A Lightweight Cryptographic Permutation Generator for Critical Infrastructure Protection

*Arman Allahverdi, *Kareem Ahmad, *^Vincent John Mooney III, *†Santiago Grijalva

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

TPEC 2026

2017-2026 10

Georgia Tech

Acknowledgement

- This work has been partially supported by the U.S. Department of Energy (DoE) Office of Cybersecurity, Energy Security, and Emergency Response (CESER) under Cybersecurity for Energy Delivery Systems (CEDS) Agreement Number #DE-CR0000055 to the Georgia Tech Research Corporation: GRIDLOGIC: Hardware/Software Codesign for Deep Grid Visibility and Security

Georgia Tech

Outline

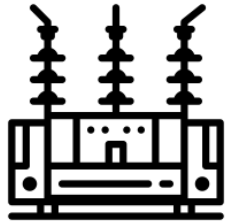
1. Problem Statement
2. Background
3. System Architecture
4. Experimental Evaluation
5. References

Outline

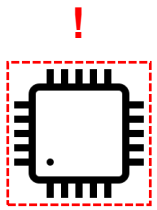
1. Problem Statement
2. Background
3. System Architecture
4. Experimental Evaluation
5. References

Problem Statement

- Field devices may be physically insecure and have strict resource/computing power limitations
  - Sensors
  - Field IEDs
  - Pole-Mounted Switches
- An adversary with access to a physically-insecure device may tamper with the memory contents of the device or otherwise cause the device to generate erroneous data
- Erroneous data transmitted upstream can falsely trigger protection mechanisms and cause device misoperation, damage, and outages



Control Center



Deployed Temperature Sensor

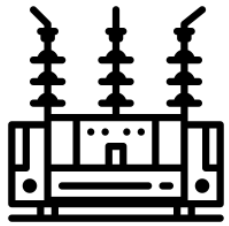
Remote Substation

Georgia Tech

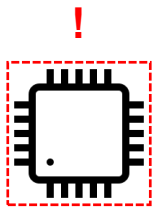
Problem Statement

- **Goal:** Design a hardware-efficient/lightweight security scheme that can be applied to devices to protect memory contents in the field

- The scheme should:
  - Not allow adversary access to plaintext data in device memory
  - Allow for the receiver to detect tampering/modification of the post-sensed data



