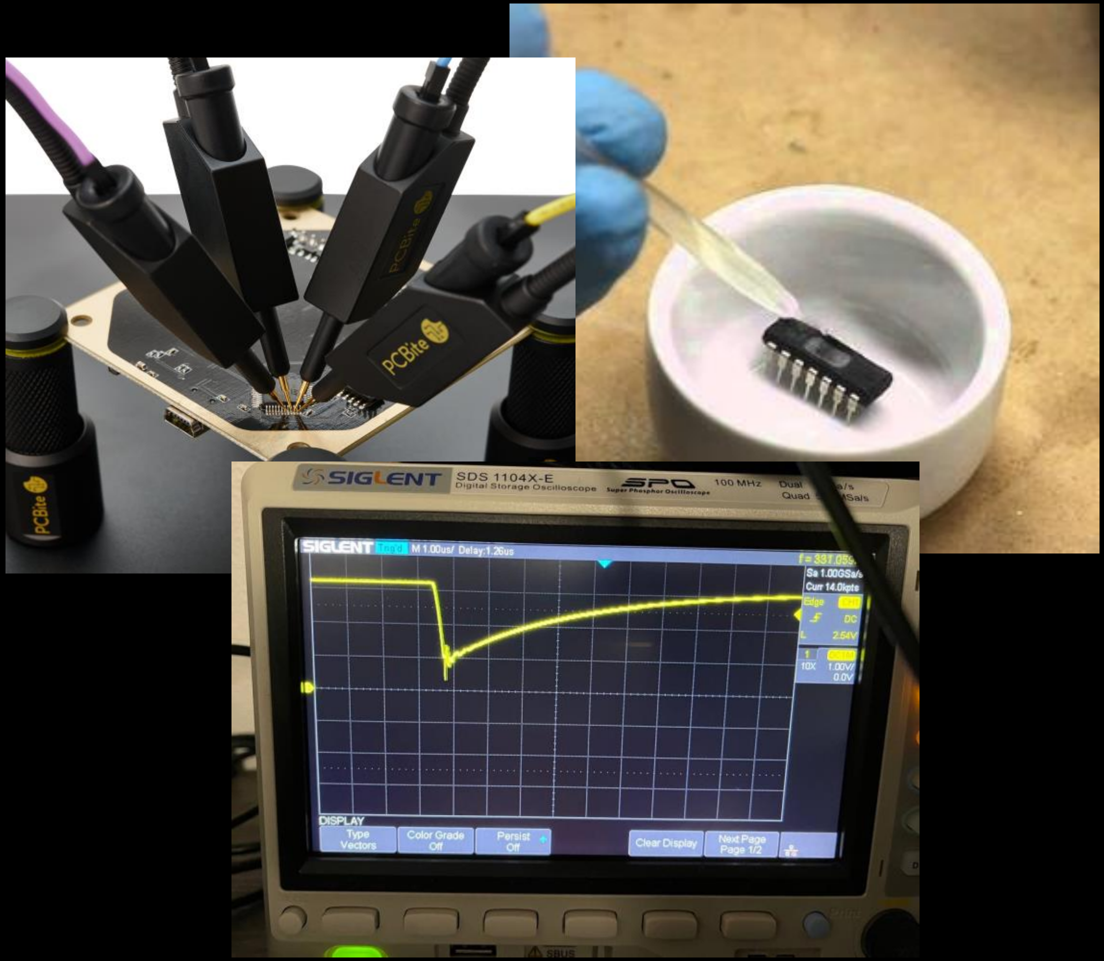
HARDWARE HACKING

AN INTRO TO EXTRACTION, FAULT INJECTION, AND POWER ANALYSIS

https://github.com/elbee-cyber

AGENDA

- Why hack hardware?
- Hardware Debugging
- Glitching
- Simple Power Analysis
- Advanced Forms of Power Analysis
- Countermeasures



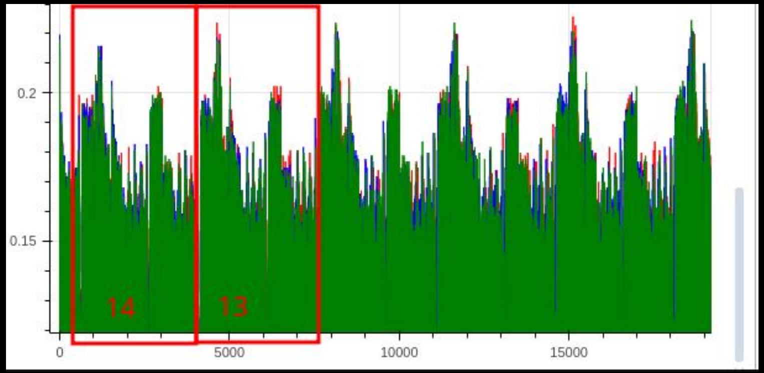
WHY DO WE HACK HARDWARE?

