Accelerating PUF-based Authentication Protocols using Programmable Switch

Divya Pathak[1] Ranjitha K[1] Krishna Sai Modali[2] Praveen Tammana[1]

Antony Franklin[1] Tejasvi Alladi[2]

IIT Hyderabad, India[1]

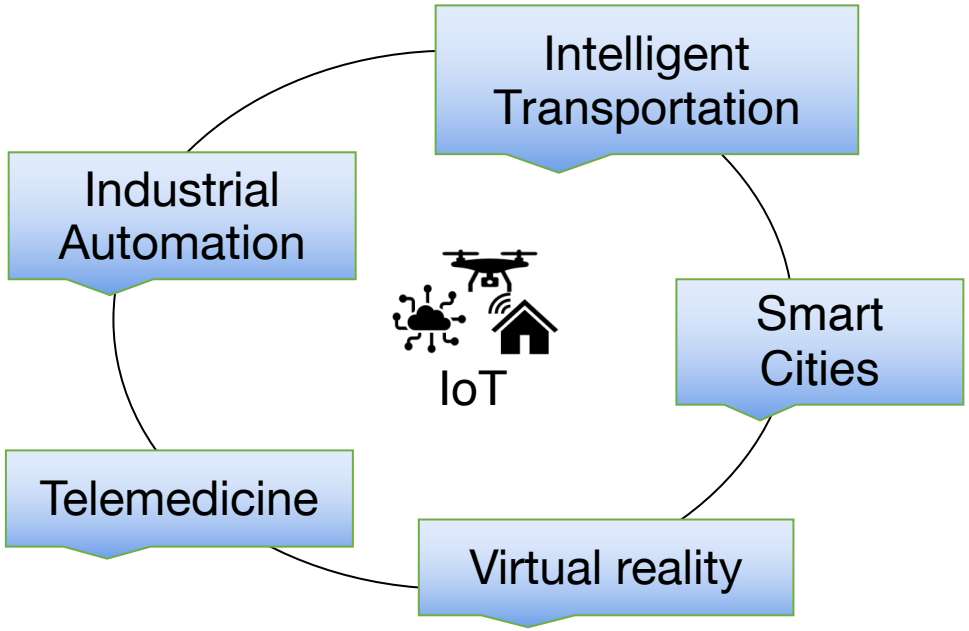
BITS Pilani, India[2]

9th May 2023

Technical Session 1.2 - SDN

IEEE/IFIP NOMS 2023

Internet of Things (IoT) security

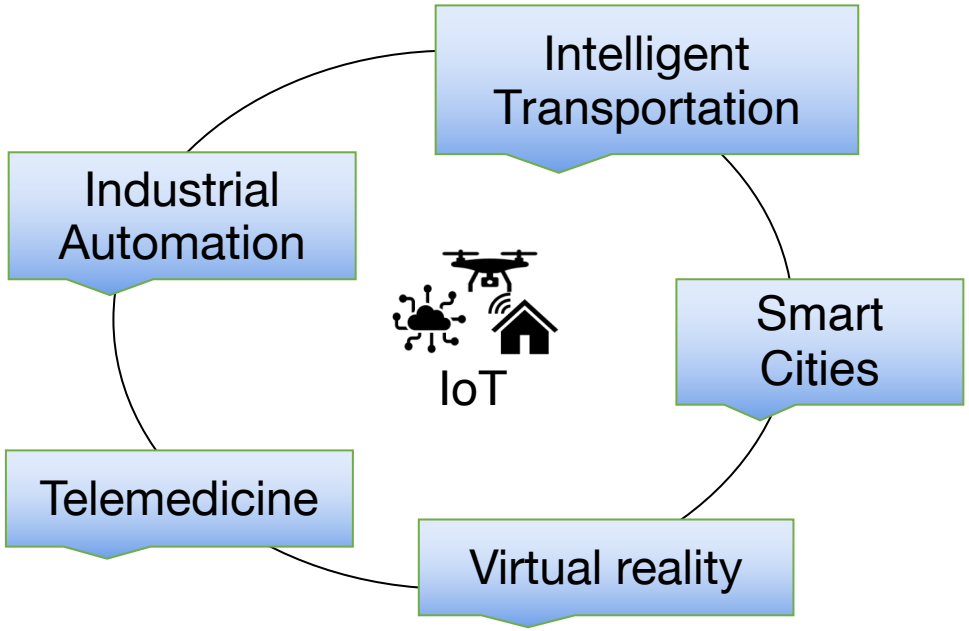


Challenges:

- **Security**
  Vulnerable to spoofing, tampering attacks
- **Resource constraints**
  Compute and energy
- **Ultra-low Latency**
  Many IoT applications require ultra-low latency (**~1 msec**)

- J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, “Security analysis of drones systems: Attacks, limitations, and recommendations,” Internet of Things, vol. 11, p. 100218, 2020.
- J.-Y. Lee, W.-C. Lin, and Y.-H. Huang, “A lightweight authentication protocol for internet of things,” in 2014 International Symposium on Next-Generation Electronics (ISNE). IEEE, 2014, pp. 1–2.
- M. A. Siddiqi, H. Yu, and J. Joung, “5g ultra-reliable low-latency communication implementation challenges and operational issues with iot devices,” Electronics, vol. 8, no. 9, p. 981, 2019. S. Gallenm¨uller, J. Naab, I. Adam, and G. Carle, “5g urllc:

Internet of Things (IoT) Security



Challenges:

- **Security**

Vulnerable to spoofing, tampering, and side channel attacks

- **Resource constraints**

Compute and energy

- **Ultra-low Latency**

Many IoT applications require ultra-low latency (**~1 msec**)

- **Authentication prevents unauthorised access**
- **Ensures only legitimate IoT device gains access to the network**

Existing cryptography-based authentication schemes

- **Authentication via encryption/decryption**

  Ex: Symmetric or Asymmetric key-based methods

Resource intensive

- **Secret keys stored on non-volatile memory (NVMs)**

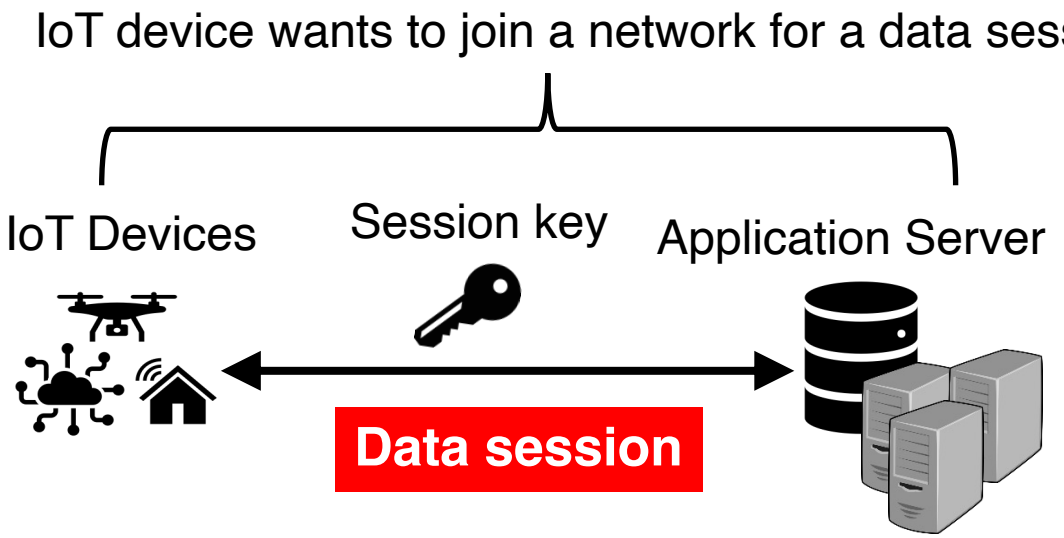
Micro probing

Side channel attacks

- A. Setyawan Sajim, “Open-source software-based sram-puf for secure data and key storage using off-the-shelf sram,” 2018.
- M. N. I. Khan and S. Ghosh, “Information leakage attacks on emerging non-volatile memory and countermeasures,” in Proceedings of the International Symposium on Low Power Electronics and Design, 2018, pp. 1–6.

