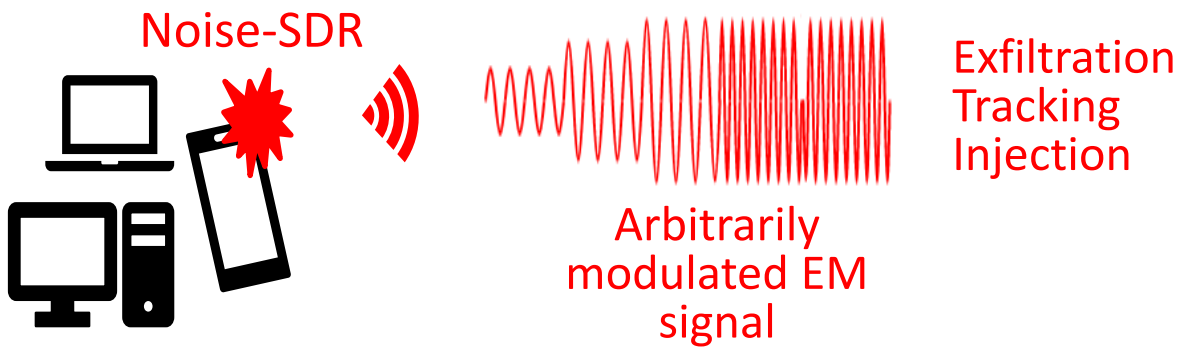
IEEE S&P 2022

Noise-SDR: Arbitrary Modulation of Electromagnetic Noise from Unprivileged Software and Its Impact on Emission Security

https://github.com/eurecom-s3/noise-sdr

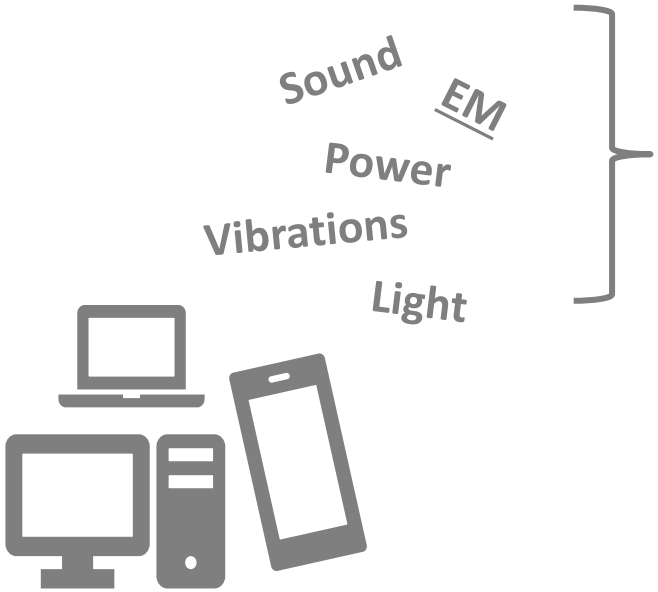
Giovanni Camurati
EURECOM (now ETH Zurich)
giovanni.camurati@eurecom.fr
@giocamurati

Aurélien Francillon
EURECOM
aurélien.francillon@eurecom.fr
@aurelsec



EURECOM
Sophia Antipolis

Context: Emission Security



Compromising emanations
Related to digital activity

R. J. Anderson, “Security Engineering - a Guide to Building Dependable Distributed Systems” (2. Ed.) (Wiley, 2008).

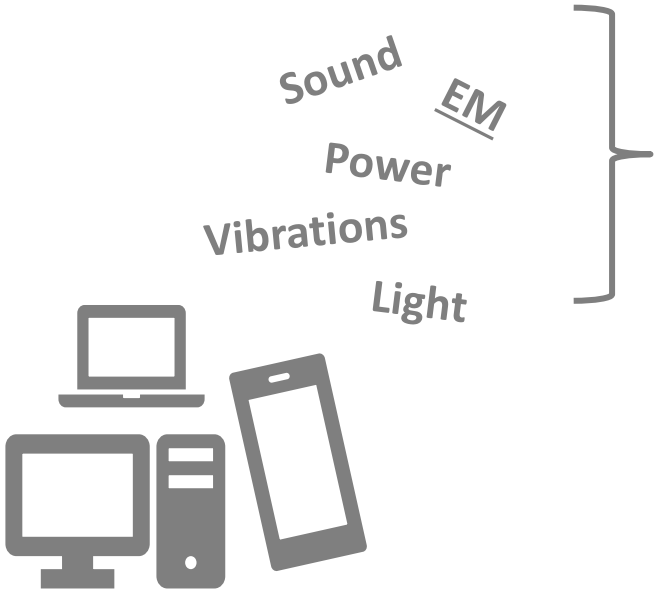
D. Agrawal et al., “The EM Side-Channel(s),” in CHES 2002.

