RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Inria

SORBONNE UNIVERSITÉ

SECURITY OF SMART DUST

Robust Key Derivation in Single-Chip Systems

Sara Faour

Supervised by:
Dr. Mališa Vučinić

Smart Dust: Promise and Security Challenge

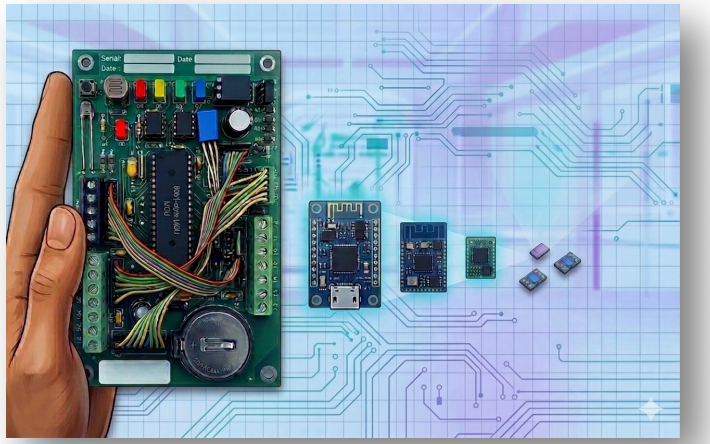
Smart Dust Vision (90's)

Millimeter-scale wireless motes

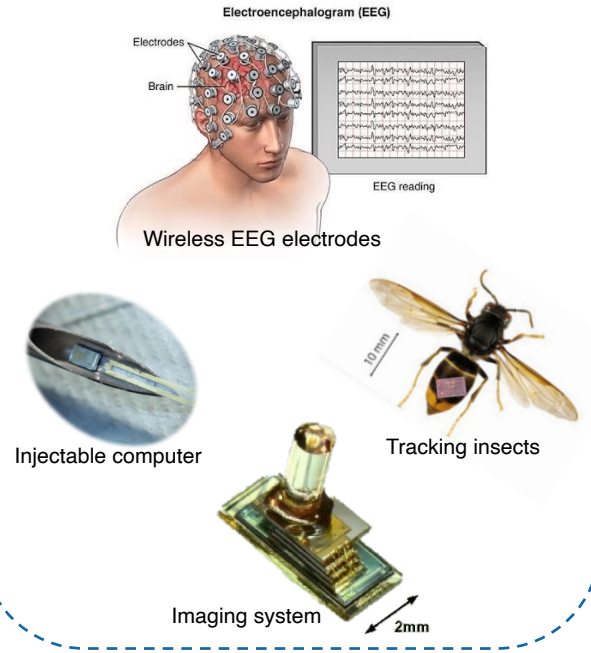
- *sensing, computation and communication*
- *deployable in large numbers*

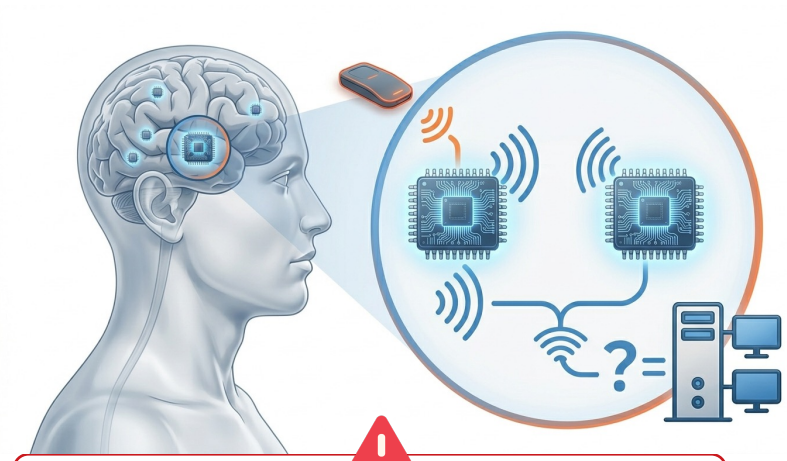
Properties

Ultra-small, low-cost, energy-efficient



Applications





Example: Medical implants

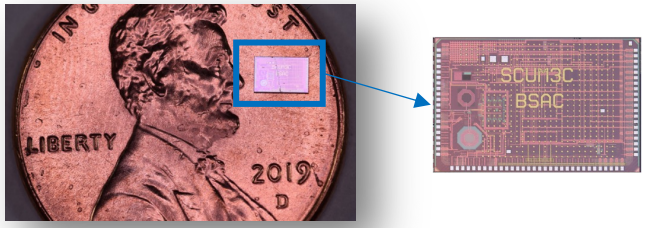
A fake implant is indistinguishable
→ attacker impersonates the patient
→ injects false health data

Grand Challenge ***How can we secure Smart Dust devices despite their extreme constraints?***

SCµM: Reality of Smart Dust

2019 - Single Chip µicro Mote (SCµM)

The Smart Dust vision realized on a single die:



- 2x3x0.3 mm^3
- Single CMOS process
- No external components (crystal-free)

No hardware root of trust is available by default

Why securing SCµM is fundamentally different?

Physically Exposed

Tampering

Cloning

Eavesdropping

Hardware Constraints

❌ **No writable Non-Volatile Memory (NVM)**
→ cannot store keys

when NVM present → not secure

❌ **No dedicated secure element**
→ additional chip contradicts Smart Dust

Objective 1

Characterize the physical attack surface of single-chip motes

SCµM is harder to attack in most categories

Objective 2

Design a lightweight and secure root of trust

Our Approach: SRAM PUF as Root of Trust

Every human has a unique fingerprint

- *random variations during development*
- *never chosen, impossible to replicate*

Every chip has a unique physical fingerprint

Physical Unclonable Function (PUF)

We can read this fingerprint from the SRAM memory (on start-up):

SRAM PUF

- present on every mote
- device-unique
- NVM-free root of trust

SRAM start-up

🔑 **Secret Key on-demand**

Design Requirements

- Self-contained
- Resource-constrained
- Standard SRAM (software-only)

Outline

- State of the Art
- Methodology

Main Contributions

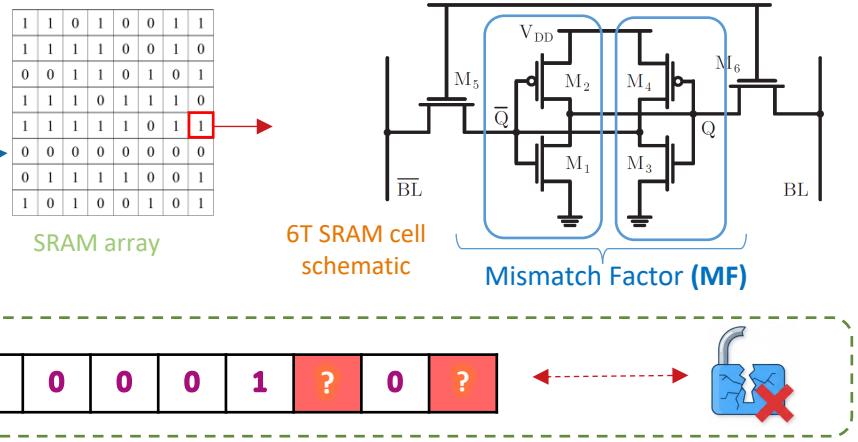
- **C1** Single-Chip Motes and SRAM PUF: A Feasibility Study
- **C2** TMVS: Threshold-based Majority Voting Scheme for Robust SRAM PUFs
- **C3** Two-Stage TMVS
- **C4** ODHD: On-Demand Helper Data Generation for Reliable NVM-Free Key Derivation from SRAM PUFs

