PUF-Based Two-Factor Authentication Protocol for Securing the Power Grid Against Insider Threat

KEVIN HUTTO*, SHUVA PAUL*, BENJAMIN NEWBERG*, VISHNU BOYAPATI^, YATHIENDRA VUNNAM*, SANTIAGO GRIJALVA*, AND VINCENT MOONEY*^
*SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
^SCHOOL OF COMPUTER SCIENCE
GEORGIA INSTITUTE OF TECHNOLOGY
ATLANTA, GEORGIA

Georgia Tech School of Electrical and Computer Engineering

Acknowledgement

- This work has been partially supported by the U.S. Department of Energy's Office of Cybersecurity, Energy Security, and Emergency Response (CESER) under Cybersecurity for Energy Delivery Systems (CEDS) Agreement Number DE-CR0000004 to the Georgia Tech Research Corporation.

Contents

I. **Problem Statement**

II. Standard Authentication

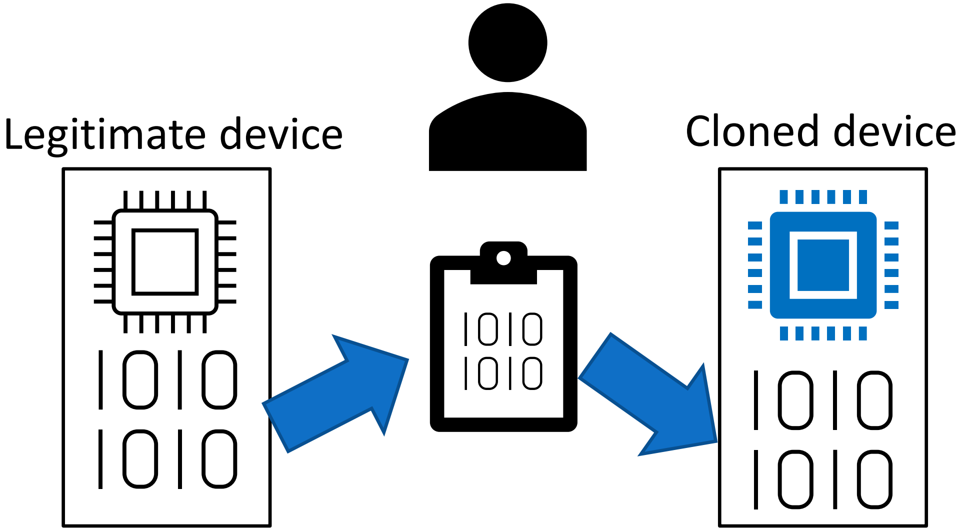
III. PUF Authentication

IV. Physical Implementation

V. Experiment and Results

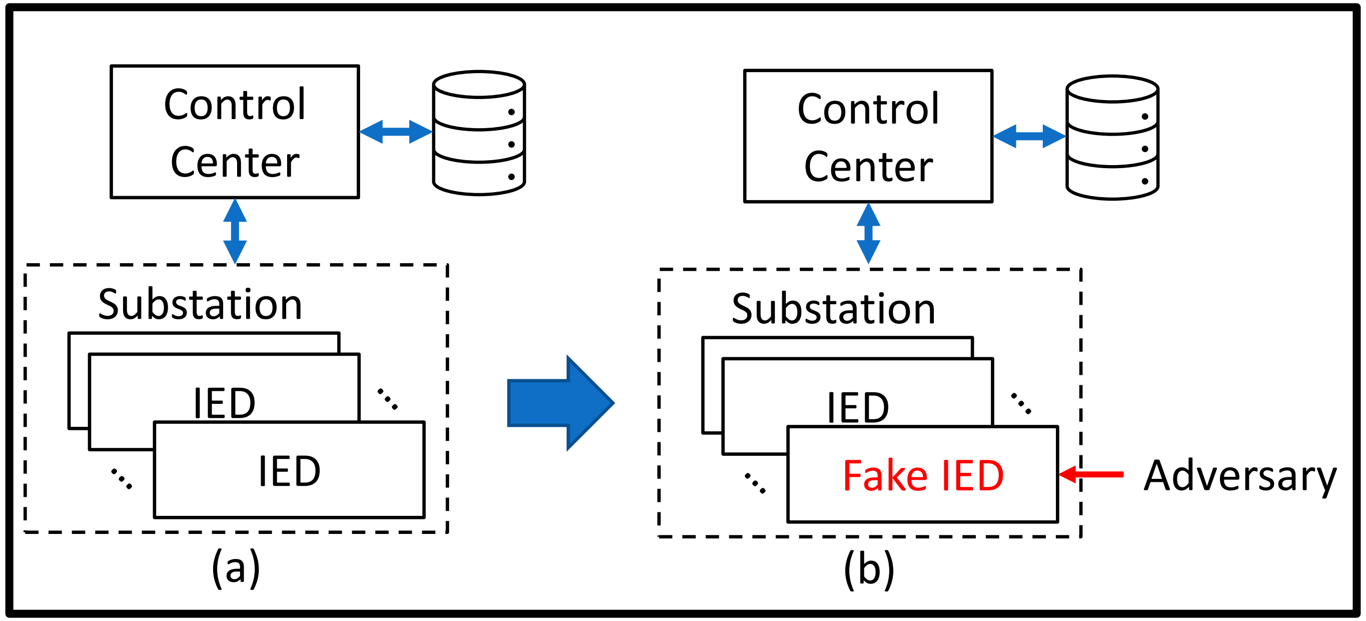
Problem Statement

- A substation may be physically vulnerable to a malicious lone wolf, low level insider
- Numerous IEDs and small devices are capable of having the software and memory contents cloned



Problem Statement

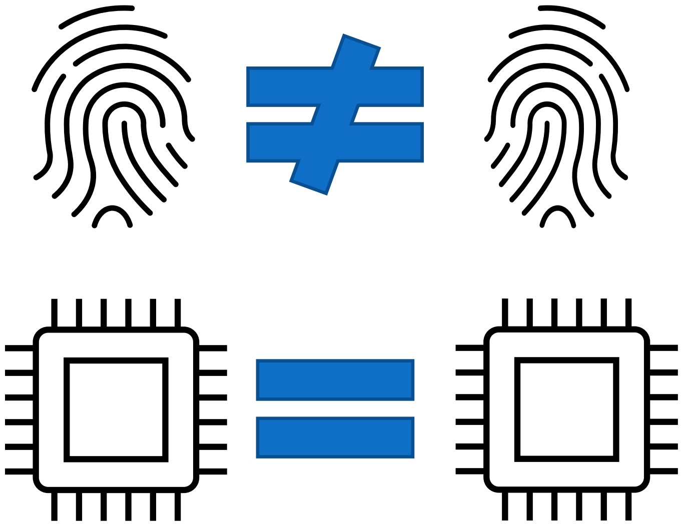
- A substation may be physically vulnerable to a malicious lone wolf, low level insider
- Numerous IEDs and small devices are capable of having the software and memory contents cloned
- An adversary may replace an IED with a fake device containing cryptographic keys cloned from the legitimate device
- The control center will determine the fake device as authentic due to the cloned cryptographic keys
- The fake device can now send erroneous data which could lead to power outages from unneeded protective actions through a false data injection attack [6]



Vulnerable Remote Substation

Problem Statement

- Utilize Physically Uncloneable Functions (PUFs) on each IED or power device to provide a second layer of authentication
  - A PUF is a hardware security primitive that utilizes tiny manufacturing variations, typically in silicon, to produce a unique digital fingerprint [10]
- We want to utilize inexpensive commercially available SRAM PUFs to provide a low overhead second form of authentication which cannot be cloned
  - An SRAM PUF obtains a source of inter-chip randomness via the power-on state of an SRAM



