Crypto in the Cloud workshop, MIT
August 4, 2009

Protecting Circuits from Computationally-Bounded Leakage

Eran Tromer — MIT

Joint work with
Sebastian Faust — K.U. Leuven
Leo Reyzin — Boston University

CSAIL
MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

Motivation

The great tragedy of Crypto –

the slaying of a provably secure scheme

by an ugly side channel.

Engineering approach

- Try preventing leakage.

  Imagine a list of

  - *all known side channel attacks*
  - *all new attacks during the device's lifetime.*

- Good luck.

Cryptographic approach

- Face the music: computational devices are not black-box.
- Leakage is a *given*, i.e., modeled by an adversarial observer.
  The device should protect itself against it.

Cryptographic Machinery

- Standard toolbox against polynomial-time adversaries (obfuscation, oblivious RAM, fully-homomorphic encryption).
  - Minimize assumptions on adversary's power.
  - Looks hard/impossible/expensive to realize.
  - Worth exploring!
- New tools for a new setting
  - Model the leakage more finely
    - What leaks
    - How much leaks
    - How is the leakage chosen
  - Devise ways to make **specific functionality**, or even **arbitrary circuits**, resilient to such leakage.

Related Work

[CDHKS00]: Canetti, Dodis, Halevi, Kushilevitz, Sahai: Exposure-Resilient Functions and All-Or-Nothing Transforms

[ISW03]: Ishai, Sahai, Wagner: Private Circuits: Securing Hardware against Probing Attacks

[MR04]: Micali, Reyzin: Physically Observable Cryptography

[GTR08]: Goldwasser, Tauman-Kalai, Rothblum: One-Time Programs

[DP08]: Dziembowski, Pietrzak: Leakage-Resilient Cryptography in the Standard Model

[Pie09]: Pietrzak: A leakage-resilient mode of operation

[AGV09]: Akavia, Goldwasser, Vaikuntanathan: Simultaneous Hardcore Bits and Cryptography against Memory Attacks

[ADW09]: Alwen, Dodis, Wichs: Leakage-Resilient Public-Key Cryptography in the Bounded Retrieval Model

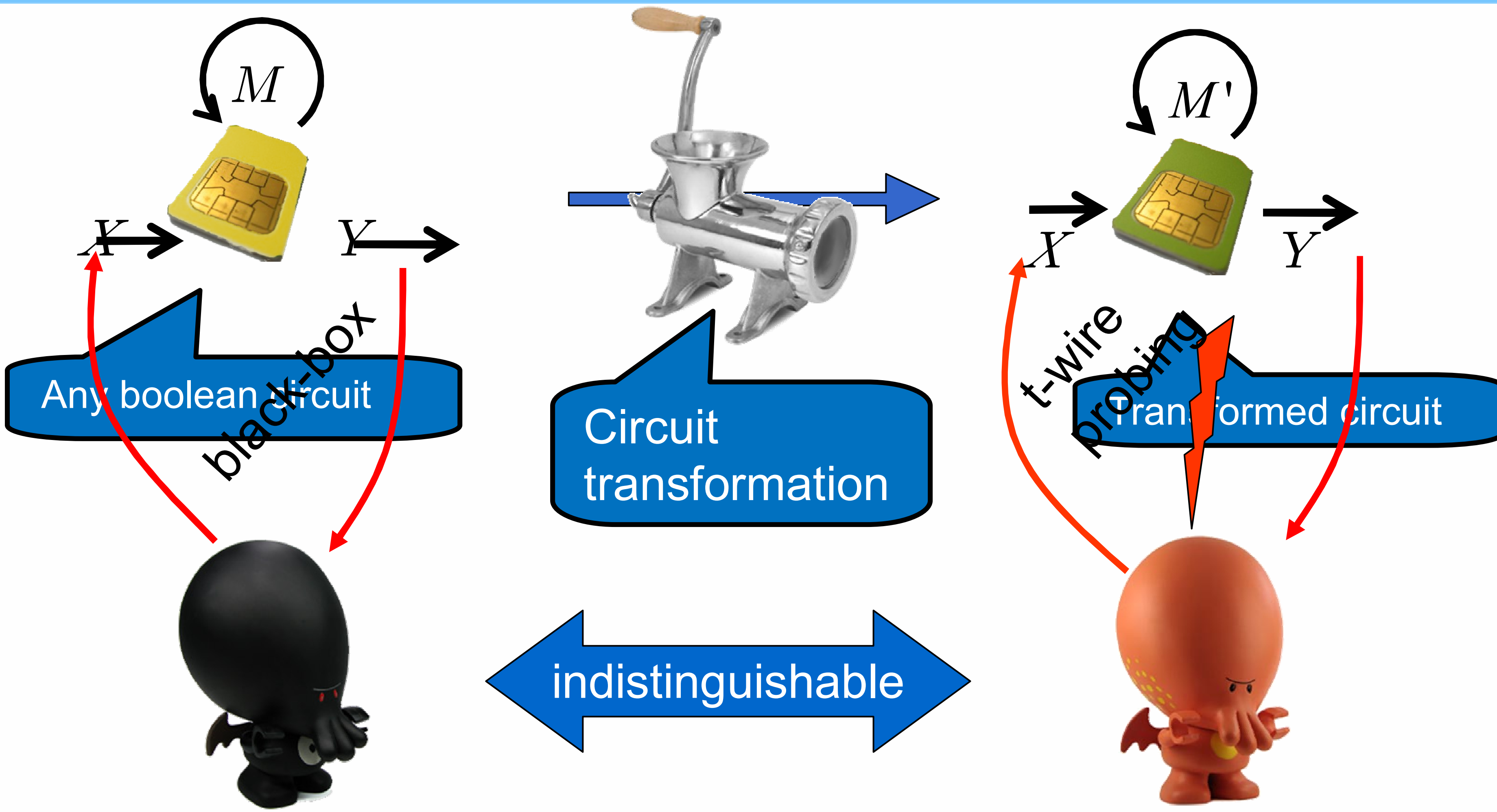
[FKPR09]: Faust, Kiltz, Pietrzak, Rothblum: Leakage-Resilient Signatures

[DHT09]: Dodis, Lovett, Tauman-Kalai: On Cryptography with Auxiliary Input

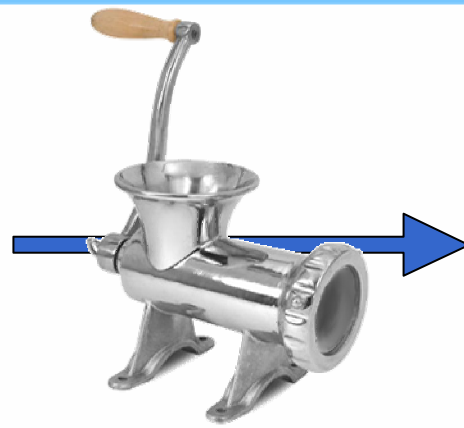
[SMY09]: Standaert, Malkin, Yung: A Unified Framework for the Analysis of Side-Channel Key-Recovery Attacks

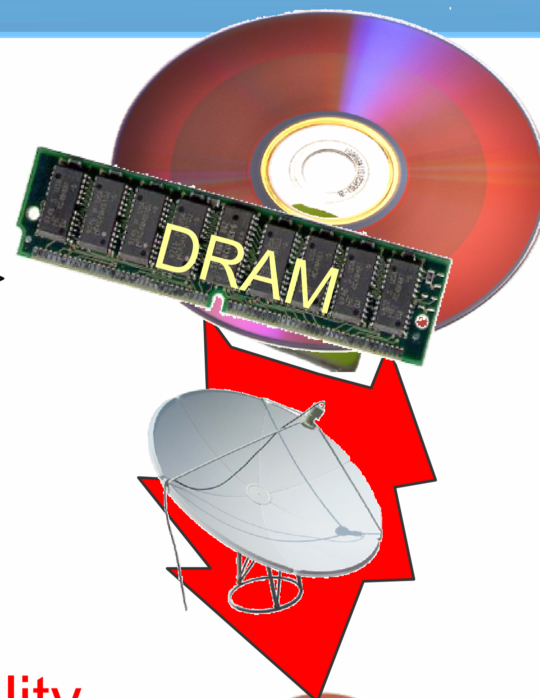
...

[Ishai Sahai Wagner '03]



Our goal





Allow much stronger leakage.

