Shaping the Glitch: Optimizing Voltage Fault Injection Attacks

Conference on Cryptographic Hardware and Embedded Systems 2019

Claudio Bozzato[4]

Riccardo Focardi[12]

Francesco Palmarini[13]

[1]Ca' Foscari University of Venice, [2]Cryptosense, [3]Yarix, [4]Talos

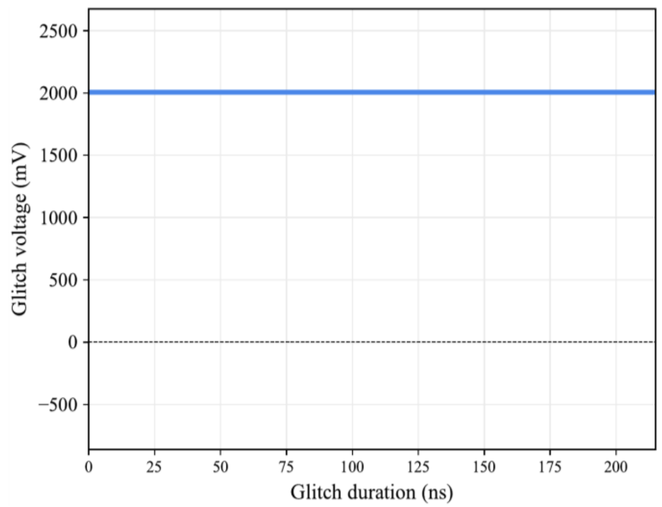
August 28, 2019

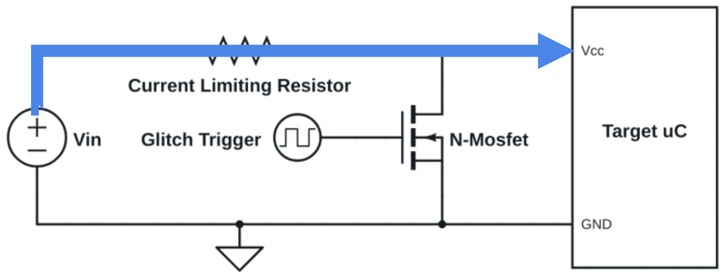
Atlanta, USA

Fault what?

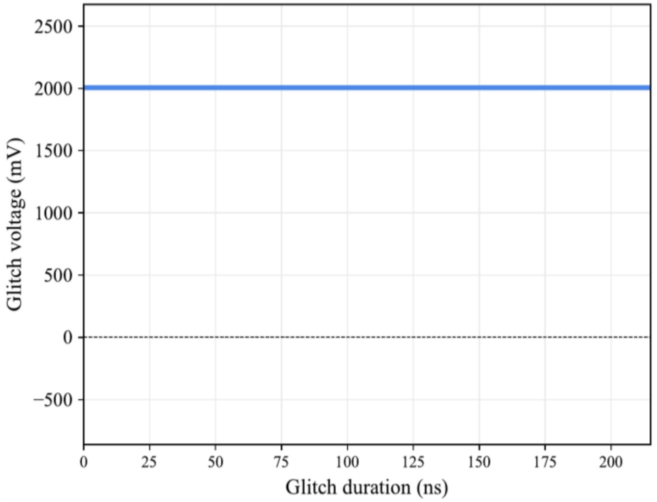
- Exploits **hardware vulnerabilities** to “create” new bugs
- Influence (inject) a system with internal / **external stimuli**
- **Alter** the intended **execution flow** / behavior
- **Skip instructions**, influence branch decisions, corrupt memory locations, etc.
- Bypass security checks, leak data or crypto material, create side-channels, etc.
- **Non-invasive to invasive** techniques: clock, voltage, EM, FIB, laser, heat, flash, etc.

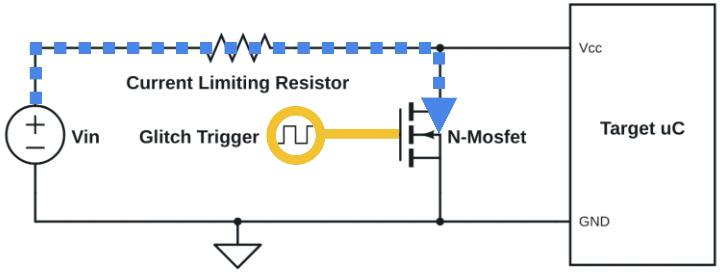
Voltage Fault Injection…
The MOSFET Way





- ✓ The most **widespread** Voltage Fault Injection setup [OC14]
- ✓ Very **easy** to setup and low-cost
- × **Low control** over glitch parameters
- × **Unpredictable:** the glitch characteristics depends on circuit properties, MOSFET, etc.

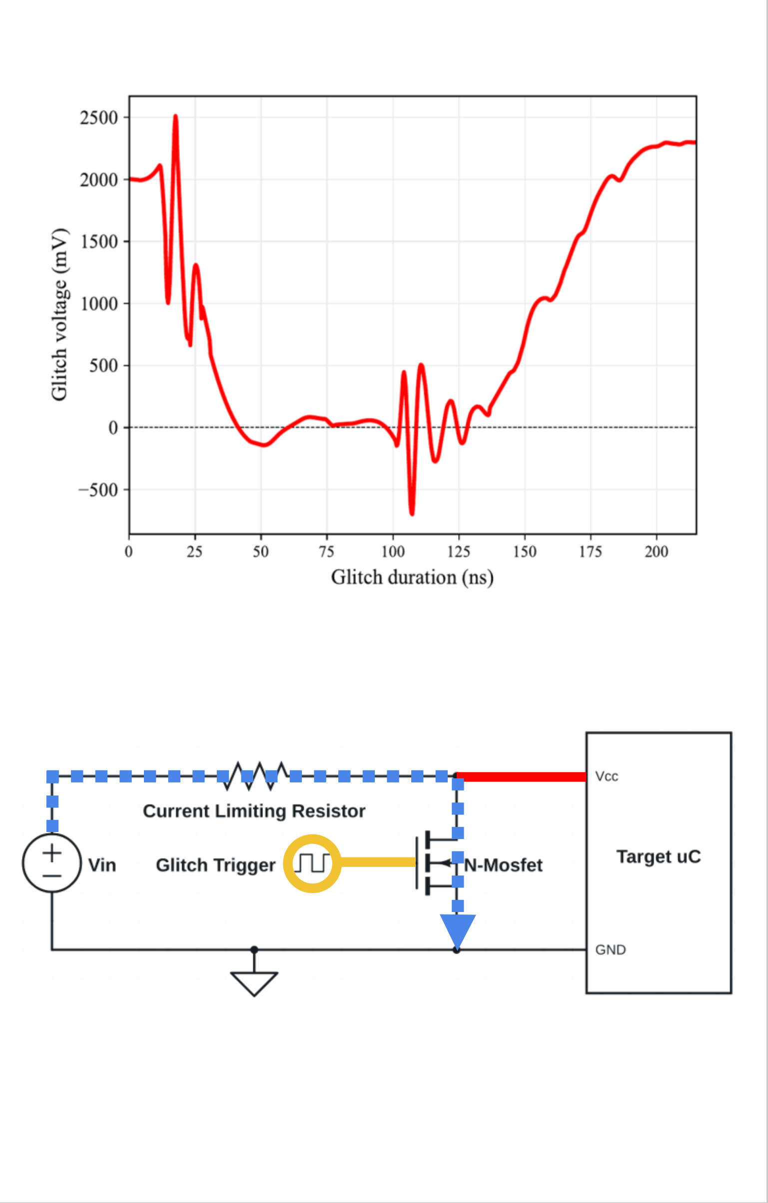




Voltage Fault Injection… The MOSFET Way

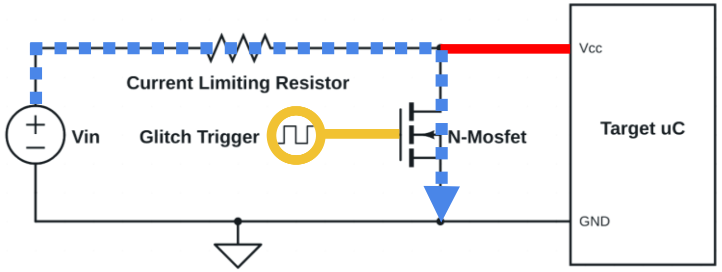
- ✓ The most **widespread** Voltage Fault Injection setup [OC14]
- ✓ Very **easy** to setup and low-cost
- × **Low control** over glitch parameters
- × **Unpredictable:** the glitch characteristics depends on circuit properties, MOSFET, etc.

Voltage Fault Injection… The MOSFET Way



- ✓ The most **widespread** Voltage Fault Injection setup [OC14]
- ✓ Very **easy** to setup and low-cost
- × **Low control** over glitch parameters
- × **Unpredictable:** the glitch characteristics depends on circuit properties, MOSFET, etc.

Voltage Fault Injection… The MOSFET Way



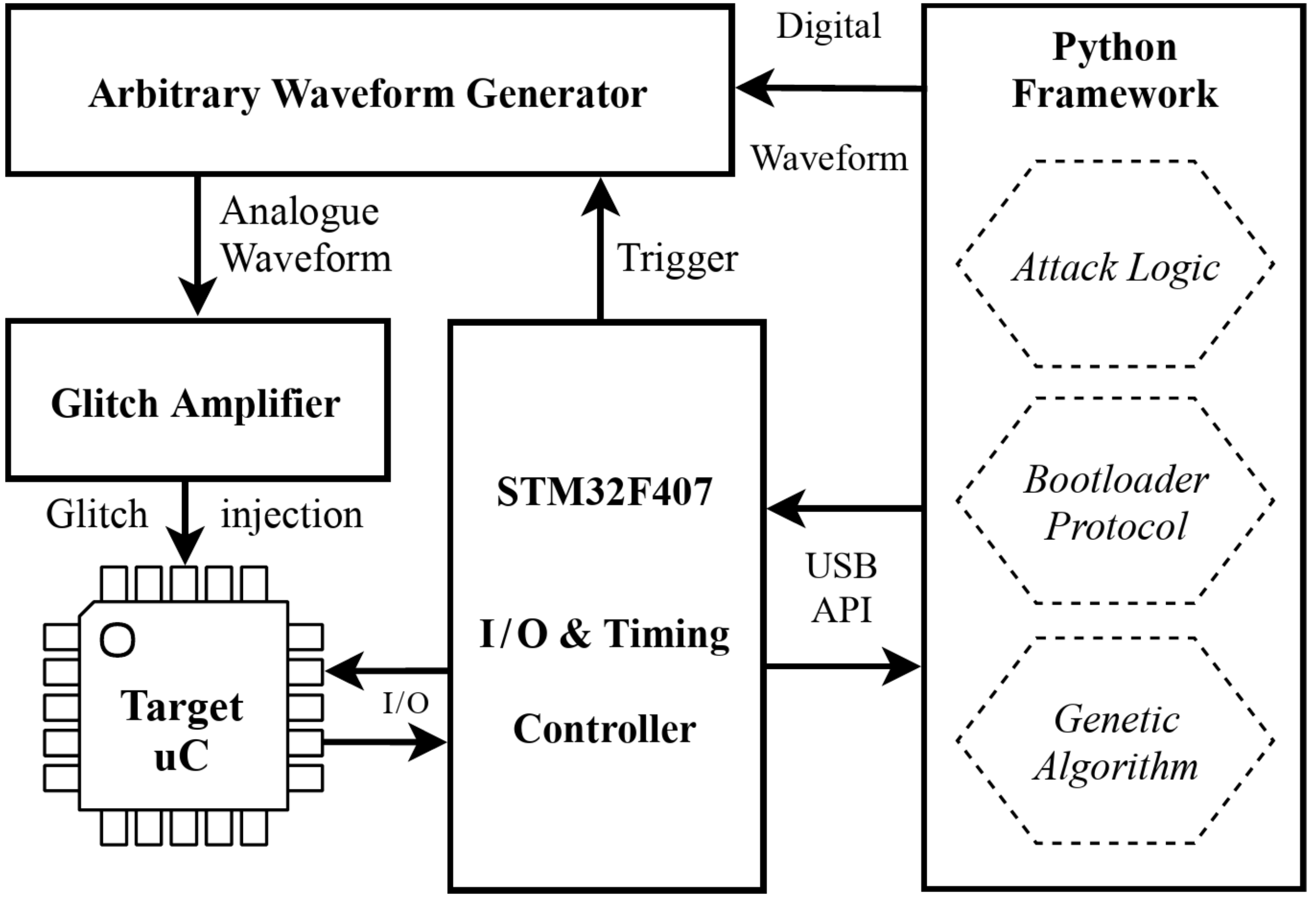
- ✓ The most **widespread** Voltage Fault Injection setup [OC14]
- ✓ Very **easy** to setup and low-cost
- × **Low control** over glitch parameters
- × **Unpredictable:** the glitch characteristics depends on circuit properties, MOSFET, etc.

Our Idea: Arbitrary Glitch Waveforms

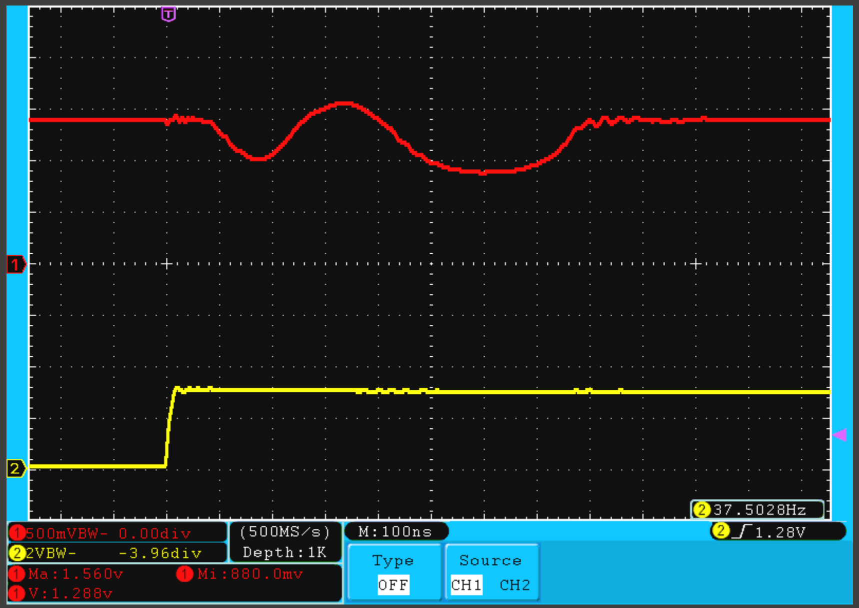
DESIDERATA

- ✓ Stable and **repeatable results**
- ✓ **High degree of freedom** in glitch generation
- ✓ **Software managed** attack parameters
- ✓ **Low-cost** and easy to build setup

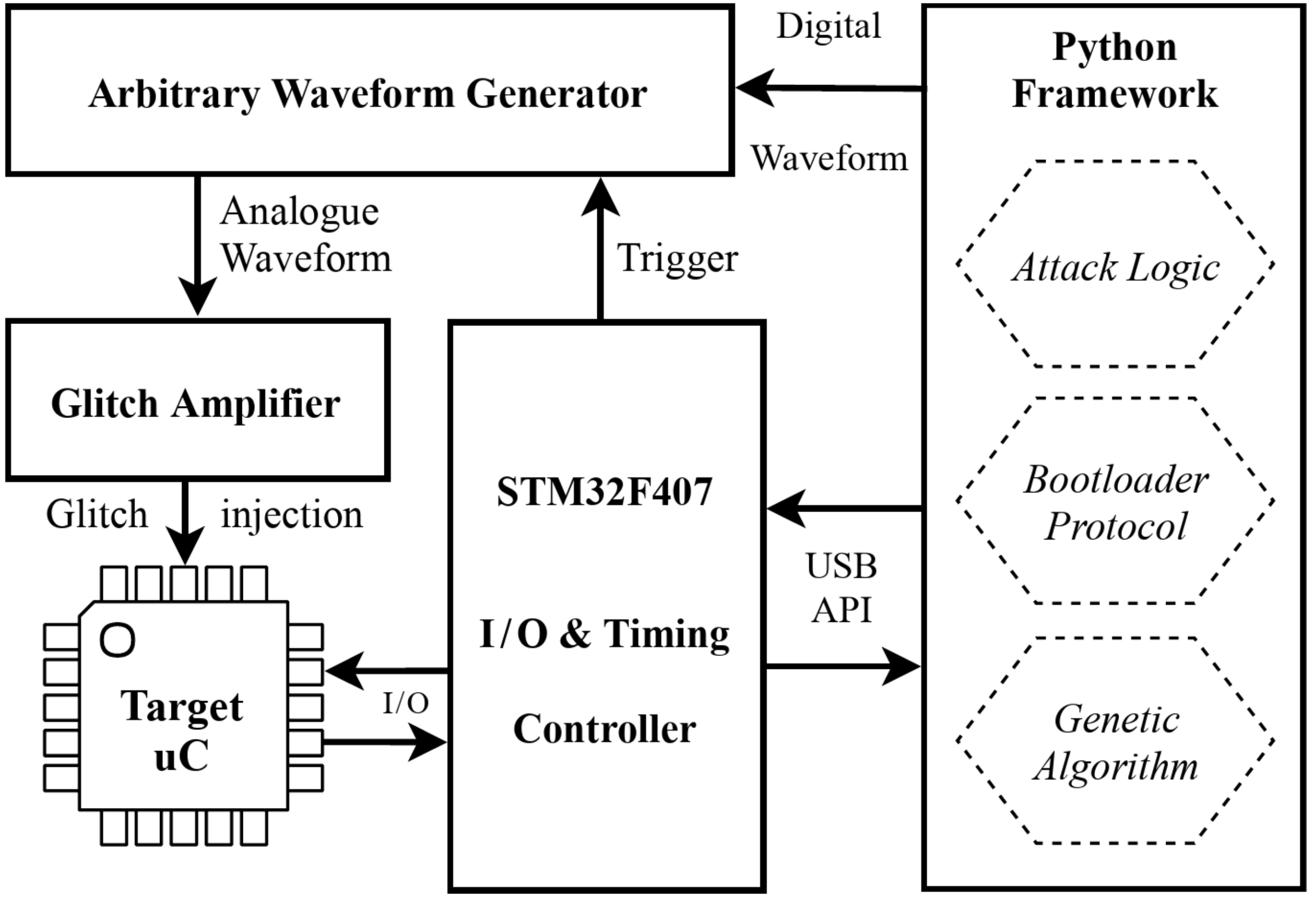
DAC-based glitch generator



Our Idea: Arbitrary Glitch Waveforms



DAC-based glitch generator



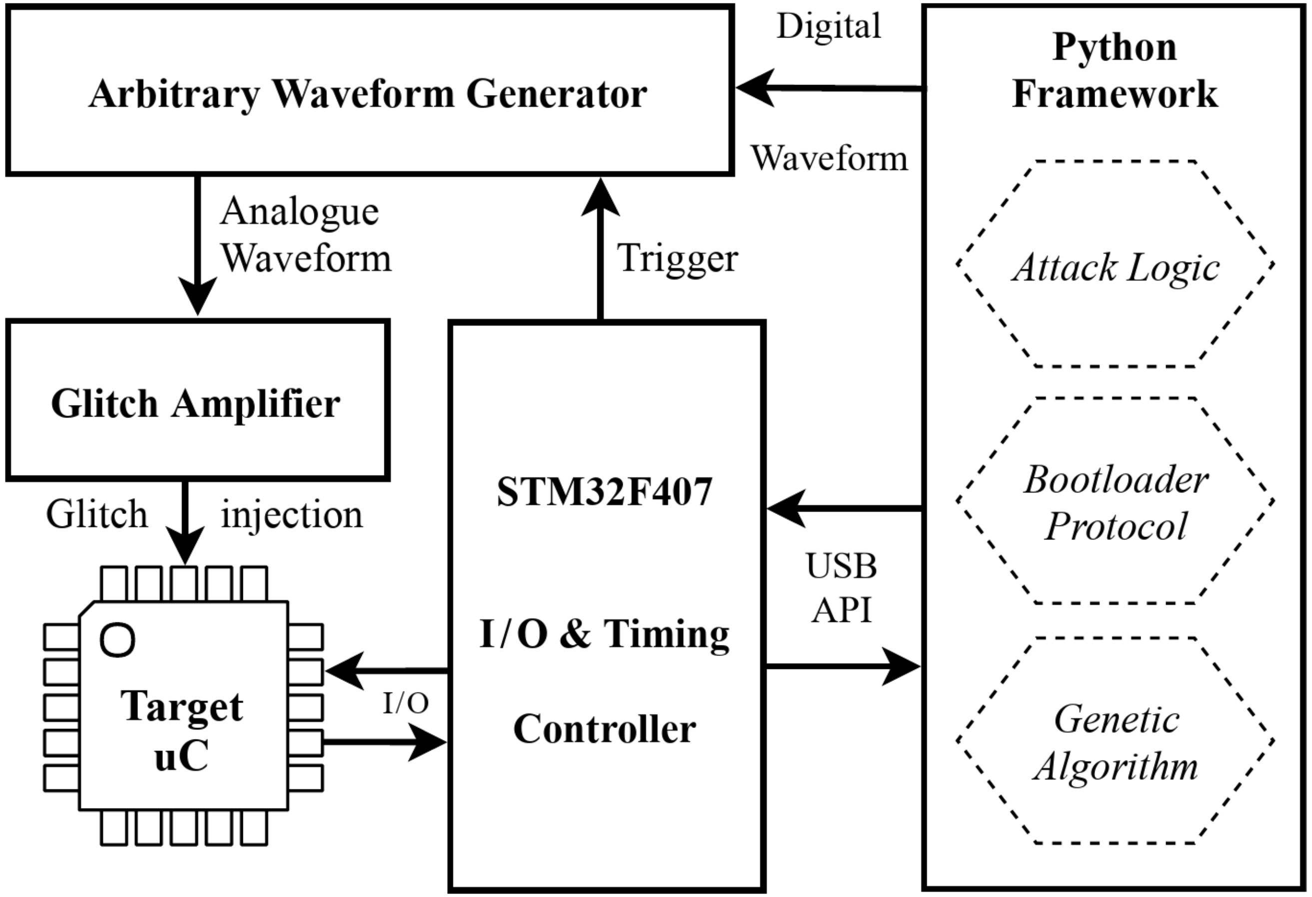
Our Idea: Arbitrary Glitch Waveforms

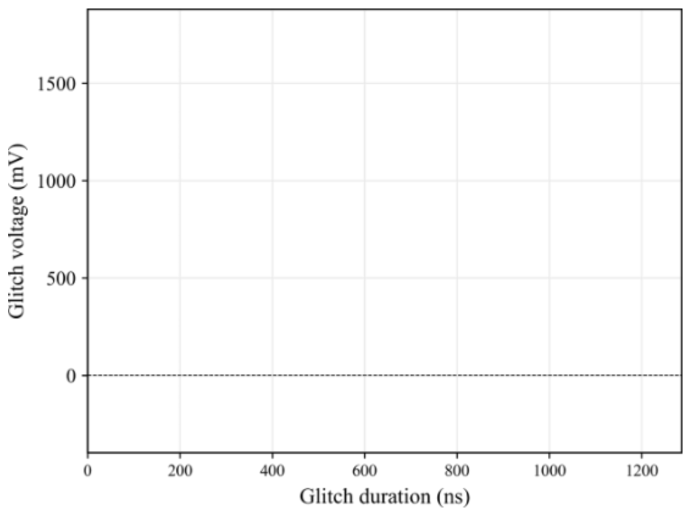
✓ Rising and falling edges affect V-FI performance [ZDCR14]

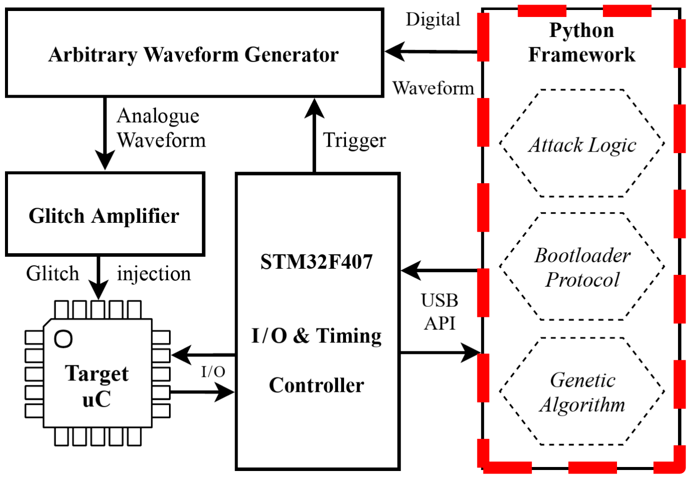
? What if different devices / attacks need **different glitch waveforms**?

? How do we identify the **best match**?

DAC-based glitch generator



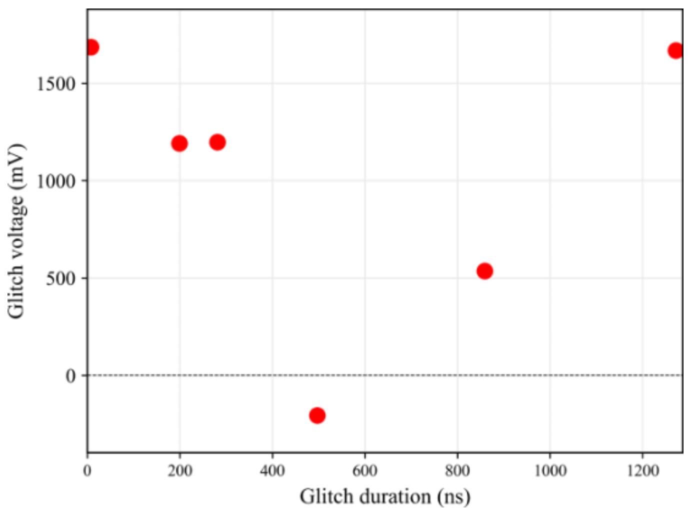


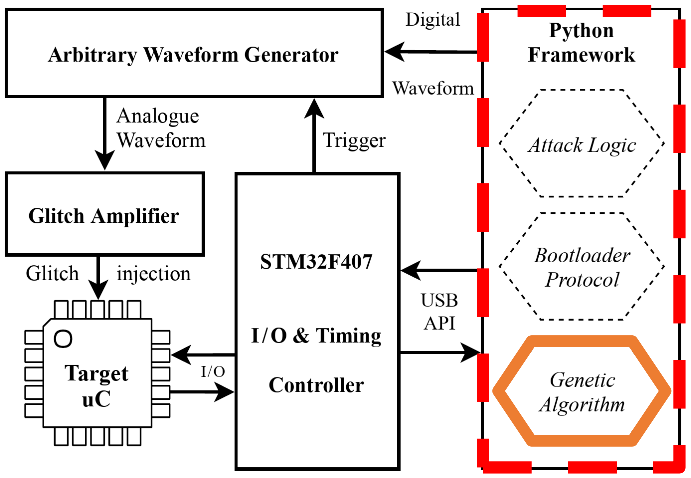


AGW: with big power comes lots of parameters

- Power **supply voltage** with < 10mV resolution
- Glitch **shape** and voltage in 2048 points
- Injection **timing** with ~20ns accuracy
- Glitch frequency / **duration**

→ **Need for automatic parameter search and optimization!**



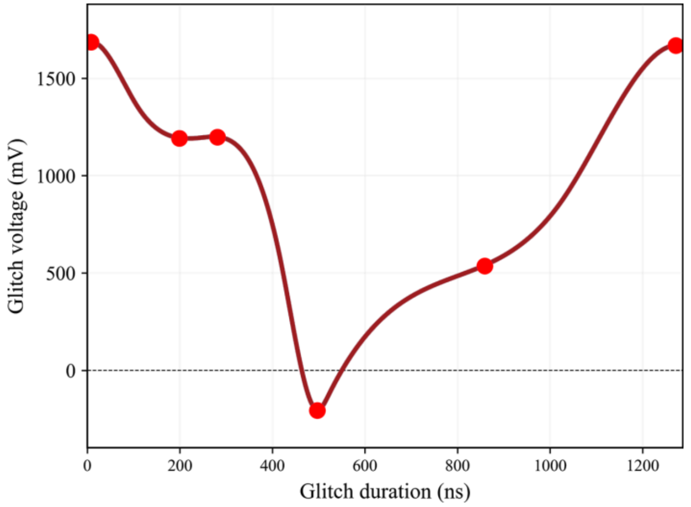


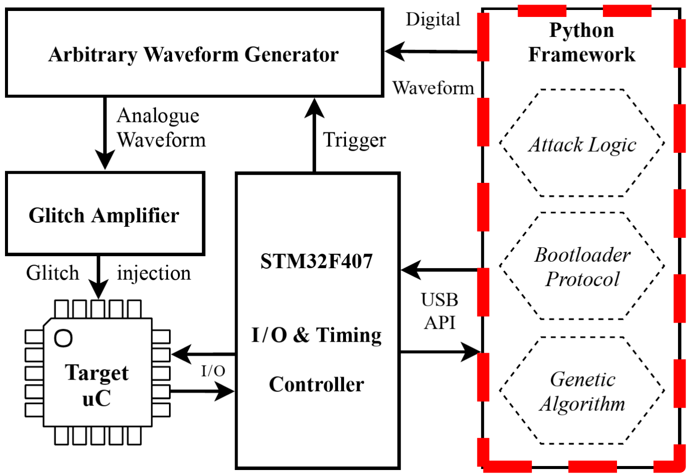
AGW: with big power comes lots of parameters

- Power **supply voltage** with < 10mV resolution
- Glitch **shape** and voltage in 2048 points
- Injection **timing** with ~20ns accuracy
- Glitch frequency / **duration**

→ **Genetic Algoritm (Selection, Crossover, Mutation, Replacement)**

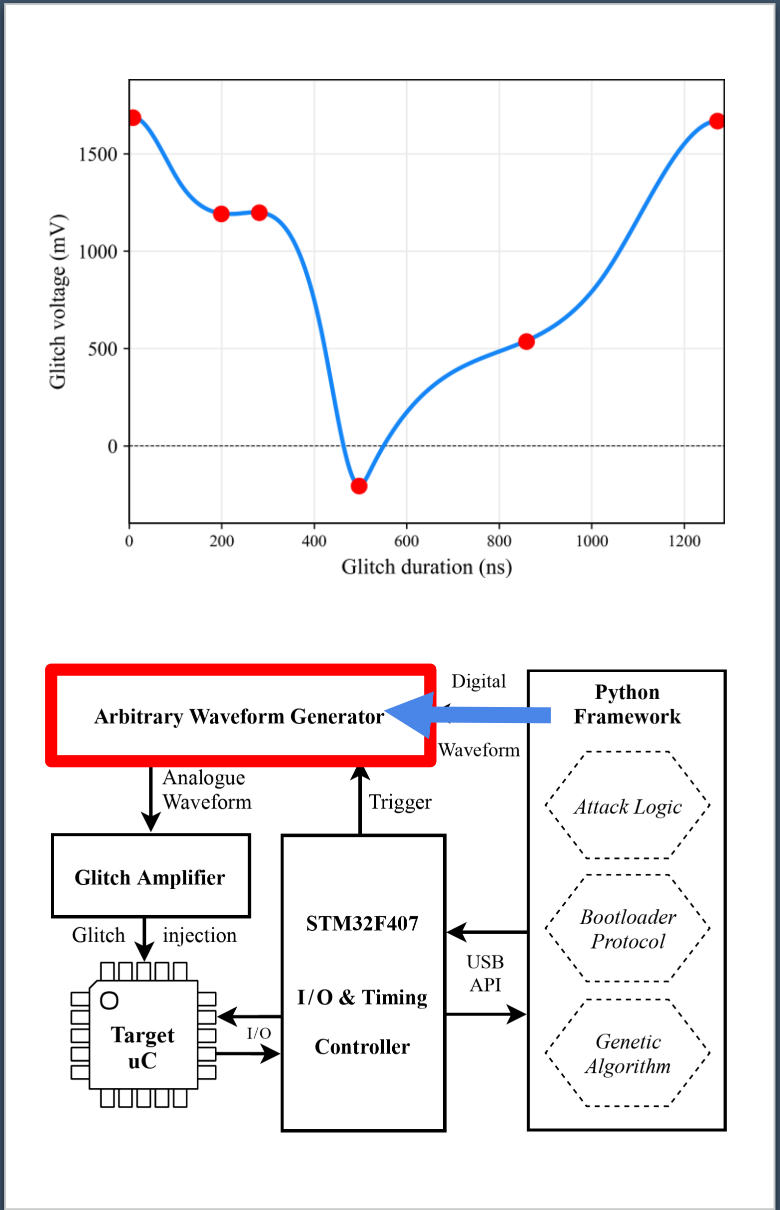
AGW: with big power comes lots of parameters





- Power **supply voltage** with < 10mV resolution
- Glitch **shape** and voltage in 2048 points
- Injection **timing** with ~20ns accuracy
- Glitch frequency / **duration**

→ **Cubic interpolation**

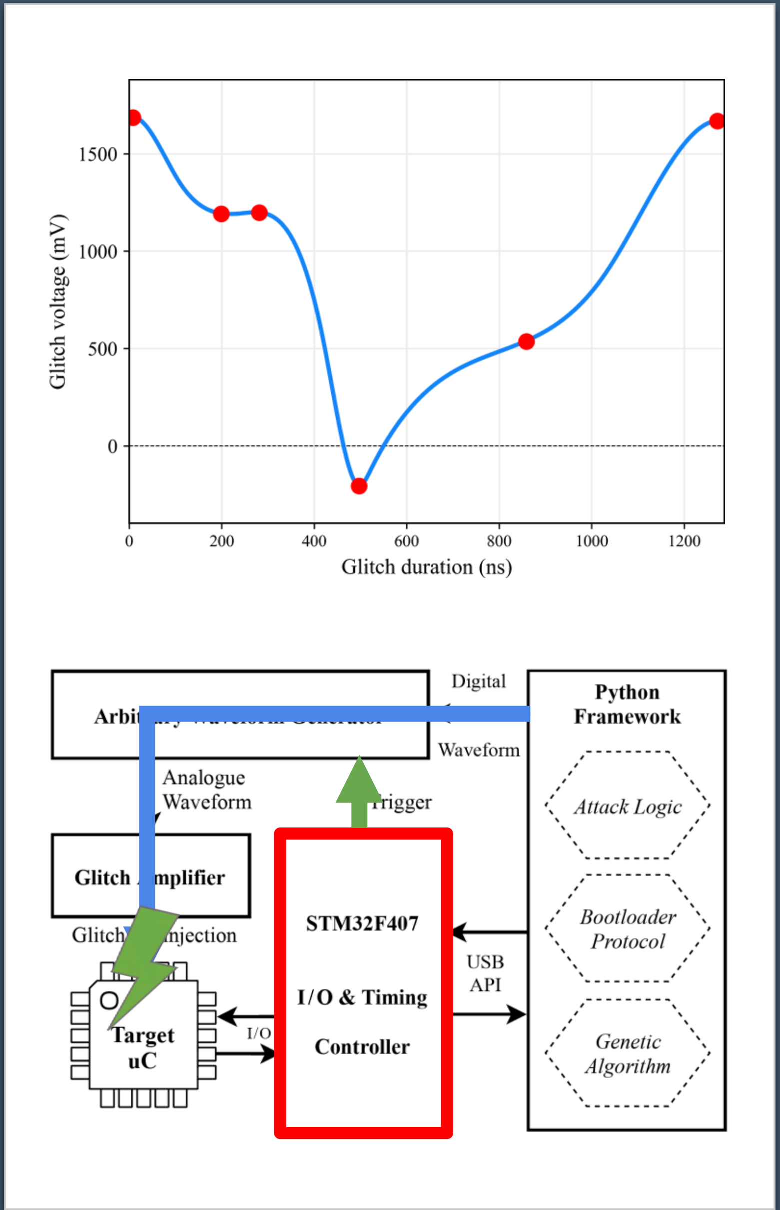


AGW: with big power comes lots of parameters

- Power **supply voltage** with < 10mV resolution
- Glitch **shape** and voltage in 2048 points
- Injection **timing** with ~20ns accuracy
- Glitch frequency / **duration**

→ **Digital-to-Analog conversion**

AGW: with big power comes lots of parameters



- Power **supply voltage** with < 10mV resolution
- Glitch **shape** and voltage in 2048 points
- Injection **timing** with ~20ns accuracy
- Glitch frequency / **duration**

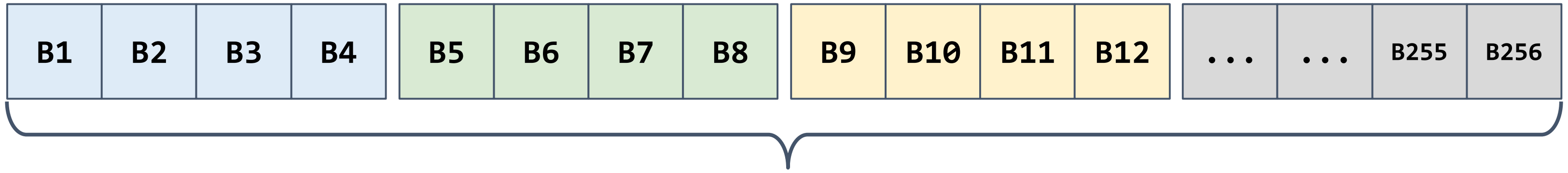
→ **Precise glitch triggering**

Case Study: Renesas 78K Firmware Extraction

- Widely used by the **automotive** industry
- 32 to 256KB **integrated flash memory** for firmware / data
- **Internal bootloader** for flash programming via PC
- No knowledge on the firmware / bootloader code → **Blackbox**
- Bootloader protocol exposes a set of **API** via serial interface
  - *Program*
  - *Erase*
  - *Checksum*
  - *Verify*
- Built-in **security mechanisms**:
  - Commands operate on **256 bytes aligned memory** blocks
  - All programming and erasing **commands can be disabled**
  - **Voltage Supervisor** / BOR

Step I: Finding Vulnerabilities

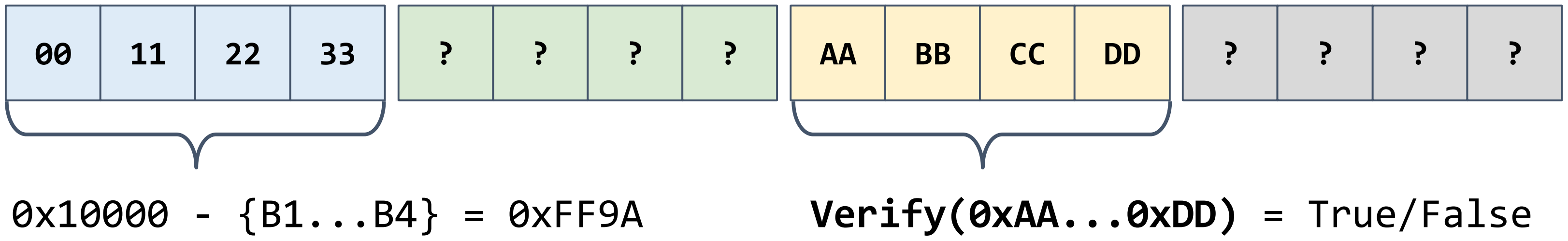
- No *read* command… **Fail** ☹
- Use *FI* to verify just one byte… **Fail** ☹
- Use *FI* to calculate the checksum of one byte… **Fail** ☹
- Use *FI* to calculate the **checksum of 4 bytes** (aligned)...
- Use *FI* to **verify 4 bytes** (aligned)...



Checksum(B1, B256) = 0x10000 - B1 - B2 - B3 - ... - B255 - B256

Step I: Finding Vulnerabilities

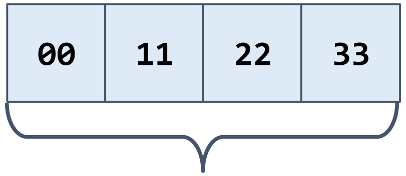
- No *read* command… **Fail** ☹
- Use *FI* to verify just one byte… **Fail** ☹
- Use *FI* to calculate the checksum of one byte… **Fail** ☹
- Use *FI* to calculate the **checksum of 4 bytes** (aligned)... **Success** ☺
- Use *FI* to **verify 4 bytes** (aligned)... **Success** ☺



Step II: Leaking Flash Memory Content

- More leaks required → more faults
- **Side-channel** from the *checksum* computation?

```
def checksum(start, end):
    if (end != start + 256):
        raise
    result = 0x10000
    for i in range(start, end + 1):
        result = result - flash[i]
    return result
```

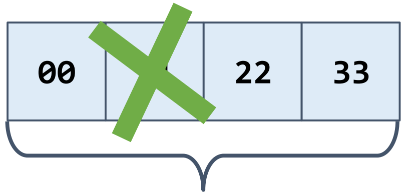


0x10000 - {B1...B4} = **0xFF9A**

Step II: Leaking Flash Memory Content

- More leaks required → more faults
- **Side-channel** from the *checksum* computation?

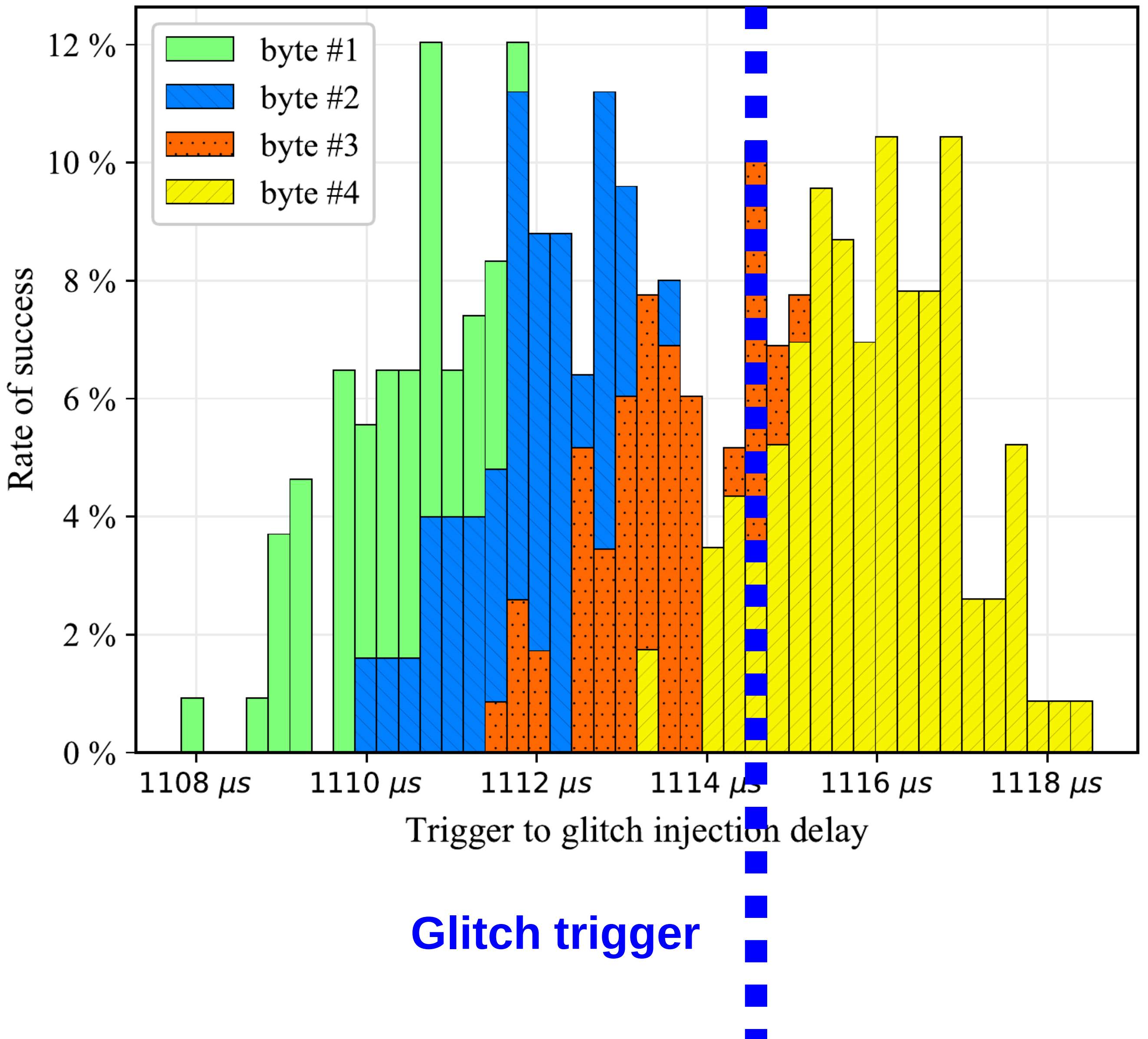
```
def checksum(start, end):
    if (end != start + 256):
        raise
    result = 0x10000
    for i in range(start, end + 1):
        result = result - flash[i]
    return result
```



0x10000 - B1 - B3 - B4 = **0xFFAB**

0xFF9A - 0xFFAB = **0x11**

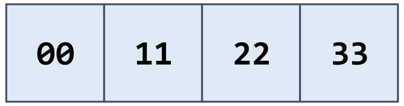
- Just inject a fault for every byte, right? **Nope.**



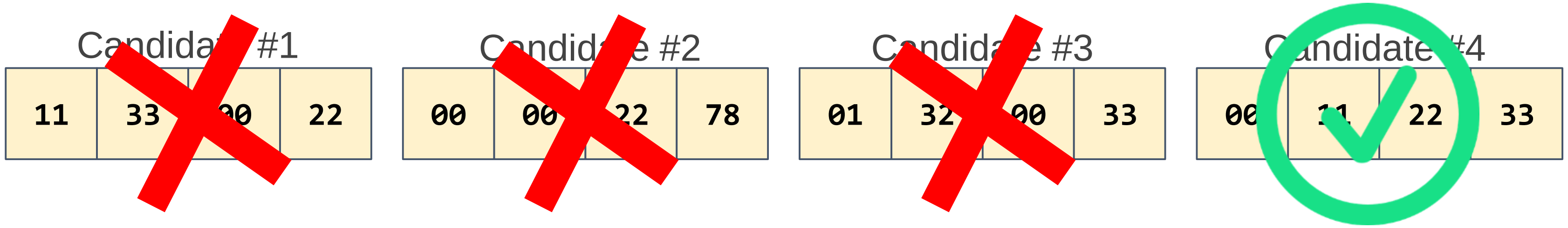
Step III: Deal With Timing Errors

- What is the **extracted value for B3**?
  - **0x22** with ~10% probability
  - **0x33** with ~4% probability
  - **0x11** with ~3% probability
  - **0x00** with <1% probability
  - **0x55** with <1% probability
  - Plus the false positives!

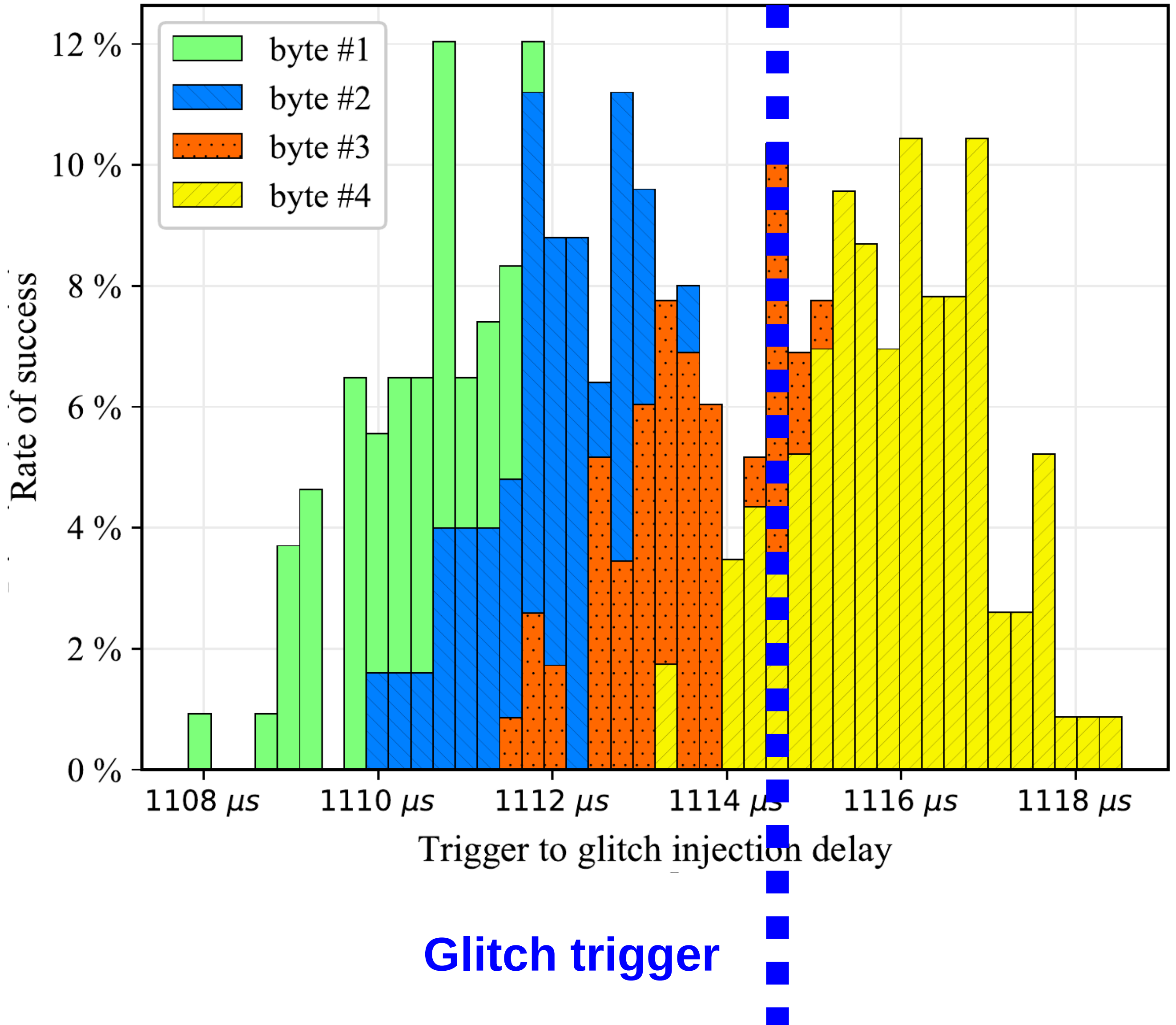
Step IV: Mount the Full Attack



- Calculate the sum of B1+B2+B3+B4 = **0x66**
- For each extracted candidate byte **Bx**:
  - Find all the 4-bytes **permutations with Bx**
  - **Discard** permutations which do not **sum to 0x66**
  - Glitch the *verify* command to **test each new permutation**
  - Stop when the **verify is successful**
- **Iterate** for {B5…B8} {B9…B12} … until the flash is dumped! **MANY hours later…**



- Let the attack go day and night, right? **Not that easy.**



Step V: Compensate for Temperature Errors

- Bootloader runs from **internal oscillator**
- The RC oscillator **drift with temperature**
- The rate is about **0.1% / ◦C**
- With **+6 ◦C** the trigger moved by **> 4 us**
- Solved by **software compensation**

Evaluation & Comparison

Comparison of the Renesas attack performance for three major V-FI techniques.

Technique	Tested combinations	# ShortVerify	# ChecksumLeak	# ShortChecksum	Total glitch count	Total dump time
Mosfet	351 k	13.9 M	3.1 M	699 k	18.1 M	6 d 19 h
Pulse	142 k	3.8 M	2.6 M	582 k	7.1 M	3 d 16 h
AGW	105 k	1.5 M	1.5 M	351 k	3.3 M	2 d 12 h

- **Speed**: our technique is **32% faster** than PULSE and **63% faster** than MOSFET
- **Efficiency**: PULSE used **~2x** the number of glitches and MOSFET **~5x**
- **Reliability:** AGW produces **30% the number of false positives** than MOSFET

Evaluation & Comparison

Comparison of the Renesas attack performance for three major V-FI techniques.

Technique	Tested combinations	# ShortVerify	# ChecksumLeak	# ShortChecksum	Total glitch count	Total dump time
Mosfet	351 k	13.9 M	3.1 M	699 k	18.1 M	6 d 19 h
Pulse	142 k	3.8 M	2.6 M	582 k	7.1 M	3 d 16 h
AGW	105 k	1.5 M	1.5 M	351 k	3.3 M	2 d 12 h

- **Speed**: our technique is **32% faster** than PULSE and **63% faster** than MOSFET
- **Efficiency**: PULSE used **~2x** the number of glitches and MOSFET **~5x**
- **Reliability:** AGW produces **30% the number of false positives** than MOSFET

Evaluation & Comparison

Comparison of the Renesas attack performance for three major V-FI techniques.

Technique	Tested combinations	# ShortVerify	# ChecksumLeak	# ShortChecksum	Total glitch count	Total dump time
Mosfet	351 k	13.9 M	3.1 M	699 k	18.1 M	6 d 19 h
Pulse	142 k	3.8 M	2.6 M	582 k	7.1 M	3 d 16 h
AGW	105 k	1.5 M	1.5 M	351 k	3.3 M	2 d 12 h

- **Speed**: our technique is **32% faster** than PULSE and **63% faster** than MOSFET
- **Efficiency**: PULSE used **~2x** the number of glitches and MOSFET **~5x**
- **Reliability:** AGW produces **30% the number of false positives** than MOSFET

Evaluation & Comparison

Comparison of the Renesas attack performance for three major V-FI techniques.

Just 60KB!

Technique	Tested combinations	# ShortVerify	# ChecksumLeak	# ShortChecksum	Total glitch count	Total dump time
Mosfet	351 k	13.9 M	3.1 M	699 k	18.1 M	6 d 19 h
Pulse	142 k	3.8 M	2.6 M	582 k	7.1 M	3 d 16 h
AGW	105 k	1.5 M	1.5 M	351 k	3.3 M	2 d 12 h

- **Speed**: our technique is **32% faster** than PULSE and **63% faster** than MOSFET
- **Efficiency**: PULSE used **~2x** the number of glitches and MOSFET **~5x**
- **Reliability:** AGW produces **30% the number of false positives** than MOSFET

Evaluation & Comparison

Comparison of the Renesas attack performance for three major V-FI techniques.

Technique	Tested combinations	# ShortVerify	# ChecksumLeak	# ShortChecksum	Total glitch count	Total dump time
Mosfet	351 k	13.9 M	3.1 M	699 k	18.1 M	6 d 19 h
Pulse	142 k	3.8 M	2.6 M	582 k	7.1 M	3 d 16 h
AGW	105 k	1.5 M	1.5 M	351 k	3.3 M	2 d 12 h

- **Speed**: our technique is **32% faster** than PULSE and **63% faster** than MOSFET
- **Efficiency**: PULSE used **~2x** the number of glitches and MOSFET **~5x**
- **Reliability:** AGW produces **30% the number of false positives** than MOSFET

Evaluation & Comparison

Comparison of the Renesas attack performance for three major V-FI techniques.

Technique	Tested combinations	# ShortVerify	# ChecksumLeak	# ShortChecksum	Total glitch count	Total dump time
Mosfet	351 k	13.9 M	3.1 M	699 k	18.1 M	6 d 19 h
Pulse	142 k	3.8 M	2.6 M	582 k	7.1 M	3 d 16 h
AGW	105 k	1.5 M	1.5 M	351 k	3.3 M	2 d 12 h

- **Speed**: our technique is **32% faster** than PULSE and **63% faster** than MOSFET
- **Efficiency**: PULSE used **~2x** the number of glitches and MOSFET **~5x**
- **Reliability:** AGW produces **30% the number of false positives** than MOSFET

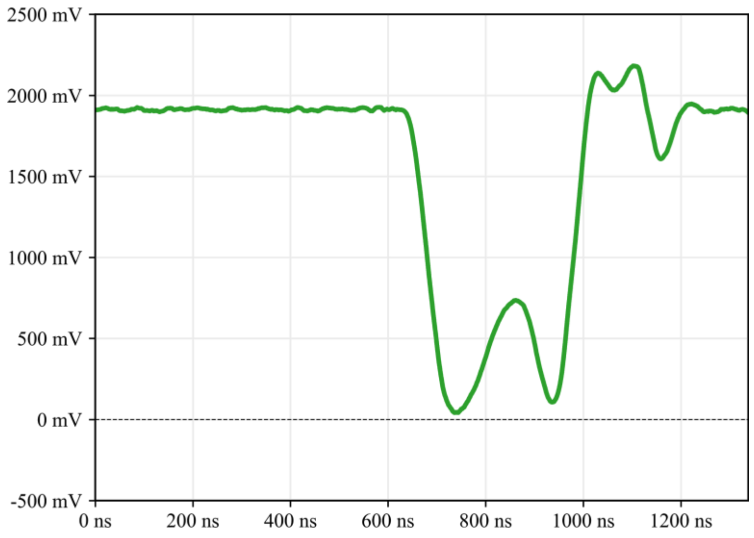
Evaluation & Comparison

Comparison of the Renesas attack performance for three major V-FI techniques.

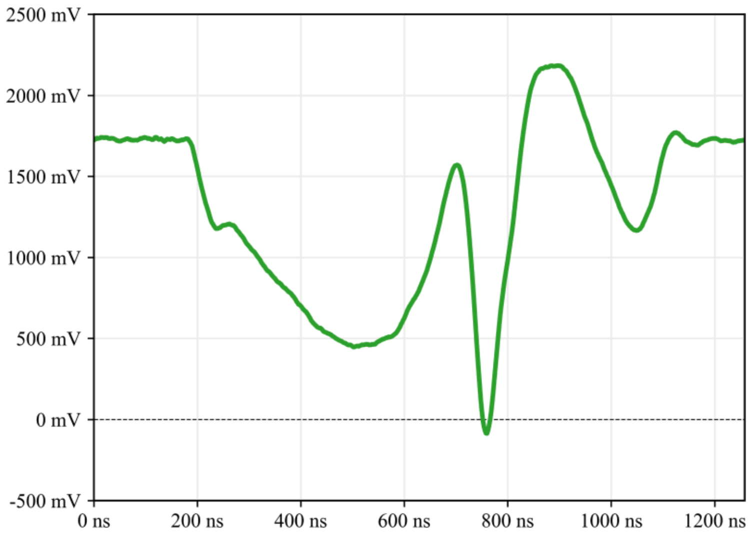
Technique	Tested combinations	# ShortVerify	# ChecksumLeak	# ShortChecksum	Total glitch count	Total dump time
Mosfet	351 k	13.9 M	3.1 M	699 k	18.1 M	6 d 19 h
Pulse	142 k	3.8 M	2.6 M	582 k	7.1 M	3 d 16 h
AGW	105 k	1.5 M	1.5 M	351 k	3.3 M	2 d 12 h

- **Speed**: our technique is **32% faster** than PULSE and **63% faster** than MOSFET
- **Efficiency**: PULSE used **~2x** the number of glitches and MOSFET **~5x**
- **Reliability:** AGW produces **30% the number of false positives** than MOSFET

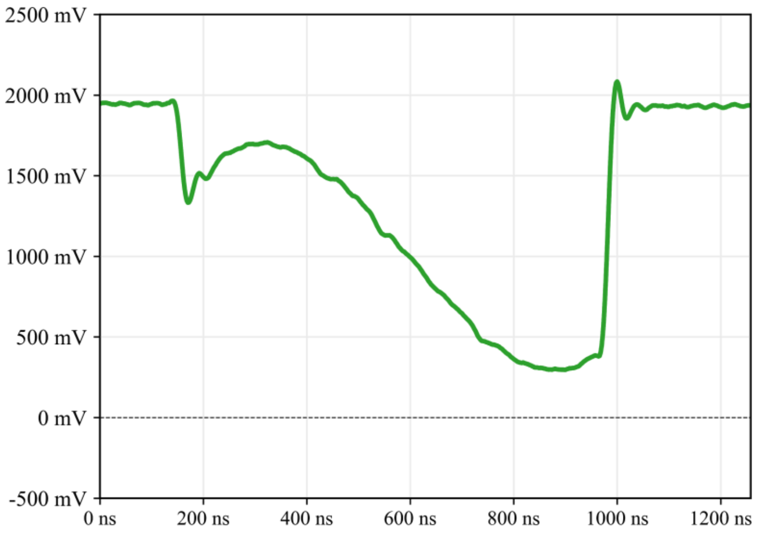
Different glitch waveforms provide the best performance for different vulnerabilities.



Verify 4-bytes



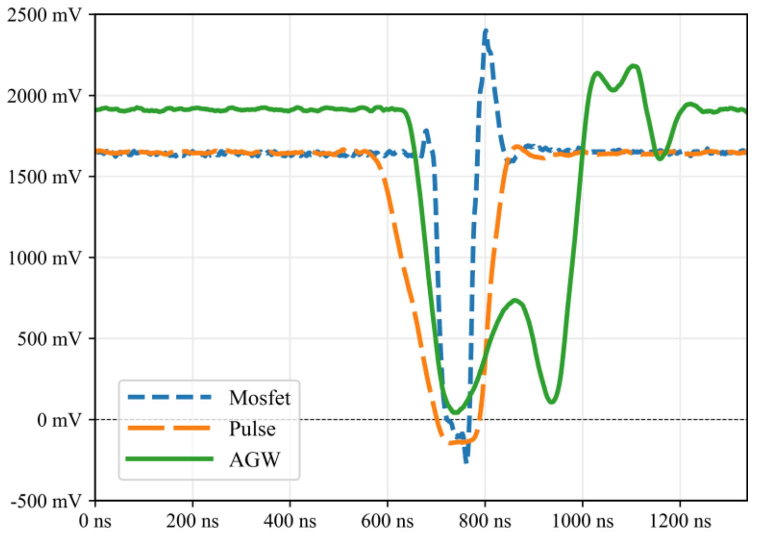
Checksum 4-bytes



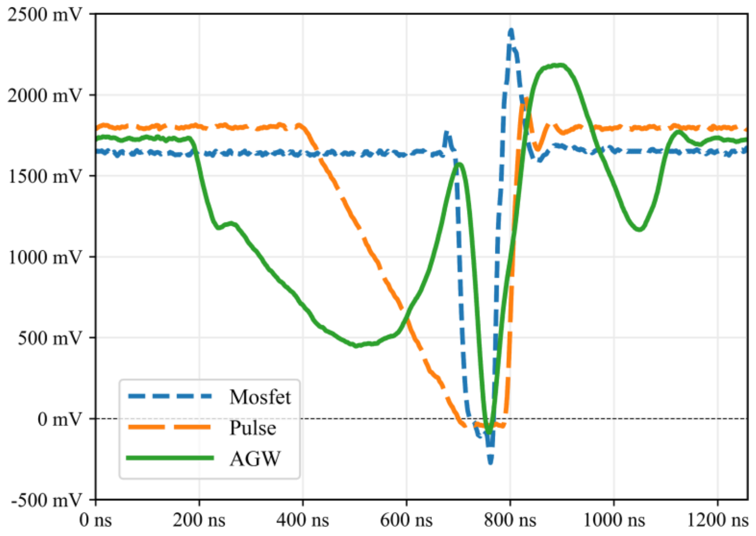
Checksum leak

Evaluation and comparison

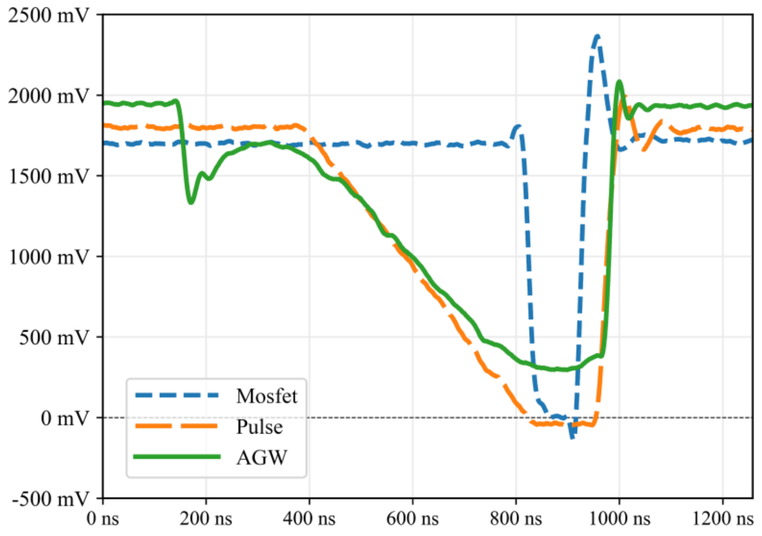
Comparison of the glitch waveforms / techniques for the Renesas attack.



Verify 4-bytes



Checksum 4-bytes



Checksum leak

Evaluation and comparison

Contributions

- Studied the **effects of Arbitrary Glitch Waveforms** on the performance of V-FI
- Investigated on the feasibility of **automatic attack parameter selection** and optimization using Genetic Algorithms
- Found unpublished vulnerabilities that enable **firmware extraction attacks** for **six microcontrollers** from by three major silicon manufacturers:
  - *STMicroelectronics* - **STM32F1** & **STM32F3**
  - *Texas Instruments* - **MSP430 F5xx** & **MSP430 FRAM**
  - *Renesas Electronics* - **78K0/Kx2** & **78K0R/Kx3-L**
- **In-depth analysis and evaluation** of the attack performance compared to other V-FI techniques

THANK YOU!

References

- [BFP19] C. Bozzato, R. Focardi, F. Palmarini. **Shaping the Glitch: Optimizing Voltage Fault Injection Attacks.** TCHES 2019.
- [ZDCR14] L. Zussa, J. Dutertre, J. Clediere, B. Robisson. **Analysis of the fault injection mechanism related to negative and positive power supply glitches using an on-chip Voltmeter.** HOST 2014.
- [OC14] C. O'Flynn, Z. Chen. ChipWhisperer. **An Open-Source Platform for Hardware Embedded Security Research.** COSADE 2014.

palmarini@unive.it