Securing data session

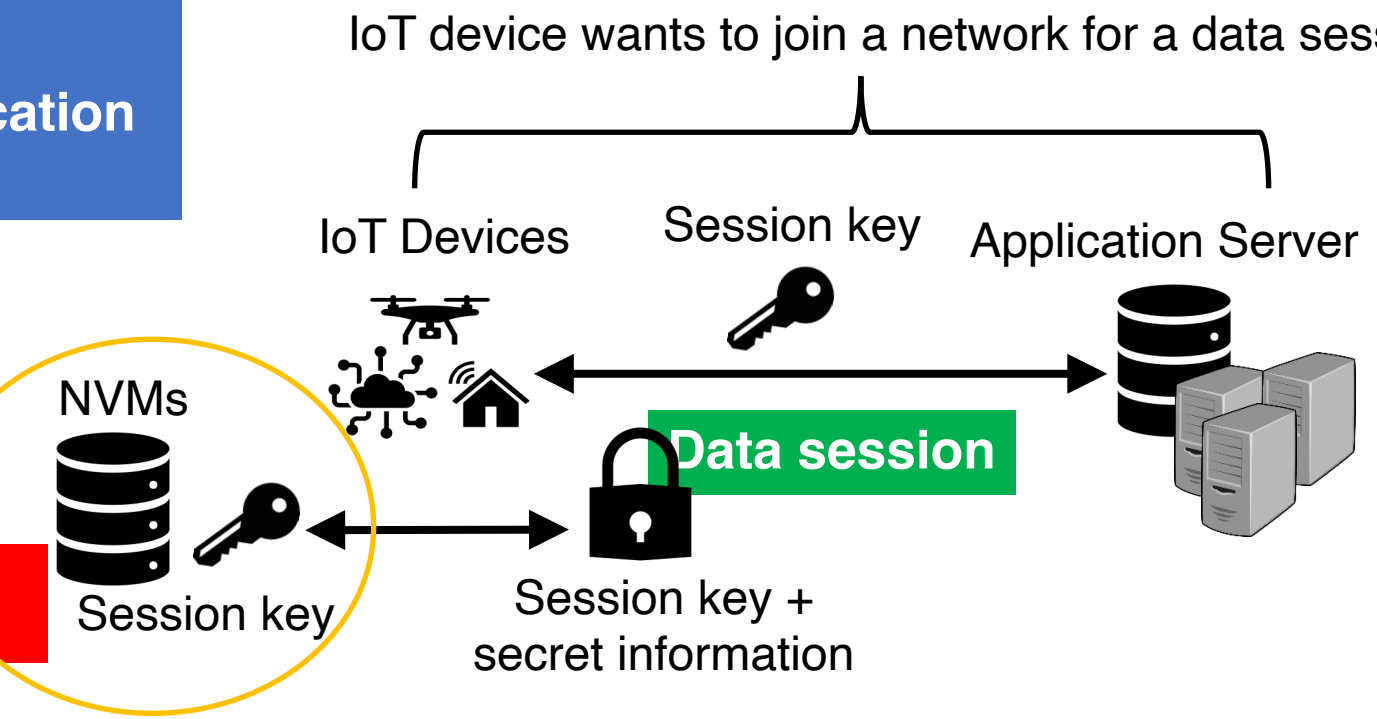
Step 1: Authentication



Step 2: Data session

Securing data session

Step 1: Authentication

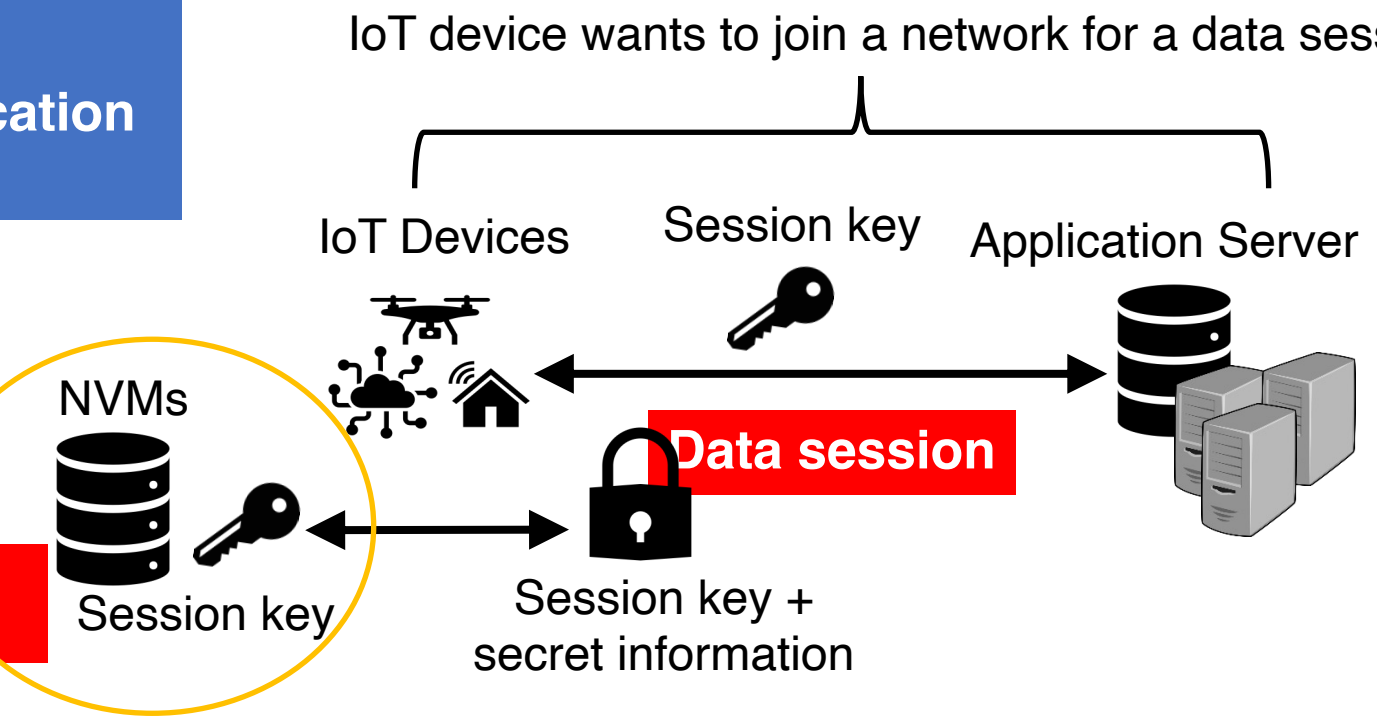


Micro probing

Step 2: Data session

Securing data session

Step 1: Authentication



Step 2: Data session

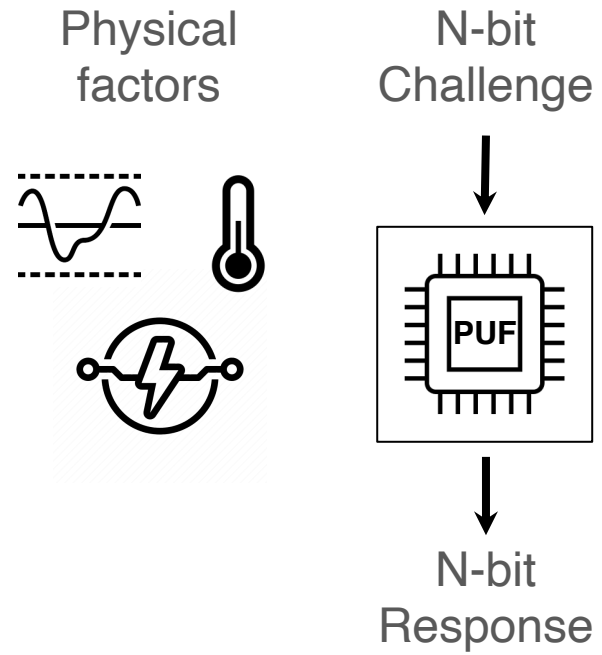
Micro probing

- **Can we generate key instead of storing**
- **Use key for secure data transfer**

- **Physical Unclonable Functions**

Physical Unclonable Functions (PUFs)

- Based on the **Challenge-Response (CR)** mechanism
- **Lightweight**
- **Unclonable**
  - Relies on inherent randomness while manufacturing
  - Two PUFs have different responses to the same challenge
- **Unpredictable:** Robust to ML attacks



Being lightweight, unclonable and unpredictable makes PUF-based security primitive a promising choice for IoT security

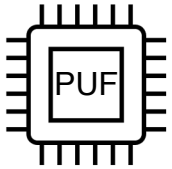
- B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, “Silicon physical random functions,” in Proceedings of the 9th ACM Conference on Computer and Communications Security, 2002, pp. 148–160.
- T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig, and R. J. Young, “A puf taxonomy,” Applied Physics Reviews, vol. 6, no. 1, p. 011303, 2019.

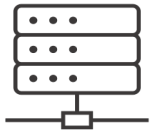
PUF-based authentication

PUF verifier at at an edge server

PUF verifier
authenticates IoT devices

IoT Devices with PUFs





Challenge-Response Pairs (CRPs) database

Enrolment phase

Challenge response pairs are generated at the IoT and securely stored at the PUF verifier DB

Authentication Phase

Request

Challenge C

Hash (R)

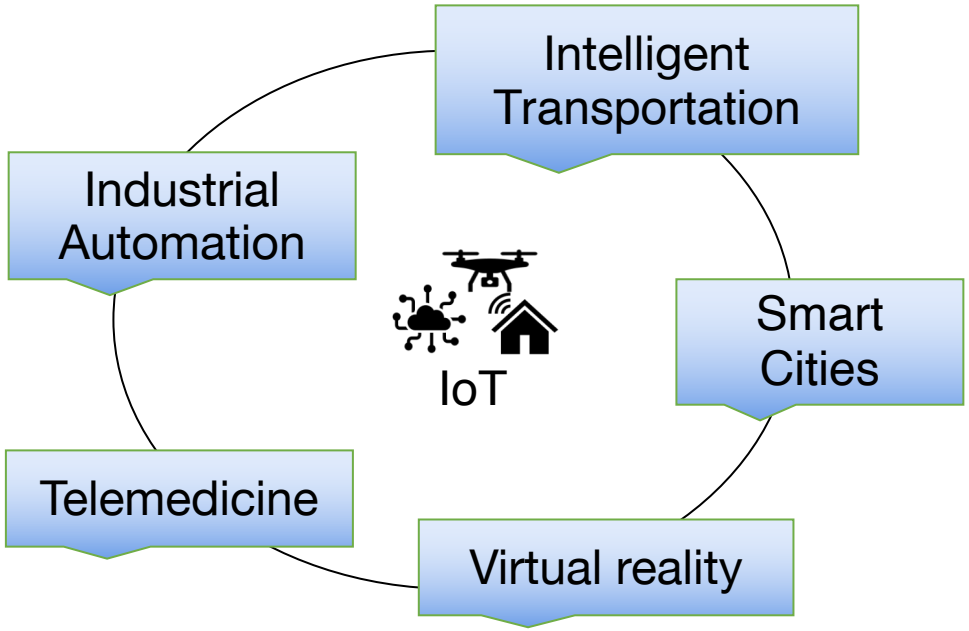
Verifies R against C

Compute response R on receiving C

Secure data transfer

Key = Response + secret information

Internet of Things (IoT) security



Challenges:

- **Security**

Vulnerable to spoofing, tampering attacks

- **Resource constraints**

Compute and energy

- **Ultra-low Latency**

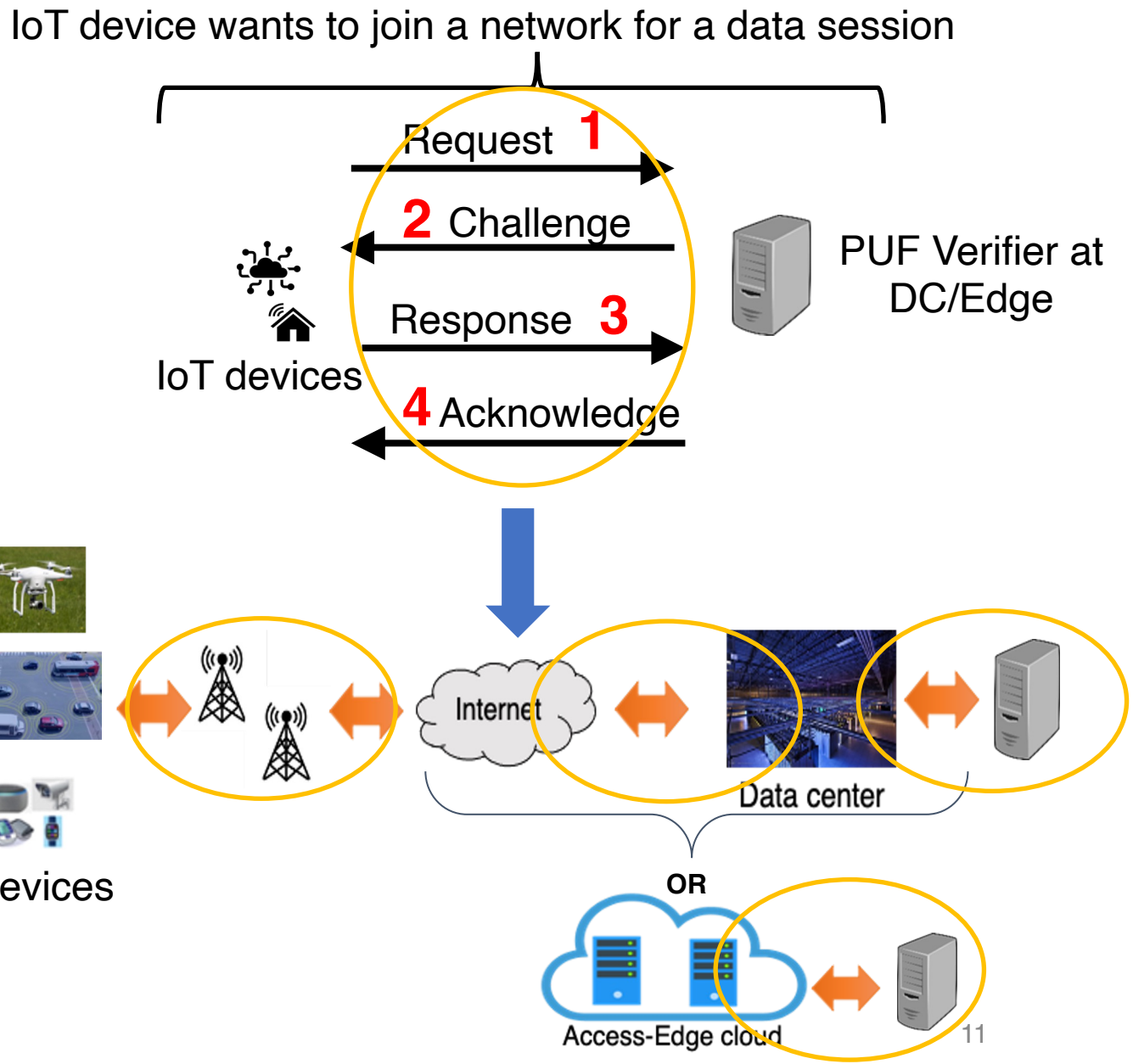
Many IoT applications require ultra-low latency (**~1 msec**)