“TEMPEST: A Signal Problem” (NSA, 1972).

W. van Eck, “Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?,” Comput. Secur. 4, no. 4 (1985).

EURECOM
Sophia Antipolis

Context: Emission Security



Compromising emanations
Related to digital activity

- **Key => Side Channel**

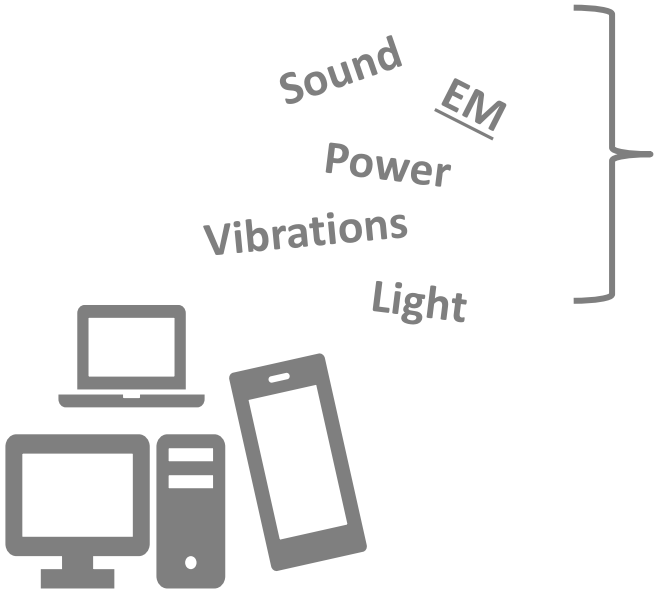
R. J. Anderson, “Security Engineering - a Guide to Building Dependable Distributed Systems” (2. Ed.) (Wiley, 2008).

D. Agrawal et al., “The EM Side-Channel(s),” in CHES 2002.

“TEMPEST: A Signal Problem” (NSA, 1972).

W. van Eck, “Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?,” Comput. Secur. 4, no. 4 (1985).

Context: Emission Security



Compromising emanations
Related to digital activity

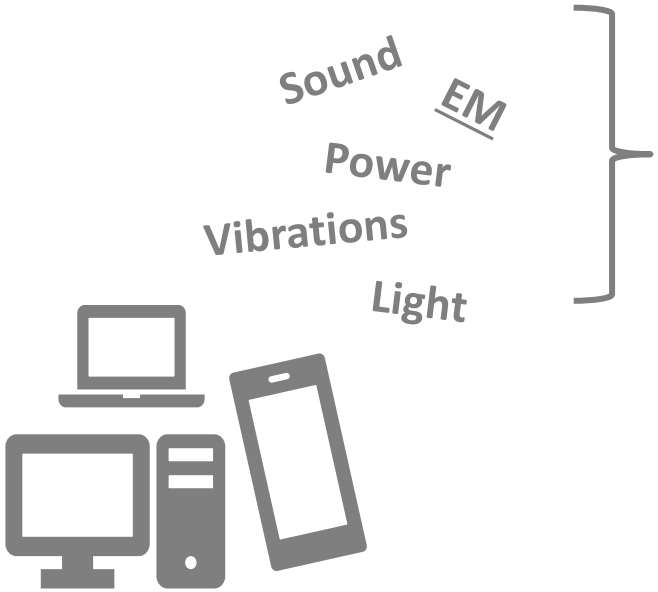
- **Key => Side Channel**
- **Plaintext => TEMPEST**

R. J. Anderson, “Security Engineering - a Guide to Building Dependable Distributed Systems” (2. Ed.) (Wiley, 2008).
D. Agrawal et al., “The EM Side-Channel(s),” in CHES 2002.

