CCCP: Secure Remote Storage for Computational RFIDs

M. Salajegheh, **S. Clark**, B. Ransford, K. Fu
(UMass Amherst)

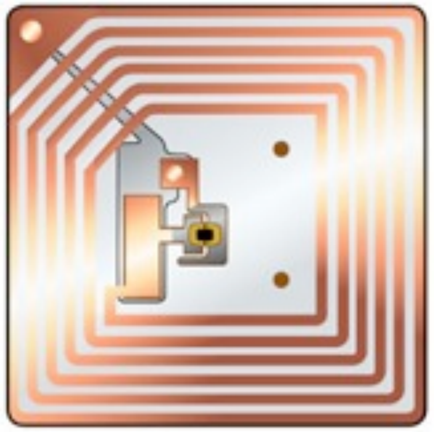
A. Juels
(RSA)

The Security Division of EMC

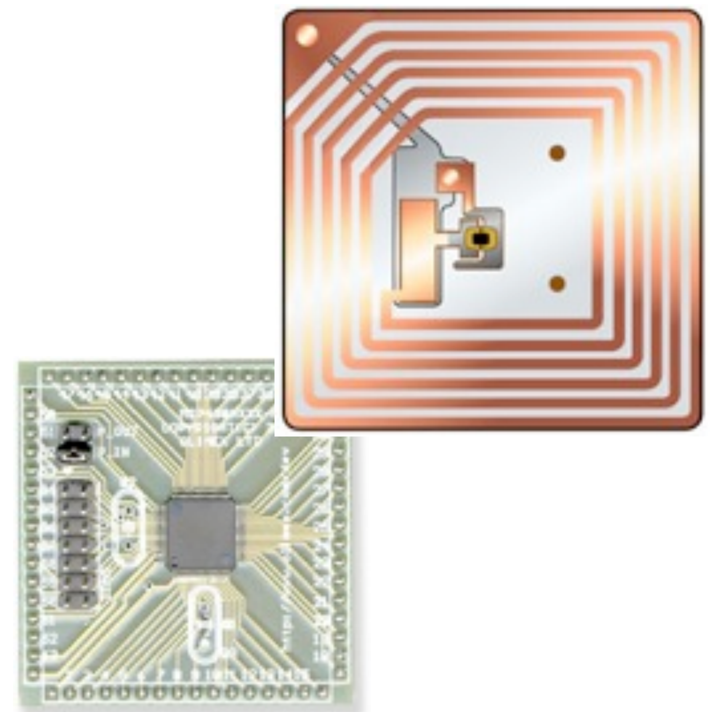
NSF-627529

RFID tags

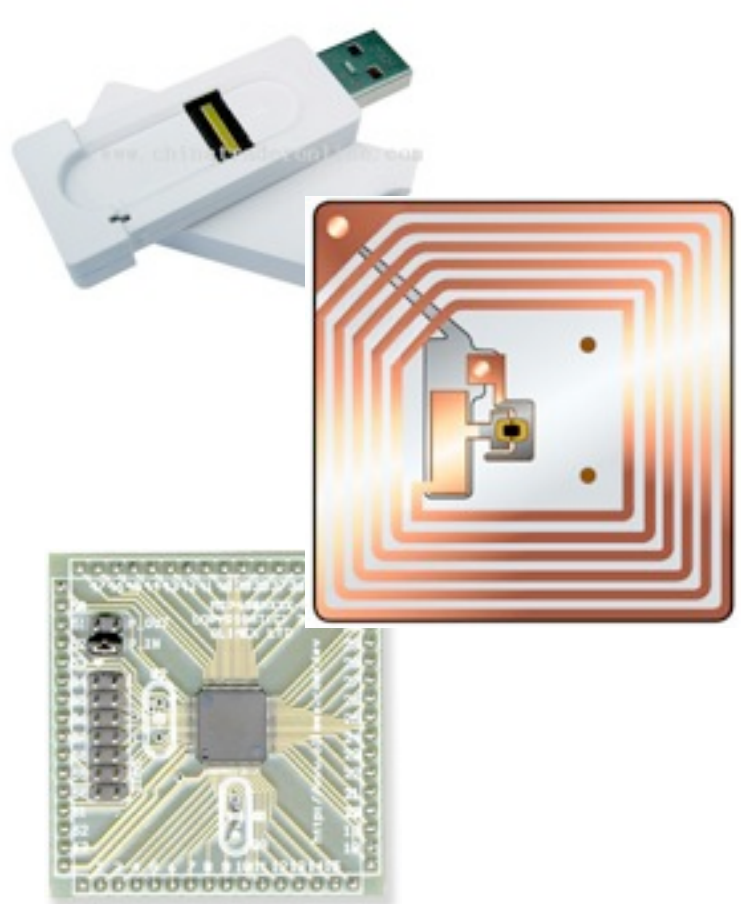
RFID tags



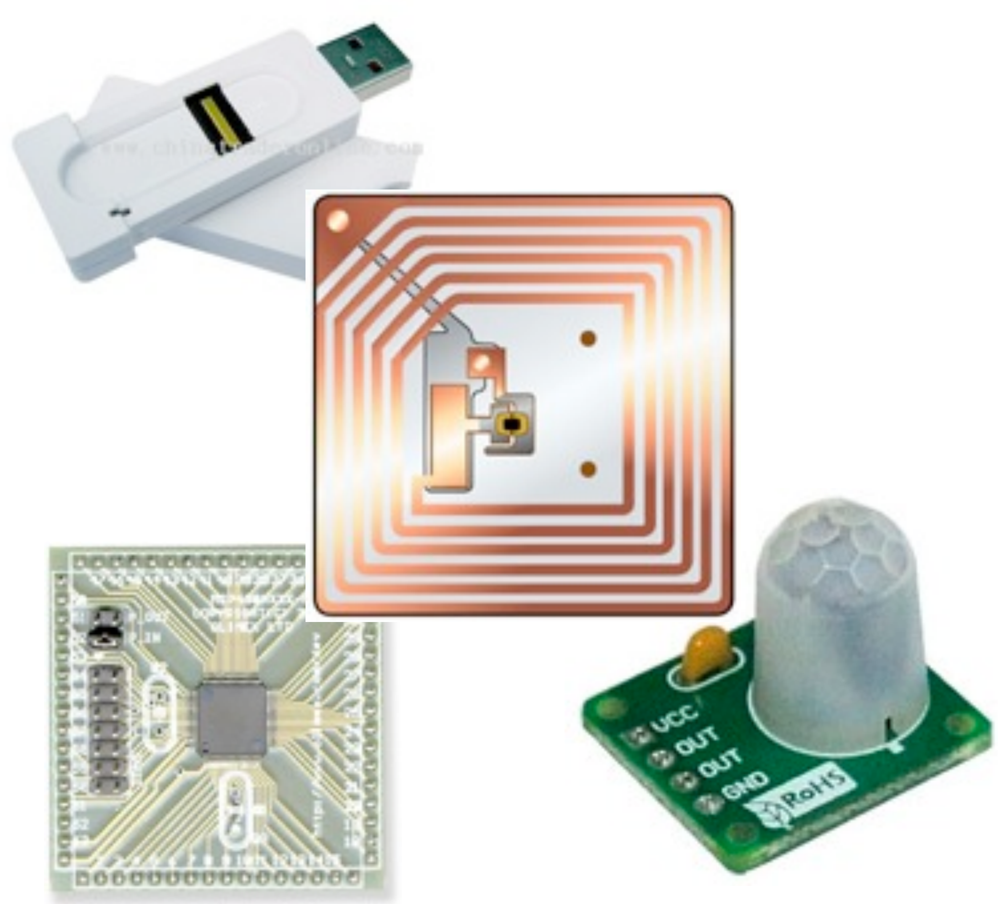
RFID tags



RFID tags

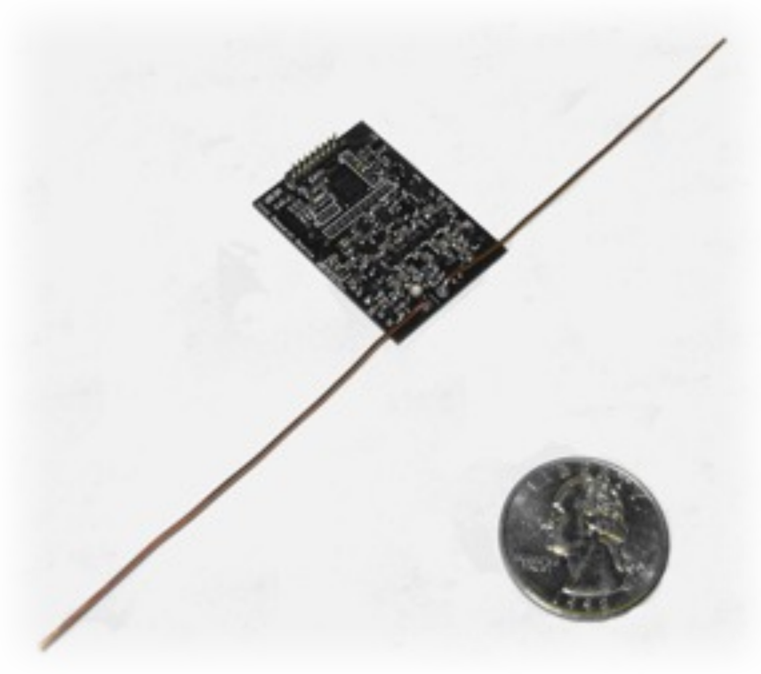


Computational RFID tags



CRFIDs

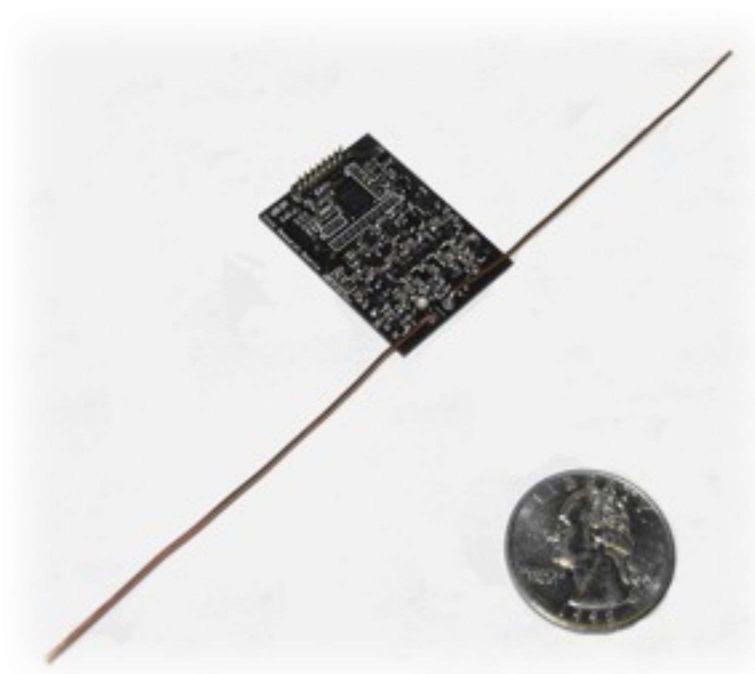
- Batteryless
- Powered by harvested energy
- Interact with RFID readers
- Programmable



WISP 1.0

CRFIDs

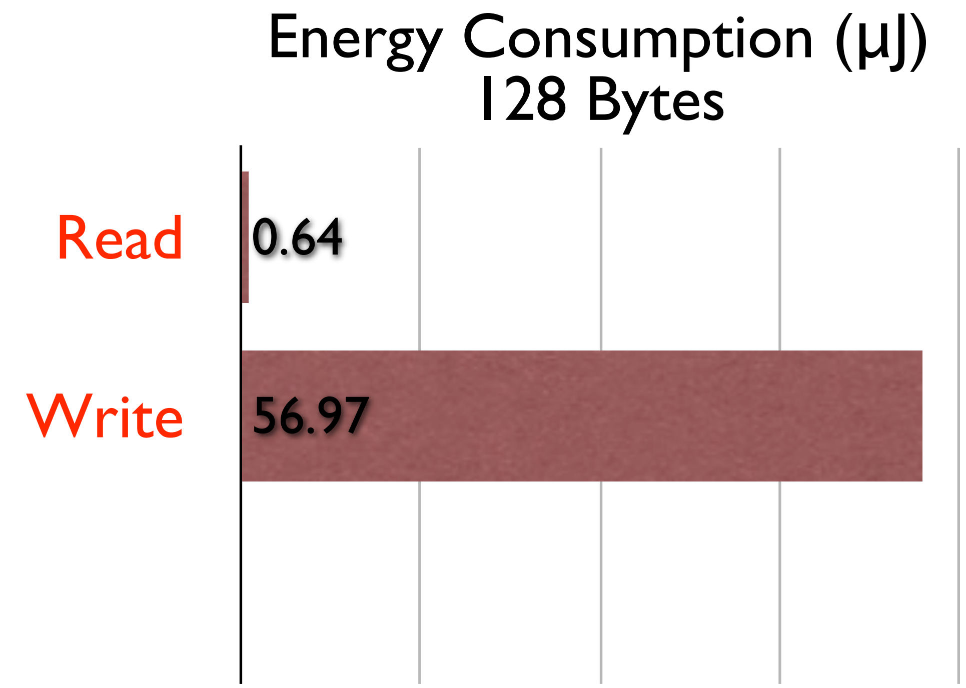
- Tiny energy reservoir
- Frequent power loss
- Limited use of local storage



WISP 1.0

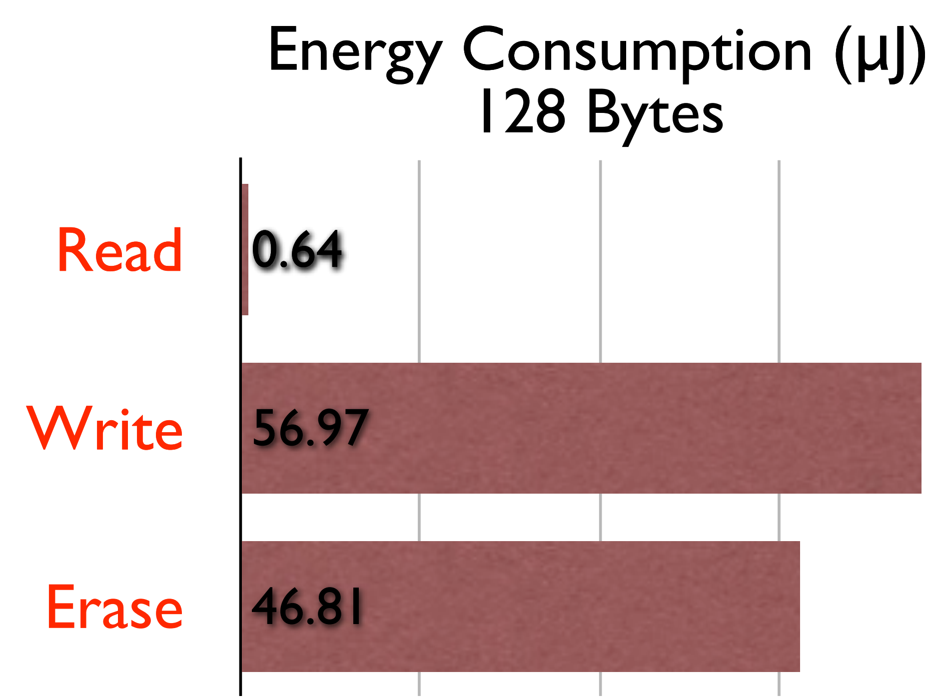
Local Storage... at a Price

- Energy intensive writes



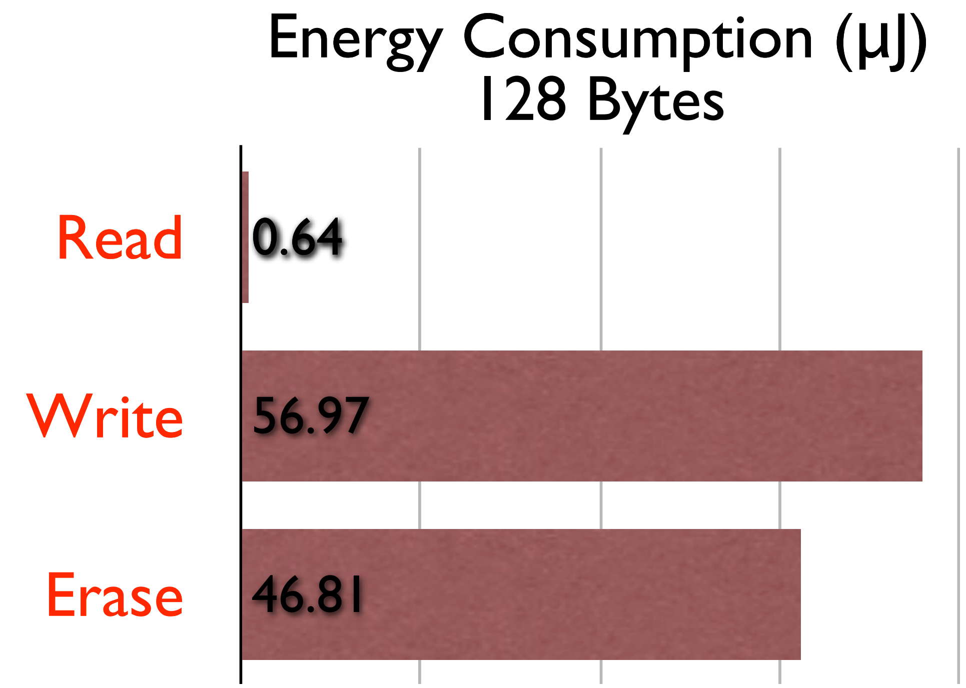
Local Storage... at a Price

- Energy intensive writes
- Erase-before-write

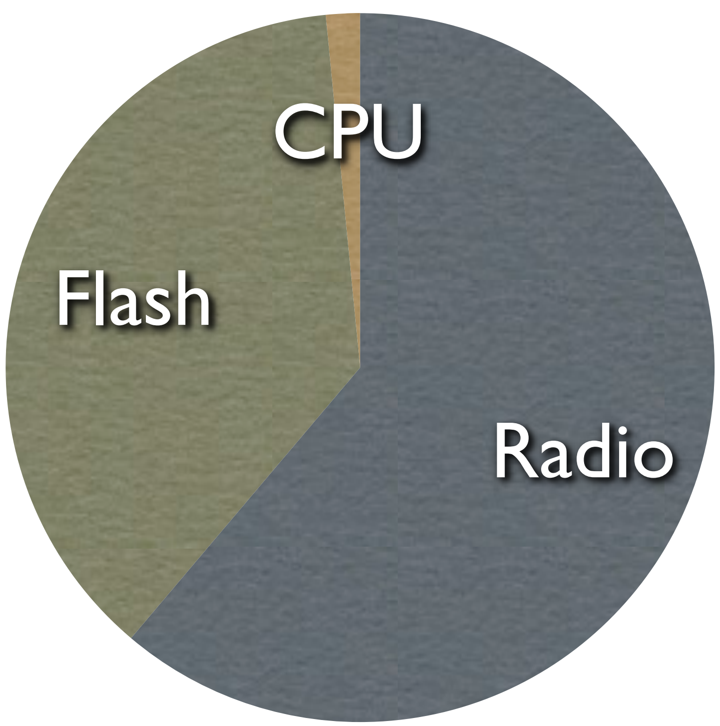


Local Storage... at a Price

- Energy intensive writes
- Erase-before-write
- Small nonvolatile memory
  - WISP 4.0: 32 KB flash

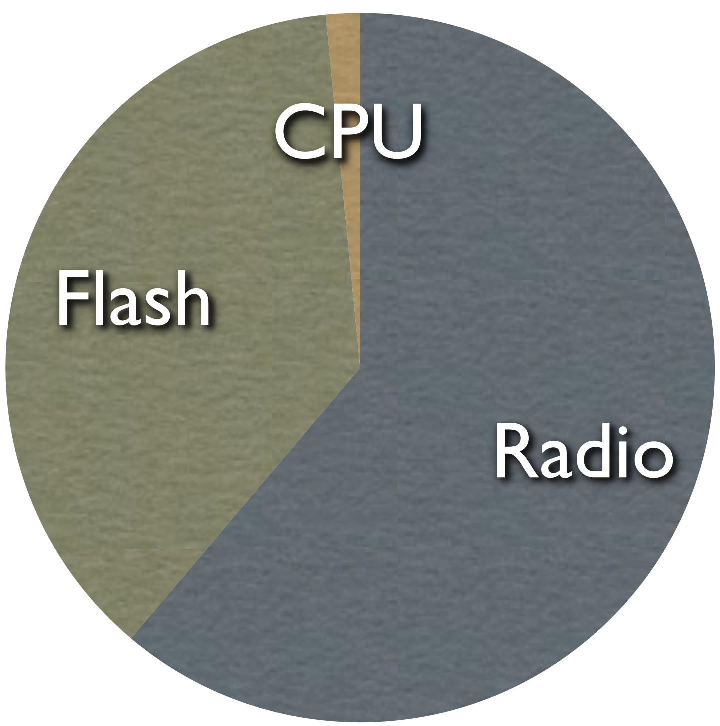


Inexpensive Radio

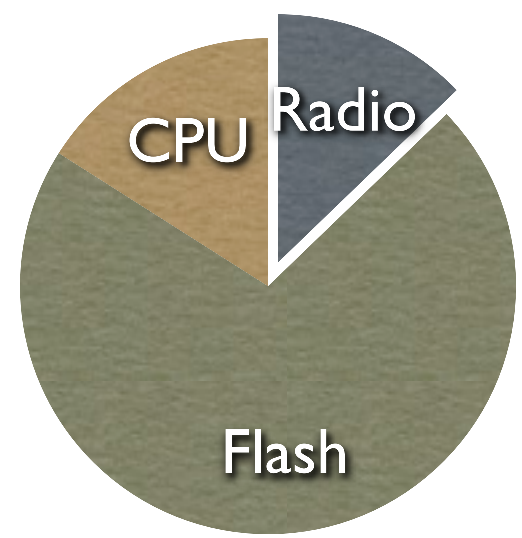


Mote
[Hydrowatch]

Inexpensive Radio

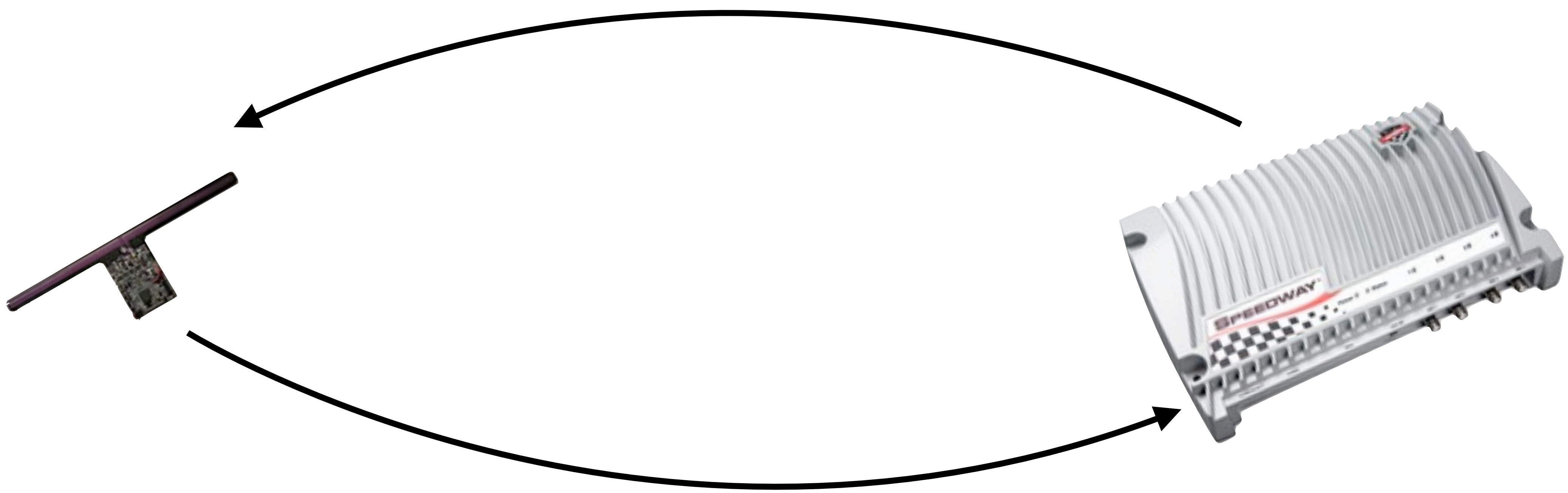


Mote
[Hydrowatch]

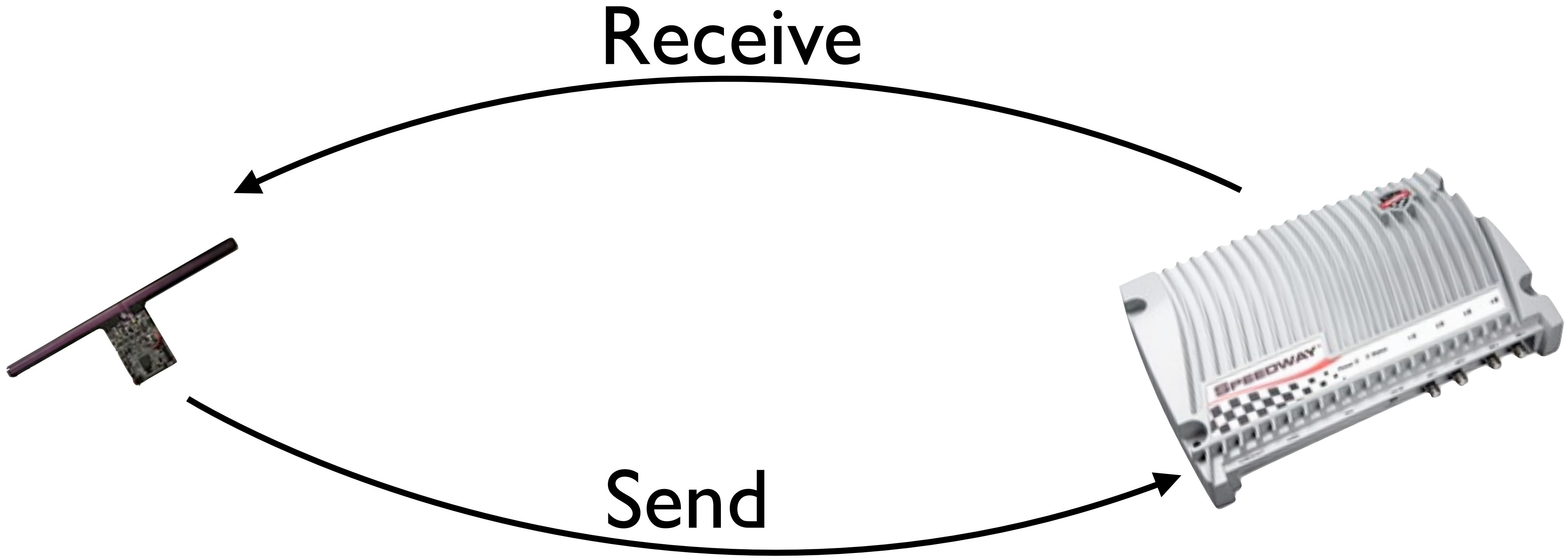


CRFID

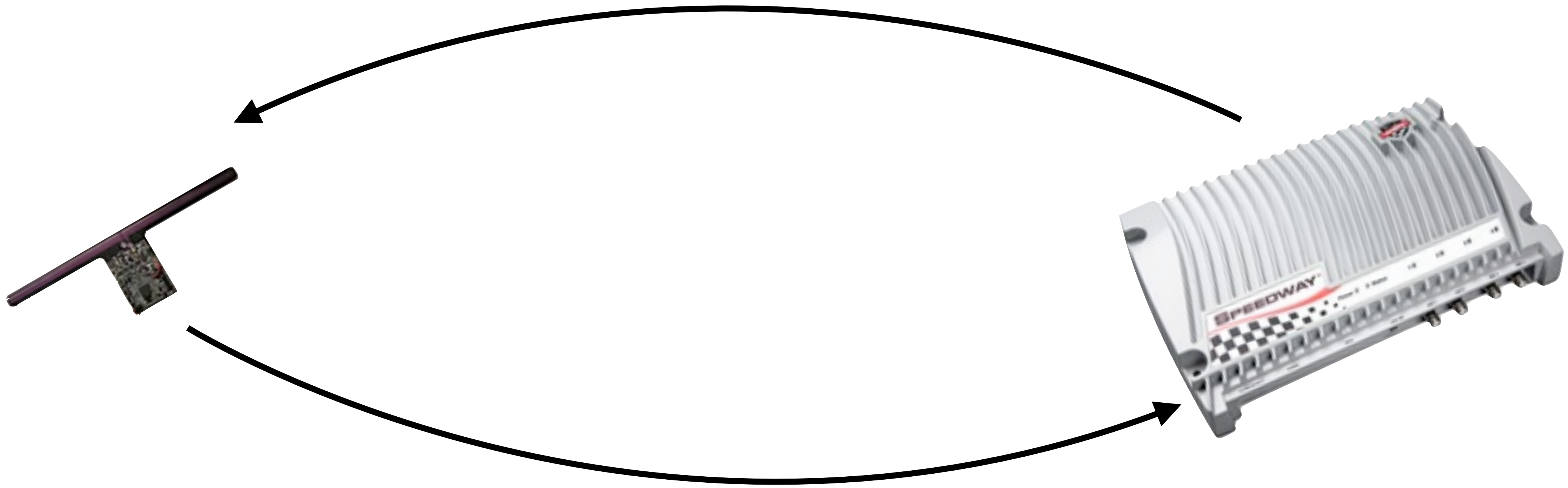
Outsource Storage?



Outsource Storage?

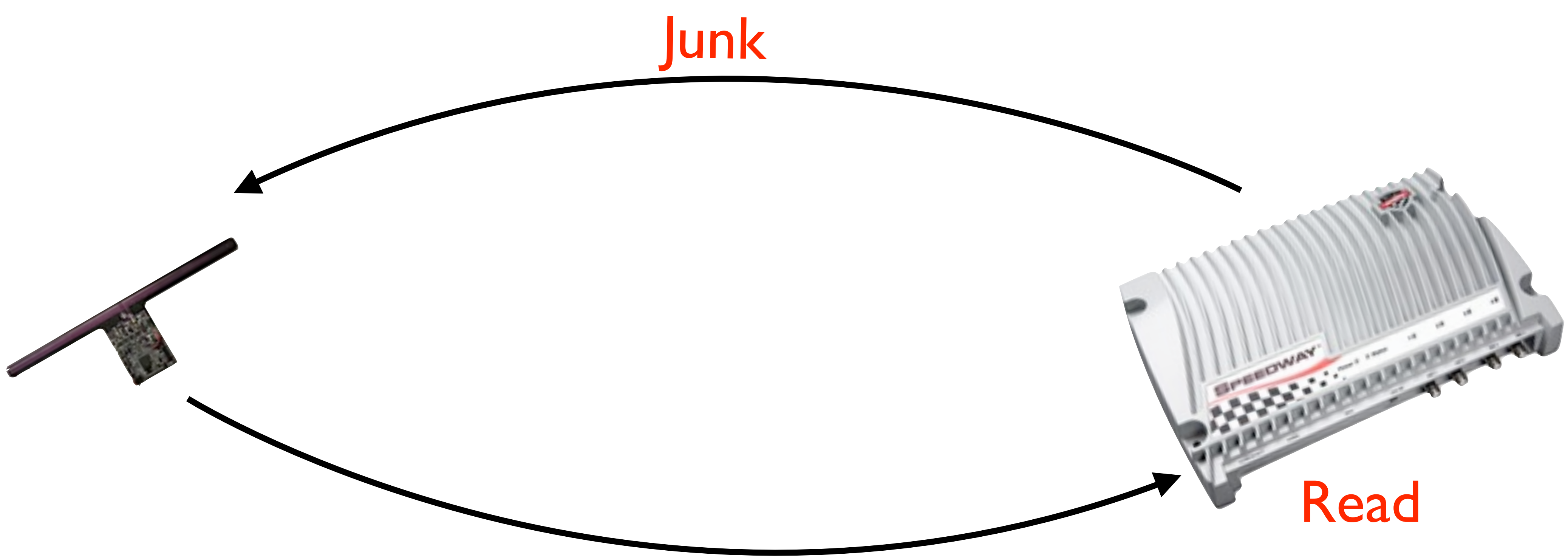


Outsource Storage?



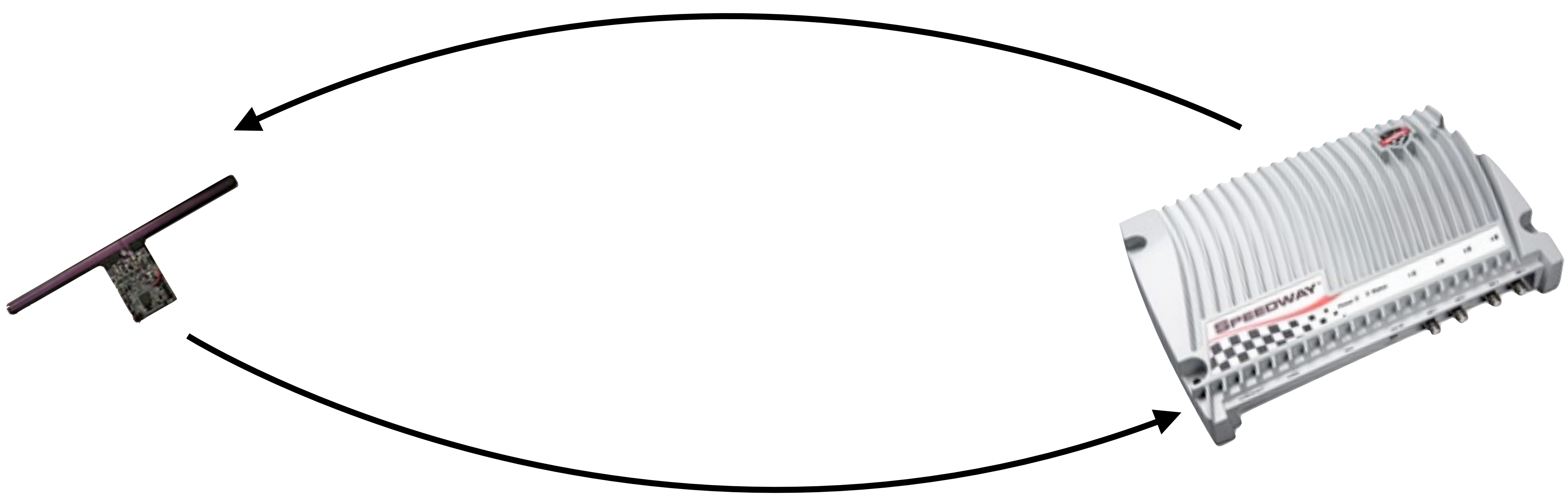
Problem: a reader is not necessarily trustworthy

Outsource Storage?

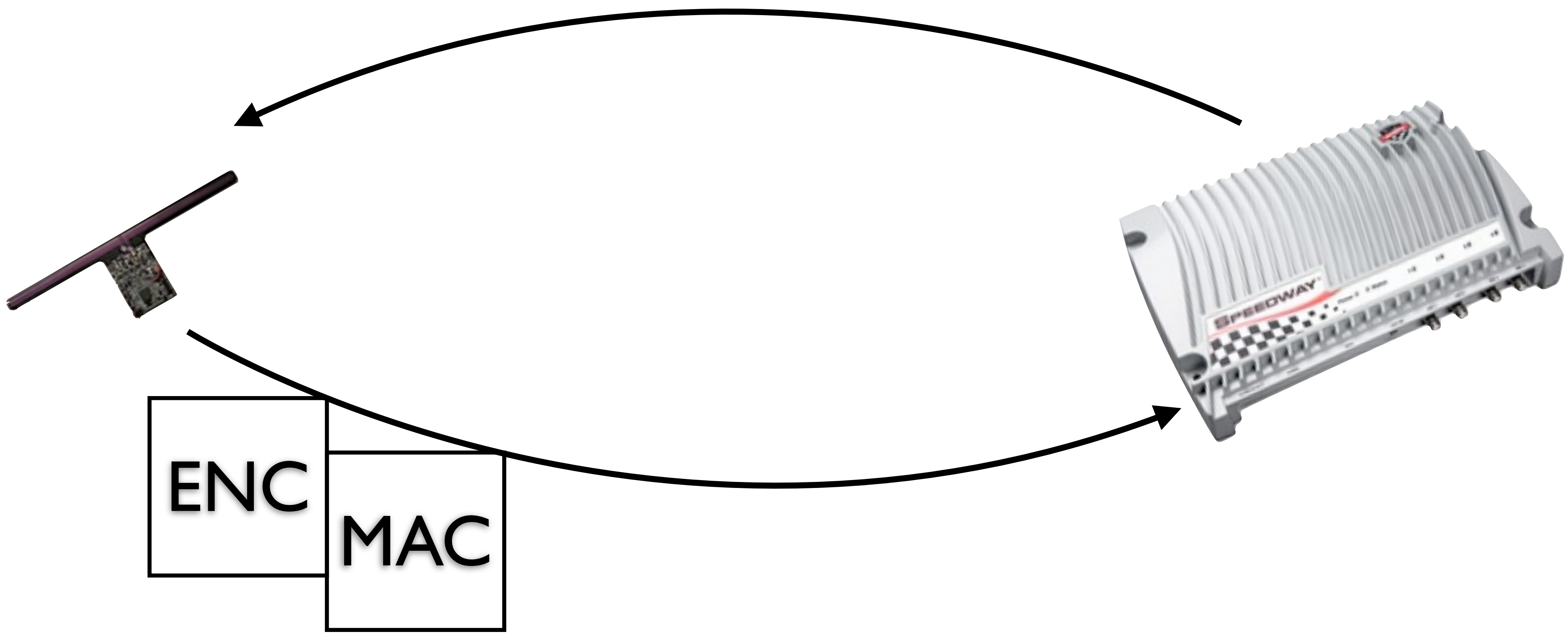


Problem: a reader is not necessarily trustworthy

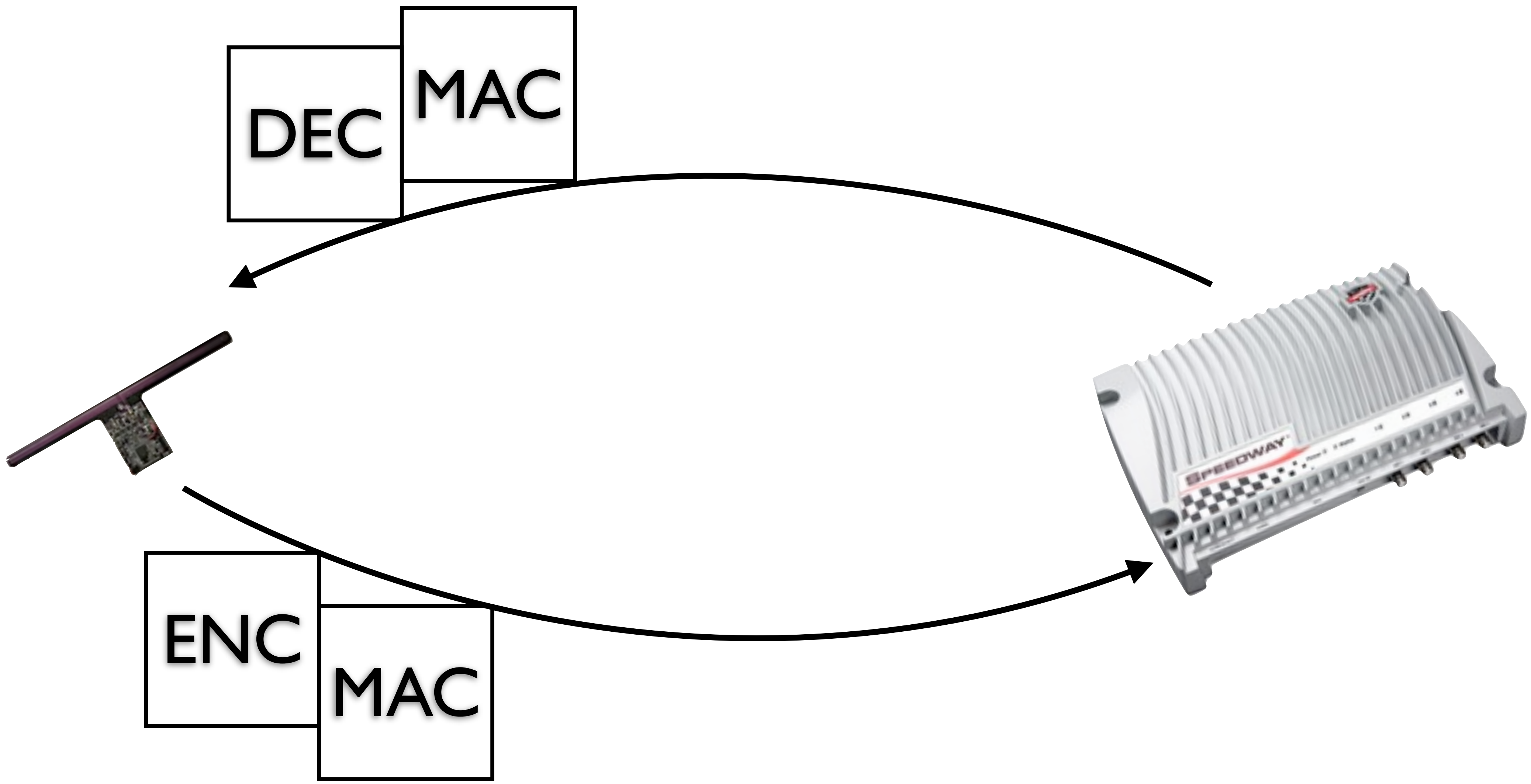
Cryptographic Computational Continuation Passing



Cryptographic Computational Continuation Passing



Cryptographic Computational Continuation Passing



Goal: Computational Progress

- Change of computational state toward a goal
  - Example: completion of a loop
- Eliminate Sisyphean tasks

Sisyphean Tasks

- Some workloads may never complete given typical energy availability

- Manually splitting tasks is not necessarily easy or effective

Mementos [Ransford '08]

Mementos [Ransford '08]

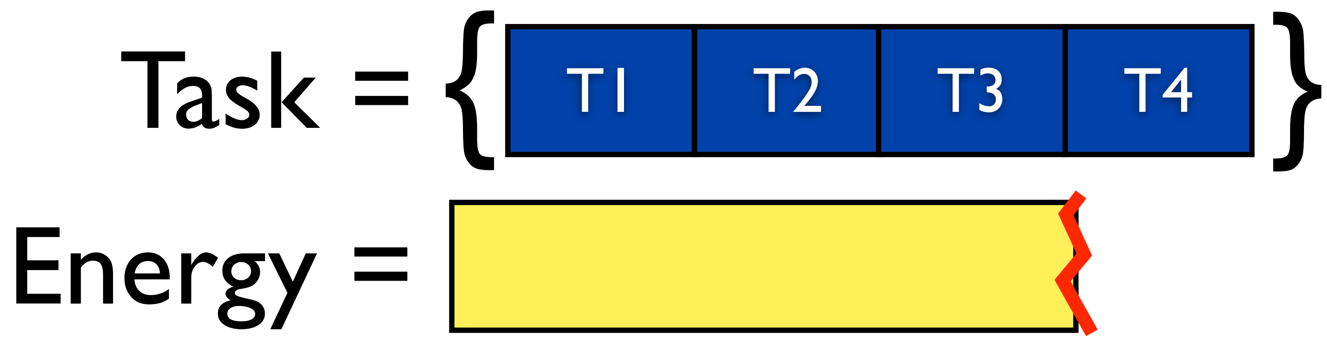
Task = { T1 | T2 | T3 | T4 }

Mementos [Ransford '08]

Task = { T1 | T2 | T3 | T4 }

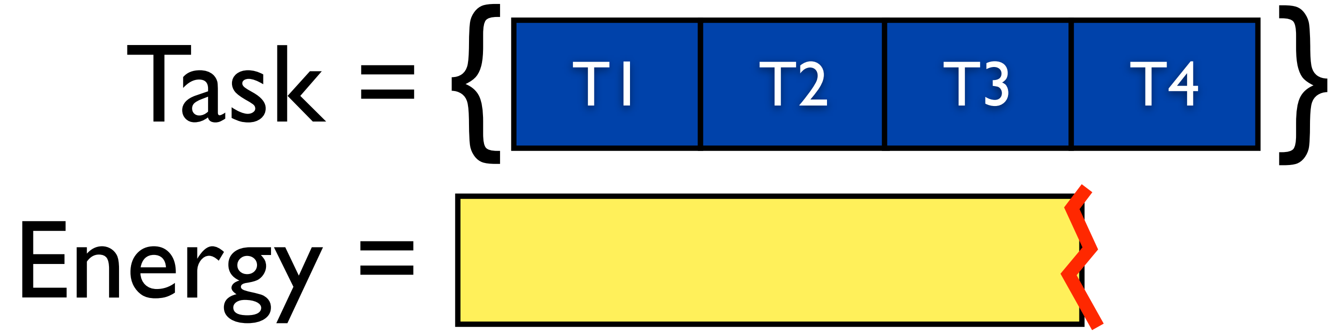
Energy =

Mementos [Ransford ’08]



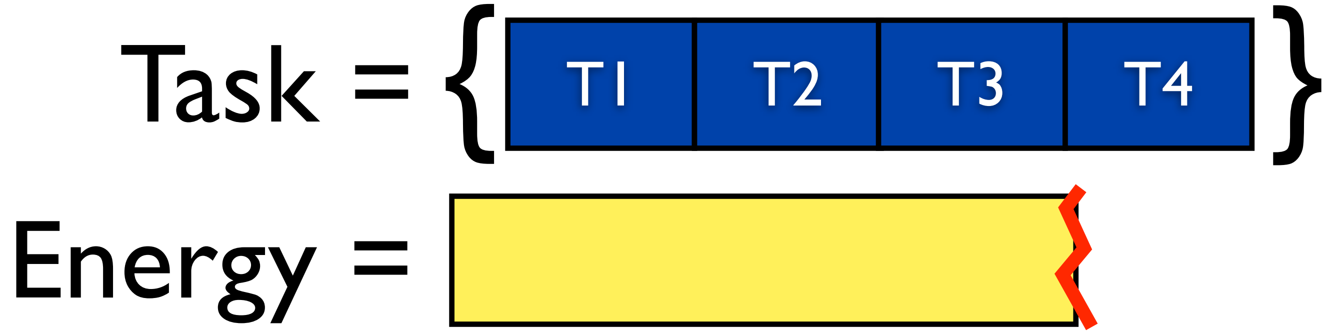
- Checkpoint state (locally) as energy wanes
- Spread computations over multiple lifecycles

Mementos [Ransford ’08]



- Checkpoint state (locally) as energy wanes
- Spread computations over multiple lifecycles

Mementos [Ransford '08]



- Checkpoint state (locally) as energy wanes
- Spread computations over multiple lifecycles

- **Problem:** flash write takes precious energy.

Security Goals

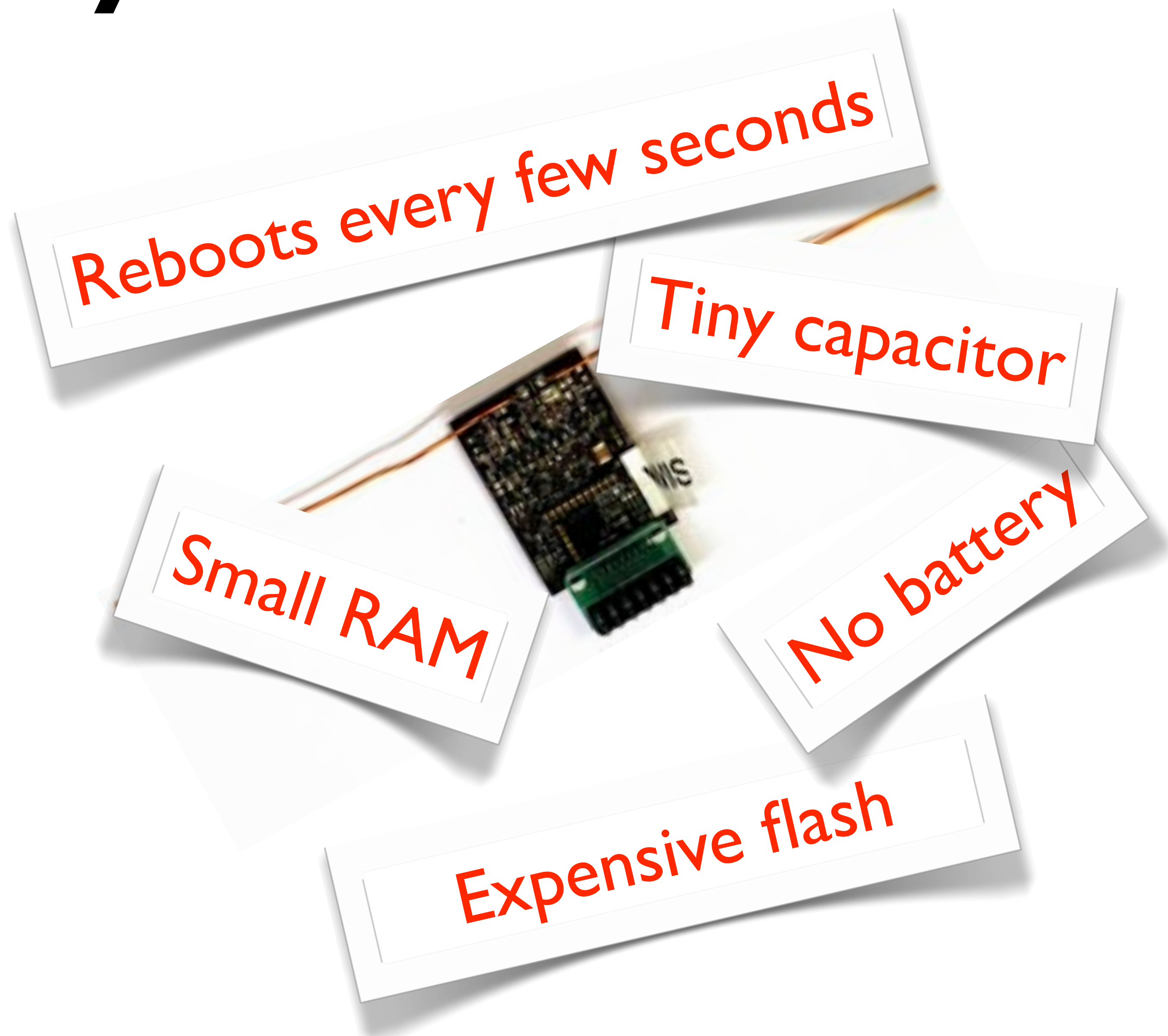
- Confidentiality
- Integrity
- Authentication
- Data Freshness
- Availability

Security Goals

- Confidentiality
- Integrity
- Authentication
- Data Freshness
- ~~Availability~~

Security Goals

- Confidentiality
- Integrity
- Authentication
- Data Freshness
- ~~Availability~~



Security Primitives

- Stream cipher for confidentiality
- UMAC for integrity/authentication [Black '99]
- Low cost in terms of energy

Security Primitives

- Stream cipher for confidentiality
- UMAC for integrity/authentication [Black '99]
- Low cost in terms of energy
- Challenge: Maintaining the *keystreams*

Precomputation?

- Keystreams are required by the cipher and the MAC
- Cannot reuse keystream bits
- Not enough energy to compute on the fly

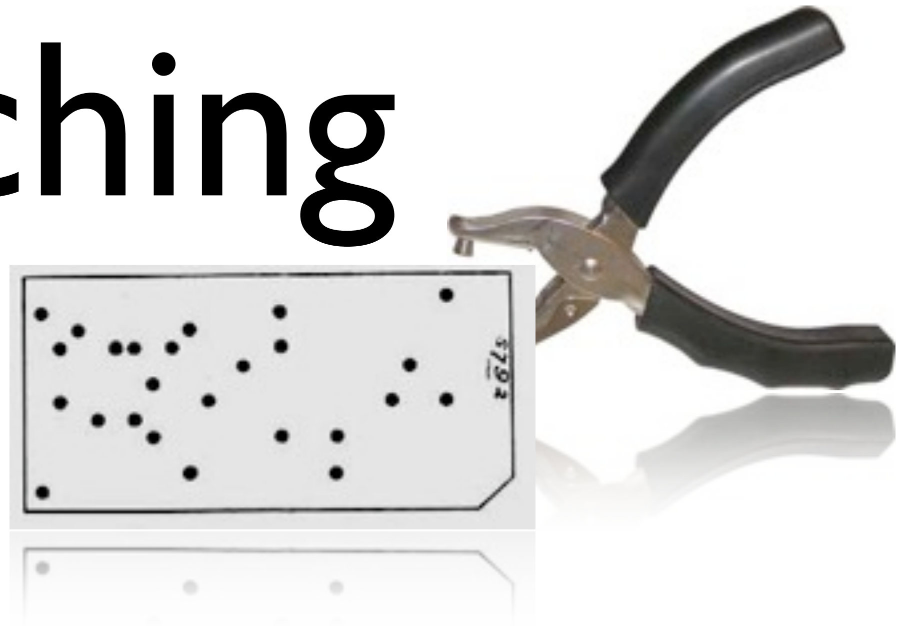
Good Power Seasons

- Times when the CRFID is idle
  - CRFID is awake and has no computation left to complete.
  - CRFID finds a reader that does not understand CCCP.
- Plentiful energy → time to produce keystream bits

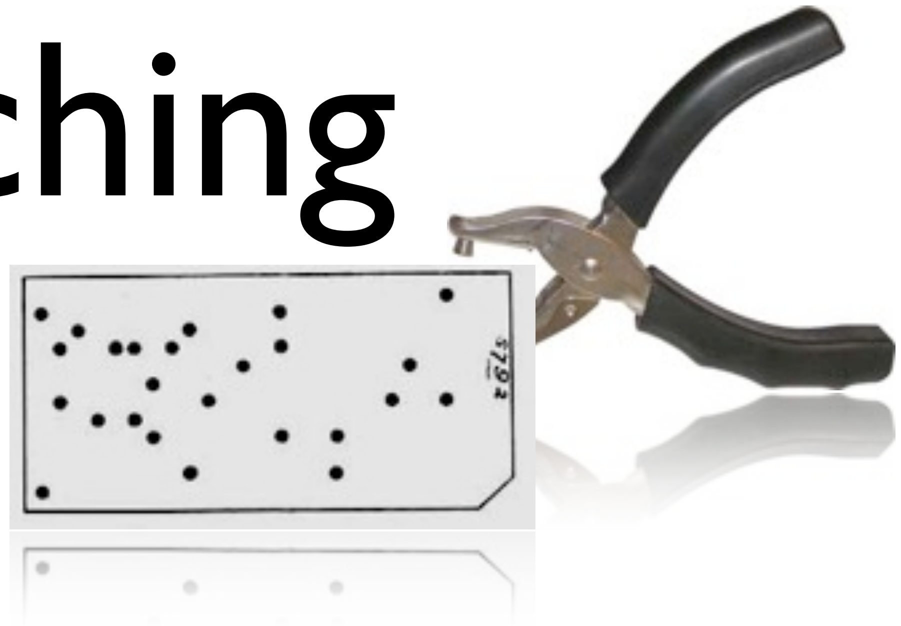
Data Freshness

- Some state must be in trusted storage
- Nonvolatile memory is too expensive to use frequently
- How can we use it frugally?

Hole Punching



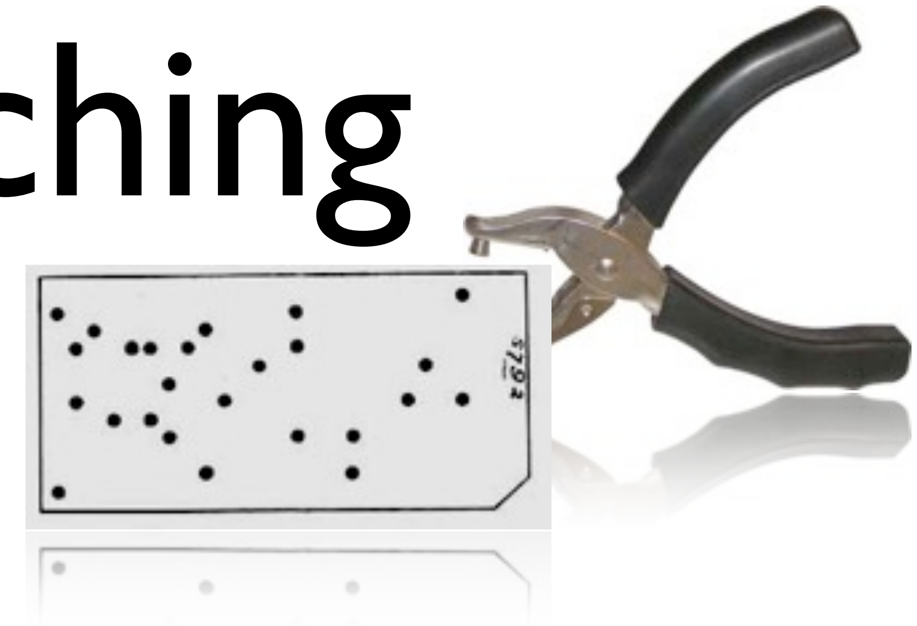
Hole Punching



$00000111_2 \quad (=7_{10})$

(a) Binary Counter

Hole Punching

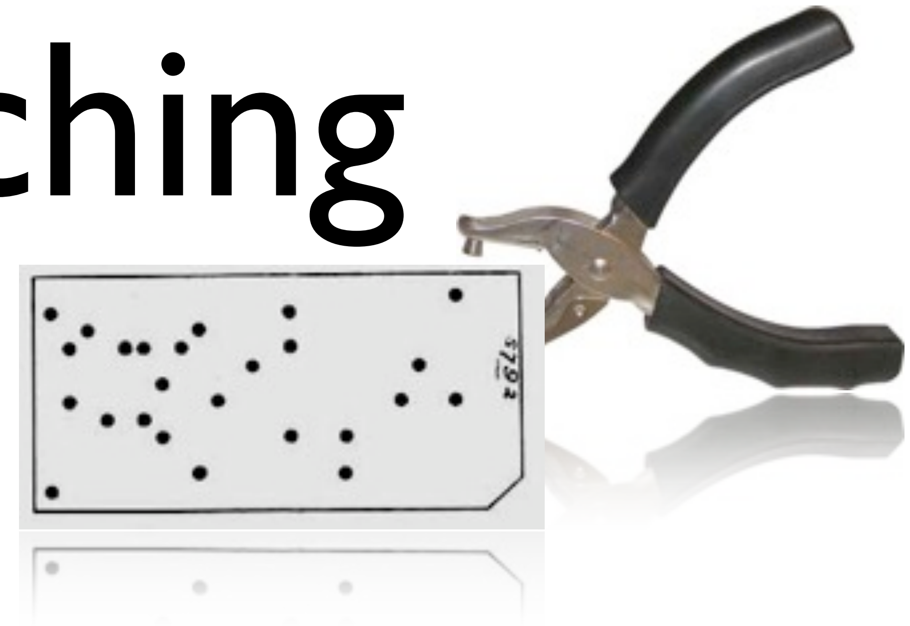


$00000111_2 \quad (=7_{10})$

$00001000_2 \quad (=8_{10})$

(a) Binary Counter

Hole Punching



00000111_2 $(=7_{10})$

↓

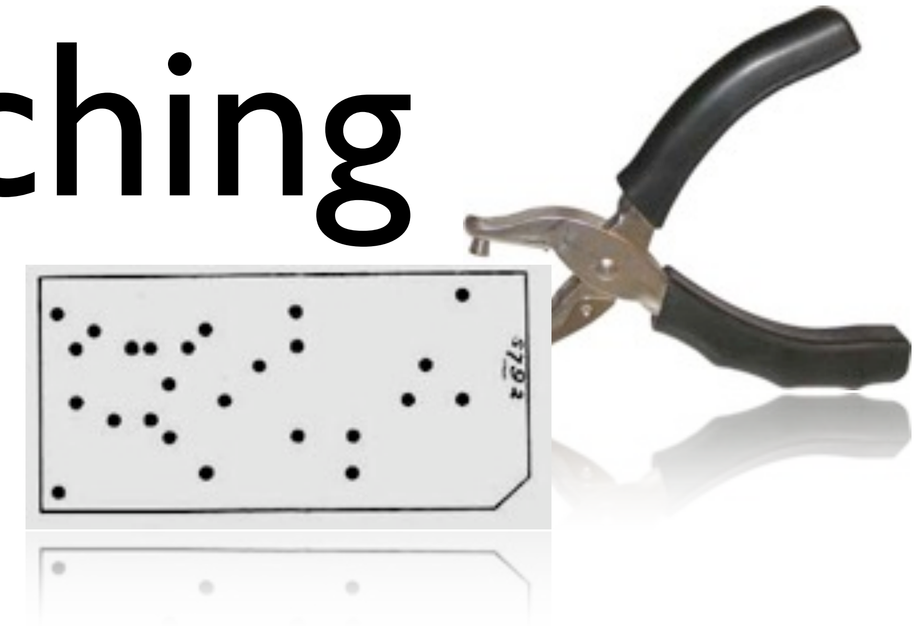
11111111_2 (erase)

↓

00001000_2 $(=8_{10})$

(a) Binary Counter

Hole Punching



00000111_2 (=7_{10})

↓

11111111_2 (erase)

↓

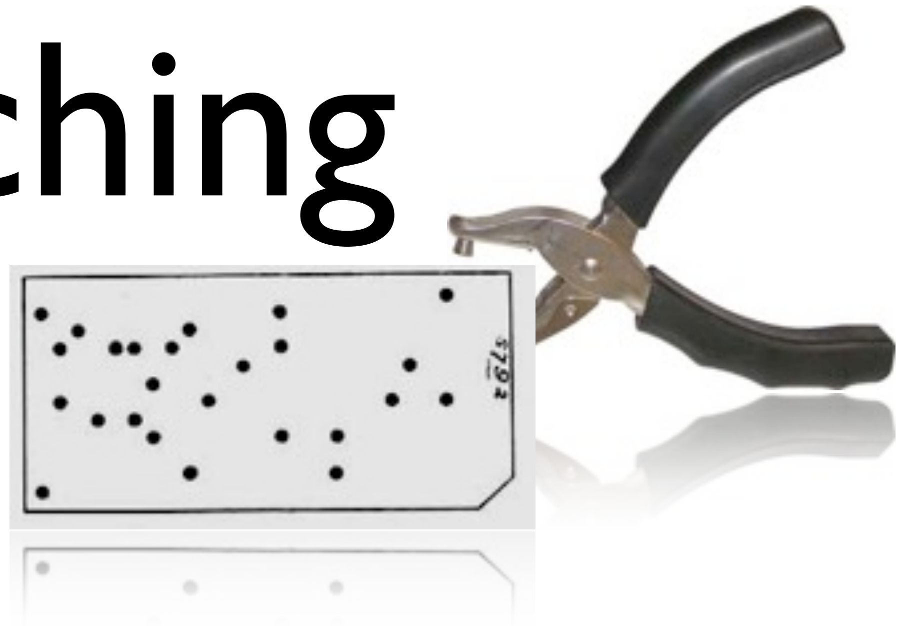
00001000_2 (=8_{10})

(a) Binary Counter

1111000000_1 (=7_{10})

(b) Unary Counter (complemented)

Hole Punching



00000111_2 $(=7_{10})$

$\downarrow$

11111111_2 (erase)

$\downarrow$

00001000_2 $(=8_{10})$

(a) Binary Counter

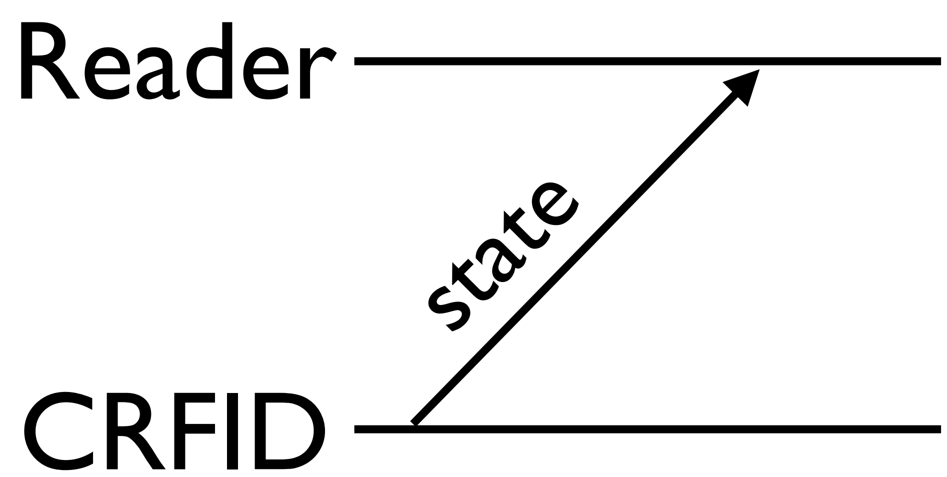
11110000000_1 $(=7_{10})$

$\downarrow$

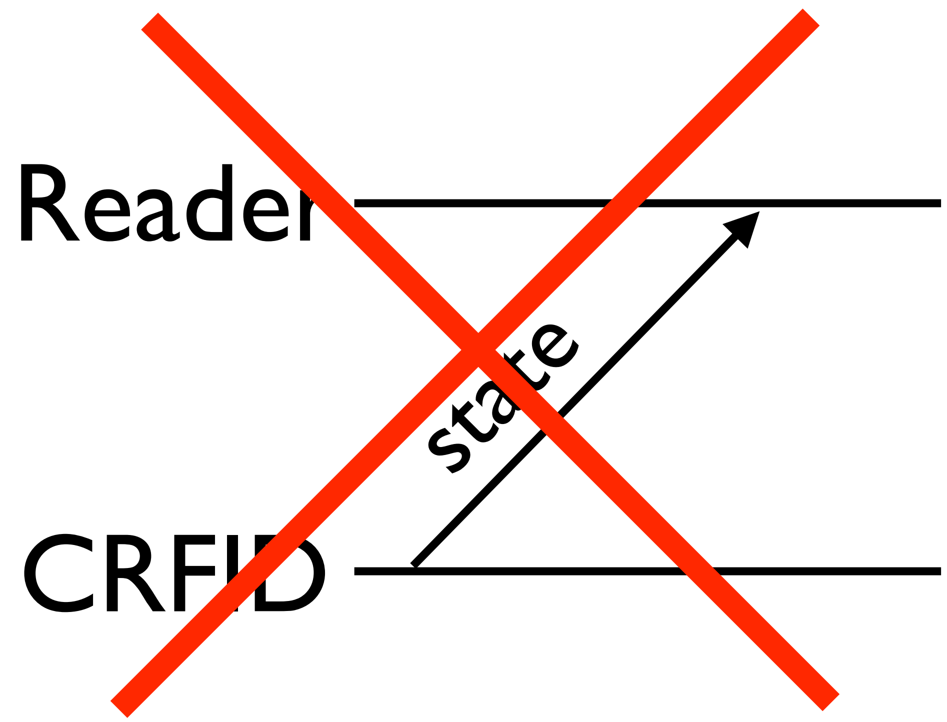
11100000000_1 $(=8_{10})$

(b) Unary Counter (complemented)

Protocol



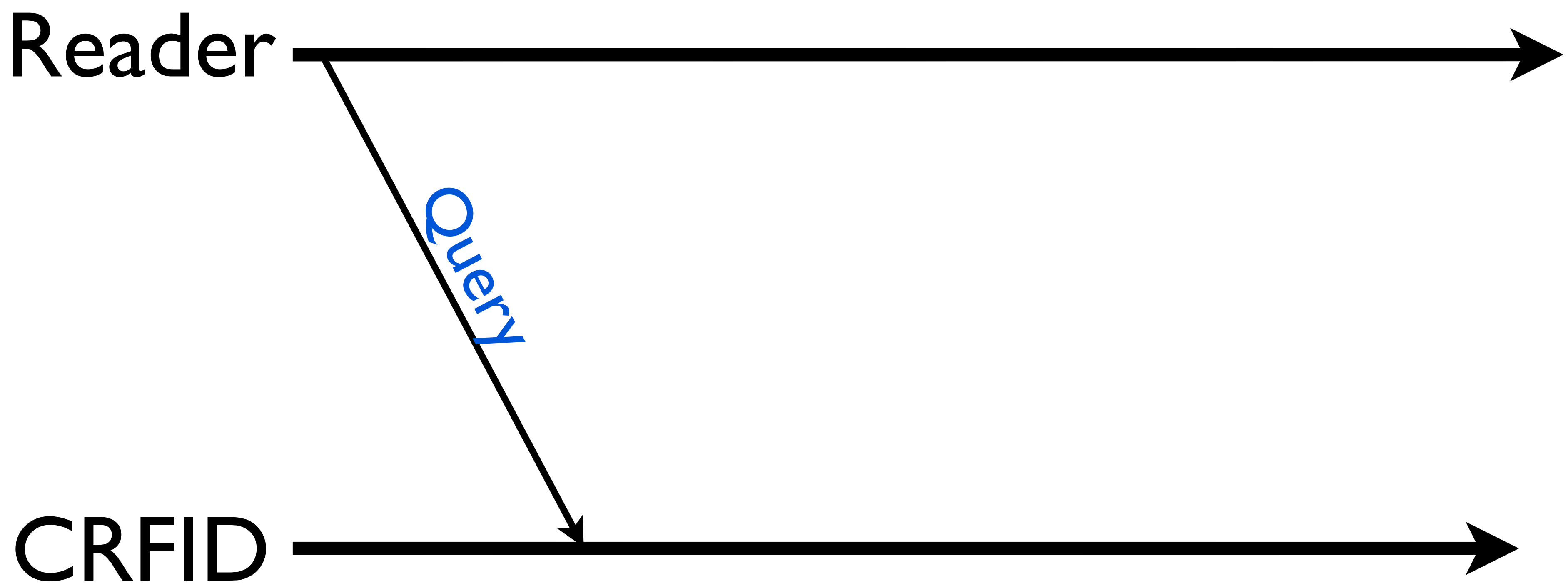
Protocol



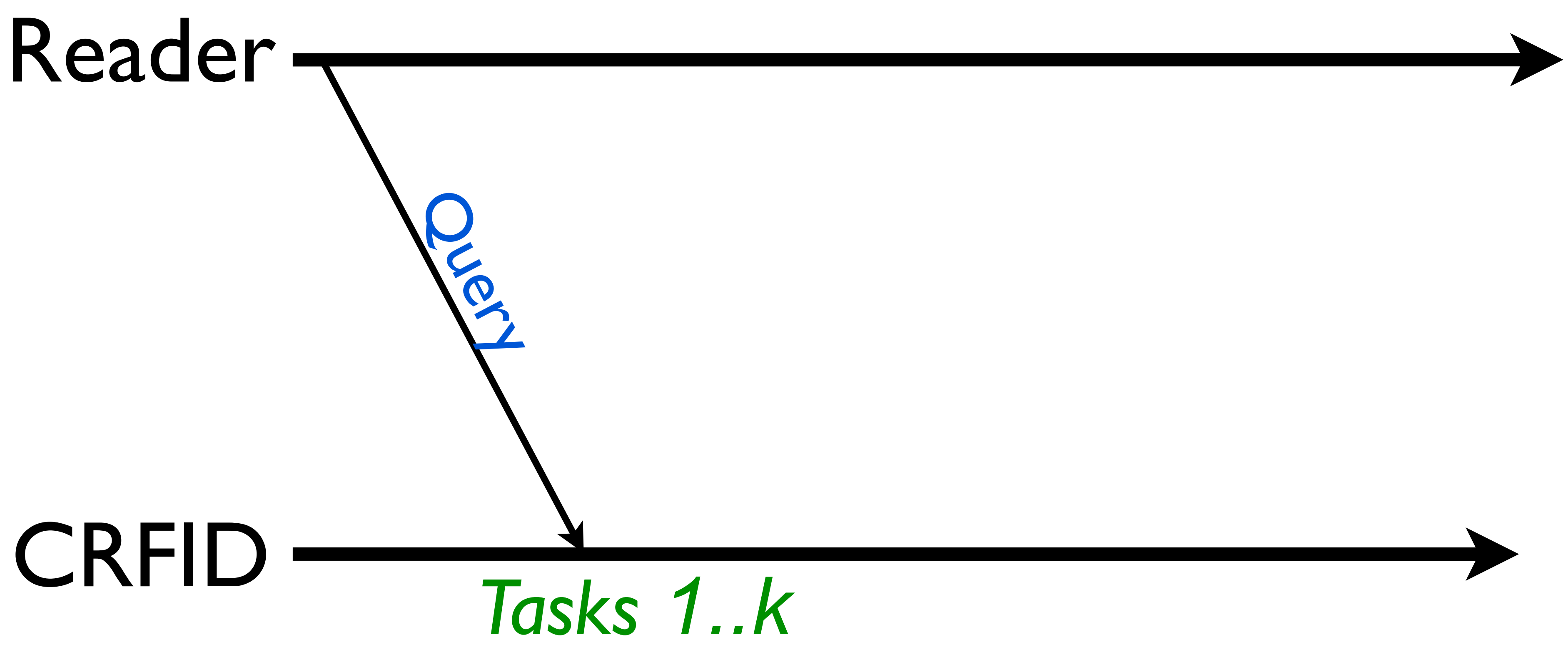
Non-autonomous communication

Store Procedure

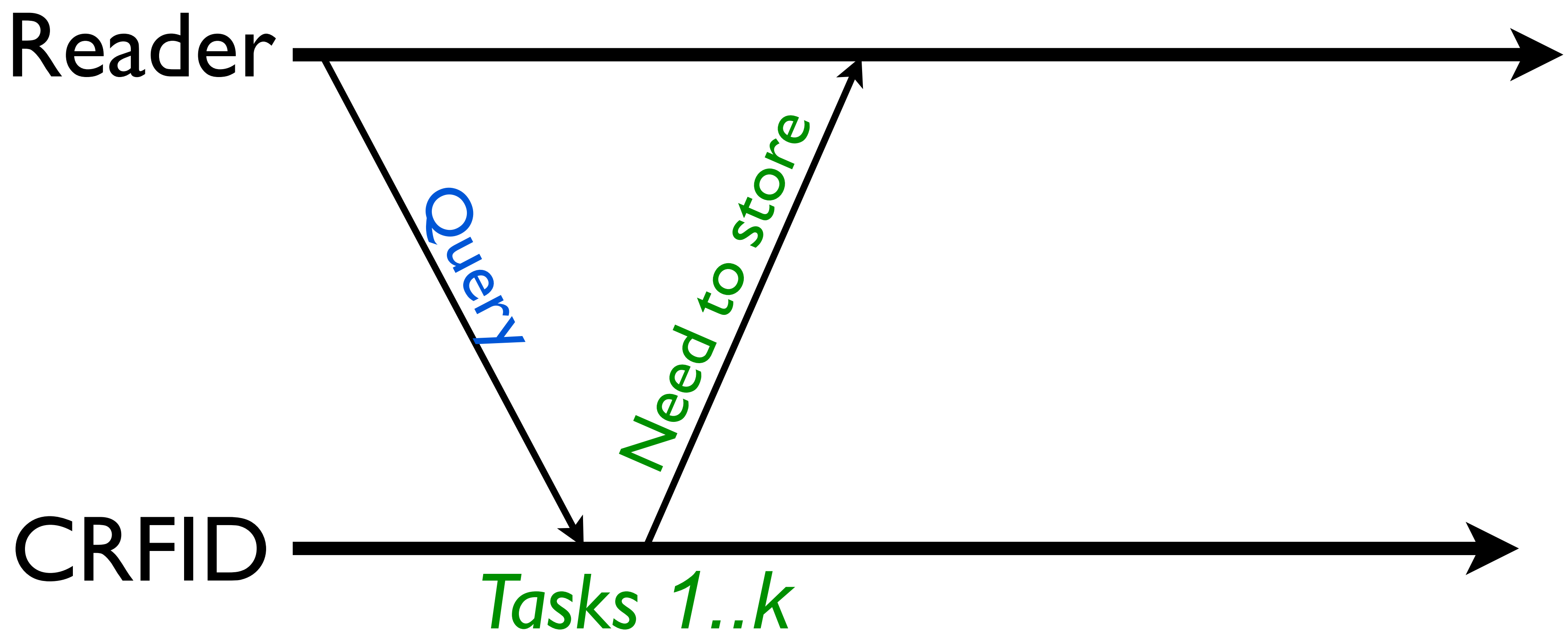
Store Procedure



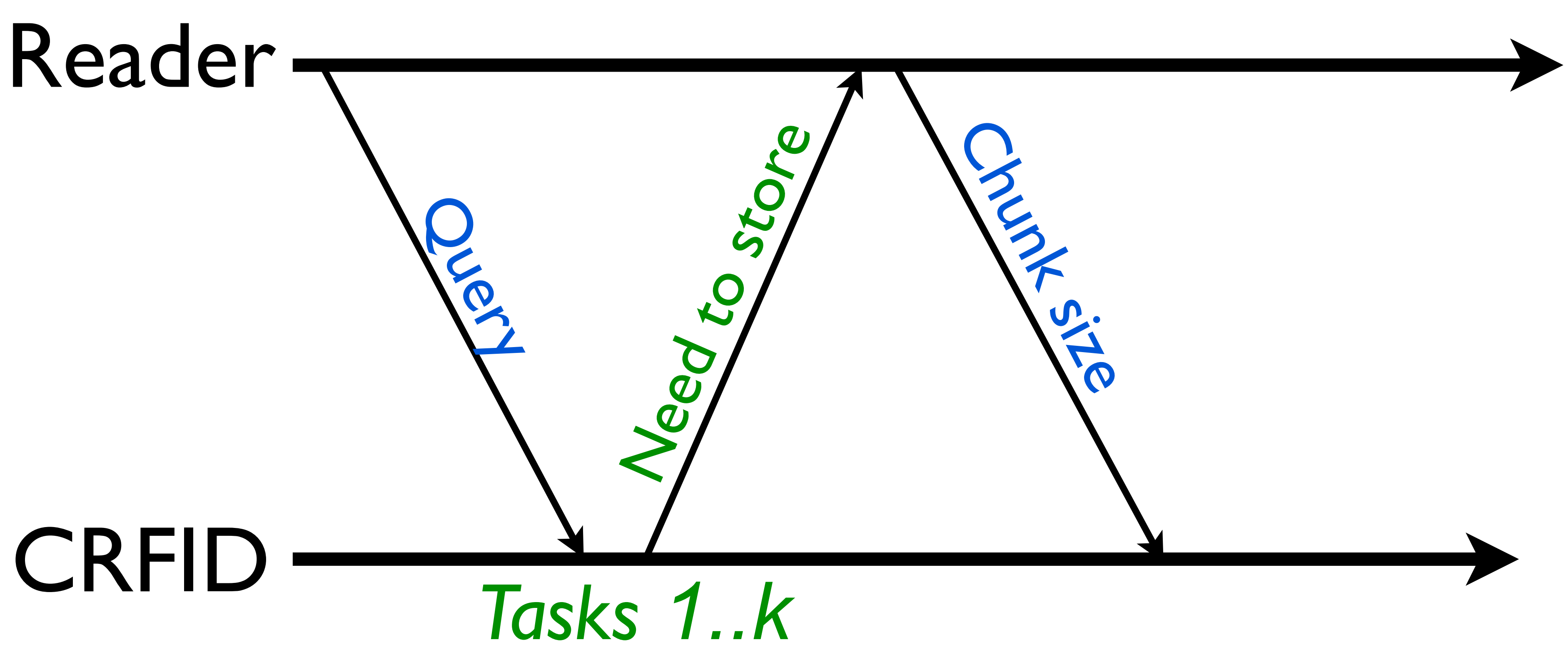
Store Procedure



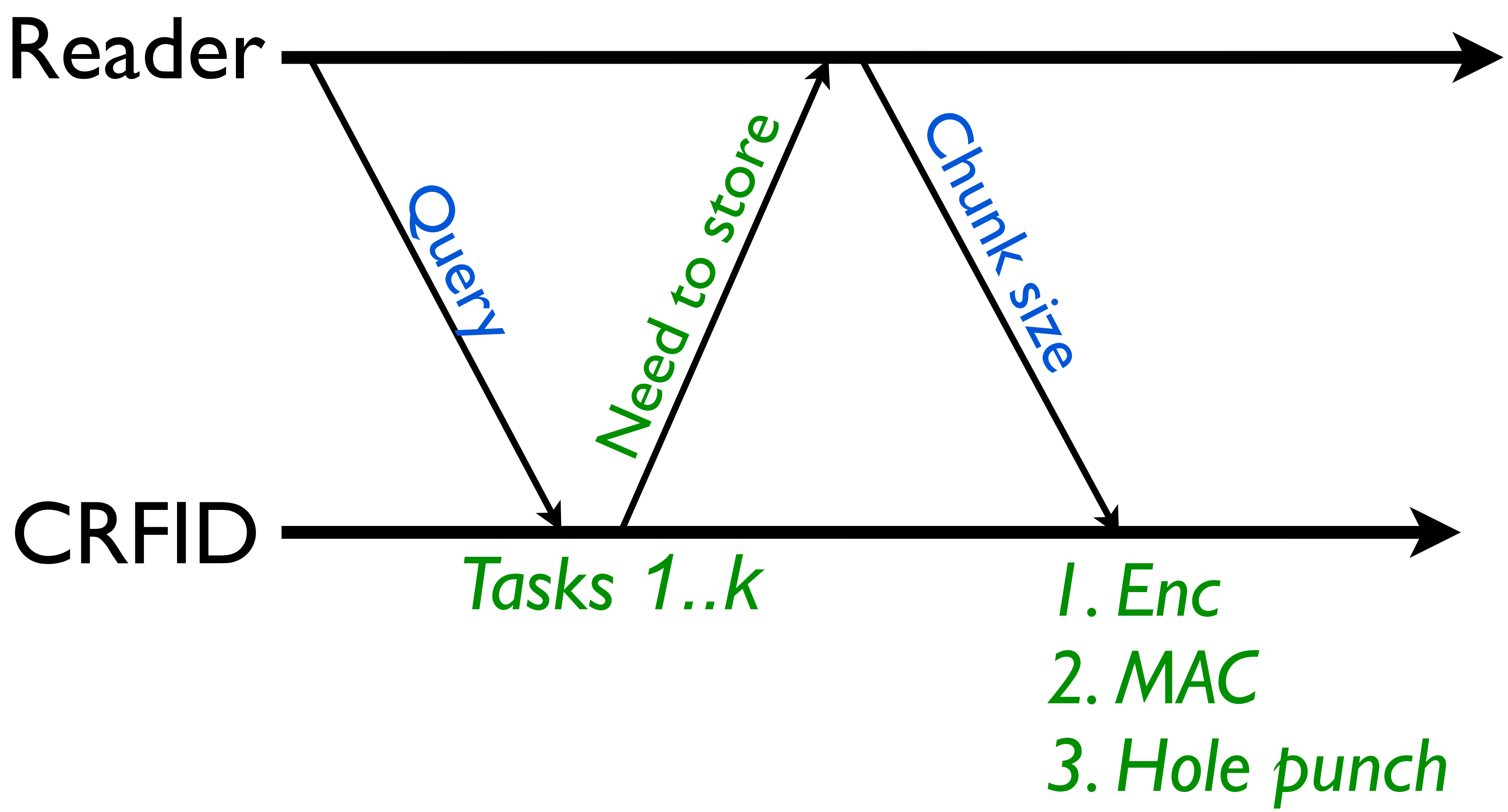
Store Procedure



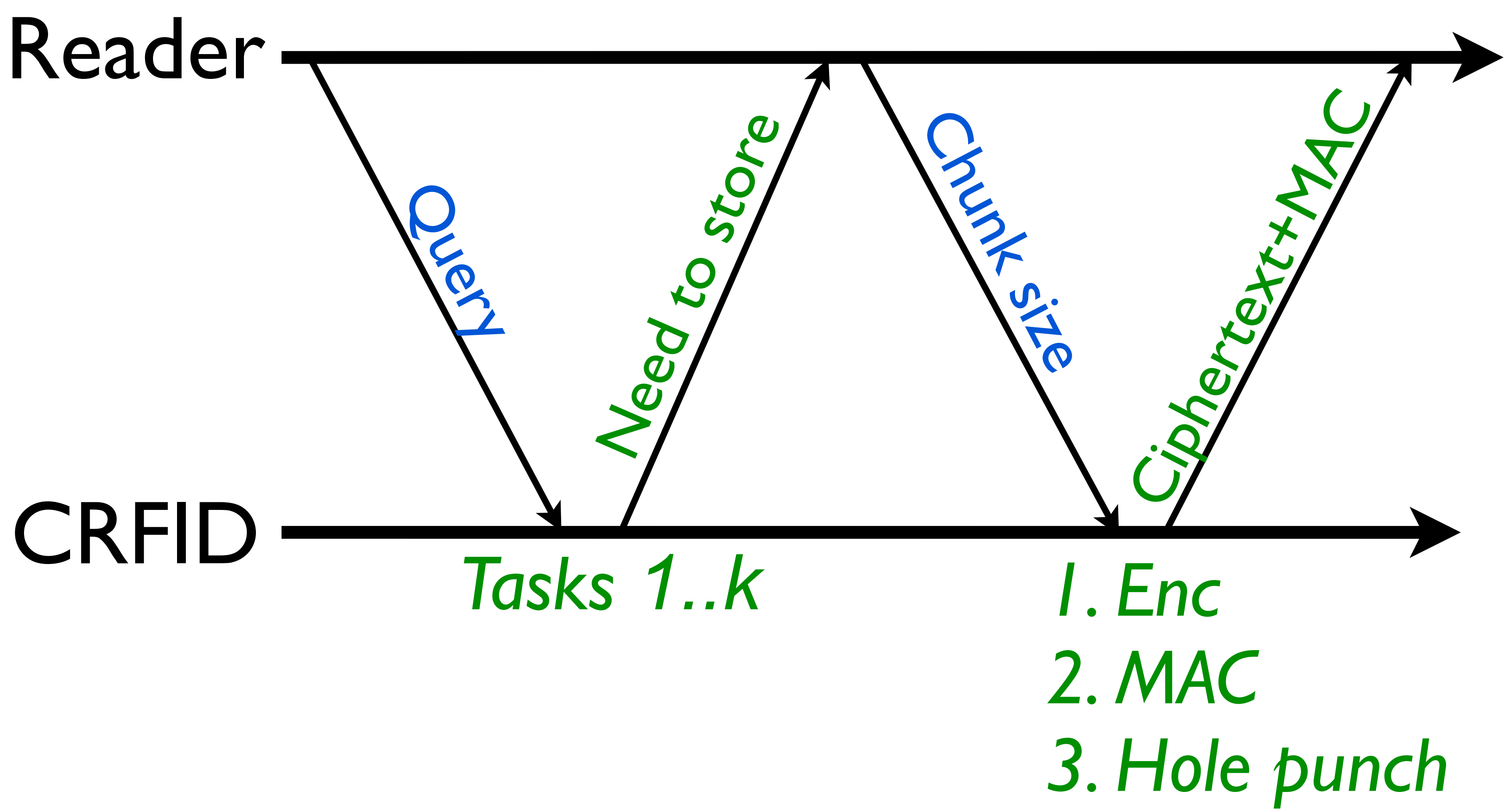
Store Procedure



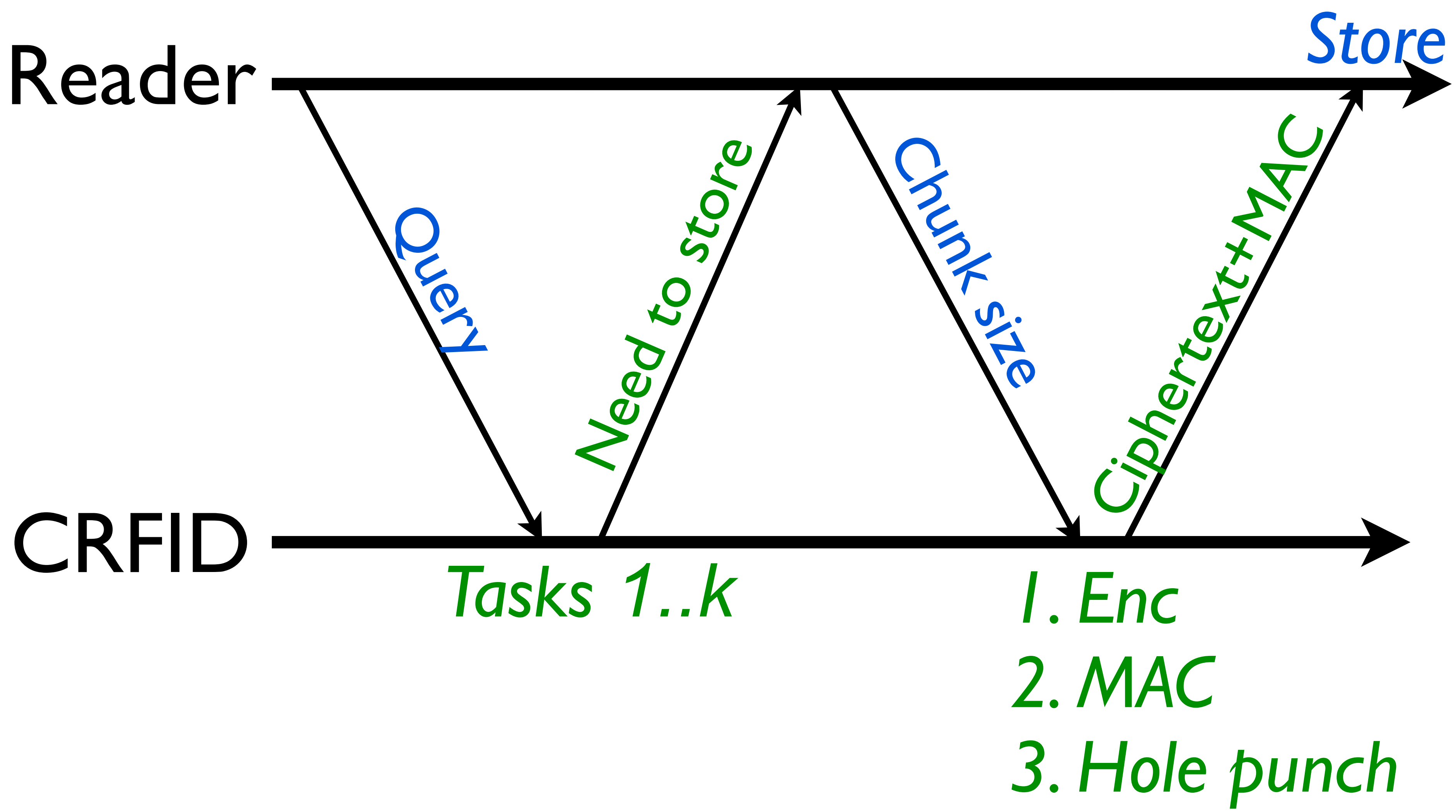
Store Procedure



Store Procedure



Store Procedure

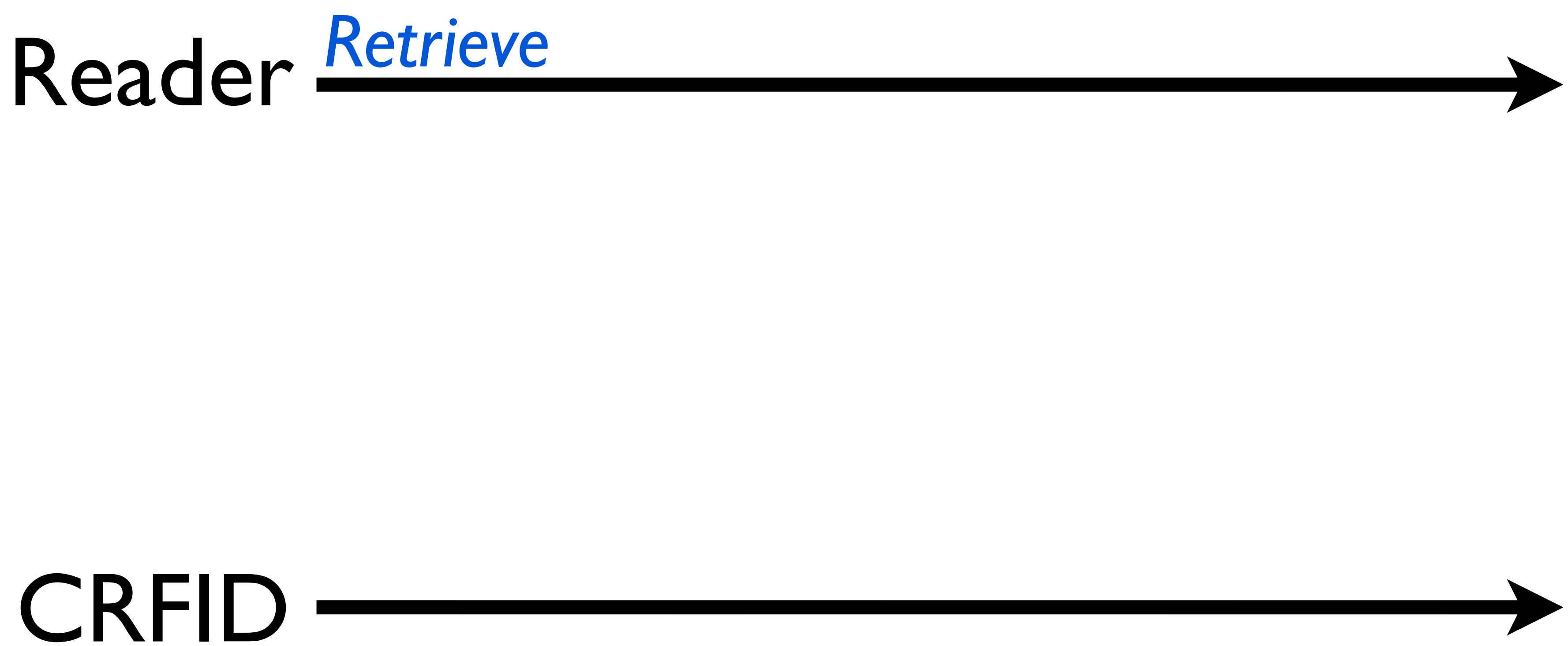


Retrieve Procedure

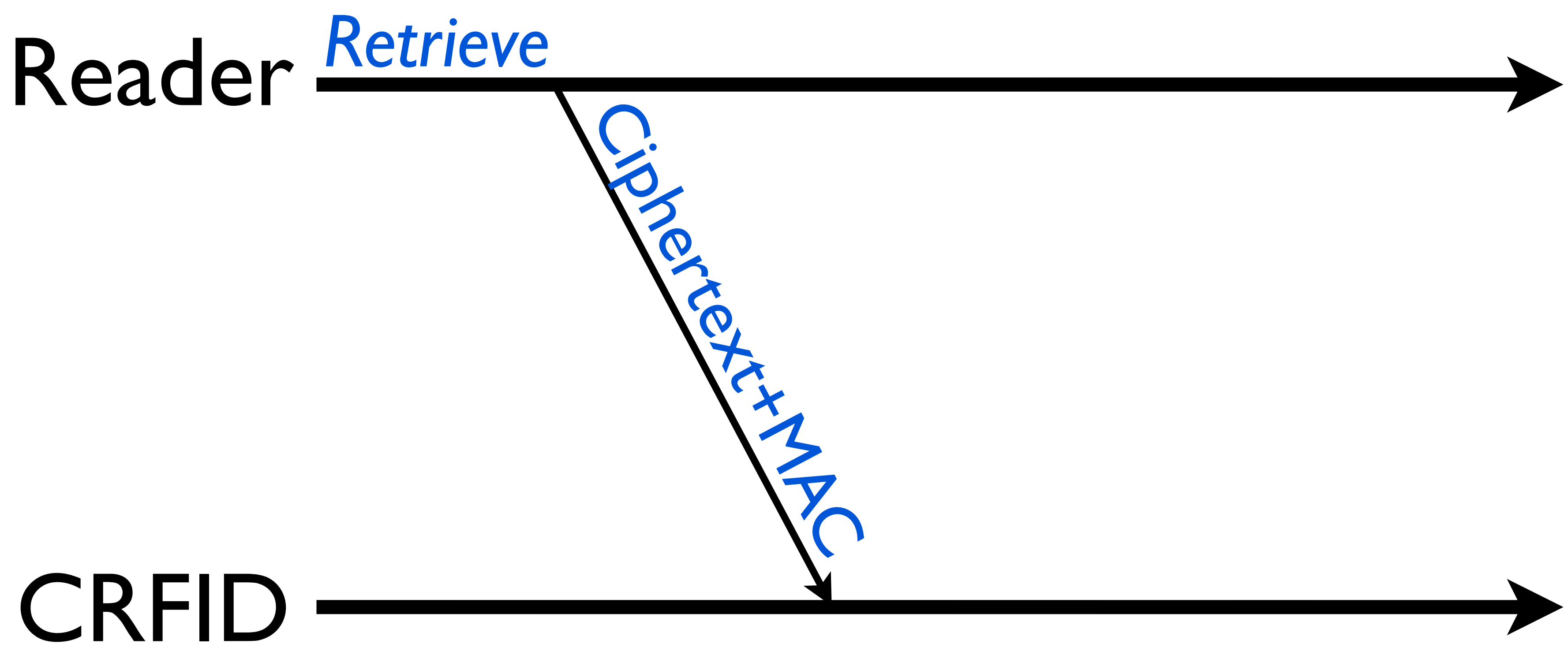
Reader

CRFID

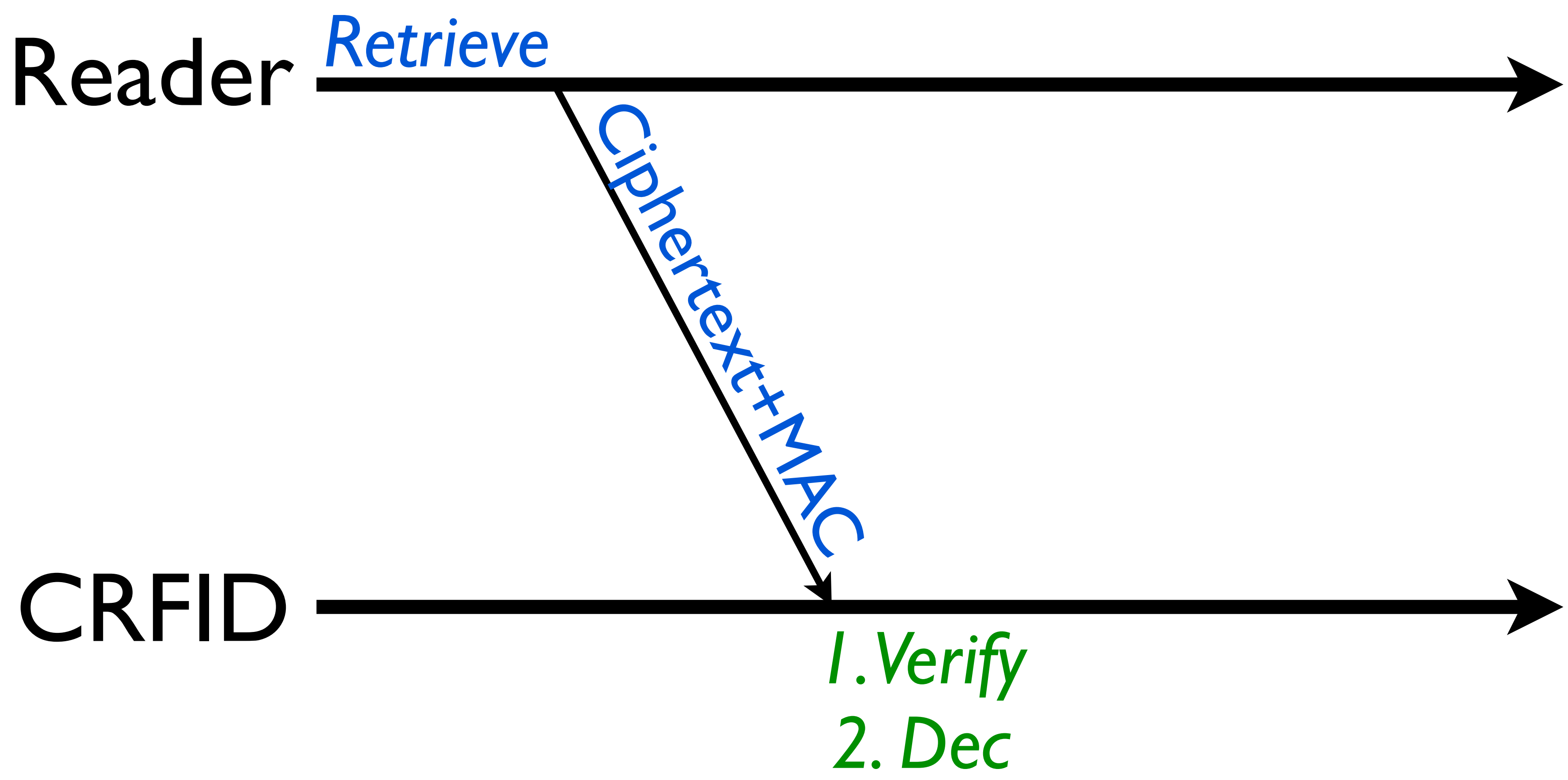
Retrieve Procedure



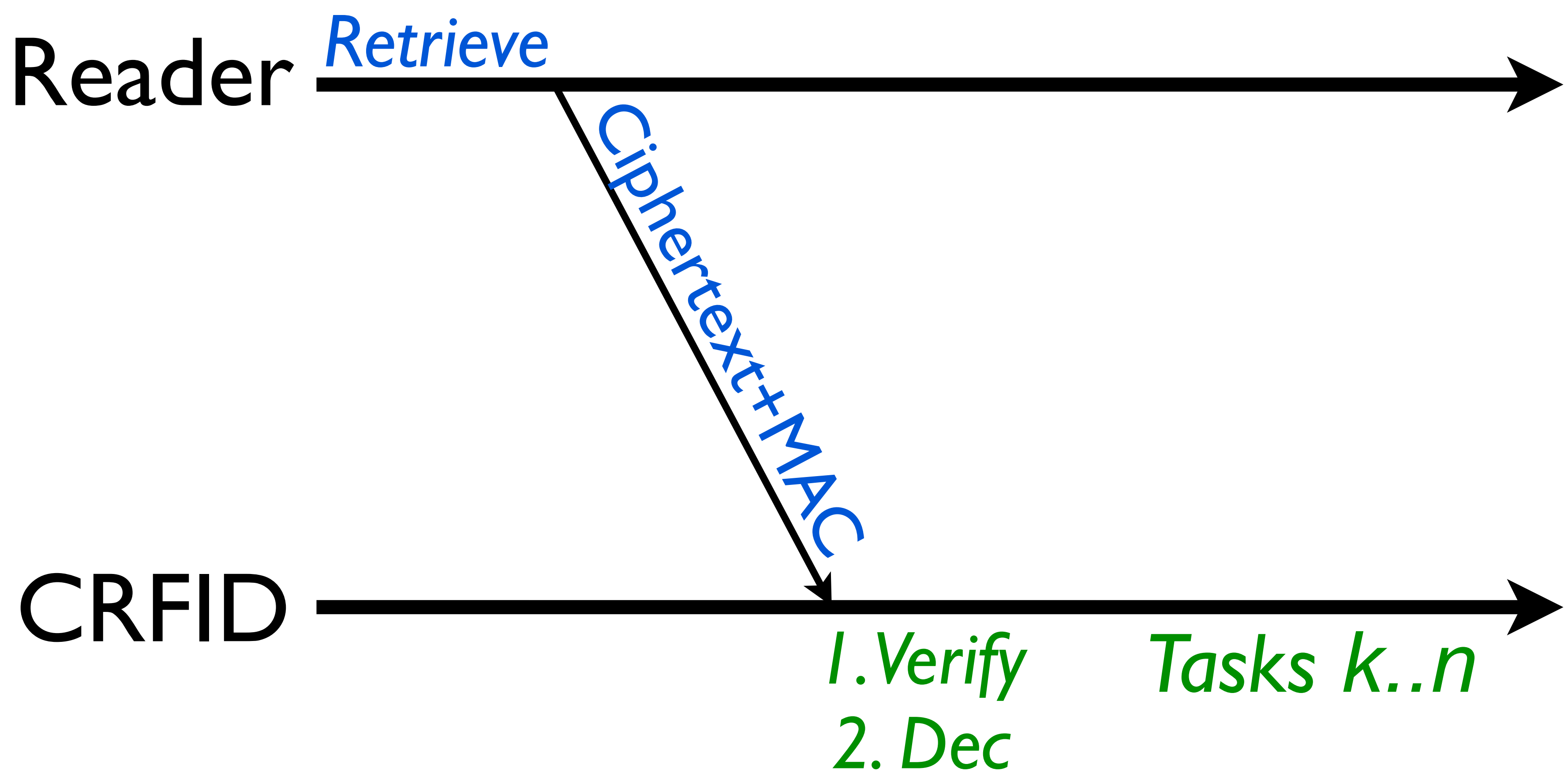
Retrieve Procedure



Retrieve Procedure



Retrieve Procedure

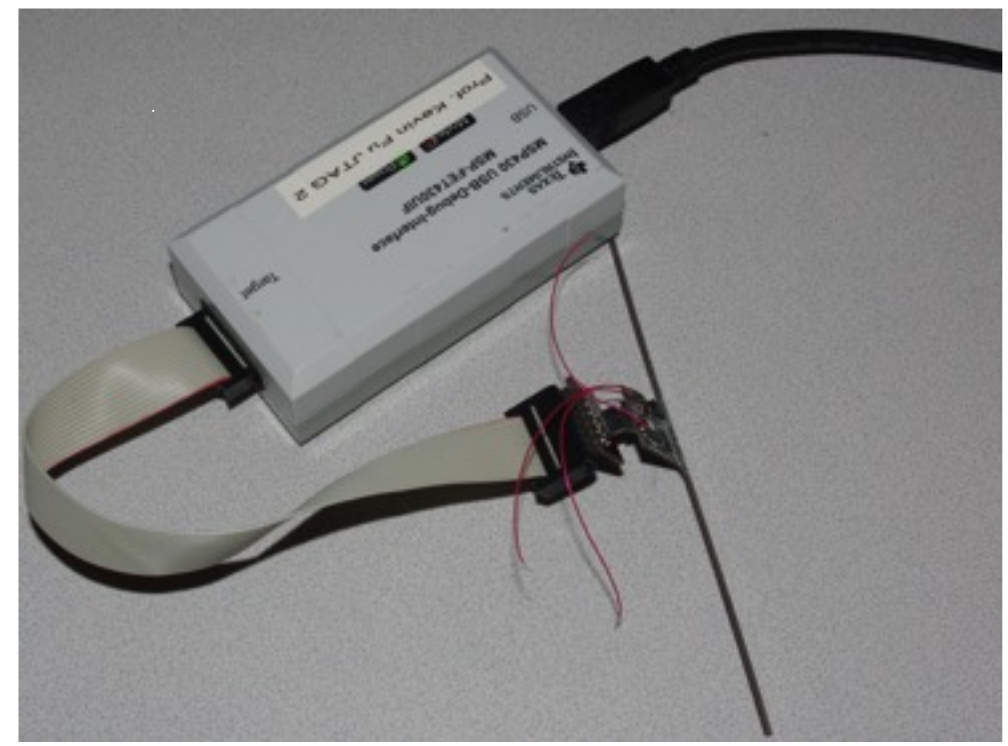


Evaluation

Experimental setup

Experimental setup

1. Program the CRFID with a task



Experimental setup

1. Program the CRFID with a task

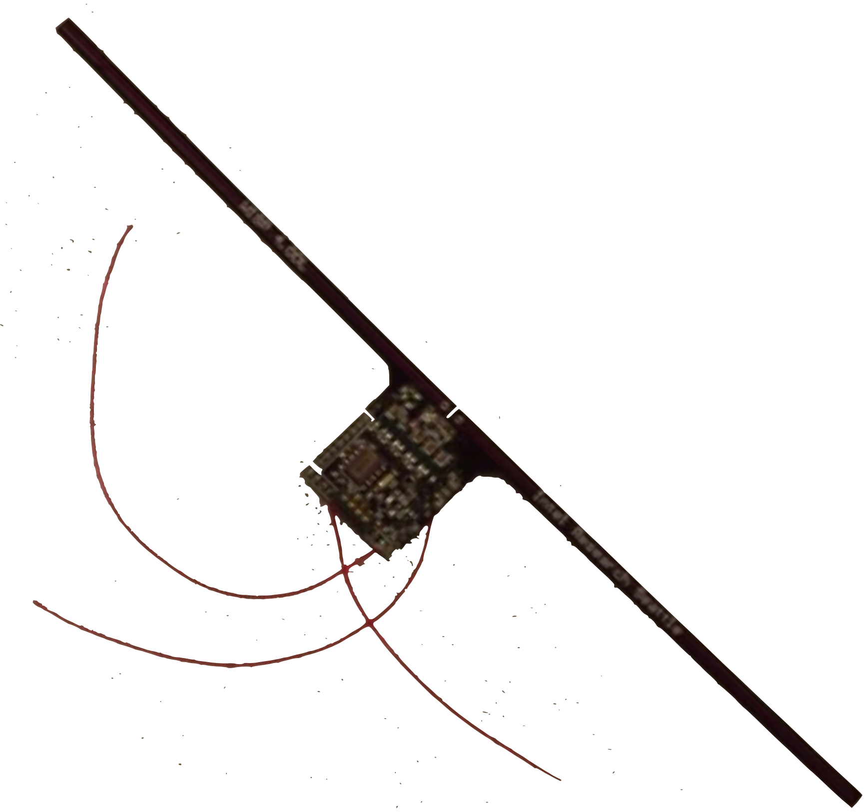
2. Charge CRFID to voltage V

Experimental setup

1. Program the CRFID with a task

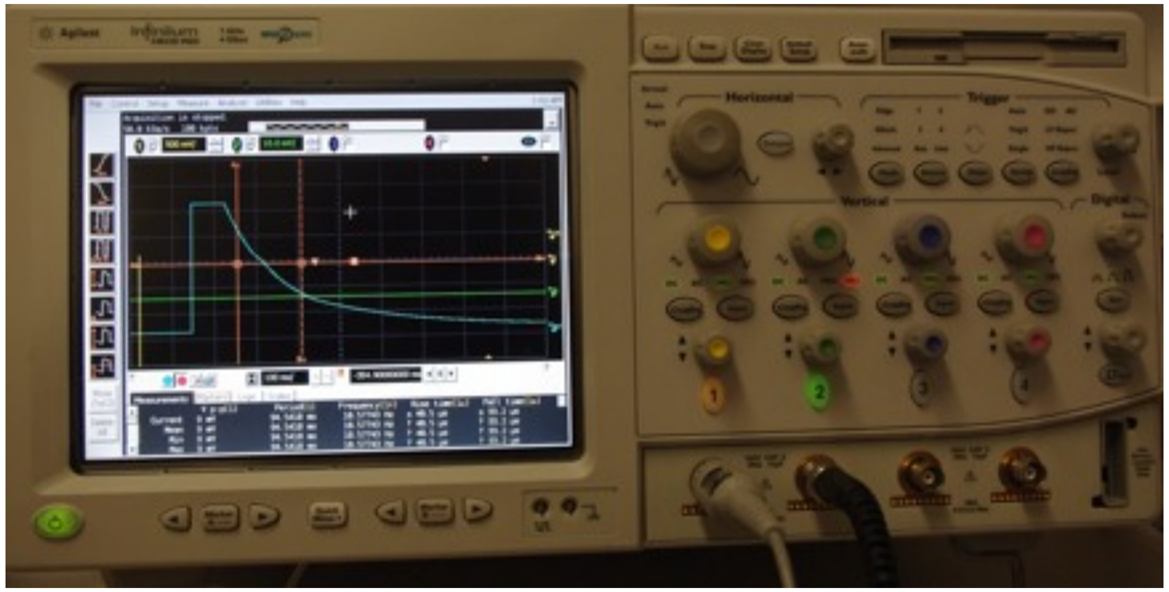
2. Charge CRFID to voltage V

3. Disconnect the power supply



Experimental setup

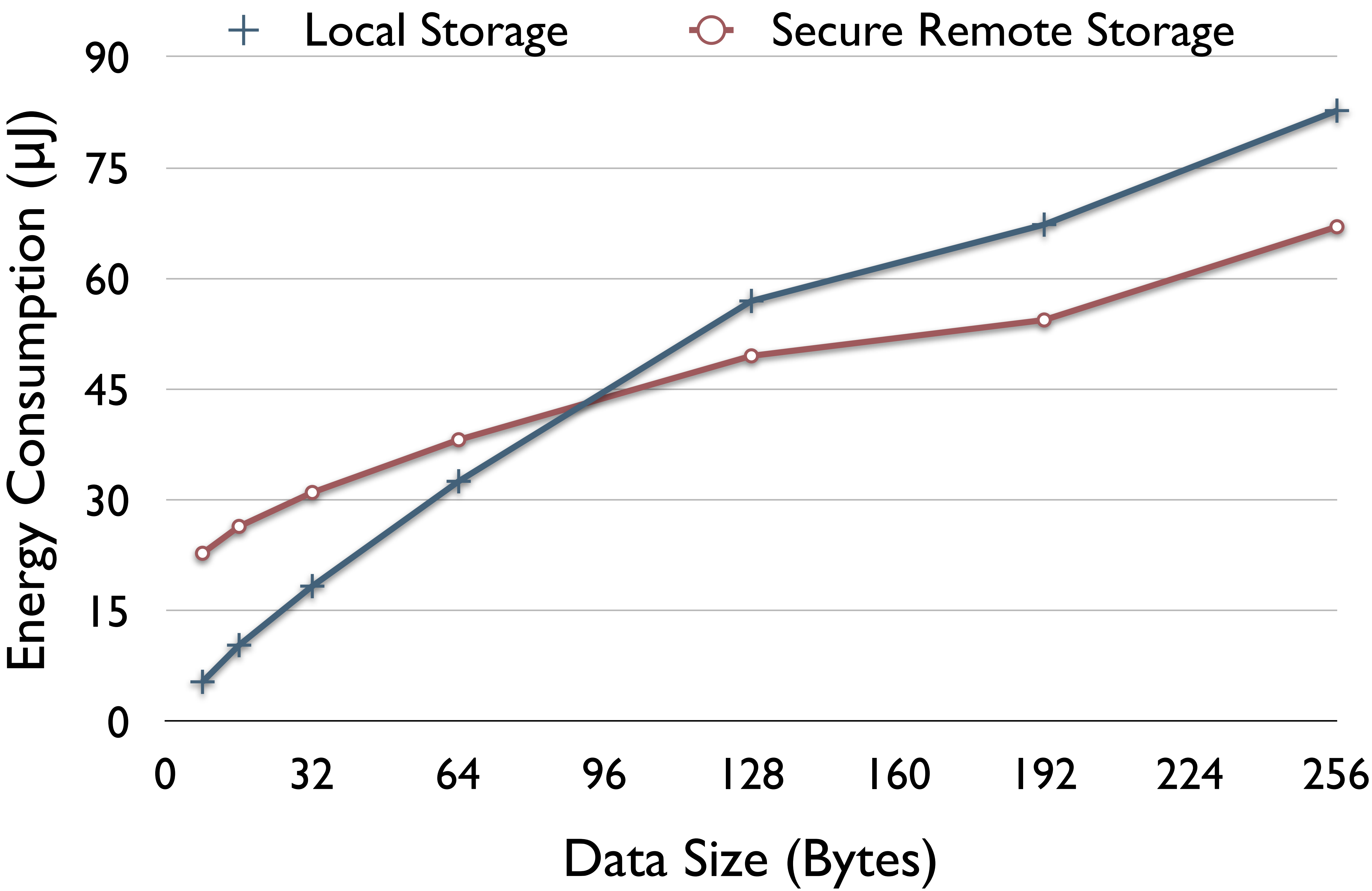
1. Program the CRFID with a task

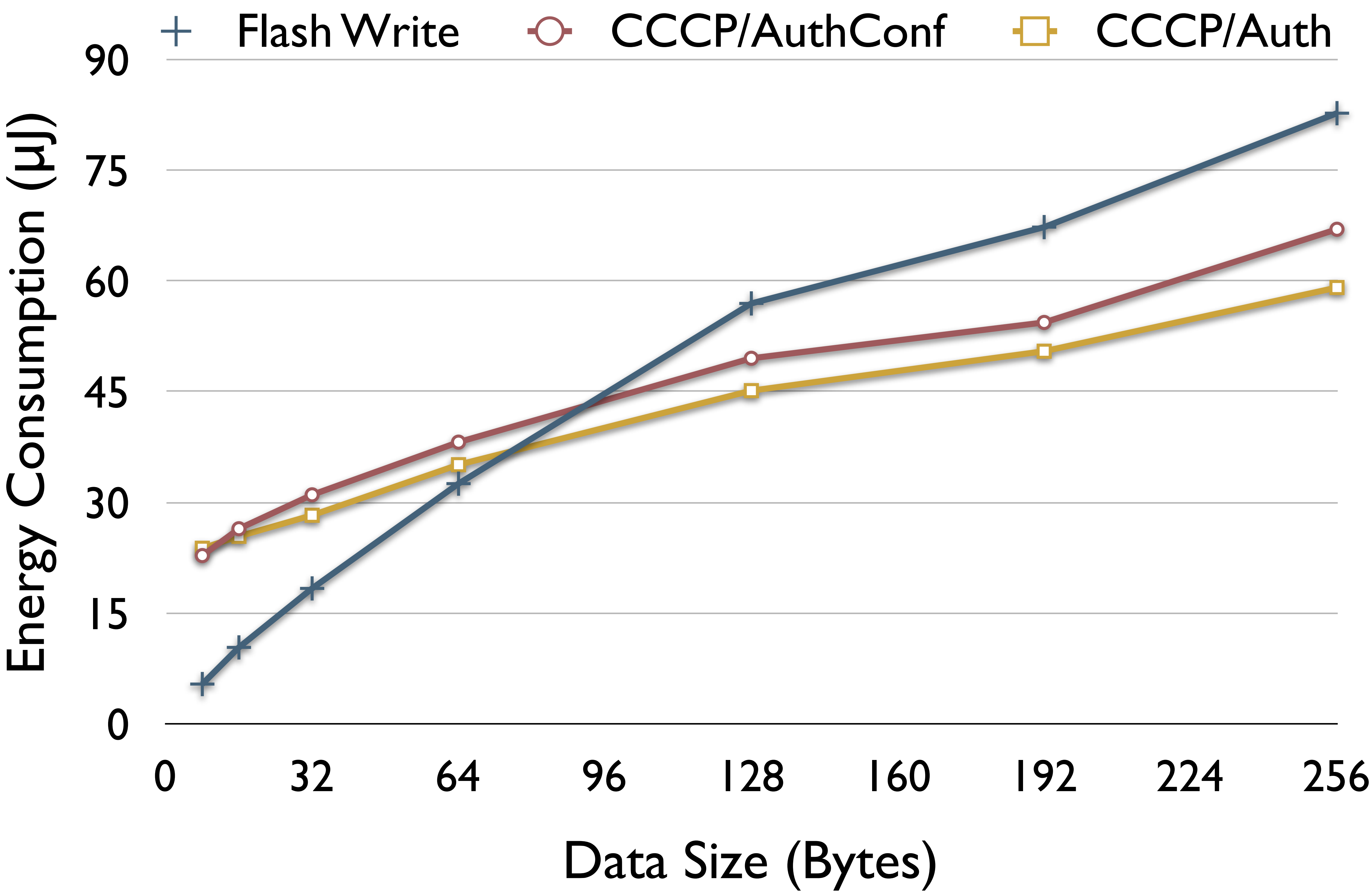


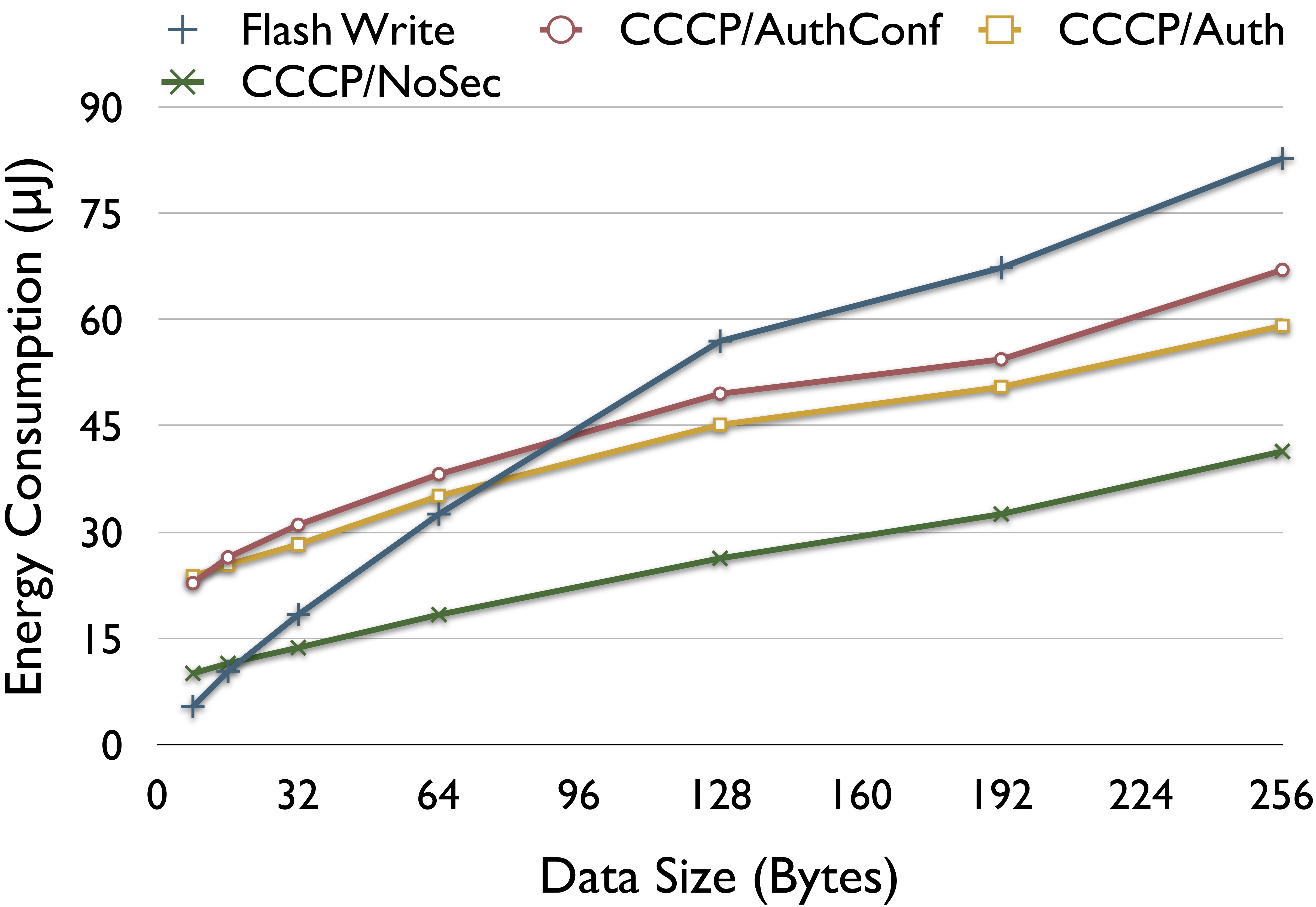
2. Charge CRFID to voltage V

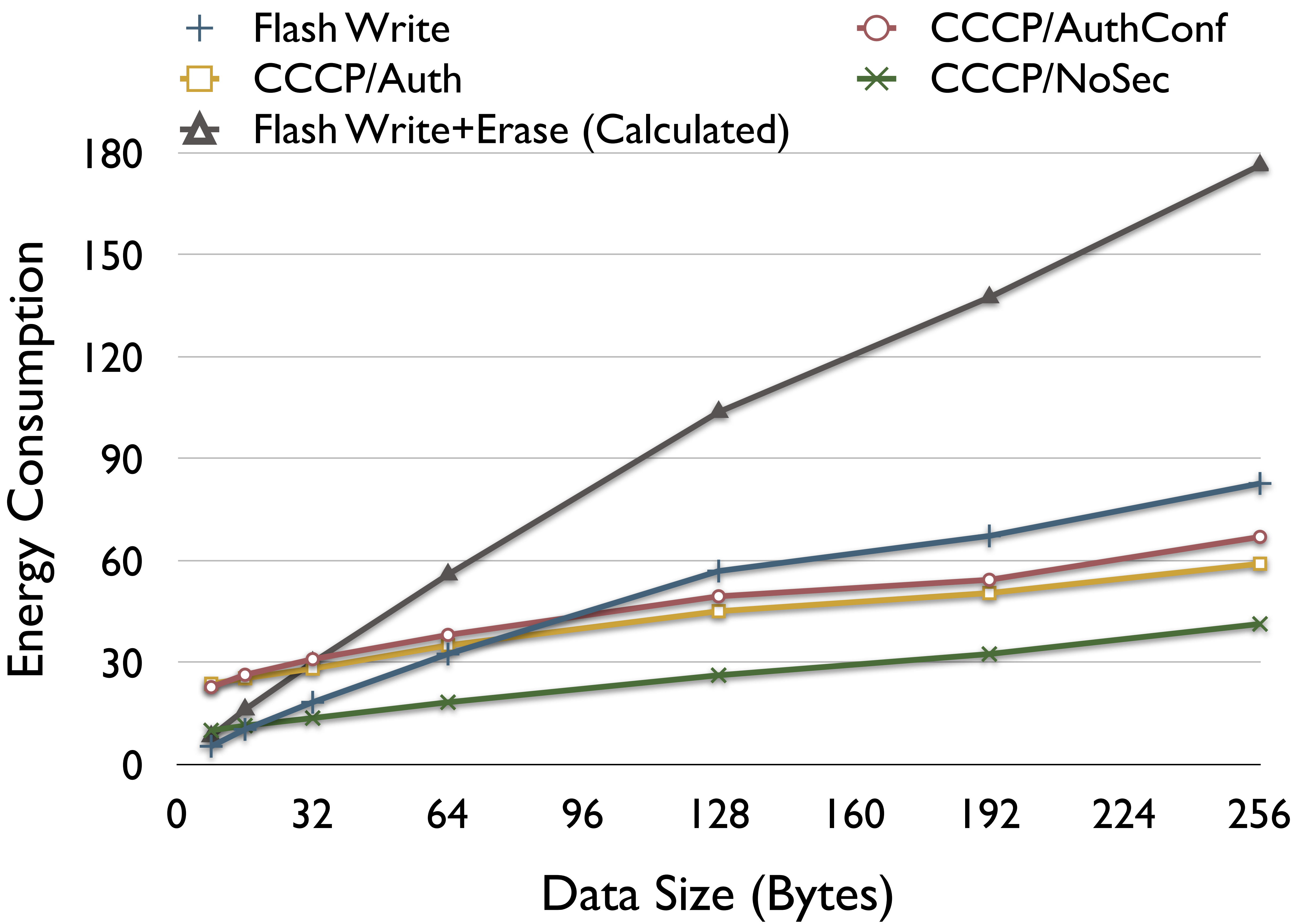
3. Disconnect the power supply

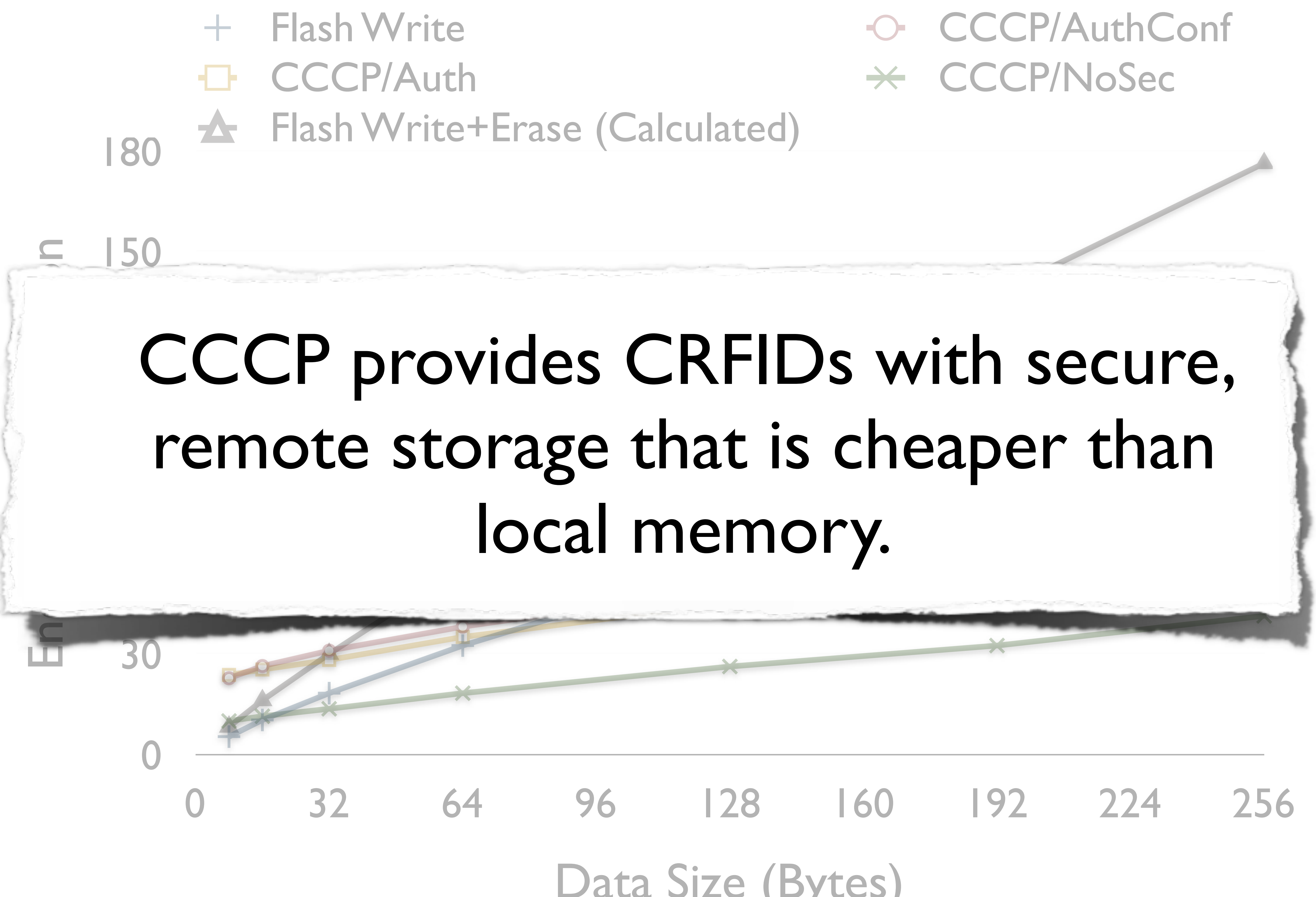
4. Observe the voltage drop and execution time


Local Storage
Secure Remote Storage
Energy Consumption (μJ)
90
75
60
45
30
15
0
0
32
64
96
128
160
192
224
256
Data Size (Bytes)


Flash Write
CCCP/AuthConf
CCCP/Auth
Energy Consumption (μJ)
90
75
60
45
30
15
0
0
32
64
96
128
160
192
224
256
Data Size (Bytes)


Flash Write
CCCP/AuthConf
CCCP/Auth
CCCP/NoSec
Energy Consumption (μJ)
90
75
60
45
30
15
0
0
32
64
96
128
160
192
224
256
Data Size (Bytes)


Flash Write
CCCP/Auth
Flash Write+Erase (Calculated)
CCCP/AuthConf
CCCP/NoSec
Energy Consumption
180
150
120
90
60
30
0
0
32
64
96
128
160
192
224
256
Data Size (Bytes)



Flash Write
CCCP/AuthConf
CCCP/Auth
CCCP/NoSec
Flash Write+Erase (Calculated)
180
150
30
0
0
32
64
96
128
160
192
224
256
Data Size (Bytes)
CCCP provides CRFIDs with secure, remote storage that is cheaper than local memory.

Extensions/Future Work

- CRFID hardware design
- Long-term storage
- WOM codes [Rivest '82]
- PKCS on CRFIDs

Summary

- CRFIDs can go where other platforms cannot
- They are limited by available energy
- Remote storage is cheap
- CCCP provides remote storage that is secure and yet less expensive than local storage.

Summary

- CRFIDs can go where other platforms cannot
- They are limited by available energy
- Remote storage is cheap
- CCCP provides remote storage that is secure and yet less expensive than local storage.

More info at: www.cs.umass.edu/~ssclark/crfid