IoT Goes Nuclear: Creating a ZigBee Chain Reaction

Official Slides :
https://eyalro.net/pdf/IoTSP17.pdf

Skyler Hau, Peter Hevrdejs, Paul George,
Brendan Freeman, Harshit Potu

Presentation Outline

I. Introduction
II. Related Work
III. Background for Attack
IV. Hardware Setup
V. Creating a Light Bulb Worm
VI. Power Analysis
VII. Take Over Attack
VIII. Worm Results
IX. Repercussions of the Worm and Countermeasures
X. Conclusion

Introduction

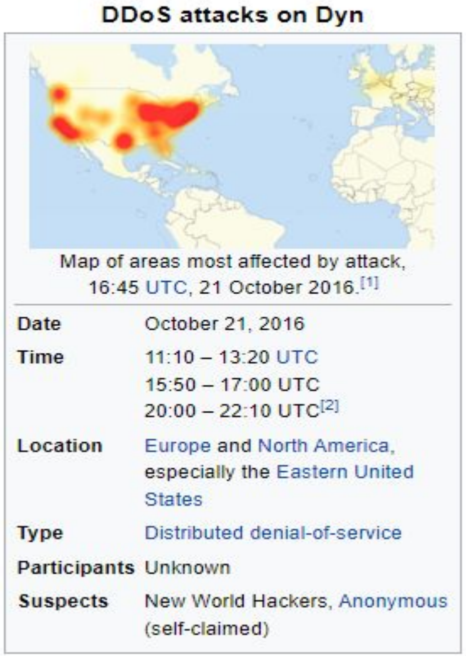
- Internet of Things (IoT) is currently going through exponential growth.
- Most of them are :

  - Cheaply made sensors

  - Cheaply made actuators
- Due to the rapid increase of IoT devices there are potential dangers.
- There were three consecutive Distributed Denial of Service (DDoS) attacks on Dyn DNS Company (Software Company) on 21st October, 2016.

  - Exploited attack vectors (default passwords)

  - Took control of millions of Web Cameras



DDoS attacks on Dyn

Map of areas most affected by attack, 16:45 UTC, 21 October 2016.[1]

Date	October 21, 2016
Time	11:10 – 13:20 UTC 15:50 – 17:00 UTC 20:00 – 22:10 UTC[2]
Location	Europe and North America, especially the Eastern United States
Type	Distributed denial-of-service
Participants	Unknown
Suspects	New World Hackers, Anonymous (self-claimed)

Fig 1 DDoS attack

Introduction

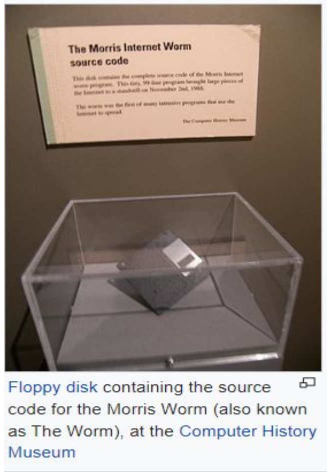
- In this paper they demonstrated an attack on Philip Hue Smart Lamps.
- Communication protocol (lamps and controller): Zigbee
- Zigbee Chip : Made by Atmel
- Initial Discovery : Atmel stack has a major bug in its proximity test.
- This bug enables any standard Zigbee transmitter to initiate a Factory reset procedure.
- Later, they checked if it can cause a permanent damage through a firmware update which had to be encrypted and authenticated by AES-CCM (Advanced Encryption Standard _ Counter)

Fig 2 Zigbee protocol

Fig 3 Philips Hue Smart lamps

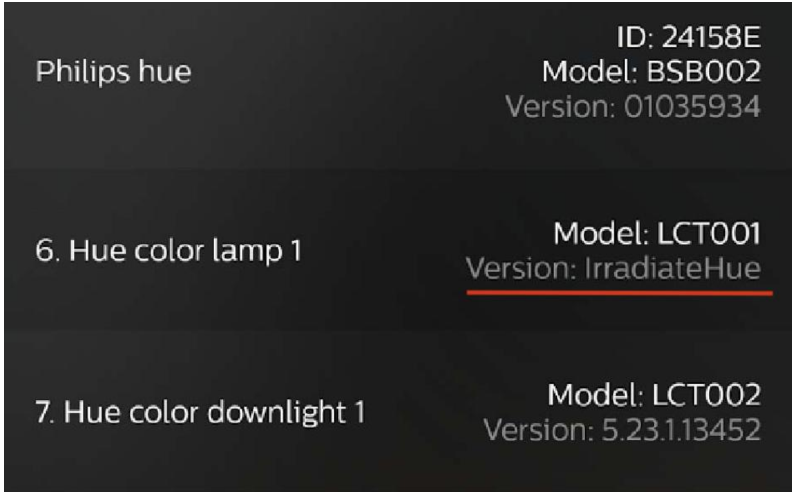
Introduction

- This attack was similar to the worm scenario triggered by Robert Morris Jr on November 2, 1988 where whole internet brought to standstill within minutes.
- To avoid accidental outcome they just changed the firmware ware version number string to “IrradiateHue”.



Floppy disk containing the source code for the Morris Worm (also known as The Worm), at the Computer History Museum

Fig 4 Morris Worm



Philips hue
ID: 24158E
Model: BSB002
Version: 01035934

6. Hue color lamp 1
Model: LCT001
Version: IrradiateHue

7. Hue color downlight 1
Model: LCT002
Version: 5.23.1.13452

Fig 5 Outcome of the new attack

Introduction

- New attack differs from previous attacks in the following ways:

  - First, previous attacks used TCP/IP packets to scan the internet whereas new attack uses unmonitored and unprotected Zigbee communication making infections jump from lamp to lamp.

  - Second, new attack spreads via physical proximity alone, disregarding the established network structures. It is similar to air borne biological infection.

  - Finally, previously reported attacks are carried out via linear scans and infections with a centrally located attacker in a star shaped structure whereas new attack is a chain reaction where each infected lamp becomes the new source of infection.

Related Work

- Regarding connected lamps, several vulnerabilities were discovered:

  - Alex Chapman[2] extracted hard coded encrypted keys which are used to extract data sent between LIFX brand bulbs and recovered Wifi password of local network.

  - Dhanjani[3] had shown denial of service (DoS) attacks against Philips Hue.

  - Ronen and Shamir[4] have shown how to use Philips hue and LimitlessLed systems to exfiltrate data from air gapped networks and to create strobes that can cause epileptic seizures.

  - Heiland[5] found weakness in Osram lightify app (such as unencrypted passwords, lack of authentication in the gateway and vulnerable usage of Zigbee Home Automation profile).

Related Work

- Regarding ZigBee Light Link (ZLL) and related products:

  - Armknecht[6] proposed a formal security model.

  - Zilner[7] and Morgner[8] demonstrated weakness in ZLL and ways to take over lamps.

  - O’Flynn[9] reverse engineered some of the Philips Hue security design choices, where he raised the possibility of a lightbulb worm, but did not bypass the firmware or provide a spreading mechanism.

  - Carettoni[10] described a type of worm which spreads through bluetooth enabled cellular phones.

Whereas this new attack on stationary IoT devices exploits previously unknown weaknesses in the implementation of ZigBee protocol.