Control Center



Deployed Temperature Sensor

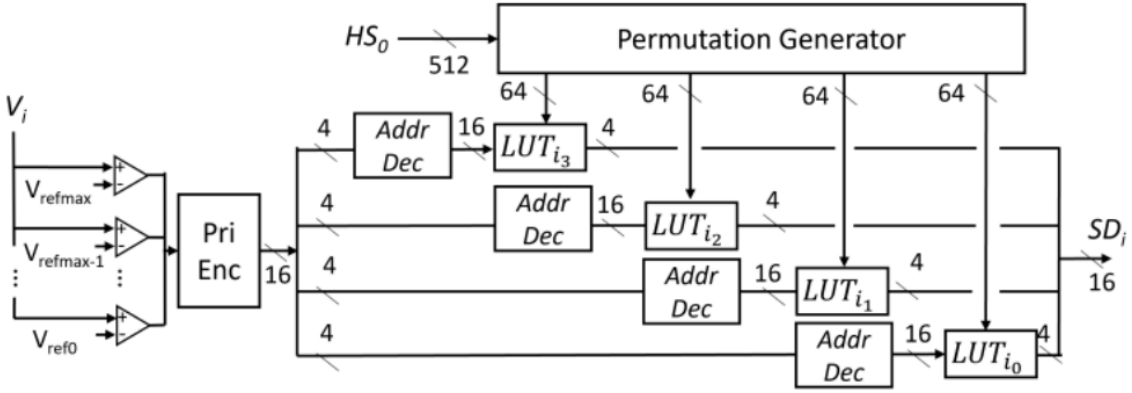
Critical Infrastructure

Outline

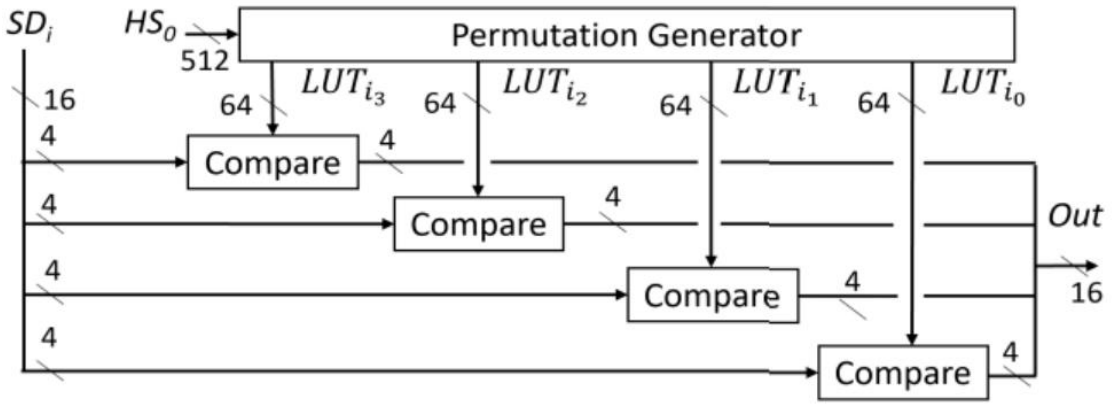
1. Problem Statement
2. Background
3. System Architecture
4. Experimental Evaluation
5. References

Background

- RanCode [1,2] is a hardware-oriented encoding scheme that uses pseudorandomly-generated lookup tables (LUTs) to encode/decode data
- To generate pseudorandom LUTs, RanCode requires the implementation of a cryptographically-secure pseudorandom number generator (PRNG), heretofore SHA-3 [1,2,3] (TPEC 2022)
  - SHA-3 has a significant hardware footprint, utilizing more hardware than the rest of the RanCode architecture