- Extract secrets (like universally used crypto keys!)
- Rooting or modification of devices (like bypassing secure boot)
- Extracting firmware (the first step in zero-day research!)
- Supply chain attacks

RESOURCES

- https://nostarch.com/hardwarehacking
- https://nostarch.com/microcontroller-exploits
- https://voidstarsec.com/blog
- Chipwhisperer juypter notebook
- Conference talks!!!

HISTORY



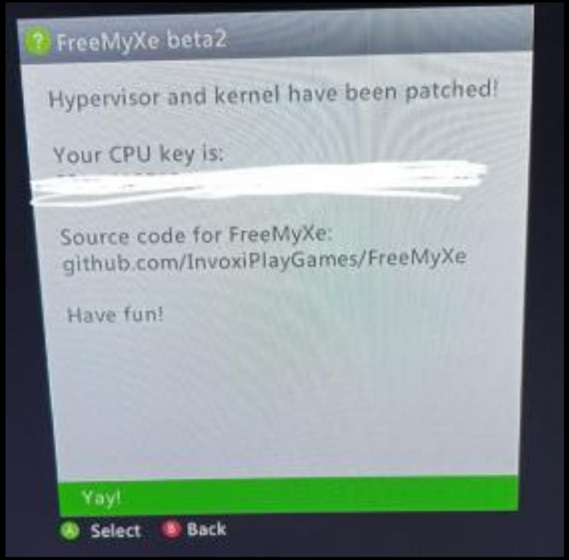
Power trace of AES decryption S-boxes.

POWER ANALYSIS

- Zigbee Hue Lights Key Extraction
  - Proximity-based worm (estimated only 15k lights need to be present for the worm to take over a city!)
- Google Titan Security Key
  - Recovery of key linked to card holder.

FAULT INJECTION

- Xbox 360 Reset Glitch
  - Booting unsigned kernel/hypervisors, resulted in large-scale modding and piracy.
- Trezor One SRAM Dump (wallet.fail)
  - Allowed dumping the seed phrase from a locked wallet.
- Airtags (nRF52)
  - Connecting to Airtag results in a rickroll.


FreeMyXe beta2

Hypervisor and kernel have been patched!

Your CPU key is:
[redacted]

Source code for FreeMyXe:
github.com/InvoxiPlayGames/FreeMyXe

Have fun!

Yay!

XB360 unlocked with modchip.