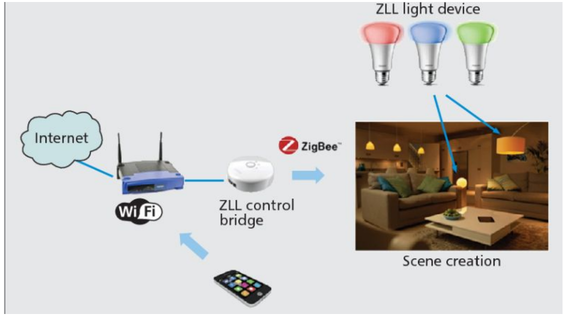


Fig 6 ZLL application

Related work

- Kizhvatov[11] had done the first power analysis attacks on Atmel AES hardware accelerators against Atmel XMEGA using AES-ECB mode.
- O'Flynn and Chen[12] used the same leakage model to attack Atmel MegaRF128RFA1 hardware, and attacked ZigBee CCM mode of operation under an assumption of a known nonce
- Jaffe[13] had shown an attack on counter mode encryption with unknown nonce but would require 2^16 sequential block operations.

Overview of Attack

- This attack is much stronger than previous attacks as it creates the first native and autonomously self spreading ZigBee worm.

- It is a combination of two novel attacks:

  - Correlation Power Analysis (CPA) attack

  - Take Over Attack

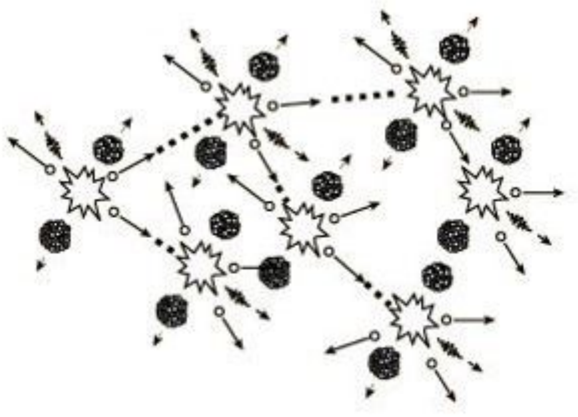


Fig 7 Zigbee chain reaction

Overview of Attack

- **CPA Attack:** Targets the verification phase of CCM Mode and has several advantages.
  - It does not assume any knowledge about the nonce.
  - It works with any type of counter implementation.
  - It does not require any validation encryption sample.
  - It requires at most the verification of messages as short as 2 blocks.
  - It requires at most twice the number of traces required to break the ECB mode.

This novel attack is used to recover the OTA update verification and encryption keys of Philips Hue Smart lamps.

Overview of Attack

- **Take Over Attack:** It uses a bug in Atmel's implementation of ZLL Touchlink protocol state machine to take over lamps from a large distance of 70m indoors or 400m outdoors.
  - This attack does not assume any prior knowledge about attacked lamps.
  - It does not require the knowledge of ZLL's secret master key.
  - This attack can run simultaneously on all lamps within range.
  - It can be used in wardriving and warflying scenarios.

Finally, this allows the individual to take full control over lamps from long distances without using custom hardware.

Creating Widespread Infections

Consider a city.

- Area is ‘A’ and shape is ‘Roughly Circular’.
- ‘N’ smart lamps at random locations.
- Defined an infection graph by connecting two lamps whose distance is smaller than ‘D’ by an edge.

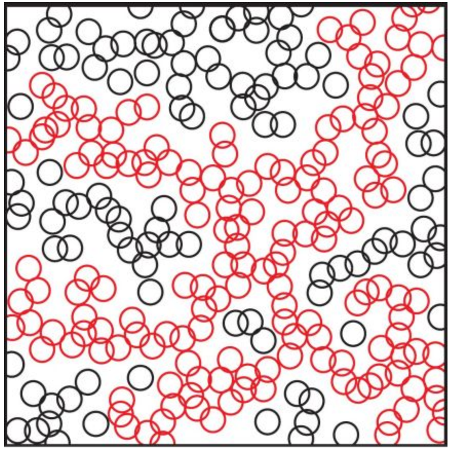


Fig 8 2D continuum with disks

- As two points are within distance D from each other if and only if the two disks of radius R = D/2 around them intersect and can use this model to find critical mass. (which is 1.128 larger than A of the city).

Now,

For Example : For a city like Paris (A = 105 square km), ZigBee (D=100m, R=50m)

$$N = 1.128A/\pi(D/2)^2,$$

N = 15,000

ZigBee Light Link and SmartLight Systems

- ZLL Architecture
  - "Interoperable and easy-to-use consumer lighting"
  - Full wireless controls over lighting systems

- Supported by big manufacturers (Phillips, GE, etc)
- 3rd parties can access bridge via Phillip's open API

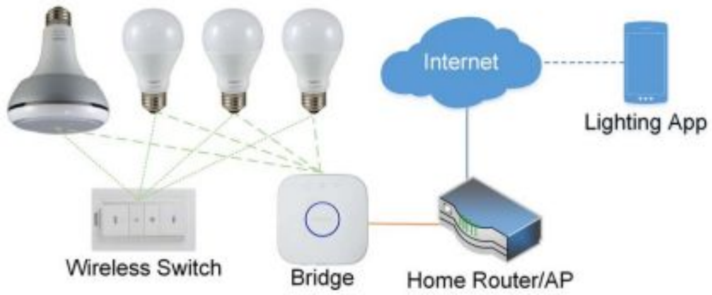


Figure 2. The ZLL architecture.

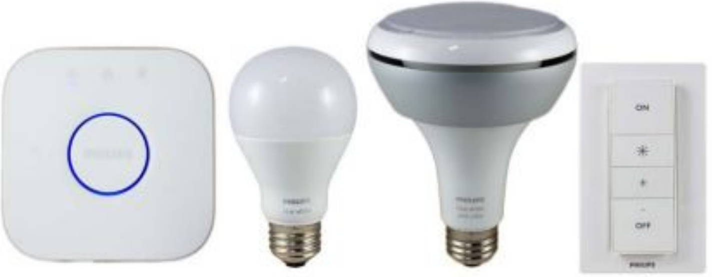


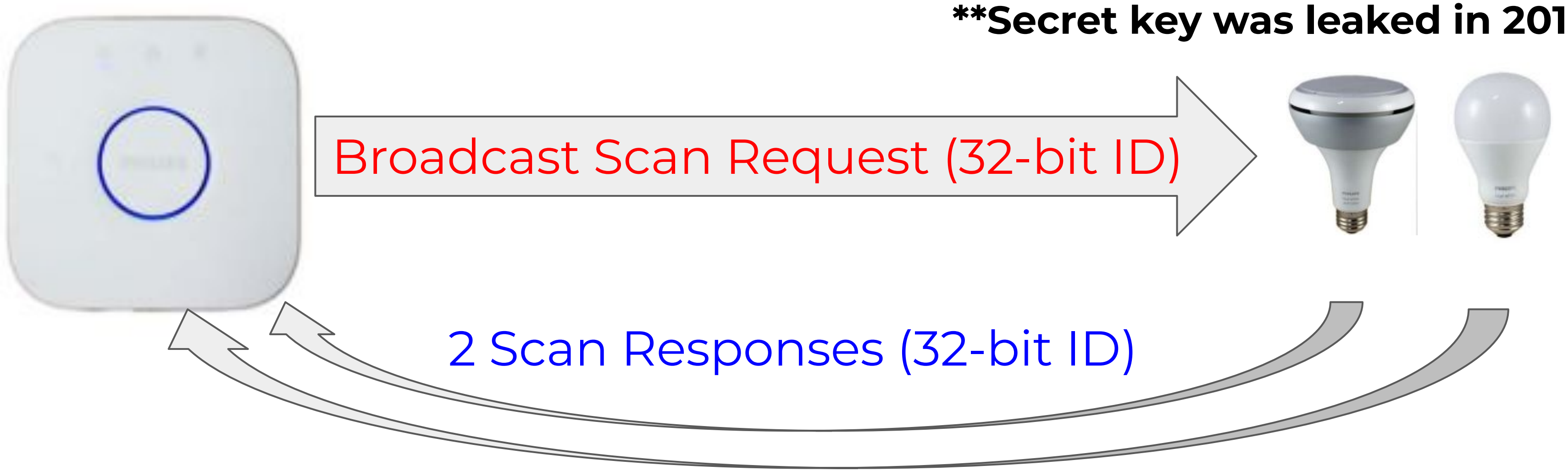
Figure 3. Philips Hue bridge (gateway), lamps, and wireless switch.

ZLL TouchLink Protocol

- PANs (Personal Area Network): collection of devices
- Lightbulbs are preconfigured to be on same PAN via shared secret key

- ZLL messages are unsigned/unencrypted, but the shared secret key for new network members is registered on all ZLL-certified products

**Secret key was leaked in 2015



Broadcast Scan Request (32-bit ID)

2 Scan Responses (32-bit ID)

Communication Capabilities

- Messages:
  - Factory Reset
  - Join (or start) network
- Join Messages includes:
  - PAN's unique key
  - ZLL master key + Transaction + Response ID
- OTA updates: do not require asymmetric verification of authenticity/integrity even though it is suggested
- Lamps use ATMEL SOC
- Authors assume usage of Atmel's open-source Bit-Cloud ZLL stack
- Touchlink proximity check (1m, 45cm, then 75cm)

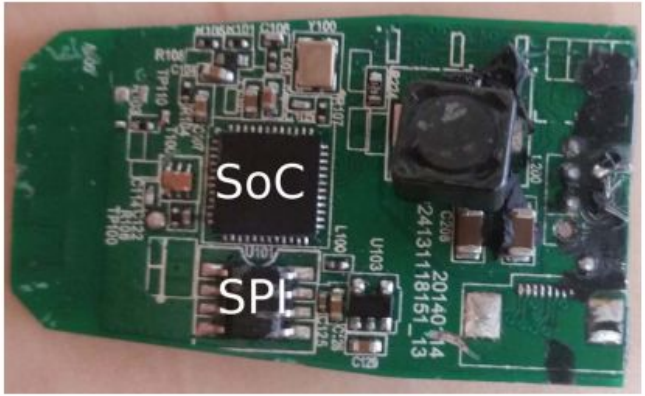


Figure 4. Philips Hue board

ATmega SOC overview

1. Atmel AVR microprocessor
2. 256KB bootloader/firmware
3. 32 KB program data
4. AES HW accelerator
5. IEEE 802.15.4 low-power radio transceiver

Experimental Setup

- Hardware Setup
  - TI CC2531EMK evaluation board: same family chip and ZLL stack as Philip Hue Lamps
  - RF transceiver to send/recv messages
- Software Setup:
  - Python impl. of ZLL for easy parsing
- Attack model uses logic in C code using TI's ZigBee stack
- Power-measurement board on right

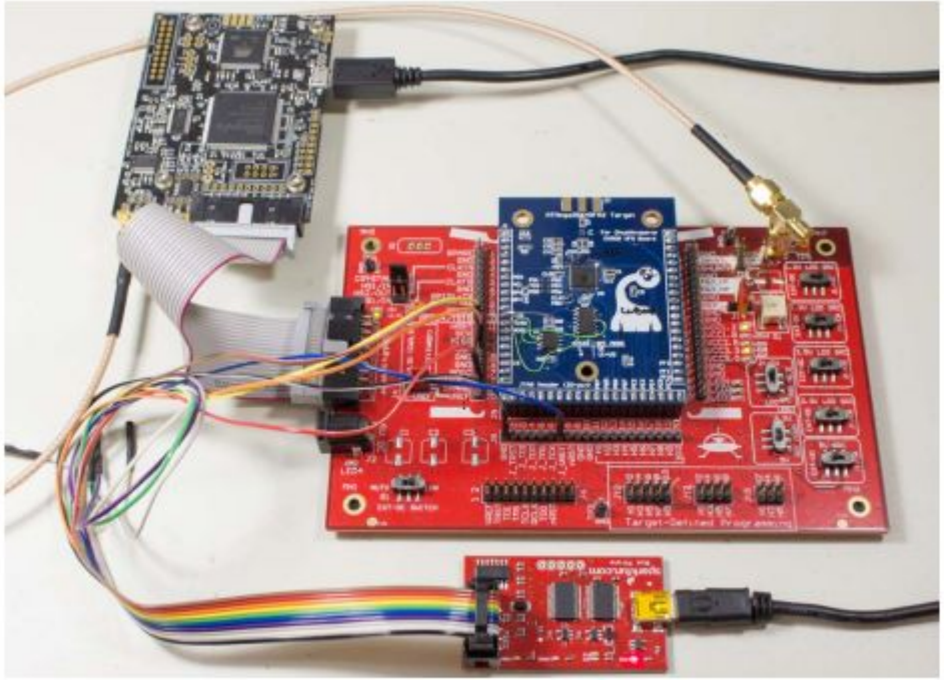


Figure 5. ChipWhisperer-Lite (top left), connected to a custom PCB with the ATMega2564RFR2 mounted (middle blue PCB) and a Bus Pirate (bottom small PCB) to reprogram the SPI flash chip

Experimental Setup Cont.d

- Power Analysis
    - Some boards loaded with production-grade Hue chips for breaking encryption/key
    - Some with blank chips for measuring power traces to break hardware AES peripheral
    - Performed using ChipWhisperer
- More details in appendix

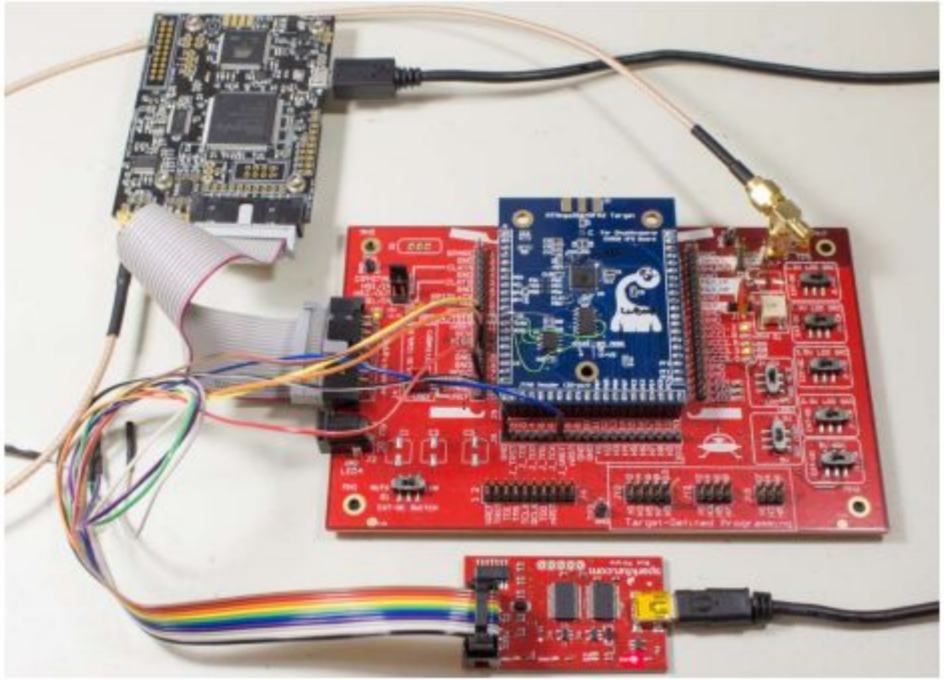


Figure 5. ChipWhisperer-Lite (top left), connected to a custom PCB with the ATMega2564RFR2 mounted (middle blue PCB) and a Bus Pirate (bottom small PCB) to reprogram the SPI flash chip

Light Bulb Worm Design

- Goals:
  - Automatically spread over a large area
  - Utilize the chain reaction of light bulbs infecting each other
- Requirements:
  - Ability to update and store new firmware
  - Ability for lights to infect each other

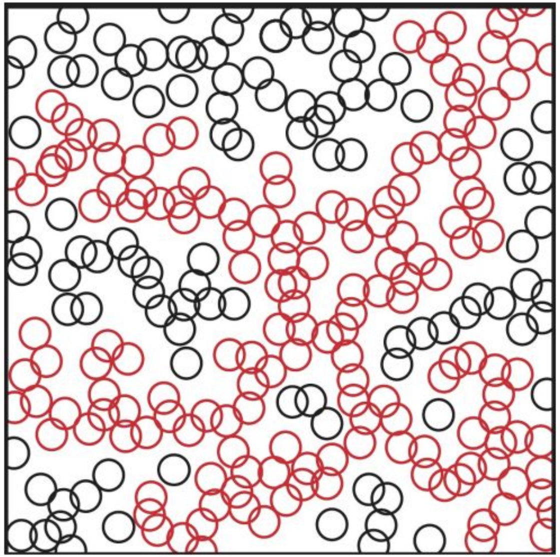


Figure 1. 2D continuum percolation with disks (figure from [15])

Installing Custom Firmware

- Relatively secure processors
- Investigated software updates using python simulation
- Found the lights likely used symmetric cryptography
- Done in five steps:
  - Using power analysis to break bootloader & retrieve keys
  - Reverse engineer firmware updates
  - Encrypt new firmware
  - Decode SPI interaction with SPI sniffer
  - Update firmware in flash via SPI

Correlation Power Analysis

- Uses power data to get information about intermediate values
- Based on DPA, which does one bit at a time
- Number of ‘1’s in bus is directly correlated with power when computed
- If measured at the right time, can yield pretty clear results

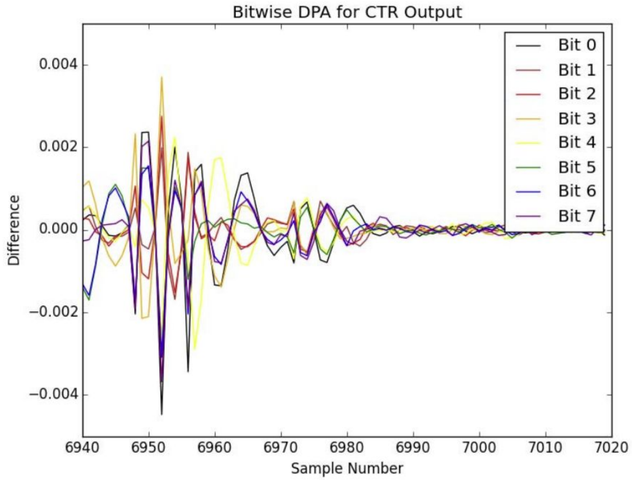


Figure 6. Bitwise DPA attack on AES-CTR ‘pad’, where all 8 bits are recovered.

Infecting Each Other

- ZigBee Over-the-Air Upgrading Cluster
- Must be on the same network
- Must have the same key
- Must be nearby
  - Requires close proximity or high strength transmitter
  - Solved with security vulnerability
- Touchlink commissioning protocol to take over each other
  - ZLL master key has been leaked

Counter with CBC-MAC Encryption

- Nonce N combined with counter
- Which is then encrypted with AES
- Calculate CMC-MAC authentication tag
- Xor AES output with data
- Combine output and tag and return

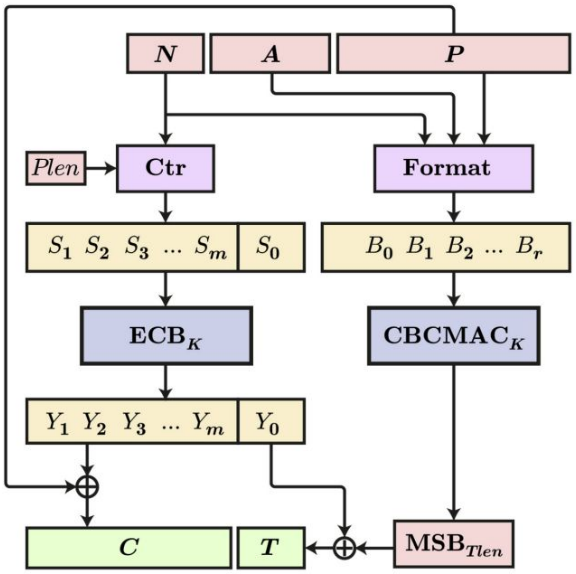


Figure 7. CCM encryption mode

Constraints

- No knowledge of key
- No knowledge of nonce
- No knowledge of signature
- No sample of a valid encrypted message
- No knowledge of input mapping
- Message length limited to ~2^14 bytes
    - One strategy is encrypting 2^16 bytes, one for each counter value

Breaking AES-CCM

Attack CBC-MAC engine used in decryption

Let ECB(electronic code book) with key k be E(k).

A CBC Initial State IV is a constant for a given decryption.

CBC Next State is a function of the Previous State , Ciphertext and Counter.

$$CBC_m = E_k(PT_m \oplus CBC_{m-1})$$

$$CBC_m = E_k(CT_m \oplus CTR_m \oplus CBC_{m-1})$$

$$CBC_m = E_k(CT_m \oplus CONST_m)$$

$$CBC_m = E_{kMOD}(CT_m)$$

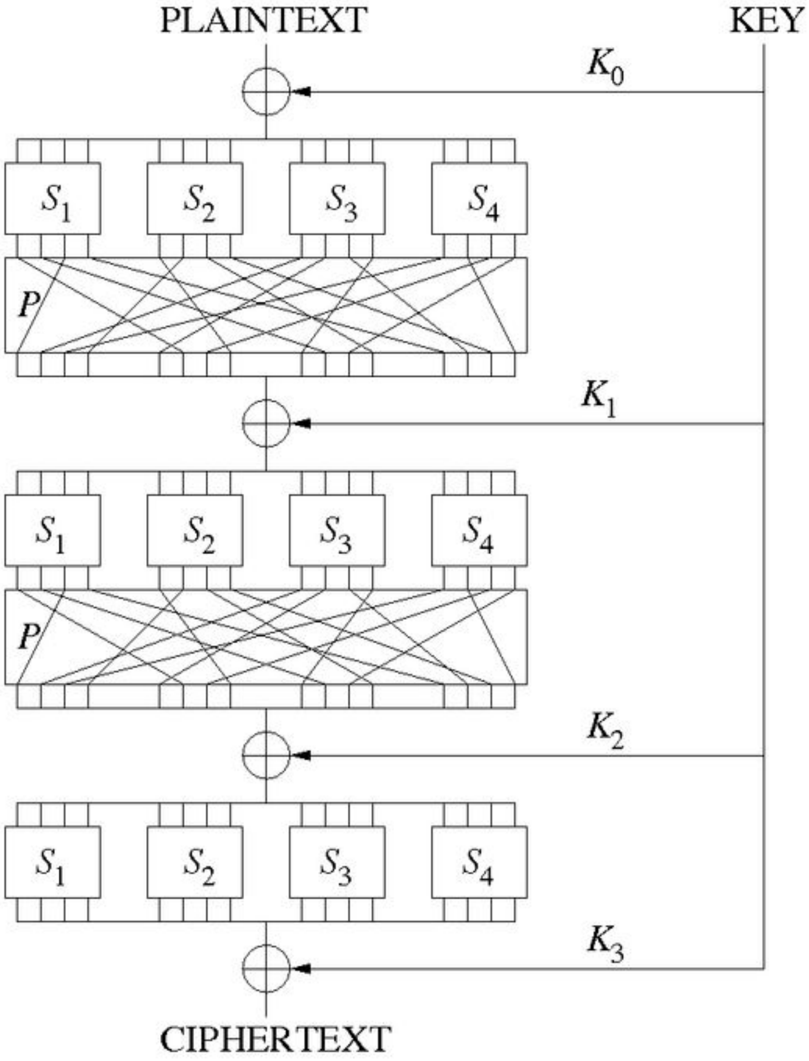
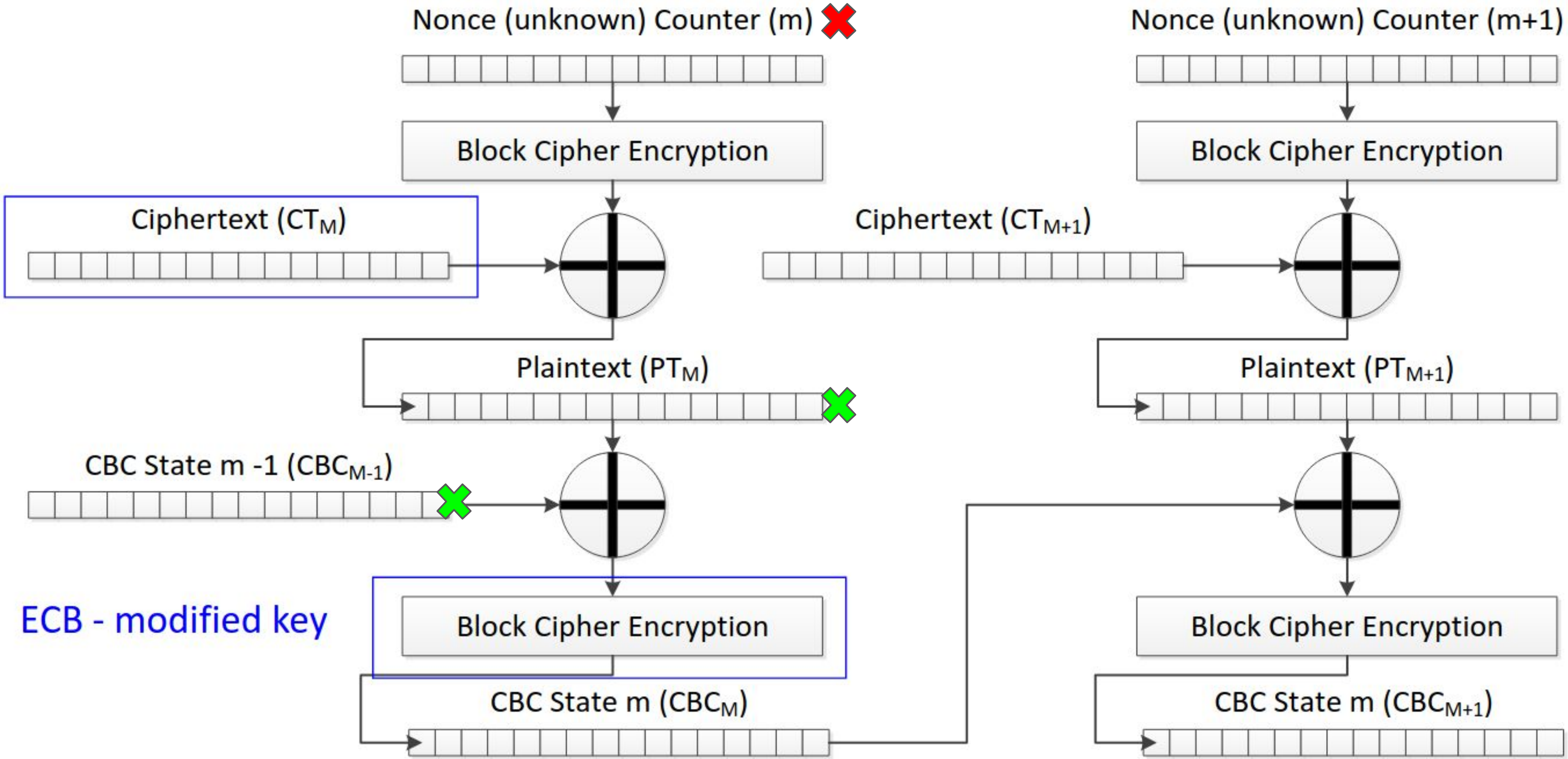


Image: https://eyalro.net/project/iotworm/IotGoesNuclearIEEESP17.pdf

New CPA attack on CCM



Leaky XOR

AES-CTR XOR operation leaked bytes {Nonce,CRT}

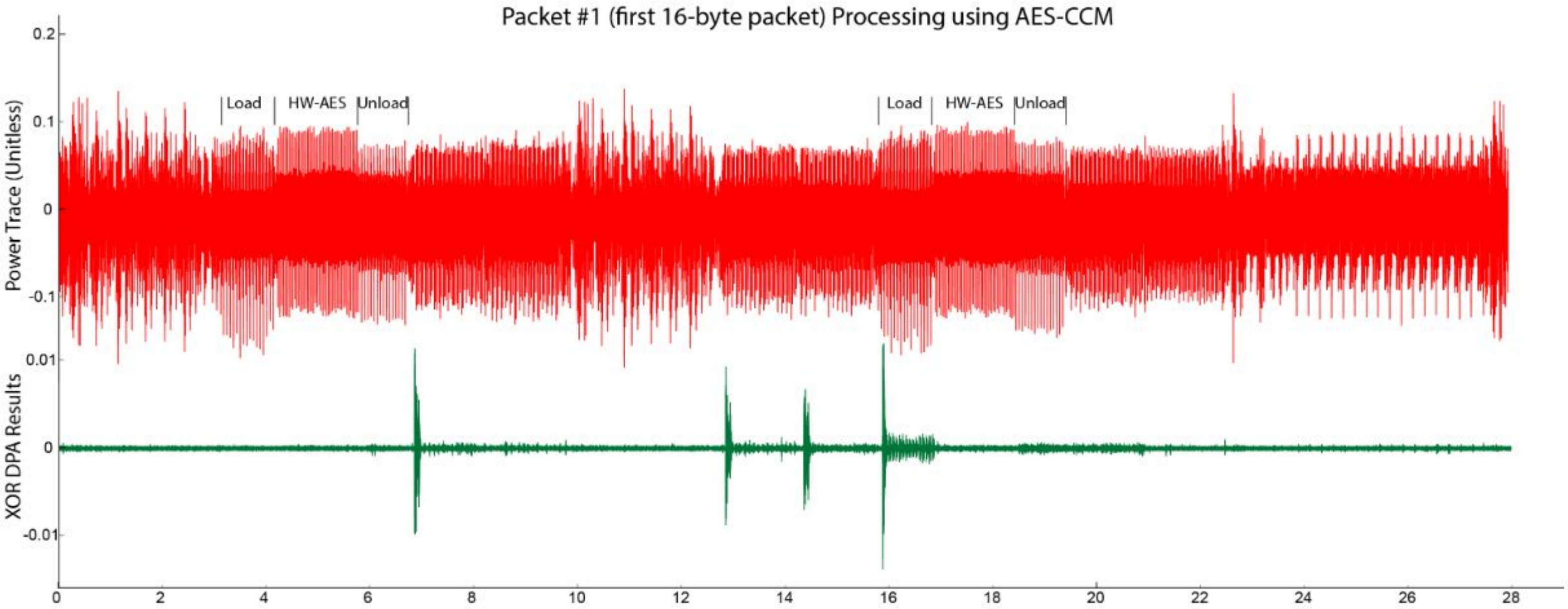


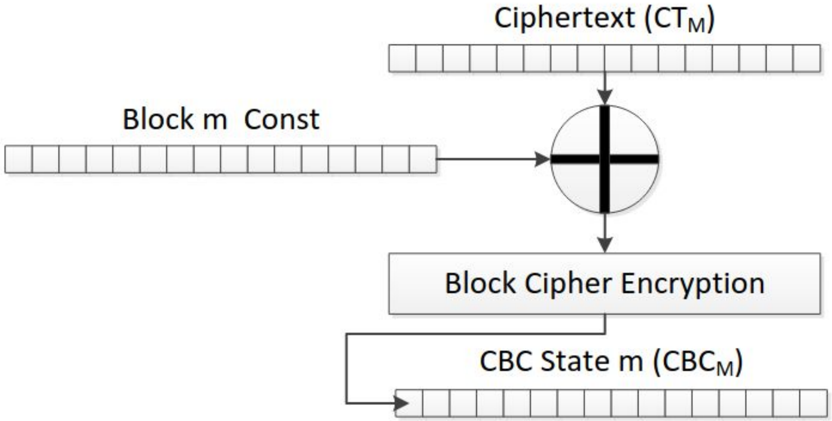
Figure 8. Power analysis of processing a single 16-byte block by the cryptographic bootloader.

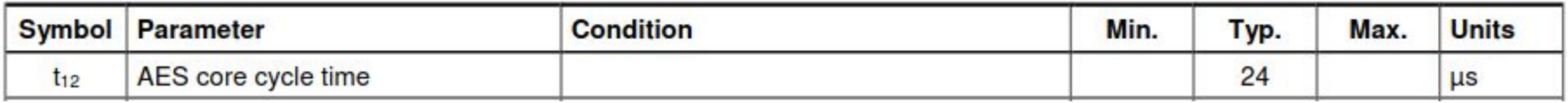
Breaking AES-CCM

Nonce format (software generated), CBC Start State (IV) are constant for a given block.

CPA applied Generally to Recover E_{kMOD}

Ciphertext with differential for E_{kMOD} Used in CPA (at second round) to recover the real CBC MAC key k of E_k





Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
t_{12}	AES core cycle time			24		µs

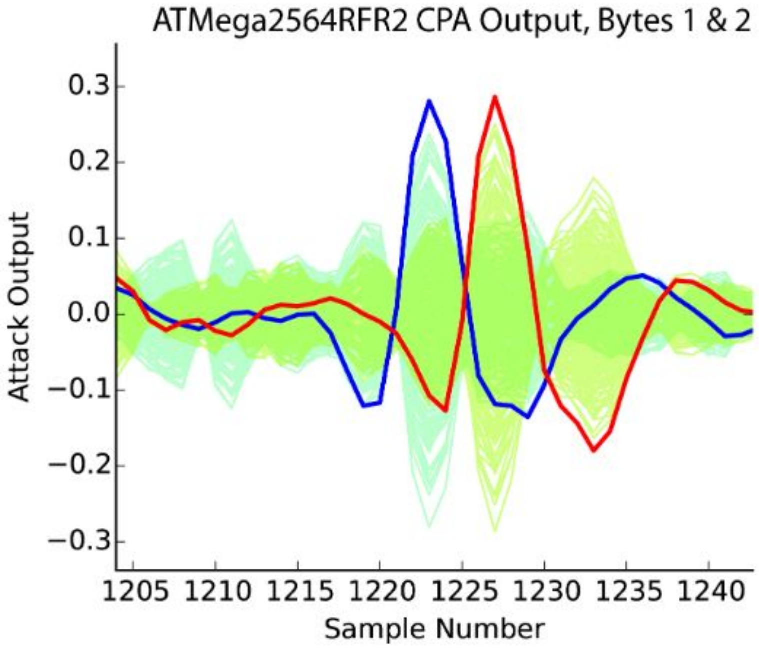


Figure 12. Correct values of the correlation analysis attack for byte 1 (in blue) and byte 2 (in red) compared to all incorrect guesses (in light cyan and green) show both the positive and negative peak we can exploit.

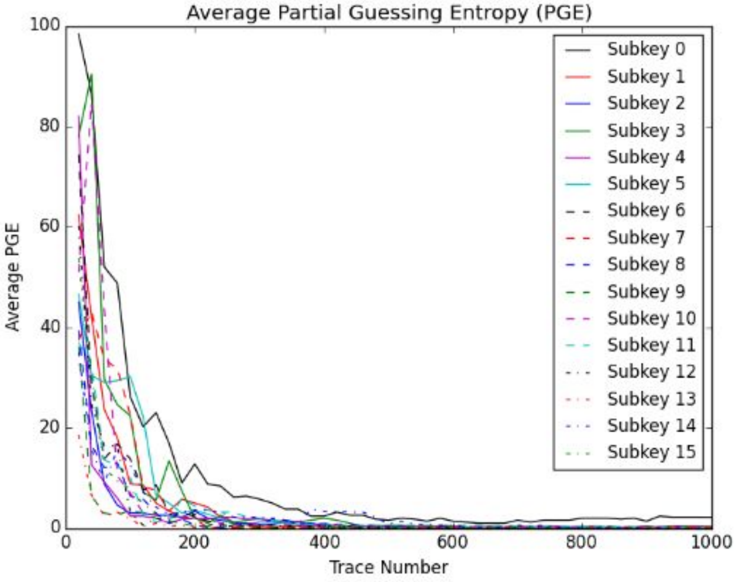
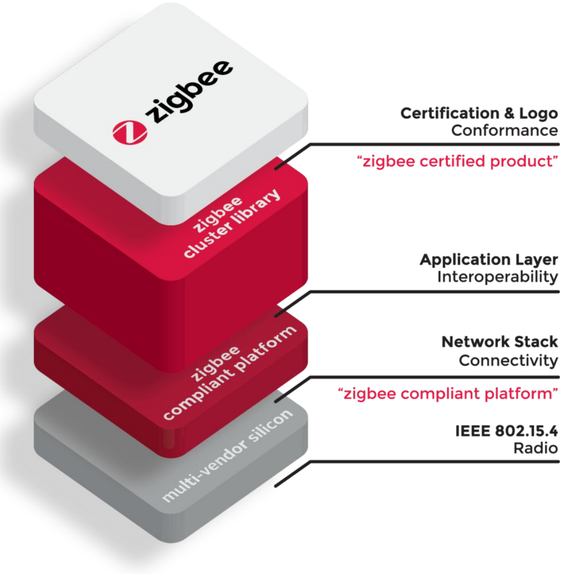


Figure 13. PGE of ATMega2564RFR2 Hardware AES Peripheral – a PGE of 0 indicates that encryption key byte is fully recovered.

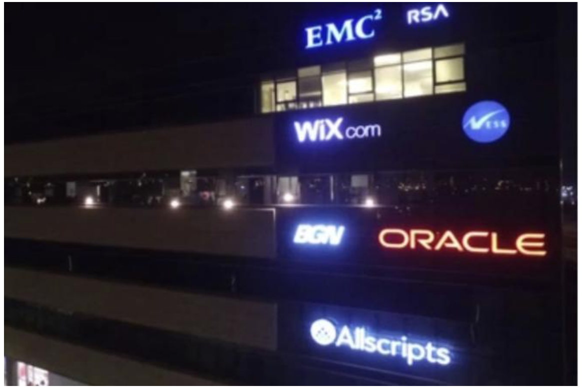
Take Over Attack ; Proximity check defeat mechanism

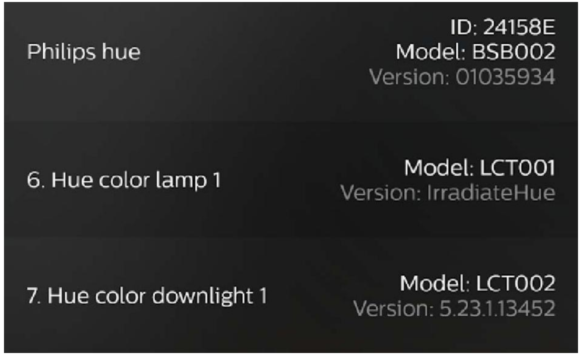
- Spec: Proximity Check (w/ RSSI) used to physically limit configuration interface access range.
- Philips used Atmel`s “Bitcloud” ZLL stack with publicly available source code. ZLL Protocol requires backwards compatibility with alternate zig-bee implementations
- Packets sent on non-ZLL protocols but signed with ZLL Master Key were not subject to Proximity/ Packet sanity checks enabling the use of broadcast commissioning Reset/Join commands.
- The Reset devices are subsequently navigated away from Zigbee primary channels so that they may not be reset again.


zigbee
zigbee cluster library
zigbee compliant platform
multi-vendor silicon
Certification & Logo Conformance
“zigbee certified product”
Application Layer Interoperability
Network Stack Connectivity
“zigbee compliant platform”
IEEE 802.15.4 Radio

Worm Results

- Attack was performed in two parts:
  - **Take Over Attack**
    - Factory reset bulbs and reassociate them
    - Demonstrates capability to infect at long distances
  - **Firmware Update Attack**
    - Breaking the bootloader with DPA / CPA
    - Changed version number



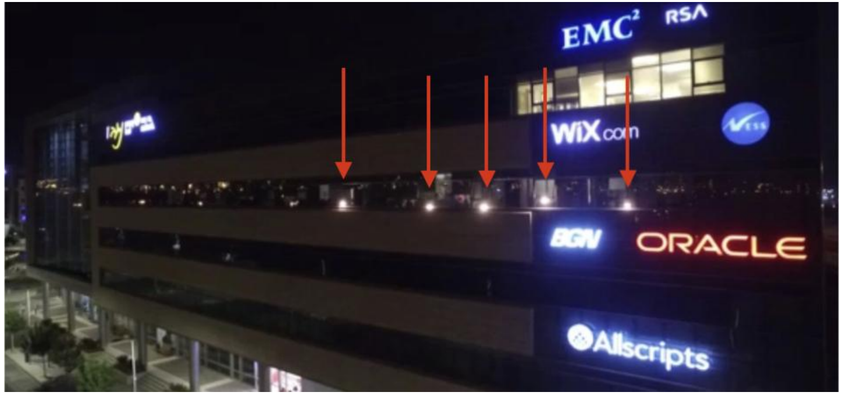

Philips hue
ID: 24158E
Model: BSB002
Version: 01035934

6. Hue color lamp 1
Model: LCT001
Version: IrradiateHue

7. Hue color downlight 1
Model: LCT002
Version: 5.23.1.13452

Worm Results - Take Over Attack Testing

- Two boards to capture lamps:
    - One emitting the factory reset command every 3 seconds
    - One focused on association / takeover
        - Lights flashed SOS when successfully taken over



- Two real life attack scenarios:
    - Wardriving
        - Full attack from a car implemented 50m away
        - Reset works 150+m away
    - Warflying
        - Attack from drone
        - Reset works at approximately 350m away

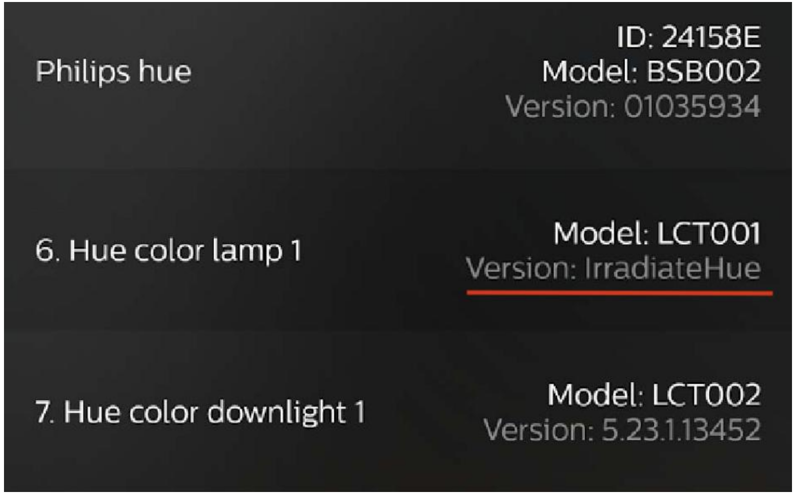
Worm Results - Firmware Update Attack Testing

- Took an arbitrary firmware image and converted it into a format accepted by the bulbs

- Hue Lamps used AES-CCM in encryption / verification of firmware binary
  - Utilization of DPA

- Exploited the fact lights of the same model use the same key
  - Figuring it out for one bulb allows for performing updates on any bulb of the same model

- Decrypted on update image, changed the version to IrradiateHue, and re-encrypted the file



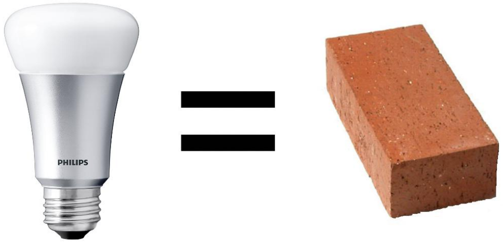
Philips hue
ID: 24158E
Model: BSB002
Version: 01035934

6. Hue color lamp 1
Model: LCT001
Version: IrradiateHue

7. Hue color downlight 1
Model: LCT002
Version: 5.23.1.13452

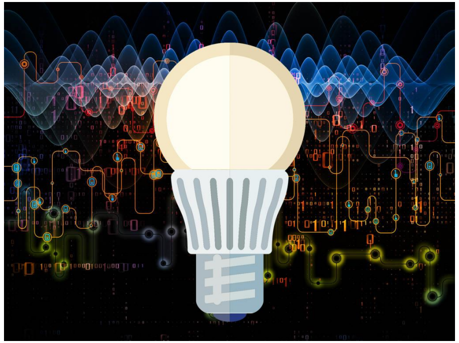
Worm Effects & Countermeasures

- **Different kinds of attack use cases:**
  - **Bricking**
    - Replaces existing firmware
    - Can pick what updates go through, if any
    - Permanent effects
      - (Unless reprogramed at PCB level)
  - **Wireless Jamming**
    - Devices can be used to continuously transmit waves on channels without checking if clear
    - Easily disrupt WiFi traffic in a given area



Worm Effects & Countermeasures Cont.d

- **Different kinds of attack use cases:**
  - **Data Exfiltration**
    - Infected bulb can create a covert channel
    - Pull data from messages sent from the bridge
  - **Data Infiltration**
    - Change data inside the network such as user readable data
  - **Epileptic Seizures**
    - Can flicker bulbs at rate which triggers seizures
    - Lack of IoT security can have human health repercussions



Worm Effects & Countermeasures Cont.d

- **Design vulnerabilities exploited:**
    - Using a single, symmetric encryption key shared across all the devices to protect a firmware update process
        - Remedy: Use a unique key per bulb or asymmetric cryptography for software verification
    - Hardware vulnerable to side-channel analysis
        - Remedy: In IoT devices, too expensive to solve this, spend efforts to make sure data leaked cannot affect entire system
    - Errors in protocols to prevent long range takeover attacks
        - Remedy: negative testing to ensure invalid cases do not break system

Conclusion

- Large scale problems can be created through the exploitation of weak security measures in IoT devices
  - If infected, only solution may be to replace all of the physical devices which could be tremendously costly
- When considering the tradeoffs between usability and security, don't sacrifice too much security just to make something easier to understand without understanding the ramifications
  - Be smart in security practice -> look to prior works
  - Eliminate the possibility / probability of chain reactions

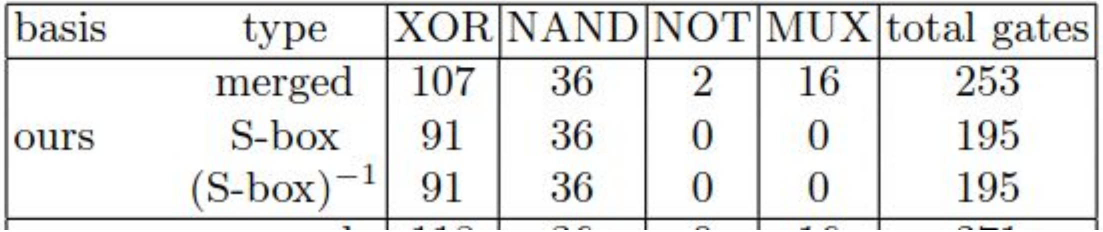
Questions?

XOR Dominated Circuits

XOR Gate has a higher Boolean Sensitivity which results in glitches causing higher power consumption.

“When XOR gates are implemented as complex static CMOS gates, the power consumption is much larger. Power consumption due to (charging and discharging) internal node capacitances in complex XOR gates is significant.”

- K. Roy, R. Drechsler and Y. Ye, "Power Consumption in XOR-Based Circuits," in Asia and South Pacific Design Automation Conference, Wanchai, Hong Kong, 1999 pp. 299
- (below) Canright, D. (2005). A Very Compact S-Box for AES. In: Rao, J.R., Sunar, B. (eds) Cryptographic Hardware and Embedded Systems – CHES 2005



basis	type	XOR	NAND	NOT	MUX	total gates
ours	merged	107	36	2	16	253
	S-box	91	36	0	0	195
	$(\text{S-box})^{-1}$	91	36	0	0	195

Hardened Microarchitectures

PARAM: A Microprocessor Hardened for Power Side-Channel Attack Resistance

Muhammad Arsath K F, Vinod Ganesan, Rahul Bodduna, and Chester Rebeiro
Department of Computer Science and Engineering
Indian Institute of Technology Madras, India
{muhammadarsath,vinodg,rahulb,chester}@cse.iitm.ac.in

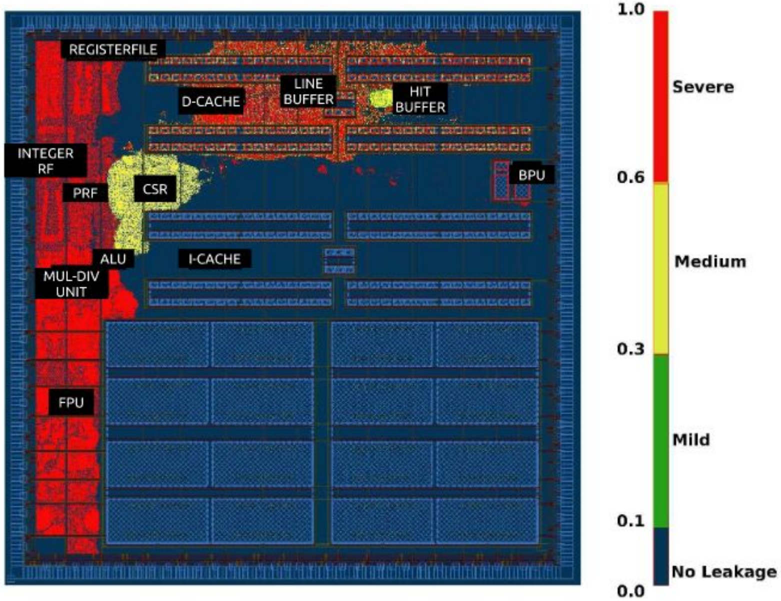
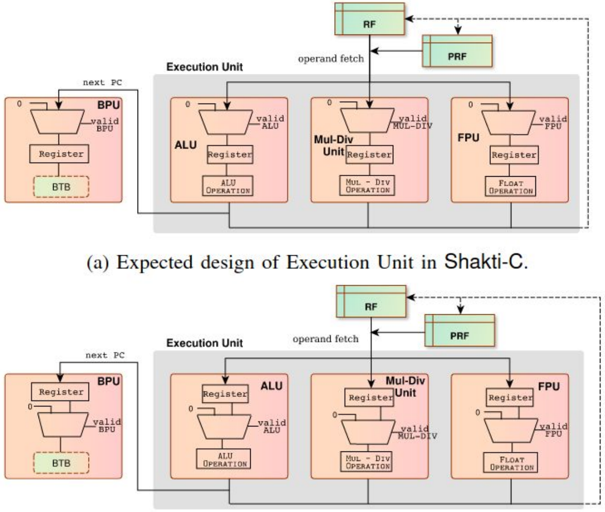


Fig. 3: Information leakage plot shows the floor plan for Shakti-C illustrating the modules with their side-channel leakage.



(a) Expected design of Execution Unit in Shakti-C.

(b) Design after Bluespec compilation. The change in the placement of register causes increased leakage.

Fig. 7: EDA tools perform translations keeping functionality intact. Some of these translations may increase side-channel vulnerability as we found in the Execute Unit of Shakti-C when compiled with Bluespec.

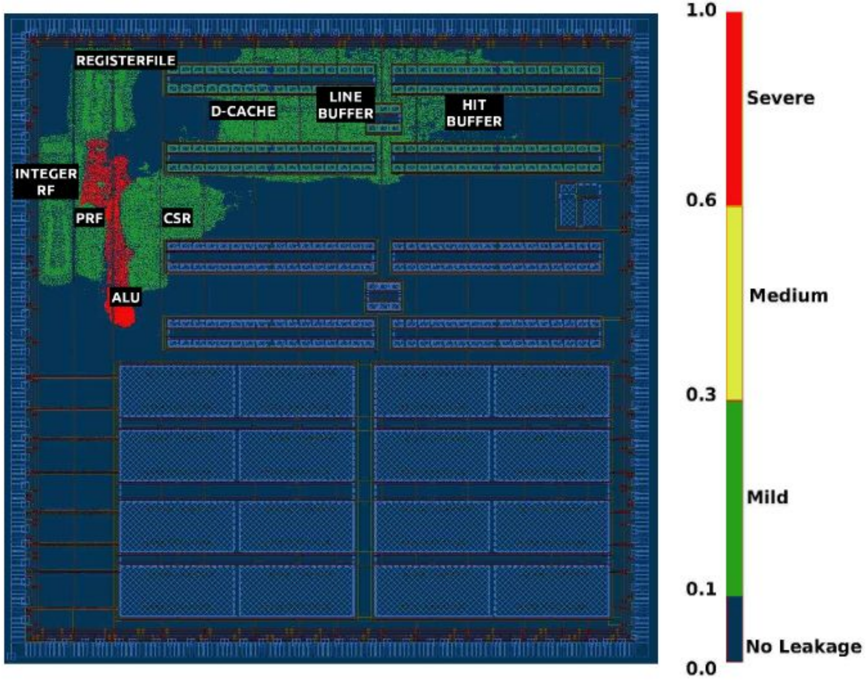


Fig. 11: Information leakage plot shows the floor plan for PARAM illustrating the modules with their SVF values in different colors.