Identical microchip at the logic level, unique fingerprint from manufacturing

Contents

I. Problem Statement

II. Standard Authentication

III. PUF Authentication

IV. Physical Implementation

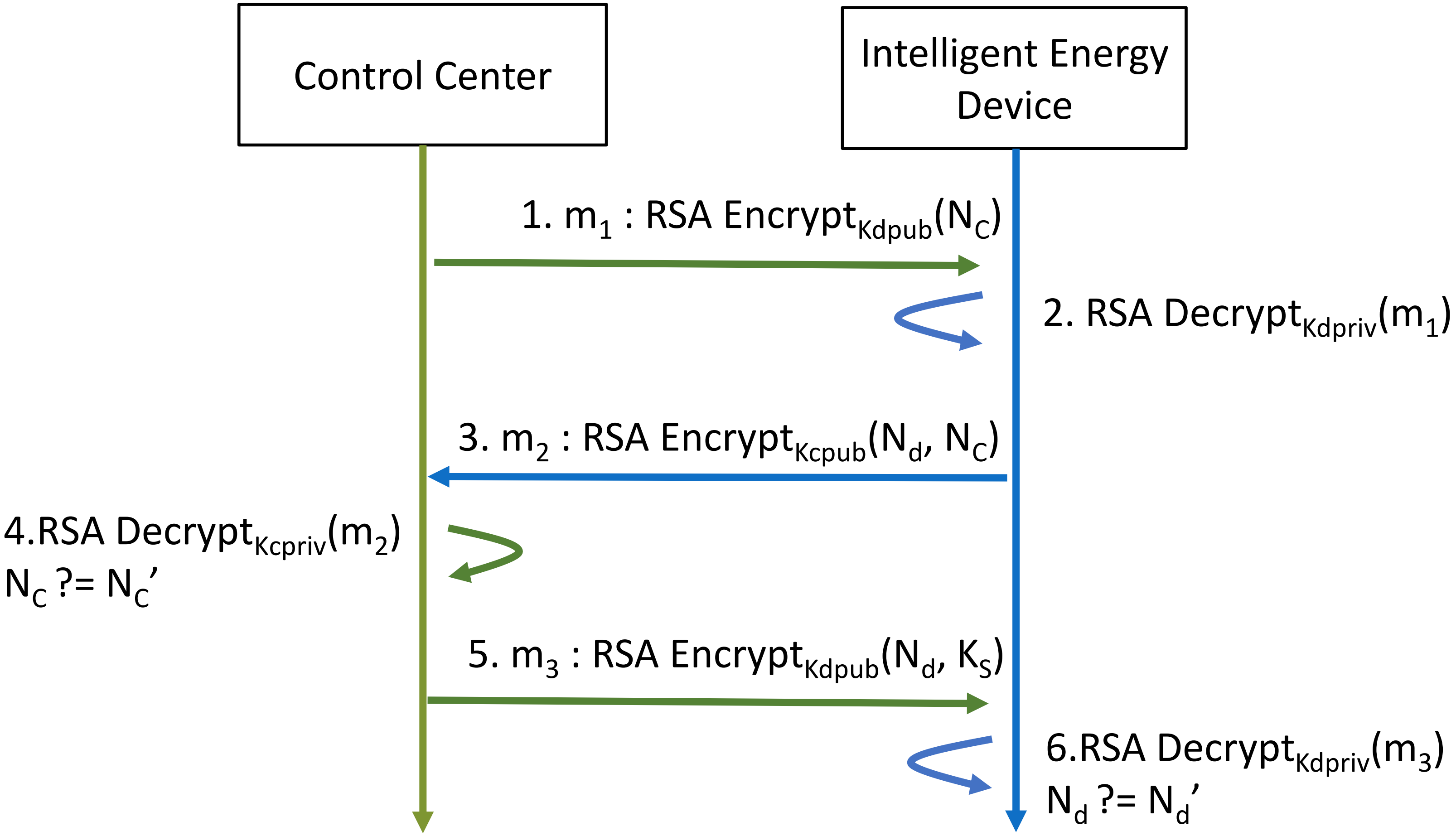
V. Experiment and Results

Standard Protocol

1) Control center sends to the IED a nonce N_c encrypted with the IED's RSA public key K_{dpub}. [7]

2) IED decrypts the nonce received from the control center using the IED's private key K_{dpriv} .

3) The IED sends back the control center's nonce N_c and a new nonce generated by the IED, N_d, back to the control center, encrypted with the control center's public key K_{cpub}.



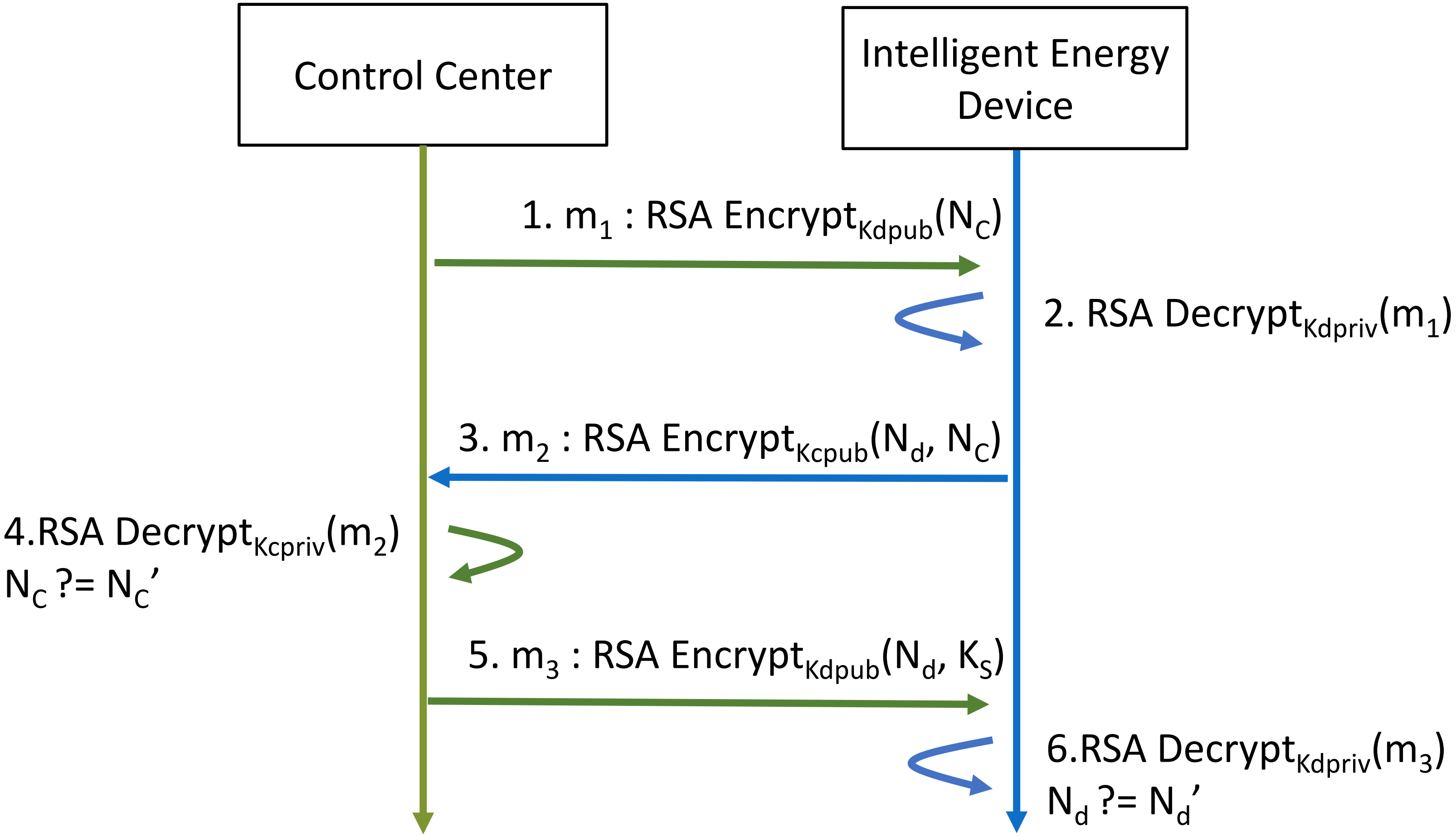
Control Center
Intelligent Energy Device
1. m_1 : RSA Encrypt$_{Kdpub}(N_C)$
2. RSA Decrypt$_{Kdpriv}(m_1)$
3. m_2 : RSA Encrypt$_{Kcpub}(N_d, N_C)$
4.RSA Decrypt$_{Kcpriv}(m_2)$
N_C ?= N_C'
5. m_3 : RSA Encrypt$_{Kdpub}(N_d, K_S)$
6.RSA Decrypt$_{Kdpriv}(m_3)$
N_d ?= N_d'

Standard Protocol

4) The control center decrypts the message using the control center's private key K_{cpriv} and verifies that the returned nonce N_c received matches what was sent.

