}^{\ell \cdot n}} [\mathscr{A}(\vec{x}) = \mathscr{L}(\vec{x})] \leq 2^{\frac{T}{2^n} - \ell(n)}$$

(Simplified definition)

⚠ T can be much larger than 2^n here!

Win-win result? either $\exists$ non-trivial crypto hardness, or $\exists$ sub-2^n av-time algo for all NP languages *when amortising across many instances*.

Our second result: No BB construction of FGOWF even from amplifiable av-case hard languages

Very technical

Can we Rule out Extreme-Pessiland?

Fine-grained one-way functions

1) Can be evaluated in time N
2) Inverting requires time $N^{1+\varepsilon}$ ($1+\varepsilon$ = *hardness gap,* typically $\varepsilon > 0$ is constant).

Block-finding average-case hardness

Sample $x_1 \in \mathscr{L}, x_2 \notin \mathscr{L}, x_3 \in \mathscr{L}, \cdots, x_m \notin \mathscr{L}$

Question: given a *pattern* $p \in \{0,1\}^t$, are there t consecutive words $x_{i+1}, \cdots, x_{i+t}$ such that $\mathscr{L}(x_{i+1}, \cdots, x_{i+t}) = s$?

Assumption: finding the answer requires $\Omega(m \cdot 2^n)$ time

😐 **Win-win (?) result?** Either ∃ non-trivial crypto hardness, or, for any NP language $\mathscr{L}$, given many instances, we can decide something about their *membership pattern* in $\mathscr{L}$ faster than brute-force.

Can we Rule out Extreme-Pessiland?

Fine-grained one-way functions

1) Can be evaluated in time N
2) Inverting requires time $N^{1+\varepsilon}$ ($1+\varepsilon$ = *hardness gap,* typically $\varepsilon > 0$ is constant).

Block-finding average-case hardness

Sample $x_1 \in \mathscr{L}, x_2 \notin \mathscr{L}, x_3 \in \mathscr{L}, \cdots, x_m \notin \mathscr{L}$

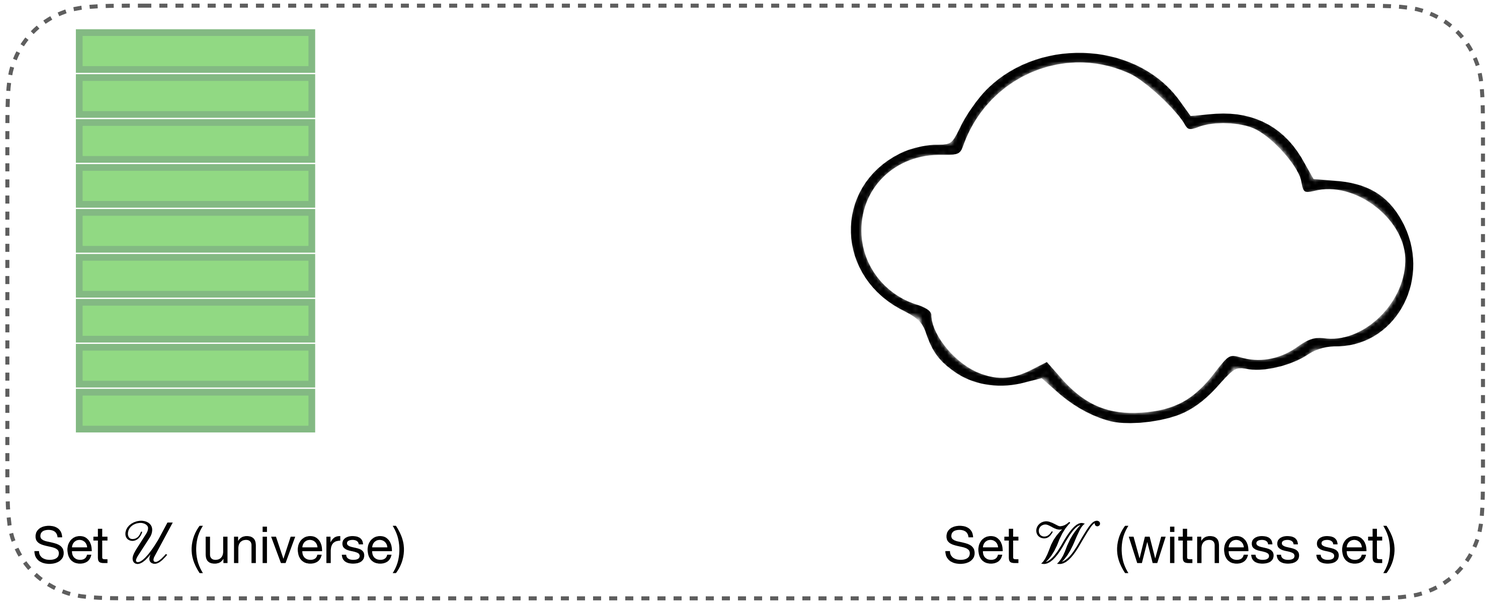
Question: given a *pattern* $p \in \{0,1\}^t$, are there t consecutive words $x_{i+1}, \cdots, x_{i+t}$ such that $\mathscr{L}(x_{i+1}, \cdots, x_{i+t}) = s$?

Assumption: finding the answer requires $\Omega(m \cdot 2^n)$ time

😐 **Win-win (?) result?** Either ∃ non-trivial crypto hardness, or, for any NP language $\mathscr{L}$, given many instances, we can decide something about their *membership pattern* in $\mathscr{L}$ faster than brute-force.

🙂 **Our third result:** if ∃ block-finding hard languages, there are n^2-hard FGOWFs

Useful Tool: Random Languages

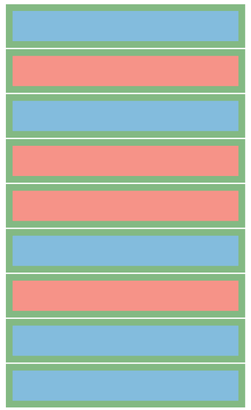


Useful Tool: Random Languages

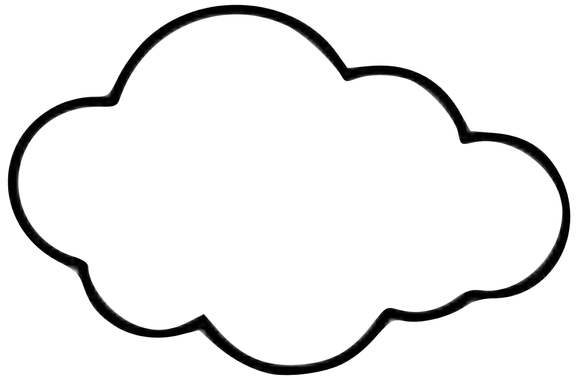
$\mathscr{L}$

$\mathscr{U} \setminus \mathscr{L}$

Flip coins, put $\sim$ half of $\mathscr{U}$ in $\mathscr{L}$



Set $\mathscr{U}$ (universe)



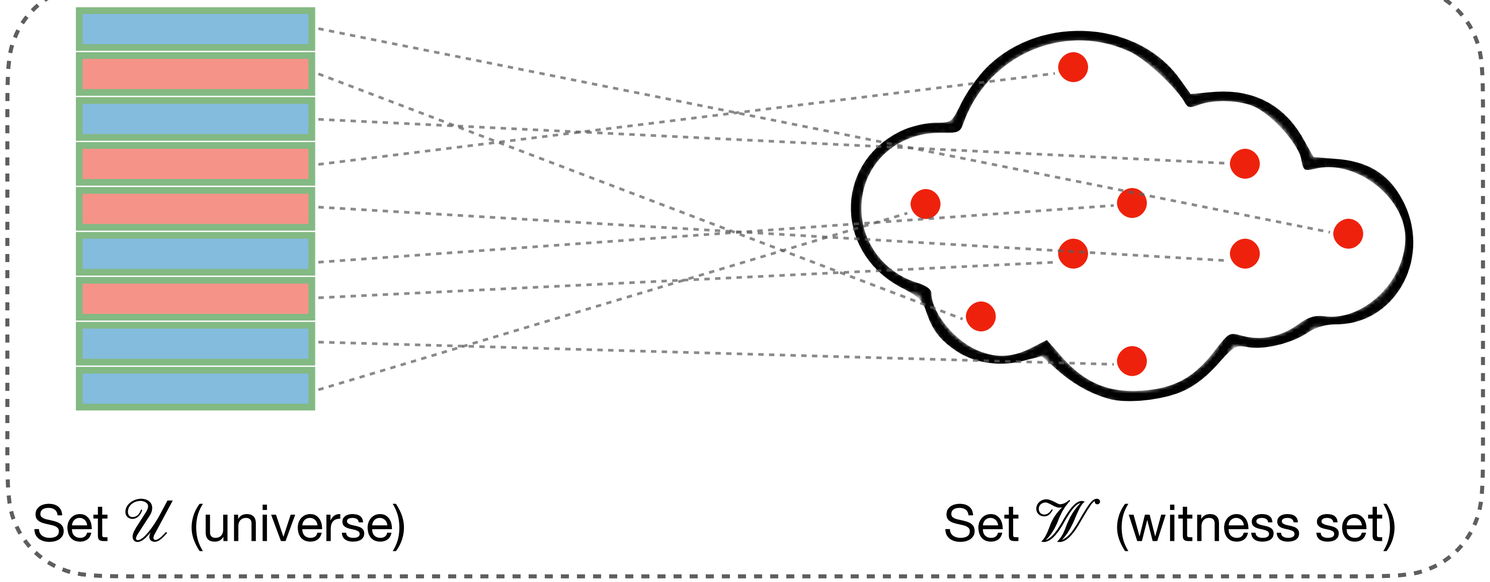
Set $\mathscr{W}$ (witness set)