“TEMPEST: A Signal Problem” (NSA, 1972).
W. van Eck, “Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?,” Comput. Secur. 4, no. 4 (1985).

EURECOM
Sophia Antipolis

Context: Emission Security



Compromising emanations
Related to digital activity

- **Key => Side Channel**
- **Plaintext => TEMPEST**
- **Covert Channel => Soft-TEMPEST (active modulation to transmit data)**

R. J. Anderson, “Security Engineering - a Guide to Building Dependable Distributed Systems” (2. Ed.) (Wiley, 2008).
D. Agrawal et al., “The EM Side-Channel(s),” in CHES 2002.

“TEMPEST: A Signal Problem” (NSA, 1972).
W. van Eck, “Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?,” Comput. Secur. 4, no. 4 (1985).

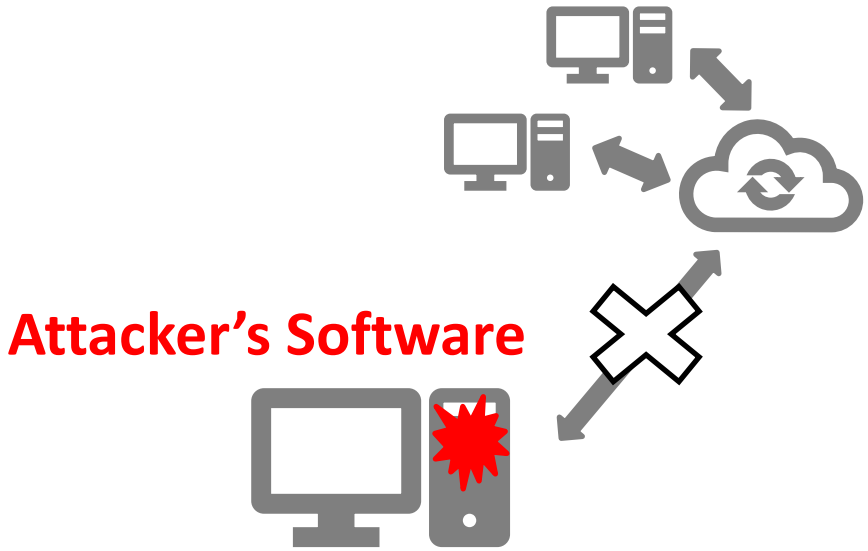
EURECOM
Sophia Antipolis