In particular, don't assume spatial locality

- t wires [ISW03]
- "Only computation leaks information" [MR04][DP08][Pie09][FKPR09]

Our main construction

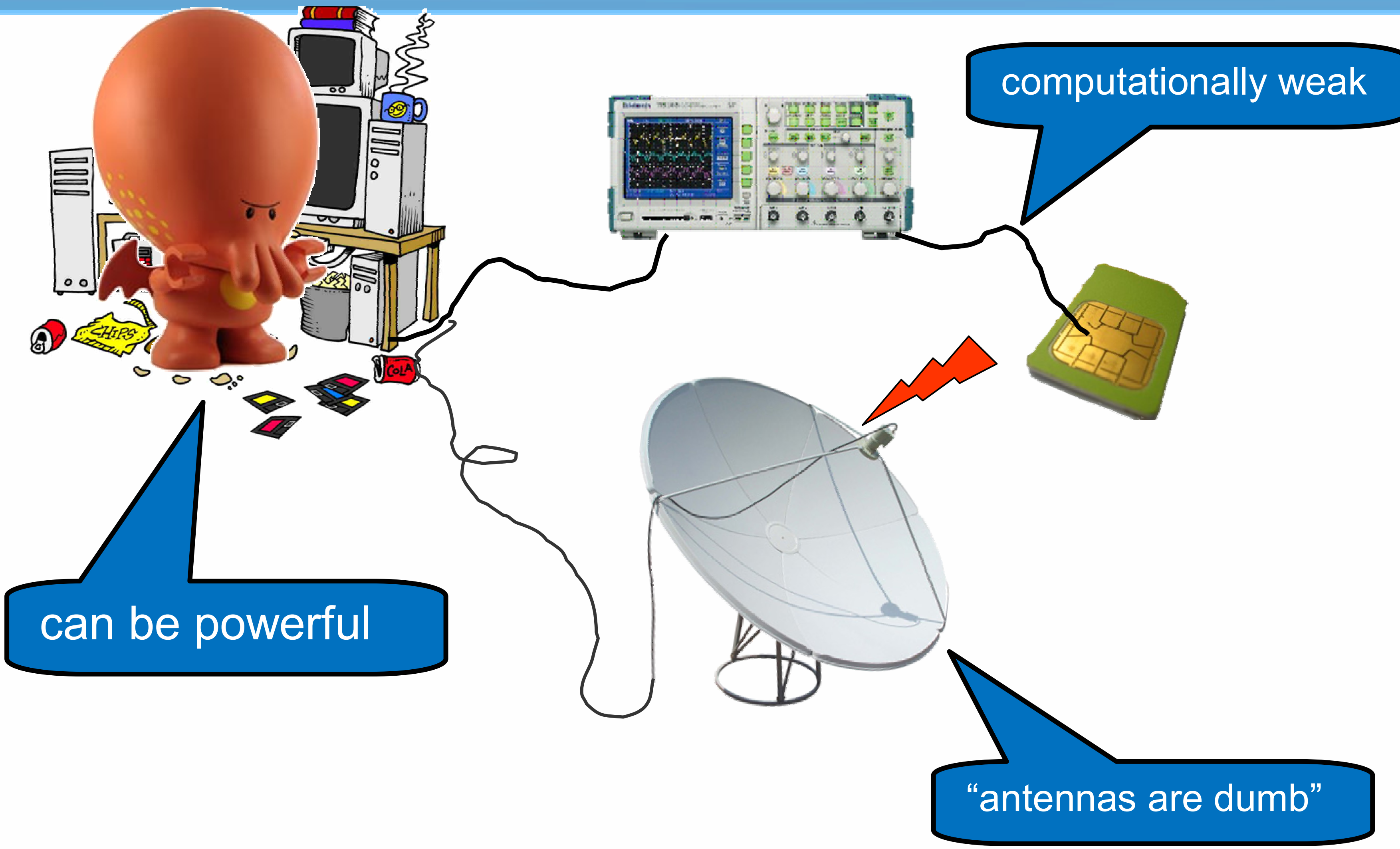
A transformation that makes **any circuit** resilient against

- **Global adaptive leakage**
  May depend on whole state and intermediate results, and chosen adaptively by a powerful on-line adversary.
- **Arbitrary total leakage**
  Bounded just per observation. [DP08]

But we must assume something:

- **Leakage function is computationally weak** [∈MR04]
- **A simple leak-free component** [∈MR04]

Computationally-weak leakage

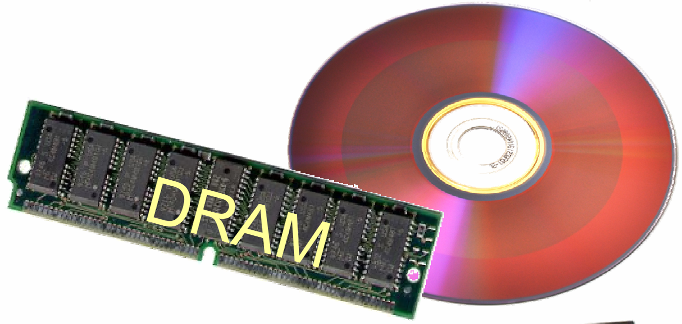


Leak-free components

- **Secure memory**

  [MR04][DP08][Pie09][FKPR09]

- **Secure processor** [G89][GO95]



- Here: simple component that samples from a fixed distribution, e.g: **securely draw strings with parity 0**.
  - No stored secrets or state
  - No input

    ⟶ Consumable leak-free "tape roll"
  - Can be relaxed

- Large leak-free components may be **necessary** in this model (more later)

Rest of this talk

1. Computation model
2. Security model
3. Circuit transformation
4. Proof approach
5. Extensions
6. Necessity of leak-free components

Original circuit

Original circuit C of arbitrary functionality (e.g., crypto algorithms). Computes over a finite field K.
Example: AES encryption with secret key M.

Original circuit

Allowed gates in C:

Multiply in K: •

Add in K: +

Coin: $

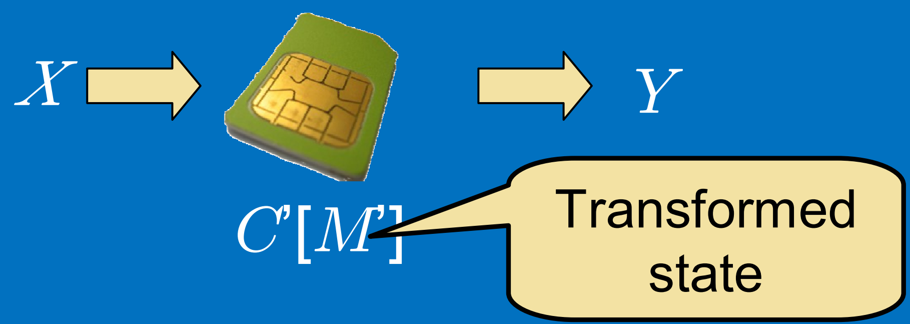
Const: 1

Memory: M

Copy: C

(Boolean circuits are easily implemented.)

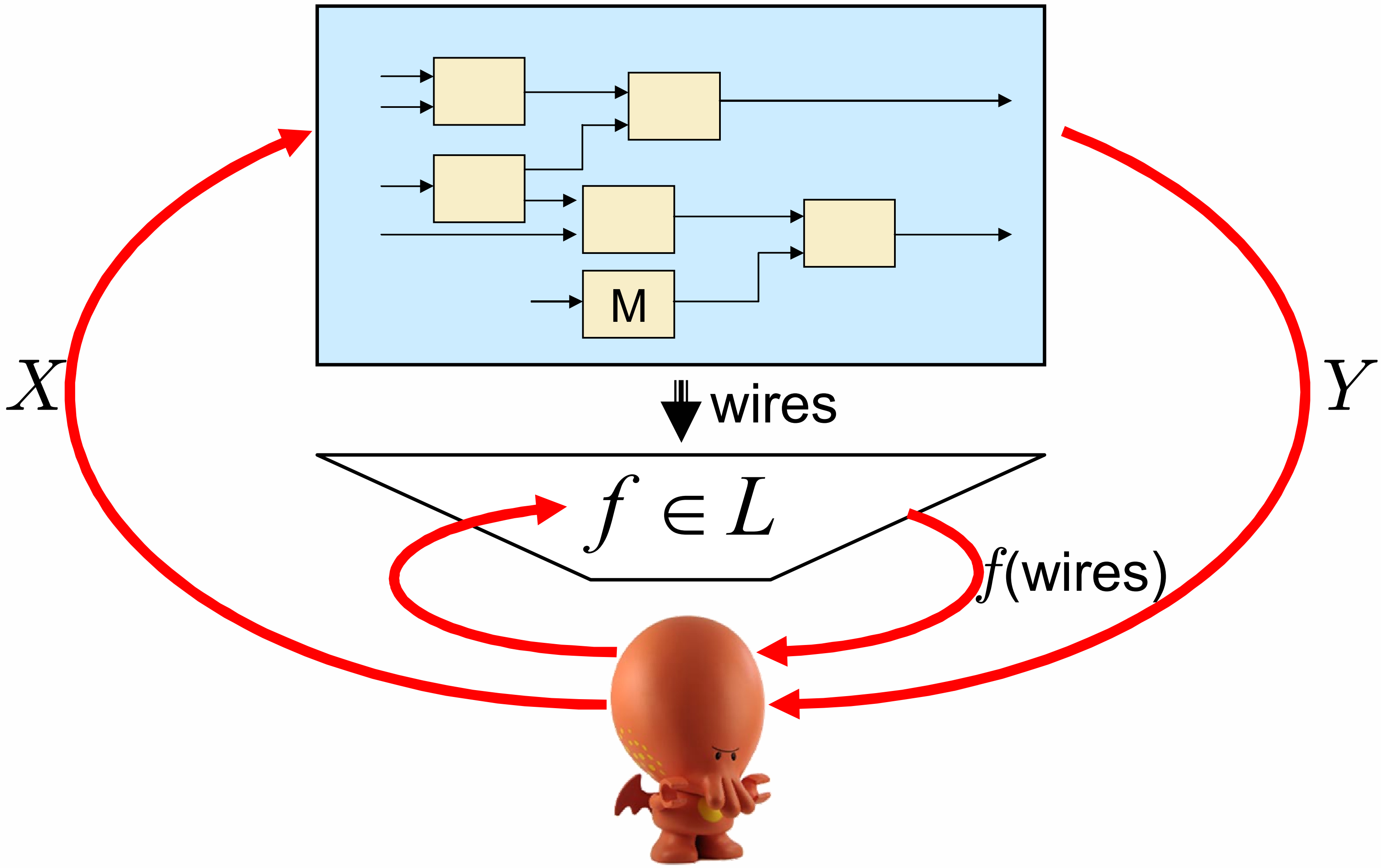
Transformed circuit [IW03]



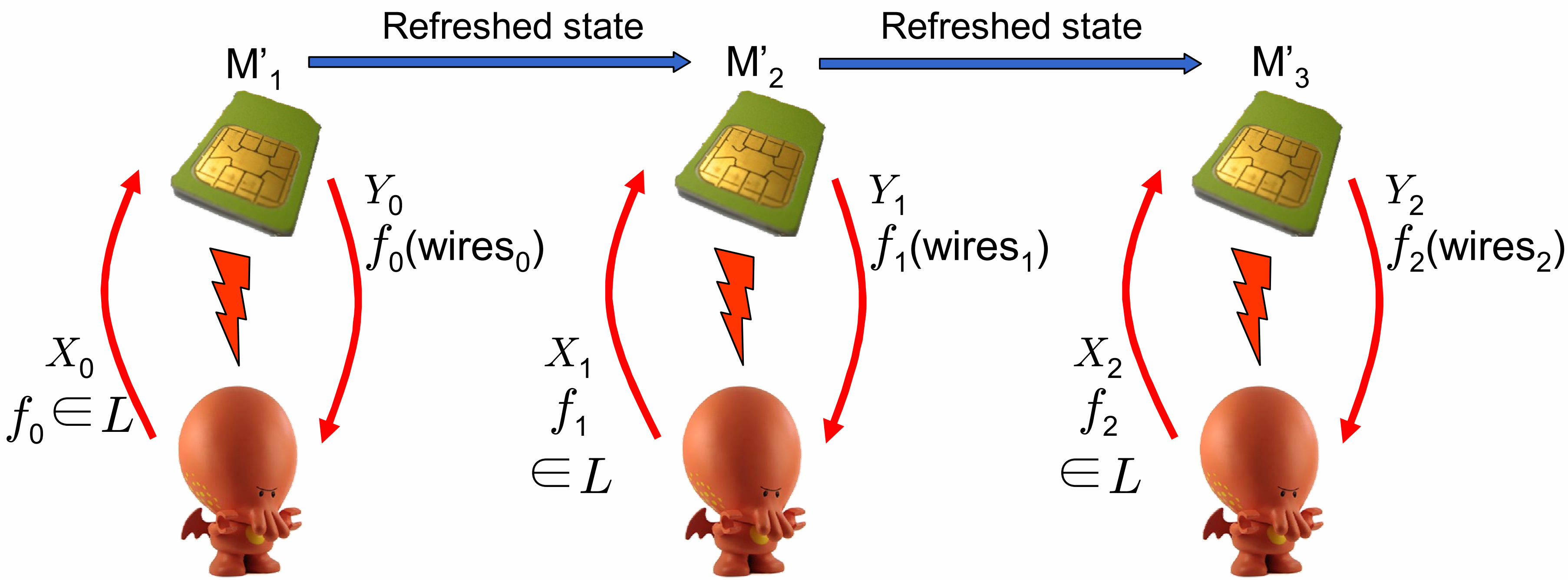
Same underlying gates as in C,
plus opaque gate (later).

Soundness: For any X,M: $C[M](X)$ = $C'[M'](X)$

Model: single observation in leakage class L



Model: adaptive observations

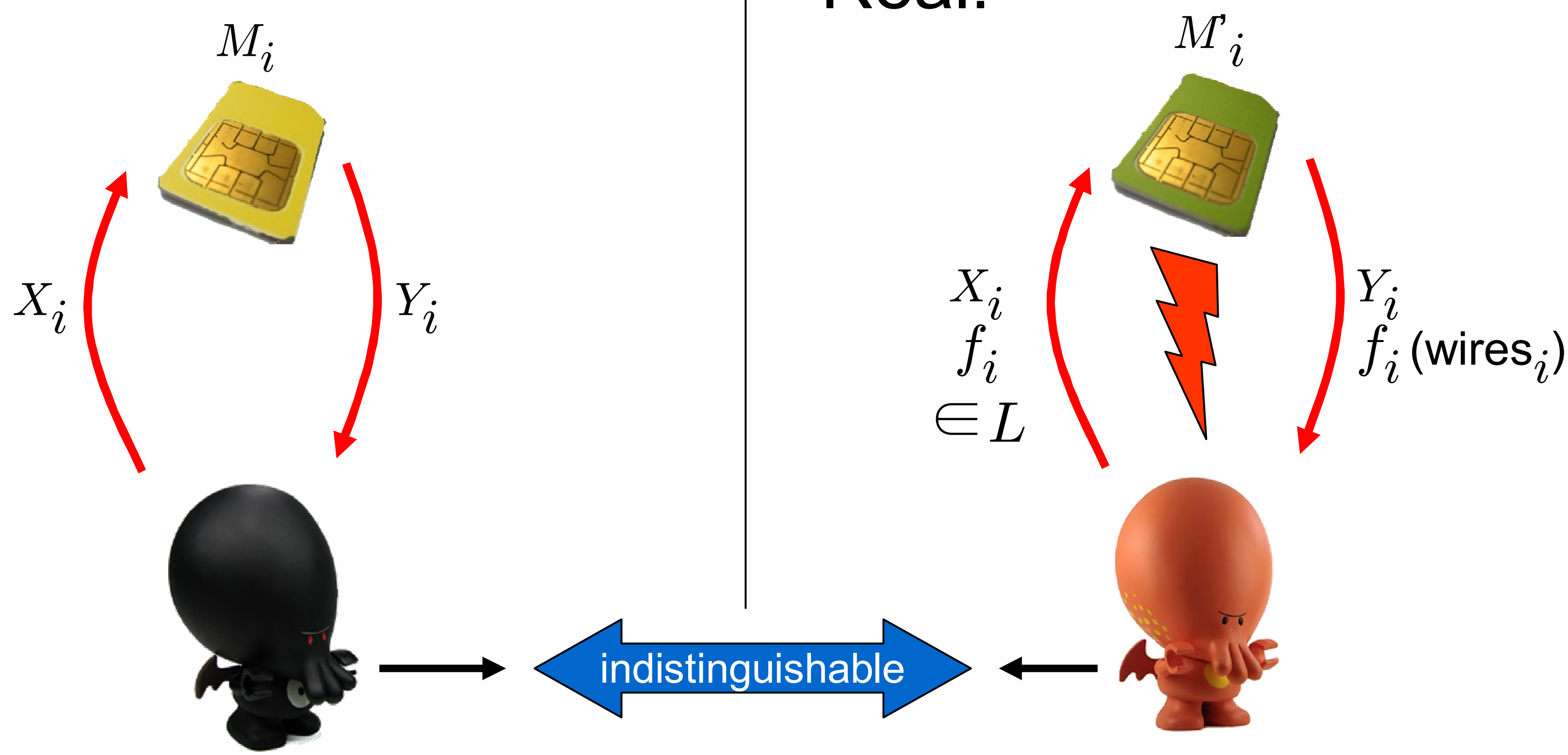


refresh state ➔ allows total leakage to be large!

Model: L-secure transformation

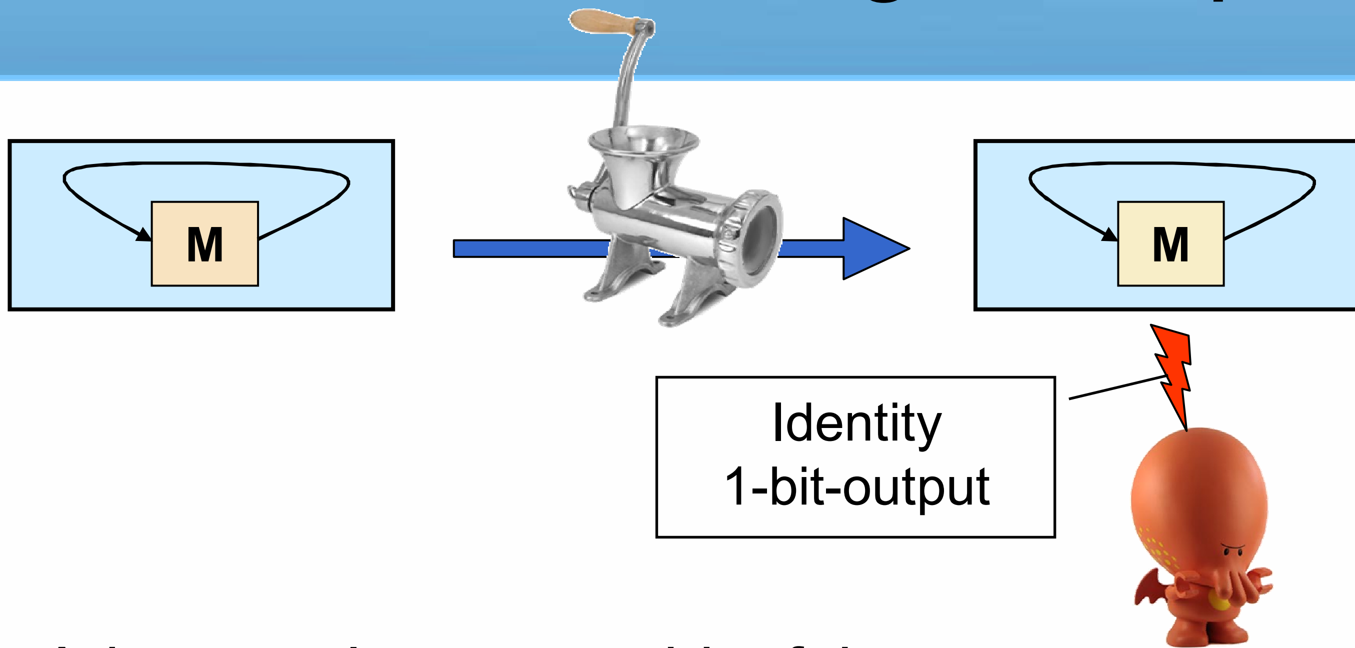
Adversary learns no more than by black-box access:

Simulation: | Real:



statistical

Motivating example



Problem: Adversary learns one bit of the state

Solution: Share each value over many wires [ISW03, generalized]

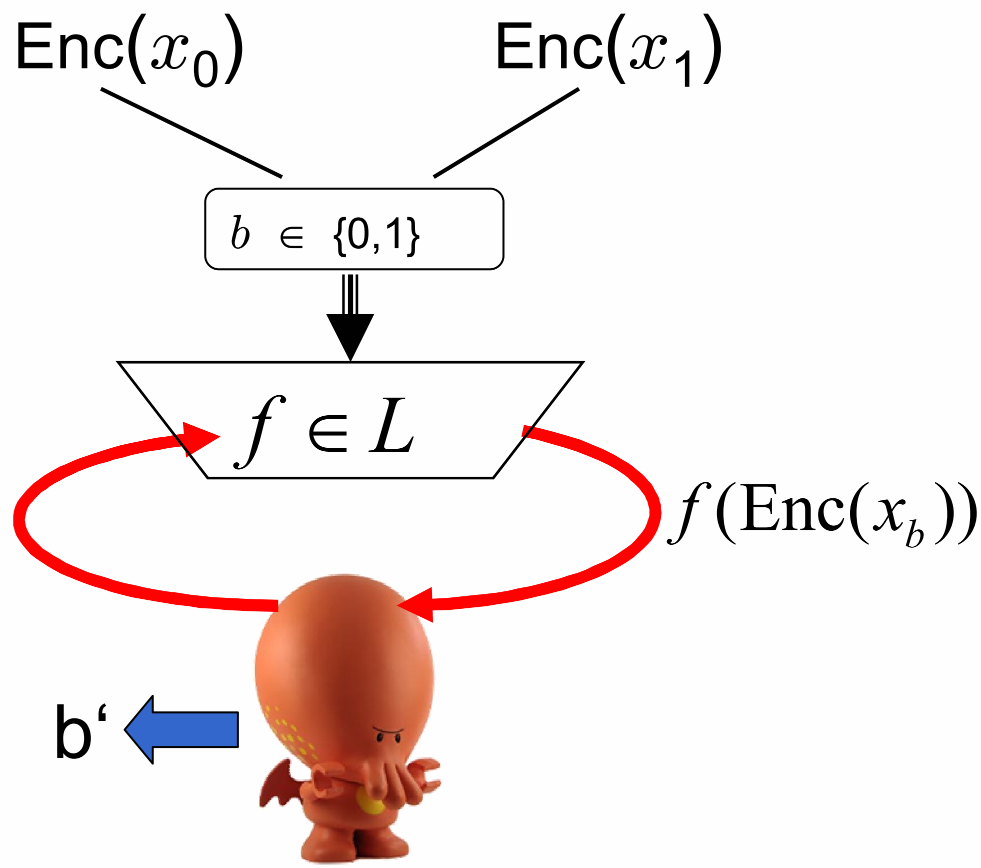
Every value encoded by a linear secret sharing scheme (**Enc**,**Dec**) with security parameter t: **Enc**: $K \to K^t$ (probabilistic)

Dec: $K^t \to K$ (surjective, linear function)

Leakage: *L*-leakage-indistinguishability

(Enc,Dec) is L-leakage-indistinguishable:

For all $x_0, x_1 \in K$:



Consequence:

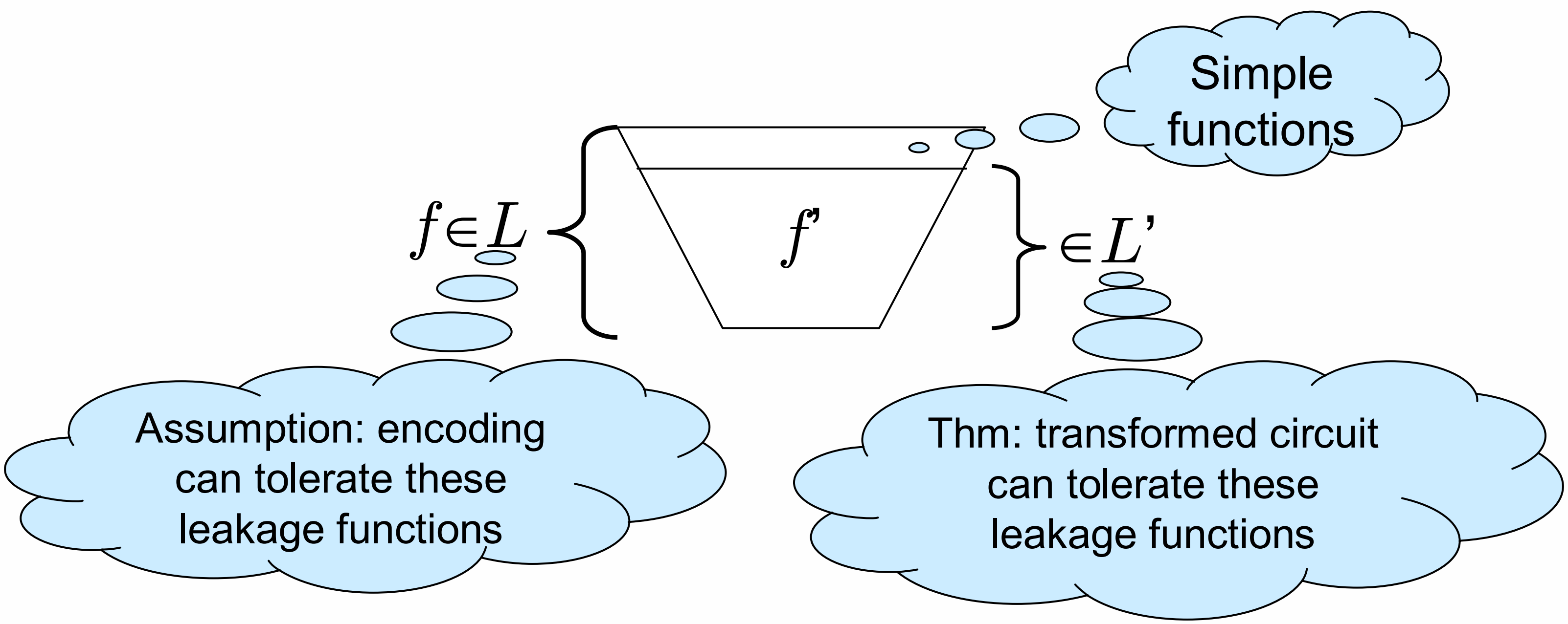
Leakage functions in L cannot decode

Pr[b' = b] - ½ ≤ negl

Main construction

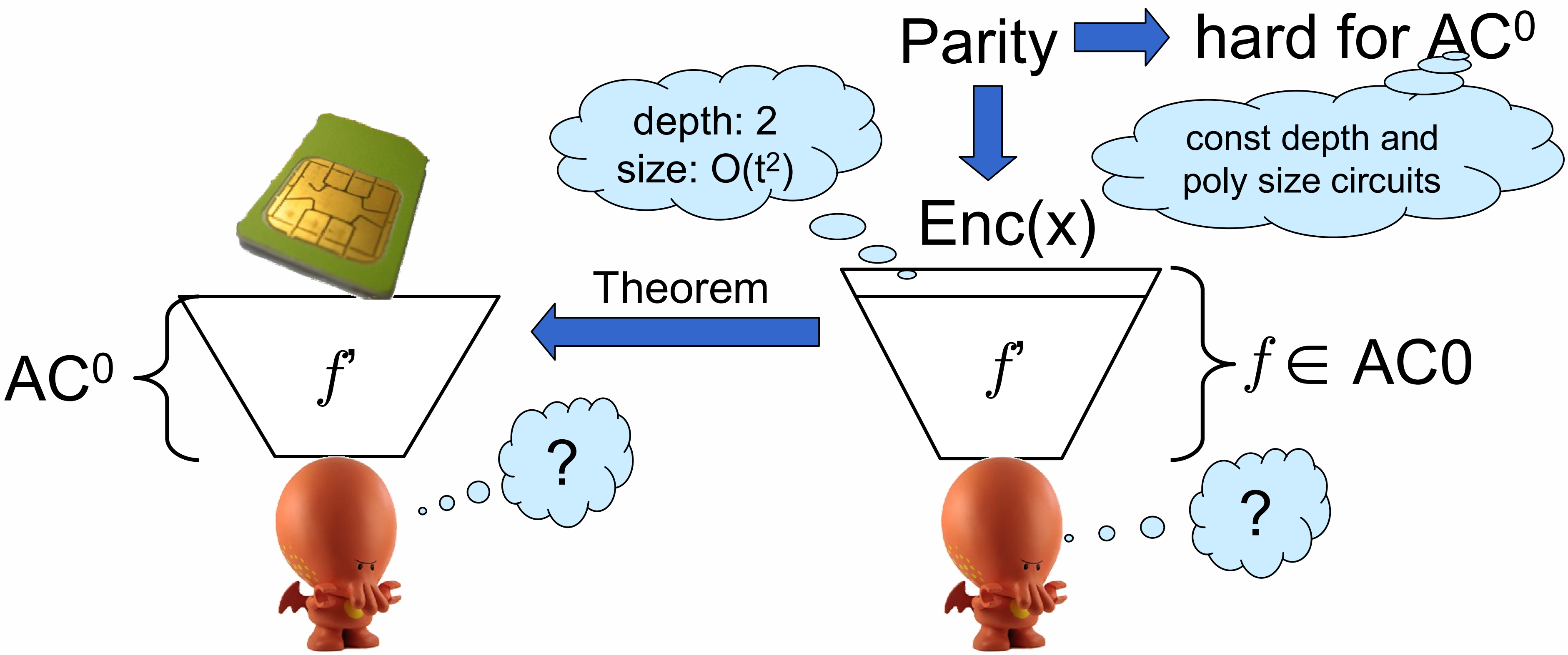
For any linear encoding scheme that is L-leakage indistinguishable

we present an L'-secure transformation for any circuit and state

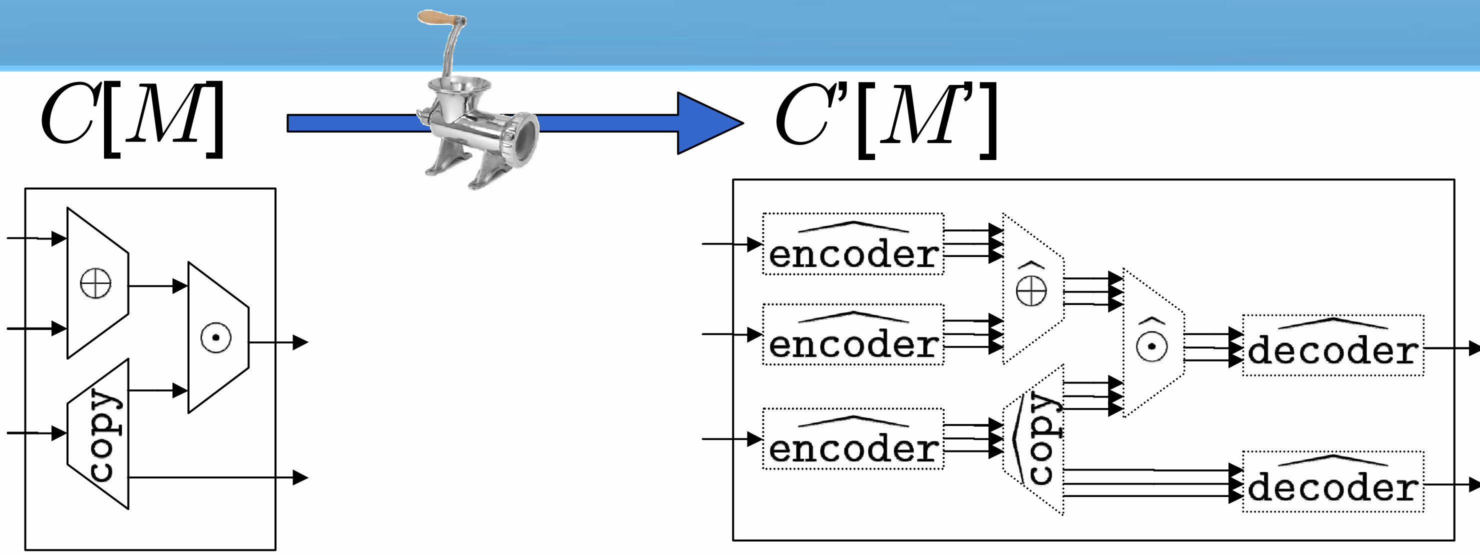


Unconditional resilience against AC^0 leakage

Some known **circuit lower bounds** imply L-leakage-indistinguishability

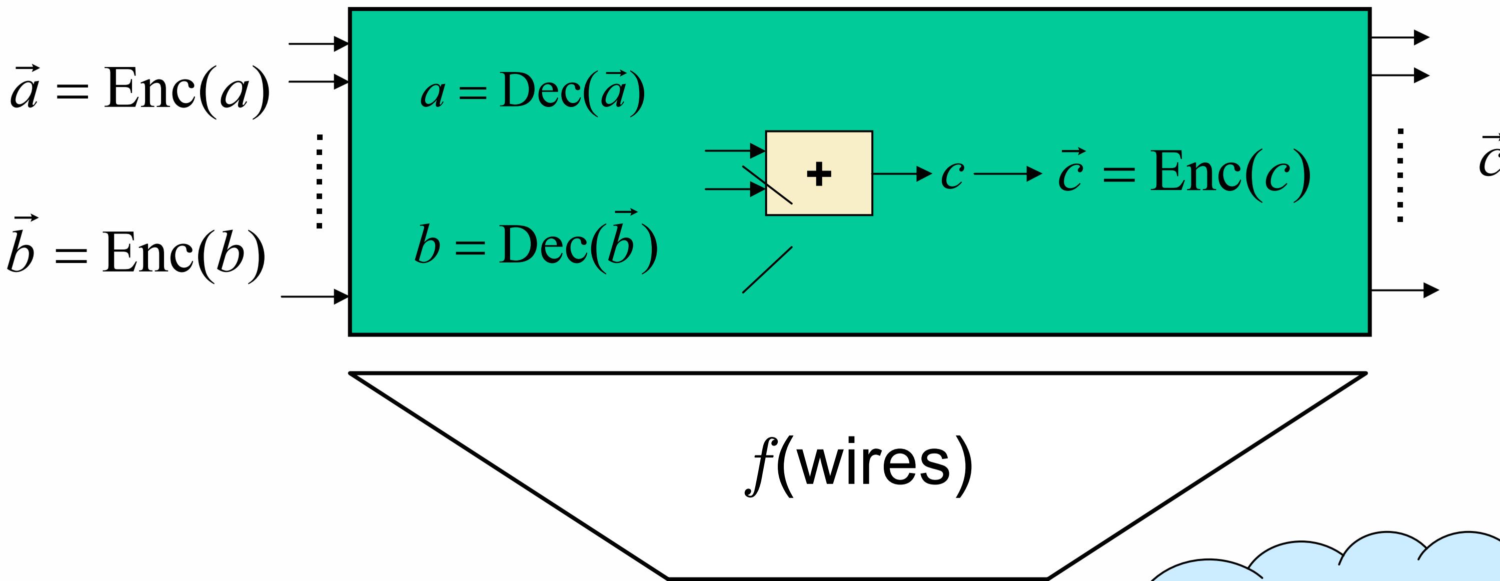


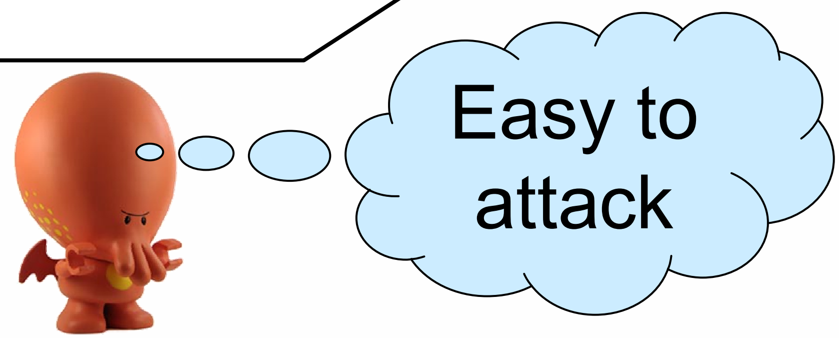
Transformation: high level



- The state is encoded: M' = Enc(M)
- Circuit topology is preserved
- Every wire is encoded
- Inputs are encoded; outputs are decoded
- Every gate is converted into a **gadget** operating on encodings

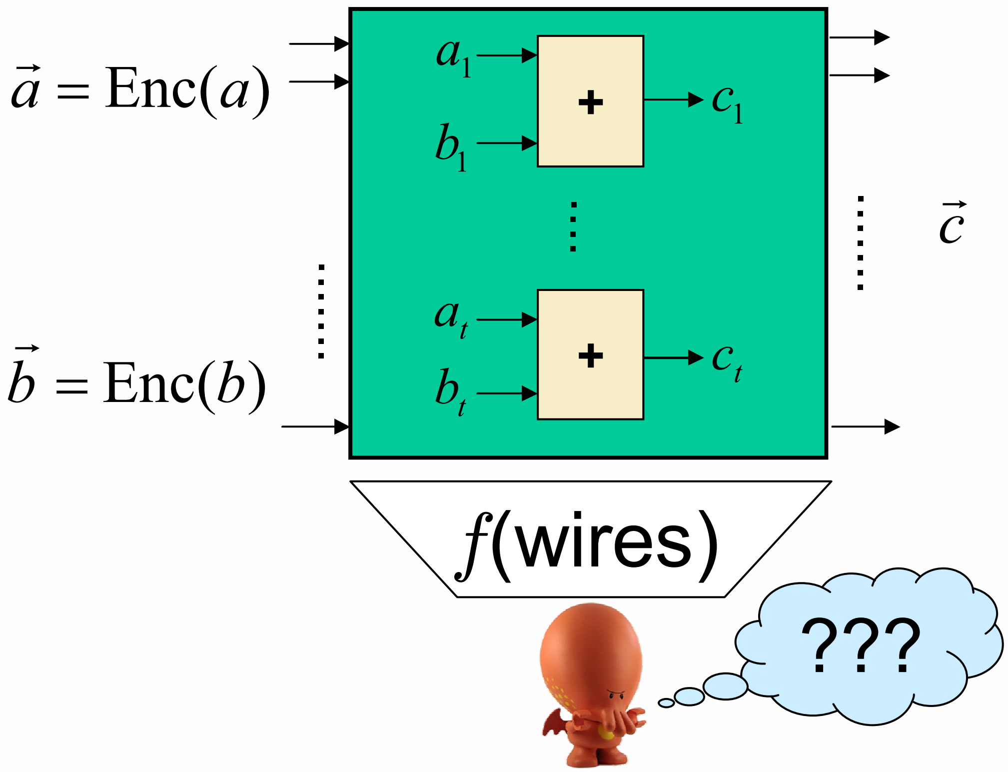
Computing on encodings *first attempt*





Notation: $\vec{x} = \mathrm{Enc}(x)$

Computing on encodings *second attempt – use linearity*



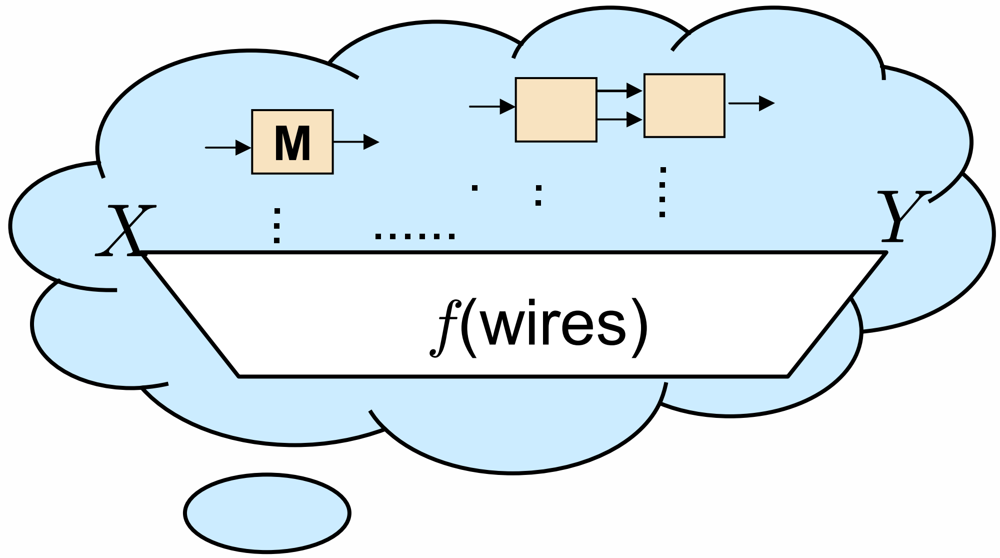
Works well for a single gate... but does not compose.
Exponential security loss (for AC^0).

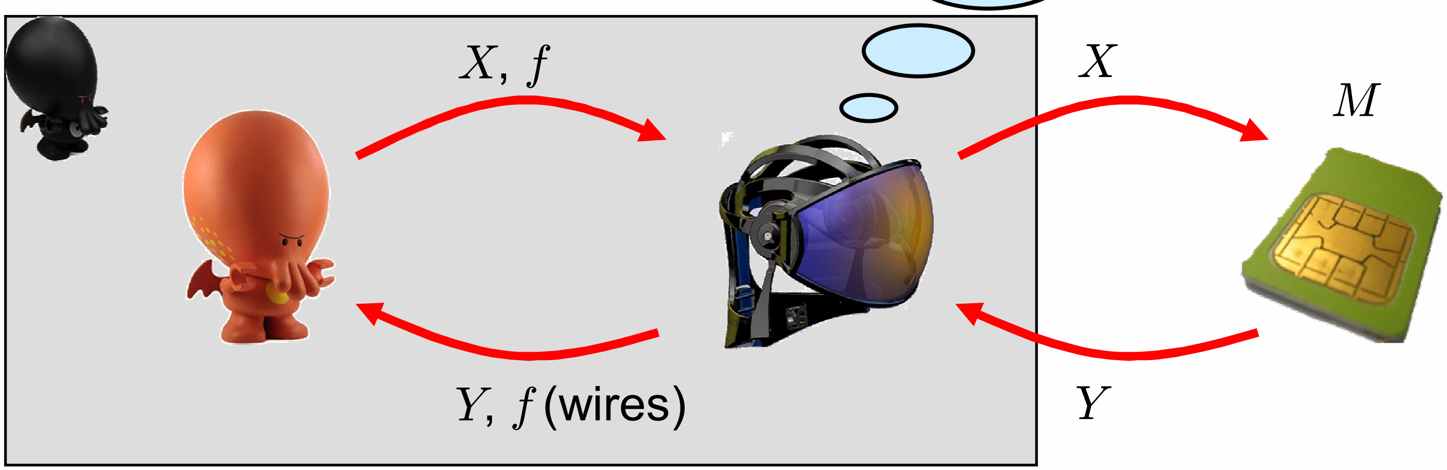
Intuition: wire simulation

Since f can verify arbitrary gates in circuit, wires must be consistent with X and Y.

Problem: simulator does not know state M

Solution: to fool the adversary, introduce **non-verifiable** atomic gate.





Opaque gate

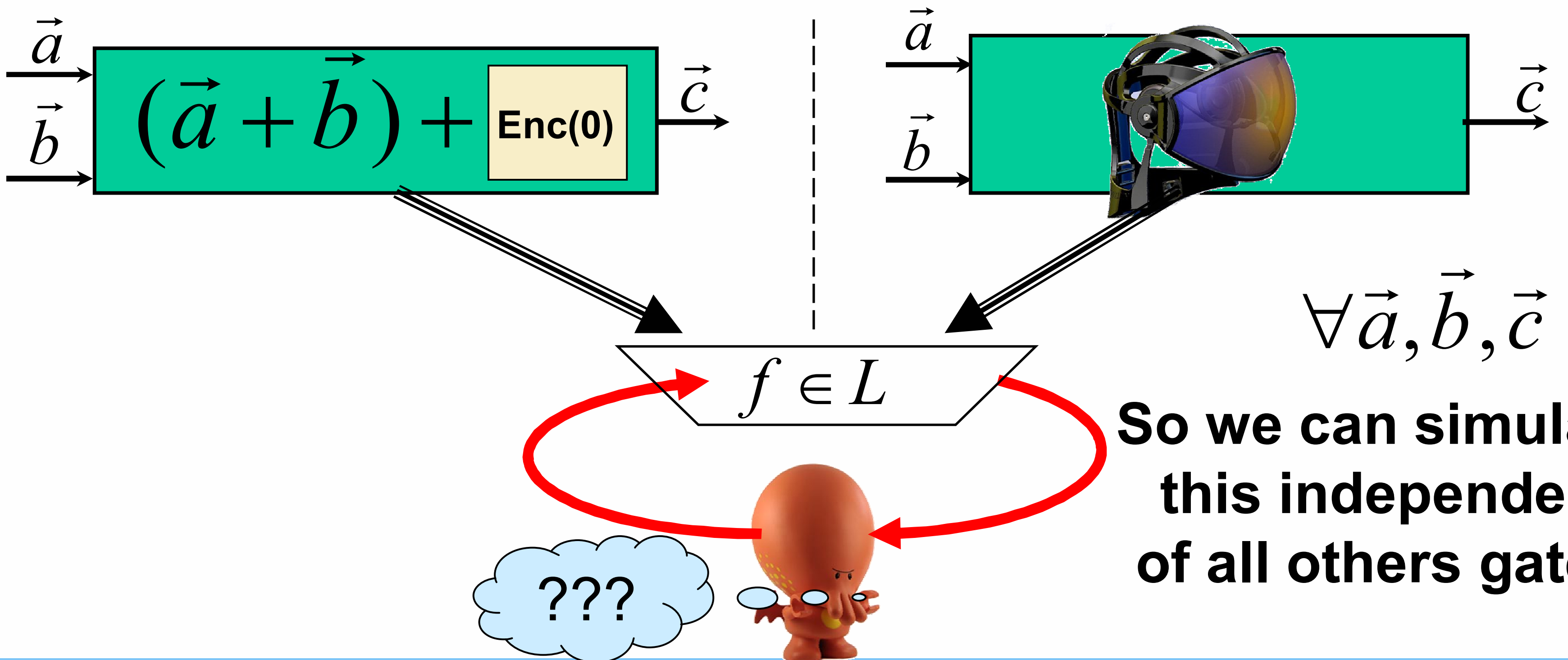
Fool adversary:
gate is non-verifiable by functions in L.

- Samples from a fixed distribution.
- No inputs
- Can be realized by a leak-free "consumable tape"

Using the opaque gate

Full transformation for [+] gate:

Wire's simulator advantage: can change output of opaque without getting noticed (L-leakage-indistinguishable)

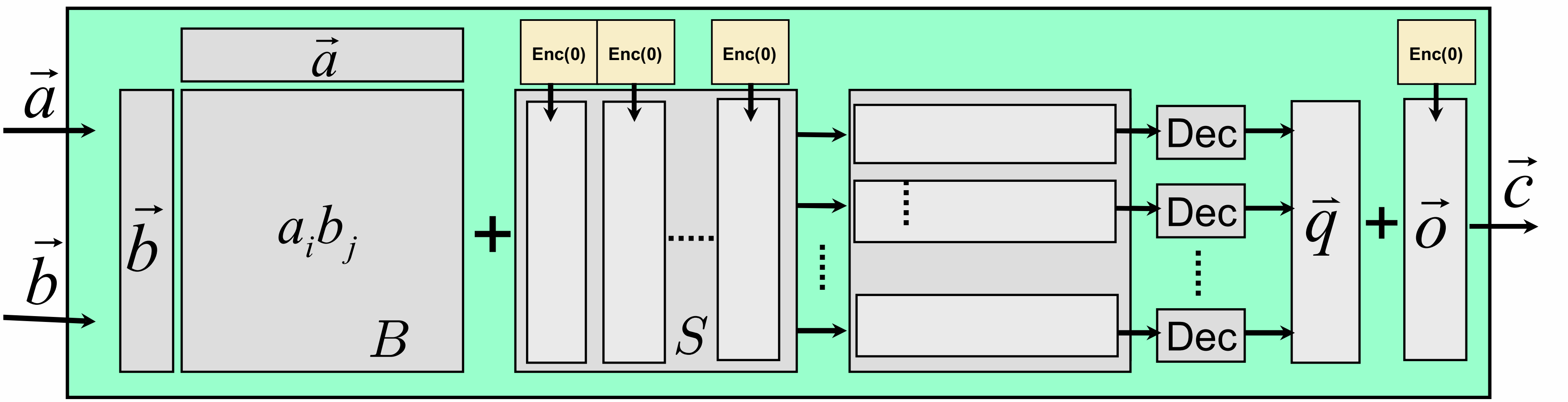


$$\forall \vec{a}, \vec{b}, \vec{c}$$

So we can simulate this independent of all others gates

Other gates

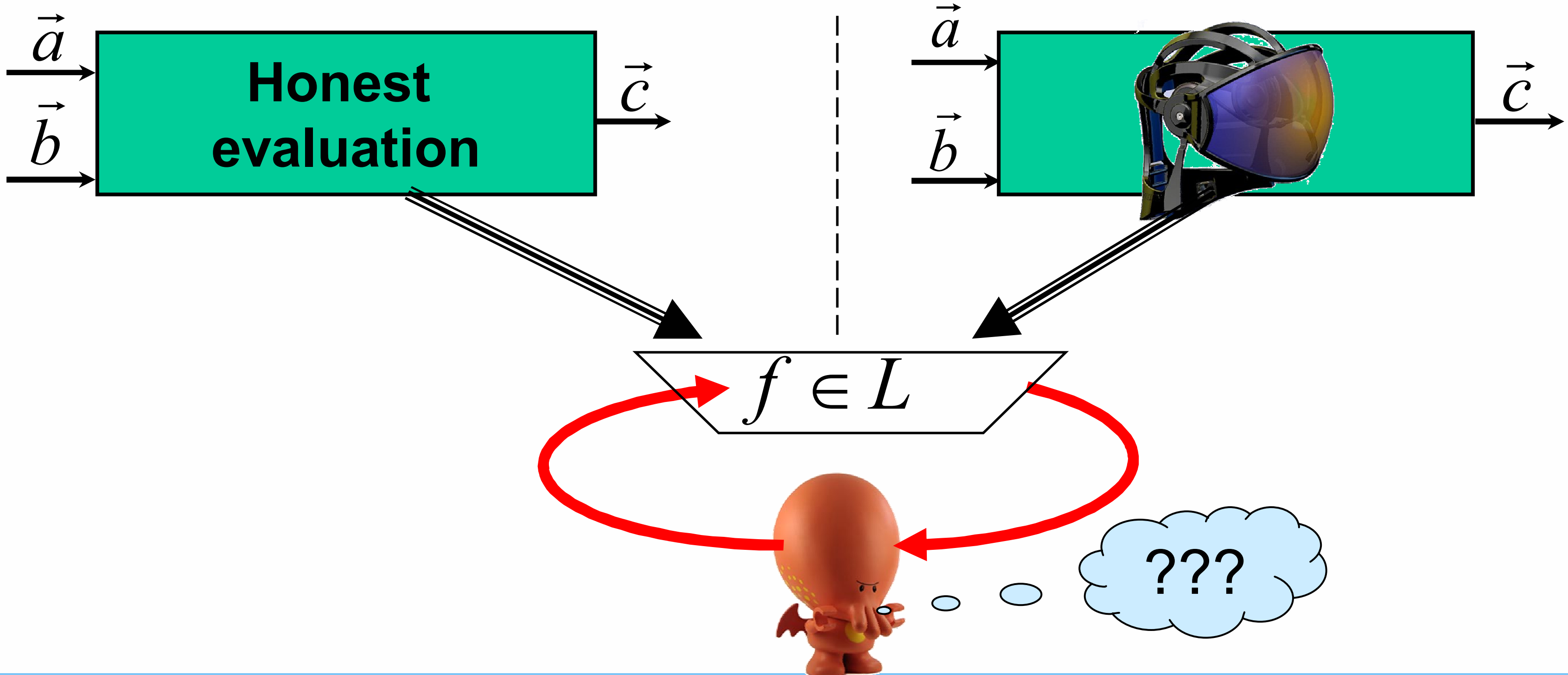
- Similar transformation for other gates.
- The challenging case is the non-linear gate, field **multiplication**. Hard to make leak-resilient; standard MPC doesn't work.
  Trick: give wire simulator enough degrees of freedom.



$$\mathsf{Dec}(\vec{c}) = \vec{r}^{\top}(\vec{q}+\vec{o}) = \vec{r}^{\top}((B+S)\vec{r}+\vec{o}) = \vec{r}^{\top}((\vec{a}\vec{b}^{\top}+S)\vec{r}+\vec{o})$$
$$= (\vec{r}^{\top}\vec{a})(\vec{b}^{\top}\vec{r})+(\vec{r}^{\top}S)\vec{r}+\vec{r}^{\top}\vec{o} = ab+\vec{0}^{\top}\vec{r}+0 = ab$$

Proof technique: wire simulators

All of our gadgets have shallow wire simulators that are L-leakage indistinguishable from honest:



Wire simulator composability

This property (suitably defined) composes!

If every **gadget**
has a (shallow) wire simulator
then the **whole transformed circuit**
has a (shallow) wire simulator.

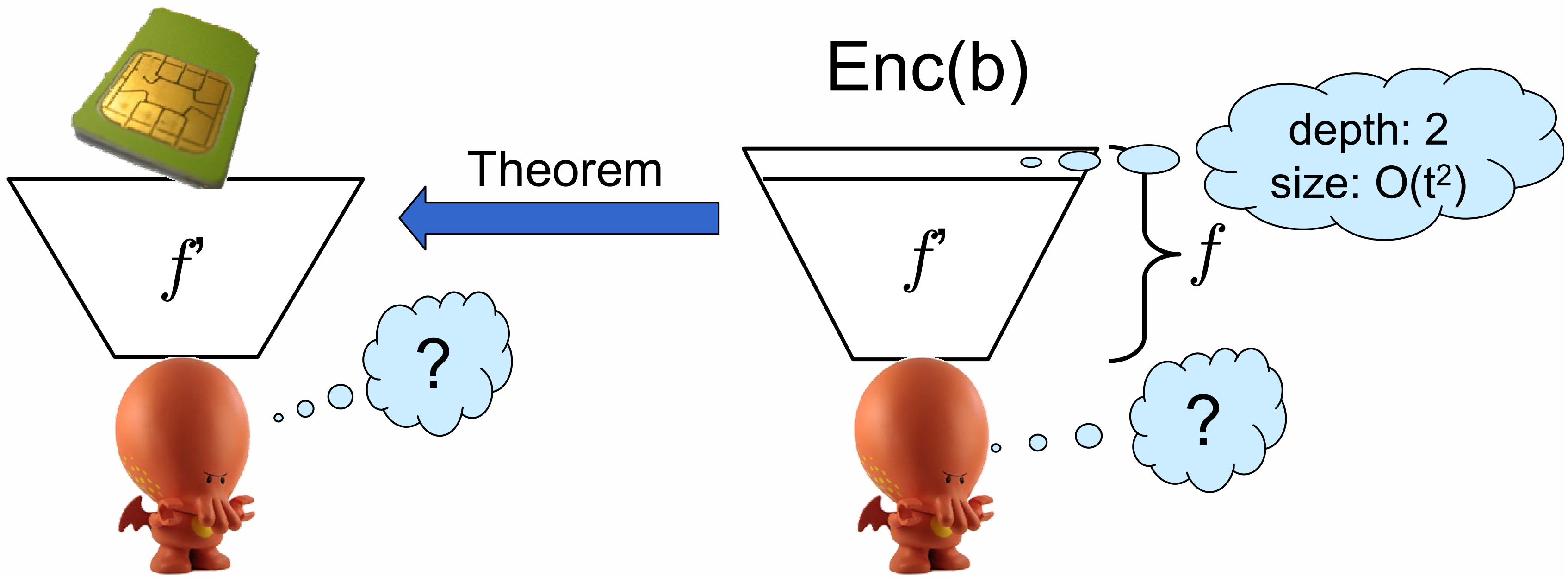
Security for single round follows easily.

For multiple rounds there's extra work due to adaptivity of the leakage and inputs.

CSAIL MIT

Security proof: bottom line

- Loss in the reduction to leakage-indistinguishability of the encoding scheme: very small.
- Necessary since we prove security against low computational classes.
- This makes the computational-security proof very delicate.



Wire simulators redux

General proof technique. Theorem:

If every gadget has (shallow) **wire simulators, then the transformation is** (almost) **as leakage-indistinguishable as the encoding.**

Applications:

- Resilience against polynomial-time leakage using public-key encryption.
  - Assumes leak-free GenKey-Decrypt-Compute-Encrypt components.
  - Proof is extremely easy!
- Resilience against noisy leakage[Rabin Vaikuntanathan 2009]
  - Easy alternative proof.
- *Theorem for hire!*

Wire simulators strike again

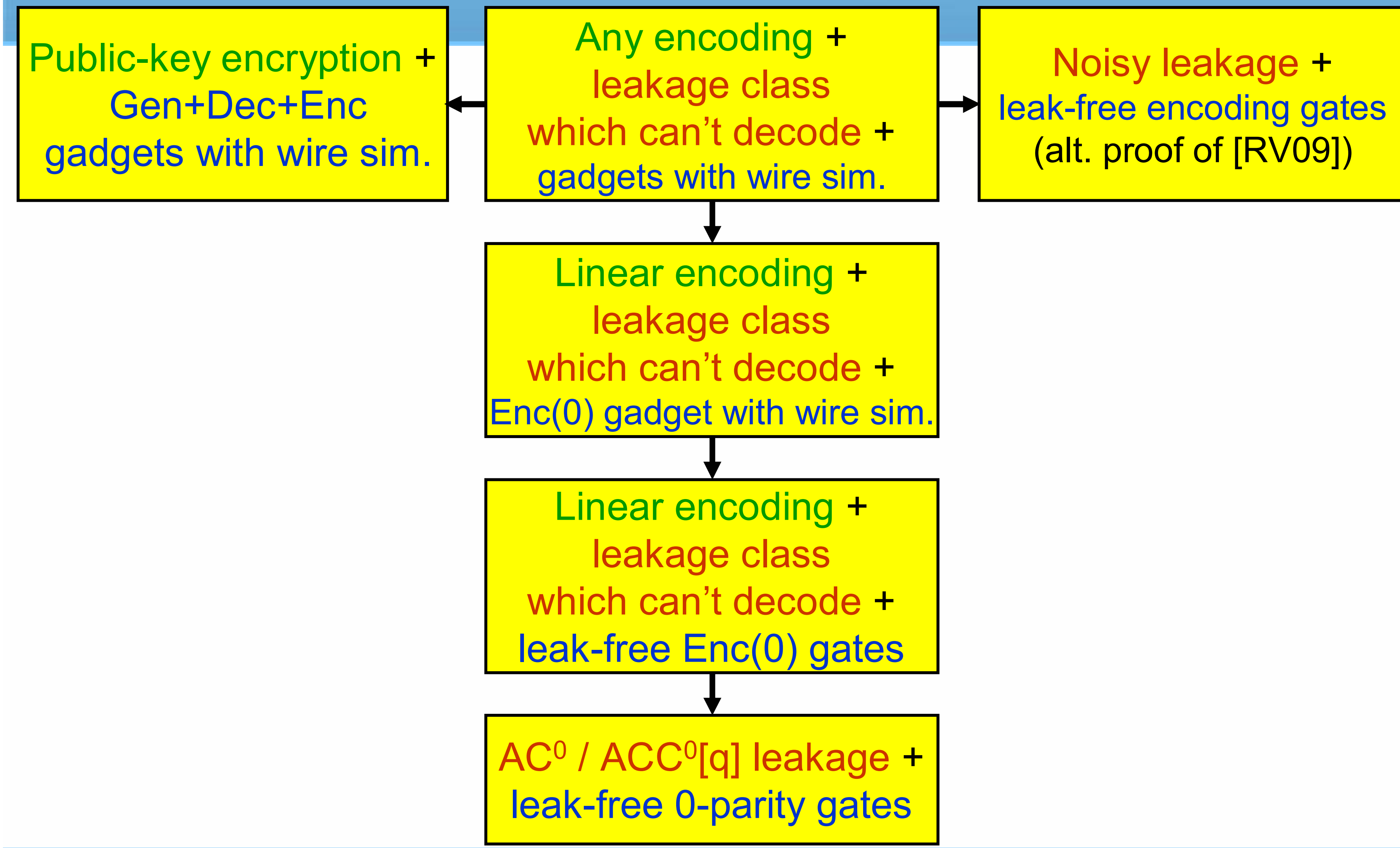
Nested-composition theorem:

Can replace each leak-free gate with a gadget of the same I/O functionality (based on different gates), if the gadget has a wire simulator that is leakage-indistinguishable.

Example: reduce randomness in the AC^0 opaque gate.

- Can be implemented using $\mathrm{polylog}(t)$ randomness + PRG. [Nis91]
- Can be implemented shallowly using any $\mathrm{polylog}(t)$-independent source. [Bra09]

Summary of (positive) results



Necessity of leak-free components

Theorem: any sound transformation that has wire simulators fooling nontrivial leakage classes **requires large leak-free components** (grow with security parameter, which grows with circuit size).

Intuition: otherwise leakage functions $f \in L$ can verify the simulated wire values, and thus force the wire simulator to honestly compute the function.

Then **shallow circuits** (wire simulators) can compute **any function computable by polysize circuits**!

- Impossible if the simulation (and encoding) are constant-depth.
- More generally, implies unlikely complexity-theoretic collapses, e.g, NC=P/poly.

Conjecture: necessity holds for all circuit transformations which are secure against nontrivial leakage via a black-box reduction to the leakage-indistignuishability of encodings.

Conclusions

Achieved

- New model for side-channel leakage, which allows global leakage of unbounded total size
- Constructions for generic circuit transformation, for example, against all leakage in AC^0.
- Partial impossibility results.
- General proof technique + additional applications.

Open problems

- More leakage classes
- Smaller leak-free components
- Proof/falsify black-box necessity conjecture
- Circumvent necessity result (e.g., non-blackbox constructions)

http://eprint.iacr.org/2009/341