- Conclusions

SRAM PUF and the Reliability Challenge

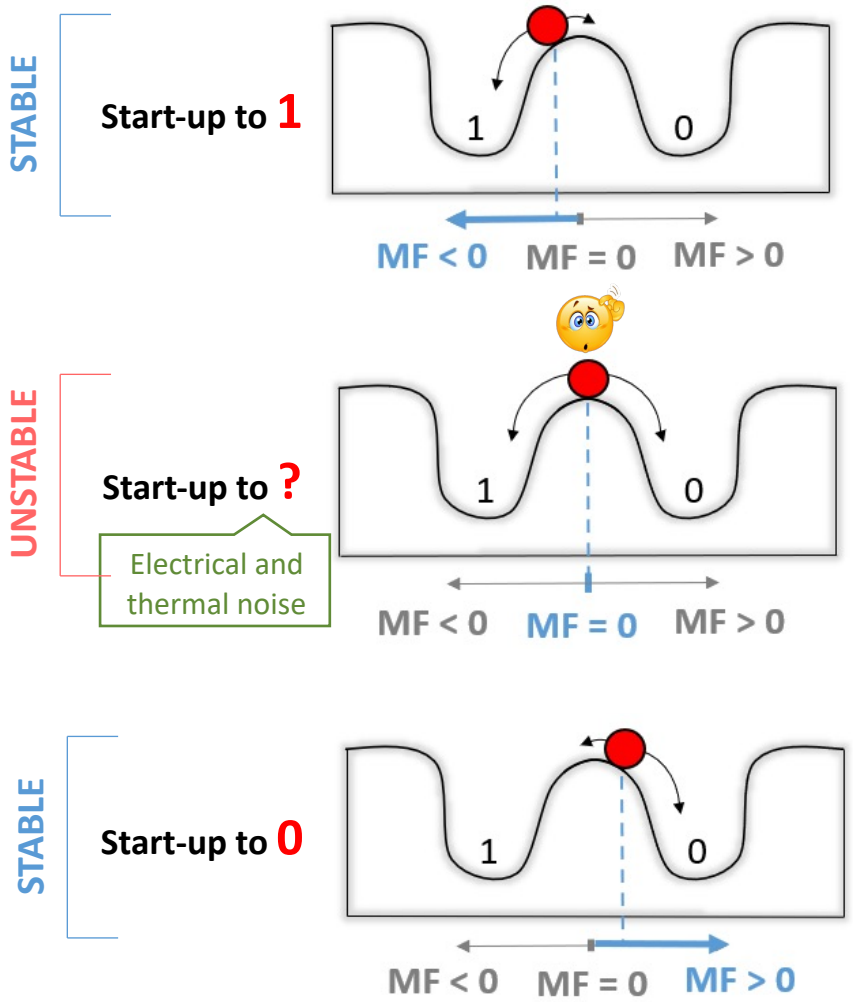
SRAM PUF: hardware root of trust without NVM

- Each 6T SRAM cell powers up to '0' or '1' based on transistors MF
- Start-up SRAM values → unique Silicon fingerprint
- **Mismatch factor (MF)**
  - |MF| large → cell is **stable**
  - MF≈0 → cell is **unstable**



Raw SRAM cannot be used directly as a key

Stabilization mechanisms are required



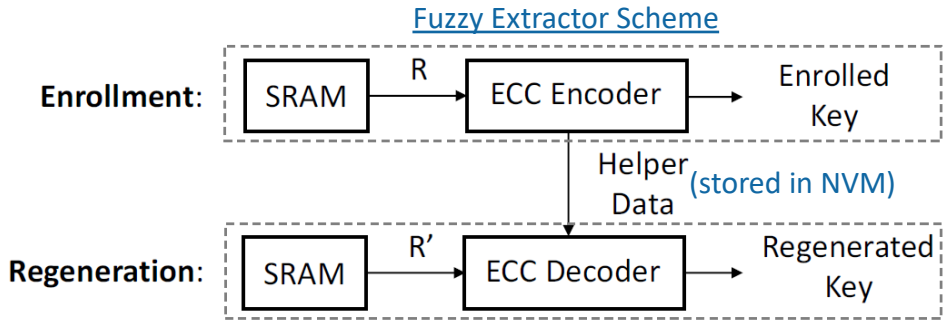
Existing SRAM PUF Stabilizers

Fuzzy Extractor

Two phases:

Enrollment (once) & **Regeneration** (every boot)

→ Helper data is public but must not leak the key



Repetition Codes (helper-based baseline)

majority=1

1	1	0	1	1

Each key bit **encoded as n copies** → **decoded** by majority vote

✅ **Trivial decoder, single SRAM read, software-only**

❌ Helper data leaks heavily → **entropy loss**

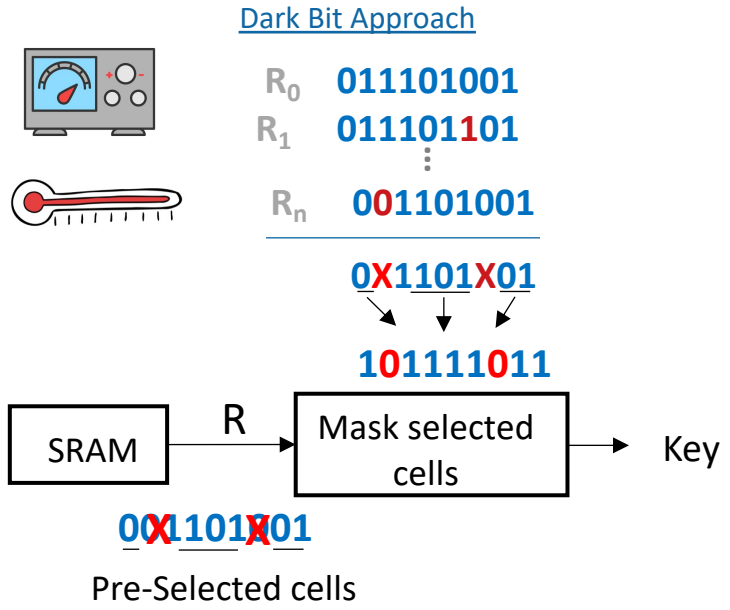
Dark Bit (helperless baseline)

Read each SRAM many times → Use only stable cells

✅ **no helper data, no NVM required**

❌ **Many power cycles during enrollment**

❌ **Residual errors too high for secret key derivation**



Open Research Gaps

	Scheme	NVM needed	Leakage	Readouts	Decoder cost†
Helper-based	Code-offset (ECC) [1]	✅ Yes	❌ High	1	❌ High
Helper-based	**Repetition codes** [1]	✅ Yes	❌ **High**	1	✅ **Low**
Helper-based	SLLC / RMCC+DSC [2][3]	✅ Yes	✅ ≈ 0	1	❌ High
Helper-based	IBS [4]	✅ Yes	✅ ≈ 0	**>1**	Medium
Helper-based	SMV [5]	✅ Yes	❌ High	1	✅ Low
Helper-based	TMV + ECC [6]	✅ Yes	As per ECC	>1	Medium
	Dark Bit* [7]	❌ **No**	(Unquantified)	❌ **High**	✅ **Low**

*weak error correction

†**Decoder cost** = memory, CPU, and energy overhead of the decoding algorithm

Helper-based gap

Lightweight schemes **leak**

Low-leakage schemes are **complex**

🎯 **Goal:** develop an efficient stabilizer with **low decoder cost + low leakage + single measurement → TMVS, TS-TMVS**

Helperless (no NVM needed) gap

Dark Bit **avoids NVM** but residual **errors remain too high**

🎯 **Goal:** ensure SRAM PUF reliability **without any NVM storage → ODHD**

Outline

- State of the Art
- **Methodology**
- **C1** Single-Chip Motes and SRAM PUF: A Feasibility Study
- **C2** TMVS: Threshold-based Majority Voting Scheme for Robust SRAM PUFs
- **C3** Two-Stage TMVS
- **C4** ODHD: On-Demand Helper Data Generation for Reliable NVM-Free Key Derivation from SRAM PUFs
- Conclusions

Threat Model


Physical access to devices (temporary or permanent)


Laboratory analysis with specialized tools

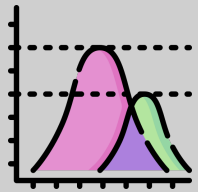

Key extraction & device cloning


Environmental stress attacks (power, temperature)


Side-channel monitoring (power, EM)

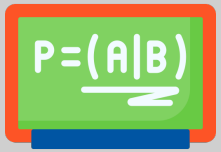
Methodology

Statistical Model



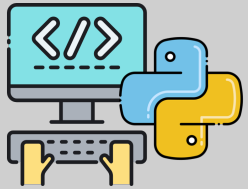
- **I.I.D.** SRAM PUF bits
- Each cell $\mathbf{R_i}$**~Bernouilli(p)**
  - Bias: **p**=P(R_i=1)
  - Unbiased: p=0.5
- Flip probability: $\mathbf{p_e}$= P($R_i \neq R'_i$)
- PUF requirements

Theoretical Analysis



- Decoding error probability
- Selection probability and memory requirements
- Min-entropy $\boldsymbol{H}_\infty$: remaining uncertainty given helper data
- Secrecy leakage $\boldsymbol{I}(\boldsymbol{K};\boldsymbol{S})$