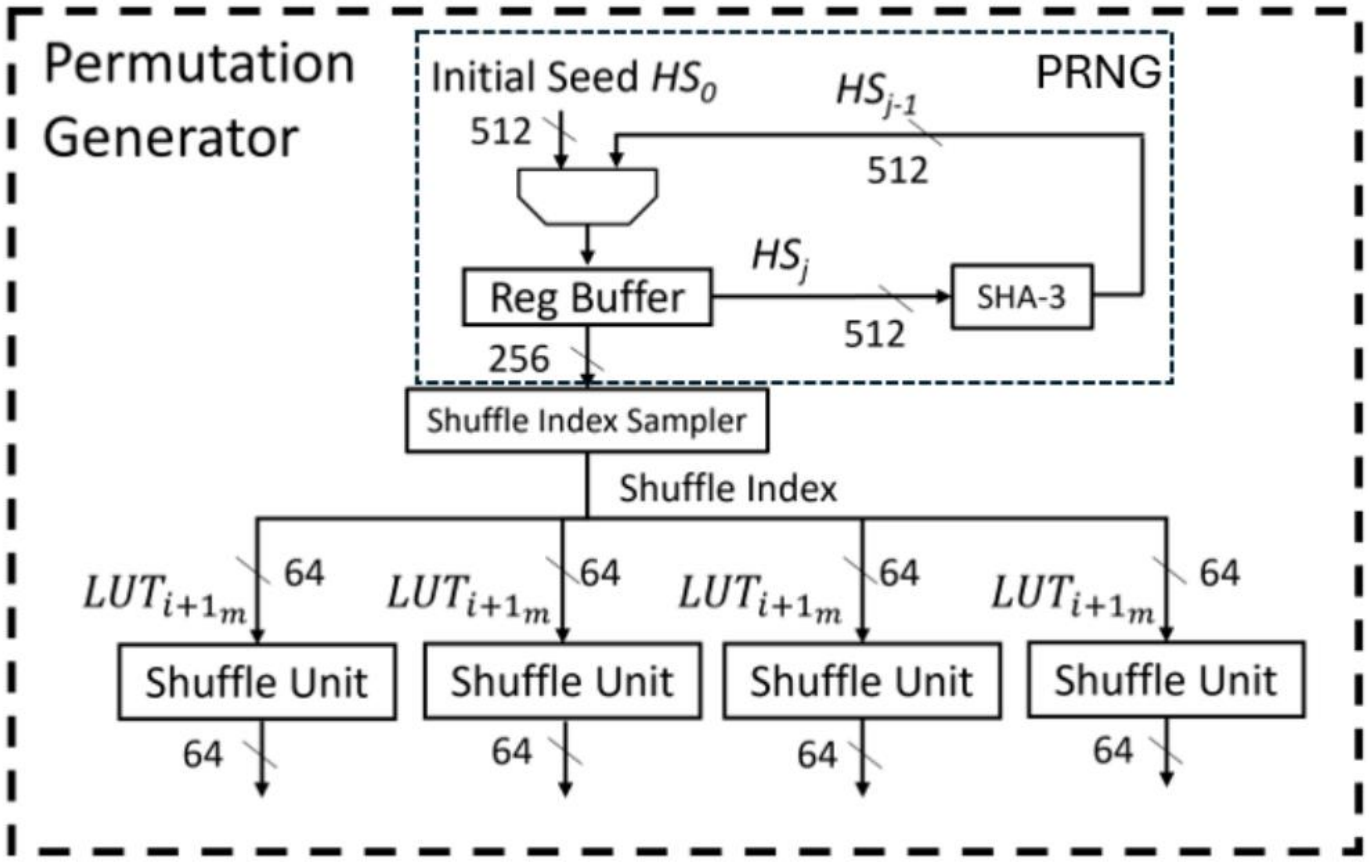
RanCode Encoding Architecture



RanCode Decoding Architecture

Georgia Tech

Background

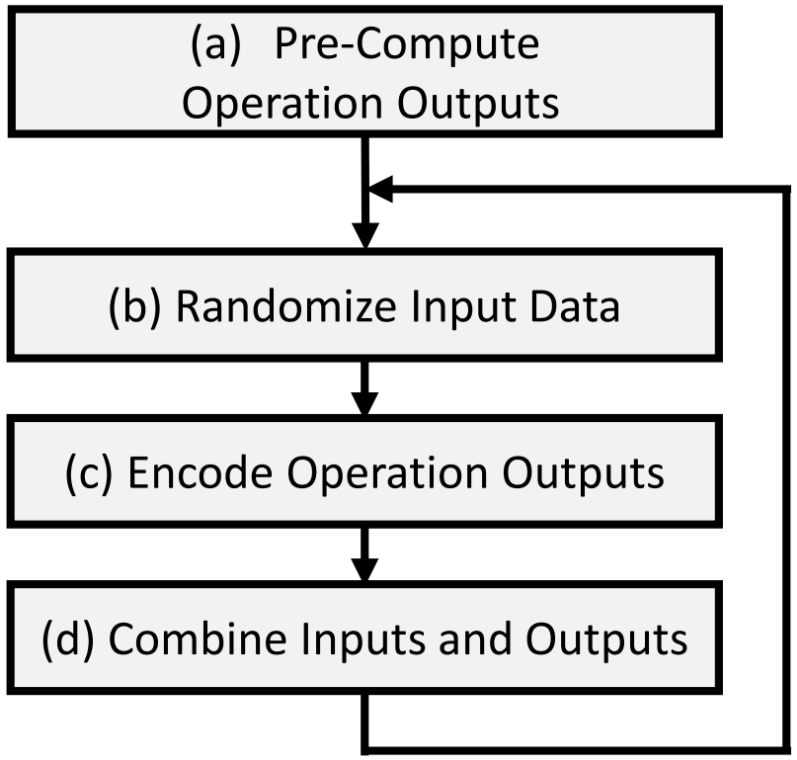


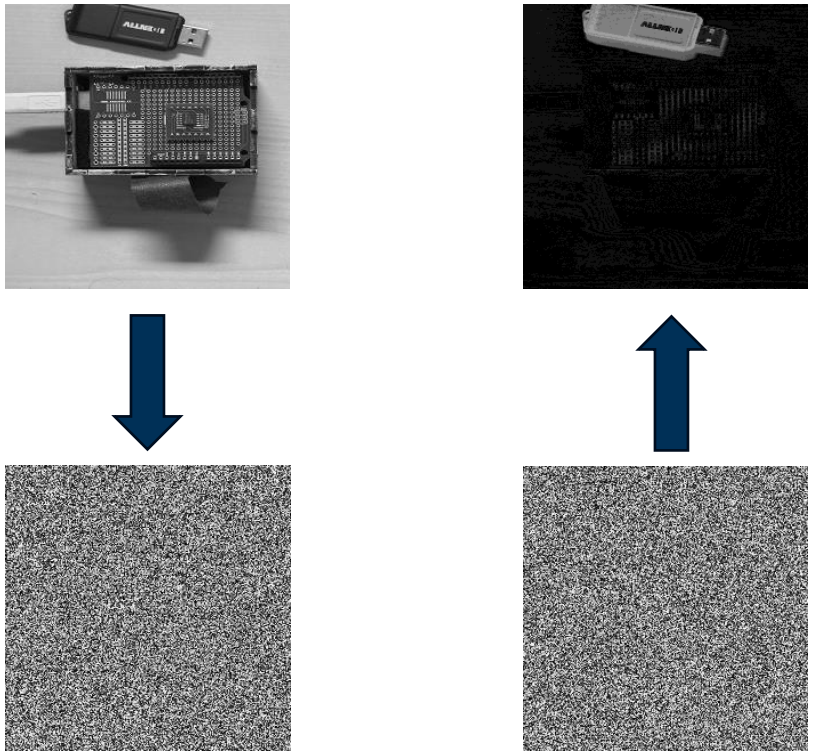
RanCode Permutation Generator

Georgia Tech

Background

- RanCompute [2] is a companion architecture to RanCode that allows for performing computation in the encoded domain