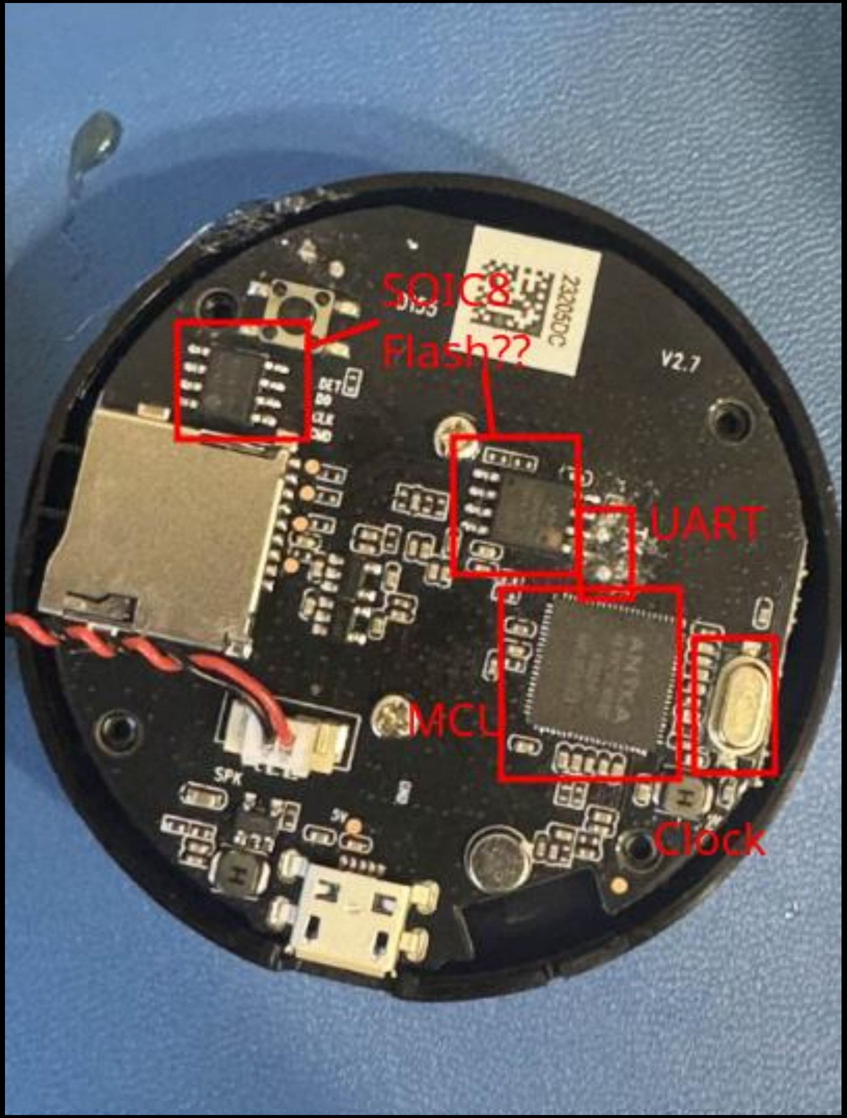
HARDWARE DEBUGGING

- How electronics work is beyond the scope of this talk.
- What they do isn't!

  UART – Serial interface (RX/TX)

  JTAG + SWD – CPU debugging

  Flash devices – Contain firmware
- A lot of the time, target interfaces are recognizable.
- These interfaces can be protected at both the firmware and chip level!



Geenie IoT camera internal photos.

FAULT INJECTION (GLITCH ATTACKS)

The kind

- Power supply glitching
- Clock/oscillator glitching
- Electromagnetic glitching
- Optical/laser glitching
- Many others!

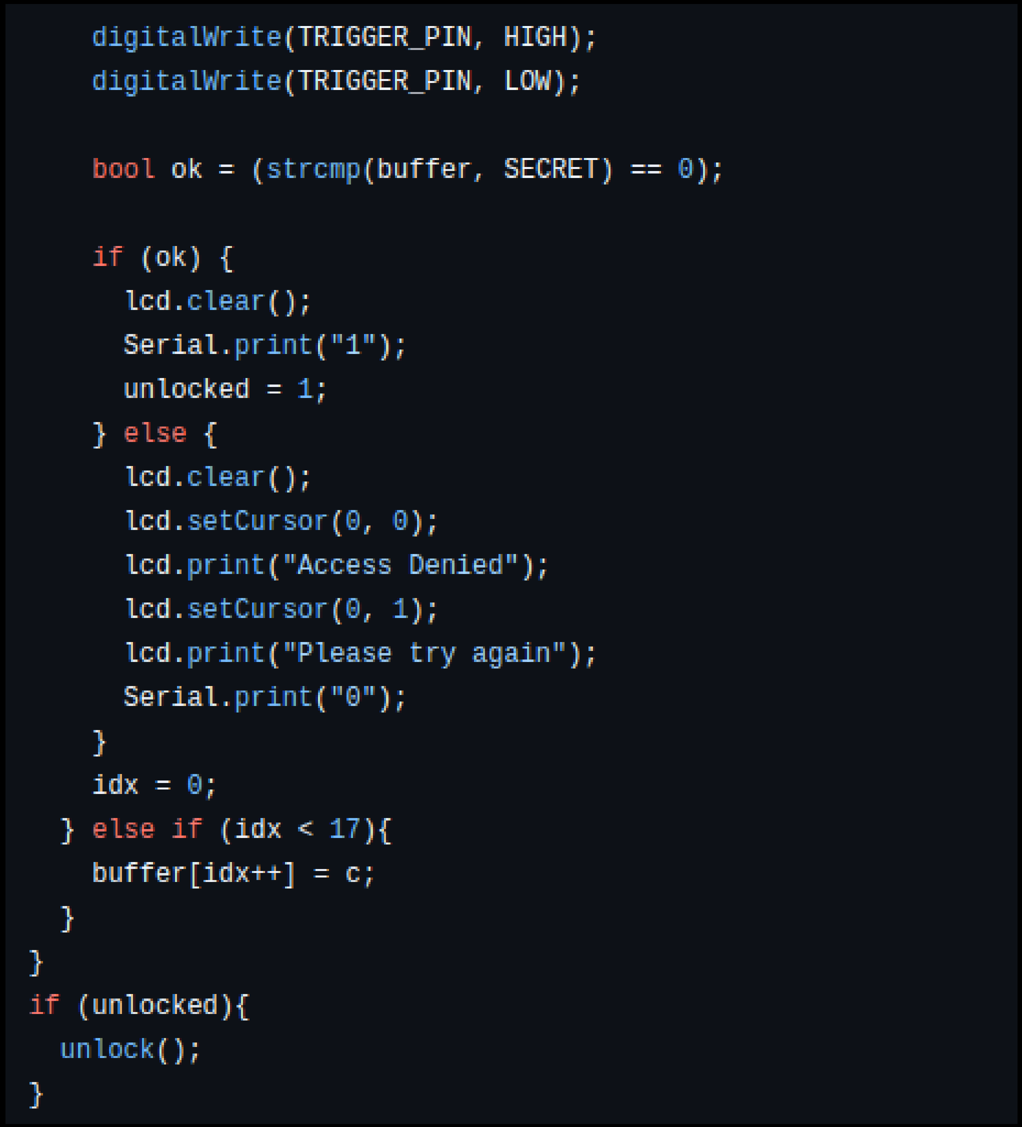
The effect

- Instruction skips
- Corrupted fetches
- Corrupted data (in registers, flash, etc)
- Resets

The desire

- Bypassing checks
- Corrupting protection bits
- Glitch -> memory corruption primitive
- Corruption of crypto (fault analysis)

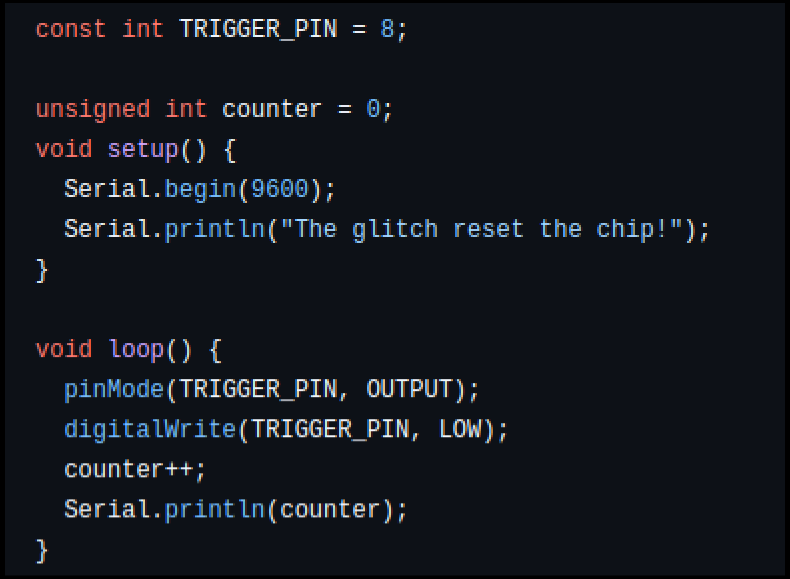
WHERE/WHAT COULD WE GLITCH TO UNLOCK?



```
    digitalWrite(TRIGGER_PIN, HIGH);
    digitalWrite(TRIGGER_PIN, LOW);

    bool ok = (strcmp(buffer, SECRET) == 0);

    if (ok) {
      lcd.clear();
      Serial.print("1");
      unlocked = 1;
    } else {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Access Denied");
      lcd.setCursor(0, 1);
      lcd.print("Please try again");
      Serial.print("0");
    }
    idx = 0;
  } else if (idx < 17){
    buffer[idx++] = c;
  }
}
if (unlocked){
  unlock();
}
```

FI: CHARACTERIZATION

- The process of building a fault model for your target.
- Parameters include: delay from a trigger, pulse width, pulse power (more depending on type of FI)
- Usually done with sweeping.
- Find the parameters that are not so high the board resets, but not so low that nothing happens.
- Flash target with custom helper firmware if possible!

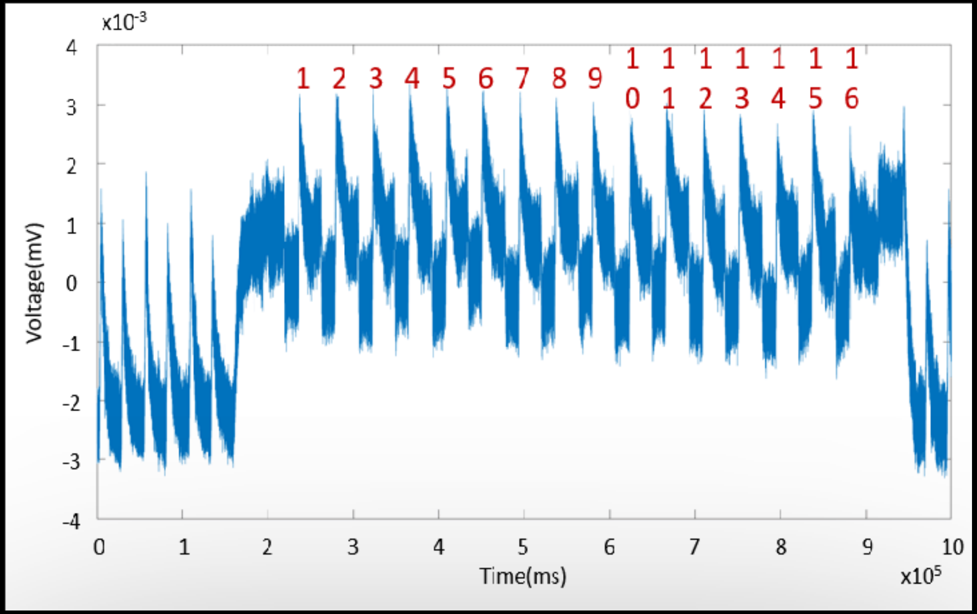


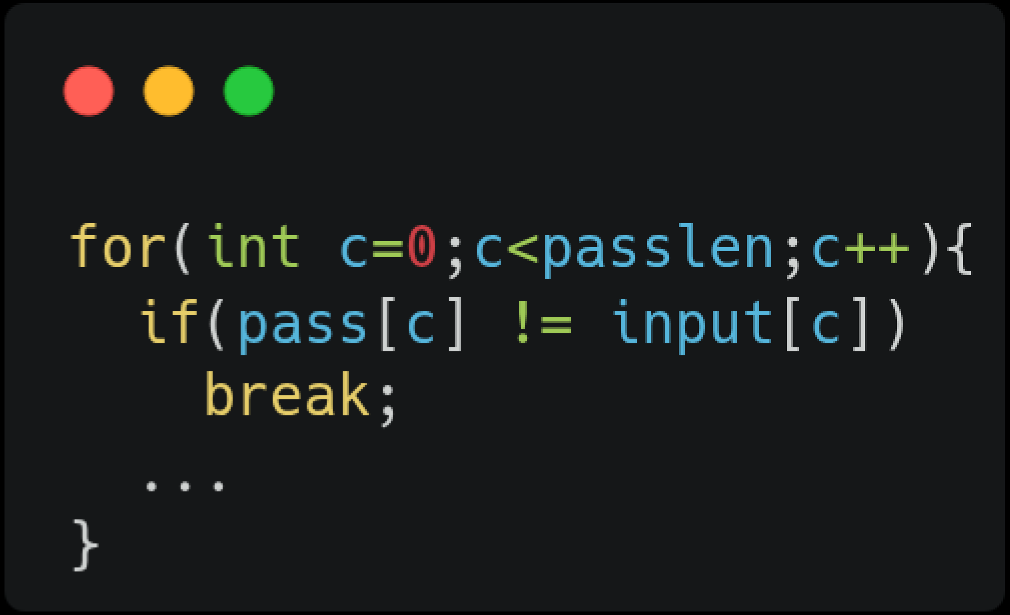
```
const int TRIGGER_PIN = 8;

unsigned int counter = 0;
void setup() {
  Serial.begin(9600);
  Serial.println("The glitch reset the chip!");
}

void loop() {
  pinMode(TRIGGER_PIN, OUTPUT);
  digitalWrite(TRIGGER_PIN, LOW);
  counter++;
  Serial.println(counter);
}
```

Characterization helper firmware.

SIMPLE POWER ANALYSIS

- Analyze a relationship between a software characteristic and the device's power consumption to leak data.
- Example: RSA square multiplyer (based on operation relationship), AES S-Box output (based on data relationship)
- For SPA, we use the relationship of program execution and the time differences in power consumption.
- Example: Char-by-char password comparison that terminates early once an incorrect character is found.

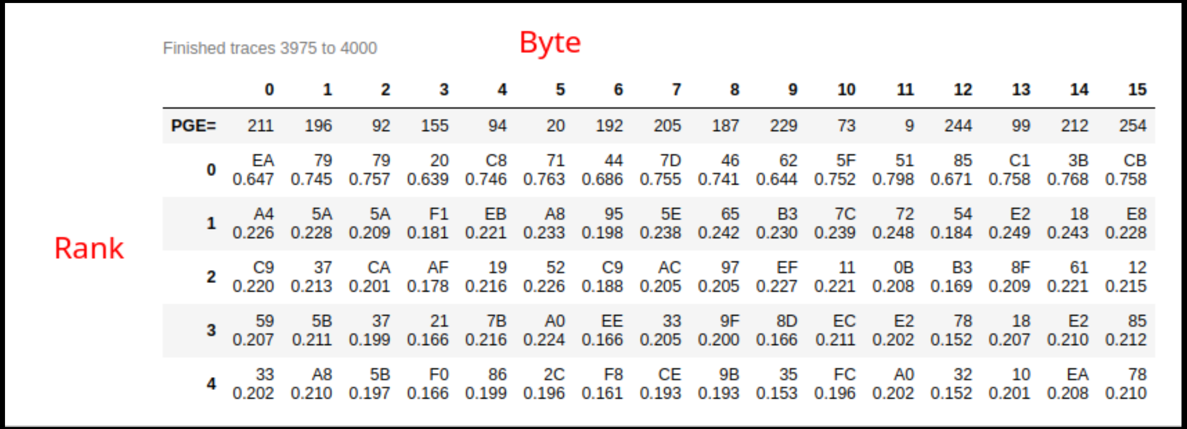




```
for(int c=0;c<passlen;c++){
  if(pass[c] != input[c])
    break;
  ...
}
```

ADVANCED POWER ANALYSIS (DATA-BASED)

- Even a change in a transistor in the die results in power differences.
- Much more subtle requires statistical analysis.



Finished traces 3975 to 4000

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PGE=	211	196	92	155	94	20	192	205	187	229	73	9	244	99	212	254
0	EA 0.647	79 0.745	79 0.757	20 0.639	C8 0.746	71 0.763	44 0.686	7D 0.755	46 0.741	62 0.644	5F 0.752	51 0.798	85 0.671	C1 0.758	3B 0.768	CB 0.758
1	A4 0.226	5A 0.228	5A 0.209	F1 0.181	EB 0.221	A8 0.233	95 0.198	5E 0.238	65 0.242	B3 0.230	7C 0.239	72 0.248	54 0.184	E2 0.249	18 0.243	E8 0.228
2	C9 0.220	37 0.213	CA 0.201	AF 0.178	19 0.216	52 0.226	C9 0.188	AC 0.205	97 0.205	EF 0.227	11 0.221	0B 0.208	B3 0.169	8F 0.209	61 0.221	12 0.215
3	59 0.207	5B 0.211	37 0.199	21 0.166	7B 0.216	A0 0.224	EE 0.166	33 0.205	9F 0.200	8D 0.166	EC 0.211	E2 0.202	78 0.152	18 0.207	E2 0.210	85 0.212
4	33 0.202	A8 0.210	5B 0.197	F0 0.166	86 0.199	2C 0.196	F8 0.161	CE 0.193	9B 0.193	35 0.153	FC 0.196	A0 0.202	32 0.152	10 0.201	EA 0.208	78 0.210

Correlation table of predictions and actual values from captured traces, 0 is no correlation, 1 is exact match. This is for leaking an AES256 key.

STEPS:

1. Physically modify the target for power analysis
   - Shunt resistor, removal of decoupling capacitors, etc, we care about noise.
2. Build a leakage hypothesis (this is what we're relating to data or operations executed!)
   - Eg: The hamming weight of the output of a round of AES.
3. Capture a lot of power traces (hundreds, thousands, sometimes millions)
4. Time alignment (if needed)
5. Do statistical analysis on captured traces.
   - Differential – Sum of Differences.
   - Correlation – Use the statistical correlation for the actual power usage and the hypothesis.

COUNTERMEASURES (BOARD)

- Decoupling capacitors, eliminates noise (SCA).
- Brownout detection (Crowbar FI).
- EM and optical shielding (FI).

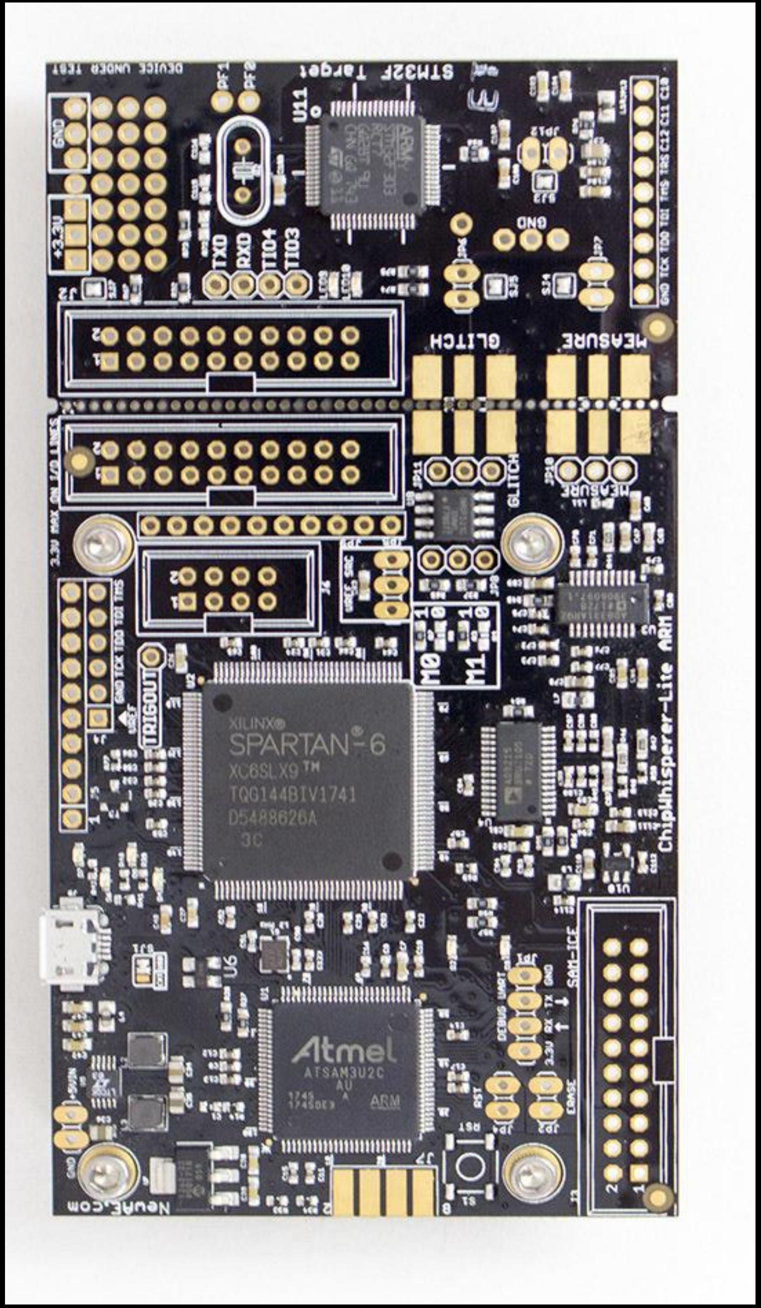
COUNTERMEASURES (FIRMWARE)

- Constant times across operations (SPA).
- Make important flags explicit (FI).
- Time desynchronization (time-based triggers).
- Redundant checks (FI).

No mitigation is good enough on its own!

CHIPWHISPERER LITE

- Connected target chip for learning (w Juypter notebook tutorials!)
- Good max clock speed for most microcontrollers.
- Features
  - Oscilloscope
  - Crowbar and clock injection
  - Pre-loaded modules for different types of leakage models and SCA attacks.



EMFI DEMO: CRYPTO WALLET UNLOCK

Target: ATMEGA2560

Faulter: FaultyCat – Based on PicoEMP, configurable via UART.

Considerations

1. Target modification?
2. Parameters?
3. Sweeping considerations and firmware?