Implementation



- Algorithms implemented in Python/C
- Available as open source:
  🔗 https://github.com/Sara-Fa/SRAM-PUF-key-generation

Validation through Experimentation

- Deployed on **ARM Cortex-M0** (SCµM-3C)
- Measured **clock cycles** for key reconstruction
- Theory vs. experiment comparison
- Comparison to baselines: SRAM size, leakage, readouts, decoder cost, failure rate

SCuM-3C Dataset

9 chips × 1000 readouts × 55 kB SRAM

Dataset Acquisition

SCµM-3C Platform

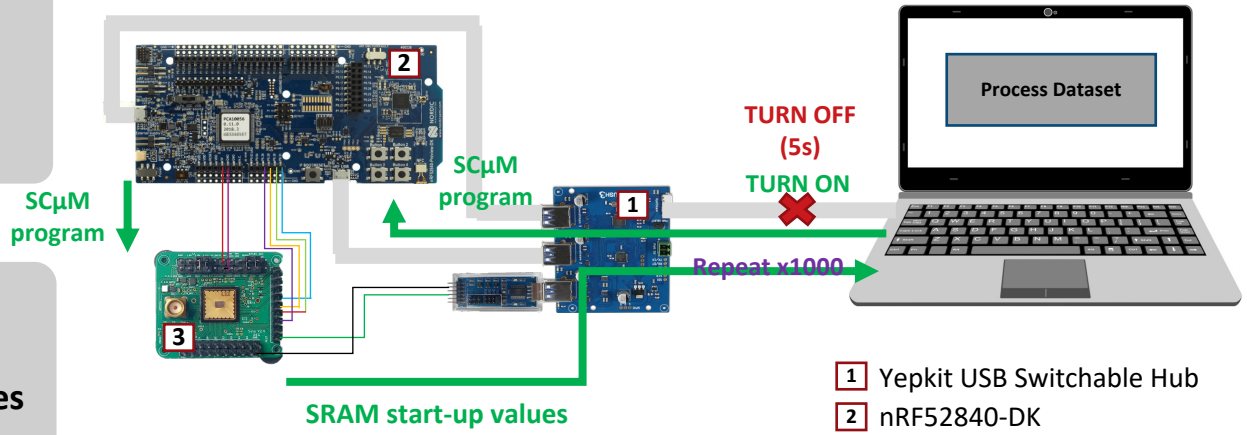
- ARM Cortex-M0
- 64 kB program SRAM + 64 kB data SRAM
- Fabricated in TSMC 65 nm LP CMOS
- No NVM — firmware reloaded at every power cycle

Resulting Dataset

- **9 SCµM-3C chips × 1000 readouts = 9000 start-up images**
- 55 kB data SRAM per image, at ambient temperature
- Scripts and dataset publicly available as open source

https://github.com/bkorecic/scum-automated-sram-read
https://github.com/bkorecic/scum-sram-evaluation

1 power-up cycle = 1 TURN ON + 1 TURN OFF



1 Yepkit USB Switchable Hub
2 nRF52840-DK
3 SCµM-3C placed on Sulu v2.0 experimental board

This dataset is the basis for all upcoming experimental evaluations

Outline

- State of the Art
- Methodology
- **C1 Single-Chip Motes and SRAM PUF: A Feasibility Study**
- **C2** TMVS: Threshold-based Majority Voting Scheme for Robust SRAM PUFs
- **C3** Two-Stage TMVS
- **C4** ODHD: On-Demand Helper Data Generation for Reliable NVM-Free Key Derivation from SRAM PUFs
- Conclusions

Single-Chip Motes and SRAM PUF: A Feasibility Study

Not all SRAMs are suitable as PUF!

Secure Key Generation

- Random
- Unique
- Reliable

Does SCµM meet these PUF requirements?

SCµM' SRAM is valid as PUF

BER is too high for direct cryptographic use

Using SCuM-3C Dataset

Randomness

≠ **cells**
same SRAM
same time

Uniqueness

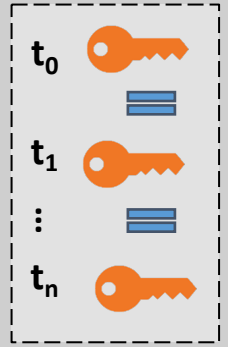
same cells
≠ **SRAMs**
same time

SRAM

another SRAM

Reliability

same cell
same SRAM
≠ **times**



BER ≈ 4.75%

All three properties satisfied

Outline

- State of the Art
- Methodology
- **C1** Single-Chip Motes and SRAM PUF: A Feasibility Study
- **C2 TMVS: Threshold-based Majority Voting Scheme for Robust SRAM PUFs**
- **C3** Two-Stage TMVS
- **C4** ODHD: On-Demand Helper Data Generation for Reliable NVM-Free Key Derivation from SRAM PUFs
- Conclusions

TMVS Principle: Threshold-Based Selection

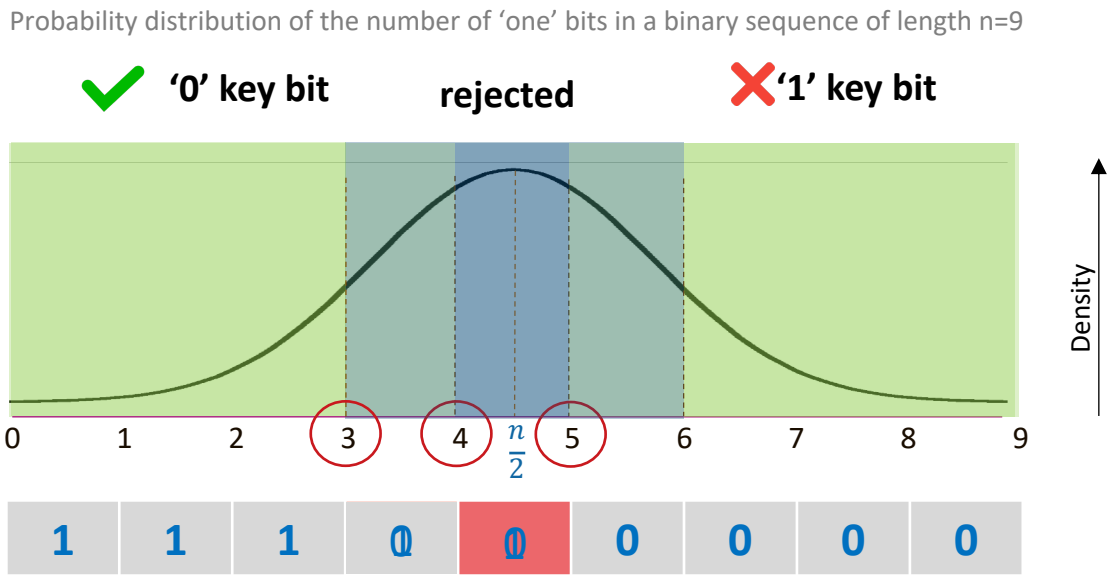
- Unbiased SRAM has p = 0.5 (probability of '1' bit)
- SRAM sequence= block of n bits
- #ones in a n-bit SRAM ~Binomial(n, 0.5)

Limitations of Majority Voting

- Selects dominant bit
- ✗ FAILS when #zeroes ≈ #ones (≈ n/2)
- Small flips → decoding errors

Threshold-based Majority Voting

- Reject patterns near n/2
- Keep patterns respecting a fixed threshold
- ✓ Tolerates bit flips without errors



Case 0: no-threshold → 1 flip causes error

Case 1: threshold = 1 away from n/2 → up to 1 flip tolerated

Wider threshold = more tolerated flips + fewer accepted sequences

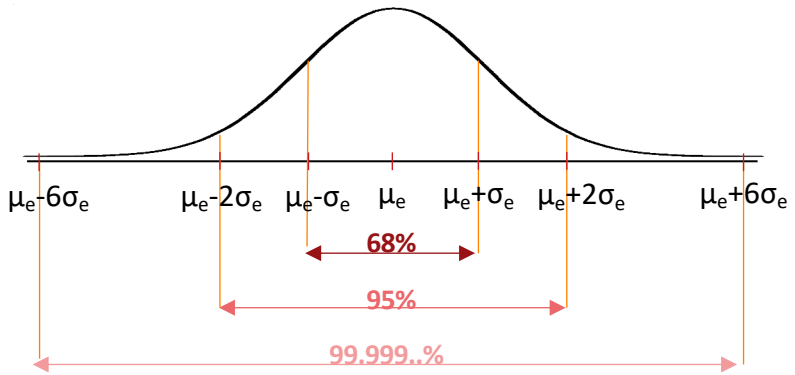
TMVS: Threshold Choice

Bit Flip Model

- Flips between two reads: n_e ~ **Binomial(n, p_e)**
  - Mean μ_e, std. dev. σ_e
- Maximum number of tolerated bit flips:

$$N_{e,max} = \lceil \mu_e + x_\sigma \sigma_e \rceil$$

- **With $x_\sigma = 6$: only $1.97 \cdot 10^{-9}$ realizations fall outside $\mu_e \pm 6\sigma_e$**
- **Error only if number of flips exceeds $N_{e,max}$**