5) Control center generates and sends an AES session key K_S, encrypted with the IED's public key K_{dpub} , to the device along with the nonce N_d generated by the device.

6) The IED once again decrypts using K_{dpriv} and verifies that the N_d received matches the one originally sent.



Control Center
Intelligent Energy Device
1. m_1 : RSA Encrypt$_{Kdpub}$(N_C)
2. RSA Decrypt$_{Kdpriv}$(m_1)
3. m_2 : RSA Encrypt$_{Kcpub}$(N_d, N_C)
4.RSA Decrypt$_{Kcpriv}$(m_2)
N_C ?= N_C'
5. m_3 : RSA Encrypt$_{Kdpub}$(N_d, K_S)
6.RSA Decrypt$_{Kdpriv}$(m_3)
N_d ?= N_d'

Contents

I. Problem Statement

II. Standard Authentication

III. PUF Authentication

IV. Physical Implementation

V. Experiment and Results

SRAM PUF

An SRAM PUF is a PUF that obtains a source of inter-chip randomness via the power-on state of an SRAM. [10]

An individual bit in an SRAM may settle as either a 0 bit or a 1 bit if no clear or reset signal is applied on power-up. [10]

The location and number for each of the individual 1 or 0 bits will vary for each physical SRAM depending on per-chip manufacturing variances.

We use an SRAM PUF which uses the random bits as a “root key” for cryptographic primitive [13][14]

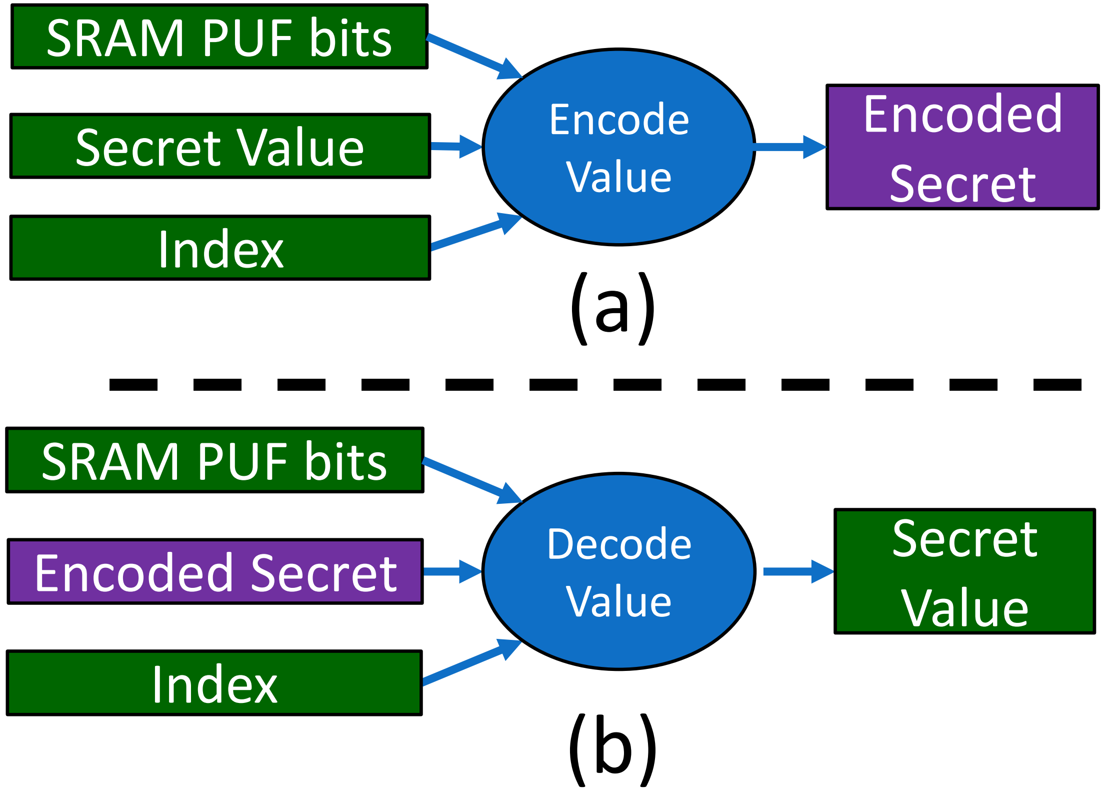
- According to documentation, the root key is not accessible externally via any hardware or software interface.

SRAM PUF Basic Functionalities

We utilize two main functionalities of the SRAM PUF:

(a) A secret value (such as a cryptographic key), is encoded using the values of the SRAM power-on state. The encoded secret can then be stored safely in non-volatile memory [14]

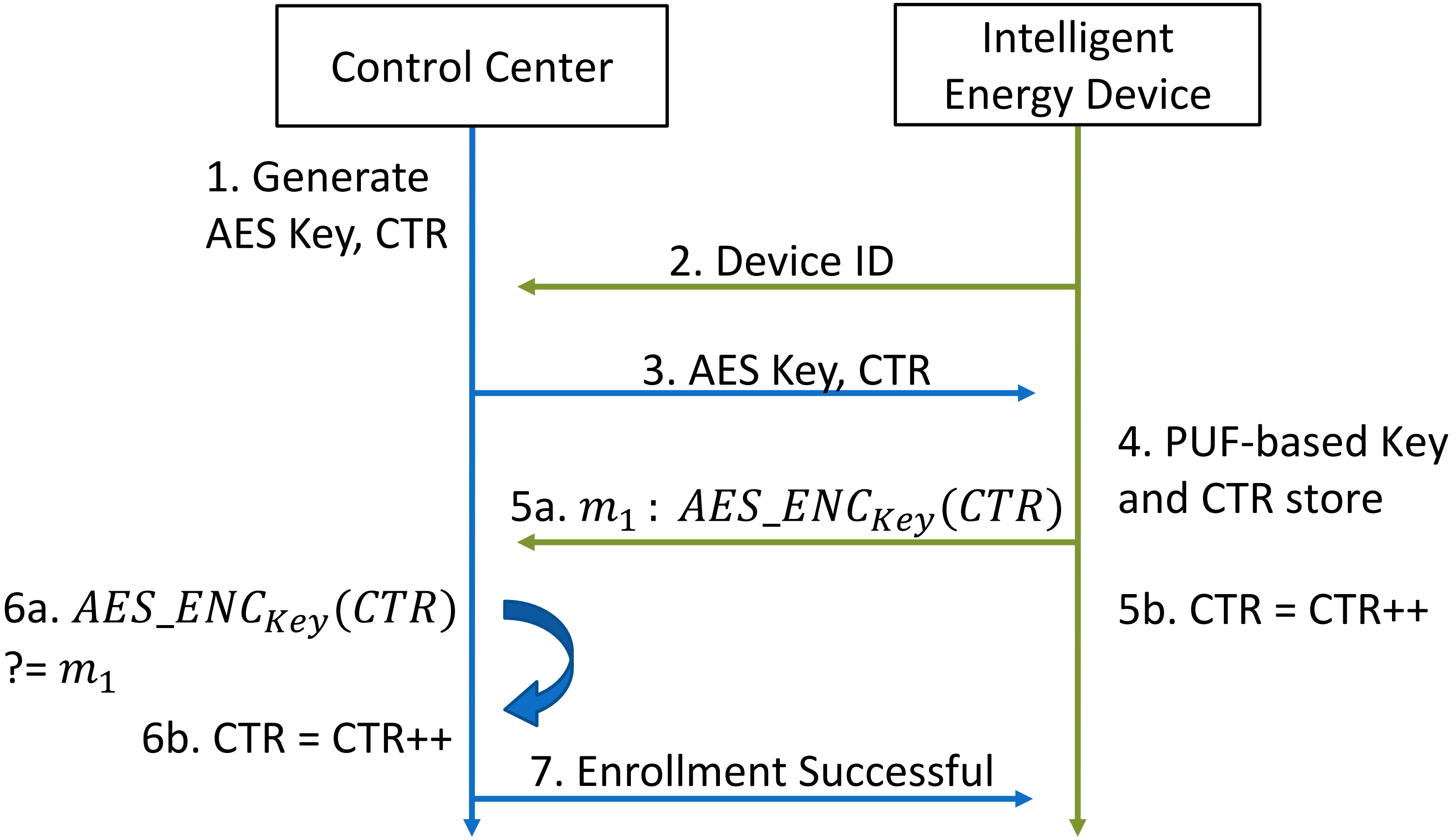
(b) The encoded secret value is decoded using the values of the SRAM power-on state [14]