Useful Tool: Random Languages

$\mathscr{L}$

$\mathscr{U} \setminus \mathscr{L}$

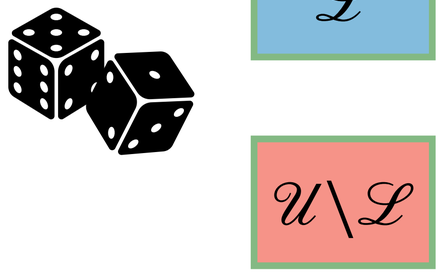
Flip coins, put $\sim$ half of $\mathscr{U}$ in $\mathscr{L}$



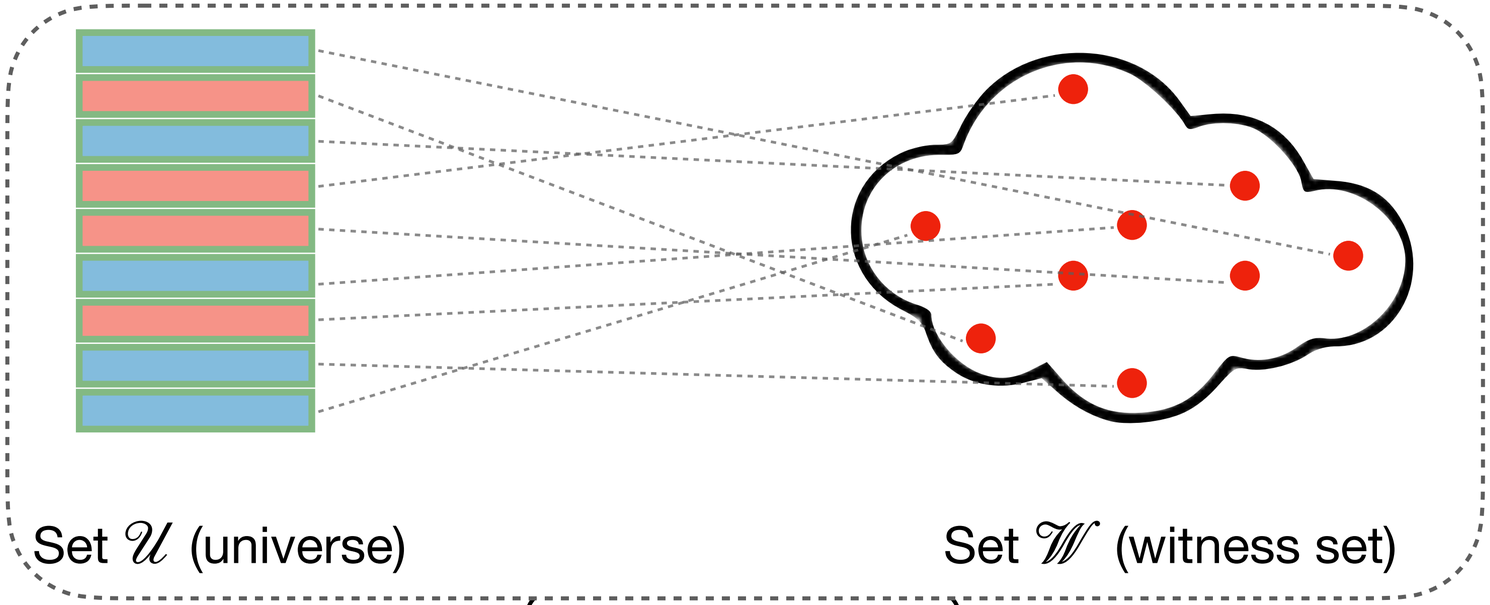
Associate to each element of $\mathscr{U}$ a uniformly random witness in $\mathscr{W}$

Notation: the witness of $x \in \mathscr{U}$ is denoted w_x, and b_x is set to 1 iff $x \in \mathscr{L}$

Useful Tool: Random Languages



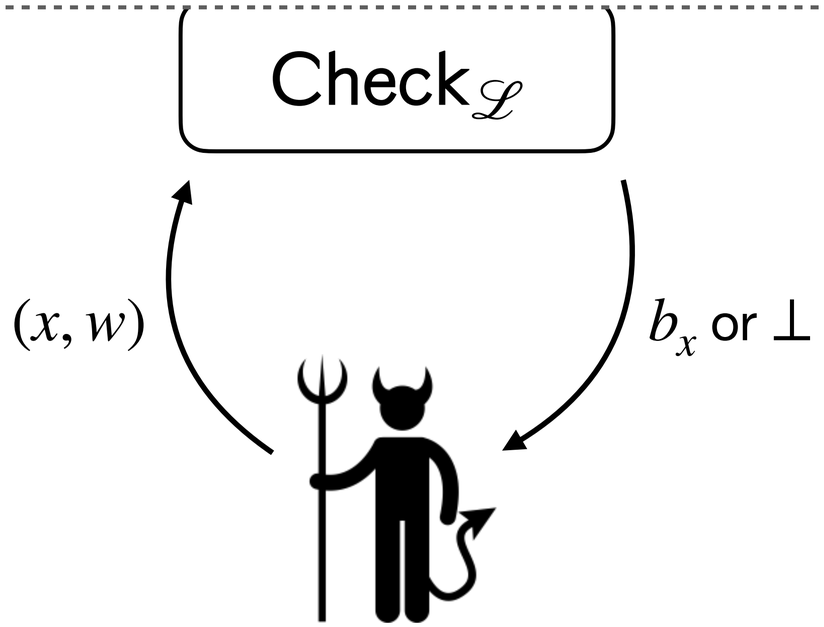
Flip coins, put ~ half of $\mathscr{U}$ in $\mathscr{L}$



Associate to each element of $\mathscr{U}$ a uniformly random witness in $\mathscr{W}$

Notation: the witness of $x \in \mathscr{U}$ is denoted w_x, and b_x is set to 1 iff $x \in \mathscr{L}$

Add a Check oracle: on input (x, w), check if $w = w_x$. If yes, return b_x, else, return $\perp$.



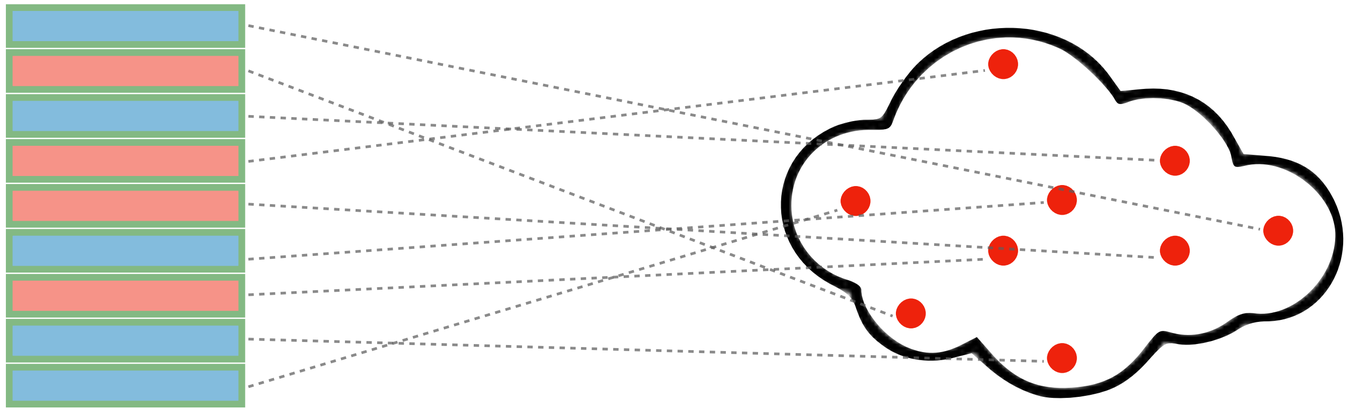
= *random language in* NP ∩ coNP

Useful Tool: Random Languages

$\mathscr{L}$

$\mathscr{U} \setminus \mathscr{L}$

Flip coins, put ~ half of $\mathscr{U}$ in $\mathscr{L}$



Set $\mathscr{U}$ (universe)