Empirical Rule for Flipping Distribution

Threshold Derivation (symmetric case)

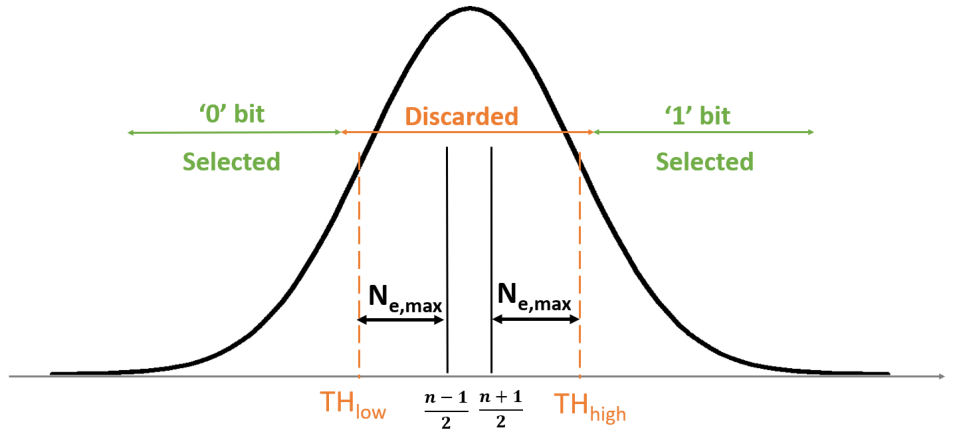
Push $N_{e,max}$ beyond the decoding boundary ($\frac{n}{2}$):

$$TH_{low} = \left\lfloor \frac{n}{2} \right\rfloor - N_{e,max}$$

$$TH_{high} = \left\lceil \frac{n}{2} \right\rceil + N_{e,max}$$

Guarantees: after up to $N_{e,max}$ flips, enrolled sequence R stays in its decoding region

TMVS tolerates up to $N_{e,max}$ flips



TMVS: Codebook

Codebook C: set of M codewords of length n

Trivial case (M=1): S^0: 000000…0 (n bits) → selection uses the number pf ones in R

General case (M>1): selection uses Hamming distance $d_H(R, S)$ between R and each codeword $S \in C$

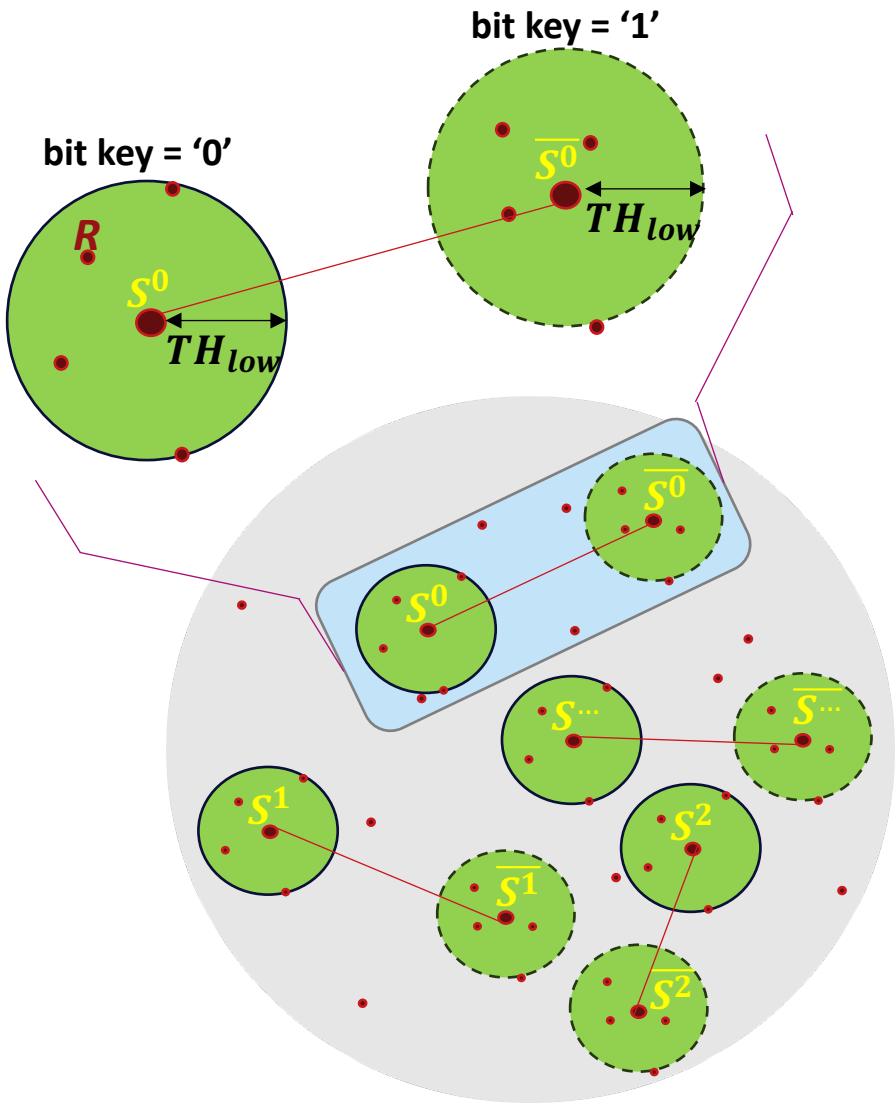
- Hamming distance (d_H) = number of different bits between two bit strings

More codewords → higher probability of selection → less SRAM required

Constraint: acceptance regions must not overlap

Minimum distance between any two codewords (and their complements)

→ Codebook can be hard-coded or stored in ROM



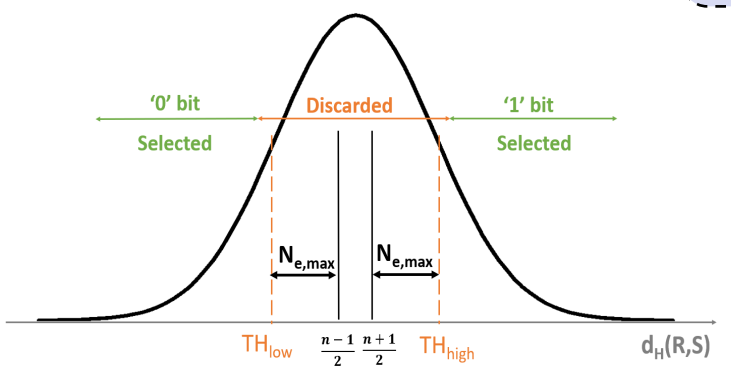
TMVS: Workflow

TMVS Setup

One-time, at design time

- Code length n
- Key length n_k
- Compute TH_{low}, TH_{high}
- Construct codebook:

$$C = \{S^1, \dots, S^M\}$$



Window of n bits
(selected)

1	0	0	0	1	1	0	1	0	0

TMVS Enrollment one-time, on device

Read n-bit sequence R from SRAM → $d_H(R,S) \le TH_{low}$? or $d_H(R,S) \ge TH_{high}$?

yes → Save address and S, Advance window by n → Repeat until n_k bits

no → S is last codeword in C?

yes → Advance window by 1 → Read n-bit sequence R from SRAM

no → Try next S → $d_H(R,S) \le TH_{low}$? or $d_H(R,S) \ge TH_{high}$?