Enhanced Protocol (PUF Enrollment)

- Enrollment establishes the conditions for the PUF authentication to function

- Enrollment is performed either in a secure facility or utilizes TLS if conducted remotely



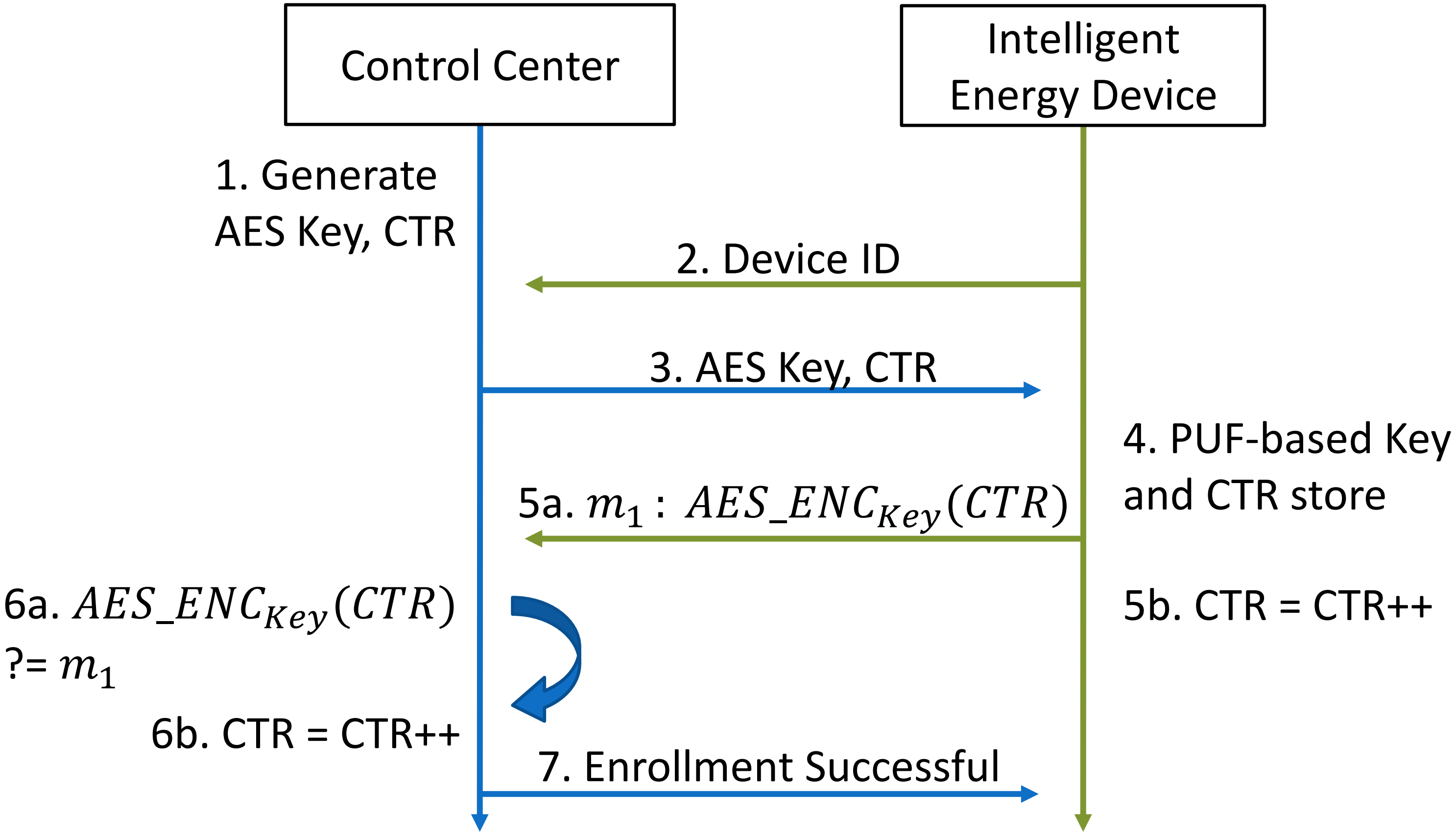
Control Center
Intelligent Energy Device
1. Generate AES Key, CTR
2. Device ID
3. AES Key, CTR
4. PUF-based Key and CTR store
5a. m_1 : $AES\_ENC_{Key}(CTR)$
5b. CTR = CTR++
6a. $AES\_ENC_{Key}(CTR)$?= m_1
6b. CTR = CTR++
7. Enrollment Successful

Enhanced Protocol (PUF Enrollment)

1) The control center generates two random 128-bit numbers, one of which will act as an AES key and one which will act as a counter.

2) The IED sends a device ID to the control center, used for database management.

3) The control center sends a unique AES key and random counter value CTR to the IED.

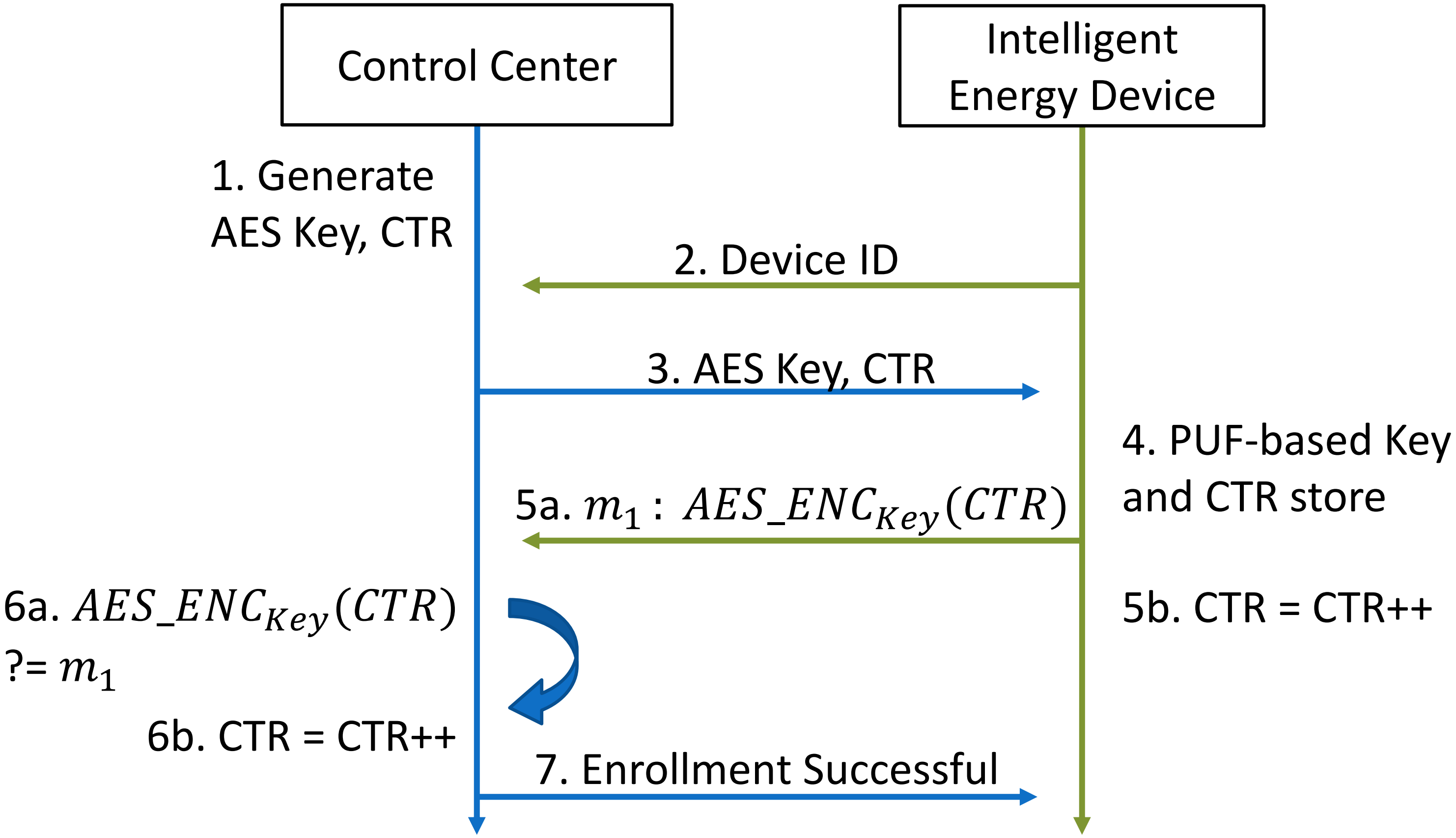

Control Center
Intelligent Energy Device
1. Generate AES Key, CTR
2. Device ID
3. AES Key, CTR
4. PUF-based Key and CTR store
5a. $m_1 : AES\_ENC_{Key}(CTR)$
5b. CTR = CTR++
6a. $AES\_ENC_{Key}(CTR)$?= m_1
6b. CTR = CTR++
7. Enrollment Successful

Enhanced Protocol (PUF Enrollment)

4) The IED utilizes the SRAM PUF to store the AES key and counter in an encoded format.

5a) The IED decodes the stored AES key and counter with the SRAM PUF. The counter is then encrypted via AES, creating m_1, which is transmitted to the control center.

5b) The IED increments the counter, encrypts the new counter value with the PUF and stores the encrypted counter.



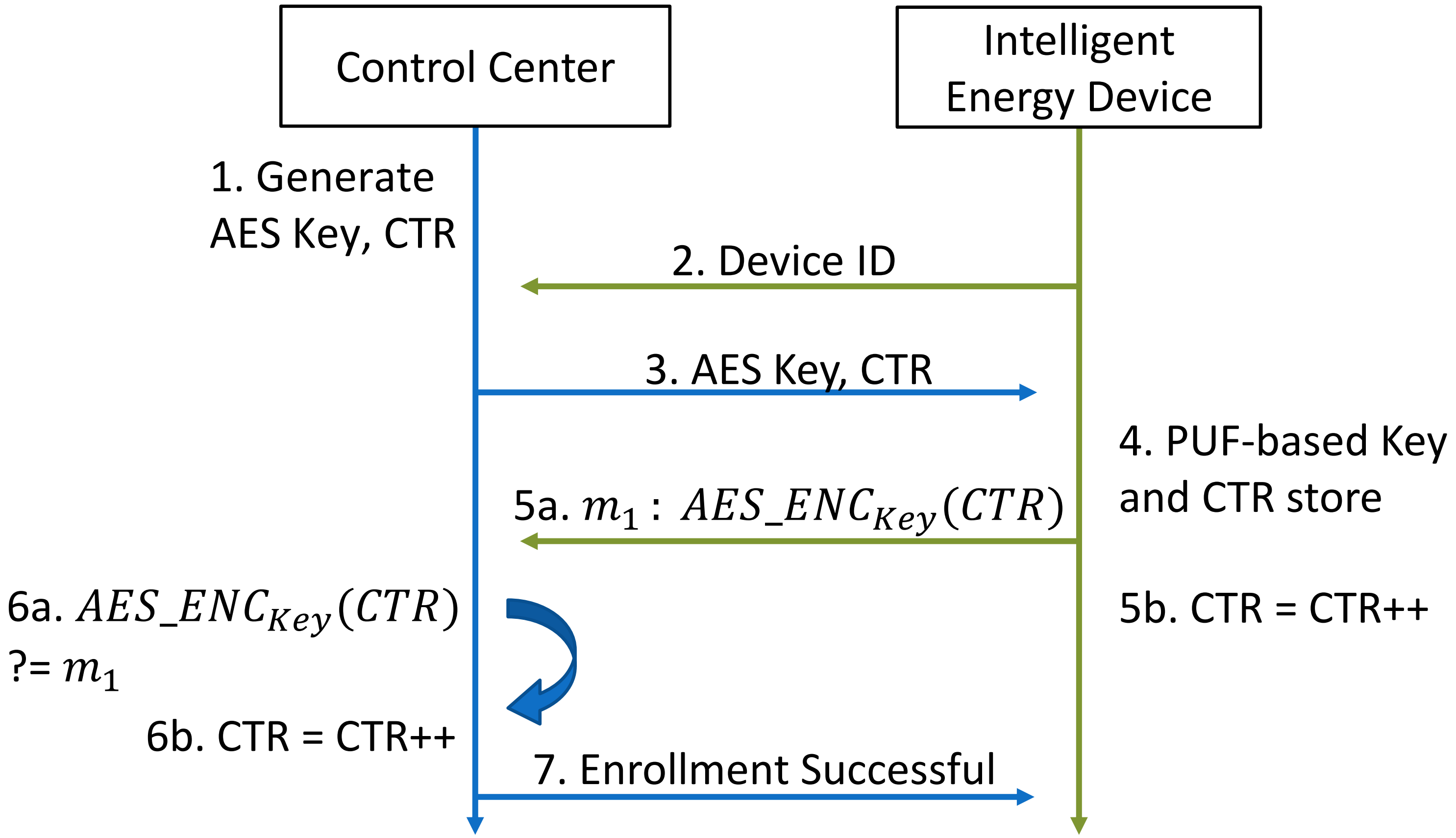
Control Center
Intelligent Energy Device
1. Generate AES Key, CTR
2. Device ID
3. AES Key, CTR
4. PUF-based Key and CTR store
5a. m_1 : $AES\_ENC_{Key}(CTR)$
5b. CTR = CTR++
6a. $AES\_ENC_{Key}(CTR)$?= m_1
6b. CTR = CTR++
7. Enrollment Successful

Enhanced Protocol (PUF Enrollment)

6a) The control center encrypts CTR with the AES key from step 3 and verifies the result matches m$_1$.

6b) The control center increments the counter CTR.

7) The control center sends a message to the IED informing of successful enrollment.


Control Center
Intelligent Energy Device
1. Generate AES Key, CTR
2. Device ID
3. AES Key, CTR
4. PUF-based Key and CTR store
5a. m_1 : $AES\_ENC_{Key}(CTR)$
5b. CTR = CTR++
6a. $AES\_ENC_{Key}(CTR)$?= m_1
6b. CTR = CTR++
7. Enrollment Successful

Enhanced Protocol (PUF Authentication)

