hardwear.io
Hardware Security Conference and Training

Exploring Dual Edges of SRAM Data Remanence in SoCs:

Covert Storage and Exfiltration Risks in TEE

Jubayer Mahmod

About me

Senior Engineer @Lucid Motors’ RedTeam

Hardware Security PhD @Virginia Tech
Advised by Dr. Matthew Hicks

My expertise

Hardware-Oriented System Security
Cloud FPGA Security
Fake Chip Detection and Anti-Counterfeit Framework Design

LinkedIn X

@jubayer0175

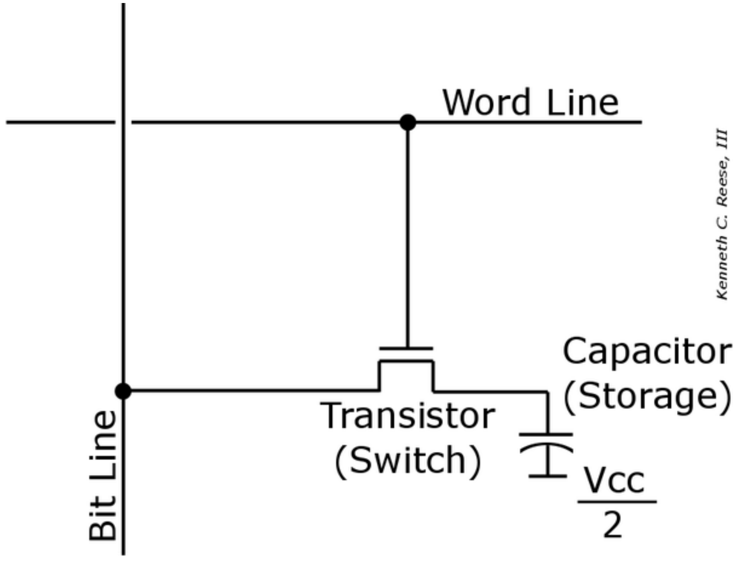
Disclaimer

The content of this presentation is based on my doctoral research conducted at Virginia Tech. All information shared here is publicly available from various publication venues.

It does not contain any proprietary technology and does not reflect the opinions or positions of Lucid.

Volatile memory does not forget data instantly

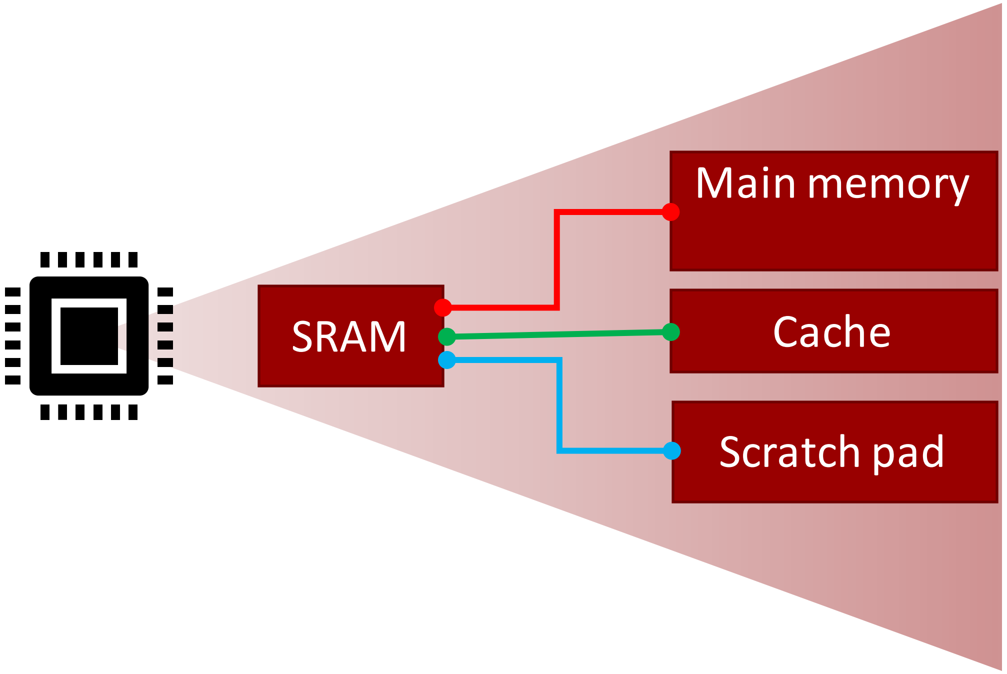
Data remanence: when a memory device retains information past when it is <u>assumed to no longer exist</u>



Typical DRAM cell

- **Battery-less timekeeping [USENIX’12, ASPLOS’2020]**
- **True random number generation [HOST’2016]**
- **Crypto-Key generation [ISLPED`17]**
- **Cold boot attacks [USENIX’08, DSD’18]**

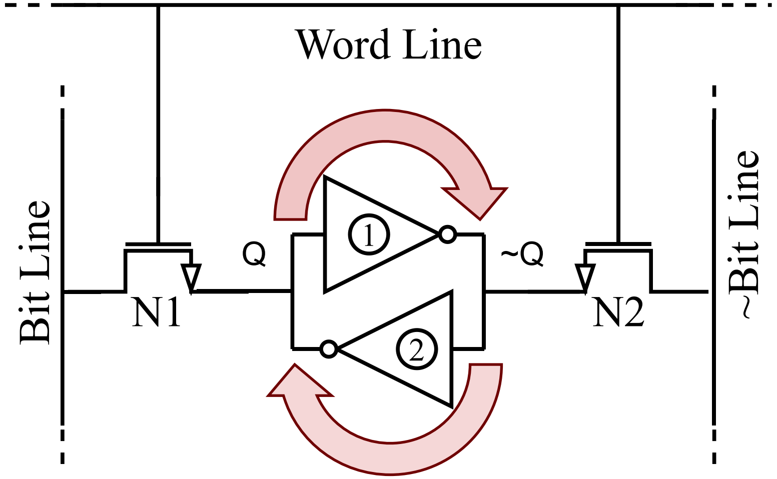
Static Random Access Memory (SRAM)

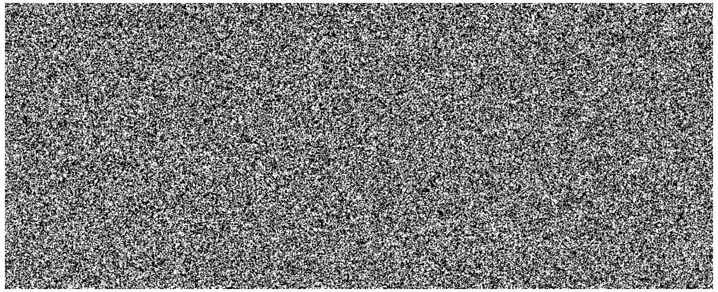


Positive feedback

No refreshing

Ultra-fast



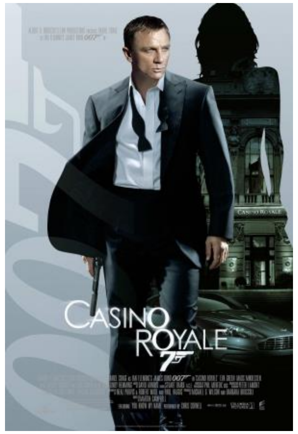


SRAM startup state: digital window into the analog world

InvisibleBits

Jubayer Mahmod and Matthew Hicks. ***Invisible Bits: Hiding Secret Messages in SRAM's Analog Domain***. In Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '22), February 28-March 4, 2022, Lausanne, Switzerland

Why Steganography?



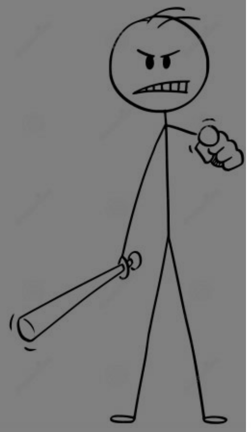
Agent 007

Not going to share!

Password?

Let me summon the best password cracking machine ever!

What do you have?



Steganography is information hiding technique

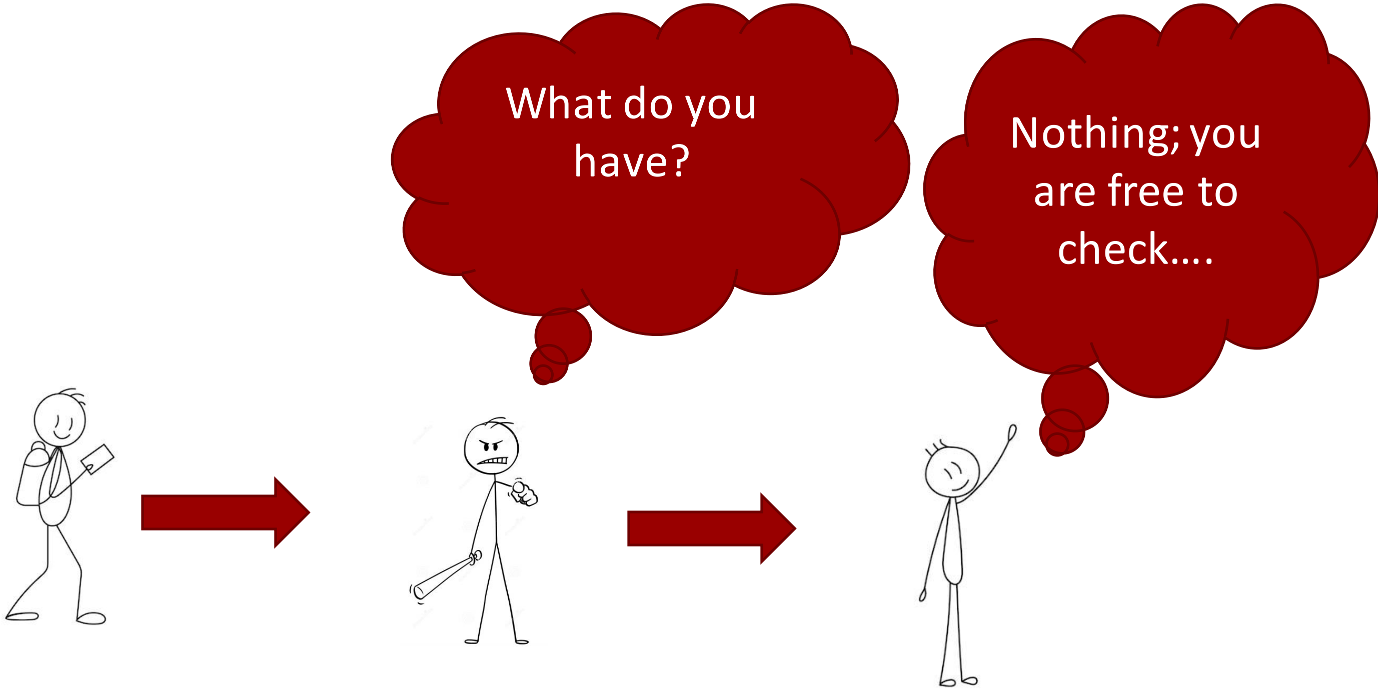
Hide information in “plain sight” to allow plausible deniability of its existence.