Helper data (NVM)

TMVS Regeneration every device boot

Read helper data: address+ codeword S → Read noisy response R' → If $d_H(R',S) < \frac{n}{2}$: key bit '0' / If $d_H(R',S) > \frac{n}{2}$: key bit '1'

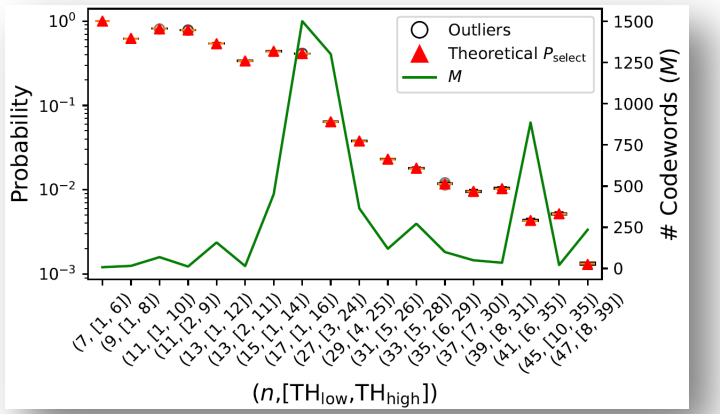
TMVS: Theoretical and Experimental Validation

Selection Probability P_{select}

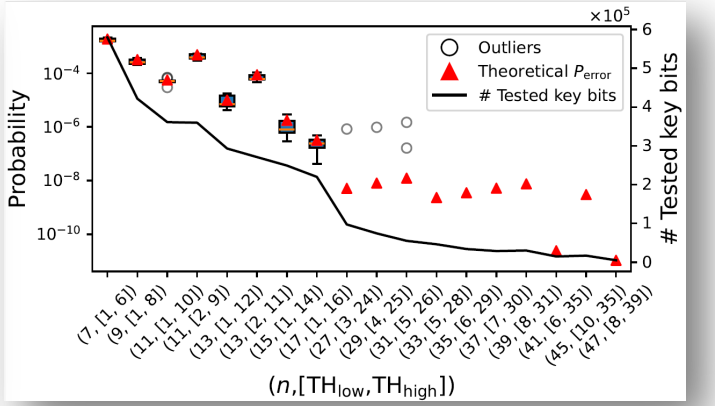
- Derived closed-form P_{select}
- ↗ with larger M
- ↘ with larger thresholds
- Governs required SRAM size:

  N_{SRAM}↗ when P_{select}↘

Experimental Validation on SCµM chip data



Theoretical vs. experimental P_{select}



Theoretical vs. experimental P_{error}

Decoding Error Probability P_{error}

- Derived closed-form P_{error}
- ↘ with larger thresholds
- Key failure probability (128-bit key):

  $P_{fail} = 1 - (1 - P_{error})^{128} < 10^{-6}$

Parameters

n=27

x_σ =7

M=1300

n_k =128

p_e=0.05

	Theoretical	Experimental
P_{fail}	$6.55 \cdot 10^{-7}$	$2.52 \cdot 10^{-6}$
N_{SRAM} (kB)	0.65	0.64

Valid theoretical model

Security Analysis of TMVS

I.I.D. SRAM PUF response bits


What is public	What attacker learns
Codeword	Which acceptance region
$TH_{low}, , TH_{high}$	Size of the region
SRAM address*	No content leakage

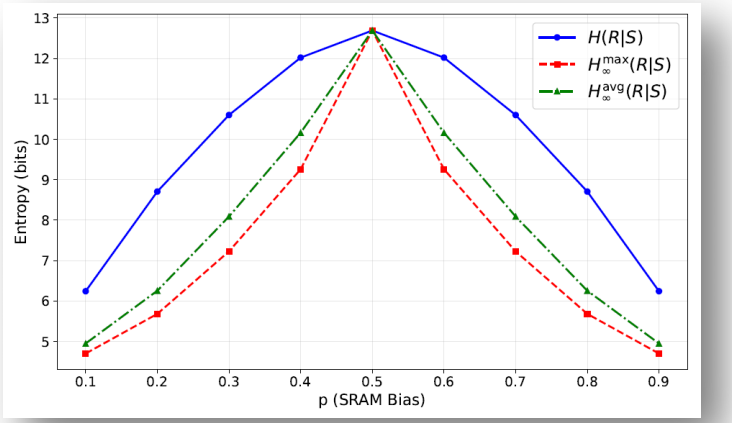
* Assuming SRAM isolation from untrusted software access

Helper data reveals acceptance region, not the response R

TMVS is not only reliable!
But also safe against leakage

Conditional Min-Entropy $H_\infty(R|S)$

- **Derived closed-form for any p**
- **Min-entropy quantifies remaining uncertainty**
- **Worst-case metric $H_\infty^{max}(R|S) > 1$ bit for all $p\epsilon(0,1)$**



Min-entropy metrics versus bias (p) for codebook parameters ($n = 27$, x_σ =7)

Secrecy Leakage $I(K;S)$

- **Derived closed-form for any p**
- **Leakage appears only when both conditions hold simultaneously:**
  - SRAM bias: p ≠ 0.5
  - Codeword imbalance: weight≠ n/2

Balanced codebook → I(K; S) = 0 for any bias

Non-I.I.D. SRAM PUF response bits

Validated on PUF4IoT public dataset [8]**:**

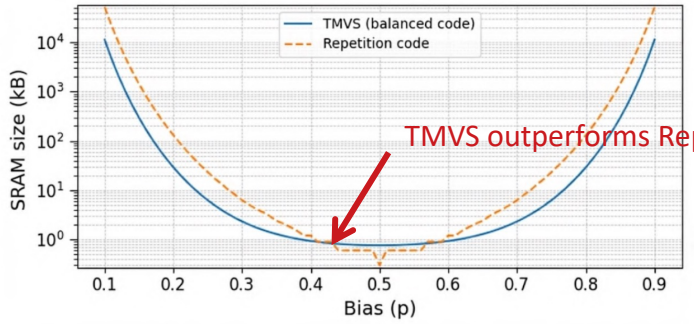
- **non-i.i.d. statistics**
- **294 STM32 Nucleo devices**

→ **empirical results confirm near-zero leakage !**

TMVS Contributions

Comparison with Repetition Codes

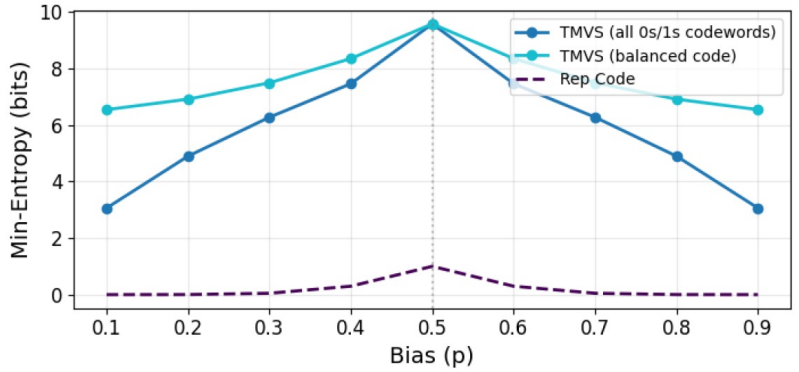
(same majority vote decoder, same single measurement)



	TMVS (26,1,5)	Rep (19,1,19)
Clock cycles (p=0.5)	**19,787**	23,541
SRAM at p=0.5	0.75 kB	**0.30 kB**
SRAM at p=0.3	**2.3 kB**	6.23 kB
SRAM at p=0.25	**6.22 kB**	23.16 kB

3.7x less!

SRAM efficiency comes from better entropy preservation:



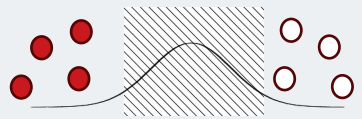
Why thresholding preserves entropy?

Repetition code

Helper data
→ 2 candidates only

Entropy collapses under bias

TMVS



Helper data
→ many feasible R

Entropy is preserved

TMVS solves the first research gap → a software–only stabilizer with
low leakage + simple majority vote decoder + single PUF measurement

IEEE Transactions on Information Forensics and Security (TIFS)

Outline

- State of the Art
- Methodology
- **C1** Single-Chip Motes and SRAM PUF: A Feasibility Study
- **C2** TMVS: Threshold-based Majority Voting Scheme for Robust SRAM PUFs
- **C3 Two-Stage TMVS**
- **C4** ODHD: On-Demand Helper Data Generation for Reliable NVM-Free Key Derivation from SRAM PUFs
- Conclusions

Limitation of TMVS

TMVS Limitation: SRAM Usage

- On SCμM, with BER≈5%
  → TMVS requires 0.64 kB of SRAM « total 64kB
- For higher BER
  → TMVS requires too many SRAM bits (>4kB)!