PUF-based Authentication is robust to security attacks and resource friendly

Issues with the existing authentication protocols

- **End-to-end completion time**

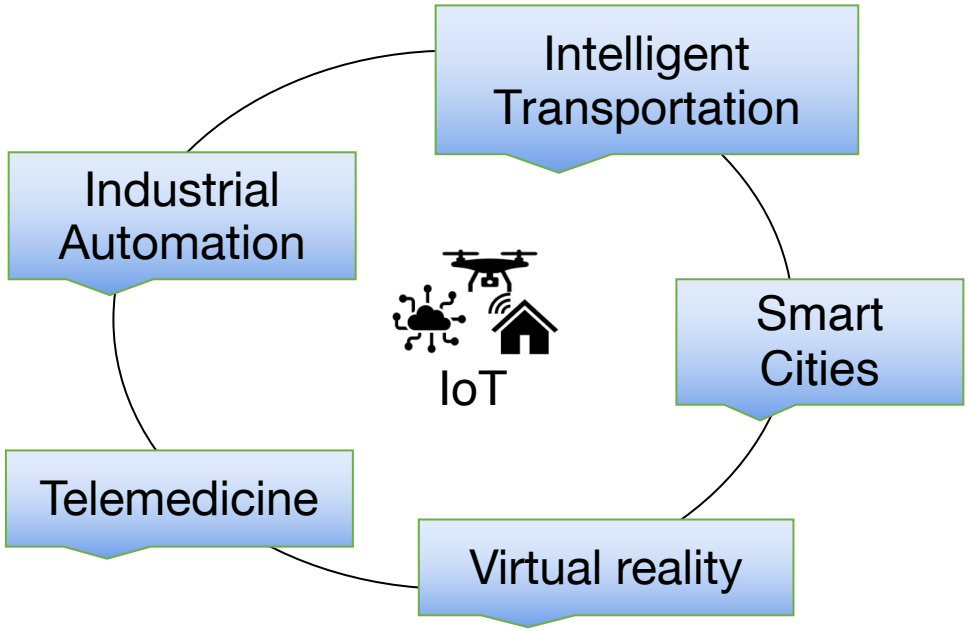
  It takes more time to complete all four steps



- **Transmission Delays**
  - Multiple RTTs
  - Multiple Hops

- **Hop-level Delays**
  - Packet copies
  - Scheduling at DC/Edge server
  - I/O interrupts

Internet of Things (IoT) Security



Challenges:

- **Security**

Vulnerable to spoofing, tampering attacks

- **Resource constraints**

Compute and energy

- **Ultra-low Latency (URLLC)**

Many IoT applications require ultra-low latency (**~1 msec**) as well as reliability

- Time budget to finish a transaction is < **1 millisecond** [1]
- Enabling PUF-based Authentication for URLLC applications is challenging due to multiple RTTs

- M. A. Siddiqi, H. Yu, and J. Joung, "5g ultra-reliable low-latency communication implementation challenges and operational issues with iot devices," Electronics, vol. 8, no. 9, p. 981, 2019. S. Gallenm¨uller, J. Naab, I. Adam, and G. Carle, "5g urllc

Research question ?

How to build **a secure and fast** PUF-based authentication system?

Assumptions

IoT devices connected to Mobile access Edge Compute (MEC)

Private edge setup

URLLC applications

Transaction-level authentication

Authentication before data transmission

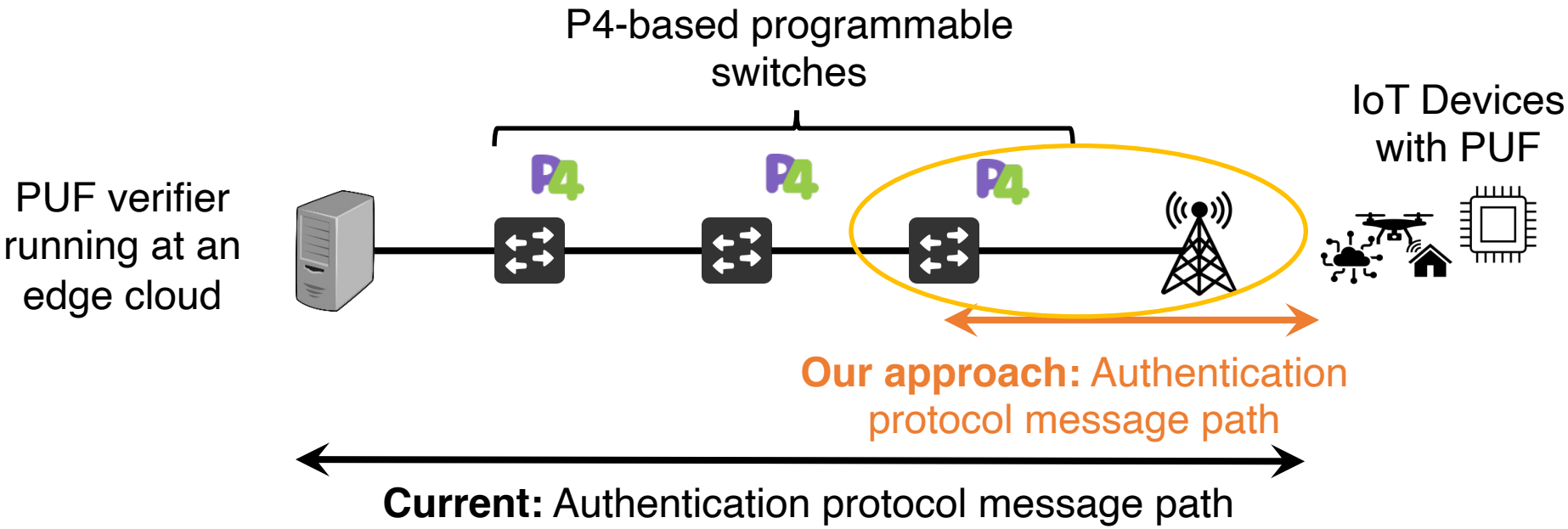
Objectives

Reduce authentication time

Scalable

Secure

Existing works and our approach

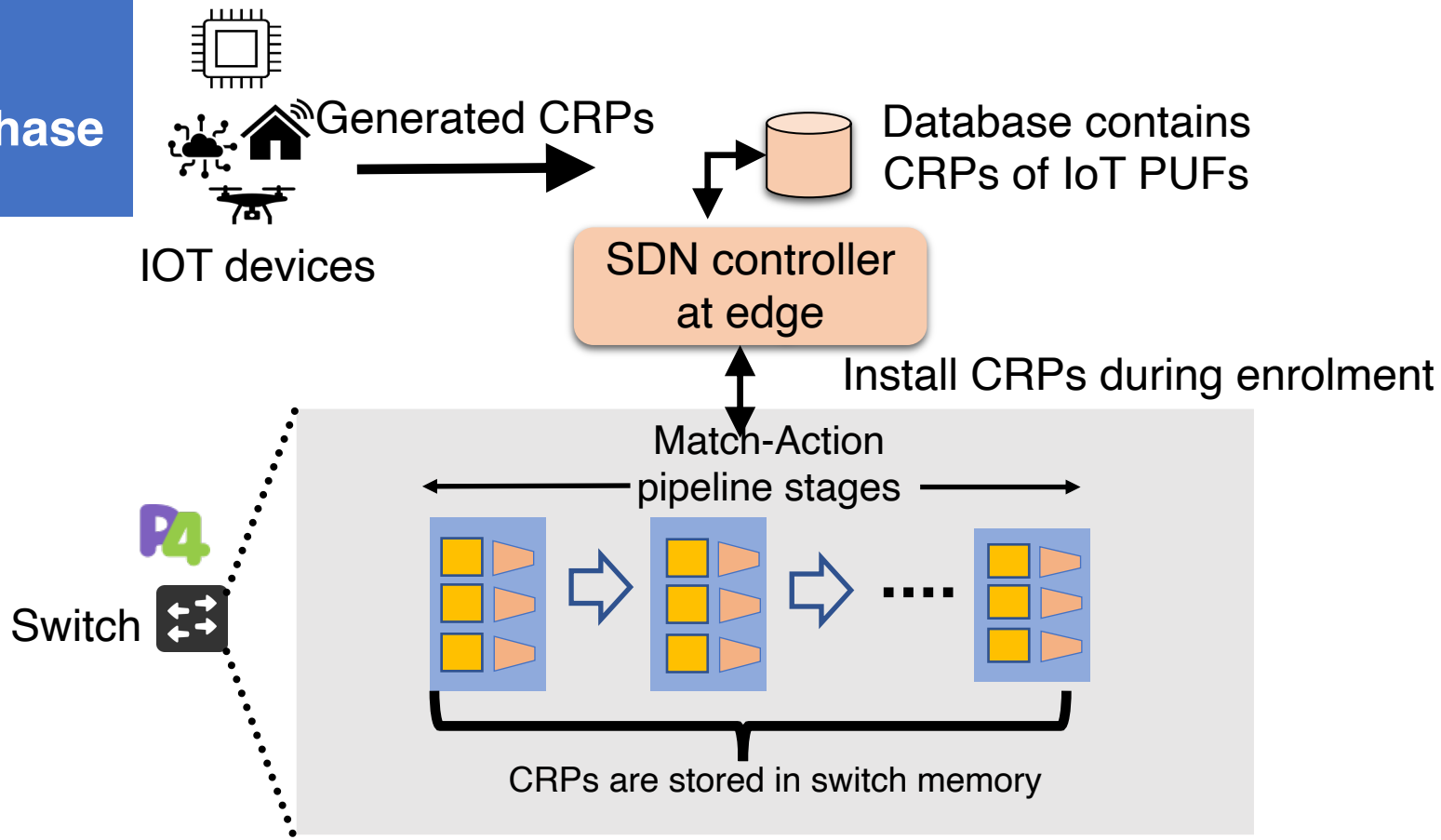


- Existing **works** lack comments on
  - Performance parameters like **latency** and **throughput**
  - Feasibility in **5G** and **edge computing environments**

We focus on URLLC applications using edge computing

Protocol workflow (EP and AP)

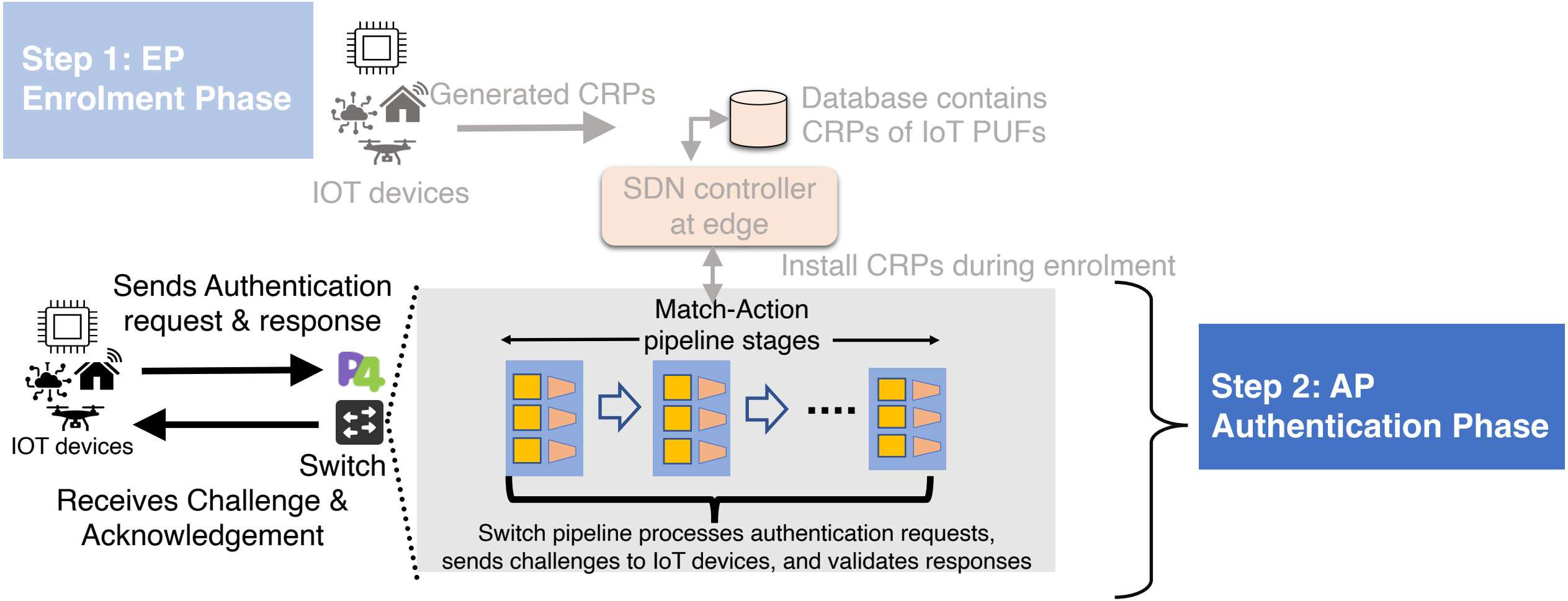
Step 1: Enrolment Phase (EP)



Enrolment is assumed to happen in a secure offline manner

Protocol workflow (EP and AP)

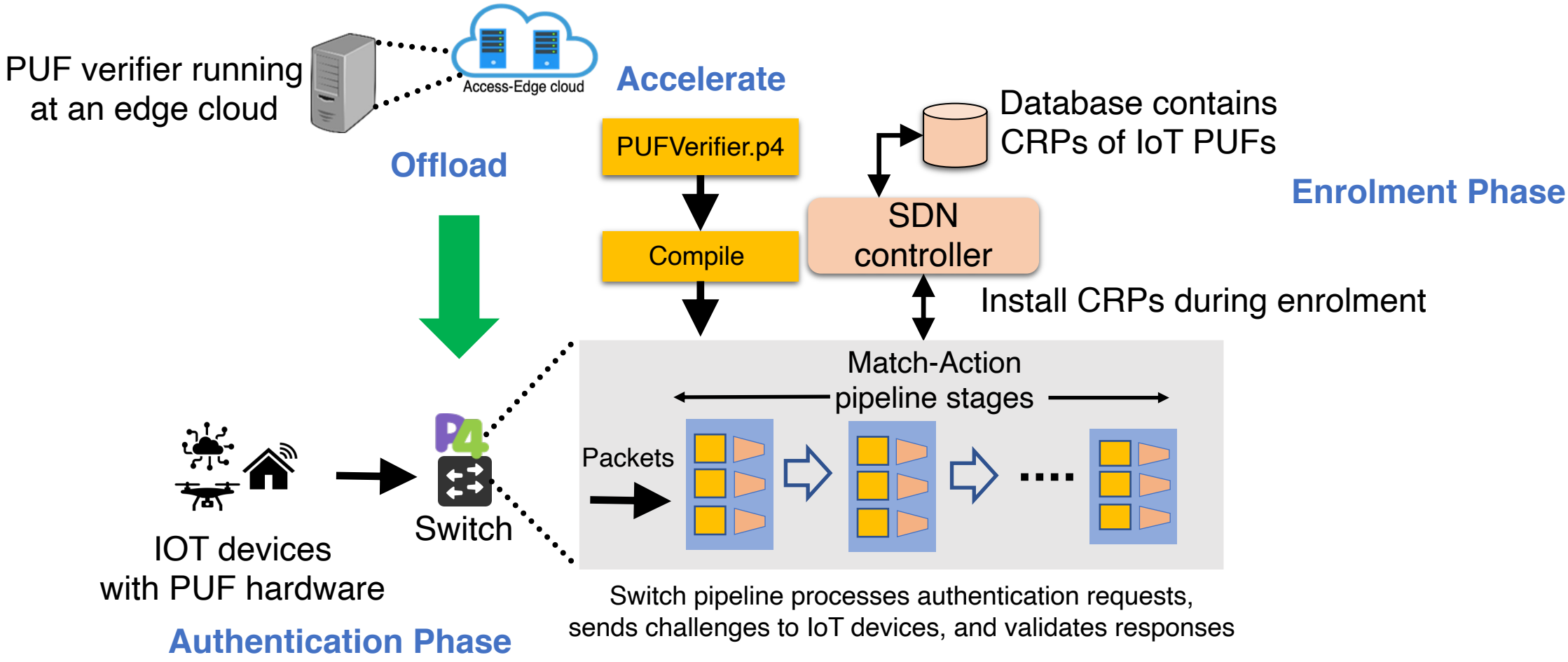
Step 1: EP Enrolment Phase



Step 2: AP Authentication Phase

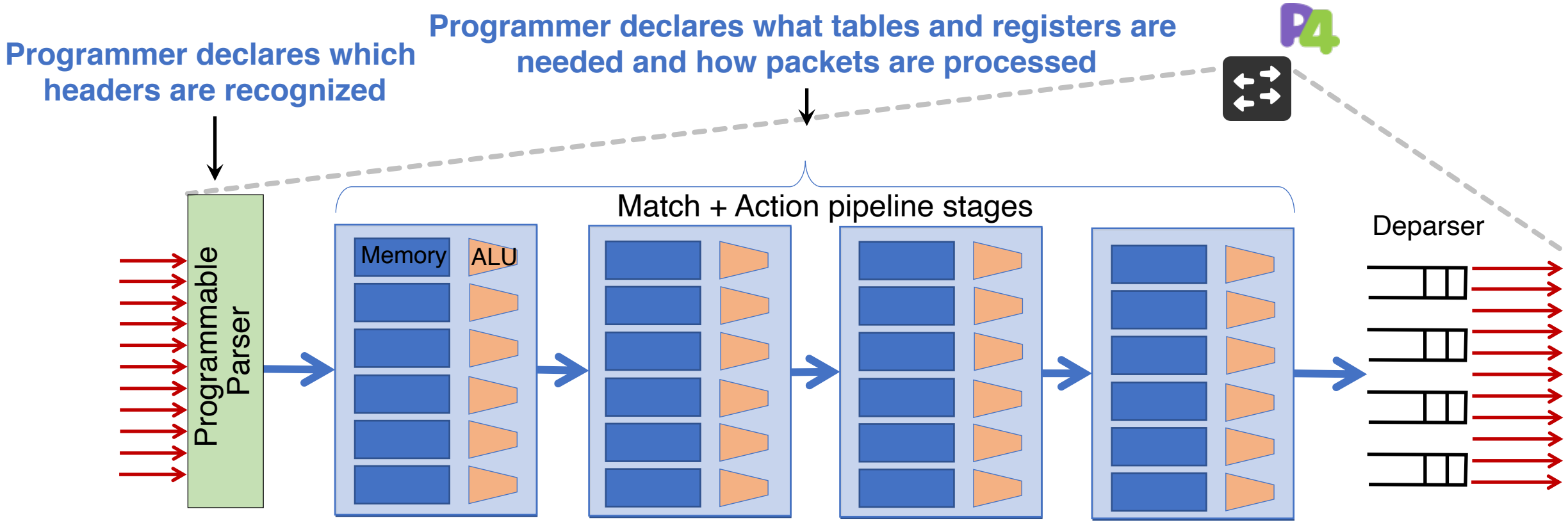
EP and AP happen when IoT device wants to join a network

This paper: Offloads PUF-based authentication protocol to P4-based Tofino switch



Background: High speed programmable switch

- Line speed packet processing at Tbps
- Programmable using P4 language



- https://opennetworking.org/
- https://github.com/p4lang/tutorials

Technical challenges imposed by P4-based switches

⚠ **Limited switch memory**

- Switch memory (SRAM) only in the order of 100 MBs
- How many IoT devices can we support?

Scalable

⚠ **Packets must be processed at line speed**

- Limited number of operations
- Reduced [domain-specific] instruction set

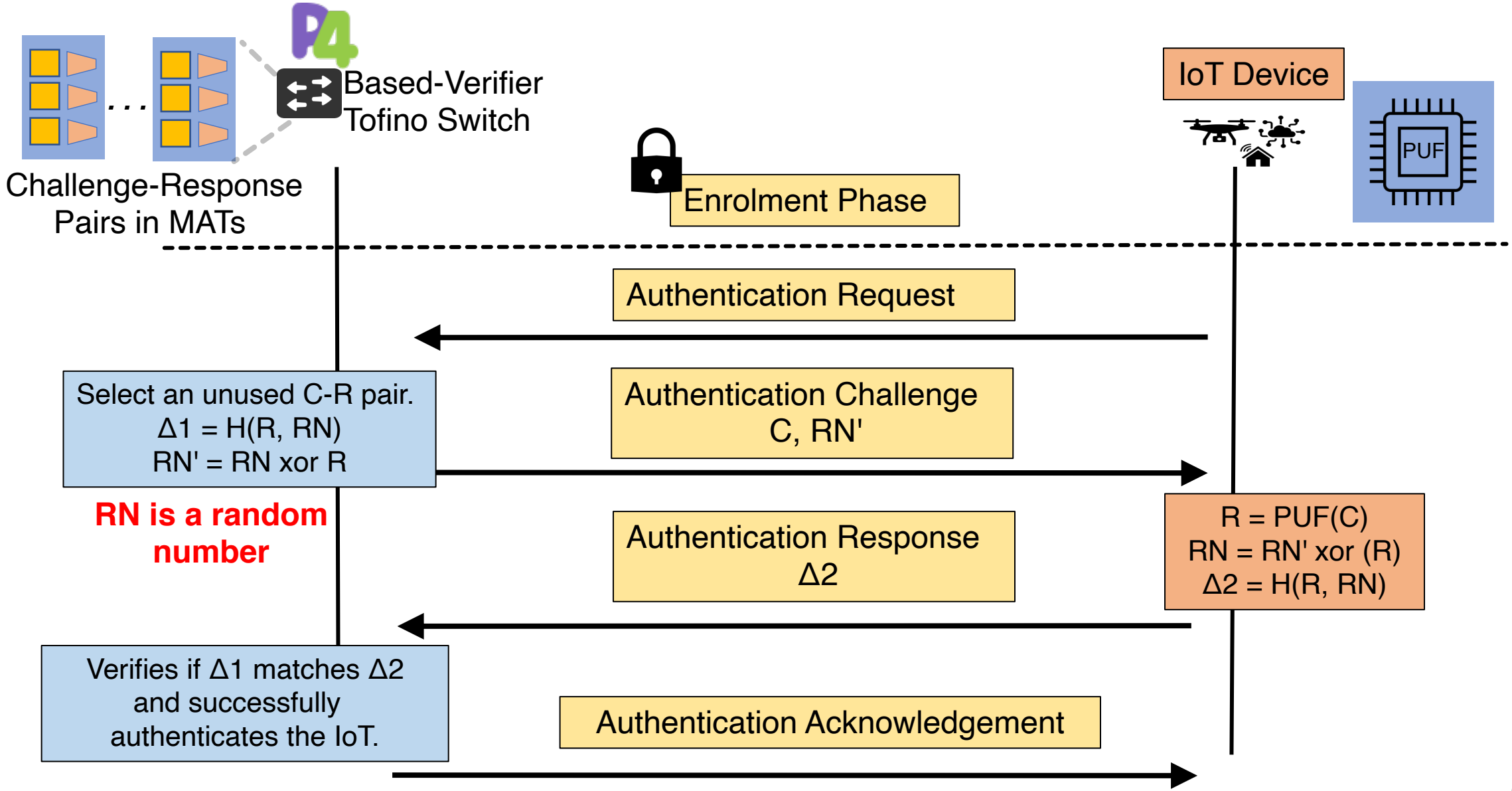
Reduces authentication time

Key contributions

Our work: Accelerating PUF-based authentication protocol on P4-based switch

- We address **technical challenges** imposed by P4-switches
- We implement a PUF-based authentication protocol **prototype**:
  - **Reduces authentication time**
  - **Scalable**
  - **Secure**
- We evaluate the prototype running on an **actual switch**

PUF-based authentication protocol on a P4-based switch



- U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, “A pufbased secure communication protocol for iot,” ACM Transactions on Embedded Computing Systems (TECS), vol. 16, no. 3, pp. 1–25,

Key contributions

Our work: Accelerating PUF-based authentication protocol on P4-based switch

- We address **technical challenges** imposed by P4-switches

- We implement a PUF-based authentication protocol **prototype**:
  - **Reduce authentication time**
  - **Scalable**
  - **Secure**

- We evaluate the prototype running on an **actual switch**

Addressing technical challenges

Limited switch memory

- Switch memory (SRAM) only in the order of 100 MBs
- How many IoT devices can we support?

Packets must be processed at line speed

- Li
- Re ction set

- **Exploiting per-stage SRAM**
- **Scalable**

Solution

MAT optimization

- Place per-IoT CRPs across different MATs
- Multi-stage MATs

Addressing technical challenges

- **P4 code fits in a sequential switch pipeline**
- **We could achieve ultra-low latency requirement**

Packets must be processed at line speed

- Limited number of operations
- Reduced [domain-specific] instruction set

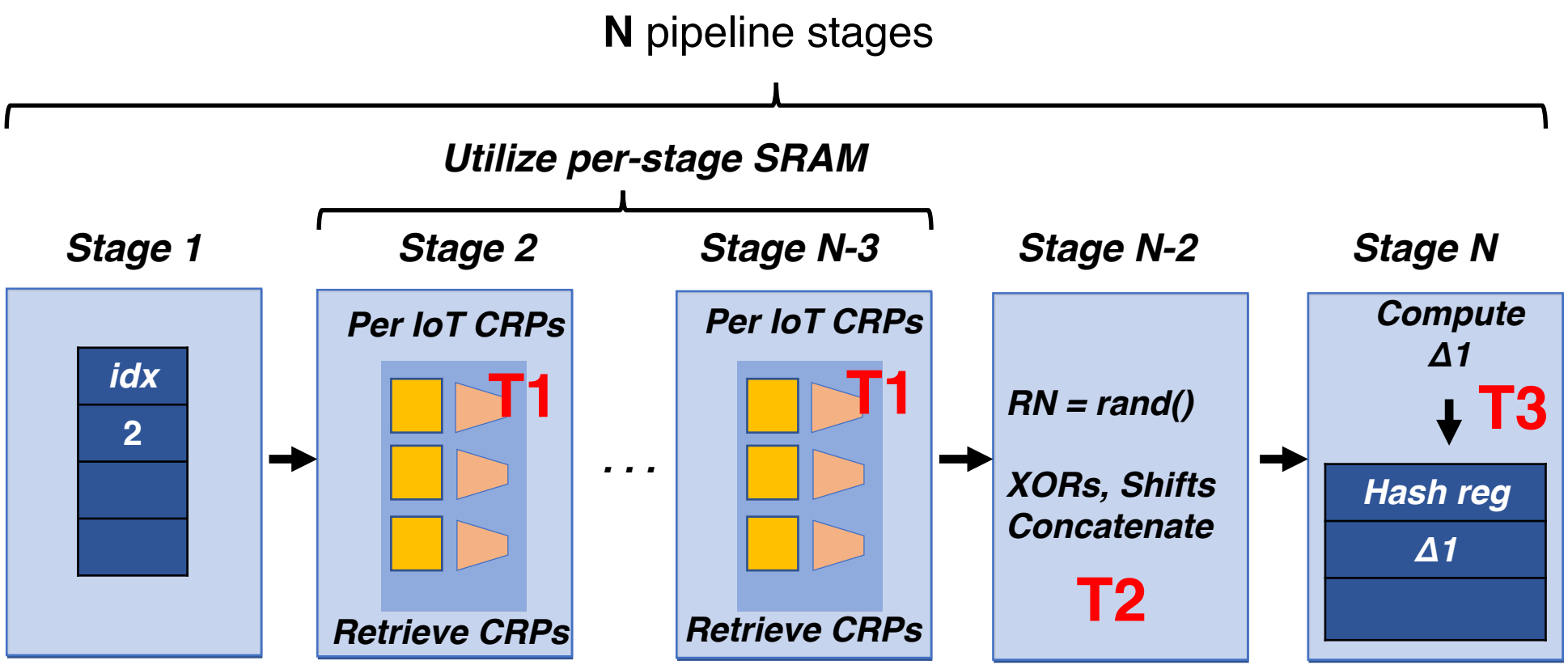
Solution

Identify a dependent set of tasks to be executed sequentially

- **T1:** Retrieve unused CRPs
- **T2:** Generate random numbers and perform XORs, shifts and concatenation
- **T3:** Hash and store in stateful register to be compared during response processing

Addressing technical challenges

MAT Optimization + Data Dependency



Scalable

Reduces authentication time

Key contributions

Our work: Accelerating PUF-based authentication protocol on programmable switch

- We address **technical challenges** imposed by P4-switches

- We implement a PUF-based authentication protocol **prototype**:
  - **Reduce authentication time**
  - **Scalable**
  - **Secure**

- We evaluate the prototype running on an **actual switch**

Custom Authentication Header

Authentication request message

msgType	Unused	Unused	Unused	Unused
1 byte				

Authentication challenge message

msgType	challenge	rndNumber	Unused	switchTime
1 byte	4 bytes	6 bytes		4 bytes

Authentication response message

msgType	Unused	Unused	Hash	Unused
1 byte			4 bytes	

Authentication acknowledgement message

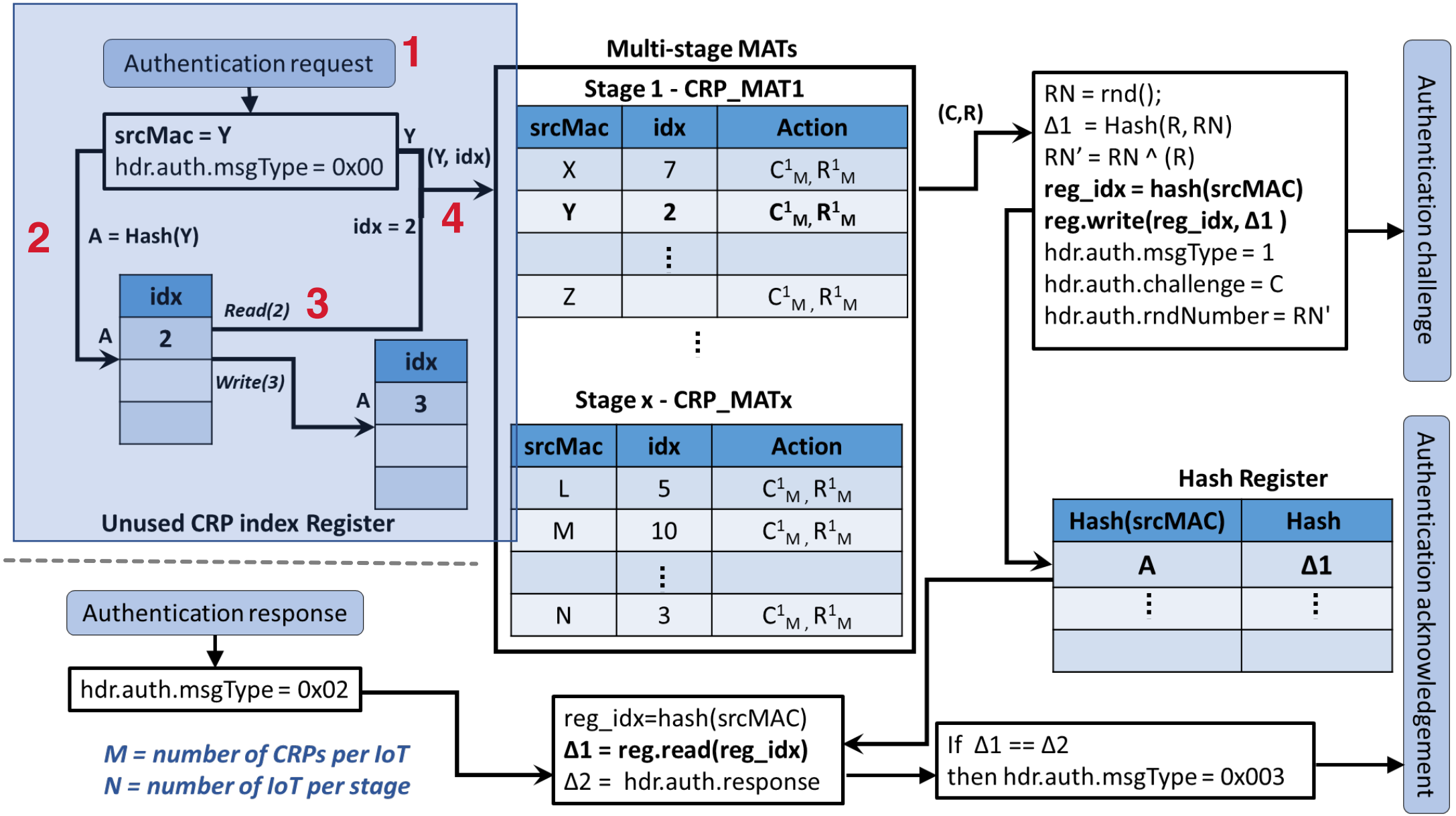
msgType	Unused	Unused	Unused	switchTime
1 byte				4 bytes

```
header auth_h
{
   bit<8> msgType;
   bit<32> challenge;
   bit<64> rndNumber;
   bit<32> Hash;
   bit<32> switchTime;
}
```

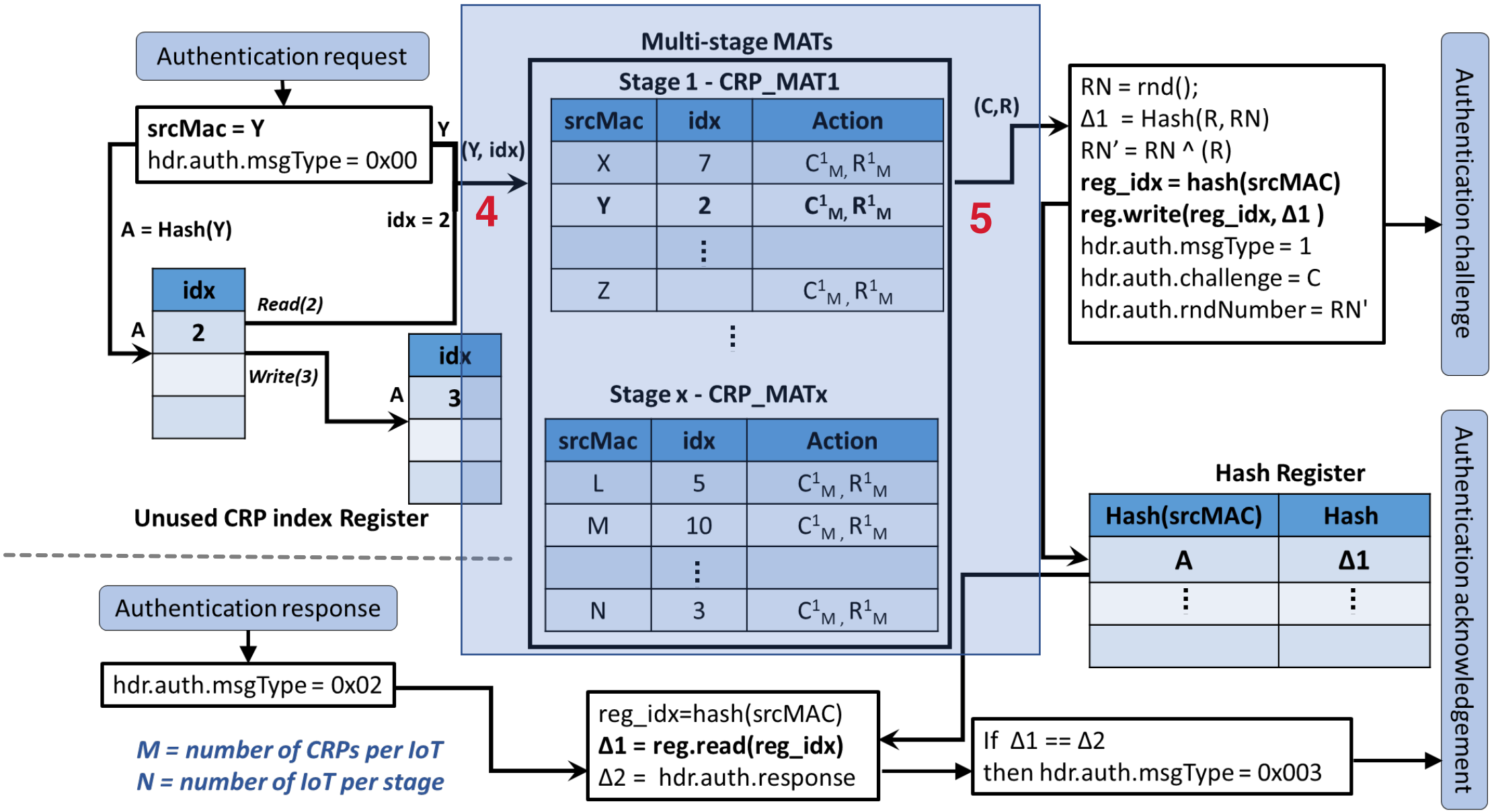
MsgType	Value
Request	0x00
Challenge	0x01
Response	0x02
Ack	0x03

Custom header defined to implement authentication phase

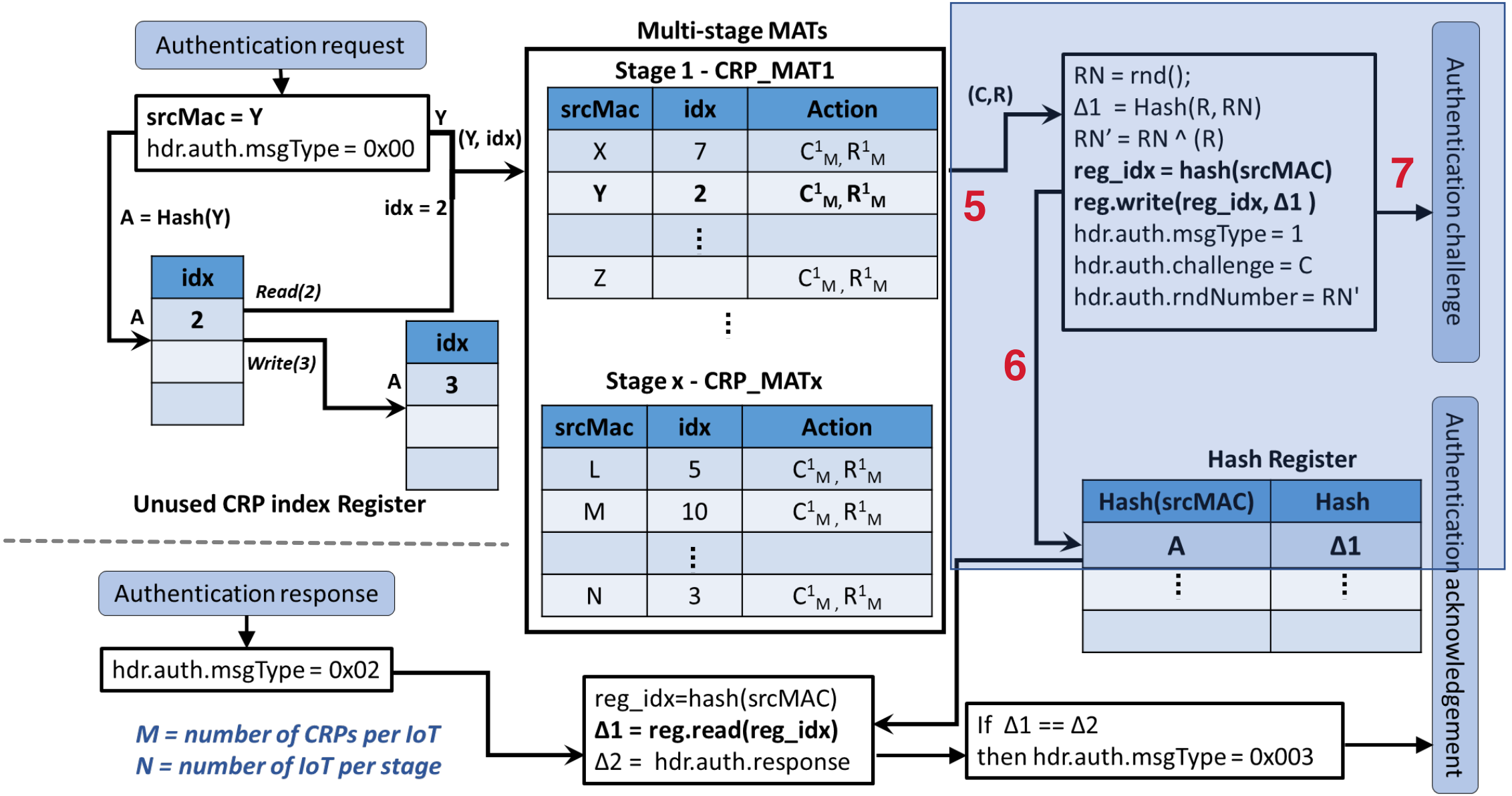
Implementation



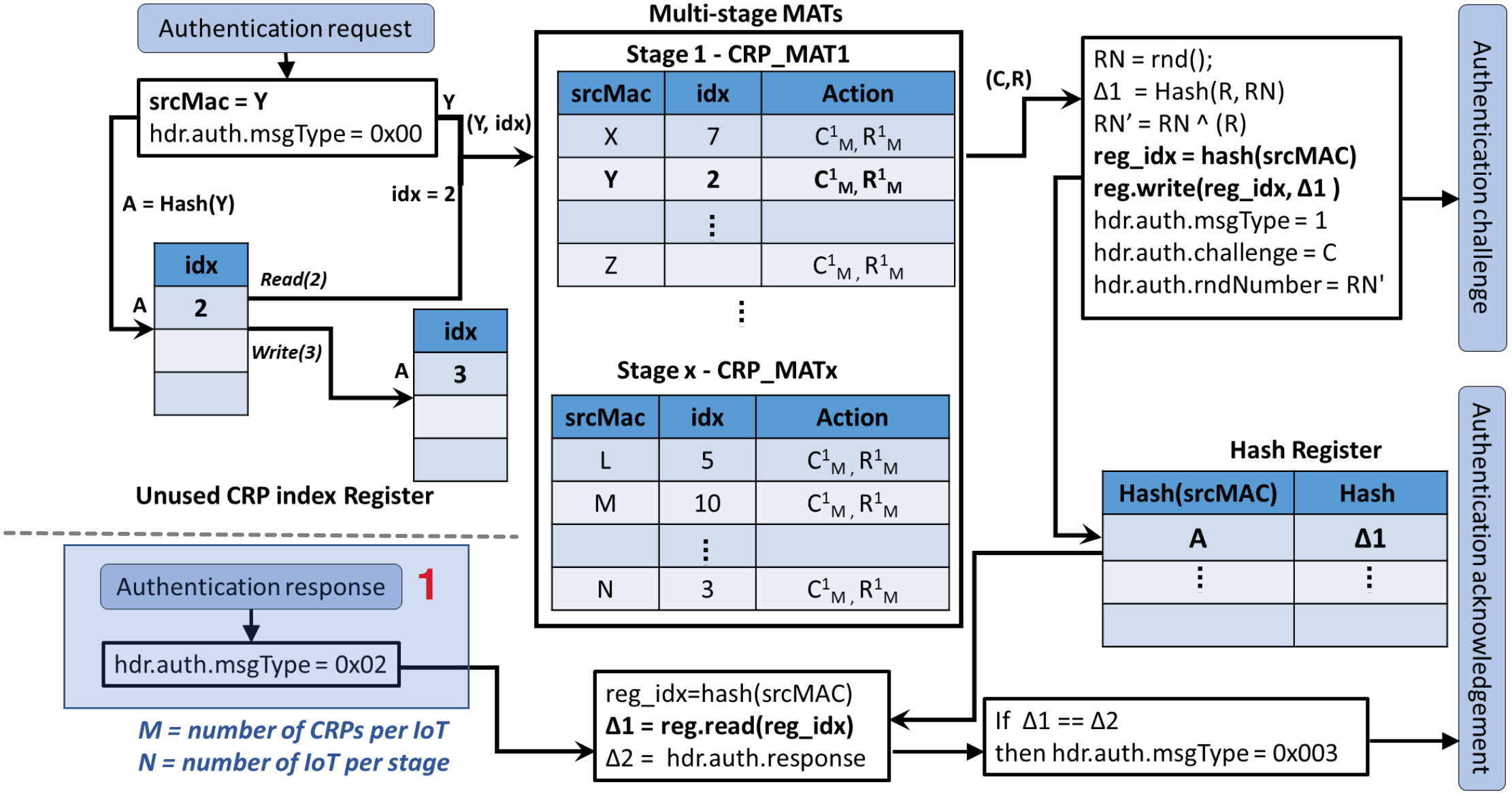
Implementation



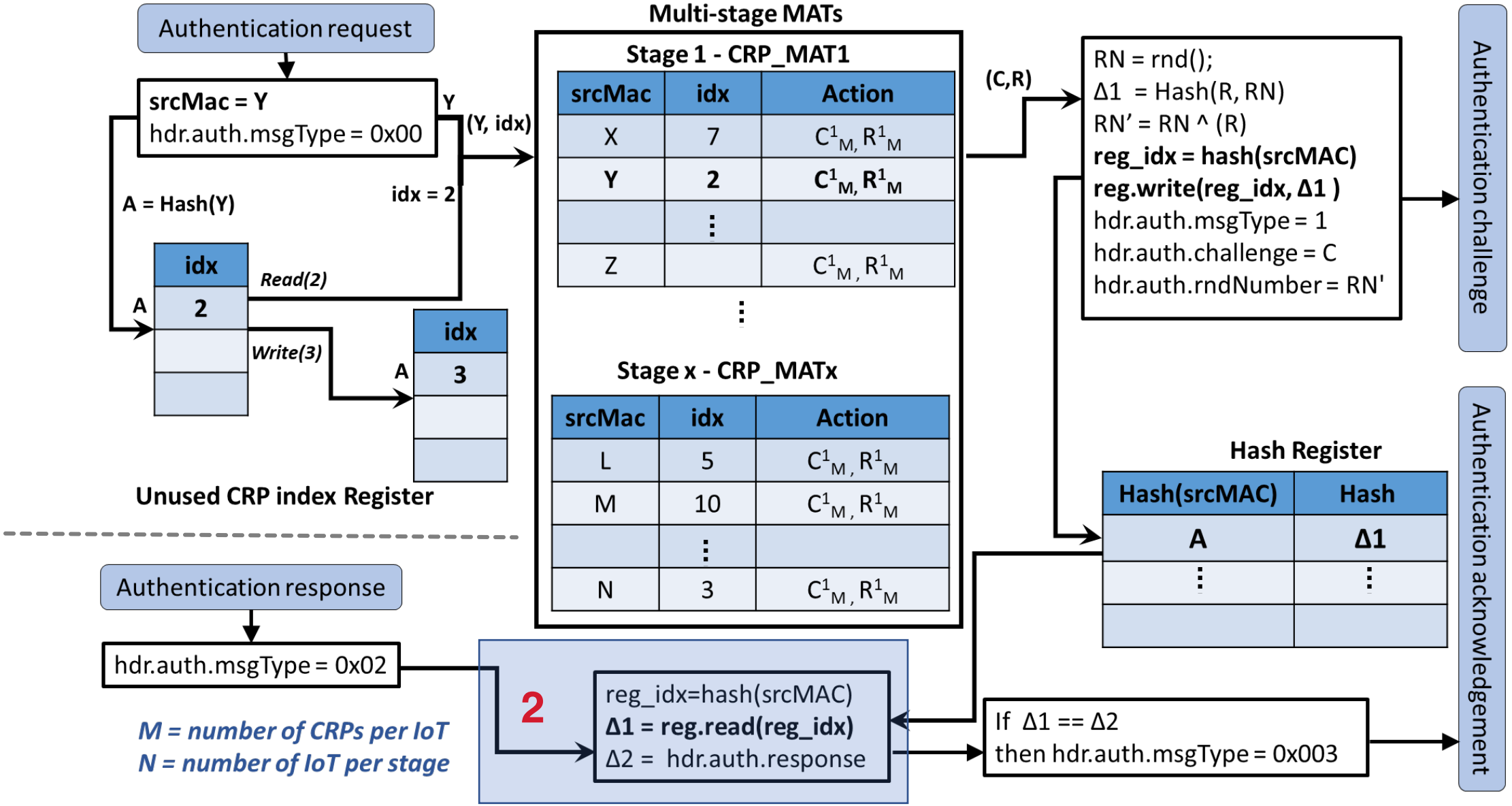
Implementation



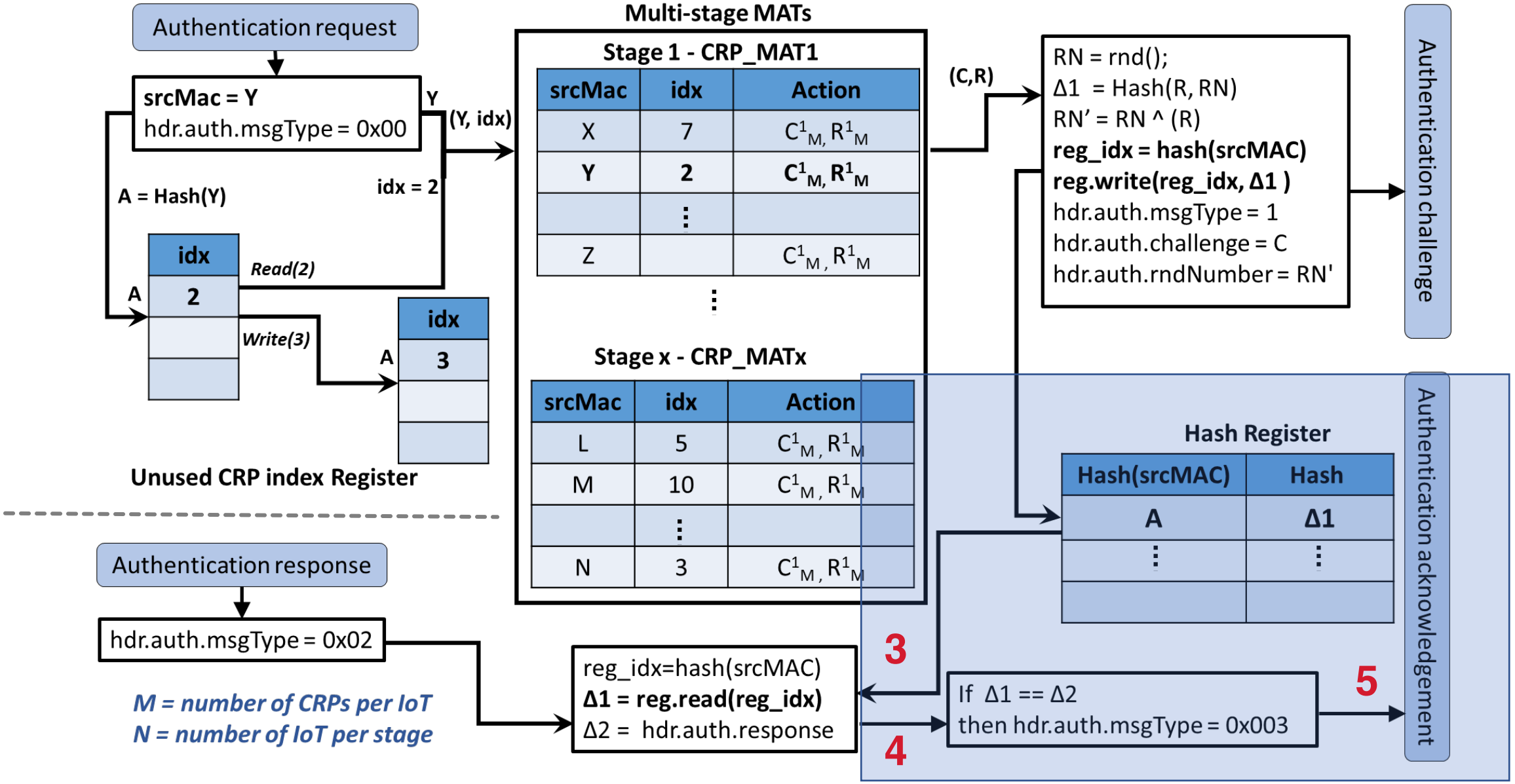
Implementation



Implementation



Implementation



Implementation

To study the offloading benefits, we implement PUF-Verifier logic on:

- **Intel Tofino switch** - Wedge100BF-32x Tofino
  - PUF-Verifier using P4-16

- **General Purpose x86 based CPU** - Intel(R) Core(TM) i9-7900X 10-core 3.3 GHz
  - PUF-Verifier using UDP socket program

IoT authentication messages are emulated using UDP socket program

Key contributions

Our work: Accelerating PUF-based authentication protocol on a programmable switch

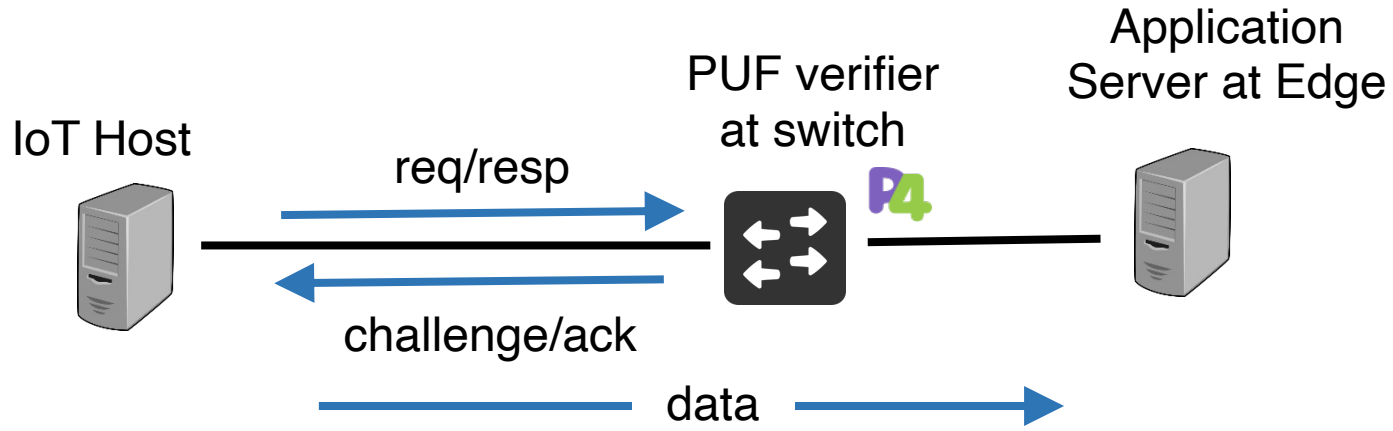
- We address **technical challenges** imposed by P4-switches

- We implement a PUF-based authentication protocol **prototype**:
  - **Reduce authentication time**
  - **Scalable**
  - **Secure**

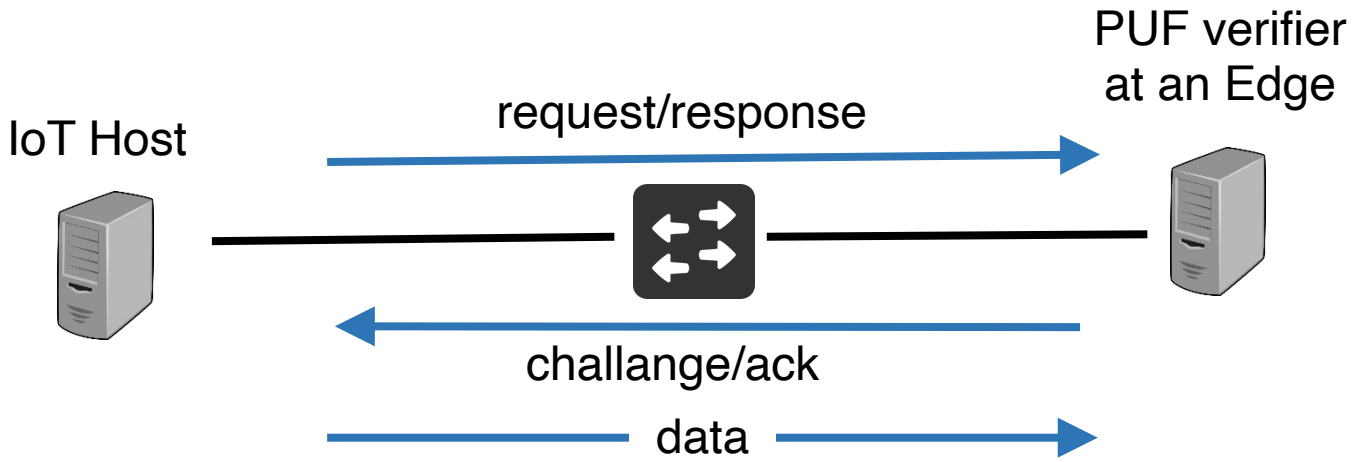
- We evaluate the prototype running on an **actual switch**

Experimental setup

- Host-Switch



- Host-Host



Evaluation

Metrics

1. Authentication completion time

- End-to-end authentication latency

2. Resource consumption

- SRAM usage

For more details on the metrics, do check out our paper…

Understand performance implications and resource consumption

Evaluation setup

- Host-Switch
- Host-Host

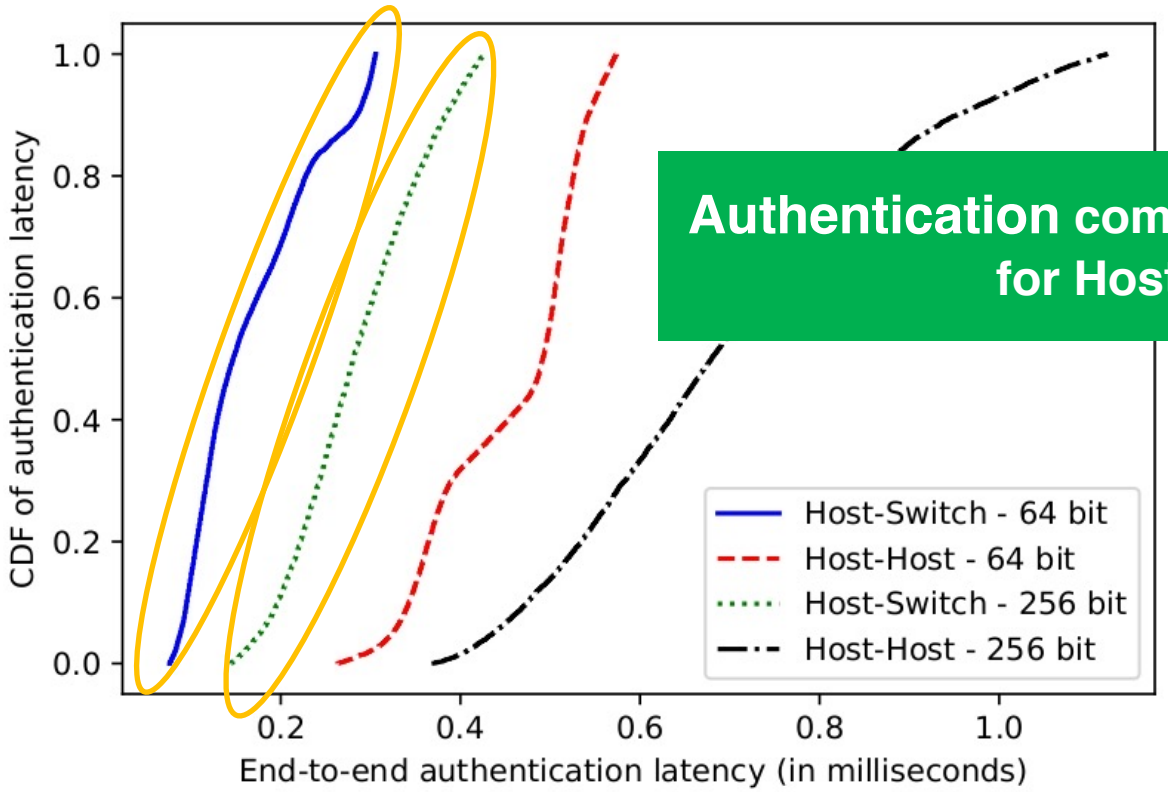
And for variable

PUF response sizes

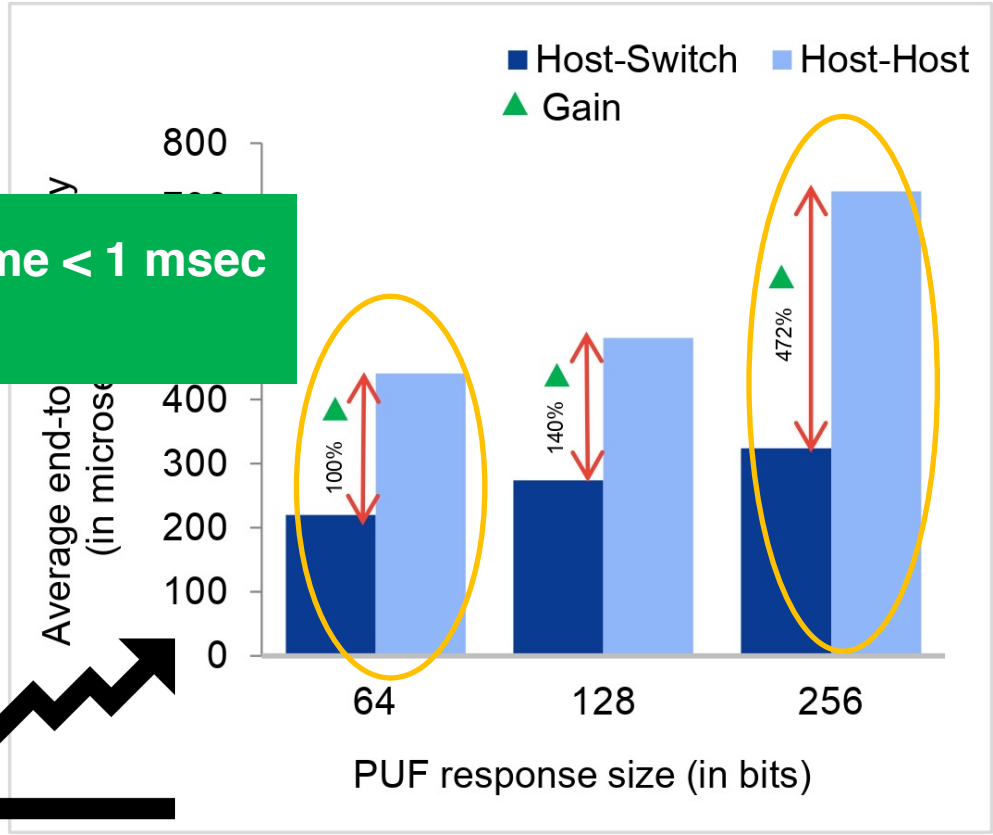
- 64-bits
- 128-bits
- 256-bits

End-to-end authentication latency

We sent 10K authentication requests to evaluate end-to-end authentication latency



Authentication completion time < 1 msec for Host-Switch



For Switch-based Verifier with PUF Response size:
(1) 64-bit -- 99% of the requests < 0.25 msec
(2) 256-bits -- 99% of the requests < 0.4 msec

Switch-based verifier significantly reduces the time taken to authenticate an IoT device

Resource (SRAM) consumption

Assume we wish to support 100 CRPs per IoT

Size of Response (in bits)	TNA1		TNA2	
		No. of IoT devices supported (per pipeline)		No. of IoT devices supported (per pipeline)
64	82	468K	82	884K
128	75.3	360K	75.3	680K
256	66	180K	66	340K

Scalable to hundreds of thousands of IoT devices

We can scale up or scale down the number of IoT devices supported by modifying the number of CRPs per IoT device

Summary

- Need for Accelerating PUF-based Authentication Protocols
- Offloaded the PUF verifier to the Tofino switch
- Average authentication latency improvement for switch-based verifier by 100-472%
- Scales up to hundreds of thousands of IoT devices

For more details, do check out our paper

Our code is available!
https://github.com/networked-systems-iith/PUFAuth

Thank You!