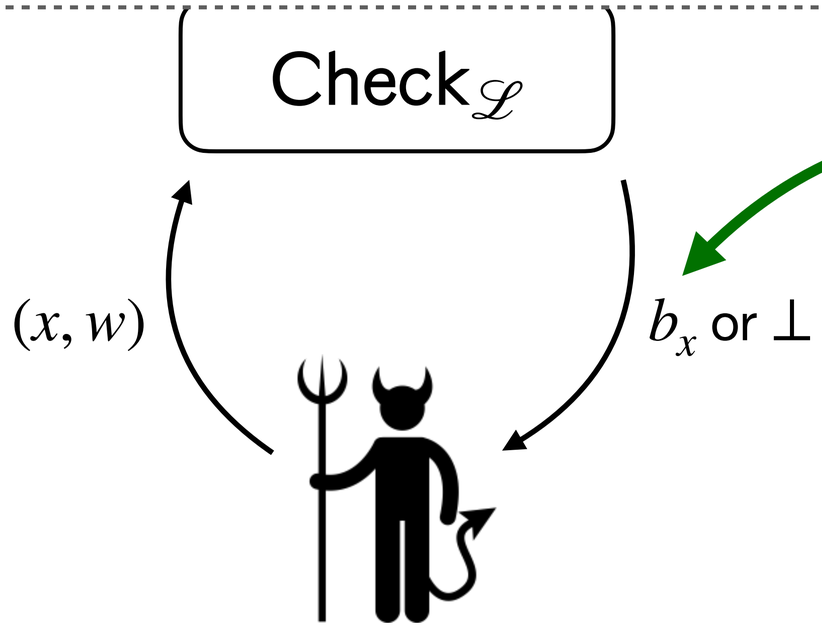
Set $\mathscr{W}$ (witness set)

Associate to each element of $\mathscr{U}$ a uniformly random witness in $\mathscr{W}$

Notation: the witness of $x \in \mathscr{U}$ is denoted w_x, and b_x is set to 1 iff $x \in \mathscr{L}$

Add a Check oracle: on input (x, w), check if $w = w_x$. If yes, return b_x, else, return $\perp$.

= *random language in* NP ∩ coNP



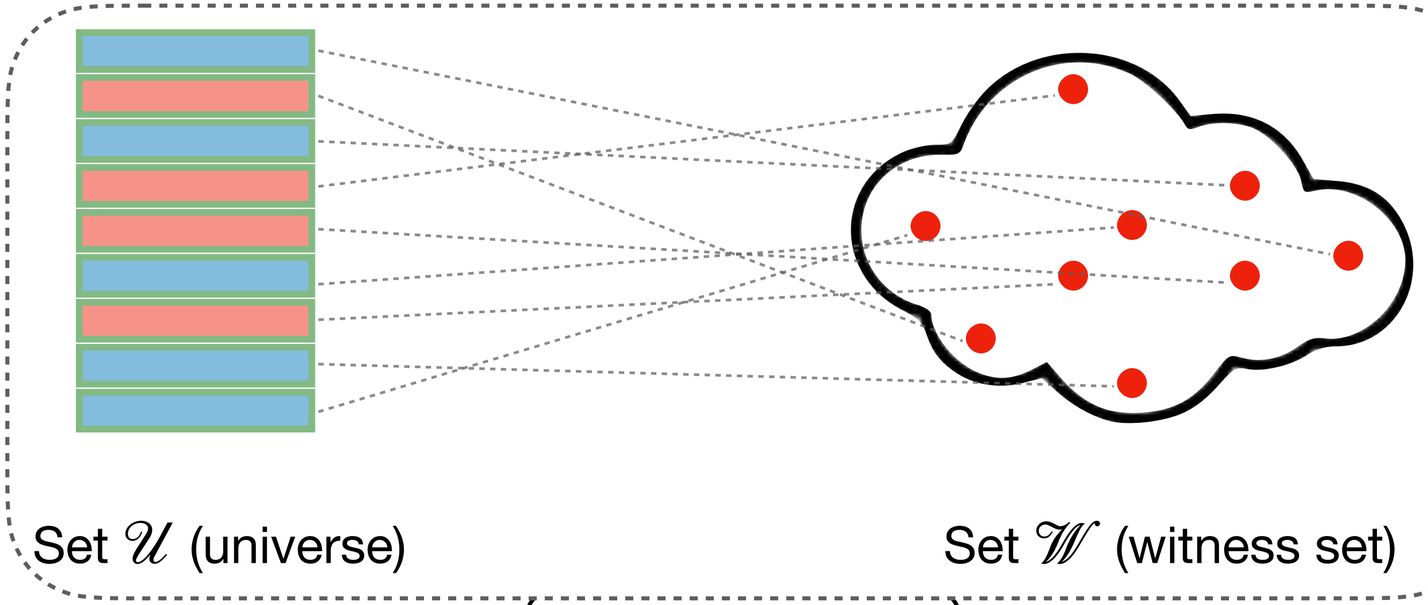
We call this a *hit*

Useful Tool: Random Languages

$\mathscr{L}$

$\mathscr{U} \setminus \mathscr{L}$

Flip coins, put ~ half of $\mathscr{U}$ in $\mathscr{L}$

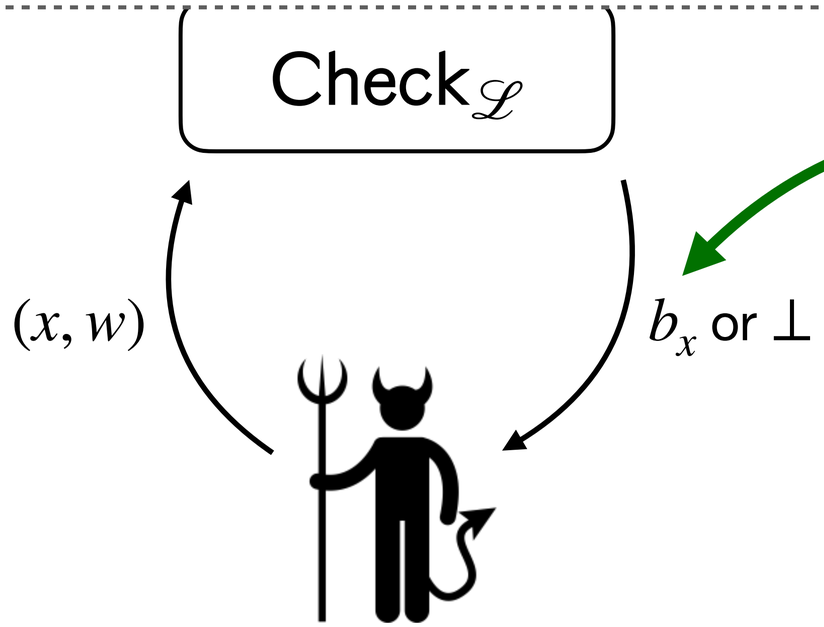


Associate to each element of $\mathscr{U}$ a uniformly random witness in $\mathscr{W}$

Notation: the witness of $x \in \mathscr{U}$ is denoted w_x, and b_x is set to 1 iff $x \in \mathscr{L}$

Add a Check oracle: on input (x, w), check if $w = w_x$. If yes, return b_x, else, return $\perp$.

= *random language in* NP ∩ coNP

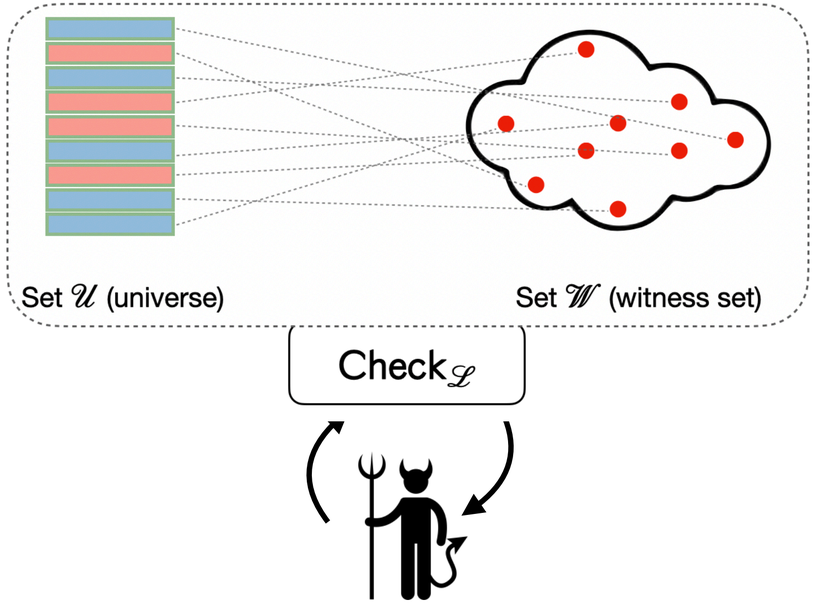


We call this a *hit*

Two goals

- Checking that block-finding hardness is plausible
- Basis for all black-box separations

Useful Tool: Random Languages



We prove three key results:

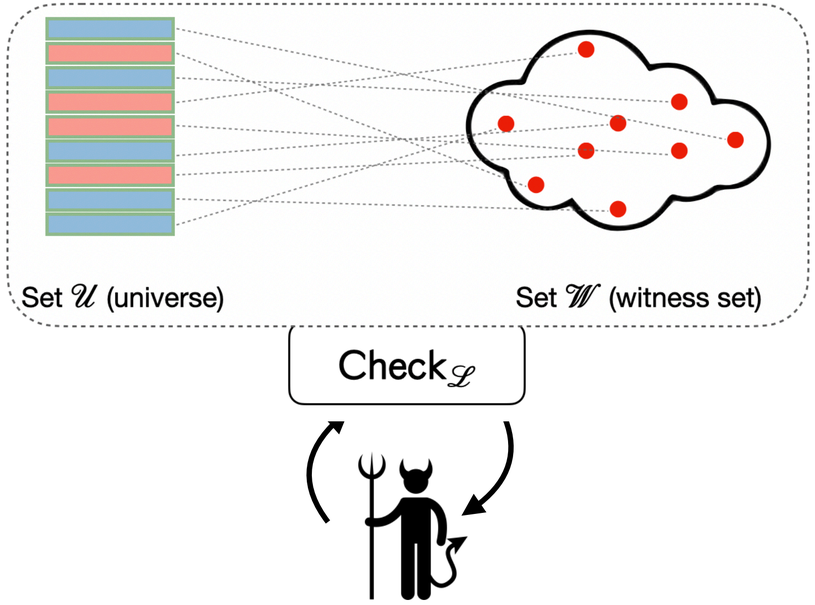
1. A random language satisfies *block-finding hardness*

2. Even given a (weakened) FG-OWF inverter Inv (that samples *sufficiently likely* preimages to any oracle circuit C^{Check}), a random language is still *amplifiable average-case hard*

3. Relative to Check, Inv, there exists no fine-grained OWF

Useful Tool: Random Languages

Provides support for the plausibility of the notion of block-finding hardness (similar to when we prove that an assumption about hash functions holds in the ROM)



We prove three key results:

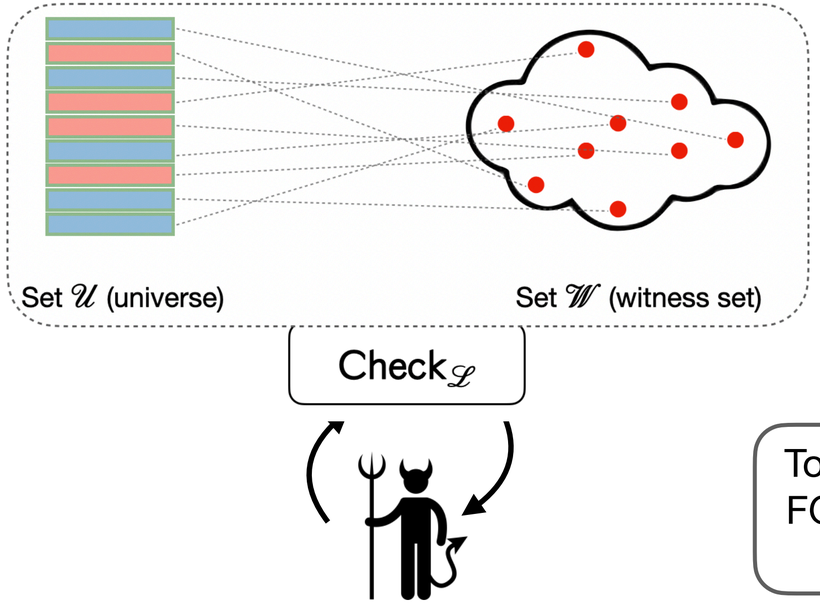
1. A random language satisfies *block-finding hardness*

2. Even given a (weakened) FG-OWF inverter Inv (that samples *sufficiently likely* preimages to any oracle circuit C^{Check}), a random language is still *amplifiable average-case hard*

3. Relative to Check, Inv, there exists no fine-grained OWF

Useful Tool: Random Languages

Provides support for the plausibility of the notion of block-finding hardness (similar to when we prove that an assumption about hash functions holds in the ROM)



Together, 2 and 3 prove our second result: FGOWF and amplifiable hardness are BB-separated

We prove three key results:

1. A random language satisfies *block-finding hardness*

2. Even given a (weakened) FG-OWF inverter Inv (that samples *sufficiently likely* preimages to any oracle circuit C^{Check}), a random language is still *amplifiable average-case hard*

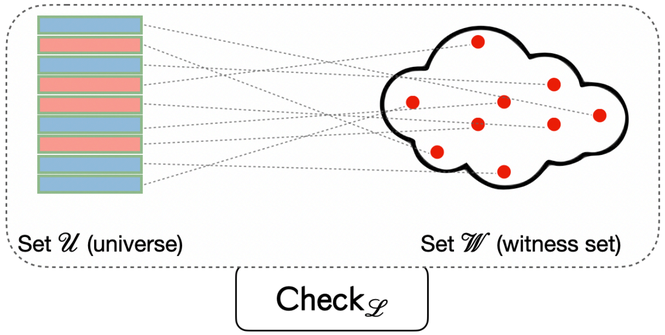
3. Relative to Check, Inv, there exists no fine-grained OWF

Separating Fine-Grained OWF and Amplifiable Hardness

Oracles:

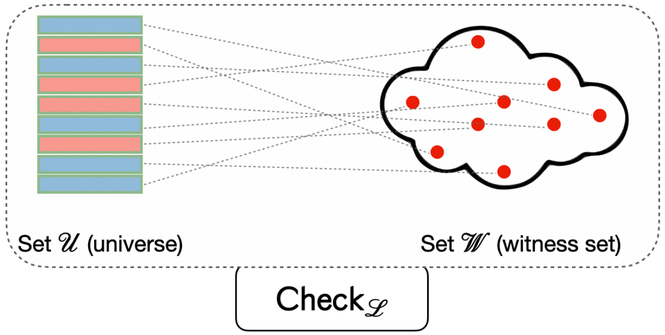
Separating Fine-Grained OWF and Amplifiable Hardness

Oracles:



Separating Fine-Grained OWF and Amplifiable Hardness

Oracles:

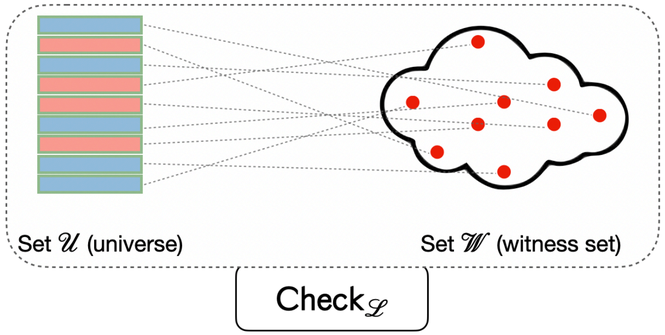


$\mathtt{Inv}\left(C^{\mathsf{Check}}, y\right)$:

- Removes heavy paths, i.e., path where the number of *hits* is higher than expected, and return a random light path
- Shaves the circuit (removing *useless on average* gates)
- Derandomizes the oracle with a universal hash function

Separating Fine-Grained OWF and Amplifiable Hardness

Oracles:



$\mathsf{Inv}\left(C^{\mathsf{Check}}, y\right)$:

- Removes heavy paths, i.e., path where the number of *hits* is higher than expected, and return a random light path
- Shaves the circuit (removing *useless on average* gates)
- Derandomizes the oracle with a universal hash function

Removing heavy paths is necessary to avoid solving $\mathscr{L}$ by inverting circuits on *lucky* outputs.
Example: $C^{\mathsf{Check}}(x, w)$ queries $b \leftarrow \mathsf{Check}(x, w)$ and returns (x, b)
Solver: on input x, queries $\mathsf{Inv}(C^{\mathsf{Check}}, (x,1))$

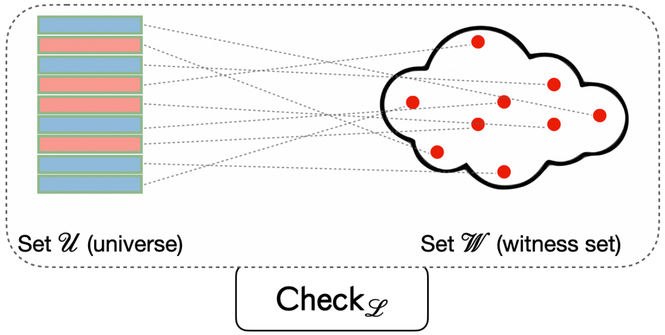
Later, we will use Inv to invert a OWF F. Hence, we need that even after removing *all* heavy paths for *all* Check gates, Inv still gives a valid preimage of $F(x)$. Due to the union bound over all gates, we need to set $\mathsf{Light}_k = \mathsf{Avg}_k + \log^2 |C|$.

Problem: for a large k, Avg_k is tiny, hence $\log^2 |C| \gg \mathsf{Avg}_k$: we are still allowing too many hits!

Solution: before answering, Inv shaves the circuit to remove all Check_k with large k, and replace them with a dummy $\perp$ -gate (this is fine since with high probability, these gates never hit!)

Separating Fine-Grained OWF and Amplifiable Hardness

Oracles:



$\mathrm{Inv}\left(C^{\mathrm{Check}}, y\right)$:

- Removes heavy paths, i.e., path where the number of *hits* is higher than expected, and return a random light path
- Shaves the circuit (removing *useless on average* gates)
- Derandomizes the oracle with a universal hash function

Proof intuition ($\mathscr{L}$ is amplifiable average-case hard relative to (Check, Inv)):

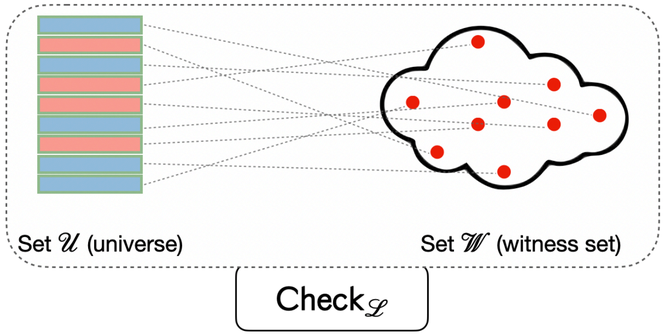
$\forall \mathscr{A}$, construct an *emulation procedure* $E_{\mathscr{A}}$ with access to Check and some carefully crafted *advice* about $\mathscr{L}$. Show that (1) this advice allows $E_{\mathscr{A}}$ to correctly emulate all answers of Inv, and (2) the size of the advice is *not too large.*

If $\mathscr{A}$ can break amplifiable hardness (= make too many *hits*), then $\mathscr{A}' = (\mathscr{A}, E_{\mathscr{A}})$ can make too many hits given *only* oracle access to Check and a not-too-long advice string.

We prove that the latter is impossible using a new technical lemma: the **Hitting Lemma with advice**

Separating Fine-Grained OWF and Amplifiable Hardness

Oracles:



$\mathsf{Inv}\left(C^{\mathsf{Check}}, y\right)$:

- Removes heavy paths, i.e., path where the number of *hits* is higher than expected, and return a random light path
- Shaves the circuit (removing *useless on average* gates)
- Derandomizes the oracle with a universal hash function

Proof intuition (there are no FG-OWF relative to $(\mathsf{Check}, \mathsf{Inv})$):

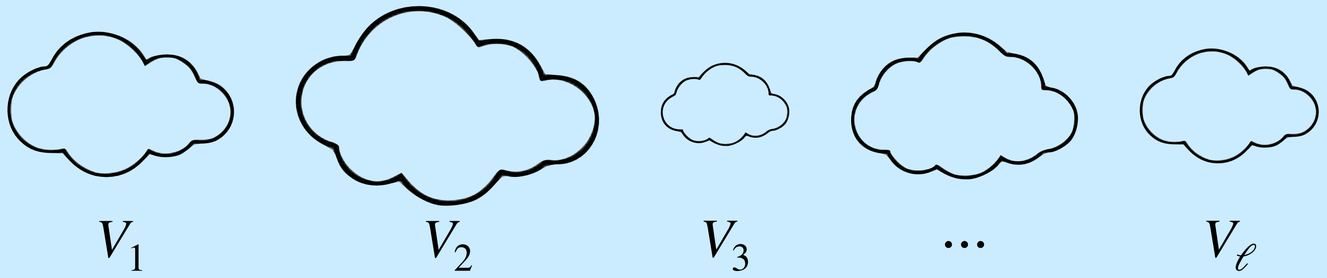
Given a candidate FGOWF F and $y = F(x)$, feed (F, y) to Inv and get a random preimage x. Now, we must prove that with high proba, Inv returns an inverse.

This boils down to showing (1) F and F_{shaved} agree on a random input with high proba, and (2) the path from x to y is a *light path* (remember: $\mathsf{Light}_k = \mathsf{Avg}_k + \log^2|C|$). Then:

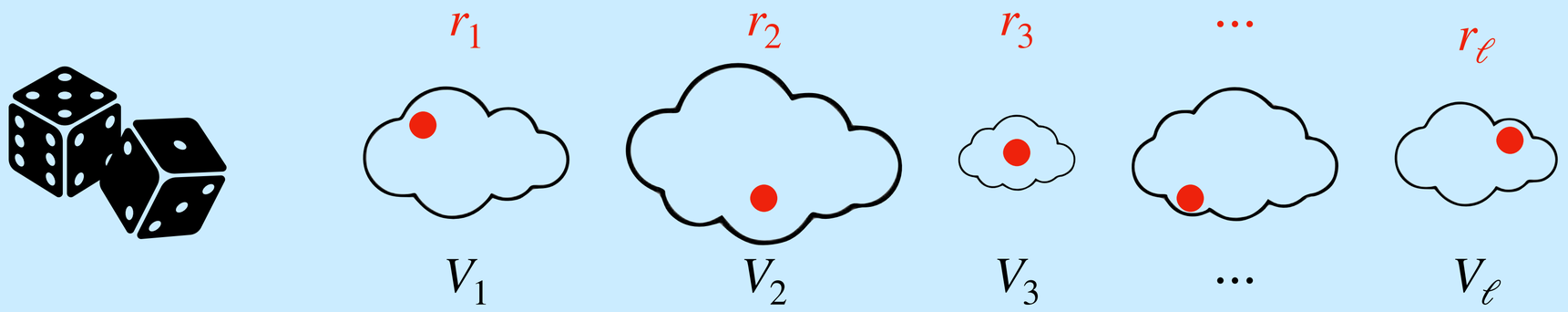
(1) Not too hard, follows from the definition of shaving.
(2) Need to show that the path from x to y cannot contain too many hits: follows again from the **Hitting Lemma**

Main Technical Tool: the Hitting Lemma

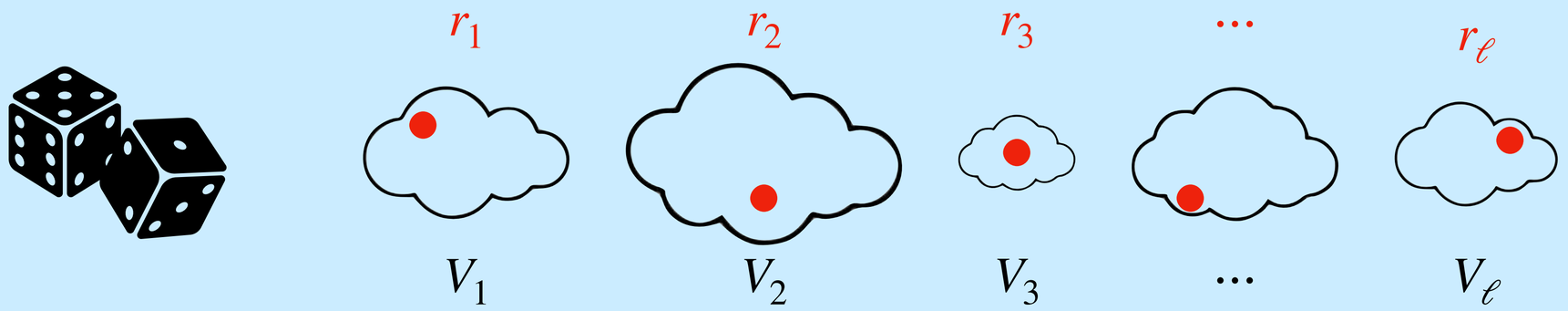
Main Technical Tool: the Hitting Lemma

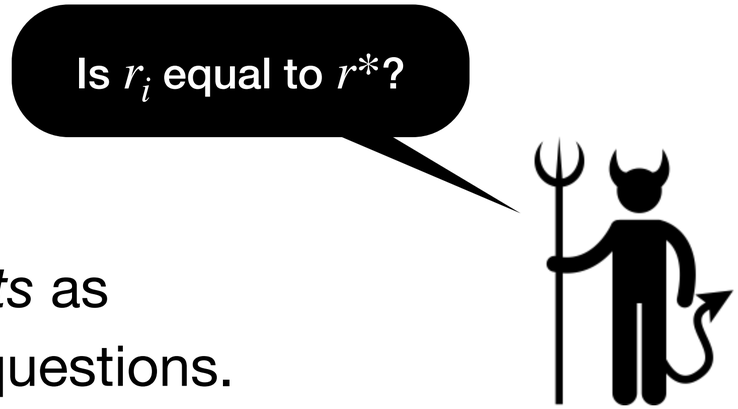


Main Technical Tool: the Hitting Lemma



Main Technical Tool: the Hitting Lemma



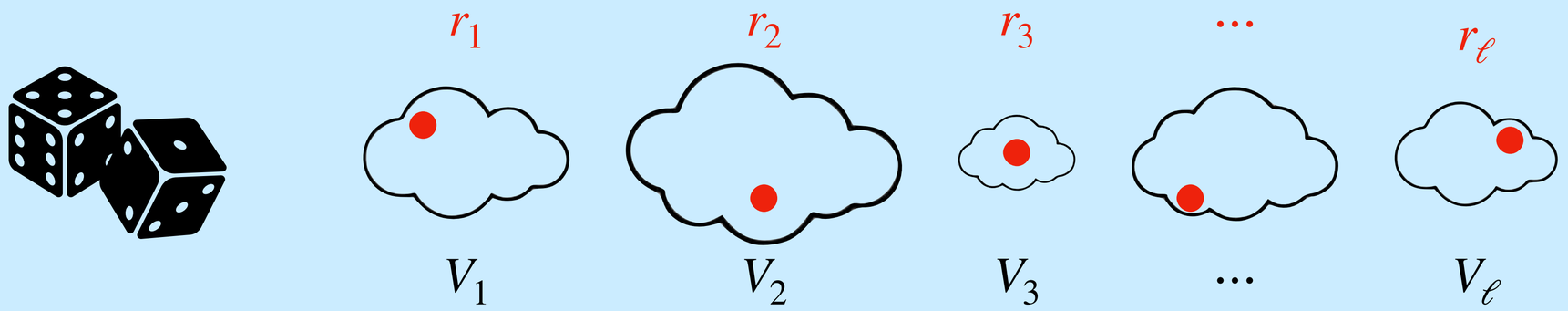


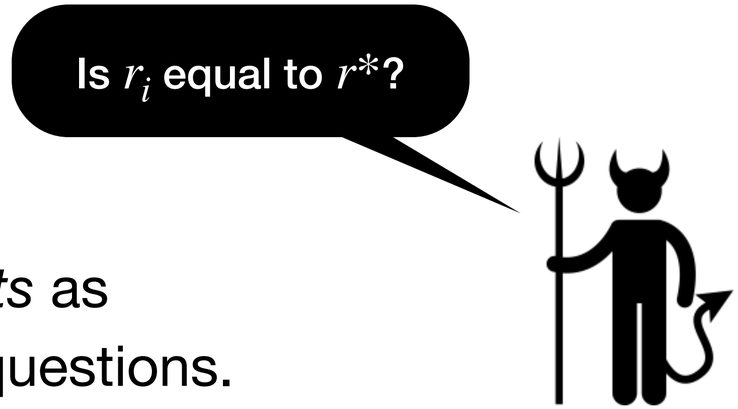
Goal: getting as many *hits* as possible after asking Q questions.

Yes: that's a *hit*

No: that's a fail

Main Technical Tool: the Hitting Lemma






Yes: that's a *hit*

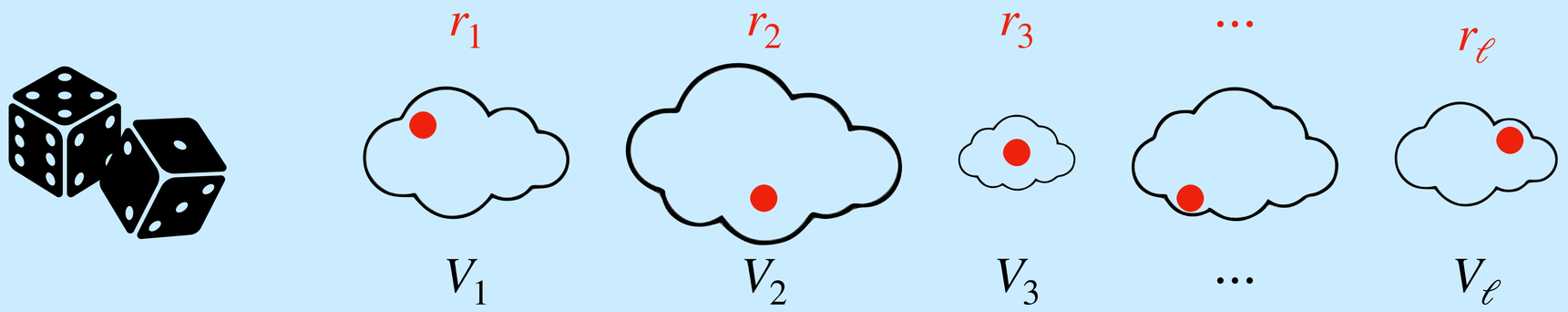
No: that's a fail

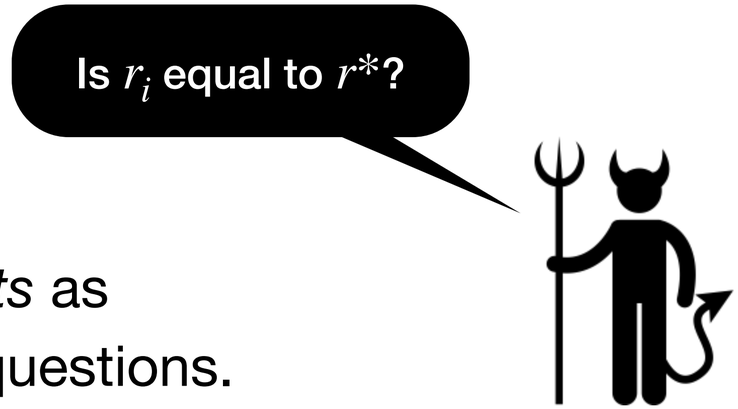
Goal: getting as many *hits* as possible after asking Q questions.

Hitting Lemma: for all sets $V_1 \cdots V_\ell$ of size at most 2^n, all Q-query adversary, any integer $c \geq 1$, there exists $\alpha > 0, \gamma > 1$ such that $\Pr_{r \leftarrow V_1 \times \cdots \times V_\ell}\left[\text{NumHits}(Q) \geq \frac{16Q+q}{2^n} + c\right] \leq \frac{\alpha}{2^{\gamma c}}$,

Where $q = \ell \cdot 2^n - \sum |V_i|$.

Main Technical Tool: the Hitting Lemma





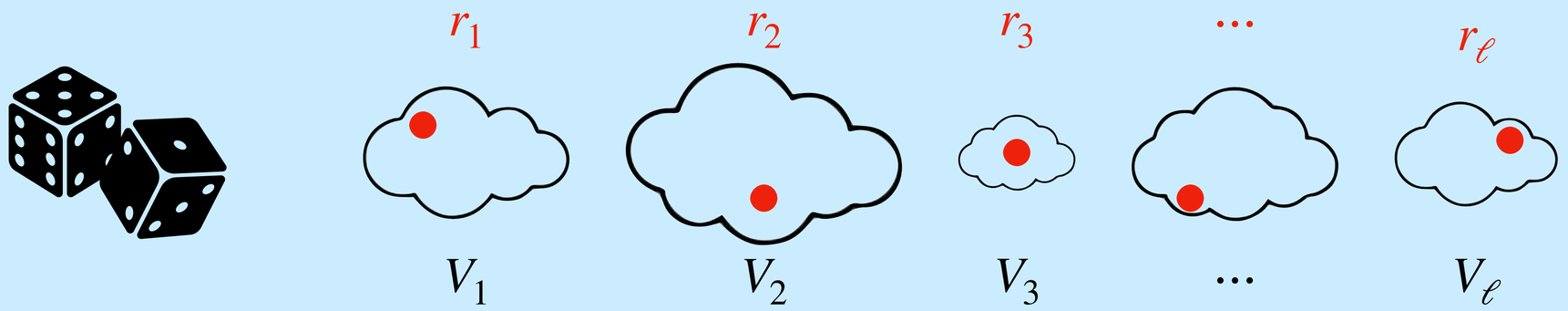
Yes: that's a *hit*

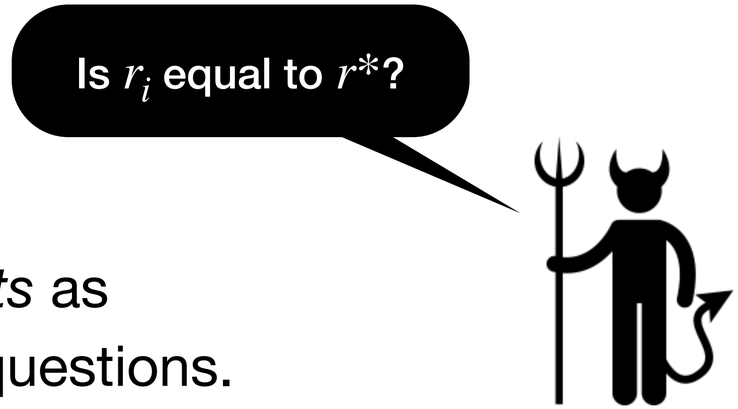
No: that's a fail

Goal: getting as many *hits* as possible after asking Q questions.

Hitting Lemma (informally): best strategy = querying 1-by-1 all elements in the smallest set V_i until we hit r_i, then querying the next smallest set, and so on. Furthermore, this makes h hits on average, where h = largest integer such that $\sum_{i=1}^{h} |V_i| \leq Q$ ($|V_i|$ in increasing order), and $\Pr[h + c \text{ hits}] = 2^{-\gamma \cdot c}$, with $\gamma > 1$.

Main Technical Tool: the Hitting Lemma





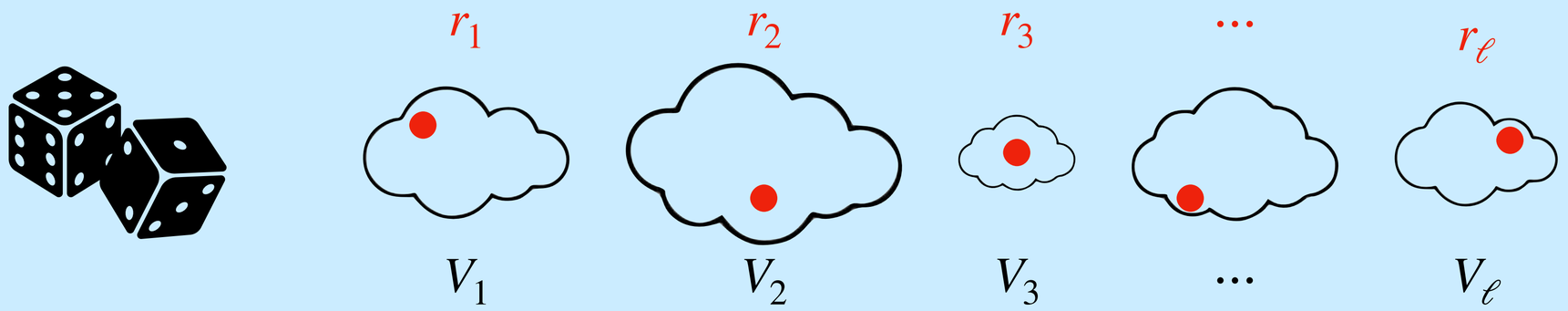

Yes: that's a *hit*

No: that's a fail

Goal: getting as many *hits* as possible after asking Q questions.

Hitting Lemma (sketch): Induction over the query number Q to prove that naive strategy = best strategy, then Bernstein's concentration bound (+ tedious calculations) to bound the naive strategy

The Hitting Lemma with Advice



Is r_i equal to r^*?

k bits of arbitrary advice

Yes: that's a *hit*

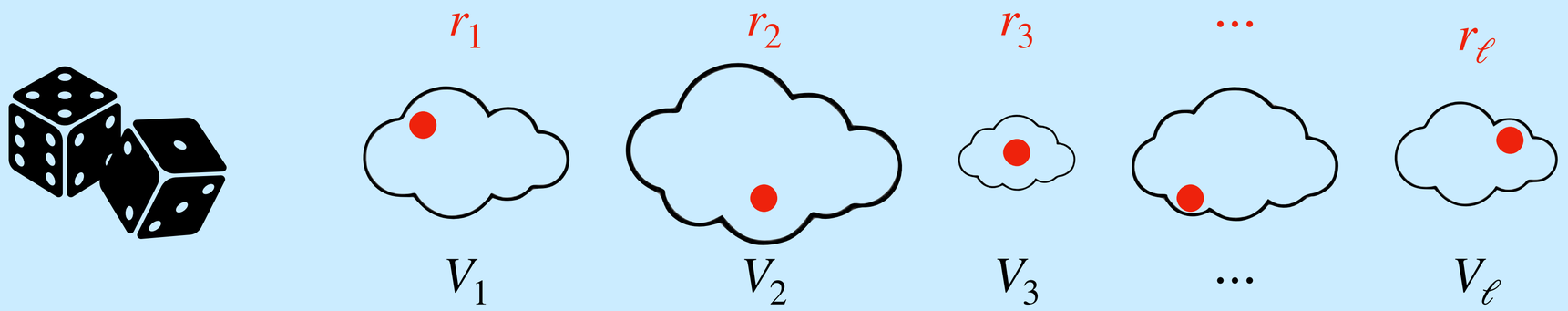
No: that's a fail

Goal: getting as many *hits* as possible after asking Q questions.

Hitting Lemma with advice: for all sets $V_1 \cdots V_\ell$ of size at most 2^n, all Q-query adversary, size-k advice, any integer $c \geq 1$, there exists $\alpha > 0, \gamma > 1$ such that $\Pr_{r \leftarrow V_1 \times \cdots \times V_\ell}\left[\text{NumHits}(Q) \geq \frac{16Q+q}{2^n} + c\right] \leq \frac{\alpha \cdot 2^k}{2^{\gamma c}}$,

Where $q = \ell \cdot 2^n - \sum |V_i|$.

The Hitting Lemma with Advice



Is r_i equal to r^*?

k bits of arbitrary advice

Yes: that's a *hit*

No: that's a fail

Goal: getting as many *hits* as possible after asking Q questions.

Hitting Lemma with advice: for all sets $V_1 \cdots V_\ell$ of size at most 2^n, all Q-query adversary, size-k advice, any integer $c \geq 1$, there exists $\alpha > 0, \gamma > 1$ such that $\Pr_{r \leftarrow V_1 \times \cdots \times V_\ell}\left[\text{NumHits}(Q) \geq \frac{16Q+q}{2^n} + c\right] \leq \frac{\alpha \cdot 2^k}{2^{\gamma c}}$,

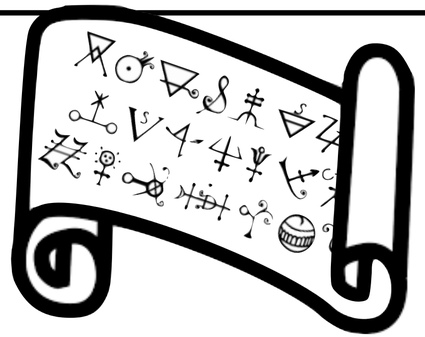
Where $q = \ell \cdot 2^n - \sum |V_i|$.

Sketch: immediate since $\gamma > 1$, just guess k!

Fine-Grained One-Way Functions from (Very) Hard Languages

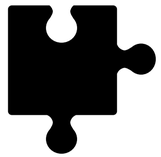
Fine-Grained One-Way Functions from (Very) Hard Languages

Block-finding hard language



Fine-Grained One-Way Functions from (Very) Hard Languages

Puzzle

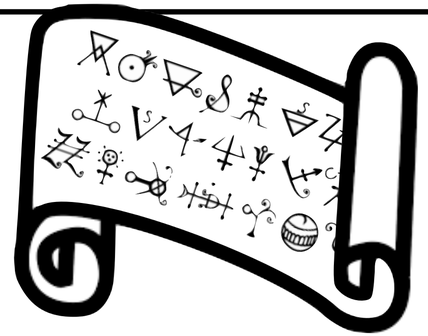
PuzzleGen: 

 : $O(1)$

PuzzleSolve() $\mapsto s \in \mathbb{N}$

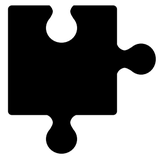
 : n

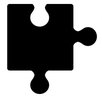
Block-finding hard language



Fine-Grained One-Way Functions from (Very) Hard Languages

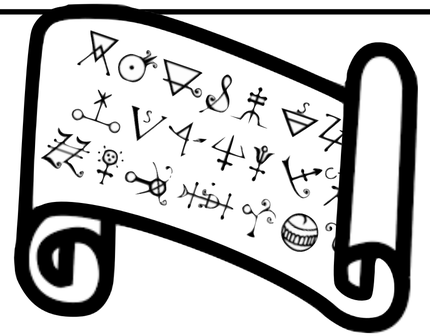
Puzzle

PuzzleGen:  : $O(1)$

PuzzleSolve() $\mapsto s \in \mathbb{N}$: n

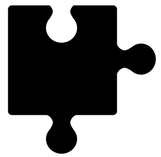
Block-finding hard language

Sample $x_1, \cdots, x_{\log|s|}$,
$s_i = 1$ iff $x_i \in \mathscr{L}$

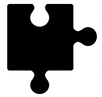


Fine-Grained One-Way Functions from (Very) Hard Languages

Puzzle

PuzzleGen: 

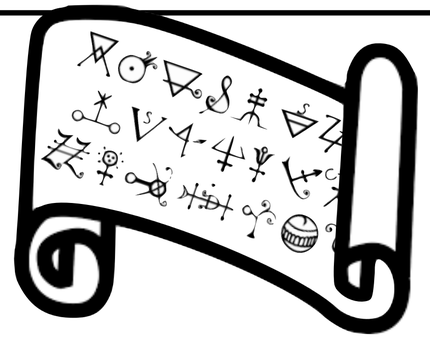
 : $O(1)$

PuzzleSolve() $\mapsto s \in \mathbb{N}$

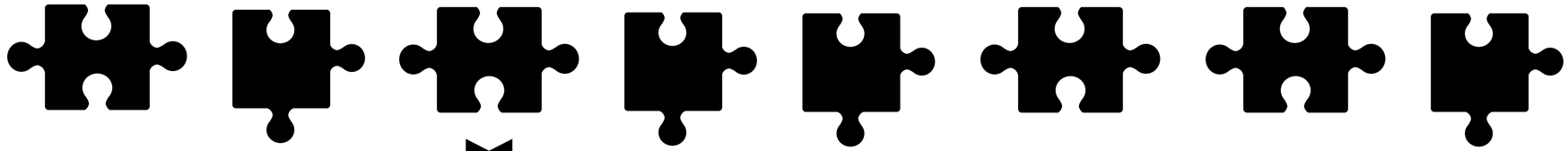
 : n

Block-finding hard language

Sample $x_1, \cdots, x_{\log|s|}$,
$s_i = 1$ iff $x_i \in \mathscr{L}$

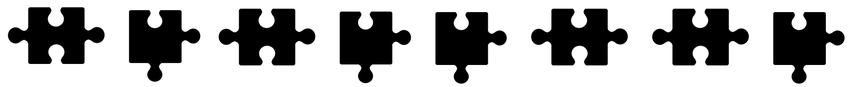


Generate n puzzles:

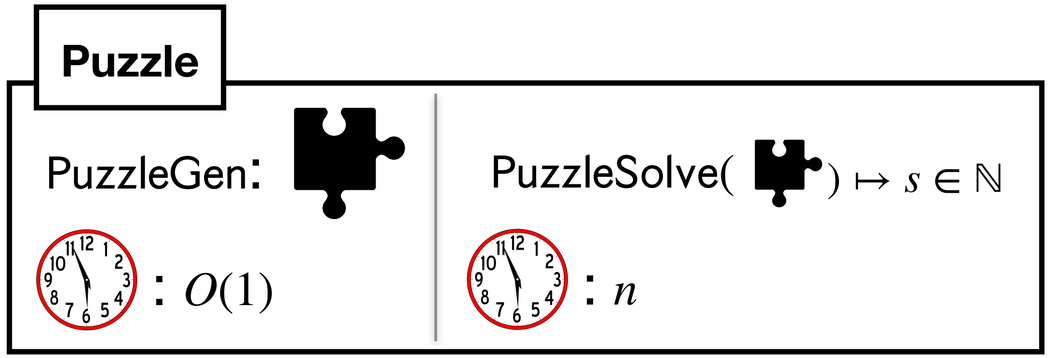


Pick a random puzzle i, solve it

s

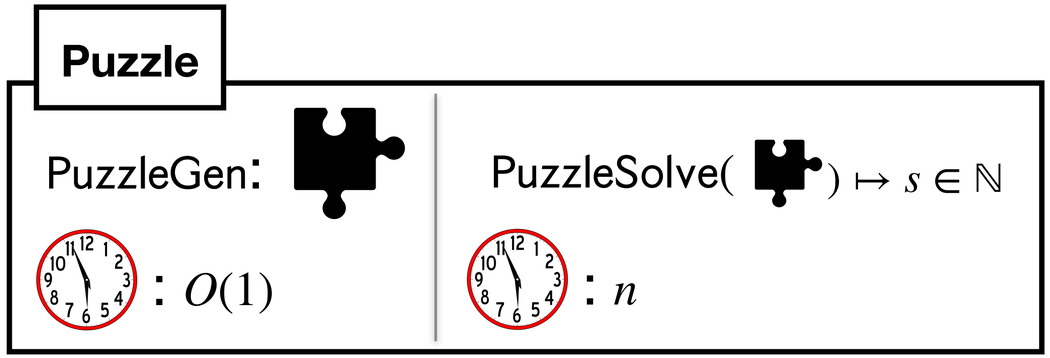
Fine-grained OWF: $F(\text{PuzzleRandomess}, \text{i}) \mapsto$ ( , s)

Fine-Grained One-Way Functions from (Very) Hard Languages



Fine-grained OWF: $F(\text{PuzzleRandomess}, \text{i}) \mapsto ($, $s)$

Fine-Grained One-Way Functions from (Very) Hard Languages



Fine-grained OWF: $F(\text{PuzzleRandomess}, \text{i}) \mapsto ($, $s)$

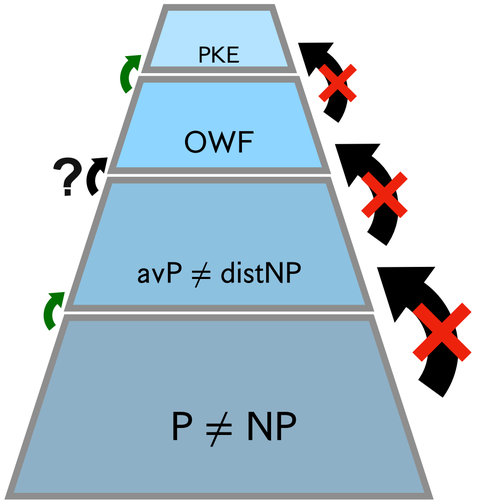
Evaluating F: $O(n)$ ($n \times$ PuzzleGen) + n ($1 \times$ PuzzleSolve)

Inverting F: $O(n^2)$ ($n \times$ PuzzleSolve) if the language is *block-finding hard*

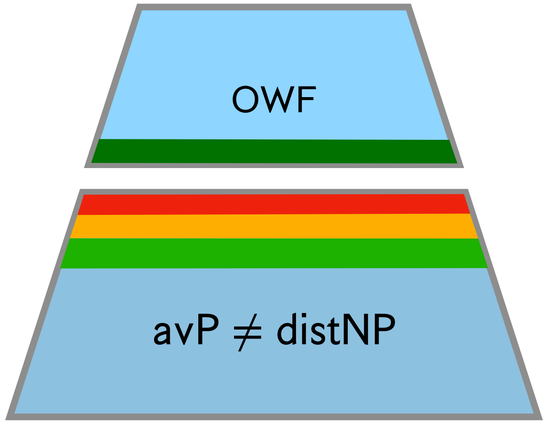
Note : A random language is provably *block-finding hard* by the Hitting Lemma with advice. The reduction

- Obtains the pattern as *advice* about $\mathscr{L}$, and
- Uses lazy sampling to guarantee that membership of a word is undefined before a hit happens
- Then, finding the pattern position implies making too many hits given a small advice, contradicting the HL

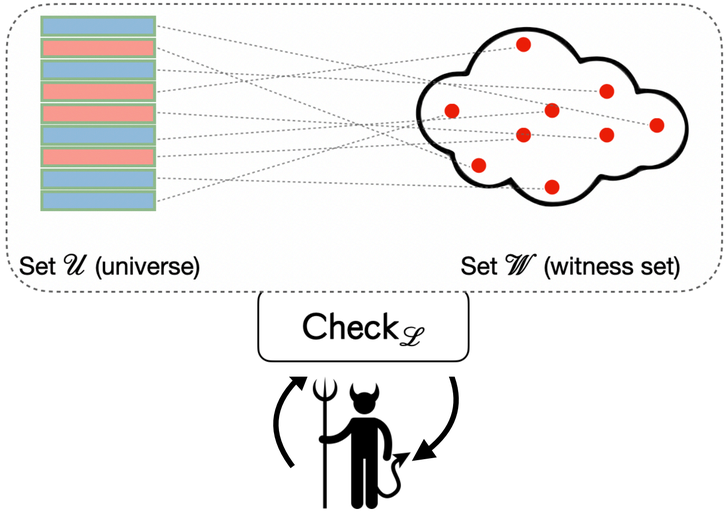
Thank You for Your Attention!



We ask whether strong average-case hardness implies weak OWFs



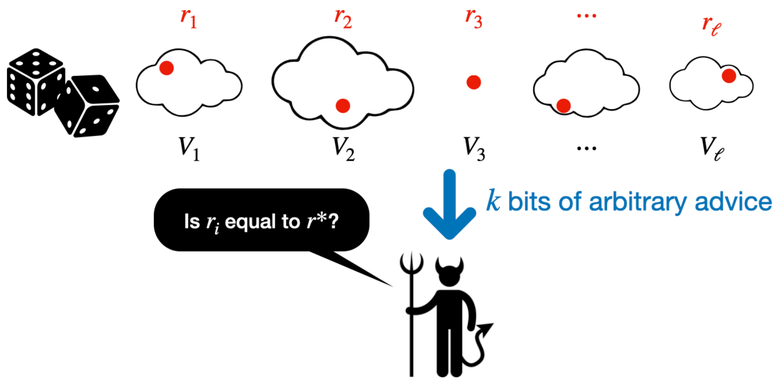
We focus on *fine-grained* OWFs and study various notions of extreme average-case hardness



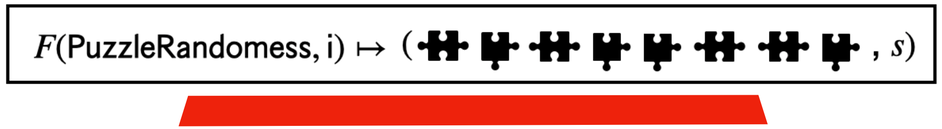
We use a random language model as a tool to study extreme average-case hardness notions and devise black-box separations

---

We introduce the *hitting lemma,* and get



One positive result from block-finding hardness



Two black-box separations from weaker hardness