BER under a Temperature Cycle Test for SRAM fabricated in TSMC 65 nm LP CMOS [9]

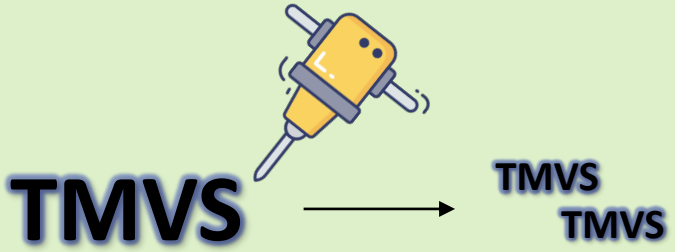
-40°C		+25°C		+85°C	
min	max	min	max	min	max
7%	8%	**5%**	6%	6.5%	**8%**

Challenge:

Can we keep the same
reliability target + majority-vote decoder + zero leakage
but use less SRAM?

Two-Stage TMVS

cascade two short TMVS stages



TS-TMVS: Two-Stage TMVS

TS-TMVS Setup

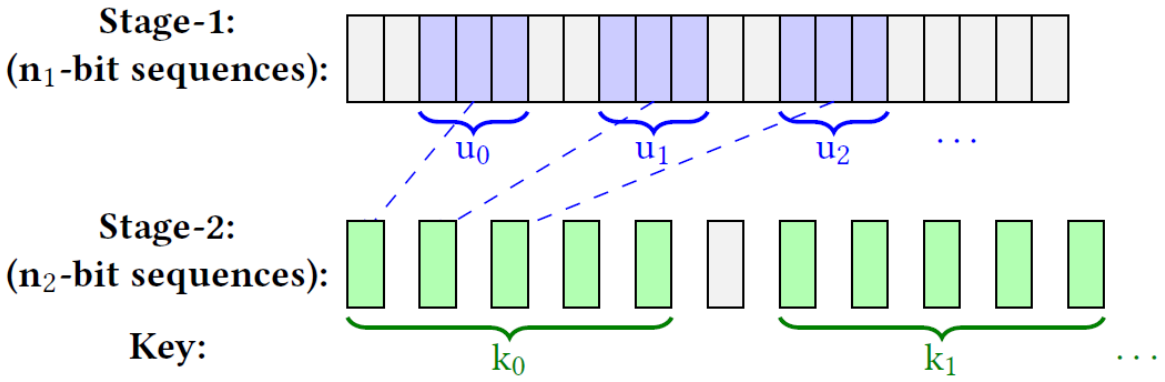
One-time, at design time

Each stage has its own setup:

Setup-1 (Stage-1): $(n_1, TH_{low,1}, TH_{high,1}, C_1)$

Setup-2 (Stage-2): $(n_2, TH_{low,2}, TH_{high,2}, C_2)$

TS-TMVS enrollment example with n_1=3 and n_2=5:



TS-TMVS Enrollment one-time, on device

Stage-1: Apply TMVS on raw SRAM using Setup-1 → Intermediate bits $u_i \in \{0, 1\}$

Stage-2: Apply TMVS on u_i bits using Setup-2 → Key bits $k_j \in \{0, 1\}$

Helper data = Stage-1 SRAM addresses of (u_i) + Stage-1 codewords indices + Stage-2 codeword indices (selected in Stage-2)

TS-TMVS Regeneration every device boot

Read helper data + noisy SRAM → Stage-1: majority vote → recover u_i bits → Stage-2: majority vote → recover k_j bits

Two-Stage TMVS: Theoretical Analysis

We derive closed-form expressions

- **Decoding error probability** $P_{error,\,TS}$
- **Memory requirements**
  - SRAM size $N_{SRAM,\,TS}$
  - Helper data size $N_{helper,\,TS}$

Single-stage TMVS Notation

- p_e: flipping probability
- n_k : key length
- $\boldsymbol{T} = [\mathbf{0}, \dots, \boldsymbol{TH_{low}}] \cup [\mathbf{0}, \dots, \boldsymbol{TH_{high}}]$
- $\boldsymbol{C}$: codebook with M codewords of code length $\boldsymbol{n}$
- $P_{error}(\boldsymbol{C}, \boldsymbol{T}, \boldsymbol{p_e})$: error probability
- $N_{SRAM}(\boldsymbol{n_k}; \boldsymbol{C}, \boldsymbol{T})$: SRAM size

$$P_{error,\,TS} = P_{error}(\boldsymbol{C_2}, \boldsymbol{T_2}, \underbrace{P_{error}(\boldsymbol{C_1}, \boldsymbol{T_1}, \underbrace{\boldsymbol{p_e}}_{\text{raw}})}_{\text{Stage-1}})}_{\text{Stage-2}}$$

$$N_{SRAM,\,TS} = N_{SRAM}(\underbrace{N_{SRAM}(\underbrace{\boldsymbol{n_k}}_{\text{final}}; \boldsymbol{C_2}, \boldsymbol{T_2})}_{\text{Stage-2}}; \boldsymbol{C_1}, \boldsymbol{T_1})}_{\text{Stage-1}}$$

$$N_{helper,\,TS} = \underbrace{3\boldsymbol{n_p}}_{\text{pointer size}} + \underbrace{\boldsymbol{n_k n_2} \lceil \log_2(N_{SRAM,\,TS}) \rceil}_{\text{addresses of intermediate bits}} + \underbrace{\boldsymbol{n_k n_2} \lceil \log_2(\boldsymbol{M_1}) \rceil}_{\text{codeword indices for intermediate bits}} + \underbrace{\boldsymbol{n_k} \lceil \log_2(\boldsymbol{M_2}) \rceil}_{\text{codeword indices for key bits}}$$

Two-Stage vs. Single-Stage TMVS

Memory Overhead: We compare SRAM and helper data size vs. p_e

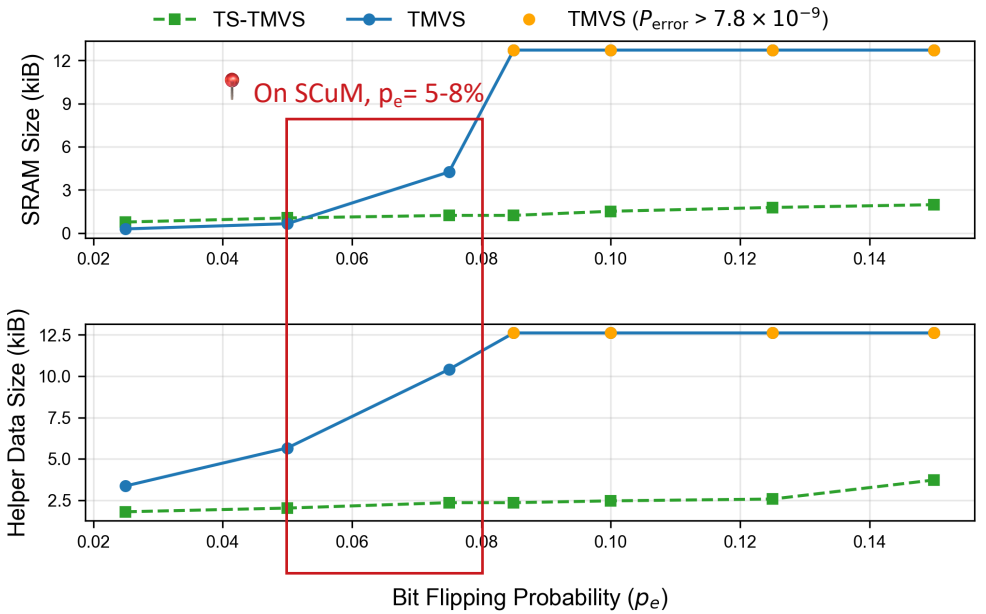
⬆️ Higher noise (p_e) → ⬆️ **longer codes (n) needed** to correct more errors → ⬇️ selection probability → ⬆️ SRAM consumption

SRAM size:

p_e (%)	TMVS (kB)	TS-TMVS (kB)	
5	0.65	1.1	TMVS slightly better
7.5	4.2	**1.2**	**3.5× reduction** ✅
8.5	>12 (❌ fails)	**1.2**	Only TS-TMVS works ✅

Helper data: TS-TMVS consistently lower → scales linearly with n_2 and log(SRAM size)