- RanCompute implements a *homomorphism*, meaning that algebraic computations on encoded values are reflected accurately in the decoded result



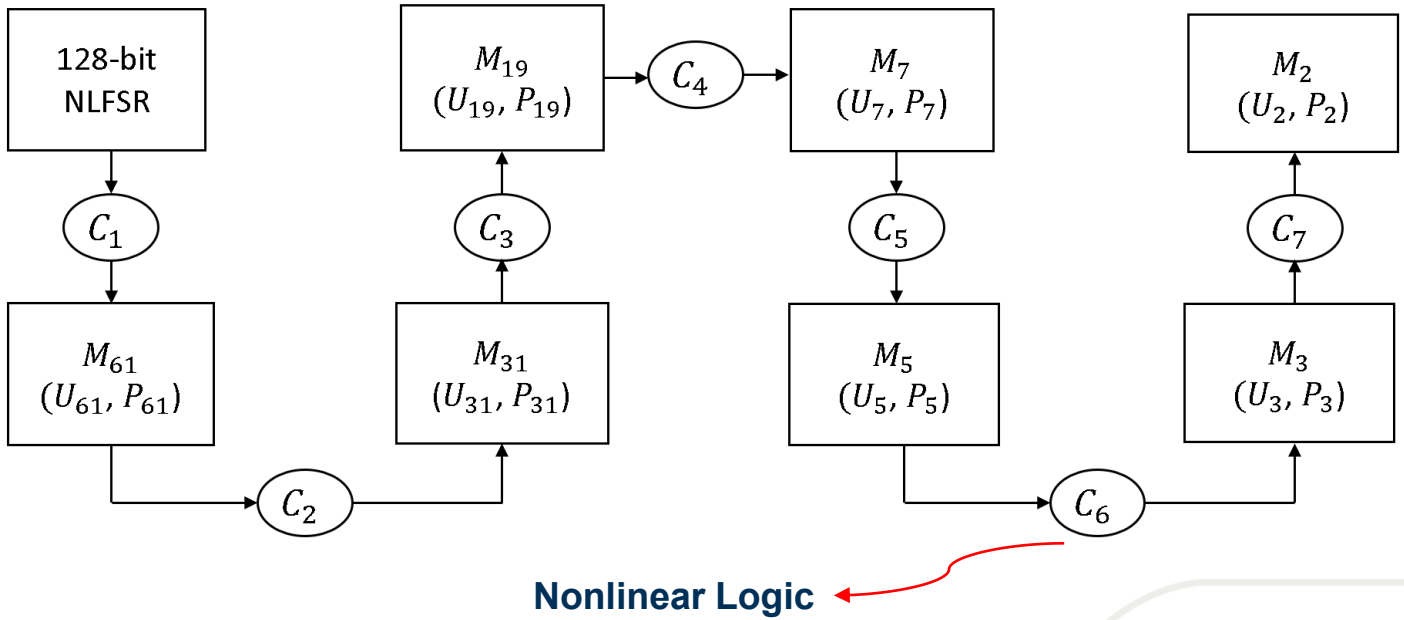


Background

- The hybrid register stream cipher [4] is a lightweight encryption scheme proposed for resource-constrained hardware environments
- This stream cipher is based on the interconnection of a Nonlinear Feedback Shift Register (NLFSR) and a Composite Mersenne Product Register (CMPR) [5]