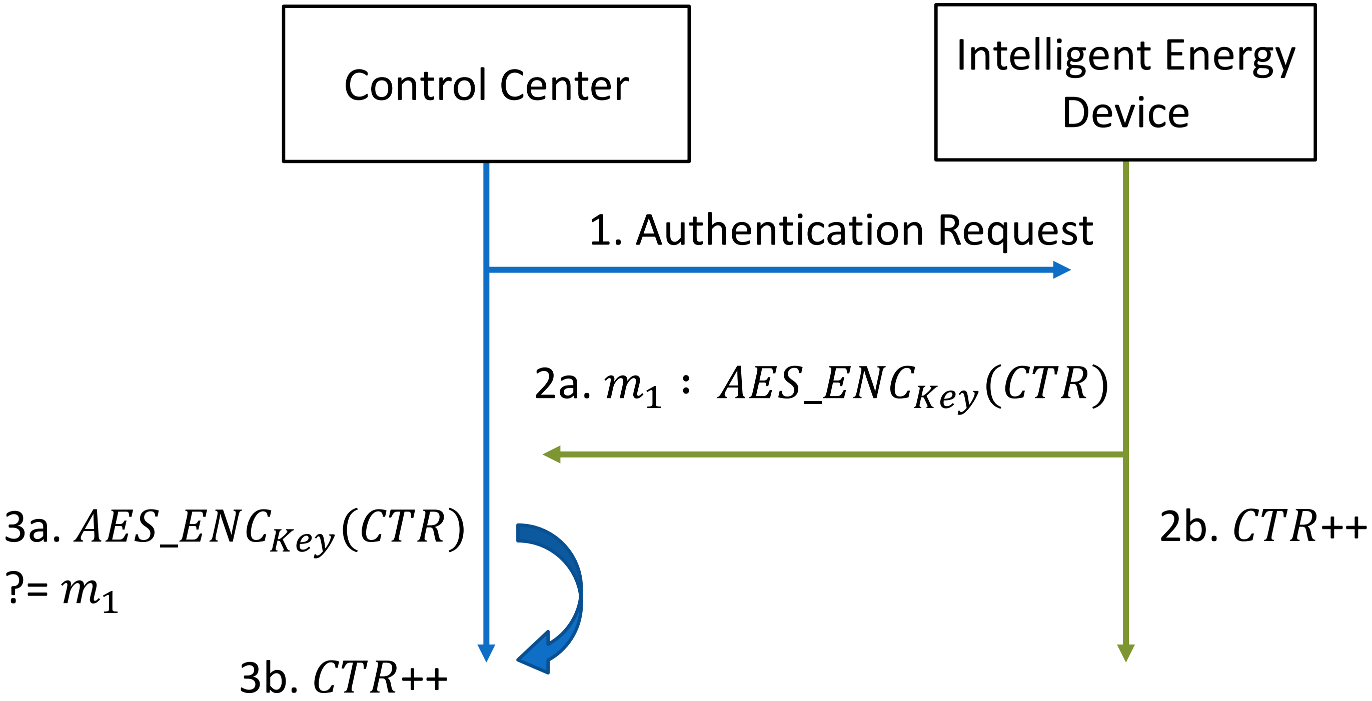
1) Control center sends a PUF authentication request to the device. This can optionally include a new counter value

2a) The IED encrypts the counter via the PUF stored AES key, producing m1, which is sent to the control center.

2b) The IED increments the counter and stores the value.

3a) The control center encrypts its copy of the counter with the IED's stored AES key and verifies a match with m_1.

3b) The control center increments the counter stored locally


Control Center
Intelligent Energy Device
1. Authentication Request
2a. m_1 : $AES\_ENC_{Key}(CTR)$
3a. $AES\_ENC_{Key}(CTR)$?= m_1
2b. CTR++
3b. CTR++

Contents

I. Problem Statement

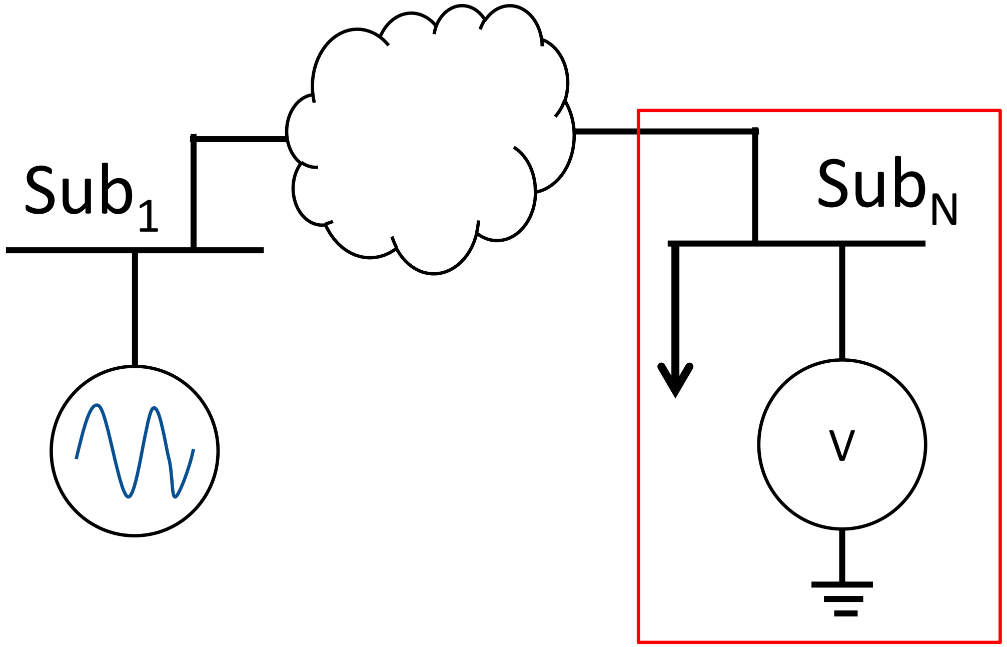
II. Standard Authentication

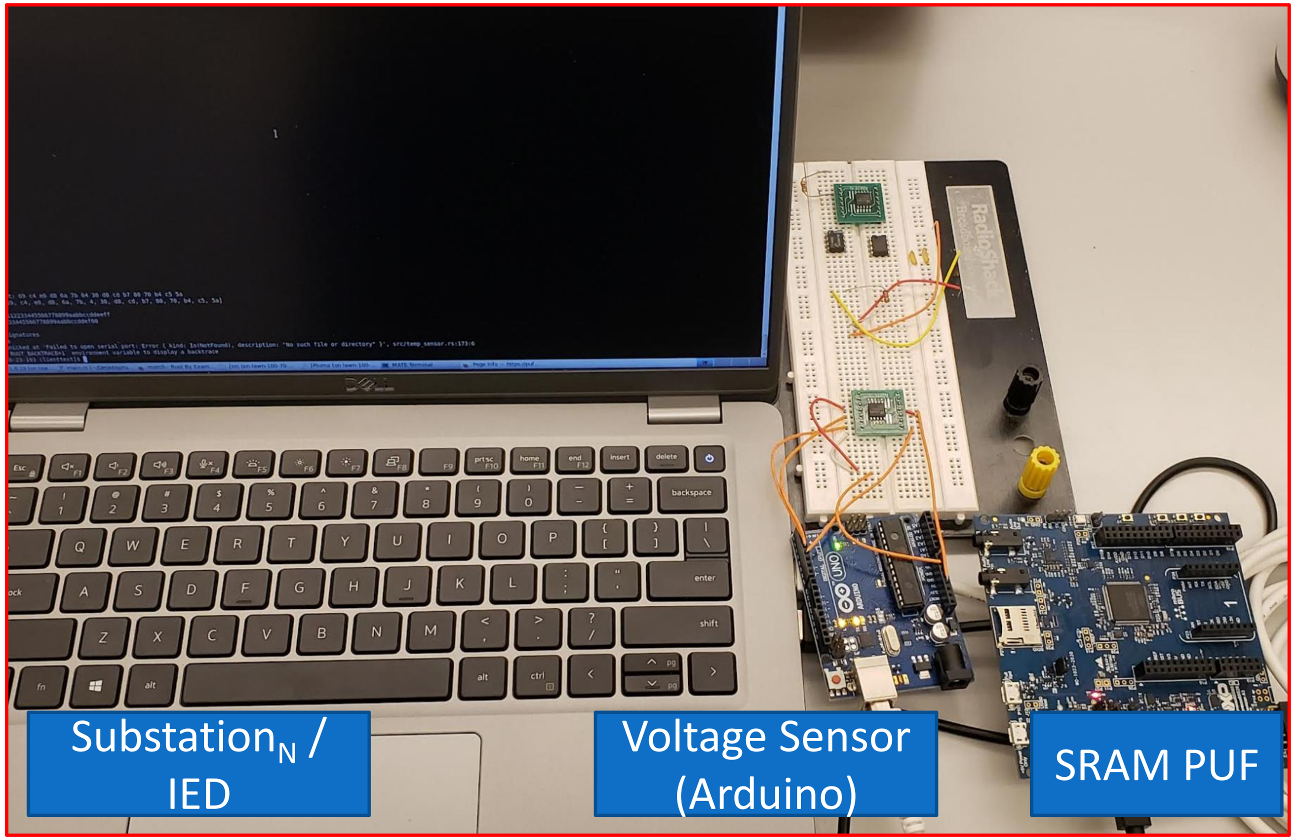
III. PUF Authentication

IV. Physical Implementation

V. Experiment and Results

Physical Implementation



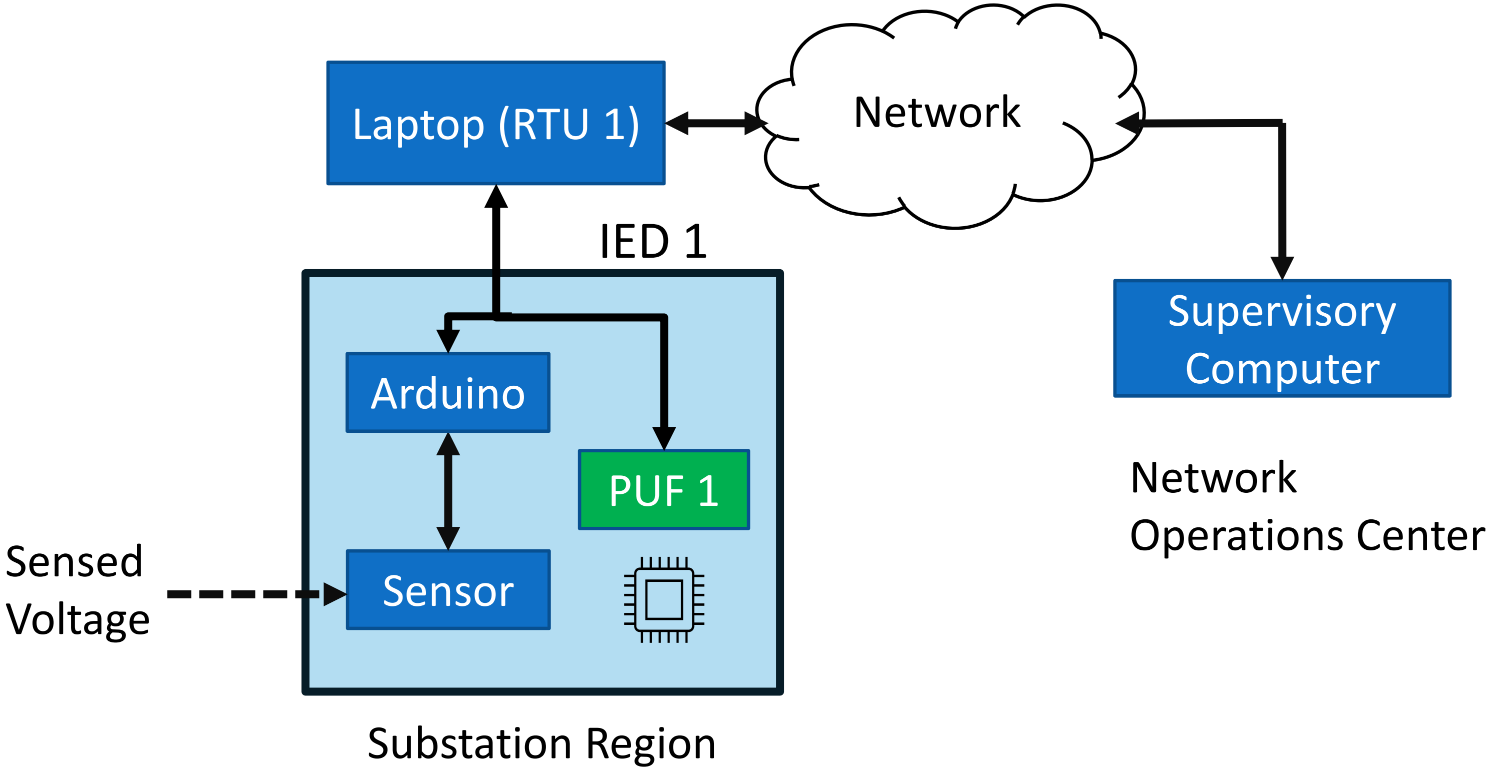


Substation_N / IED

Voltage Sensor (Arduino)

SRAM PUF

Physical Implementation



Contents

I. Problem Statement

II. Standard Authentication

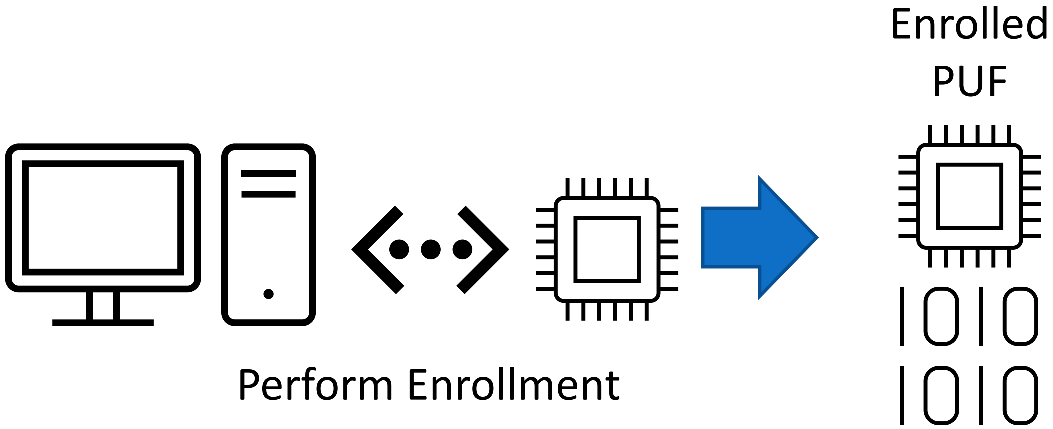
III. PUF Authentication

IV. Physical Implementation

V. Experiment and Results

Experiment Setup

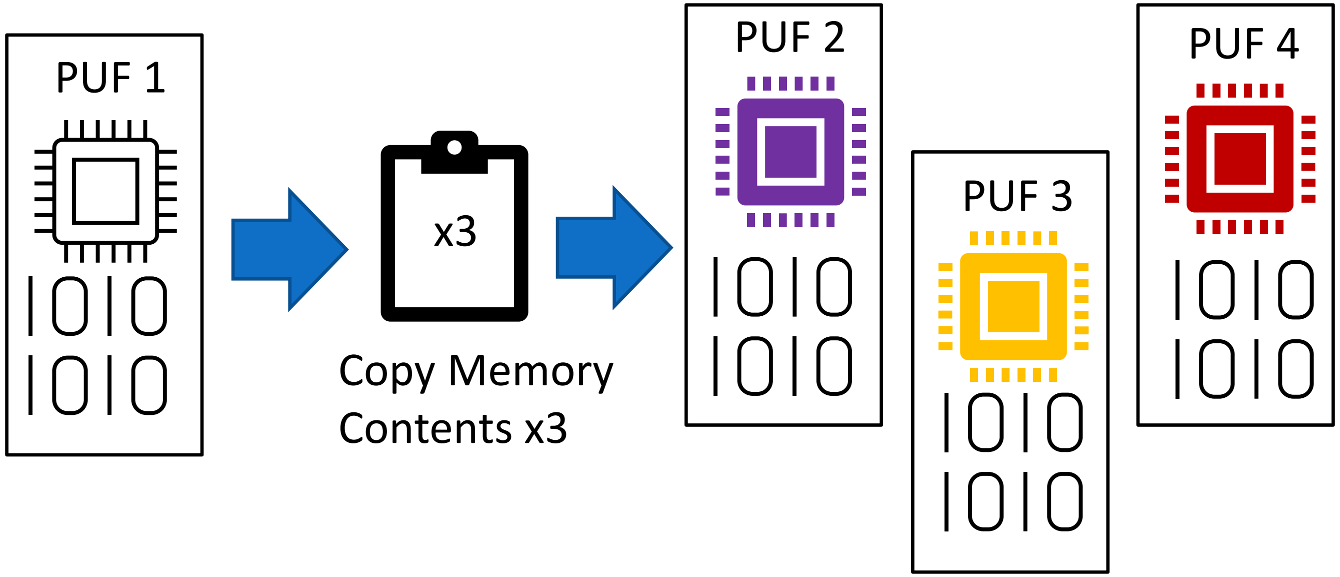
Performed enrollment on 1 NXP LPC55S69 [3][4] microprocessor and obtained the non-volatile memory contents