Typical steganography media

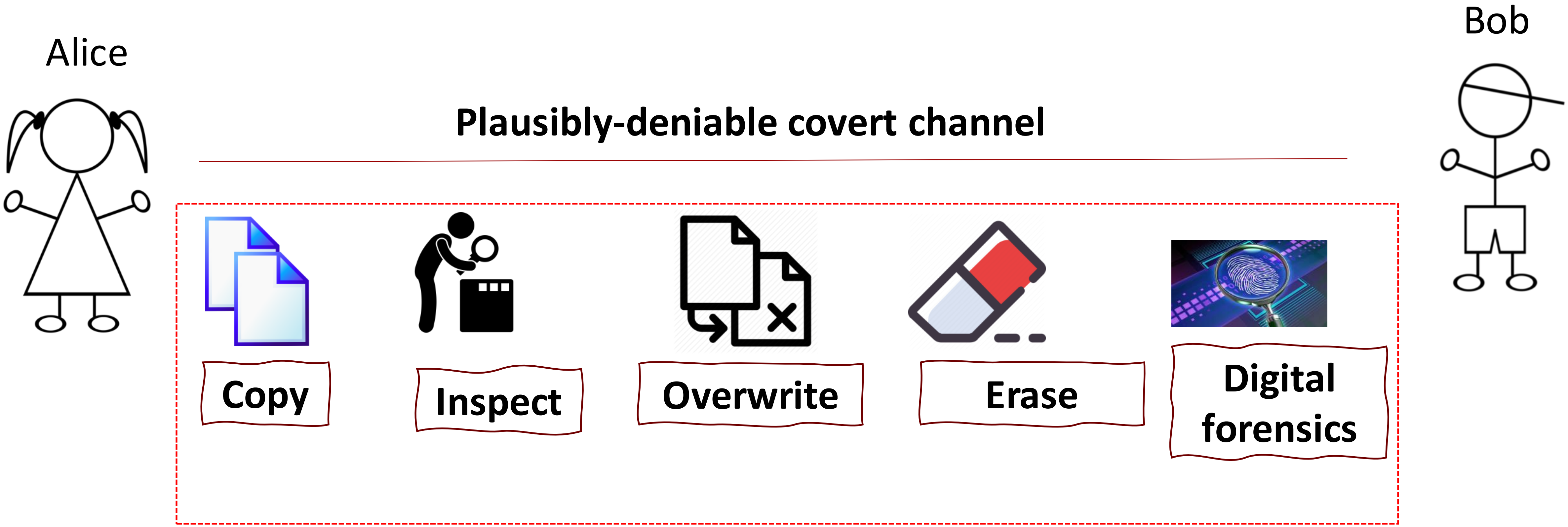
- **Audio files**
- **Images**
- **Flash memory’s hidden sectors**

Why?

- **To avoid oppressive government scrutiny**
- **To keep crypto wallet “invisible”**



Threat model



Alice

Bob

Plausibly-deniable covert channel

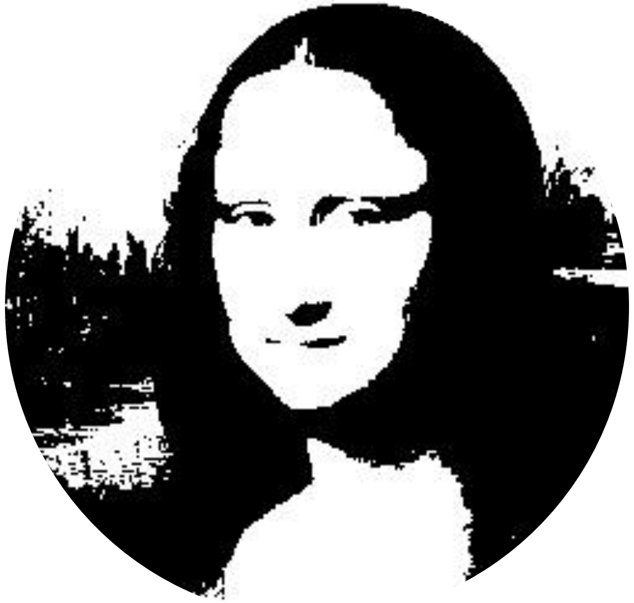
Copy

Inspect

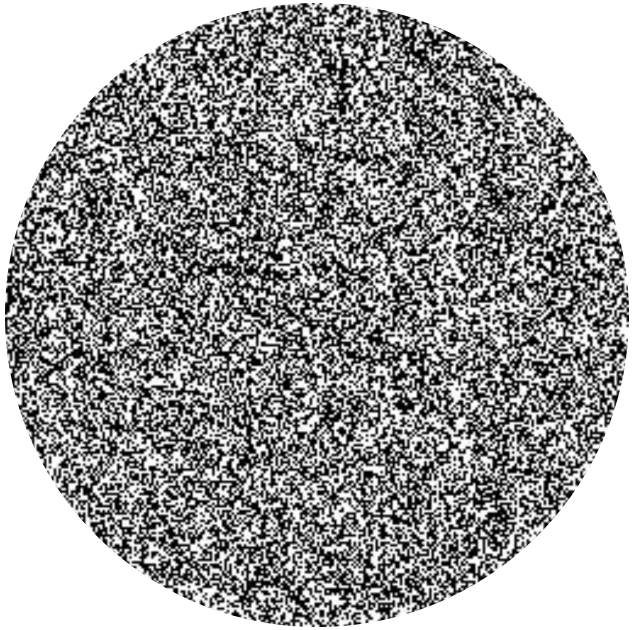
Overwrite

Erase

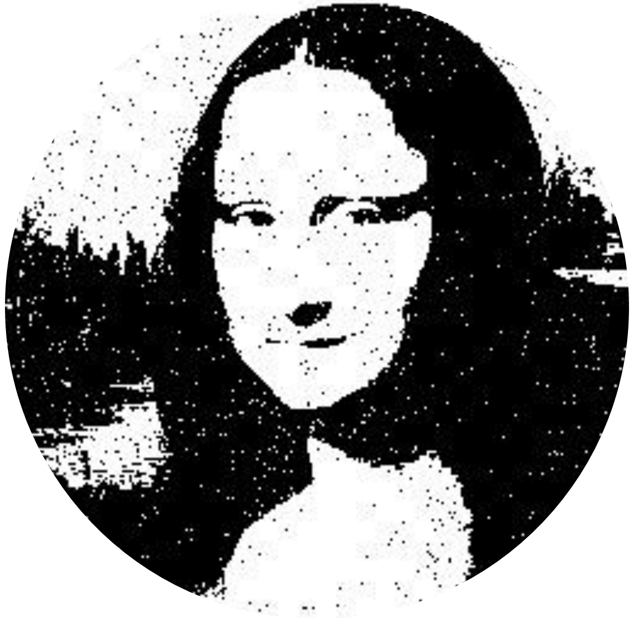
Digital forensics



Message to Encode



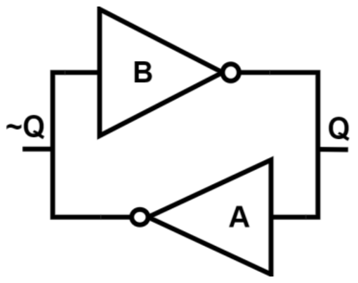
SRAM’s power-on state

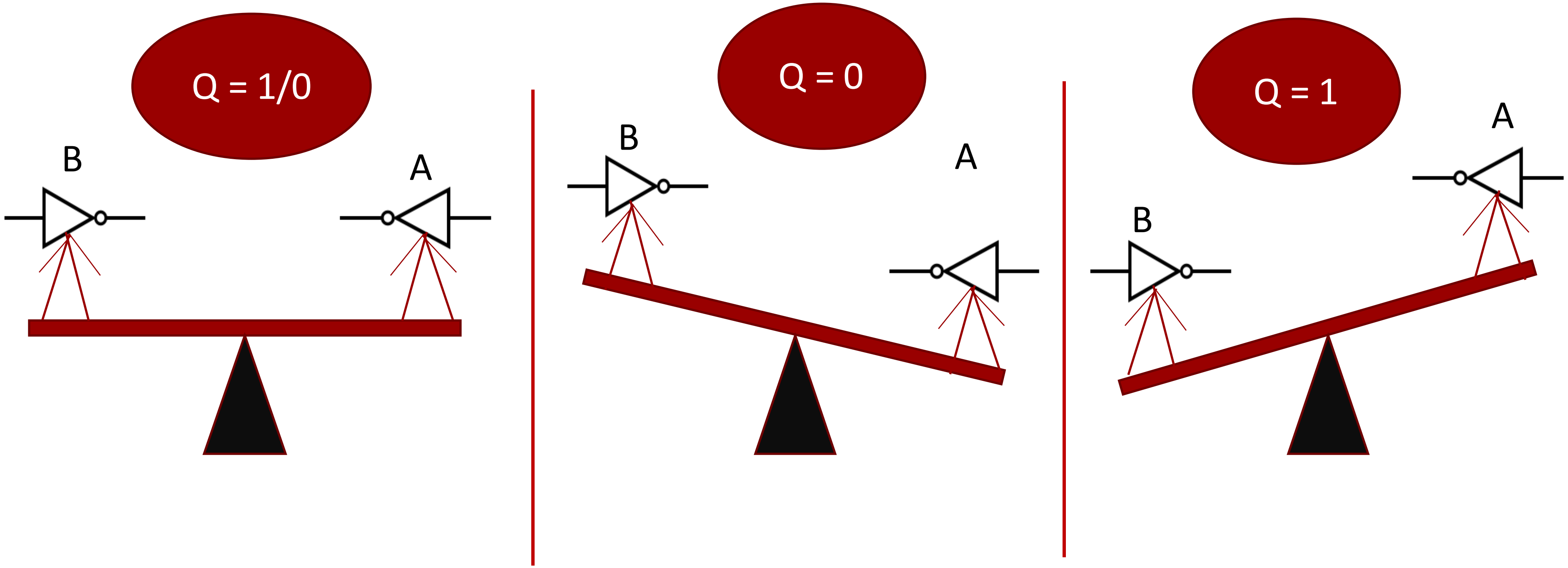


Power-on state after message encoding

InvisibleBits overview

SRAM cell and its power on-state

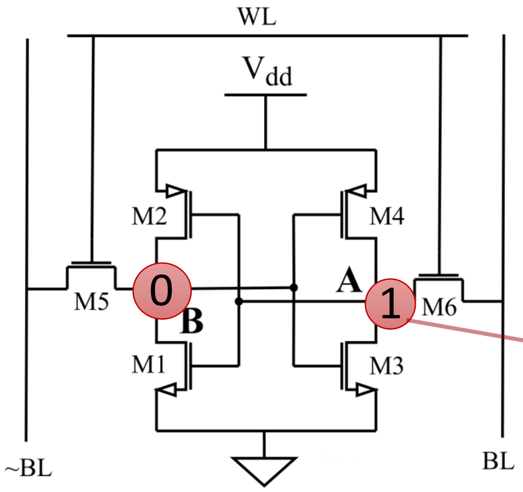




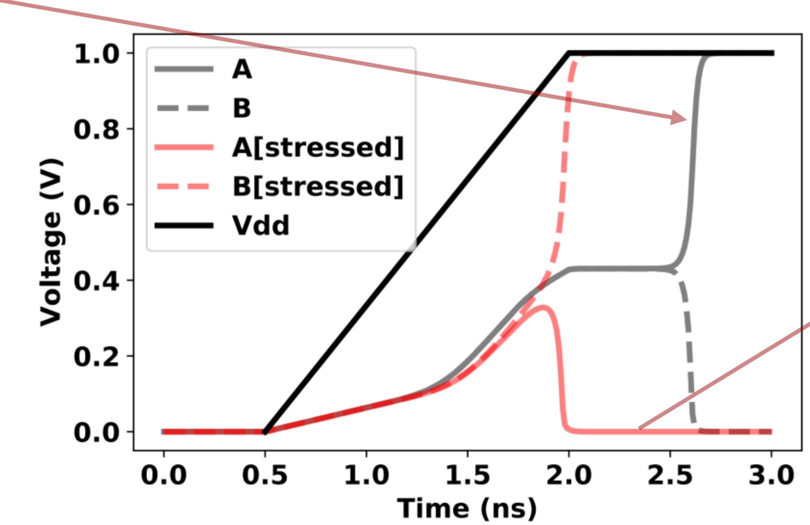
- Designed to be balanced.
- At startup, one of the inverters wins the race condition.

In this case, by winning I mean relatively faster rise time of an inverter's pull up network.

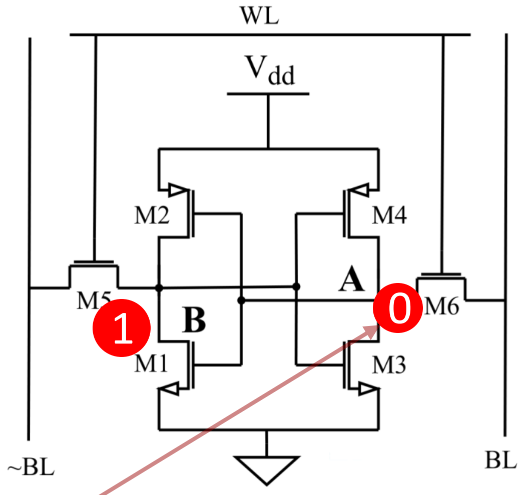
Aging burns in data in SRAM cell



Before aging



A 45nm SRAM cell's transient analysis before and after aging



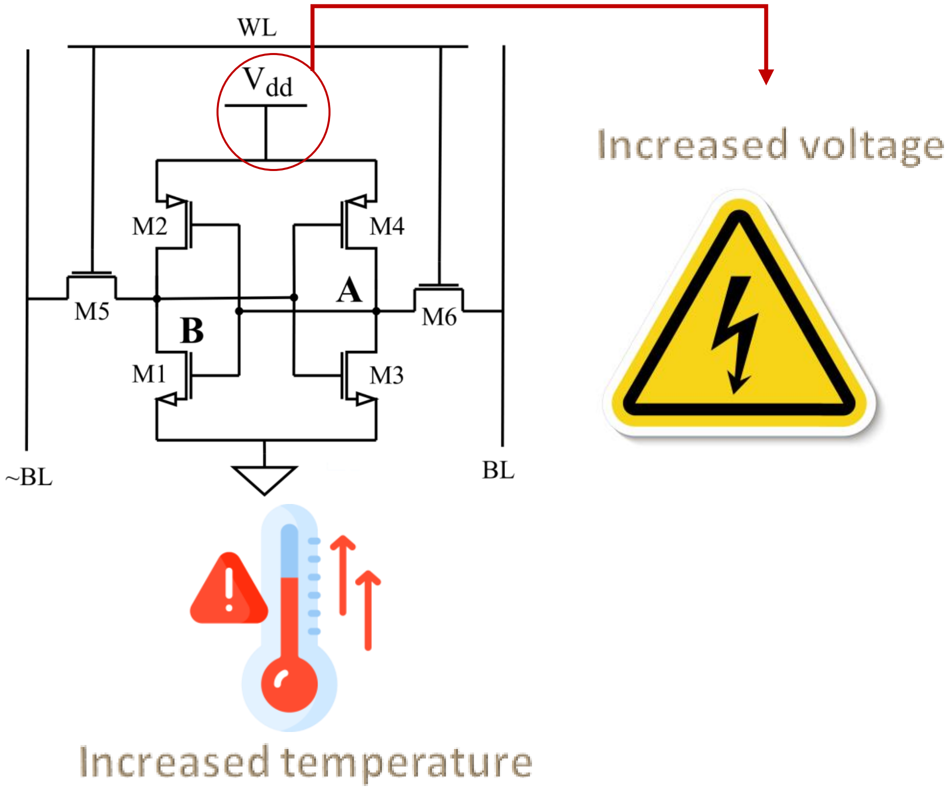
After aging

Like negative in photography, payload gets hidden as complement

Accelerating aging condition

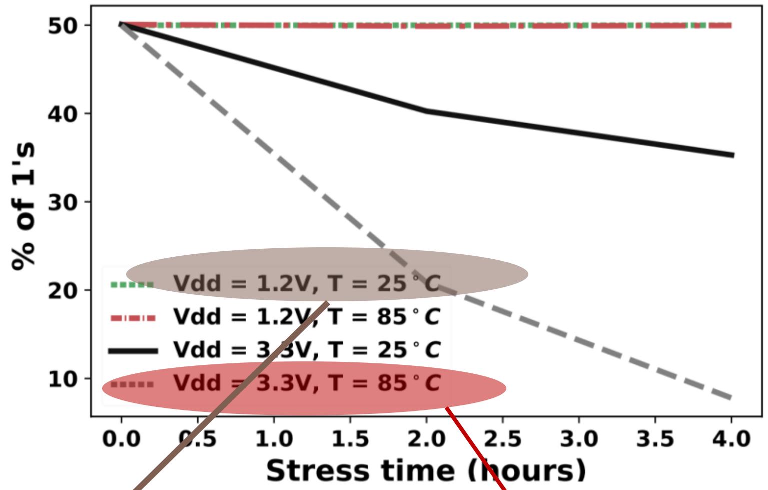
Aging takes decades to impact performance.

- Aging SRAM with all 1s in it**, reduces** number of 1s in subsequent power on
- Aging **effect is logarithmic**, over time rate of change decreases



Increased voltage

Increased temperature



Nominal operating condition

Best aging condition

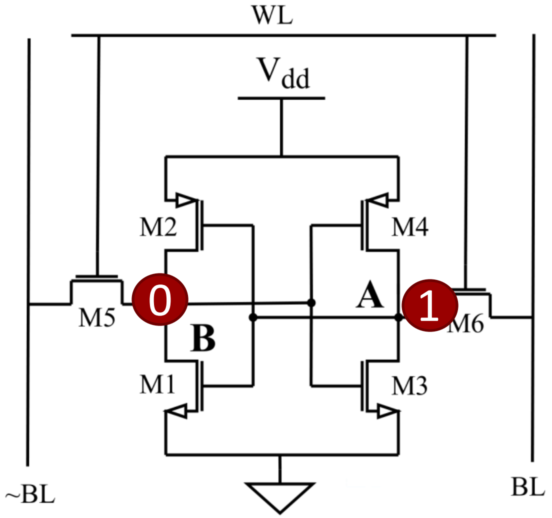
Data encoding process

1. A = 1
Write the value

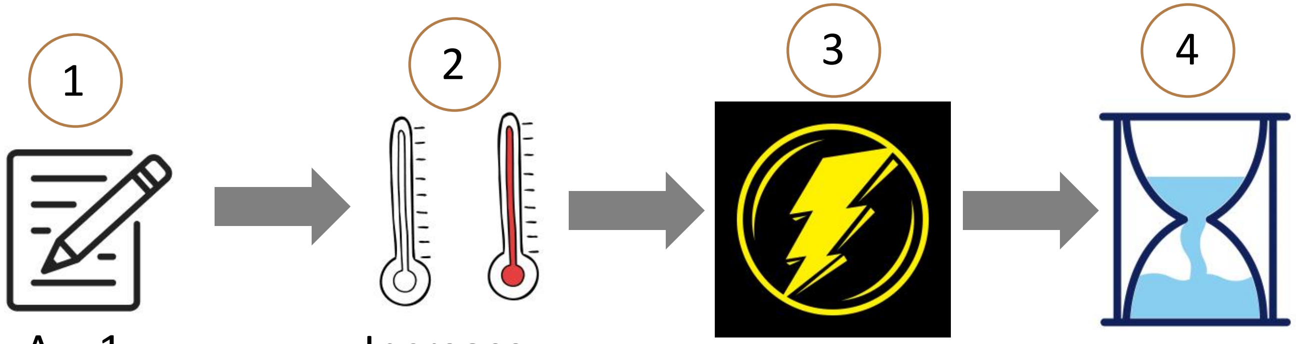
2. Increase
the temperature

3. Increase the
operating voltage

4. Wait



Data encoding process



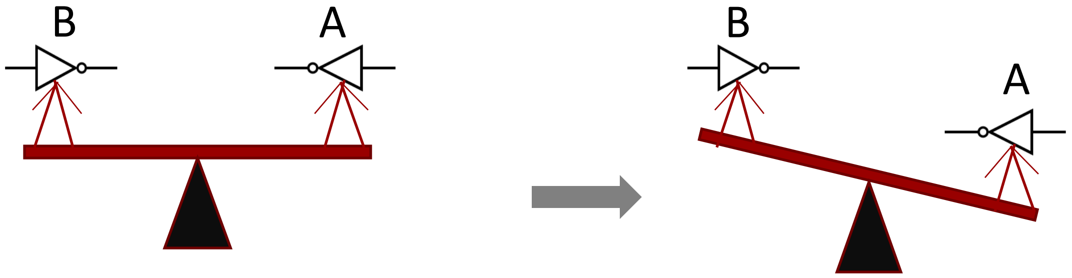
A = 1
Write the value

Increase
the temperature

Increase the
operating voltage

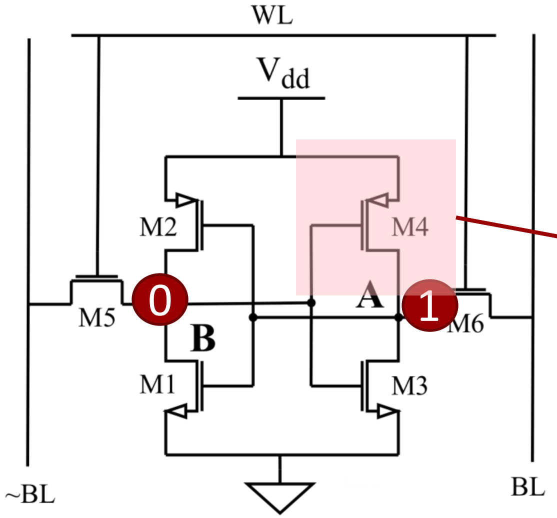
Wait

Result



Before encoding

After encoding



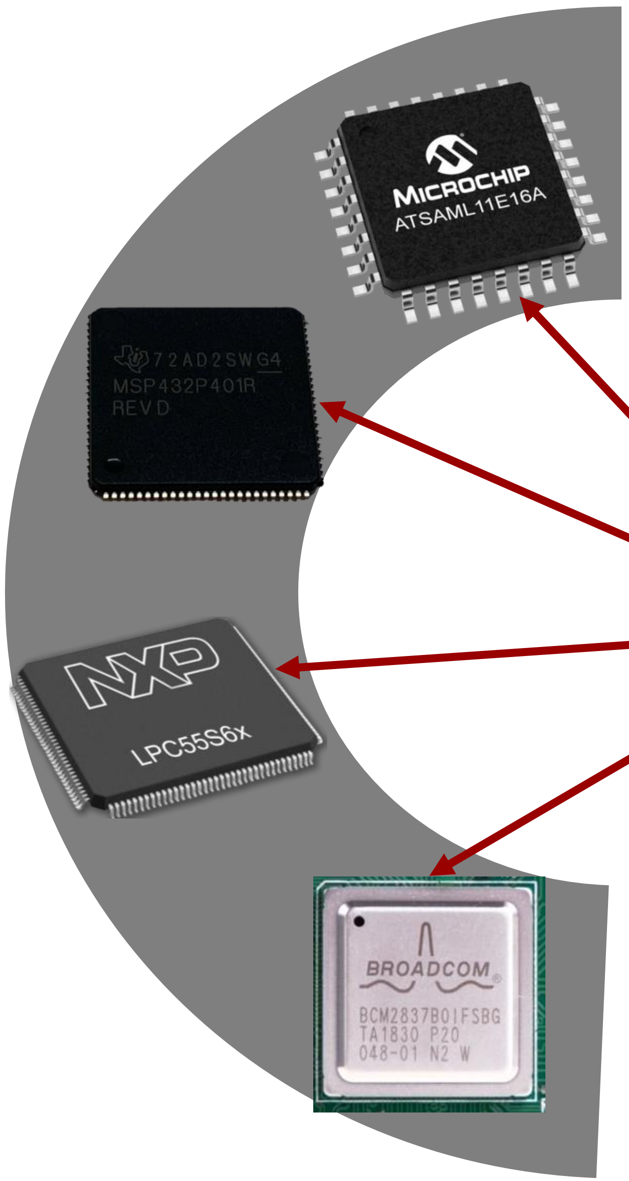
Slowing down the startup speed of this PMOS

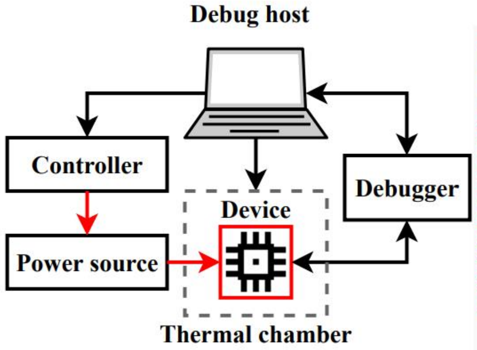
InvisibleBits evaluation

Retrieval error?

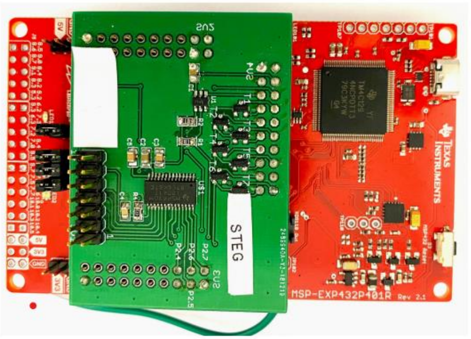
Plausibly deniable?

Errors without any ECC





(a)



(b)

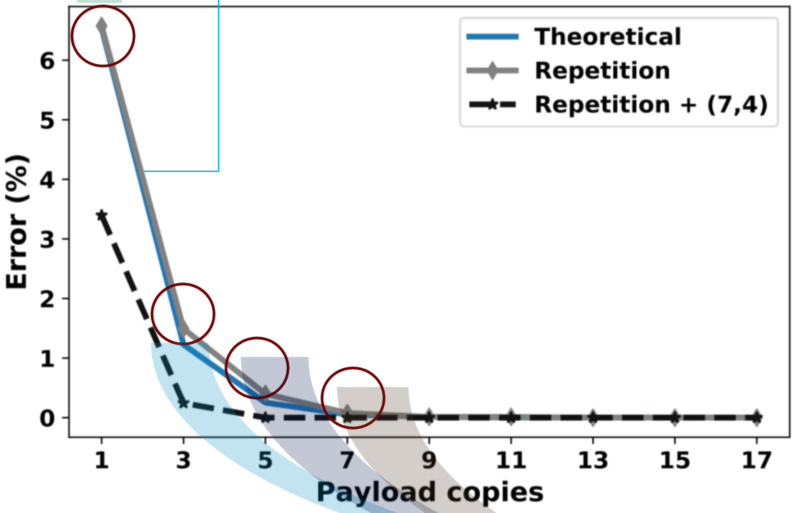
Device	SRAM usage	$V_{acc.}$	$T_{acc.}$	Accuracy	Encoding time
ATSAML11E16A	Main memory	4.8V	$85^\circ C$	97.2%	16 hours
MSP432P401	Main memory	3.3V	$85^\circ C$	93.5%	10 hours
LPC55S69JBD100	Main memory	5.5V	$85^\circ C$	88.5%	24 hours
BCM2837	Cache	2.2V	$85^\circ C$	79.2%	120 hours

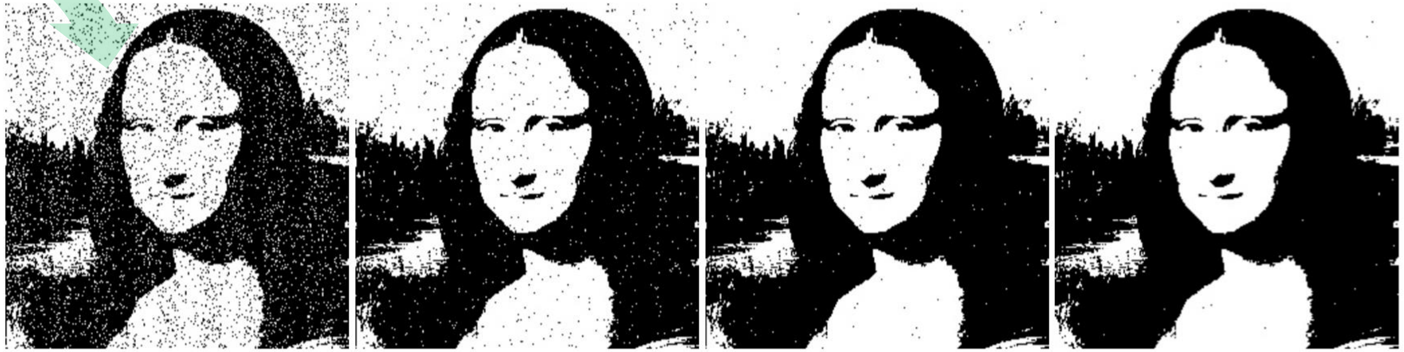
*Acceleration voltages are derived from experiments & datasheets

Improving accuracy

$$error = 1 - \sum_{i=\frac{(n+1)}{2}}^{n} \binom{n}{i} \times p^{i} \times (1-p)^{n-i}$$

Bernoulli trials





1 copy

3 copies

5 copies

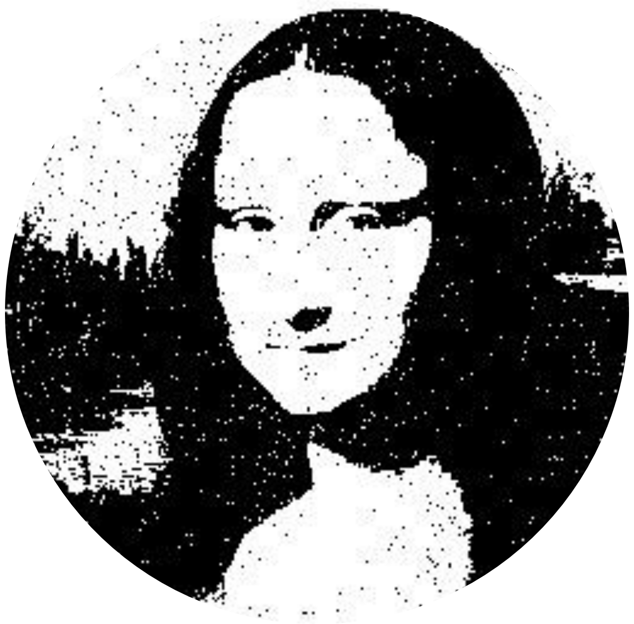
7 copies

Plausible deniability

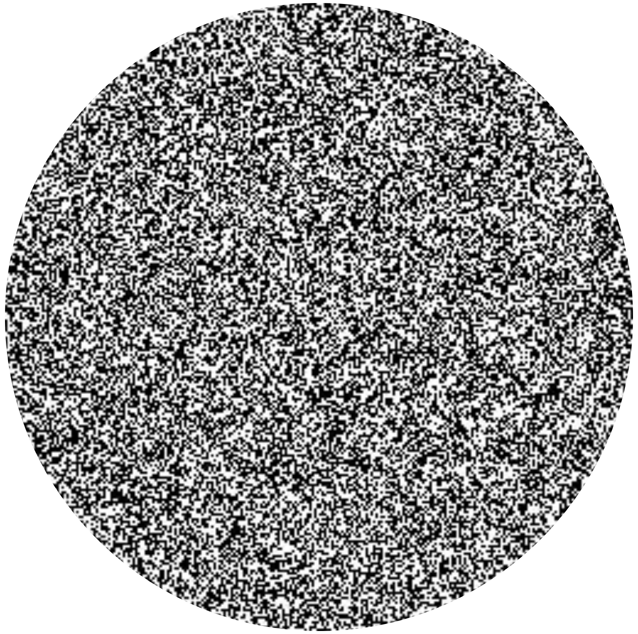
The information is **not really stored (digitally)**

System is free to **use entire SRAM** – No restriction

Condition	Spatial autocorrelation	Mean power-on state bias
Clean device	0.001	0.500
Hidden message	0.502	0.537
Hidden message(Encrypted)	0.008	0.501

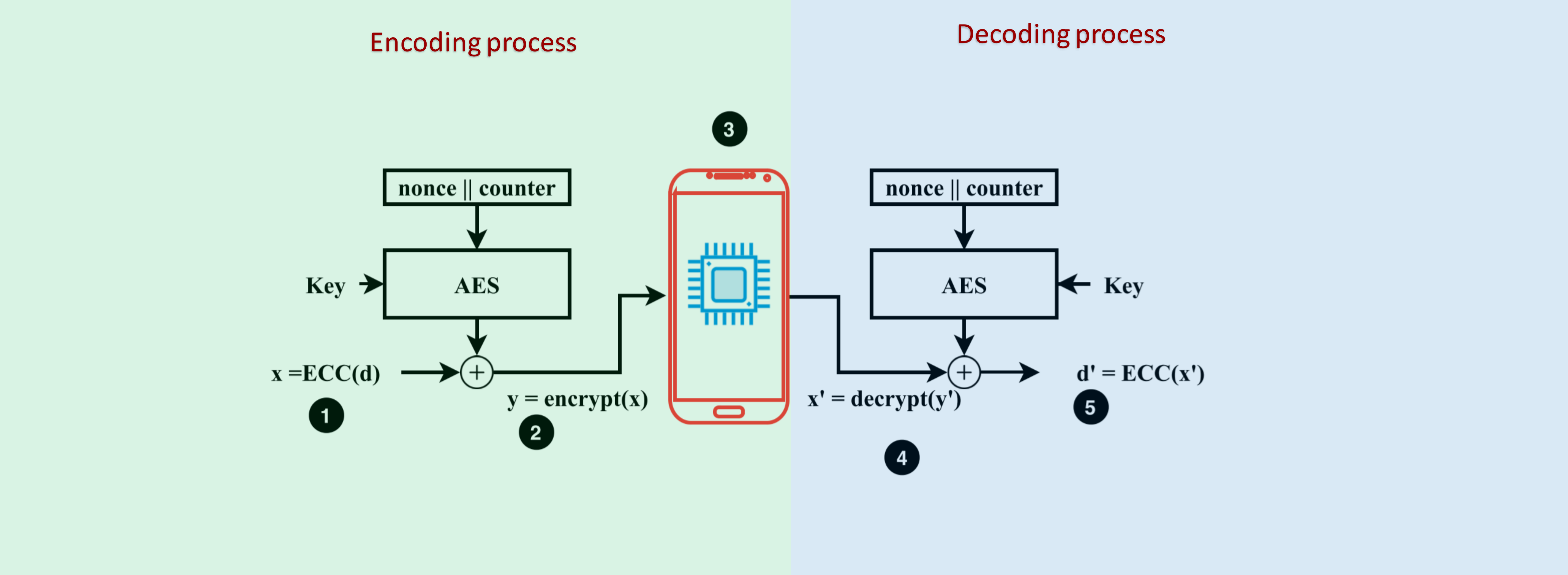


Plain-text encoding



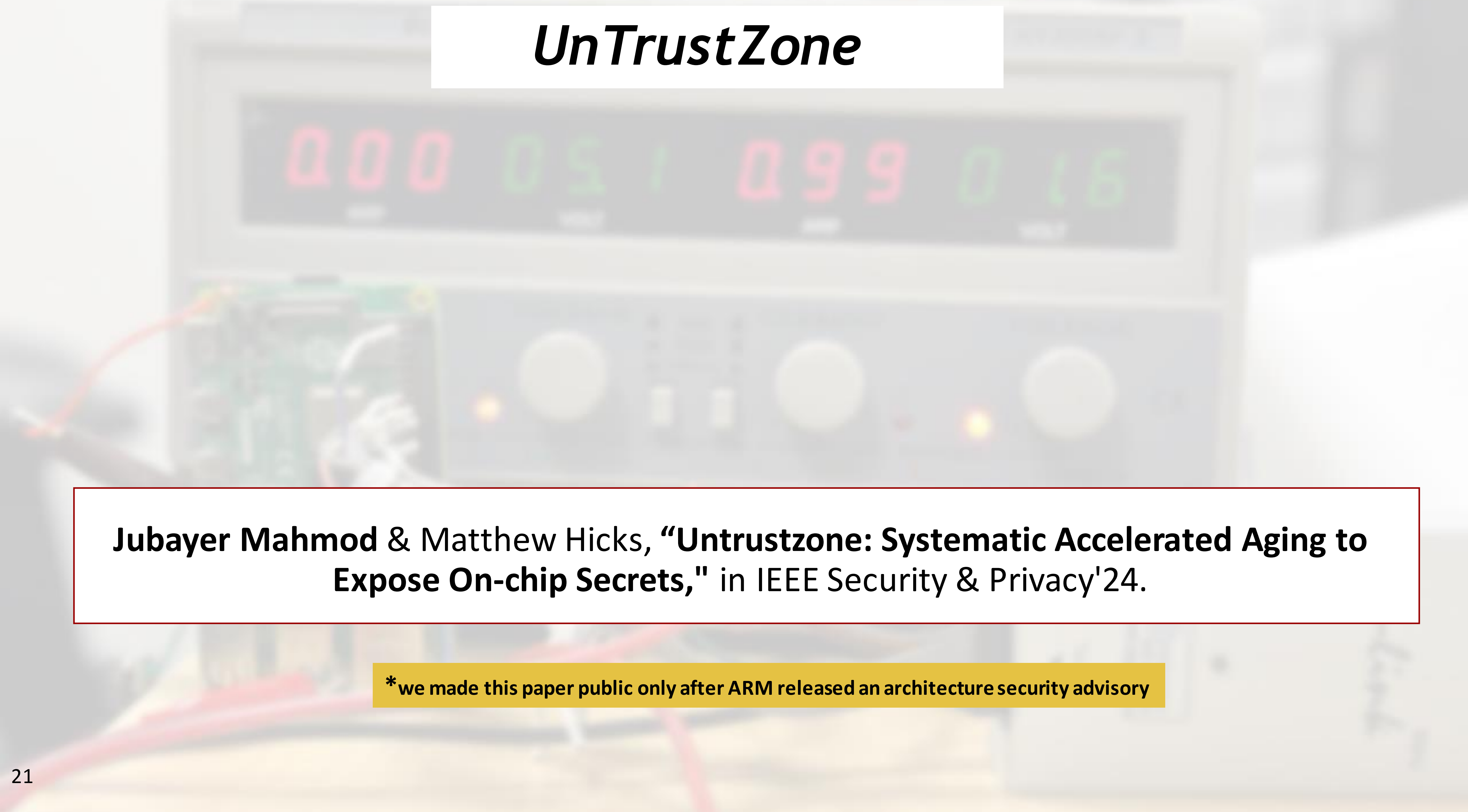
Encrypted hidden massage looks like regular power on state!

Full system implementation



Other evaluation performed: 1) Source of error 2) Recovery 3) multi-snapshot adversary

UnTrustZone



Jubayer Mahmod & Matthew Hicks, **“Untrustzone: Systematic Accelerated Aging to Expose On-chip Secrets,"** in IEEE Security & Privacy'24.

*we made this paper public only after ARM released an architecture security advisory

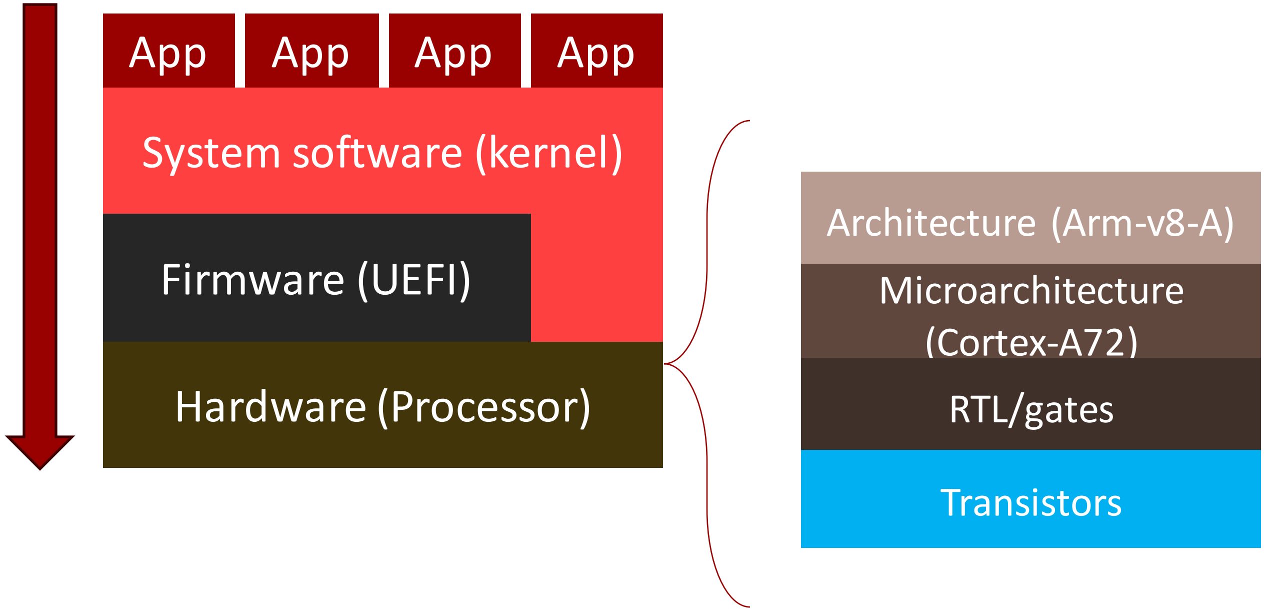
Takeaways InvisibleBits

SRAM power bus is accessible from outside of the SoC

Stored data directs future power-on state

Security threats are on the rise: hardware is in the spotlight

As software security enhances, attackers shift their focus to exploiting lower-level system components.



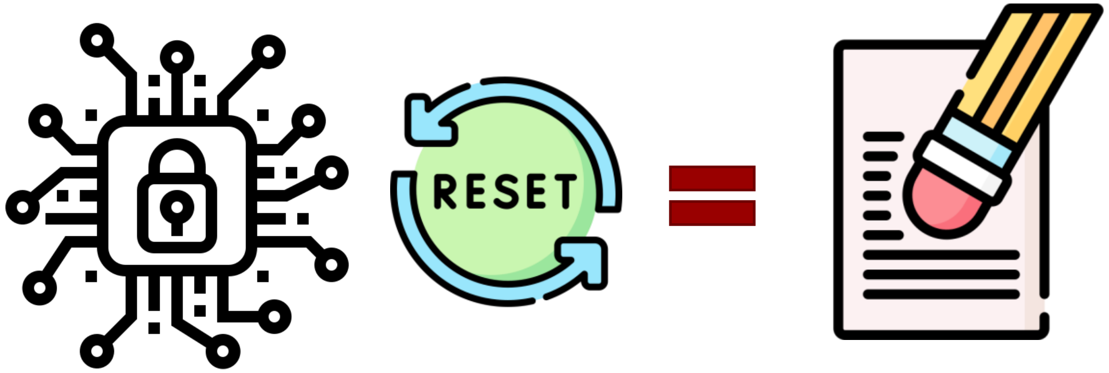
> “Ultimately, **hardware** is the **foundation for digital trust**. A **compromised physical** component can **undermine all additional layers** of a system’s cybersecurity to devastating effect. Hardware security, therefore, focuses on protecting systems against the vulnerabilities at the physical layer of devices”

Security perimeter reduction helps preventing many physical attacks

Keeping sensitive plaintext in on-chip SRAM reduces risks off-chip physical attacks (e.g. cold boot)

Enforcing secure execution prevents illegitimate access to secure memory area

Security attribute change = Memory erasure

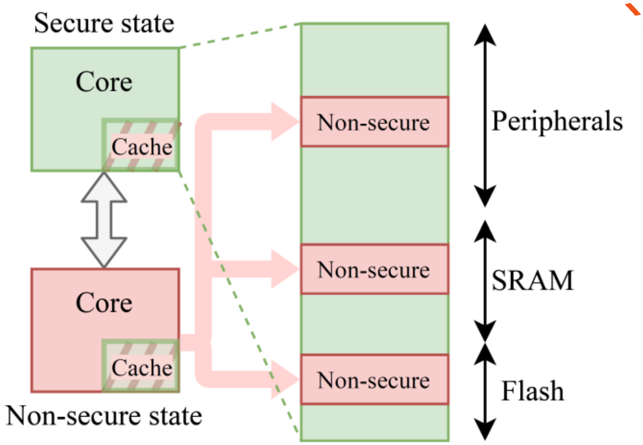


TrustZone fundamentals

Divides a system into the **Secure World & Normal World**

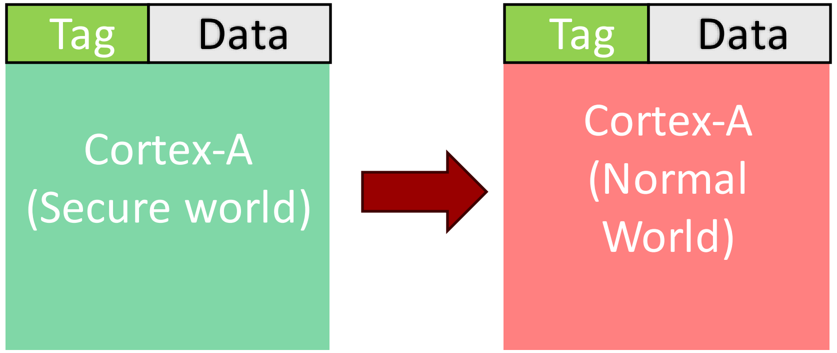
Non-secure state **cannot** access secure memory area

Software **bug** in non-secure state cannot access secure memory



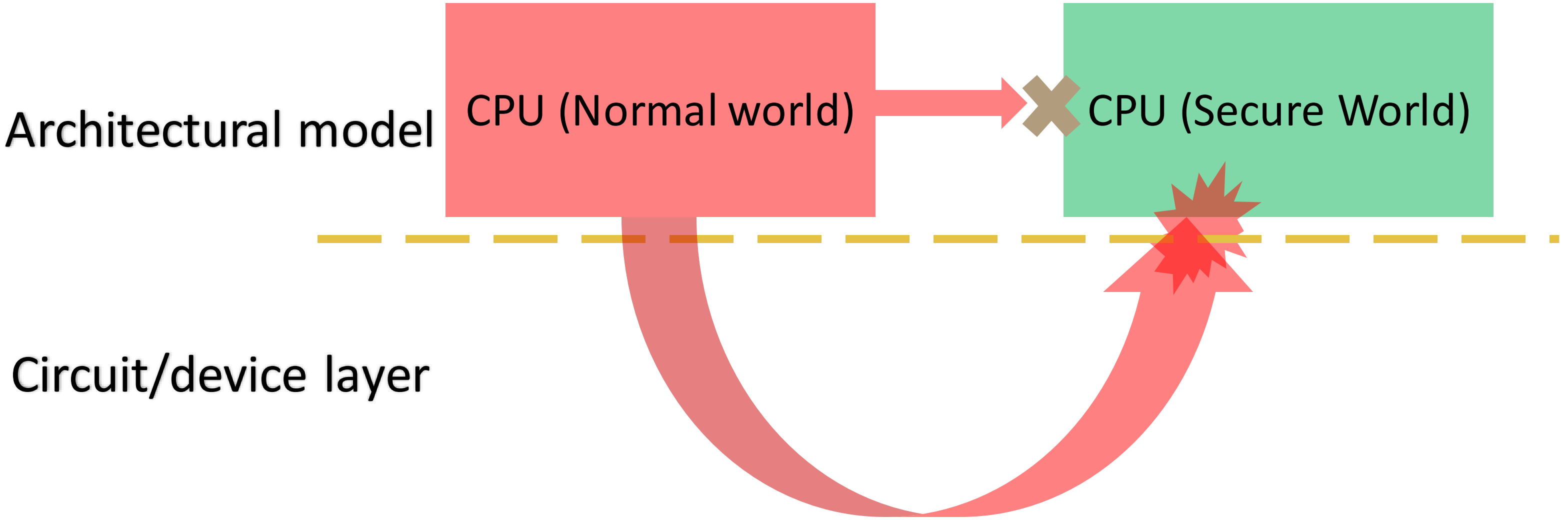
Cache lines are **physically shared** between the Worlds

NS tag bit indicates security levels of a cache line



TrustZone controls security attributes, but physical memory is shared between the *Worlds*.

Going beyond architecture



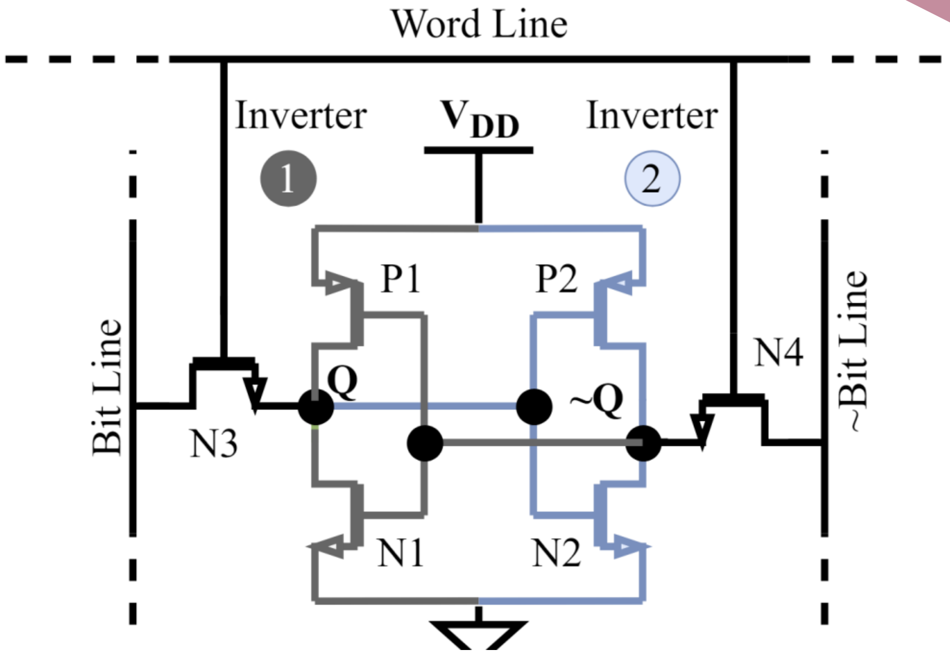
Architectural model

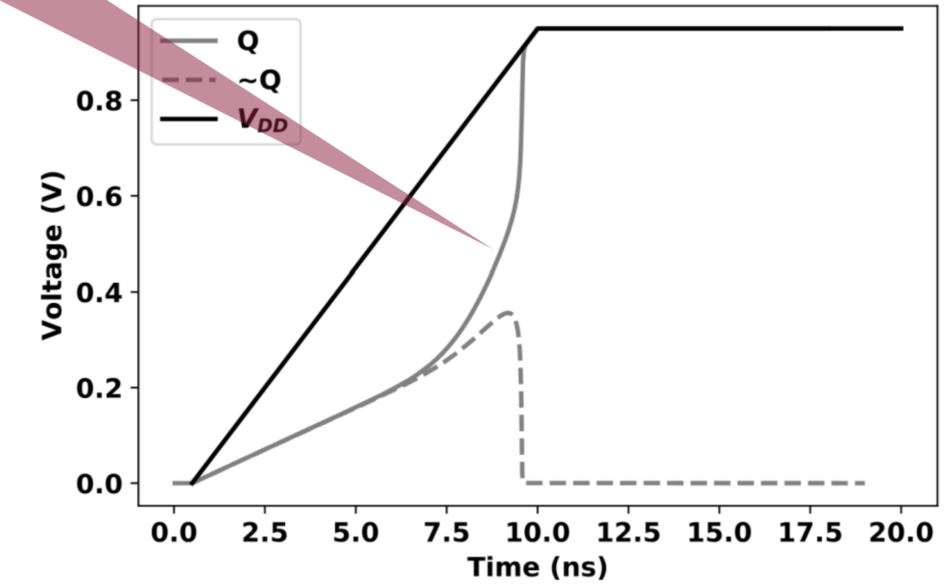
CPU (Normal world)

CPU (Secure World)

Circuit/device layer

Aging allows ‘analog programming’ of the startup race condition.





Overarching threat model

Secrets on-chip SRAM guarded by TrustZone

Attackers have physical access

Target-information- and SoC-specific threat models

Exfiltrate cryptographic key

Exfiltrate proprietary firmware

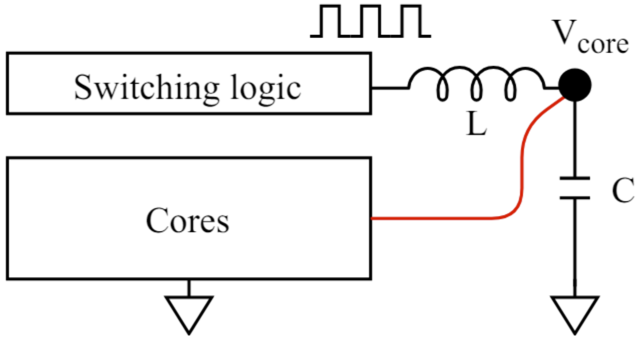
Exfiltrate secrets from cache

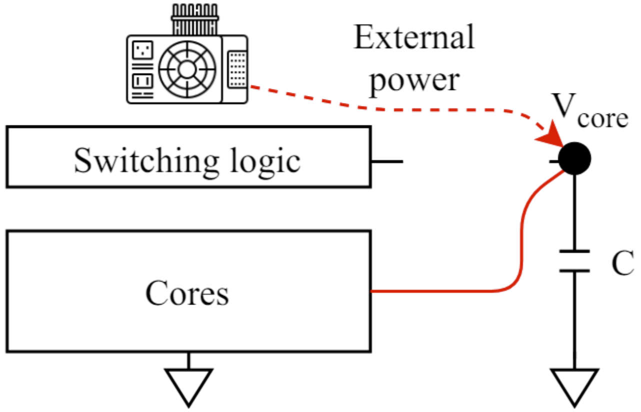
Technical challenges

Overdrive SRAM’s power bus

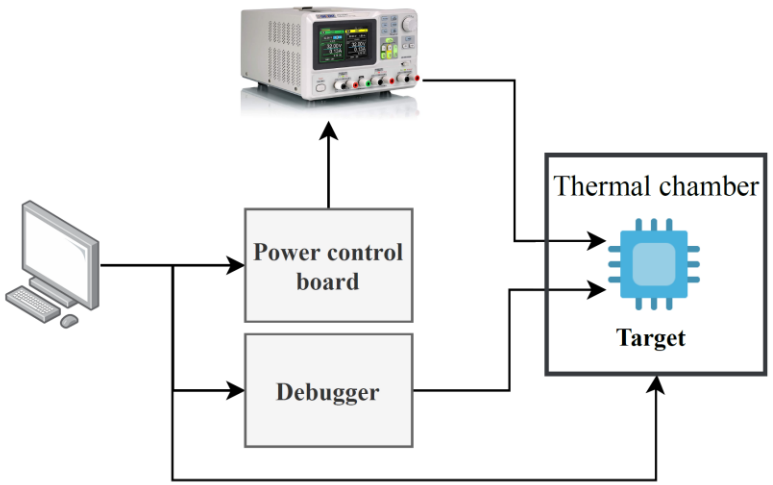
Capture SRAM’s power-on state using software interface

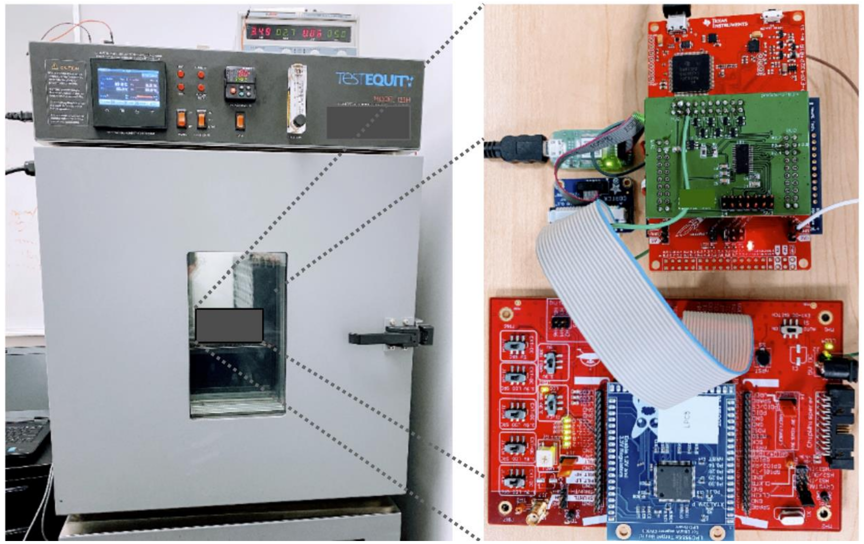
Reduce contamination of SRAM power-on state





Test Platforms and SoCs



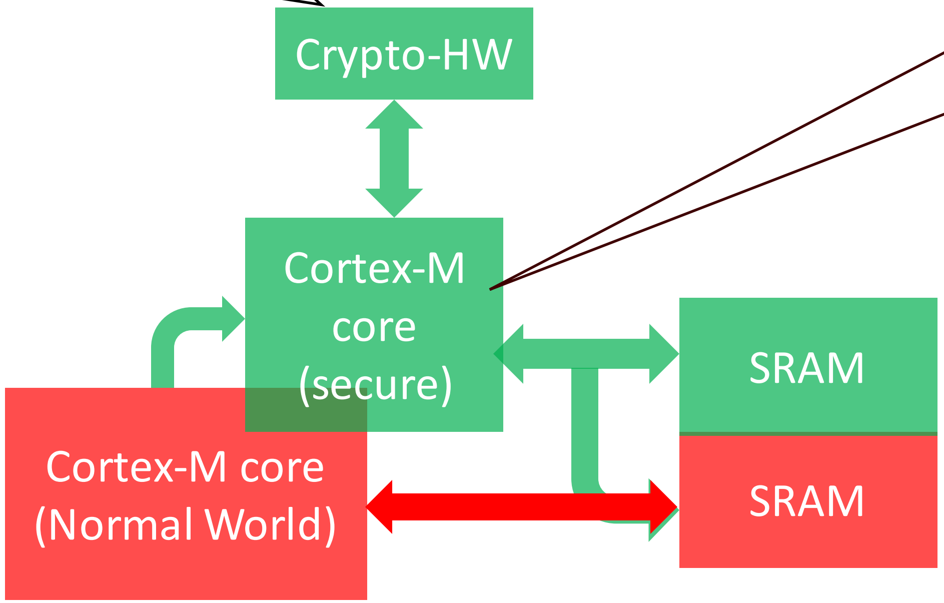


System-on-Chip	Core	SRAM size	TrustZone	Access to uncontaminated power-on state	Aging acceleration	Manufacturer
ATSAML11E16A [59]	ARM Cortex-M23	16KB	✓	✓	✓	Microchip
LPC55S69JBD100 [62]	Dual-core ARM Cortex-M33	320KB	✓	✓	✓	NXP
M263KIAAE [21]	ARM Cortex-M23	96KB	✓	✓	✓	Nuvoton
M2351SFSIAAP [19]	ARM Cortex-M23	96KB	✓	✓	✓	Nuvoton
M252KG6AE [20]	ARM Cortex-M23	32KB	✓	✓	✓	Nuvoton
M251SD2AE [20]	ARM Cortex-M23	12KB	✓	✓	✓	Nuvoton
STM32L562 [85]	ARM Cortex-M33	40KB	✓	✓	✓	STMicroelectronics
BCM2837 (RPi3) [69]	Quad-core ARM Cortex-A53	L1:128KB, L2:512KB	✓	✓	✓	Broadcom
BCM2711 (RPi4) [70]	Quad-core ARM Cortex-A72	L1:320KB, L2:1MB	✓	✓	✓	Broadcom
R7FS1JA783A01CFM [25]	ARM Cortex-M23	32KB	✗	✓	✓	Renesas Electronics
MSP432P401 [35]	ARM Cortex-M4	64KB	✗	✓	✓	Texas Instruments
MSP430G2553 [36]	MSP430 single cycle	0.5KB	✗	✓	✓	Texas Instruments
EFM32WG990F256 [82]	ARM Cortex-M4	32KB	✗	✓	✓	Silicon Labs

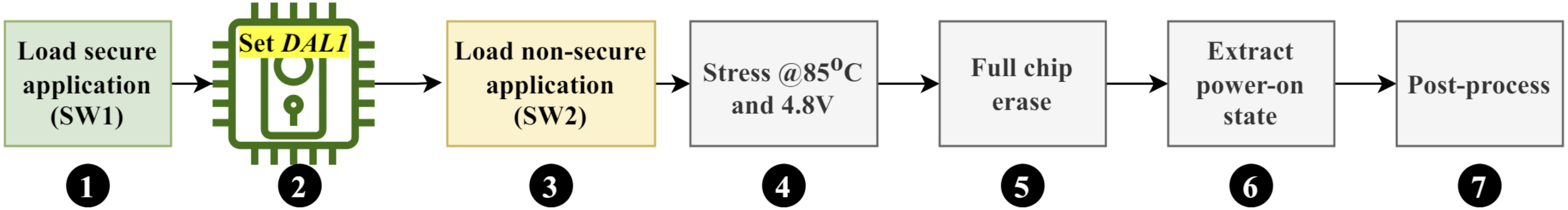
Exfiltrate an AES key from TrustZone

Secure state provides crypto support through non-secure callable (NSC) functions.

NSC functions switch the processor to secure world.



Memory type	Base address	Size (bytes)	Security attributes
Flash	0x00000000	0x7C00	Secure
Flash	0x00007C00	0x0400	NSC
Flash	0x00008000	0x8000	Non-secure
SRAM	0x20000000	0x2000	Secure
SRAM	0x20002000	0x2000	Non-secure



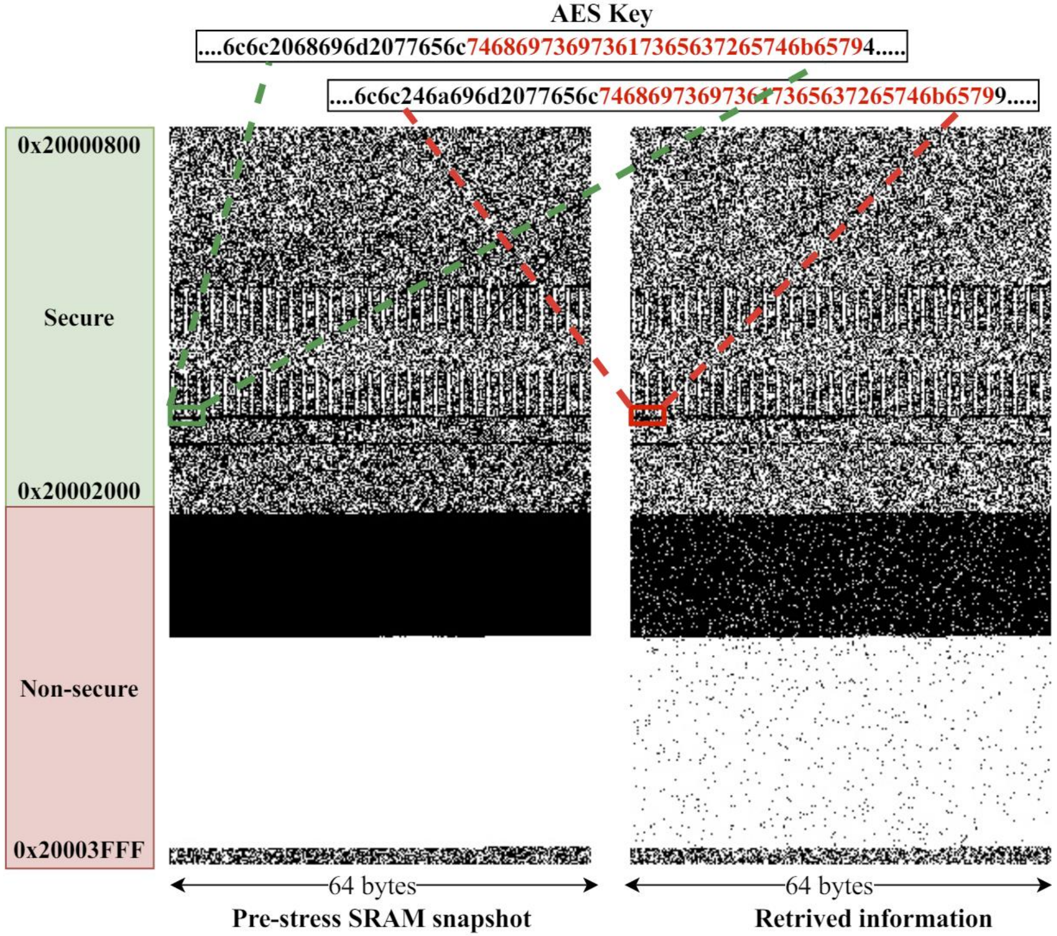
1. Load secure application (SW1)
2. Set *DAL1*
3. Load non-secure application (SW2)
4. Stress @85°C and 4.8V
5. Full chip erase
6. Extract power-on state
7. Post-process

Exfiltrate an AES key from TrustZone

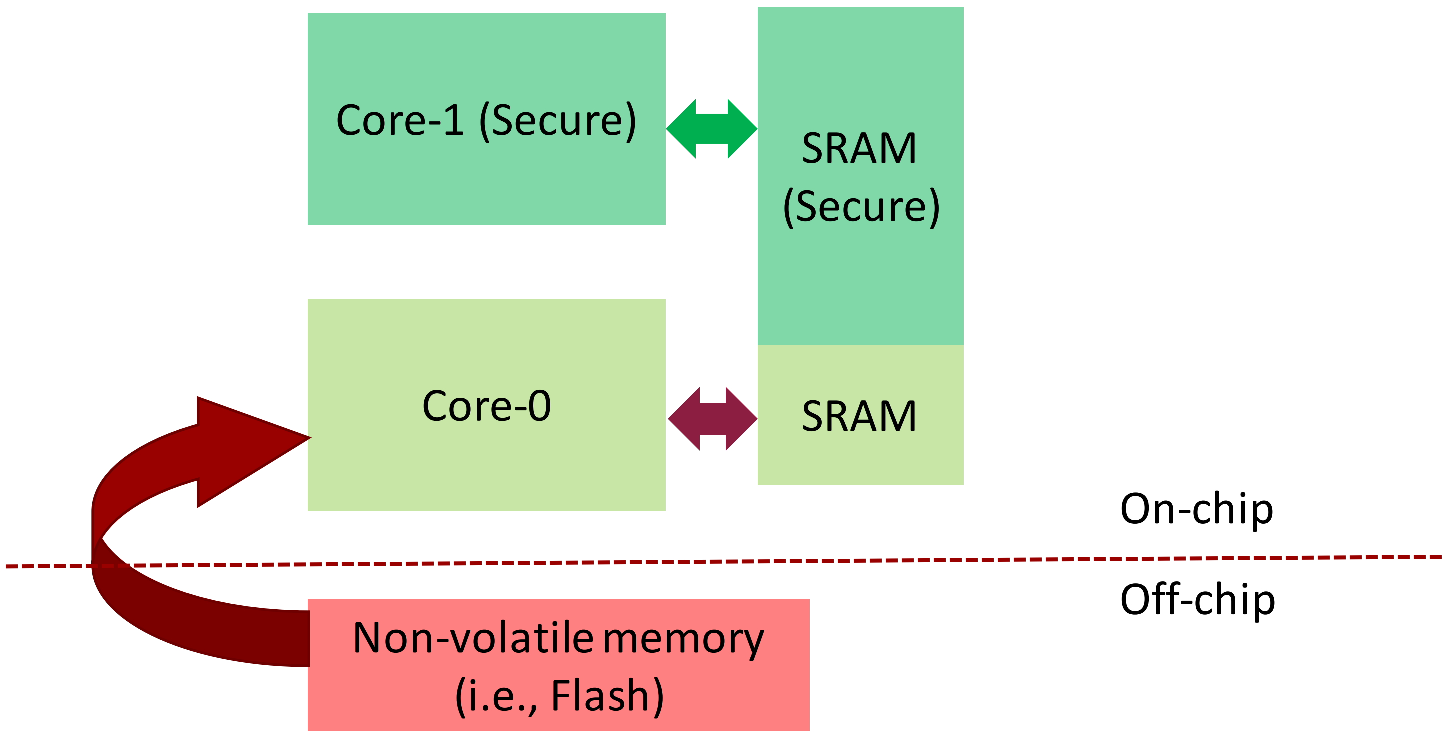
Stored data	Power-on state		Interpreted data	Correctness	% of bits	Transition type
	Pre-stress	Post-stress				
0	0	0	1	✗	2.19%	Flipping failure
0	0	1	0	✓	30.31%	Flipping success
0	1	1	0	✓	23.11%	Reinforcing
1	0	0	1	✓	26.41%	Reinforcing
1	1	0	1	✓	17.38%	Flipping success
1	1	1	0	✗	0.61%	Flipping failure

Key extraction scenario #1
- Error rate: 2.8%
- Key search space$\approx$ 2^{23}

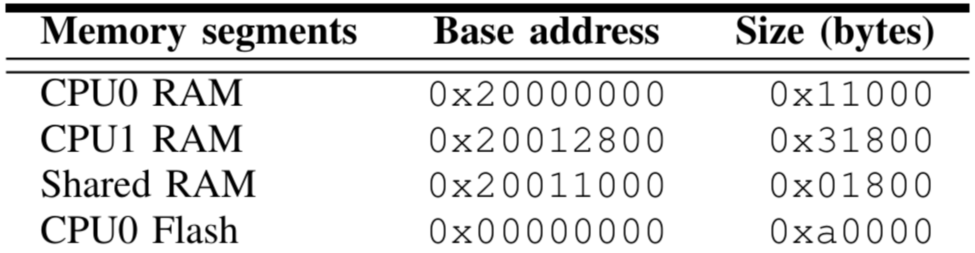
Key extraction scenario #2
- Error rate: 1.27%
- Key search space$\approx$ 2^{13}


AES Key
....6c6c2068696d2077656c74686973697361736563726574 6b65794.....
....6c6c246a696d2077656c746869736973617365637265746b65799.....
0x20000800
Secure
0x20002000
Non-secure
0x20003FFF
64 bytes
64 bytes
Pre-stress SRAM snapshot
Retrived information

Exfiltrate proprietary firmware



Tested on Dual-core Cortex-M33 (LPC55S69)

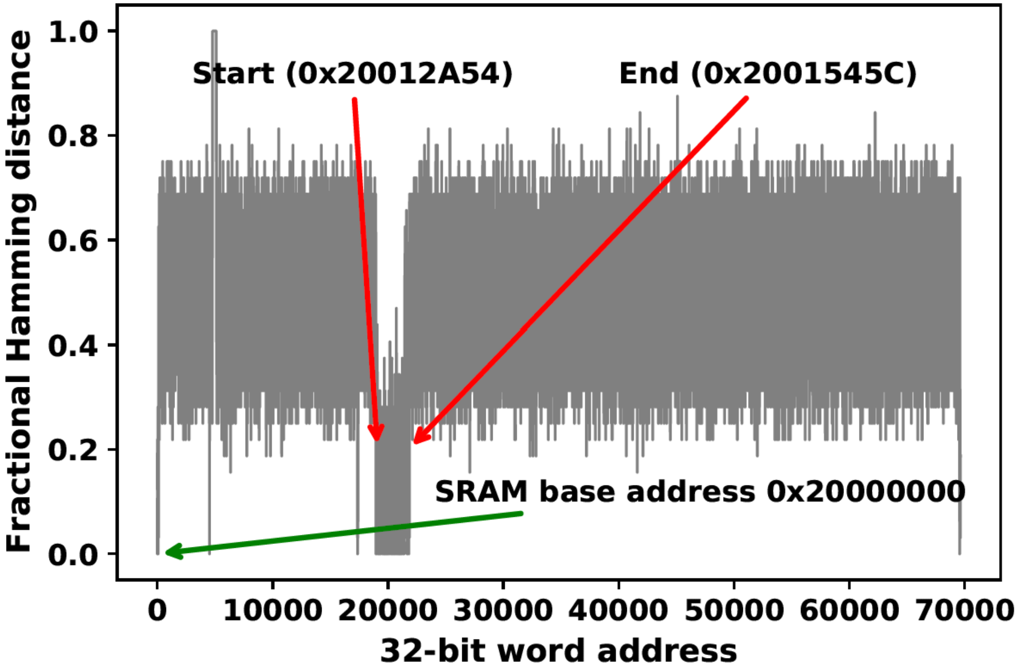


Memory segments	Base address	Size (bytes)
CPU0 RAM	0x20000000	0x11000
CPU1 RAM	0x20012800	0x31800
Shared RAM	0x20011000	0x01800
CPU0 Flash	0x00000000	0xa0000

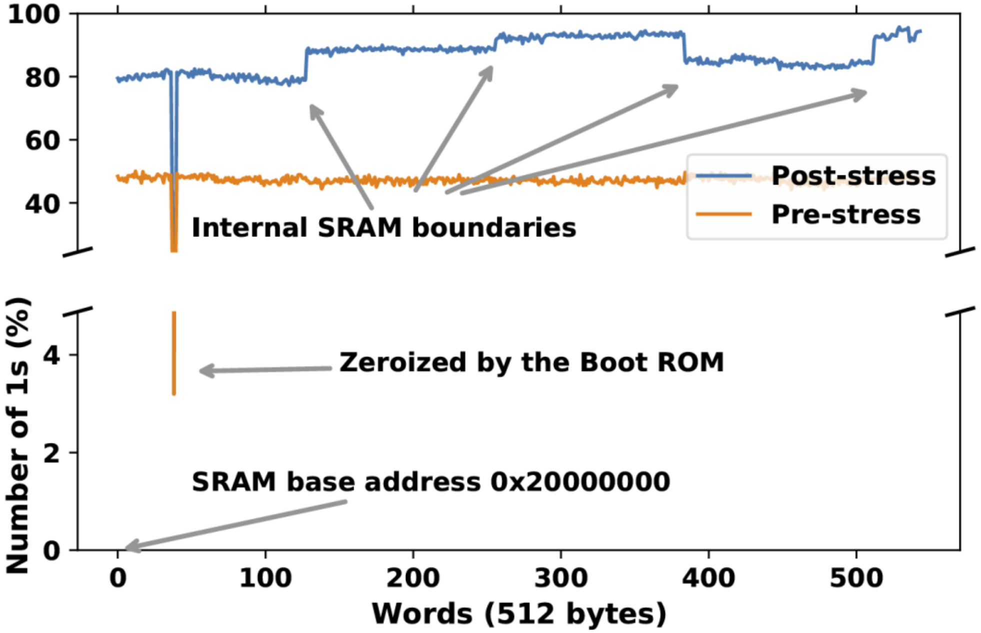
"Case: Cache-assisted Secure Execution on ARM Processors" Oakland'16

Exfiltrate proprietary firmware

	LPC1	LPC2	LPC3	Combined
Scenario # 1 accuracy	87.70%	86.70%	88.50%	95.82%
Scenario # 2 accuracy	93.20%	91.76%	93.36%	98.29%



Visual demonstration of firmware burn-in



Secret placement influences accuracy

Exfiltrate secrets from cache

Victim software executes from CPU

Accelerated aging burns in cache lines in the analog domain

Elevated voltage

Heat

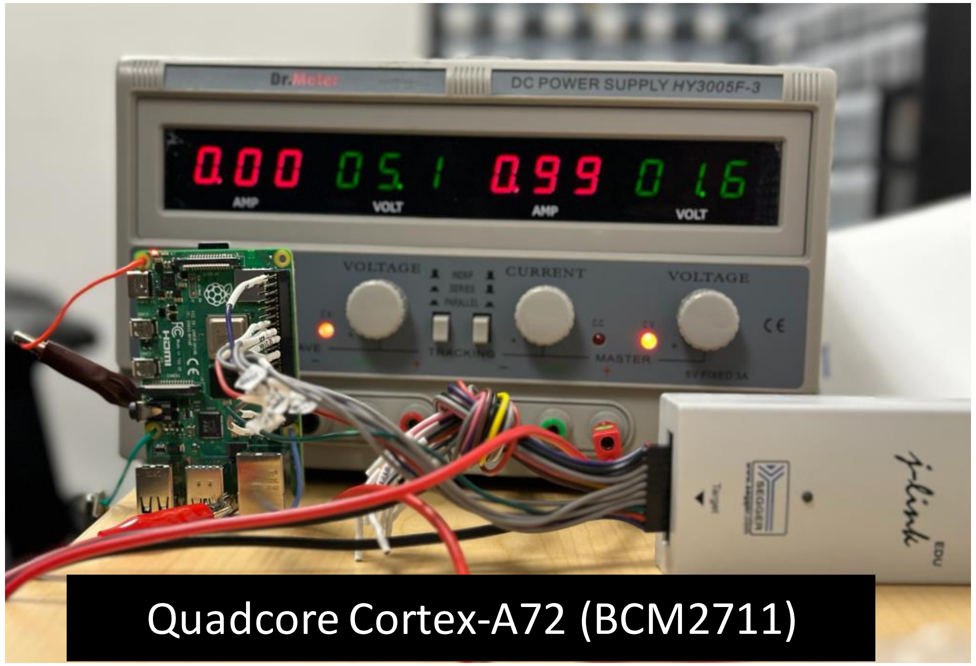
Stress time

Post-stress data extraction

Introduces a ‘fake kernel’

Stops cores from enabling caches (disabled MMU)

Upon request dumps cache lines into the system RAM (using co-processor interface & ram Indexing)



Quadcore Cortex-A72 (BCM2711)

Assumes secret data (attack #1) and proprietary software (attack #2) **are in the on-chip cache** (attack #3)

The AES key extraction accuracy **reaches 93.2% after** 120 hours of aging (2.025× nominal voltage and $T = 85$°C)

Q & A

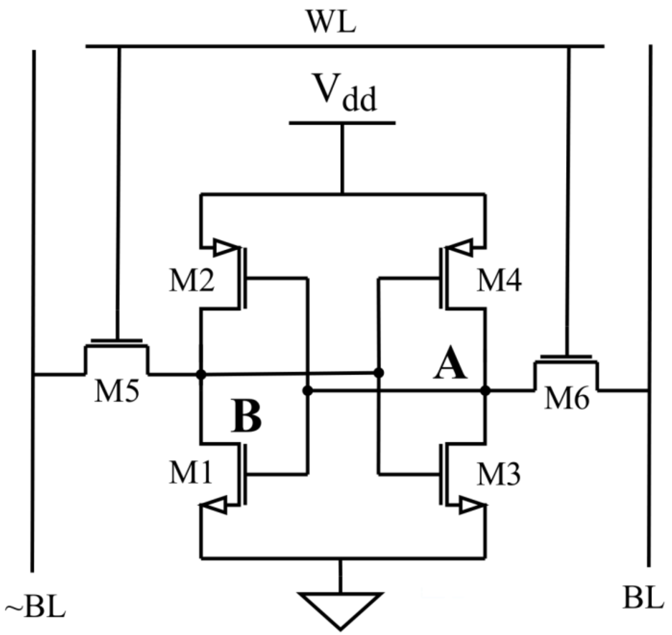
LinkedIn 𝕏

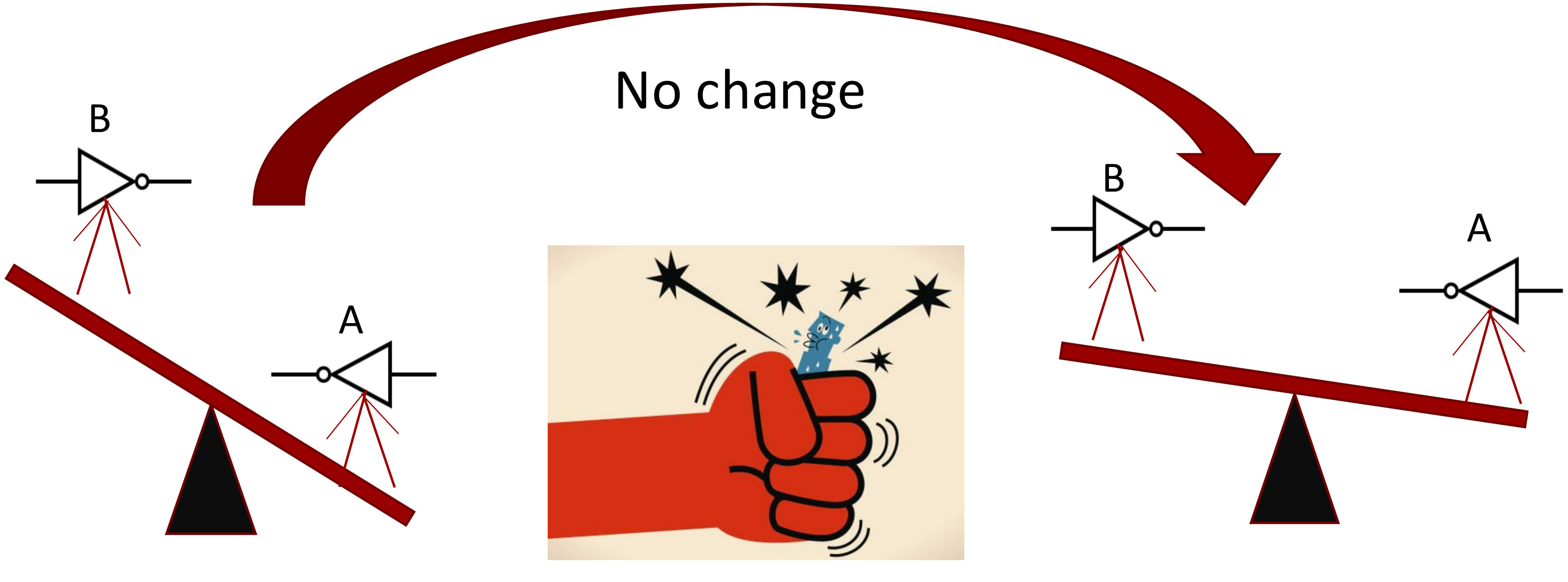
@jubayer0175

Backup slides

Message extraction error: source (1)

Primary source of error: **failing** to change the power-on state.





Power-on state **before** stress

Stress

- Electric
- Thermal
- Long time

Power-on state **after** stress

Performance comparison

Flash **program-time-based** scheme achieves **0.05%** capacity (256KB Flash carries 131B)[Oakland'15]

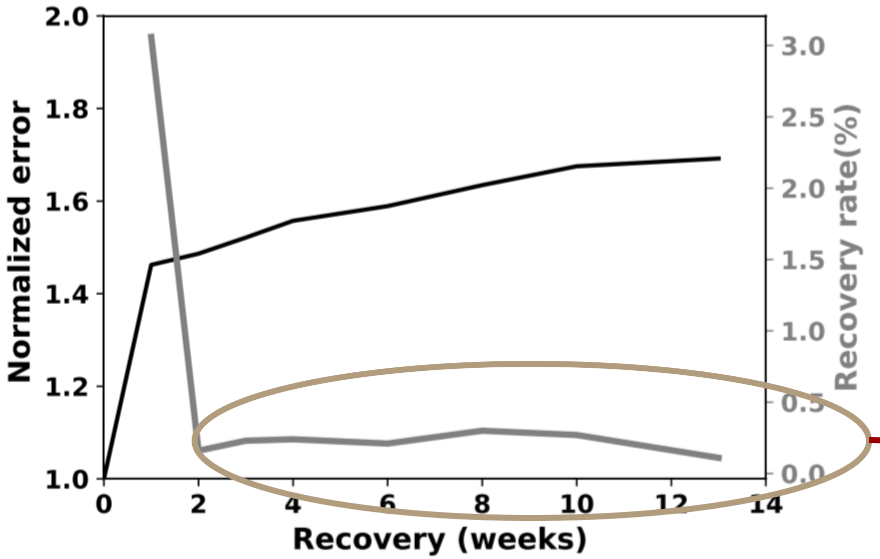
Flash **program-voltage-based** scheme improves capacity by **2x** [Usenix Fast'18]

Invisible bits (with 5 copies @<3% error) carries 12.8KB **(100x)**

	Ubiquity	Capacity	Resilience	Read stable
Flash Program-time-based	◓	●	●	◉
Flash program-voltage-based	◓	●	◓	◉
Invisible bits	◉	◉	◉	◉

◉ = Excellent, ◓ = Very good, ○ = Good, ◓ = Fair, and ● = Poor.

Message extraction error: source (2)



Aging-induced degradation partially recovers.

Error ~10% even after one month

The recovery rate flattens in a few days.

Performance comparison

Flash **program-time-based** scheme achieves **0.05%** capacity (256KB Flash carries 131B)[Oakland'15]

Flash **program-voltage-based** scheme improves capacity by **2x** [Usenix Fast'18]

Invisible bits (with 5 copies @<3% error) carries 12.8KB **(100x)**

	Ubiquity	Capacity	Resilience	Read stable
Flash Program-time-based	⊖	●	●	◉
Flash program-voltage-based	⊖	●	⊖	◉
Invisible bits	◉	◉	◉	◉

◉ = Excellent, ⊖ = Very good, ○ = Good, ◓ = Fair, and ● = Poor.

Takeaways

A new data hiding technique

- Covert: Information stays in the hardware layer
- Erase/write tolerant: Digitally indestructible
- Ubiquitous: Can be implemented in almost any device
- High capacity: 100x compared to state-of-the-art

Qualitative exploration of defensive landscape

Initializing the SRAM at startup

- Needs to wipe out the SRAM at startup
- Slows down boot speed
- Eliminates useful application of SRAM power-on state

Scrambling SRAM data at runtime

- Complement data at runtime to reduce burn in effect (0xAA→ 0x55)
- Core freezing will prevent software mitigation

Preventing aging acceleration

- Prevent over voltage
- Bypassing excess energy before reaching the core