- The PRNG of the hybrid register stream cipher is a 256-bit nonlinearly-updating register (128-bit NLFSR + 128-bit CMPR)
- The hybrid register stream cipher provides 128 bits of security

The Hybrid Register Construction



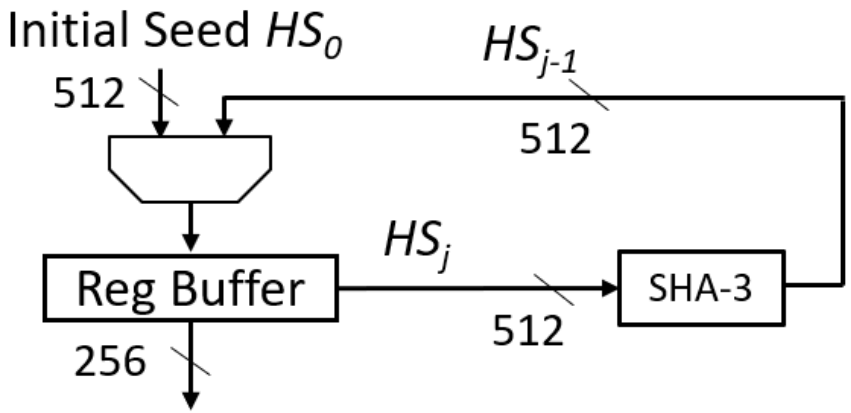
Georgia Tech

Outline

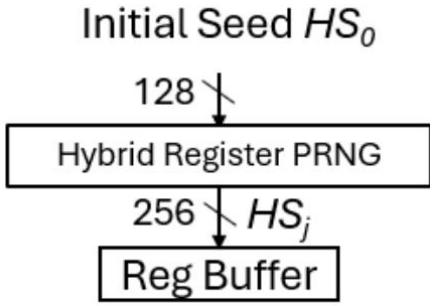
- Problem Statement
- Background
- System Architecture
- Experimental Evaluation
- References

System Architecture

- We propose replacing SHA-3 with the hybrid register in the RanCode permutation generator architecture
- Since the hybrid register was designed for stream cipher applications, it has exponential period
  - The hybrid register can generate a pseudorandom bitstream of up to 2^{81} bits [4]
- Therefore, the PRNG reseeding previously required in the RanCode architecture can be eliminated



Original Permutation Generator



Proposed Lightweight Permutation Generator

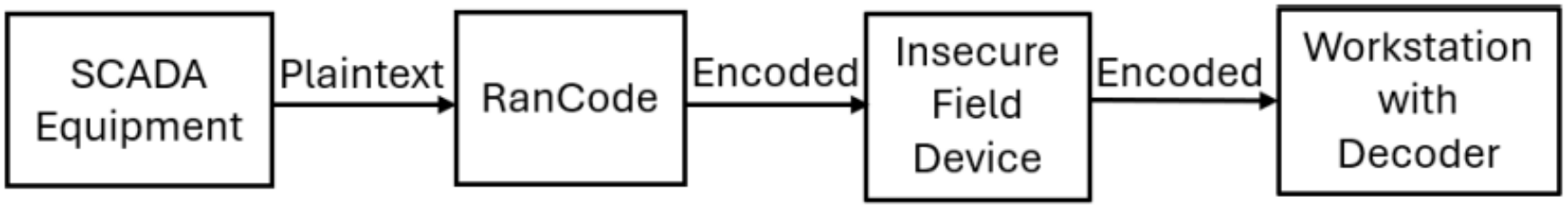
Outline

- Problem Statement
- Background
- System Architecture
- Experimental Evaluation
- References