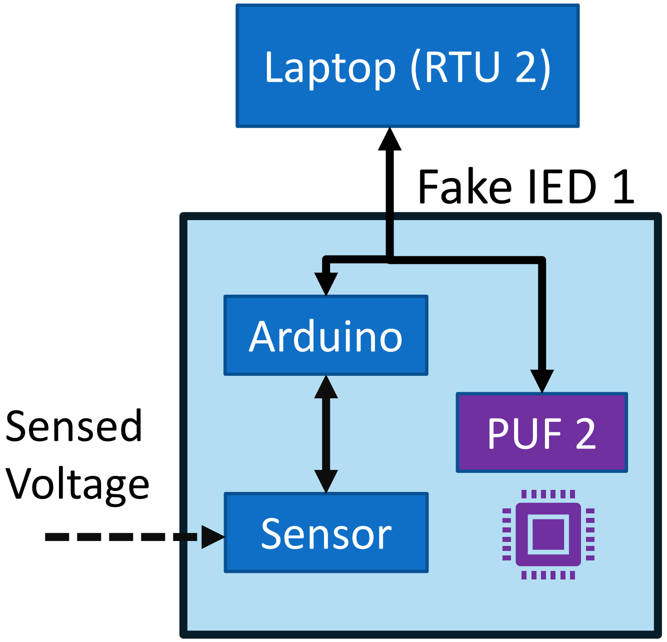
Experiment Setup

Loaded the non-volatile memory (NVM) contents of the “valid” PUF onto three other NXP LPC55S69 microprocessors [13][14]

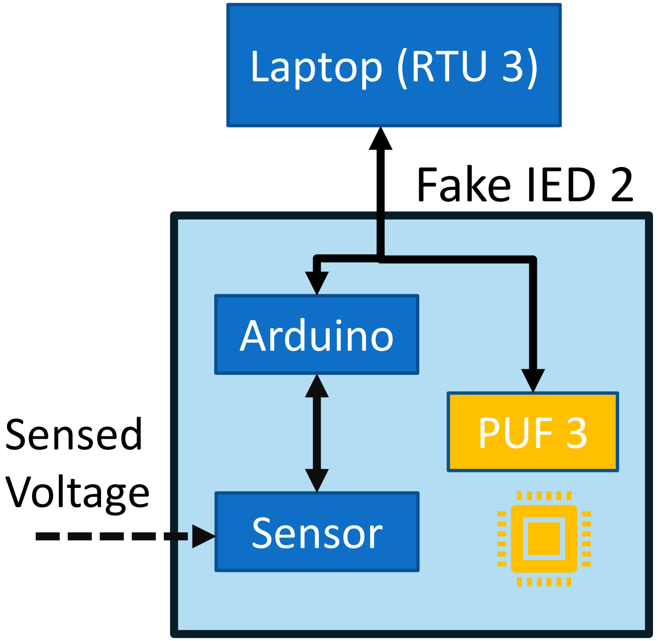
We now have four microprocessors running identical programs with identical NVM contents at time of power-up



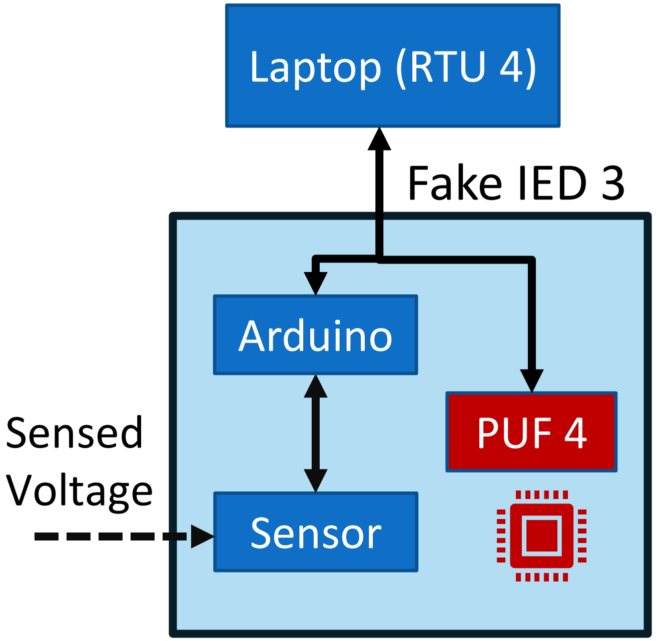
Experiment Setup



IED 2



IED 3



IED 4

Experiment Results and Discussion

Prevented authentication from succeeding despite cloning the non-volatile memory contents and running the exact same program on physically different microchips

PUF Authentication is very lightweight

- Dedicated AES module on NXP chip encrypts the counter
- Network data packets only increase in size by 128 bits

Commercial SRAM PUFs can be obtained inexpensively and provide protection against fake devices by a lone wolf insider

Experiment	Result
No PUF, No Attack	Normal Operation
No PUF, Spoofing Attack	Attack Succeeds
PUF Protected, No Attack	Normal Operation
PUF Protected, Spoofing Attack	Attack Fails

Thank You

Kevin Hutto*, Shuva Paul*, Benjamin Newberg*, Vishnu Boyapati^, Yathiendra Vunnam*, Santiago Grijalva*, and Vincent Mooney*^

*School of Electrical and Computer Engineering
^School of Computer Science
Georgia Institute of Technology, Atlanta, Georgia

khutto30@gatech.edu, spaul94@gatech.edu, bnewberg@gatech.edu, vboyapati6@gatech.edu, yvunnam3@gatech.edu

sgrijalva@ece.gatech.edu, mooney@ece.gatech.edu

Georgia Tech School of Electrical and Computer Engineering

References

[1] A. J. Wood, Power generation, operation, and control /, 3rd ed., 2014.

[2] K. Zetter, “Inside the cunning, unprecedented hack of ukraine’s power grid,” Mar 2016. [Online]. Available: https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/

[3] W. Turton and K. Mehrotra, “Hackers breached colonial pipeline using compromised password,” Jun 2021. [Online]. Available: https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password

[4] V. Chukwuka, Y.-C. Chen, S. Grijalva, and V. Mooney, “Bad data injection attack propagation in cyber-physical power delivery systems,” in 2018 Clemson University Power Systems Conference (PSC). IEEE, 2018, pp. 1–8.

[5] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, “The 2015 Ukraine blackout: Implications for false data injection attacks,” IEEE transactions on power systems, vol. 32, no. 4, pp. 3317–3318, 2017.

[6] Y. Liu, P. Ning, and M. Reiter, “False data injection attacks against state estimation in electric power grids,” in Proceedings of the 16th ACM conference on computer and communications security, ser. CCS ’09. ACM, 2009, pp. 21–32.

[7] J. Katz and Y. Lindell, Introduction to modern cryptography. CRC Press/Taylor & Francis, 2015.

[8] J. Kurose and K. Ross, Computer networking : a top-down approach /, seventh edition.. ed., 2016.

[9] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3,” RFC 8446, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8446

[10] R. Maes, Physically Unclonable Functions Constructions, Properties and Applications, 1st ed. Springer, 2013.

[11] P. Mall, R. Amin, A. K. Das, M. T. Leung, and K.-K. R. Choo, “Puf-based authentication and key agreement protocols for iot, wsns and smart grids: A comprehensive survey,” IEEE Internet of Things Journal, pp. 1–1, 2022.

[12] U. R¨uhrmair, J. S¨olter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, “Puf modeling attacks on simulated and silicon data,” IEEE Transactions on Information Forensics and Security, vol. 8, no. 11, pp. 1876–1891, 2013.

[13] “Quiddikey - Intrinsic ID: Home of PUF Technology,” Feb 2022. [Online]. Available: https://www.intrinsic-id.com/products/quiddikey/

[14] NXP, “LPC55S6x datasheet rev. 2.3,” August 6, 2021.

[15] N. D. Matsakis and F. S. Klock, “The rust language,” in Proceedings of the 2014 ACM SIGAda Annual Conference on High Integrity Language Technology, ser. HILT ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 103–104. [Online]. Available: https://doi.org/10.1145/2663171.2663188

[16] M. Turan, E. Barker, J. Kelsey, K. McKay, M. Baish, and M. Boyle, “Nist special publication 800-90b: Recommendation for the entropy sources used for random bit generation,” US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2018.