Understanding Screaming Channels: From a Detailed Analysis to Improved Attacks

Giovanni Camurati*, Aurélien Francillon*, François-Xavier Standaert**

*EURECOM, **Université catholique de Louvain
Who am I?

Giovanni Camurati
Ph.D. Student at EURECOM, Sophia-Antipolis, France
@GioCamurati
https://giocamurati.github.io

Side Channels and Radios
What happens if radio transceivers are close to computing devices?

Computer Architectures, Electronics, Embedded Systems
Hardware Design, Firmware Rehosting,
Hack@DAC with NOPS
Why radios and computing devices?
Modern Connected Devices Have Radios

Mixed-signal architecture
CPU + Crypto + Radio
Same chip
Modern Connected Devices Have Radios

**Mixed-signal architecture**

CPU + Crypto + Radio

Same chip

**Benefits**

Low Power, Cheap, Small

Easy to integrate
Modern Connected Devices Have Radios

**Mixed-signal architecture**
CPU + Crypto + Radio
Same chip

**Benefits**
Low Power, Cheap, Small
Easy to integrate

**Examples**
BT, BLE, WiFi, GPS, etc
What can go wrong?
Screaming Channels [1], The Idea

Mixed-signal chip

64 MHz 2.4 GHz
Screaming Channels [1], The Idea

- Mixed-signal chip
- Strong noise source
- Digital Logic
- Memory

- Noise sensitive transmitter

64 MHz
2.4 GHz
Screaming Channels [1], The Idea

Mixed-signal chip

Strong noise source

Digital Logic

Memory

Radio

Noise sensitive transmitter

64 MHz

2.4 GHz
Screaming Channels [1], The Idea

- Mixed-signal chip
- Strong noise source
- Digital Logic
- Memory
- Radio
- Noise sensitive transmitter
- Easy propagation

64 MHz
2.4 GHz
Screaming Channels [1], The Idea

Mixed-signal chip

Conventional Side
Channel Leak

Strong noise source

Digital Logic

Memory

Radio

Easy propagation

Noise sensitive transmitter

64 MHz

2.4 GHz
Screaming Channels [1], The Idea

Mixed-signal chip

Conventional Side Channel Leak

Strong noise source

Easy propagation
Leak Propagation

Noise sensitive transmitter

- Mixed-signal chip
- Digital Logic
- Memory
- Radio

Frequency:
- 64 MHz
- 2.4 GHz
Screaming Channels [1], The Idea

- Conventional Side Channel Leak
  - Strong noise source
  - Mixed-signal chip
  - Digital Logic
  - Memory
  - Radio

- Easy propagation
- Leak Propagation

- Noise sensitive transmitter

- Leak Is Broadcast

- 64 MHz
- 2.4 GHz
Screaming Channels [1] in Action

Antenna + SDR RX

Cortex-M4 + BT TX
Screaming Channels [1] in Action

Antenna + SDR RX

Radio Off

Cortex-M4 + BT TX

Noise
Screaming Channels [1] in Action

Antenna + SDR RX

Cortex-M4 + BT TX

Radio Off  Radio TX

Noise  Packet

2m
Screaming Channels [1] in Action

Antenna + SDR RX

Radio Off

Radio TX

Cortex-M4 + BT TX

2m

Wait loop

Noise

Packet
Screaming Channels [1] in Action

Antenna + SDR RX

Cortex-M4 + BT TX

Radio Off  Radio TX  AES On

2m

Wait loop

Noise  Packet
Screaming Channels [1] in Action

Antenna + SDR RX

Cortex-M4 + BT TX

Radio Off  Radio TX  AES On

Wait loop

AES Starts

Noise  Packet
Screaming Channels [1] in Action

Antenna + SDR RX

Cortex-M4 + BT TX

Radio Off  Radio TX  AES On

Wait loop  AES Starts

Noise  Packet

Time domain

2m
A New Threat [1]

EM Leak, proximity

Radio Leak, e.g. 10m

Intended Transmission e.g. 1m
The "Screaming Channels" Leak Vector

Idea, Root Cause, First Attack
Intuition and root cause
10m in anechoic chamber
Countermeasures
The "Screaming Channels" Leak Vector

Idea, Root Cause, First Attack
Intuition and root cause
10m in anechoic chamber
Countermeasures

Camurati, Poeplau, Muench, Hayes, Francillon
The "Screaming Channels" Leak Vector

Idea, Root Cause, First Attack
Intuition and root cause
10m in anechoic chamber
Countermeasures

Camurati, Poeplau, Muench, Hayes, Francillon

Systematic Analysis
Data/leak coexistence
Distortion, profile reuse, etc.

Improved Attacks
Realistic environment up to 15m
Google Eddystone Beacons
The "Screaming Channels" Leak Vector

Idea, Root Cause, First Attack
Intuition and root cause
10m in anechoic chamber
Countermeasures

Systematic Analysis
Data/leak coexistence
Distortion, profile reuse, etc.

Improved Attacks
Realistic environment up to 15m
Google Eddystone Beacons

Camurati, Poeplau, Muench, Hayes, Francillon

TCHES 2020
Camurati, Francillon, Standaert
Some Other Interesting Cases

“LeakyNoise”
CPU to ADC side channel in mixed-signal chips
CHES2019 [14]

Second-Order Soft-TEMPEST
Soft-TEMPEST + (un)intentional cascaded effects
EMC Europe 2018 [15]
AP-RASC 2019 [16]
Let us answer some open questions about Screaming Channels
What is the difference with conventional leakages?
Intuitively

Coupling on chip

Radio channel (data + leakage)

Near-field probe
Intuitively

Coupling on chip  
Radio channel (data + leakage)

1. SNR?
2. Distortion?

Near-field probe
Intuitively

1. SNR?
2. Distortion?

3. SNR & Distortion
   - Distance & Setup
   - BLE Channel

4. Data/Leakage modulation

5. Discrete packets
6. Frequency hopping

Coupling on chip

Radio channel (data + leakage)

Near-field probe
Necessary Steps Before We Can Start

1. **Extract traces (in the specific case of our BLE device)**
   1. Data (GFSK) and leakage (AM) are orthogonal
   2. Trigger on a peculiar frequency
   3. Fix the channel (we will consider hopping later)
   4. Time diversity to deal with deep fade between packets
Necessary Steps Before We Can Start

1. Extract traces (in the specific case of our BLE device)
   1. Data (GFSK) and leakage (AM) are orthogonal
   2. Trigger on a peculiar frequency
   3. Fix the channel (we will consider hopping later)
   4. Time diversity to deal with deep fade between packets

2. Normalize
   1. Z-score normalization inspired by [3,4,5,6]
   2. Per-trace normalization removes the effect of the channel!
Necessary Steps Before We Can Start

1. **Extract traces (in the specific case of our BLE device)**
   1. Data (GFSK) and leakage (AM) are orthogonal
   2. Trigger on a peculiar frequency
   3. Fix the channel (we will consider hopping later)
   4. Time diversity to deal with deep fade between packets

2. **Normalize**
   1. Z-score normalization inspired by [3,4,5,6]
   2. Per-trace normalization removes the effect of the channel!

   \[ y(t) = Gx(t) \]

   \[ y' = \frac{y - \text{avg}(y)}{\text{std}(y)} = \frac{Gx - G\text{avg}(x)}{G\text{std}(x)} = x' \]
Understanding the Leakage

**Leakage variable** $y = \text{SBox}(p \text{ xor } k)$

**Leakage model** $m(y) = \text{HW}[y]$

**Leakage** $l(y)$
Understanding the Leakage

**Leakage variable** \( y = \text{SBox}(p \oplus k) \)

**Leakage model** \( m(y) = \text{HW}[y] \text{ model}(y) \)  Estimate (nonlinear) leakage model for each \( y \), using the profiling set

**Leakage** \( l(y) \)
Understanding the Leakage

Leakage variable $y = \text{SBox}(p \text{ xor } k)$

Leakage model $m(y) = \text{HW}[y] \text{ model}(y)$ Estimate (nonlinear) leakage model for each $y$, using the profiling set

Leakage $l(y)$ Estimate the linear correlation between $m(y)$ and $l(y)$ on test set
Understanding the Leakage

Leakage variable $y = \text{SBox}(p \oplus k)$

Leakage model $m(y) = \text{HW}[y] \text{ model}(y)$ Estimate (nonlinear) leakage model for each $y$, using the profiling set

Leakage $l(y)$ Estimate the linear correlation between $m(y)$ and $l(y)$ on test set

This is the r-test [7]
Understanding the Leakage

(a) $\rho$-test with $p \oplus k$ (green) and $HW(Sbox(p \oplus k))$
(b) Screaming 10 cm: $\rho$-test with $p \oplus k$ (green) and $HW(Sbox(p \oplus k))$ (red)
Understanding the Leakage

Leakage variable $y$

Leakage model $m(y)$

$\text{Leakage} = SBox(p \oplus k) = HW[y] = \text{model}(y)$

Estimate (nonlinear) leakage model for each $y$, using the profiling set

Estimate the linear correlation between $m(y)$ and $l(y)$ on test set

This is the $r_{\text{test}}$

Results for Screaming vs. Conventional

• Less POIs
• Slightly lower but still high correlation
• HW is not a good model

SNR is comparable
But the leakage is distorted
Understanding the Leakage

**Leakage variable** $y = \text{SBox}(p \text{ xor } k)$

**Leakage model** $m(y) = \text{HW}[y]$

**Leakage** $l(y)$
Understanding the Leakage

**Leakage variable** $y = \text{SBox}(p \oplus k)$

**Leakage model** $m(y) = \text{HW}[y]$ Linear combination of the bits of $y$

**Leakage** $l(y)$ Estimate a linear model of the bits of $y$ using linear regression [7]
(a) Conventional
(b) Screaming at 10 cm
Understanding the Leakage

Leakage variable $y$

Leakage model $m(y)$

$\text{Leakage} \quad l(y) \quad = \quad S\text{Box}(p \text{xor} k) \quad = \quad \text{HW}[y]$

Linear combination of the bits of $y$

Estimate a linear model of the bits of $y$ using linear regression [7]

Results for Screaming vs. Conventional

- Confirm leakage from Sbox output
- Linear model is good for conventional traces
- Bad for screaming traces The leakage model is nonlinear

(a) Conventional

(b) Screaming at 10 cm
Understanding the Leakage

**Leakage variable** \( y \)

**Leakage model** \( m(y) \)

**Leakage** \( l(y) \)

Templates [9] can capture a second order relation between \( m(y) \) and \( l(y) \).
Understanding the Leakage

Leakage variable $y$

Leakage model $m(y)$

Leakage $l(y)$

Templates [9] can capture a second order relation between $m(y)$ and $l(y)$

Results for Screaming vs. Conventional

- Templates attacks are not considerably better than profiled correlation attacks

  First-order leakage (for our sample size)
Conclusion

1. Comparable SNR, distorted leakage model
2. Nonlinear leakage model
3. First order leakage

Profiled Correlation Attacks
Can we reuse the profiles?

2/4
How To Compare Profiles

#Traces for key recovery [10]
Given profile P and attack traces A

P1, A1       P2, A2

Distance & Device
How To Compare Profiles

#Traces for key recovery [10]
Given profile P and attack traces A

\[ N_{11} \propto r^{-2}(P_1, A_1) \quad \text{N22} \propto r^{-2}(P_2, A_2) \]
How To Compare Profiles

#Traces for key recovery [10]
Given profile P and attack traces A

\[ N_{11} \propto r^{-2}(P_1, A_1) \quad N_{22} \propto r^{-2}(P_2, A_2) \]
How To Compare Profiles

#Traces for key recovery [10]
Given profile P and attack traces A

Reuse P1

\[ N_{11} \propto r^{-2}(P_1, A_1) \]
\[ N_{22} \propto r^{-2}(P_2, A_2) \]
\[ N_{12} \propto r^{-2}(P_1, A_2) \]
\[ r(P_1, A_2) = r(P_2, A_2)r(P_1, P_2) \]

The higher the better
Distance

- Quadratic power loss, but we can amplify
- Normalization cancels the multiplicative channel gain
- No extra distortion (different from conventional [11])
Distance, Setup, Channel Frequency, Instance, Time

Distance
- Quadratic power loss, but we can amplify
- Normalization cancels the multiplicative channel gain
- No extra distortion (different from conventional [11])

Environment (noise) and setup
- Bigger role than distance, but we can improve the setup
- Some connections are better
Distance
- Quadratic power loss, but we can amplify
- Normalization cancels the multiplicative channel gain
- No extra distortion (different from conventional [11])

Environment (noise) and setup
- Bigger role than distance, but we can improve the setup
- Some connections are better

Device instance
- No significant impact, per-trace normalization helps
Distance
• Quadratic power loss, but we can amplify
• Normalization cancels the multiplicative channel gain
• No extra distortion (different from conventional [11])

Environment (noise) and setup
• Bigger role than distance, but we can improve the setup
• Some connections are better

Device instance
• No significant impact, per-trace normalization helps

Big Advantage
• Profile in good conditions, attack another instance in harsh conditions
Example: Distance

<table>
<thead>
<tr>
<th>d (m)</th>
<th>environment</th>
<th>antenna</th>
<th>( \hat{r}(P_i, P_2) ), -log10(p)</th>
<th>max ( \rho, r_z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_2 )</td>
<td>0.10</td>
<td>home</td>
<td>standard</td>
<td>1.00, inf</td>
</tr>
<tr>
<td>( P_3 )</td>
<td>0.20</td>
<td>home</td>
<td>standard</td>
<td>0.96, 142.77</td>
</tr>
<tr>
<td>( P_4 )</td>
<td>1.00</td>
<td>office</td>
<td>directional</td>
<td>0.40, 10.32</td>
</tr>
<tr>
<td>( P_5 )</td>
<td>5.00</td>
<td>anechoic</td>
<td>directional</td>
<td>0.96, 139.51</td>
</tr>
<tr>
<td>( P_6 )</td>
<td>10.00</td>
<td>anechoic</td>
<td>directional</td>
<td>0.92, 107.80</td>
</tr>
</tbody>
</table>

High correlation between profiles at each distance
Can we attack more challenging targets?

3/4
Attacks with obstacles and spatial diversity

Spatial Diversity
Different paths
Uncorrelated noise
Combine with Maximal Ratio

Attack
55cm in home environment
37k x 500 profiling traces
1990 x 500 attack traces
Rank $2^{26}$
Attacks in an office environment

Simple Profiling
Connection via cable
(10k x 500 traces)

Complex Attack
Different instance and time
10m (1.5k x 1000 traces, 2^28)
15m (5k x 1000 traces, 2^23, hard)
Attacks in an office environment

Simple Profiling
Connection via cable
(10k x 500 traces)

Complex Attack
Different instance and time
10m (1.5k x 1000 traces, 2^28)
15m (5k x 1000 traces, 2^23, hard)

Setup tuning becomes critical
Attacks in an office environment

Simple Profiling
Connection via cable
(10k x 500 traces)

Complex Attack
Different instance and time
10m (1.5k x 1000 traces, $2^{28}$)
15m (5k x 1000 traces, $2^{23}$, hard)

Setup tuning becomes critical

34m (2k x 1000 traces, t-test only)
60m (extraction only)
What about the hardware AES block?

**Simple Setup**
- 10cm in office
- USRP N210
- 350k x 100 traces

**Leaks from Memory Transfers**
- Firmware `memcpy` of p,c,k
- Hardware DMA of p,c,k
- No leak detected inside the AES

**Attacks**
- Only SPA attack are possible
- As of now we have not succeeded
Can we attack a real system?

4/4
What are Google Eddystone Beacons [12]?
What are Google Eddystone Beacons [12]?

UID identifier
URL e.g., www.museumshop.com
(e)TML (encrypted) telemetry
EID ephemeral id
What are Google Eddystone Beacons [12]?

UID identifier
URL e.g., www.museumshop.com
(e)TML (encrypted) telemetry
EID ephemeral id

Physical Web, Proximity Marketing, ...
Really used, though less popular now
What are Google Eddystone Beacons [12]?

**UID identifier**
**URL** e.g., www.museumshop.com
**(e)TML** (encrypted) telemetry
**EID** ephemeral id

**Physical Web, Proximity Marketing, ...**
Really used, though less popular now

**Configuration**
Authentication at GATT layer
Preshared key
AES128
What are Google Eddystone Beacons [12]?

- **UID** identifier
- **URL** e.g., www.museumshop.com
- **(e)TML** (encrypted) telemetry
- **EID** ephemeral id

**Configuration**
- Authentication at GATT layer
- Preshared key
- AES128

**Security & Privacy**
- Considered during design of the protocol

**Physical Web, Proximity Marketing, ...**
- Really used, though less popular now
Triggering AES encryptions with known plaintext

Beacon

Pre-shared key K

Owner/Attacker

P = Random()

Read Unlock Characteristic

CB = AES128(P,K)

Write Unlock Characteristic

CO = AES128(P,K)

Unlocked = (CB == CO)
Reducing the problem of frequency hopping

Frequency Hopping
A form of spread spectrum
Channel changes randomly
Hard to follow (sequence, speed, bandwidth)
Reducing the problem of frequency hopping

Frequency Hopping
A form of spread spectrum
Channel changes randomly
Hard to follow (sequence, speed, bandwidth)

Channel Map
E.g., `hcitool cmd 0x08 0x0014 0x000000000003`
The attacker can block up to 35 channels
The complete attack

Threat Model
Beacon with no physical access
• Not protected from EM/Power side channels
• Always connectable
The complete attack

Threat Model
Beacon with no physical access
- Not protected from EM/Power side channels
- Always connectable

Realistic Demo
Unmodified Nordic SDK demo [13]
- Optimized code (O3)
- Hopping Enabled (reduced with channel map)
- TinyAES software (hardware in later versions)
The complete attack

**Threat Model**
Beacon with no physical access
- Not protected from EM/Power side channels
- Always connectable

**Realistic Demo**
Unmodified Nordic SDK demo [13]
- Optimized code (O3)
- Hopping Enabled (reduced with channel map)
- TinyAES software (hardware in later versions)

**Proof-of-Concept Attack (connection via cable on PCA10040)**
70k x 1 profiling traces, 33k x 1 attack traces, rank $2^{30}$
Countermeasures?
Countermeasures

Resource constraint devices:
Cost, power, time to market, etc.
Countermeasures

Resource constraint devices:
Cost, power, time to market, etc.

Classic HW/SW:
Masking, noise, key refresh, limit attempts, use hardware block, ...
Countermeasures

Resource constraint devices:
Cost, power, time to market, etc.

Classic HW/SW:
Masking, noise, key refresh, limit attempts, use hardware block, ...

Specific (SW):
Radio off during sensitive computations
Force use of HW encryption (for now)
Countermeasures

Resource constraint devices:
Cost, power, time to market, etc.

Classic HW/SW:
Masking, noise, key refresh, limit attempts, use hardware block, ...

Specific (SW):
Radio off during sensitive computations
Force use of HW encryption (for now)

Specific (HW):
Consider impact of coupling on security during design and test
Conclusion
Conclusion

**General Problem:** Radios and Side Channels

**New threat point:** Digital activity visible from a large distance
Conclusion

General Problem: Radios and Side Channels
New threat point: Digital activity visible from a large distance

Peculiar: Not a conventional side channel vector
Easier: Amplified leak, large distance, simple and cheap setup
Harder: Distortion, channel noise, data/leak coexistence
Conclusion

**General Problem:** Radios and Side Channels

**New threat point:** Digital activity visible from a large distance

**Peculiar:** Not a conventional side channel vector

**Easier:** Amplified leak, large distance, simple and cheap setup

**Harder:** Distortion, channel noise, data/leak coexistence

**Threat:** More and more realistic attacks

**Potential threat:** More devices or new devices are vulnerable

**Countermeasures:** Clever, specific countermeasures
Conclusion

**General Problem:** Radios and Side Channels

**New threat point:** Digital activity visible from a large distance

**Peculiar:** Not a conventional side channel vector

**Easier:** Amplified leak, large distance, simple and cheap setup

**Harder:** Distortion, channel noise, data/leak coexistence

**Threat:** More and more realistic attacks

**Potential threat:** More devices or new devices are vulnerable

**Countermeasures:** Clever, specific countermeasures

**WiFi?** Possible even if not orthogonal?

**Hardware AES?** Attack the memory transfers?
Open Source!

https://eurecom-s3.github.io/screaming_channels/
Code + Data + Instructions
Thank You!
Come to the live session for questions!

Or write me:

@GioCamurati
https://giocamurati.github.io
camurati@eurecom.fr
Acknowledgements

• The authors acknowledge the support of SeCiF project within the French-German Academy for the Industry of the future, as well as the support by the DAPCods/IOTics ANR 2016 project (ANR-16-CE25-0015).

• We would like to thank the FIT R2lab team from Inria, Sophia Antipolis, for their help in using the R2lab testbed.
References

Third-Party Images

• "nRF51822 - Bluetooth LE SoC : weekend die-shot" - CC-BY– Modified with annotations. Original by zeptobars https://zeptobars.com/en/read/nRF51822-Bluetooth-LE-SoC-Cortex-M0