Georgia Tech

Experimental Evaluation

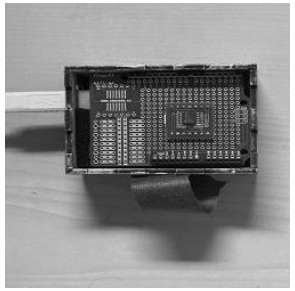
- **Sensor Data Encoding:**
  - Encoded 57,600 bits of temperature samples from a TI LM95172 temperature sensor IC
  - Each temperature sample consists of a fixed-point 16-bit temperature value
  - Randomness testing using the NIST Statistical Test Suite [6] indicates that the encoded data is pseudorandom and does not exhibit statistical vulnerabilities
    - An adversary with access to temperature sensor memory contents would not learn anything from reading the encoded data without attempting to break the hybrid register cryptography

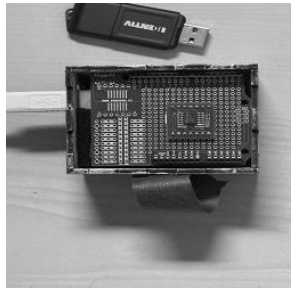


Sensor Data Encoding Experimentation Flow

Experimental Evaluation

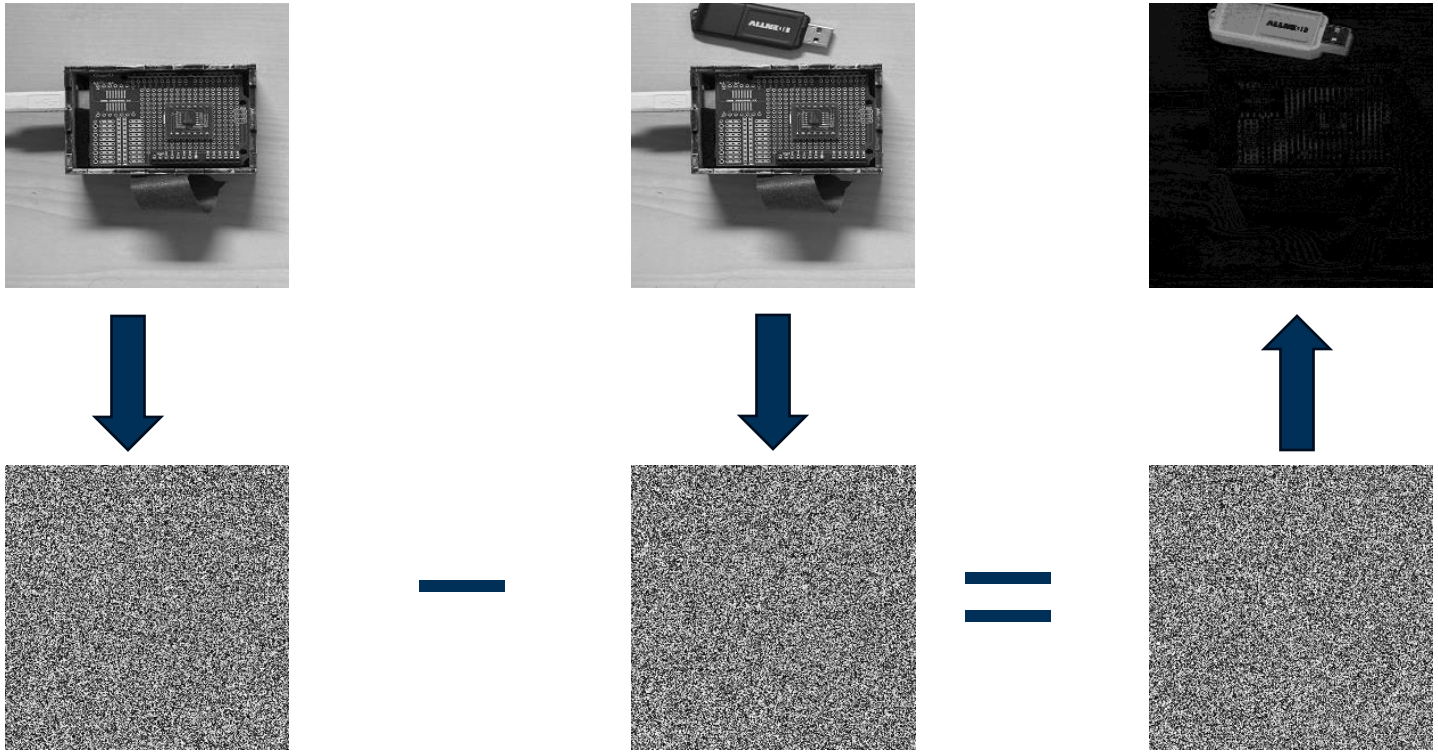
- **RanCompute Image Difference:**
  - Since RanCompute implements a homomorphism, it is possible to take the pixel-wise difference of images encoded with RanCode and correctly decode the result
  - Image difference is applicable to video feed monitoring and motion detection scenarios
  - The images below depict a scenario where a temperature sensor circuit is monitored by a camera, with one of the images including a USB stick above the circuit
  - This scenario ties into the physical attack surface we seek to address, where an adversary has access to a deployed device and can interact with the hardware, potentially using peripheral devices (e.g., USB sticks)