Two shorter codes → higher selection probability per stage → lower memory overhead than one long code



p_e	0.025	0.050	0.075	0.085	0.100	0.125	0.150
n	17	27	41	47	47	47	47
P_{error}	1.5 × 10^{-9}	5.1 × 10^{-9}	5.8 × 10^{-9}	1.9 × 10^{-8}	1.6 × 10^{-7}	2.8 × 10^{-6}	2.5 × 10^{-5}
n_1/n_2	7/7	9/7	11/7	11/7	13/7	15/7	11/11
$P_{error,TS}$	2.9 × 10^{-10}	5.8 × 10^{-10}	9.2 × 10^{-10}	5.1 × 10^{-9}	1.7 × 10^{-9}	4.2 × 10^{-9}	7.8 × 10^{-9}

P_{error} > 7.8 × 10^{-9}

Discussion: Comparison with Prior Work

All 9 approaches are implemented on SCµM

Balanced codebooks
→ zero secrecy leakage

Two-Stage TMVS

achieves the lowest clock cycle count among all zero secrecy leakage schemes

Trade-off:

larger PUF size than code-based schemes

→ acceptable on devices with sufficient SRAM

(e.g. SCµM: 64 kB)

Work	Codes	# PUF bits	P_{fail}	Clock cycles	Secrecy leakage
Fuzzy extractor	Rep(3,1,3) + BCH(255,99,47)	1530	5.4×10^{-7}	4,551,760	< 65.5
DSC	DSC(2.5,1) + RM(64,42,8)	640	3.5×10^{-7}	3,164,916	< 37.1
IBS	Rep(5,1,5) + BCH(127,64,21)	1778	4.6×10^{-7}	990,963	**0**
VN	Rep(8,1,8) + BCH(63,36,11)	2471	9.7×10^{-8}	703,771	**0**
DSC+RMCC	DSC(2.75,1) + RM(128,99,8)	704	2.5×10^{-7}	3,477,369	< 0.06
DSC+RMCC	DSC(5.75,1) + RM(64,57,4)	1472	6.6×10^{-7}	7,060,515	< 0.01
Polar CC	Polar(512,134,3)	512	2.8×10^{-7}	414,325	< 2.36
ERFE	ERFE + Rep(5,1,5)	> 43,726	2.6×10^{-6}	> 239,584	—
TS-TMVS (this work)	**TMVS(10,1,3) + TMVS(12,1,3)**	**25,838**	**9.2×10^{-7}**	**247,867**	**0**

Comparison at: $p_e = 0.1$, bias $p = 0.54$, key failure probability $\leq 10^{-6}$, 128-bit key

Outline

- State of the Art
- Methodology
- **C1** Single-Chip Motes and SRAM PUF: A Feasibility Study
- **C2** TMVS: Threshold-based Majority Voting Scheme for Robust SRAM PUFs
- **C3** Two-Stage TMVS
- **C4** **ODHD: On-Demand Helper Data Generation for Reliable NVM-Free Key Derivation from SRAM PUFs**
- Conclusions

ODHD: Motivation

Software-based stabilizers (including TMVS) require:
persistent helper data storage in NVM

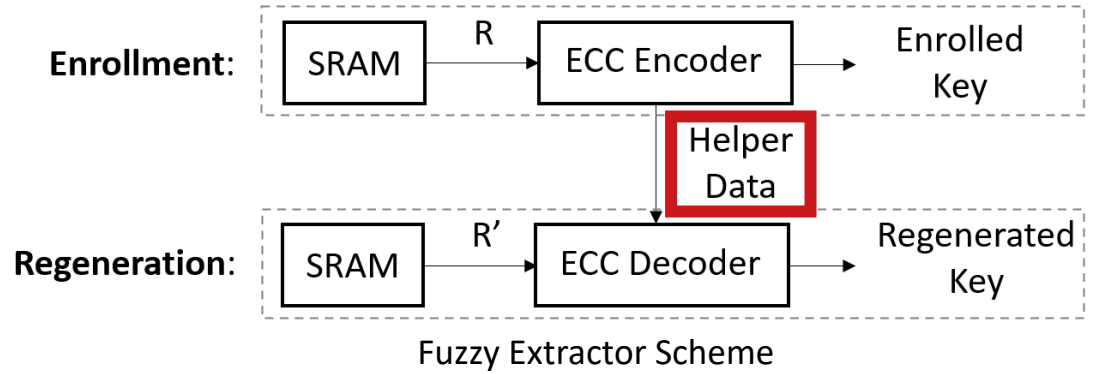
Challenge: No NVM available

Single-chip motes like SCµM-3C have no writable NVM

- Pure CMOS logic process (cost-optimized)
- Limited ROM reserved for firmware only

Existing NVM-free Solutions

- **Remote NVM (server) for helper data** → not scalable
- **Dark Bit:** repeated measurements to find stable cells → errors still too high



Fuzzy Extractor Scheme

new enrollment → ≠ Helper Data → ≠ enrolled key

✗ Cannot be deployed on NVM-free platforms

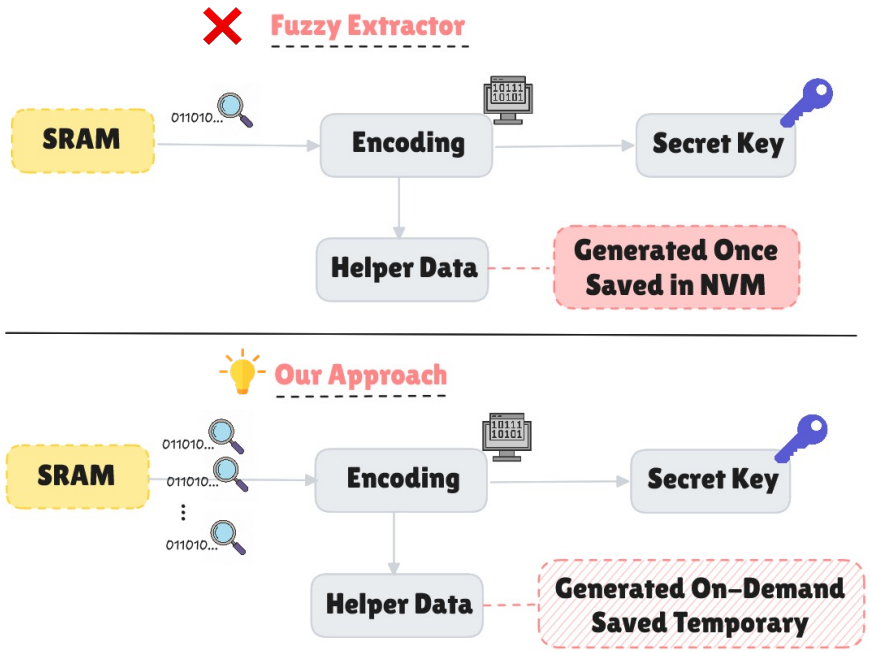
Goal

Ensure SRAM PUF reliability without using any NVM storage

ODHD: On-Demand Helper Data Generation

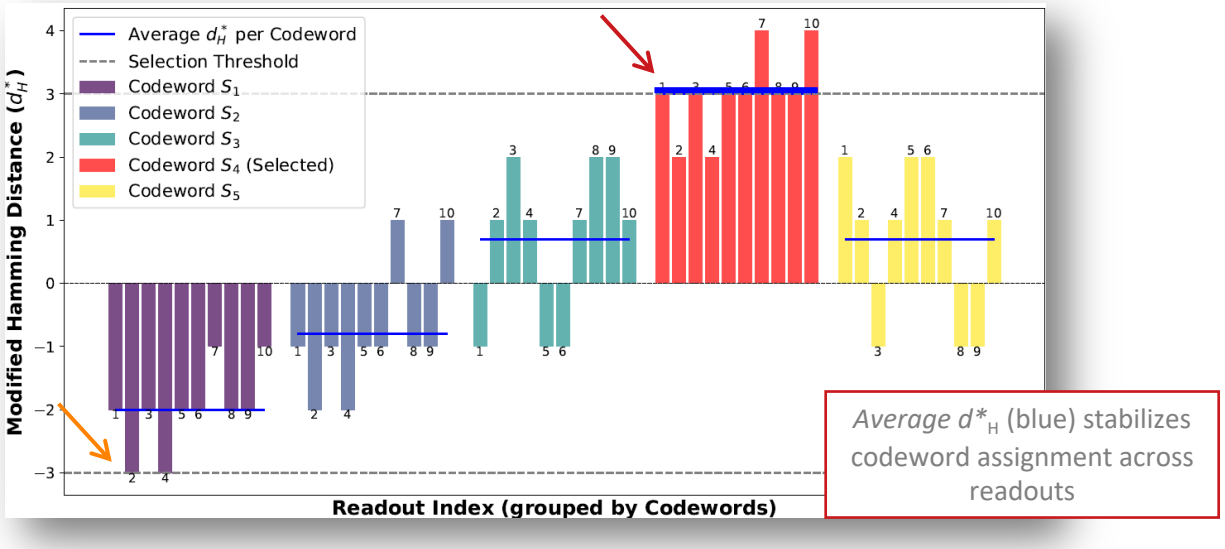
Key Innovation:

1. **Generate consistent helper data** on-demand using multiple SRAM measurements
2. **Store it temporarily** in volatile memory during key derivation



Based on TMVS Algorithm:

- Enrollment: Multiple SRAM readouts → averaged Hamming distances → threshold selection
- Regeneration: Single SRAM readout → decoder reconstructs key



Current Status: We performed full experimental evaluation, including enrollment and regeneration reliability, and comparison with Dark Bit baseline. **ODHD enhances reliability!**

Computers & Security (Elsevier)

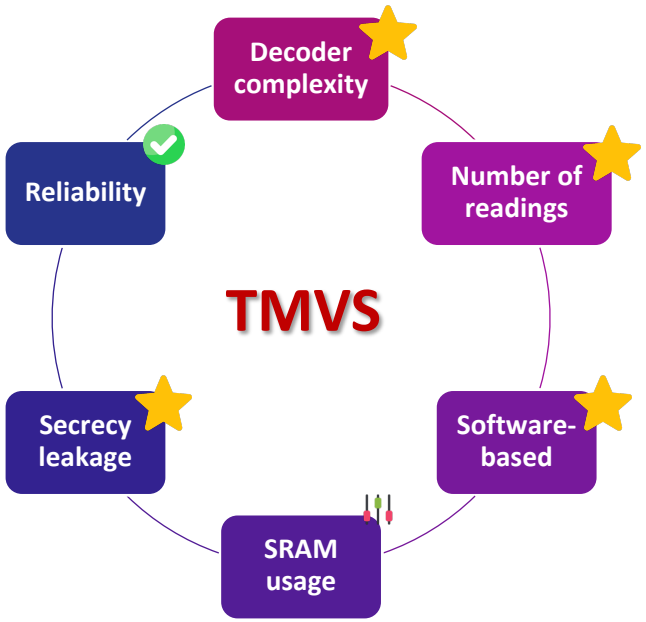
Outline

- State of the Art
- Methodology
- **C1** Single-Chip Motes and SRAM PUF: A Feasibility Study
- **C2** TMVS: Threshold-based Majority Voting Scheme for Robust SRAM PUFs
- **C3** Two-Stage TMVS
- **C4** ODHD: On-Demand Helper Data Generation for Reliable NVM-Free Key Derivation from SRAM PUFs
- **Conclusions**

Conclusions

Grand Challenge

How can we secure Smart Dust devices despite their extreme constraints?



Physical Security Analysis

- Single-chip motes harder to attack
- Critical gap: no writable NVM

SRAM PUF Feasibility on SCµM

We evaluate SRAM properties on SCµM and demonstrated that the SRAM is feasible as PUF for secret key generation

TMVS & TS-TMVS

SRAM PUF stabilizers: software-only, zero secrecy leakage, single readout, simplest decoder, achieving failure probability $<10^{-6}$

ODHD

NVM-free key derivation: on-demand helper data, without persistent storage

References

[1] Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data,” in Advances In Cryptology EUROCRYPT 2004: International Conference On The Theory And Applications Of Cryptographic Techniques, Berlin, Heidelberg: Springer, 2004, pp. 523–540.

[2] M. Hiller, M.-D. Yu, and M. Pehl, “Systematic low leakage coding for physical unclonable functions,” in Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, 2015.

[3] M. Hiller and A. G. Önalan, “Hiding secrecy leakage in leaky helper data,” in International Conference on Cryptographic Hardware and Embedded Systems, Springer, 2017.

[4] M.-D. Yu and S. Devadas, “Secure and Robust Error Correction for Physical Unclonable Functions,” IEEE Design & Test of Computers, 2010.

[5] P. Koeberl, J. Li, and W. Wu, “A Spatial Majority Voting Technique to Reduce Error Rate of Physically Unclonable Functions,” in International Conference on Trusted Systems (INTRUST), 2013.

[6] J. Delvaux, D. Gu, I. Verbauwhede, M. Hiller, and M.-D. Yu, “Efficient fuzzy extraction of PUF-induced secrets: Theory and applications,” in International Conference on Cryptographic Hardware and Embedded Systems, Springer, 2016.

[7] F. Armknecht, R. Maes, A.-R. Sadeghi, B. Sunar, and P. Tuyls, “Memory leakage-resilient encryption based on physically unclonable functions,” Towards Hardware-Intrinsic Security: Foundations and Practice, 2010.

[8] Vinagrero, S., Martin, H., de Bignicourt, A. et al. SRAM-Based PUF Readouts. Sci Data 10, 333 (2023). https://doi.org/10.1038/s41597-023-02225-9

[9] Maes, R., Rozic, V., Verbauwhede, I., Koeberl, P., Van der Sluis, E., & Van der Leest, V. (2012, September). Experimental evaluation of physically unclonable functions in 65 nm CMOS. In 2012 Proceedings of the ESSCIRC (ESSCIRC) (pp. 486-489). IEEE.

Contributions

Journal Articles

1. ODHD: On-Demand Helper Data Generation for Reliable NVM-Free Key Derivation from SRAM PUF. Sara Faour, Filip Maksimovic, Thomas Watteyne, Kristofer Pister, Mališa Vučinić. **Computers & Security**, **2026.**
2. TMVS: Threshold-based Majority Voting Scheme for Robust SRAM PUFs. Sara Faour, Filip Maksimovic, David Burnett, Paul Muhlethaler, Thomas Watteyne, Kristofer Pister, Mališa Vučinić. **IEEE Transactions on Information Forensics and Security (TIFS)**, **2026**.

Conference Papers

1. ODHD: On-Demand Helper Data Generation for Reliable NVM-Free Key Derivation from SRAM PUF. Sara Faour, Mališa Vučinić, Filip Maksimovic, Thomas Watteyne, Kristofer Pister. **IEEE International Conference on Information Security and Cryptology (ISC)**, Ankara, Türkiye, 22–23 October, **2025**.
2. Two-Stage Threshold-based Majority Voting Scheme (TS-TMVS) for Robust SRAM PUFs. Sara Faour, Mališa Vučinić, Thomas Watteyne, Kristofer Pister. **Workshop on Crystal-Free/-Less Radio and System-based Research for IoT(CrystalFreeIoT)**, International Conference on Embedded Wireless Systems and Networks (EWSN), Leuven, Belgium, 22 September, **2025**.
3. TMVS: Threshold-based Majority Voting Scheme for Robust SRAM PUFs. Sara Faour, Mališa Vučinić, Filip Maksimovic, David Burnett, Paul Muhlethaler, Thomas Watteyne, Kristofer Pister. **IEEE Symposium on Computers and Communications (ISCC)**, Paris, France, 26–29 June, **2024**.
4. Single-Chip Motes and SRAM PUF: Feasibility Study. Sara Faour, Blaz Korecic, Mališa Vučinić, Filip Maksimovic, David Burnett, Paul Muhlethaler, Thomas Watteyne. **Workshop on Crystal-Free/-Less Radio and System-based Research for IoT (CrystalFreeIoT)**, Cyber-Physical Systems and Internet-of-Things Week (CPS-IoT Week), Hong Kong, 13–16 May, **2024**.
5. Electromagnetic Side Channel Leakage Improvements Using Free-Running Oscillator Clock Reference. Jacob N. Louie, Sara Faour, David C. Burnett. **Workshop on Crystal-Free/-Less Radio and System-based Research for IoT (CrystalFreeIoT)**, Cyber-Physical Systems and Internet-of-Things Week (CPS-IoT Week), Hong Kong, 13–16 May, **2024**.
6. Implications of Physical Fault Injections on Single Chip Motes. Sara Faour, Mališa Vučinić, Filip Maksimovic, David Burnett, Paul Muhlethaler, Thomas Watteyne, Kristofer Pister. **IEEE World Forum on Internet of Things (WF-IoT)**, Aveiro, Portugal, 12–27 October **2023**.

Patent

Procédé d'extraction d'une séquence de bits d'une fonction physique non clonable. Sara Faour, Mališa Vučinić. Patent submitted. N° de dépôt : FR2509204.

Open-source Contributions

Open source implementation of TMVS, Two-Stage TMVS, ODHD and their evaluation (https://github.com/Sara-Fa/SRAM-PUF-key-generation).