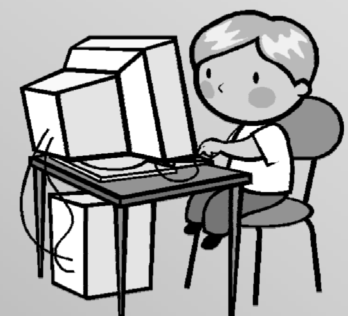
Machine Learning and Reductions in Cryptography

Nick Hopper, CMU
(hopper@cs.cmu.edu)

Cryptography:

"Computing in the presence of an adversary"

Alice



Bob

"The Adversary"

Outline of a cryptography paper

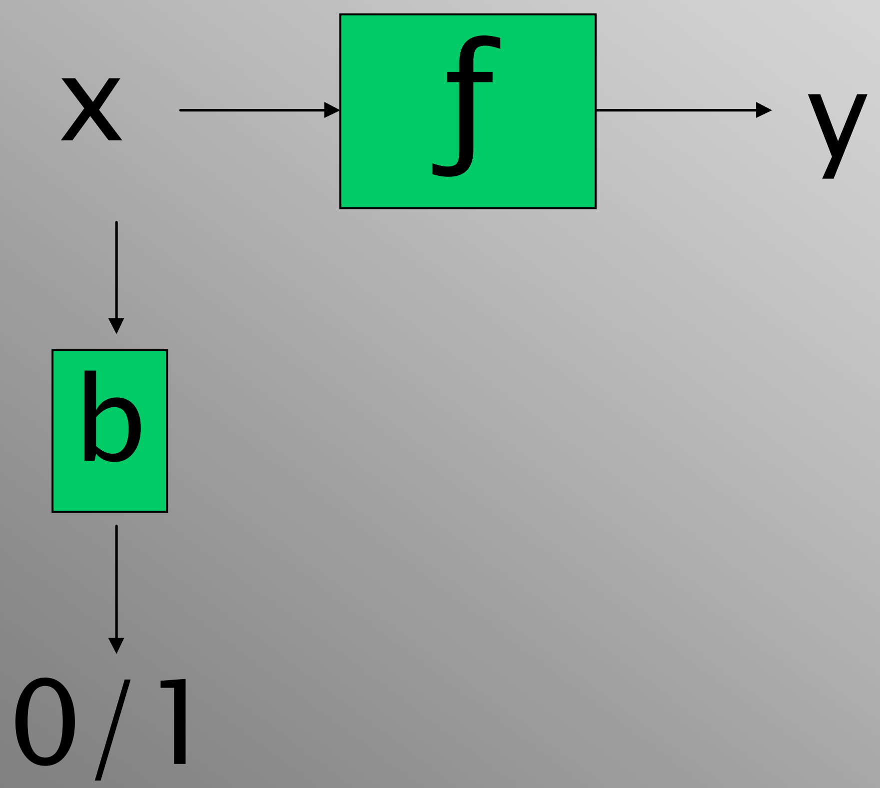
Outline of a cryptography paper

- **Start with a learning problem P**
  - hard to learn even weakly

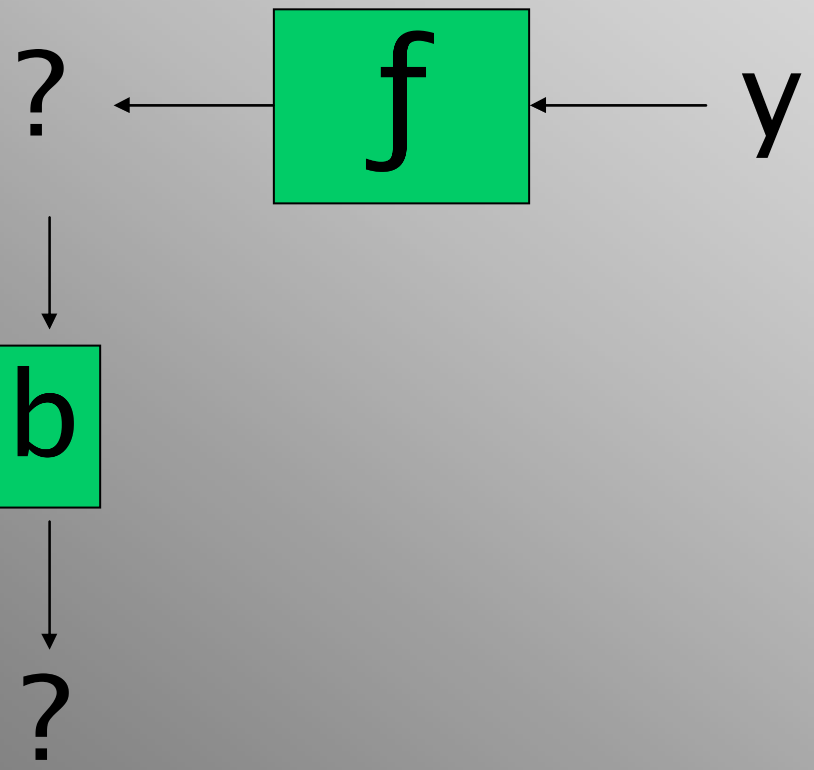
Outline of a cryptography paper

- **Start with a learning problem P**
  - hard to learn even weakly
- An efficiently computable bijection $f : \{0,1\}^n \rightarrow \{0,1\}^n$ is *one-way* if it is hard to weakly predict f^{-1} on the uniform distribution.
- A predicate $b: \{0,1\}^n \rightarrow \{0,1\}^n$ is *hard-core* for f if it is hard to weakly predict $b(x)$ given $f(x)$ (on uniform x)

Example



Example



Outline of a cryptography paper

- Start with a hard problem, P
- **Define a new kind of problem, Q**
  - hard to learn, even weakly

Outline of a cryptography paper

- Start with a hard problem, P
- **Define a new kind of problem, Q**
  - New security goal: (something else) is also hard to learn, even weakly
- A function $G : \{0,1\}^n \to \{0,1\}^{2n}$ is pseudorandom if it is hard to distinguish G(X) for uniform X from a uniform choice of 2n-bit Y.
- $G(\{0,1\}^n)$ is hard to weak-learn for an instance distribution that is ½G(X), ½Y

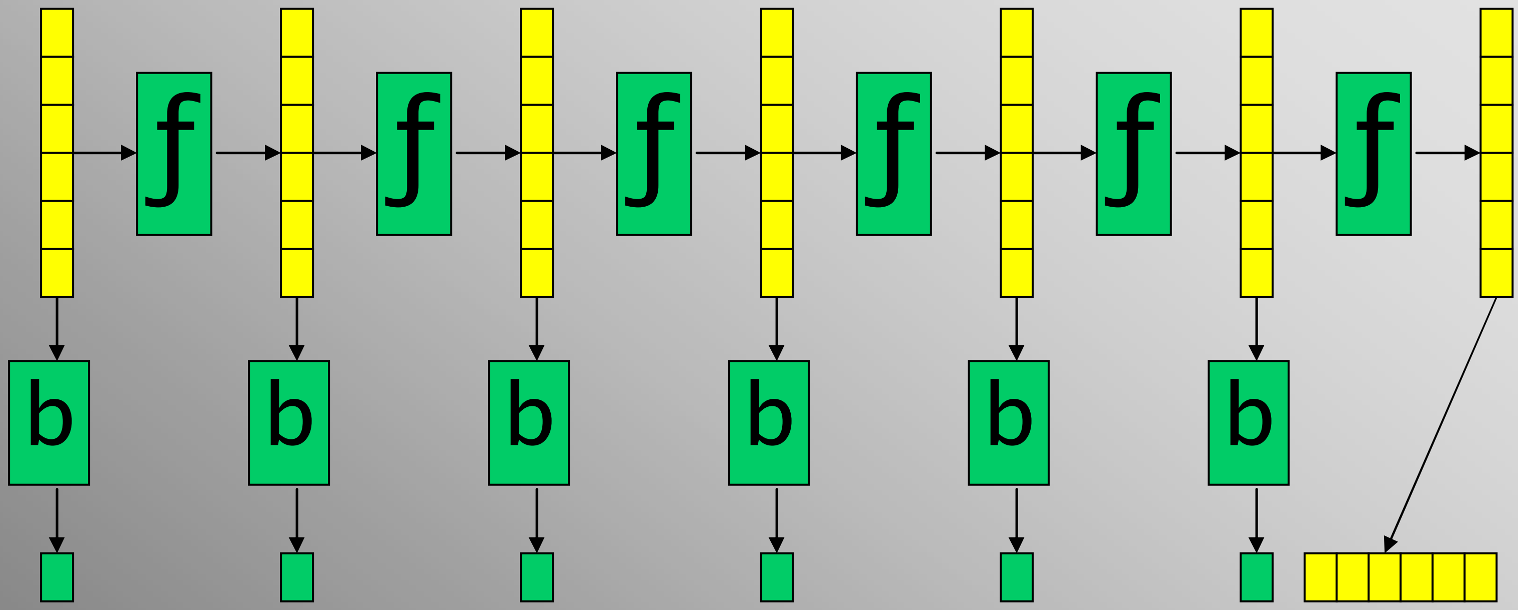
Example

AI

Outline of a cryptography paper

- Start with a hard problem, P
- Define a new kind of problem, Q
- **Give a construction, C; and associated learning problem Q_C**

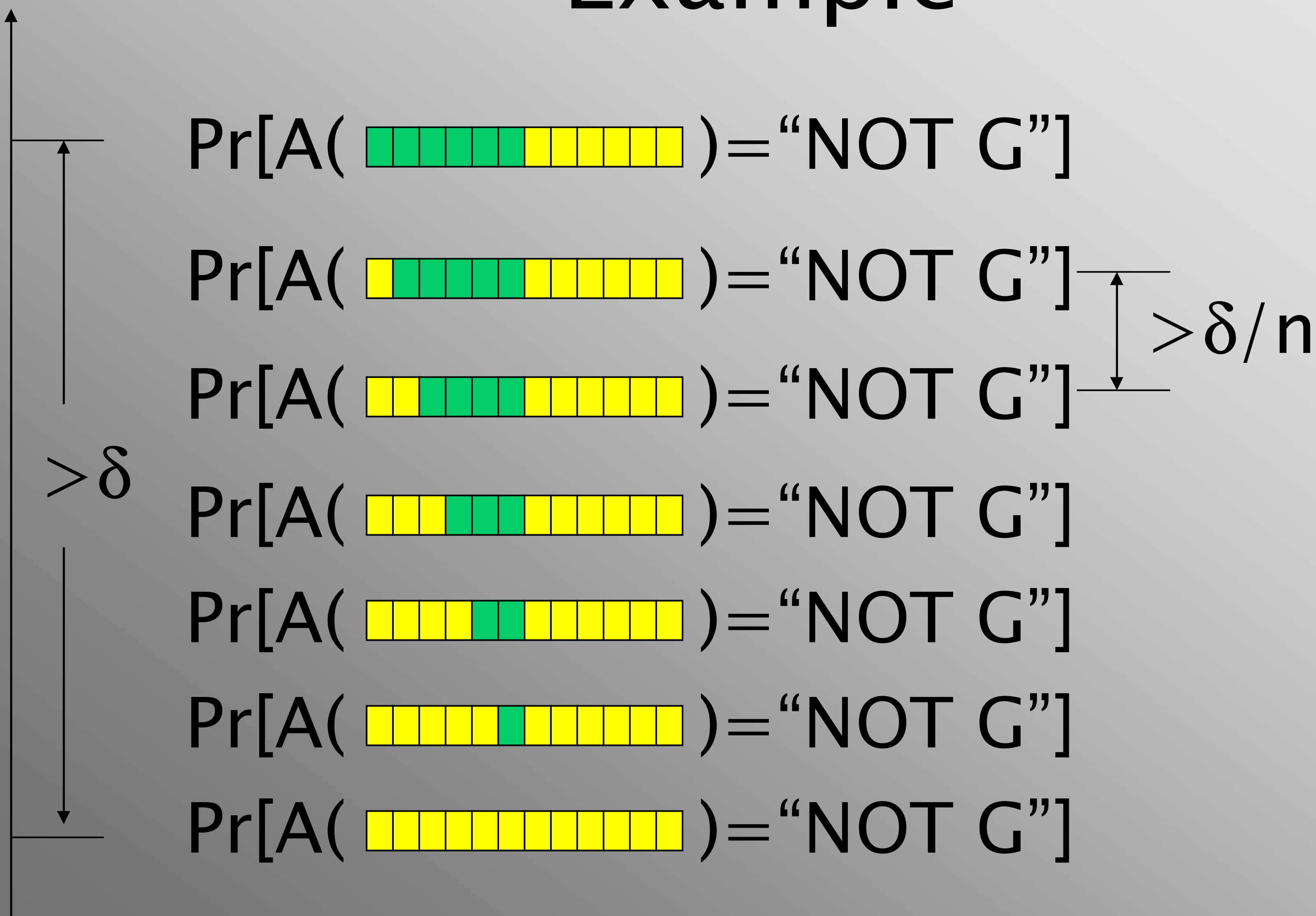
Example



Outline of a cryptography paper

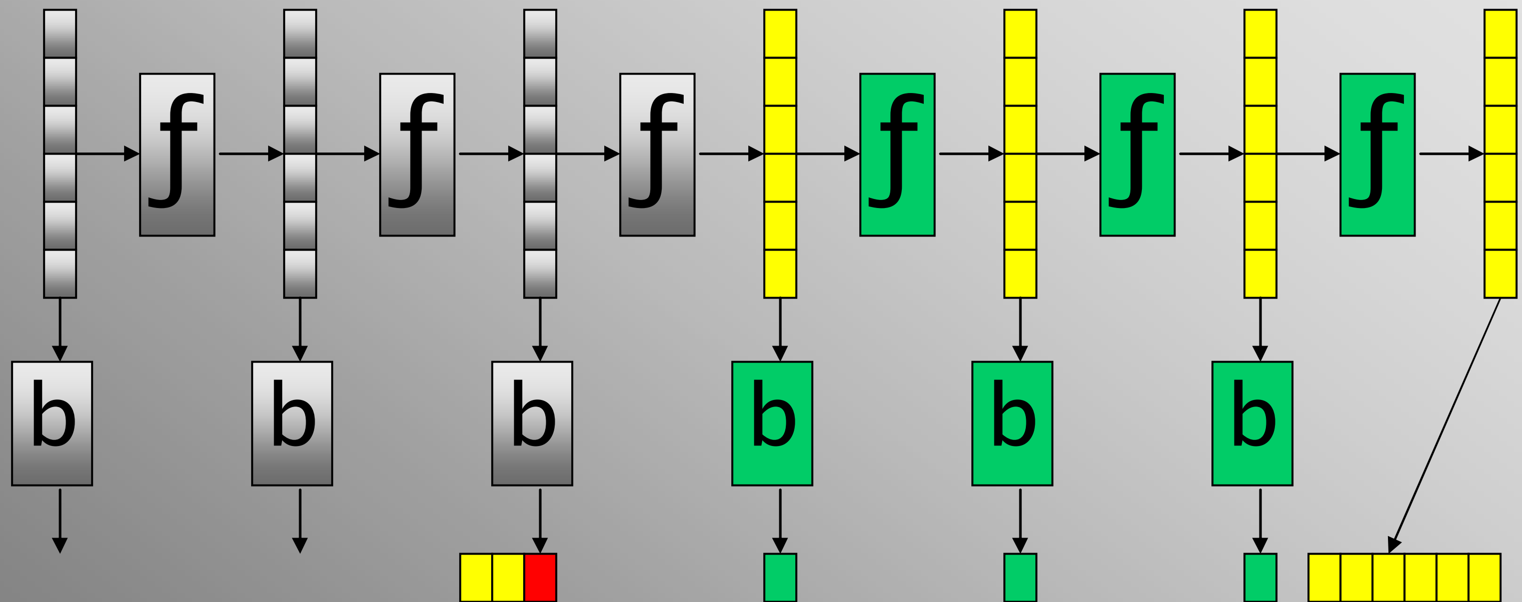
- Start with a hard problem, P
- Define a new kind of problem, Q
- Give a construction, C; and associated learning problem Q_C
- **Prove a reduction between weak learning Q_C and learning P.**

Example



Pr[A( )="NOT G"]
Pr[A( )="NOT G"]
Pr[A( )="NOT G"]
Pr[A( )="NOT G"]
Pr[A( )="NOT G"]
Pr[A( )="NOT G"]
Pr[A( )="NOT G"]
>δ/n
>δ

To predict b(x)



Outline of a cryptography paper

- Start with a hard problem, P
- Define a new kind of problem, Q
- Give a construction, C; and associated learning problem Q_C
- Prove a reduction between weakly learning Q_C and learning P.
- **Conclude that, since P is hard to learn, C is secure against an adversary learning Q.**

Outline of a cryptography paper

Cryptographers have been writing
"Machine Learning Reductions"
For the past 20 years!

- *Prove a reduction between weak learning Q_C and learning P.*
- Conclude that, since P is hard to learn, C is secure against an adversary learning Q.

Crypto Reduction Toolkit

- Vocabulary of standard reduction "patterns"
- Extensive literature of reductions
  - Including some incorrect results…
- Notational syntax for describing the relationship that a reduction gives between two problems

Reduction "Patterns"

- The Hybrid Argument
- The Simulation Argument
- The "Series of Games" Argument (really a Hybrid)
- The "Random Oracle Methodology" (only a *heuristic* argument!)

The Hybrid Argument

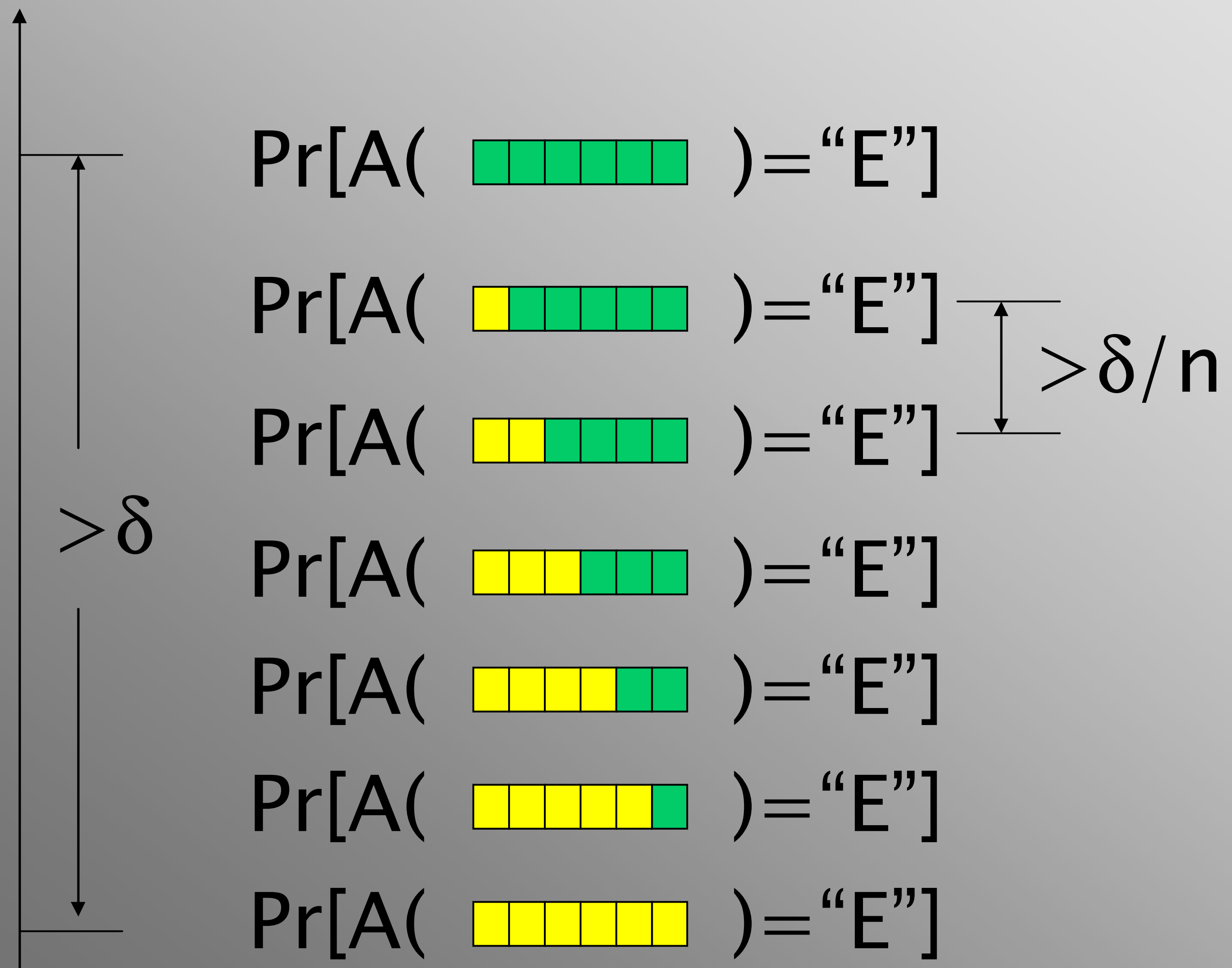
- Let $\mathcal{D}$ and $\mathcal{E}$ be distributions.

$\mathcal{D}$ $\mathcal{E}$

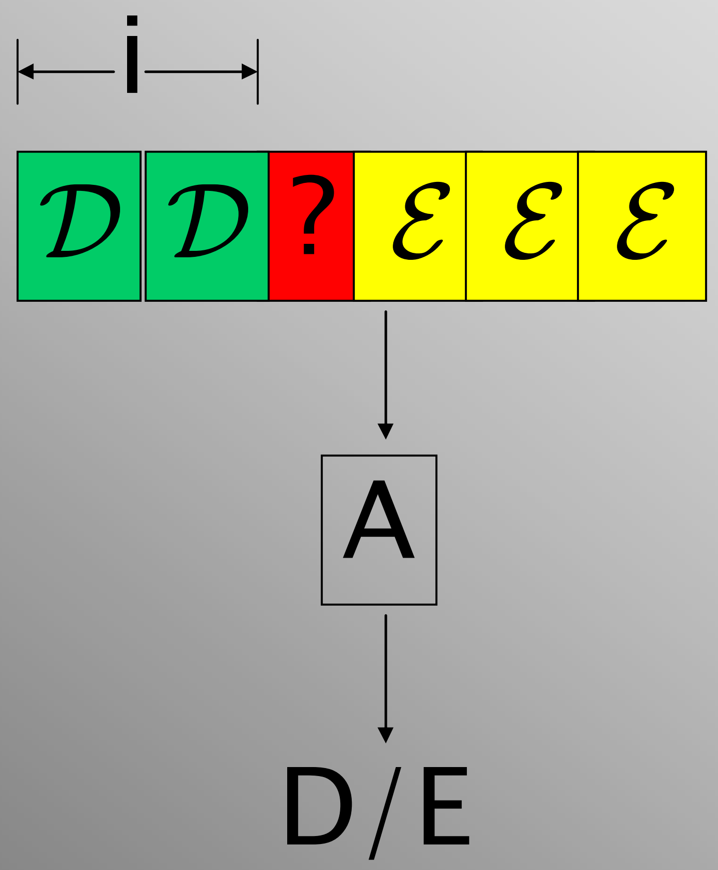
- If A can distinguish between $\mathcal{D}^n$ and $\mathcal{E}^n$, A can be used to distinguish between $\mathcal{D}$ and $\mathcal{E}$.

$\mathcal{D}\,\mathcal{D}\,\mathcal{D}\,\mathcal{D}\,\mathcal{D}\,\mathcal{D}$ $\mathcal{E}\,\mathcal{E}\,\mathcal{E}\,\mathcal{E}\,\mathcal{E}\,\mathcal{E}$

Hybrids



Distinguishing $\mathcal{D}$ from $\mathcal{E}$:



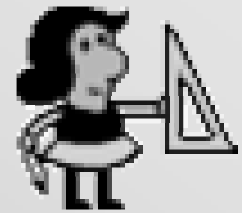
Simulations

- Show that interacting with C can be simulated noninteractively with an instance of P
- So (online) weak-learning Q_C implies weak-learning P offline.

Example: How to prove $\sqrt{a}$ exists mod N

P

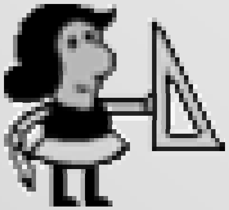
(knows $a=r^2$ mod N)



V

Example: How to prove $\sqrt{a}$ exists mod N

P



V

(knows $a=r^2$ mod N)

Picks random x, sets $y=x^2$.

y has a square root, too.

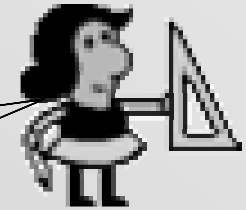
Example: How to prove $\sqrt{a}$ exists mod N

P

(knows $a=r^2 \bmod N$)

Picks random x, sets $y=x^2$.

Oh yeah? What's $\sqrt{w}$?



V

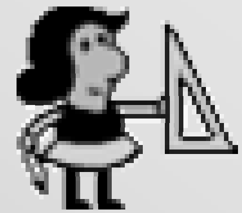
Picks a random bit, b sets $w=a^b y$.

Example: How to prove $\sqrt{a}$ exists mod N

P

(knows $a=r^2$ mod N)

Why, it's v, of course.



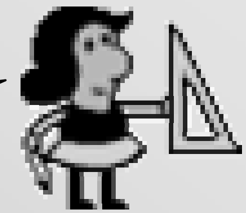
V

Picks random x, sets $y=x^2$.

Picks a random bit, b sets $w=a^b y$.

sets $v=r^b x$.

Example: How to prove $\sqrt{a}$ exists mod N



Interesting…

P

V

(knows $a=r^2$ mod N)

Picks random x, sets $y=x^2$.

Picks a random bit, b sets $w=a^b y$.

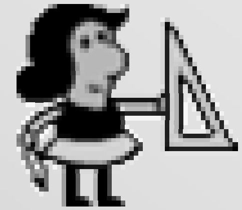
sets $v=r^b x$.

Checks $v^2=w$

Simulated Professor

P

Claim: Anything Vicky learns from the professor, she could learn herself



V

(knows $a=r^2 \bmod N$)

Picks random x,
sets $y=x^2$.

Picks a random bit, b
sets $w=a^b y$.

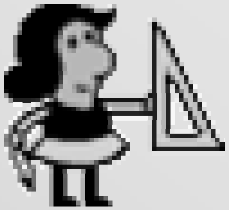
sets $v=r^b x$.

Checks $v^2=w$

Simulated Professor

Sim-P

(knows a)



V

Guesses b'.
Picks random z,
sets $u=z^2$.
Sets $y=u/a^{b'}$.

Vicky only has to guess b' twice (on average.)

Picks a random bit, b
sets $w=a^b y$.

If b=b',
sets v=z.
Else, FAIL

Checks $v^2=w$

At least 90% of crypto reductions fit the “hybrid” or “simulation” pattern

Crypto Reduction Toolkit

- Vocabulary of standard reduction "patterns"
- Extensive literature of reductions
  - Including some incorrect results…
- **Notational syntax for describing the relationship that a reduction gives between two problems**

Notational Convergence

- An "emergent standard" of notation
- Suppose A weak-learns P using resources r with advantage ε, this is denoted by

  $$\mathrm{Adv}_A^P(r) \leq \varepsilon\ .$$

  To state reduction between P and Q,

  $$\mathrm{Adv}_P(r) \leq f(\mathrm{Adv}_Q(g(r))$$

  meaning given machine A for (r,ε) learning P, we give B that $(g(r),f(\varepsilon))$-learns Q.
- Results are easier to compare…

Workshop Central Questions

1. What empirical evidence do we have for and against reductions?
2. How can we differentiate between "good" reductions and "bad" reductions?
3. Can we understand the difficulty of learning problems by their placement in a reduction hierarchy?
4. What is a mapping of available reductions?
5. What learning problems are unrelated by reduction?
6. What reductions are not possible?

What is a “good” reduction?

To practitioners, a “good” reduction is:

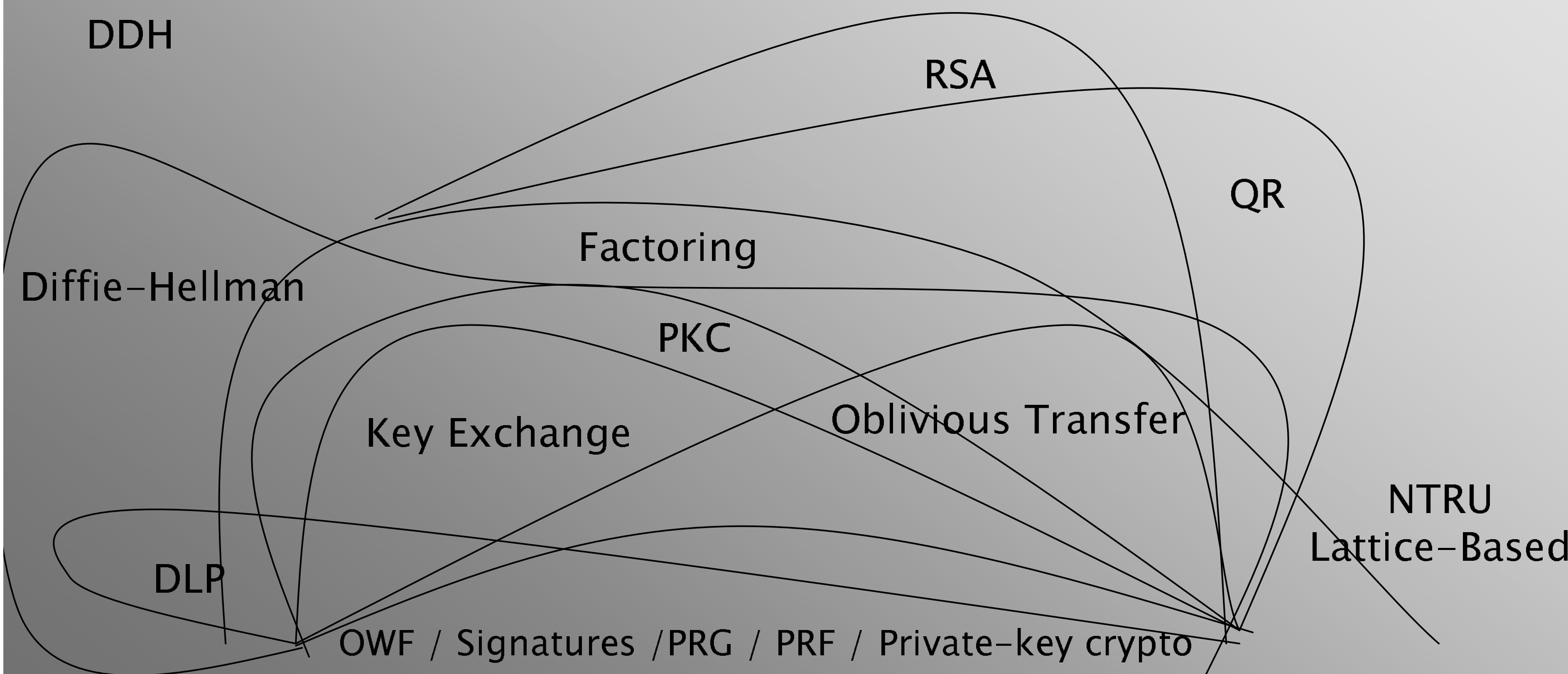
- **correct.**
- **security-preserving.**
- **efficient.**
- General – works with many primitives
- Easily understood/verified

Security Preservation and Efficiency

- Suppose we can (f(r),g(ε))-learn P given an (r,ε)-learner for Q. then the reduction is:
  - "linear preserving" – f and g are linear
  - "poly-preserving" – g linear, f is poly
  - "weak-preserving" – f and g are poly
- Typically only "linear-preserving" reductions are considered useful.

Reductions and difficulty

"Can we understand the difficulty of learning problems by their placement in a reduction hierarchy?"



So… what?

- Reductions can establish that a problem is minimal.
- Reductions can establish that a problem is hard.
  - Some learning problems have already been shown hard via cryptographic reductions
- Reductions probably won't give an exact hierarchy…

What reductions are not possible?

Cryptographer's approach: prove that a reduction would be "surprising":

- Boneh: an "algebraic" reduction from factoring to breaking RSA implies factoring is easy.
- Impagliazzo-Rudich: if P=NP, there is no reduction from breaking a KA protocol to breaking a OWP
- Gennaro-Trevisan: an "efficient" construction of PRG from OWP implies an unconditional OWP!

The Literature

- A. Blum et al, “Cryptographic Primitives Based on Hard Learning Problems”
  - www.cs.cmu.edu/~avrim/Papers/cryptofinal.ps
- Goldwasser and Bellare, “Lecture Notes on Cryptography”
  - www-cse.ucsd.edu/~mihir/papers/gb.html
- Oded Goldreich, “Foundations of Cryptography”