Experimental Evaluation

- **RanCompute Image Difference:**
  - Encoded two 250x250 PNG images of a temperature sensor device, where the second image includes a flash drive above the sensor
  - In the encoded domain, the pixel-wise difference of the images was taken, and the decoded result accurately shows the flash drive only



Experimental Evaluation

- RanCode implementation results on the Intel® DE10-Standard FPGA board:
  - -63% ALM utilization
    - ALMs (adaptive logic modules) are the combinational logic blocks used in Intel® FPGAs
  - -86% reduction register utilization
  - +20MHz encoding frequency

Circuit	ALMs	Registers	DSP	F_{max} (MHz)
Encoder	4044	3694	44	180
Decoder	4088	3461	44	200

RanCode with SHA-3

Circuit	ALMs*	Registers	DSP	F_{max} (MHz)
Encoder	1490	503	44	200
Decoder	1534	470	44	200

RanCode with Hybrid Register

Experimental Evaluation

- Additional Improvements:
  - 830 clock cycles per encoding -> 256 clock cycles per encoding
  - RanCompute image difference throughput for encoding 720p video feed at 200MHz*:
    - Prior work [2] demonstrates 15 FPS
    - LUT query takes 8 cycles [2]
    - 720p frame contains 1280 x 720 = 921,600 pixels
    - Per-pixel compute time: $t_p = \frac{8}{200} = 40$ ns
    - Frame time: $t_f = 921{,}600 * 40 \approx 37$ ms
    - Maximum throughput w/hybrid register: $1/37 \approx 27$ FPS

*Our throughput/FPS computation corresponds to the $\mathrm{F_{max}}$ obtained from the FPGA synthesis software; we did not use a dynamically reconfigurable FPGA target or implement a real-time video controller on an FPGA.

Outline

1. Problem Statement
2. Background
3. System Architecture
4. Experimental Evaluation
5. References

References

[1] K. Hutto, S. Grijalva and V. Mooney, “RanCompute: Computational Security in Embedded Devices via Random Input and Output Encodings,” *2022 11th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro*, 2022, pp. 1-8, doi: 10.1109/MECO55406.2022.9797150.

[2] K. Hutto, “Remote Sensor Security Through Encoded Computation and Cryptographic Signatures,” Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA, USA, 2024. [Online]. Available: https: //hdl.handle.net/1853/75289

[3] K. Hutto, S. Grijalva and V. Mooney, “Hardware-Based Randomized Encoding for Sensor Authentication in Power Grid SCADA Systems,” 2022 IEEE Texas Power and Energy Conference (TPEC), College Station, TX, USA, 2022, pp. 1-6, doi: 10.1109/TPEC54980.2022.9750706.

[4] A. Allahverdi and V. Mooney, “A hardware-efficient AEAD stream Cipher based on a hybrid nonlinear feedback register structure,” *2025 IEEE International Conference on Cyber Security and Resilience (CSR)*, Chania, Crete, Greece, 2025, pp. 1016-1023, doi: 10.1109/CSR64739.2025.11130096.

[5] D. Gordon, A. Allahverdi, S. Abrelat, A. Hemingway, A. Farooq, I. Smith, N. Arora, A. Chang, Y. Qiang, and V. Mooney, “Scalable nonlinear sequence generation using Composite Mersenne Product Registers,” *IACR Commun. Cryptol.*, vol. 1, no. 4, Jan. 2025, doi: 10.62056/a3tx11zn4.

[6] A. Rukhin *et al.*, “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” NIST Special Publication 800-22 Rev. 1a, Apr. 2010. [Online]. Available: https://nvlpubs. nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf