ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

IBM

Practical Composable Cryptographic Protocols Resistant Against Adaptive Attacks

Robert R. Enderlein

Examination Committee:

Prof. Dr. Ueli Maurer, ETH Zurich
Dr. Jan Camenisch, IBM Research – Zurich
Prof. Dr. Ralf Küsters, University of Trier

Chair:

Prof. Dr. Marc Pollefeys, ETH Zurich

Introduction

Cryptography is pervasive in digital communication:

E-banking | Online shopping | E-mail | Social media | Search engine | Encyclopedia

- Cryptography is concerned with the design of systems that need to resist malicious attempts to abuse them. [Goldreich]
- Other uses: e-auctions, e-voting, digital cash, distributed computation.
- Before provable security, schemes were regularly broken.
- Even today, security often secondary to UX and costs.

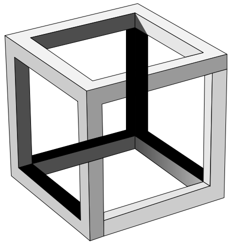
→ Need for protocols that are both **secure** and **practical.**

Provable Security

- Proving large protocols secure is challenging.
- Practical schemes often proven in isolation.
  - Security not guaranteed if run concurrently with itself/others.
- Better guarantees with **composition frameworks.**
  - Secure in arbitrary environments.
  - Modular proofs thanks to composition.
  - Typically slower than protocols proven in isolation.

Goal: Practical Protocols with Strong Security

- **Realistic assumptions.**
  No random oracles. Allow CRS.



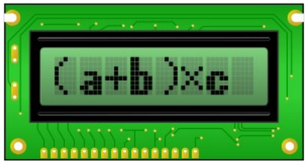
- **Provably secure in arbitrary contexts.**
  Designed in a composition framework.

- **Secure against adaptive adversaries.**
  Real computers can be compromised at any time.

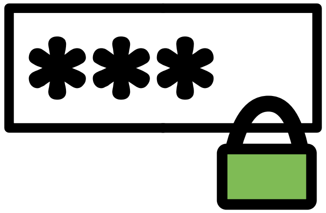
- **Efficient beyond PPT.**
  Avoid cut-and-choose, avoid generic reductions to NP-hard problems, preserve algebraic structure, minimize expensive operations.

Contributions

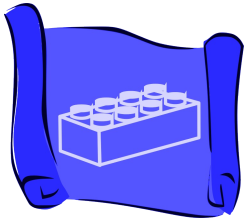
- New protocols:
  - Two-party protocol for arithmetic circuits over $\mathbb{Z}_n$ [C**E**S13]. Best student paper at ESORICS 2013.
    - Parties compute f(input$_A$, input$_B$). Useful sub-protocol.



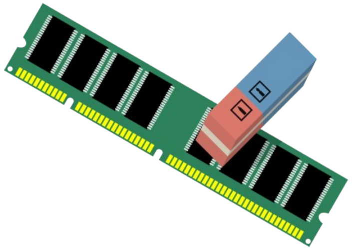
  - Two-server password-authenticated secret sharing [C**E**N15]. Published: PKC 2015.
    - Store & retrieve key with weak password. No brute-force attack against password if 1 server corrupt.

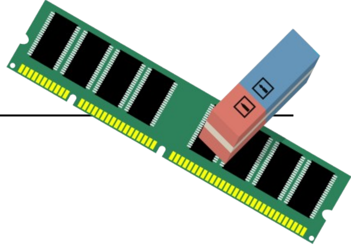


- Improve frameworks & modelling of protocols:
  - Conventions for complete and unambiguous protocol specifications [C**E**KKR16].
    - Framework to specify protocols concisely but precisely.



  - Memory erasability amplification [C**E**M16].

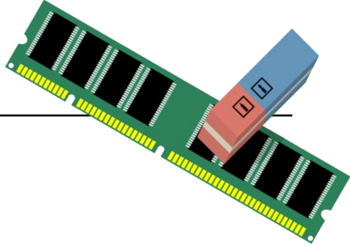




Memory Erasability Amplification

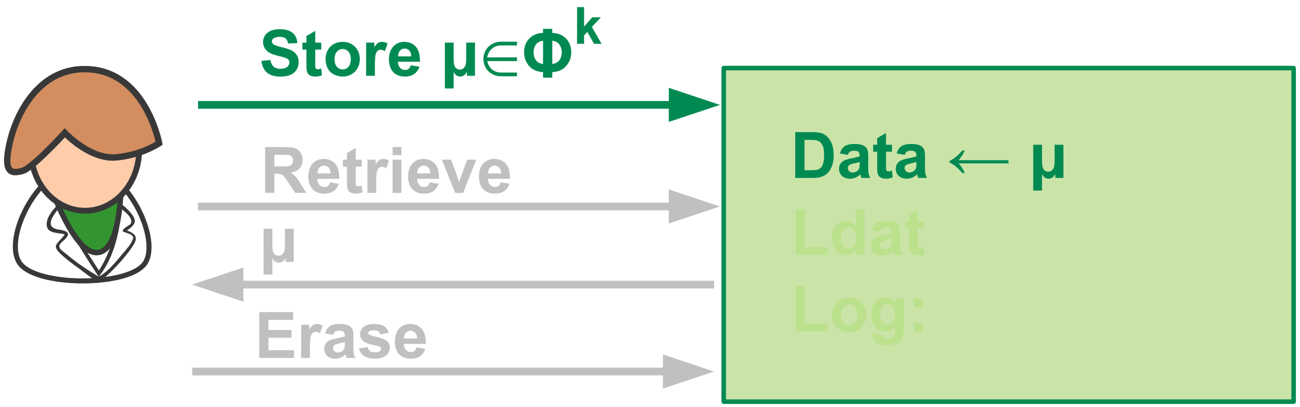
- Erasable memory crucial for most practical adaptively secure protocols.
- Not always available in reality:
  - Computers designed to preserve data, not erase it.
  - File systems don't erase deleted files; keep traces in journal.
  - SSD's don't flash blocks containing overwritten data right away.
- Important to model imperfectly erasable memory.
  - Attempt by [CEGL08, Lim08], but needed to change framework.
- Re-use existing protocols by constructing perfect memory from imperfect one.

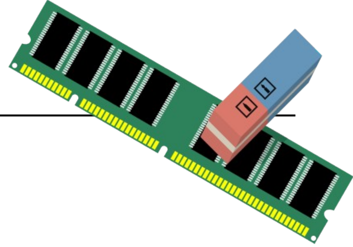
[CEGL08]: Canetti, Eiger, Goldwasser, Lim.
How to Protect Yourself without Perfect Shredding. *ICALP 2008.*
[Lim08]: Lim. *The Paradigm of Partial Erasures*. PhD thesis, MIT, 2008.



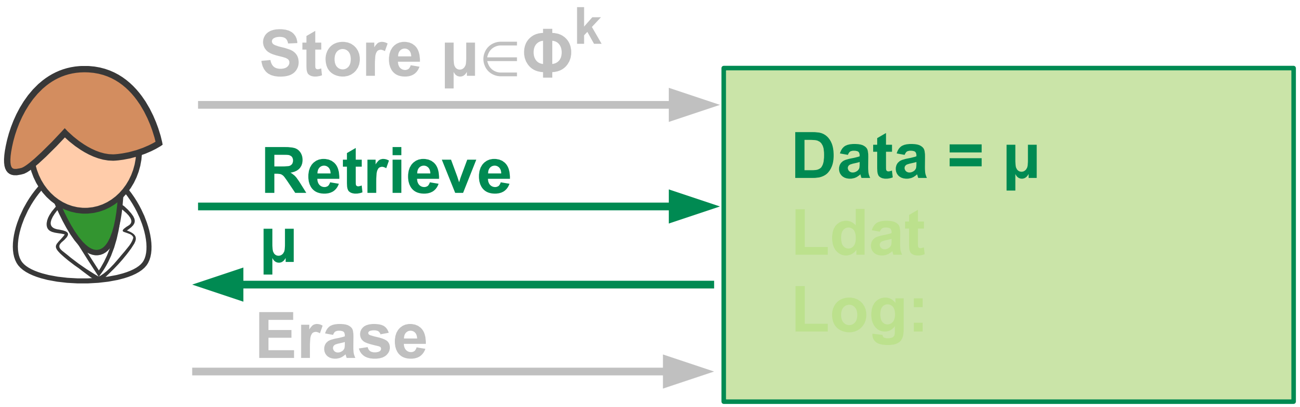
Modeling Erasable Memory

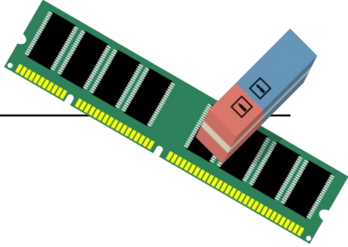
- Memory can be written once.
  - If multiple writes: use multiple resources.





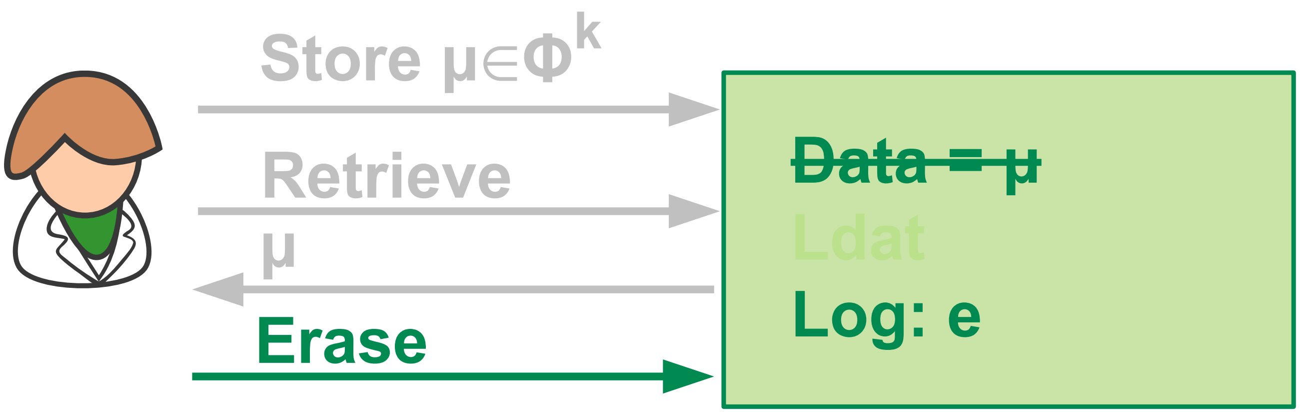
Modeling Erasable Memory



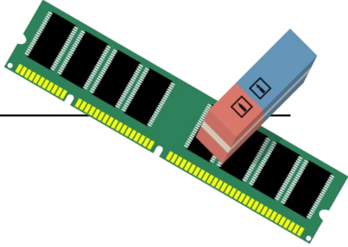


Modeling Erasable Memory

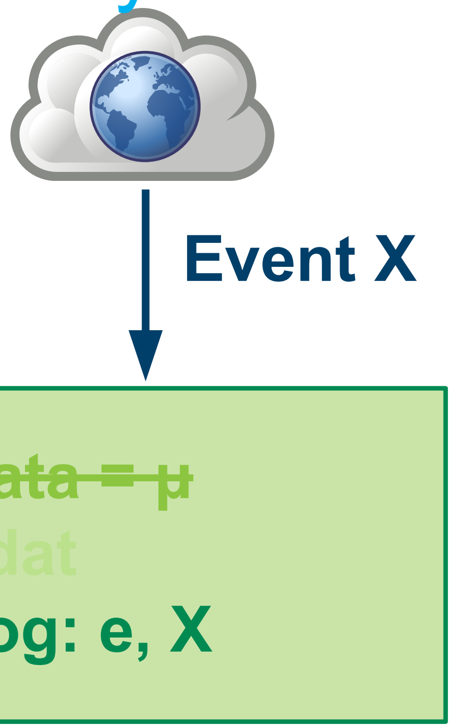
- Entire memory is erased.
  - –For more granularity: use multiple resources.



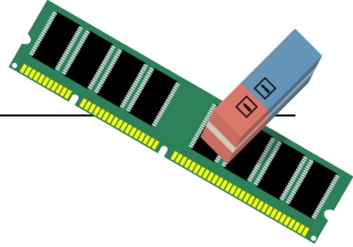
- Erasure event is logged.



Modeling Erasable Memory

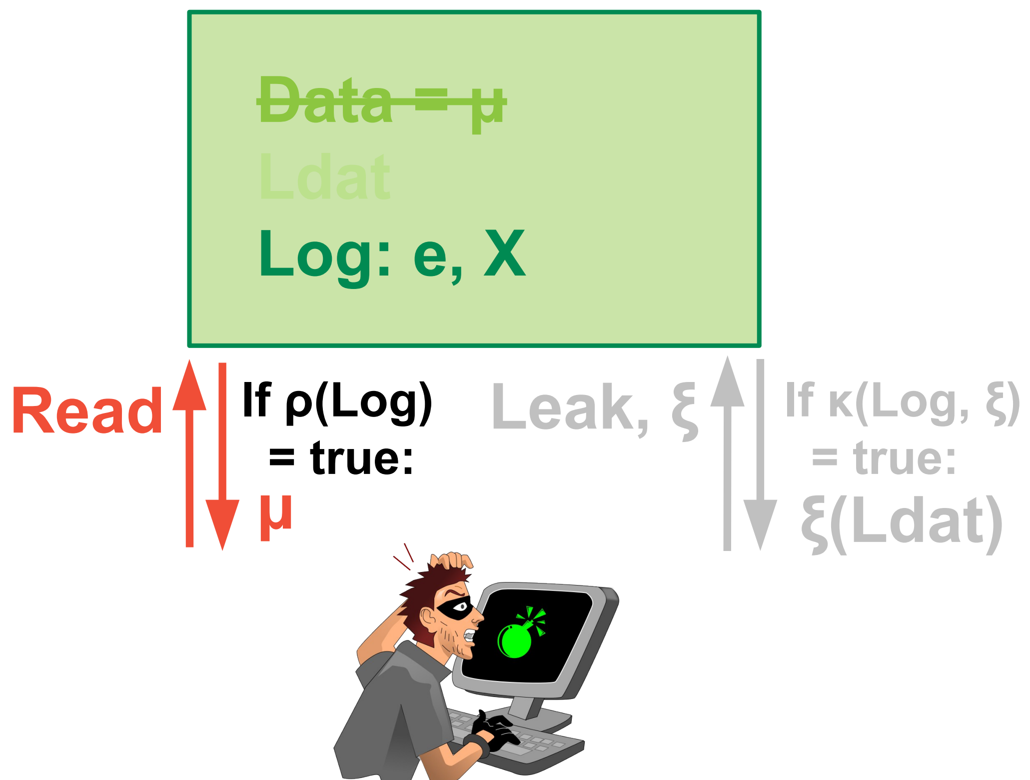


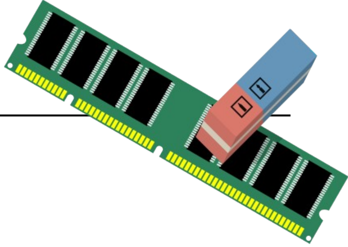
- Environment can influence resource through events.
  - Malware, adversary gets physical access, or even environmental conditions.
  - Events not triggered by the adversary: otherwise no checks & balances.
- Security guarantees of resource depends on those events.
- Events are logged.



Modeling Erasable Memory

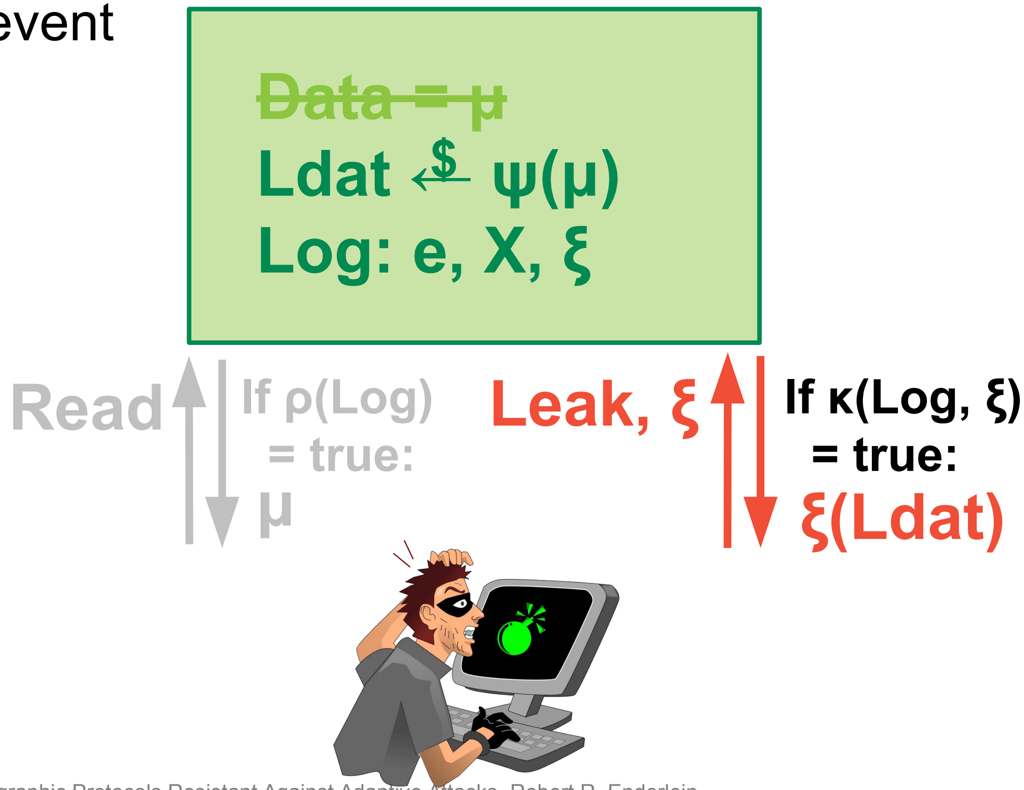
- Adversarial access: none, total (Read), or partial (Leak).
- Total access if predicate **ρ** on event log is true.
  - Typically: “critical” event before/without erasure.

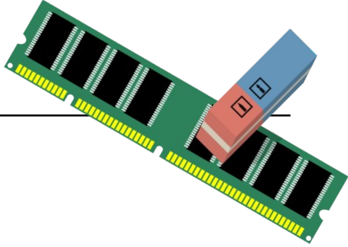




Modeling Erasable Memory

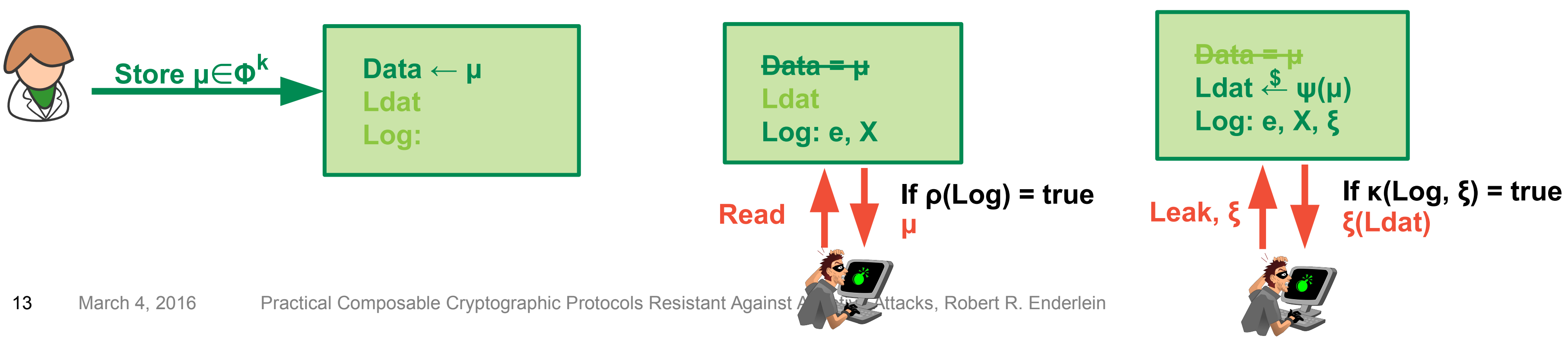
- Adversary might influence result: deterministic function ξ.
- Potential leakage **Ldat** dependent on random function **ψ**.
- Gets **ξ(Ldat)=ξ(ψ(μ))** if predicate **κ** on event log & **ξ** is true.
  - Typically: “critical” event after erasure and **ξ** is OK.
- Adaptive queries.

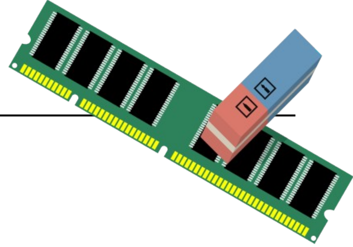




Types of Erasable Memory

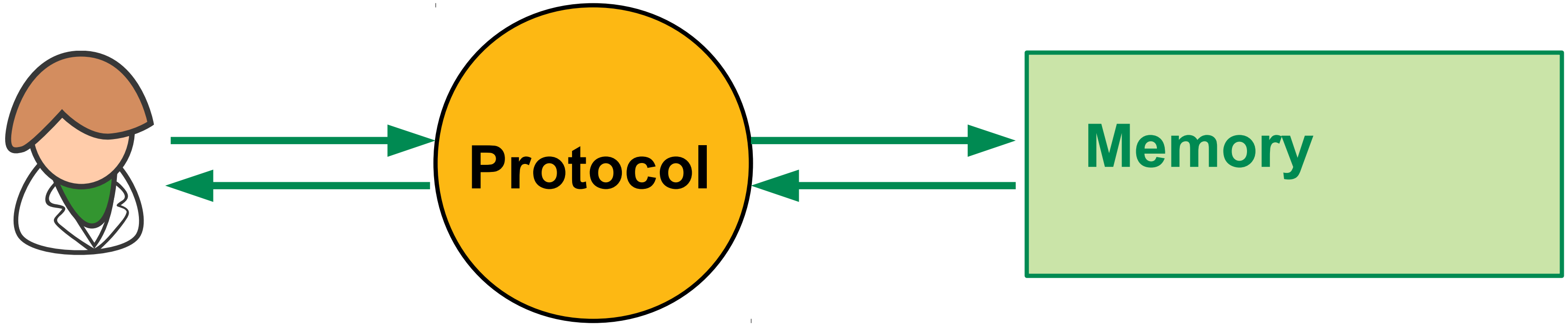
- Typical types of memory are just specializations:
  - –Perfectly erasable memory.
    - **ρ** is true if memory was attacked before/without erase.
    - **κ** returns false.
  - –Imperfectly erasable memory:
    - ♦ Memory leaking a constant number of bits.
      - **ρ** idem.
      - **ψ**(**μ**)=**μ**.
      - **κ** is true if **Log**=(**e**, **X**) and **ξ** reads **d** bits of **Ldat** (and thus of **μ**).
    - ♦ Memory leaking a noisy version of the data.
  - –Non-erasable memory.

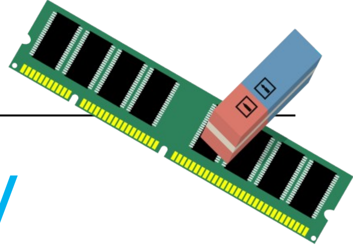




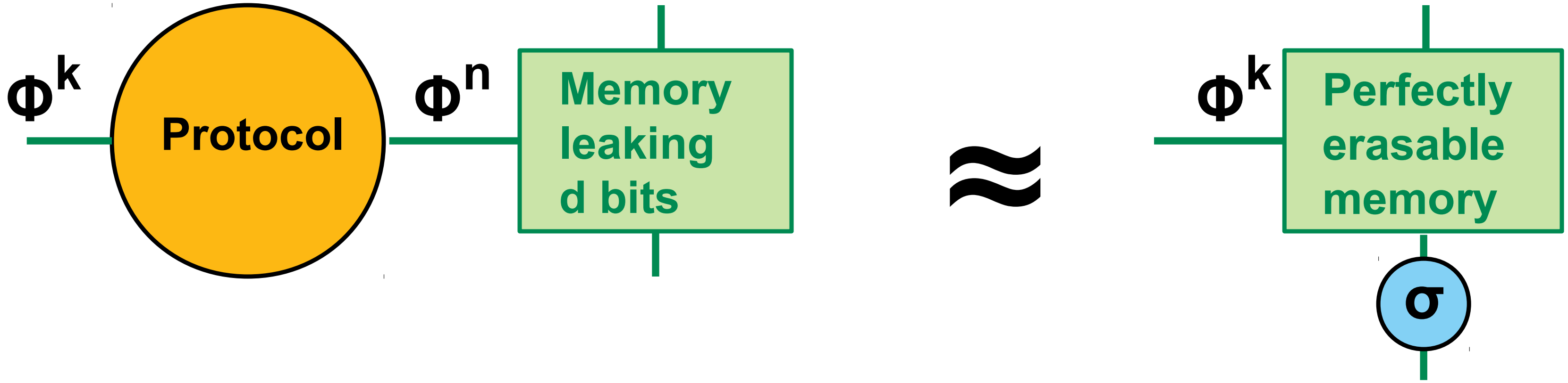
Building Protocols using our Memory

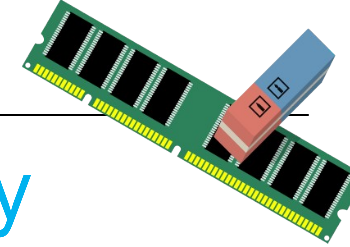
- Goal: protocols that work with imperfectly erasable memory.
- Protocols must not circumvent the memory resource:
  - –Maintain no internal state between computation phases.
  - –But can use temporary storage (registers) during phase (to avoid strong dependency on actual implementation).



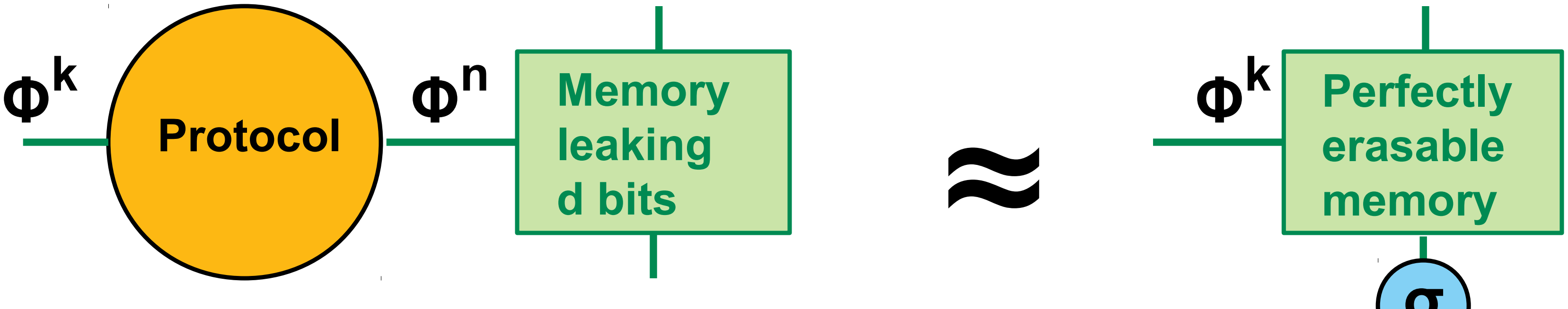


Constructing Perfectly Erasable Memory

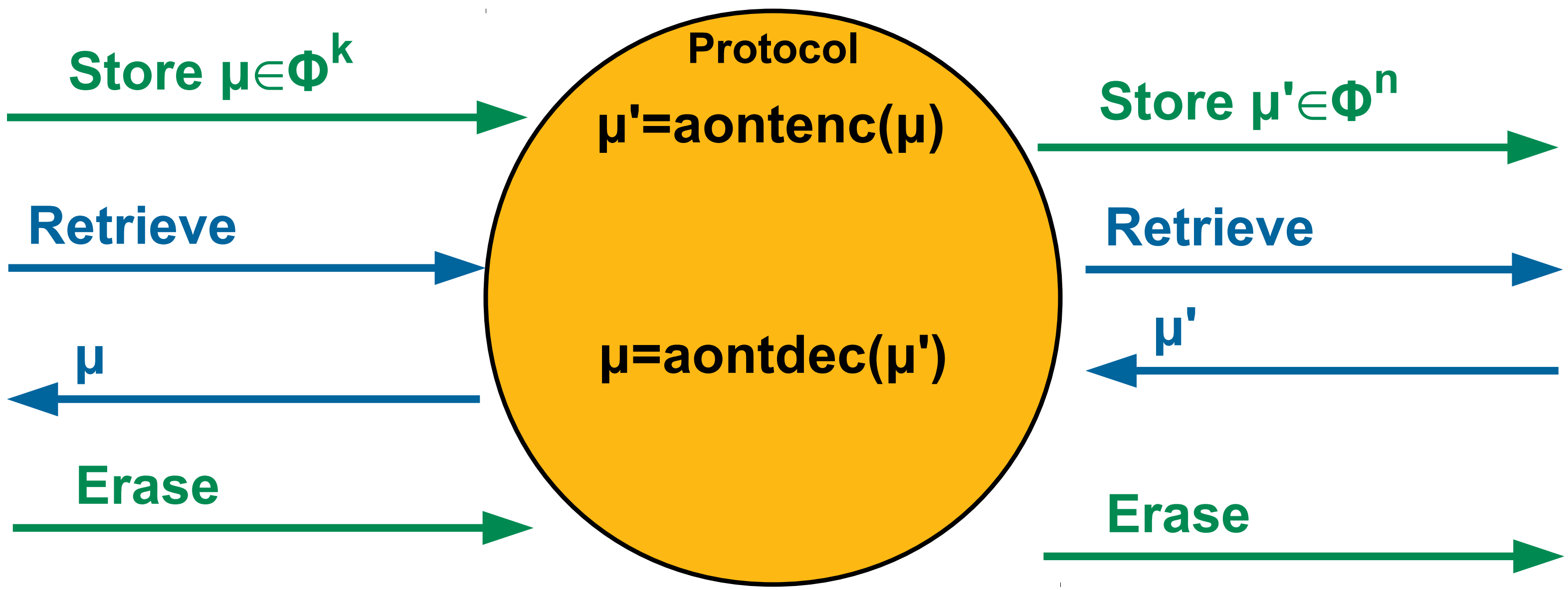




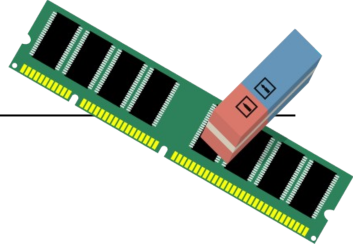
Constructing Perfectly Erasable Memory



- The protocol uses an AoNT [CDH+00].

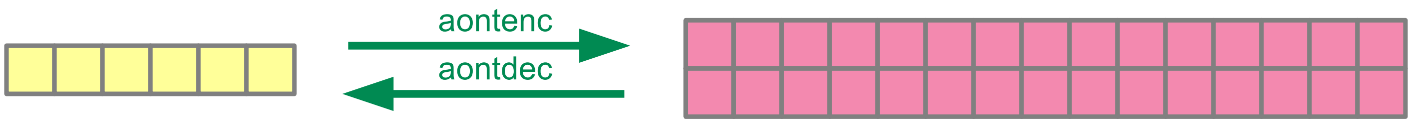


[CDH+00]: Canetti, Dodis, Halevi, Kushilevitz, Sahai. Exposure-Resilient Functions and All-or-Nothing Transforms. *Eurocrypt 2000*.



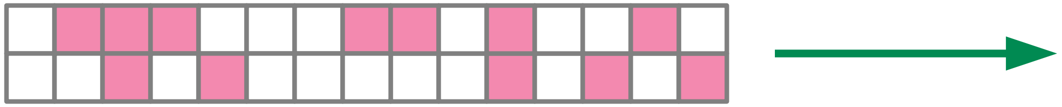
All-or-Nothing Transform [CDH+00]

- Completeness:
  - $\forall \boldsymbol{\mu} \in \boldsymbol{\Phi}^{\mathbf{k}}$: $\boldsymbol{\mu}$ = aontdec(aontenc($\boldsymbol{\mu}$)).



- Privacy:
  - For all sets **L** of size **d**, $\boldsymbol{\mu}_0 \in \boldsymbol{\Phi}^{\mathbf{k}}$, $\boldsymbol{\mu}_1 \in \boldsymbol{\Phi}^{\mathbf{k}}$:

    ($\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_1$, [aontenc($\boldsymbol{\mu}_0$)]$_L$) ≈ ($\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_1$, [aontenc($\boldsymbol{\mu}_1$)]$_L$).

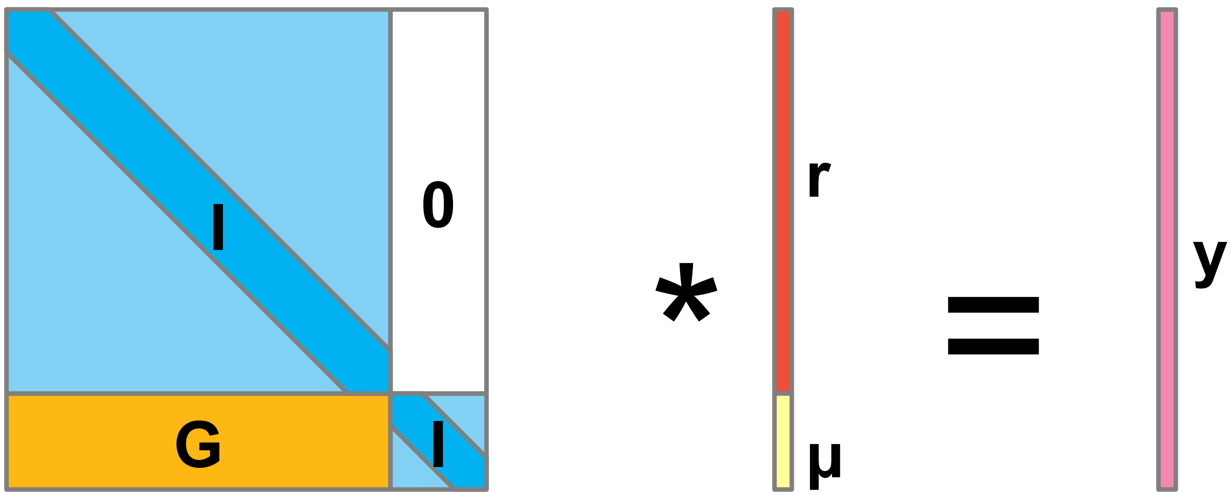


[CDH+00]: Canetti, Dodis, Halevi, Kushilevitz, Sahai. Exposure-Resilient Functions and All-or-Nothing Transforms. *Eurocrypt 2000*.

Examples of AoNT

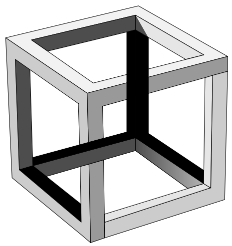
- (Ramp) secret sharing scheme:
  - –Based on Shamir secret sharing (only for large **Φ**). [BM84]
  - –For **Φ**={0, 1}, construction using linear block code. [CDH+00]

Generator matrix **G** of minimum distance **d**.



[BM84]: Blakley, Meadows. Security of Ramp Schemes. *Crypto 1984*.

Conclusion



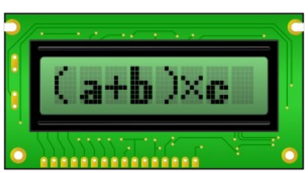
Realistic assumptions

Provably secure in arbitrary contexts

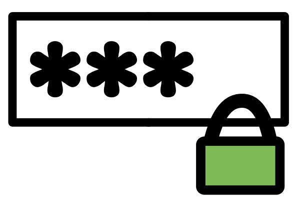
Secure against adaptive adversaries

Efficient beyond PPT

New protocols:

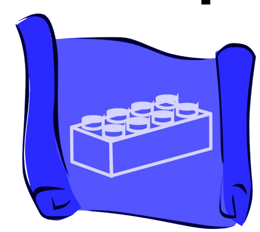


2-party protocol for arithmetic circuits

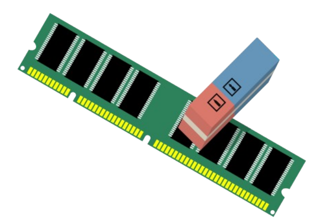


2-server password-authenticated secret sharing

Improving modelling:



Conventions for complete and unambiguous protocol specifications



Memory erasability amplification