Context: Soft-TEMPEST

In theory…

Fully disconnected

Even an attacker able to execute code cannot exfiltrate data

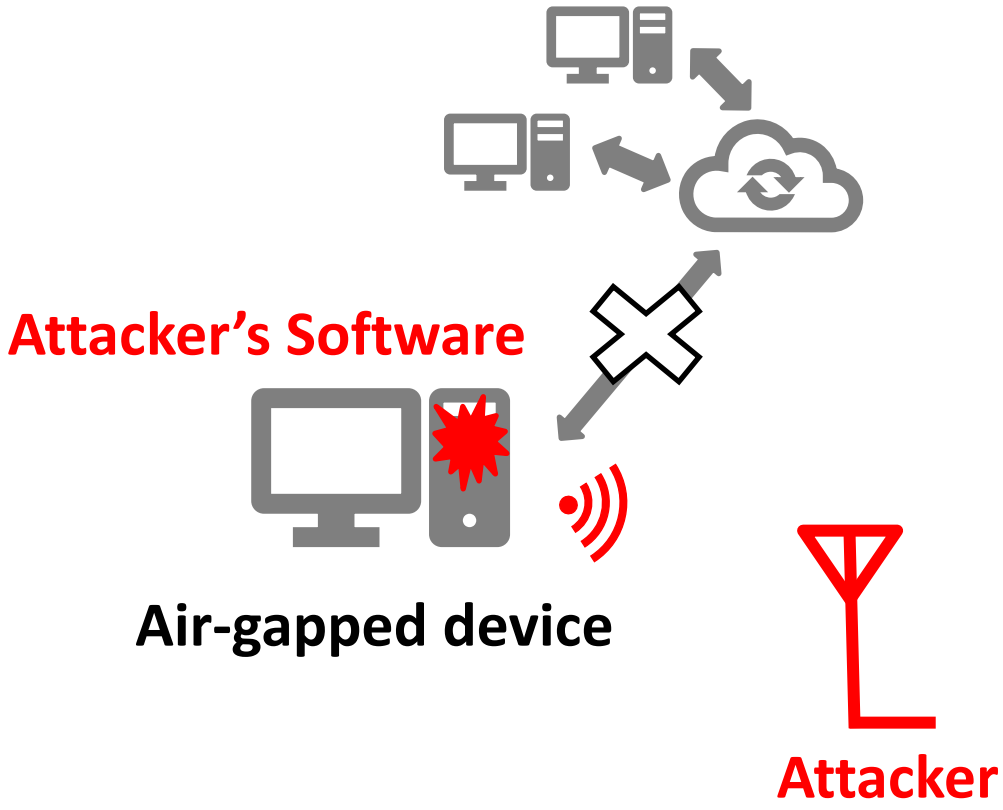


M. G. Kuhn and R. J. Anderson, "Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations," in Information Hiding (1998).

Context: Soft-TEMPEST

In theory…
Fully disconnected
Even an attacker able to execute code cannot exfiltrate data

Physical leakage…
Software execution triggers and modulates EM radiation



M. G. Kuhn and R. J. Anderson, "Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations," in Information Hiding (1998).

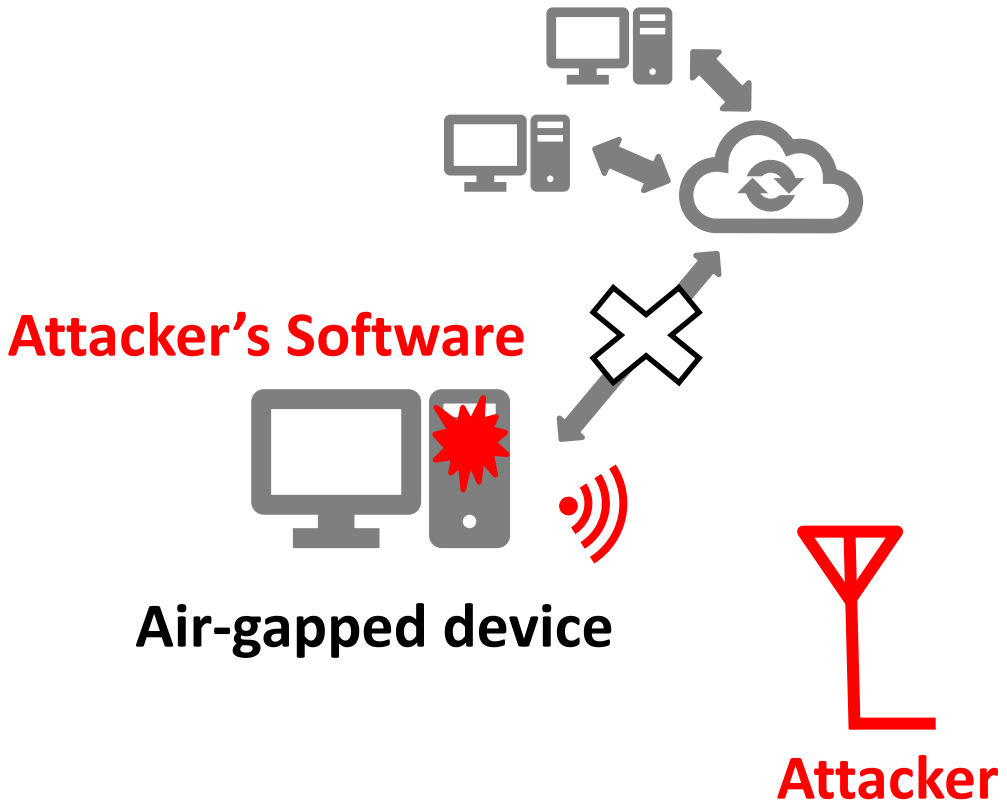
EURECOM
Sophia Antipolis

Context: Soft-TEMPEST

In theory…
Fully disconnected
Even an attacker able to execute code cannot exfiltrate data

Physical leakage…
Software execution triggers and modulates EM radiation

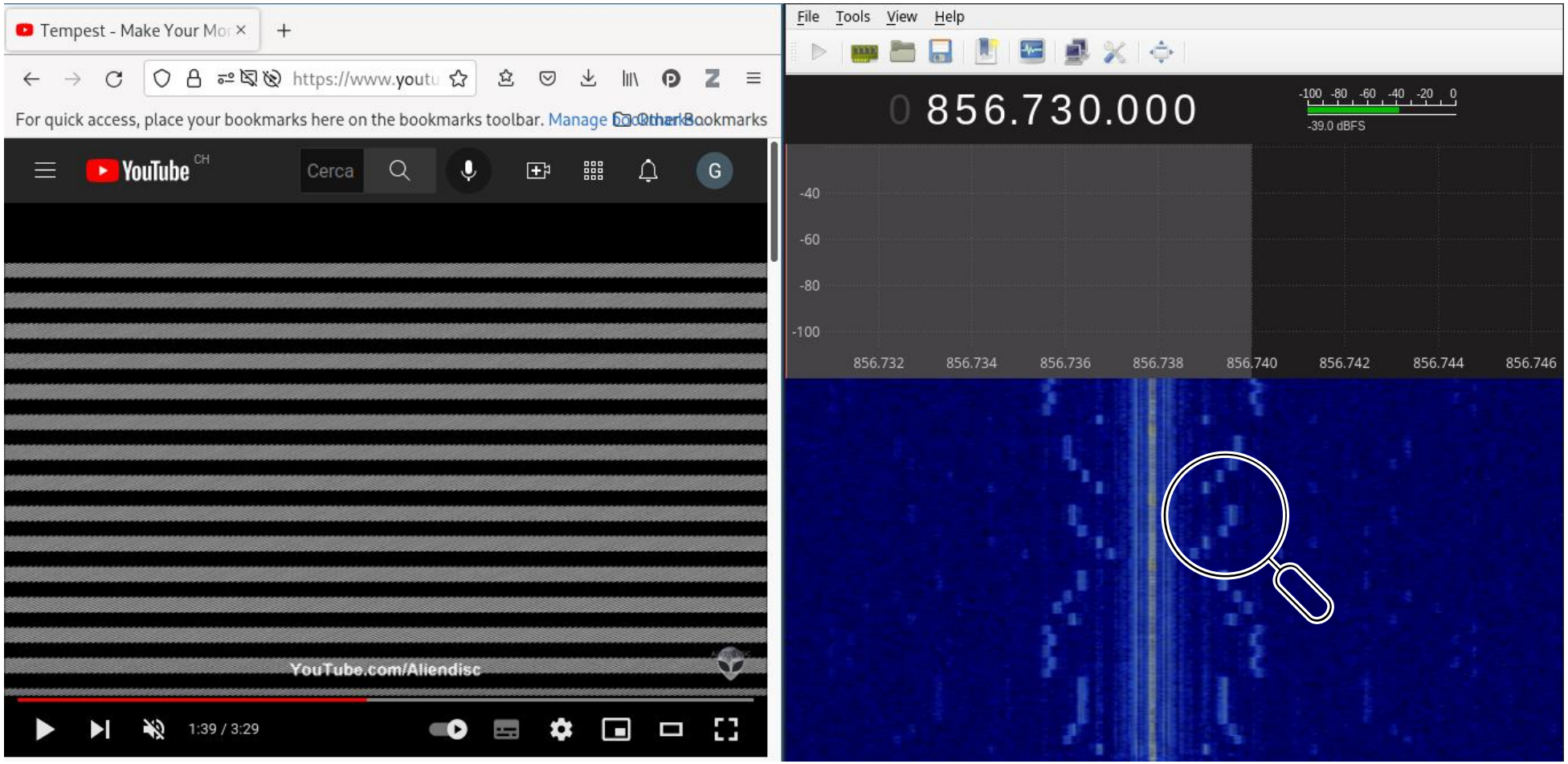
In practice…
Exfiltrate data via EM radiation
Communication is possible!



M. G. Kuhn and R. J. Anderson, "Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations," in Information Hiding (1998).

EURECOM
Sophia Antipolis

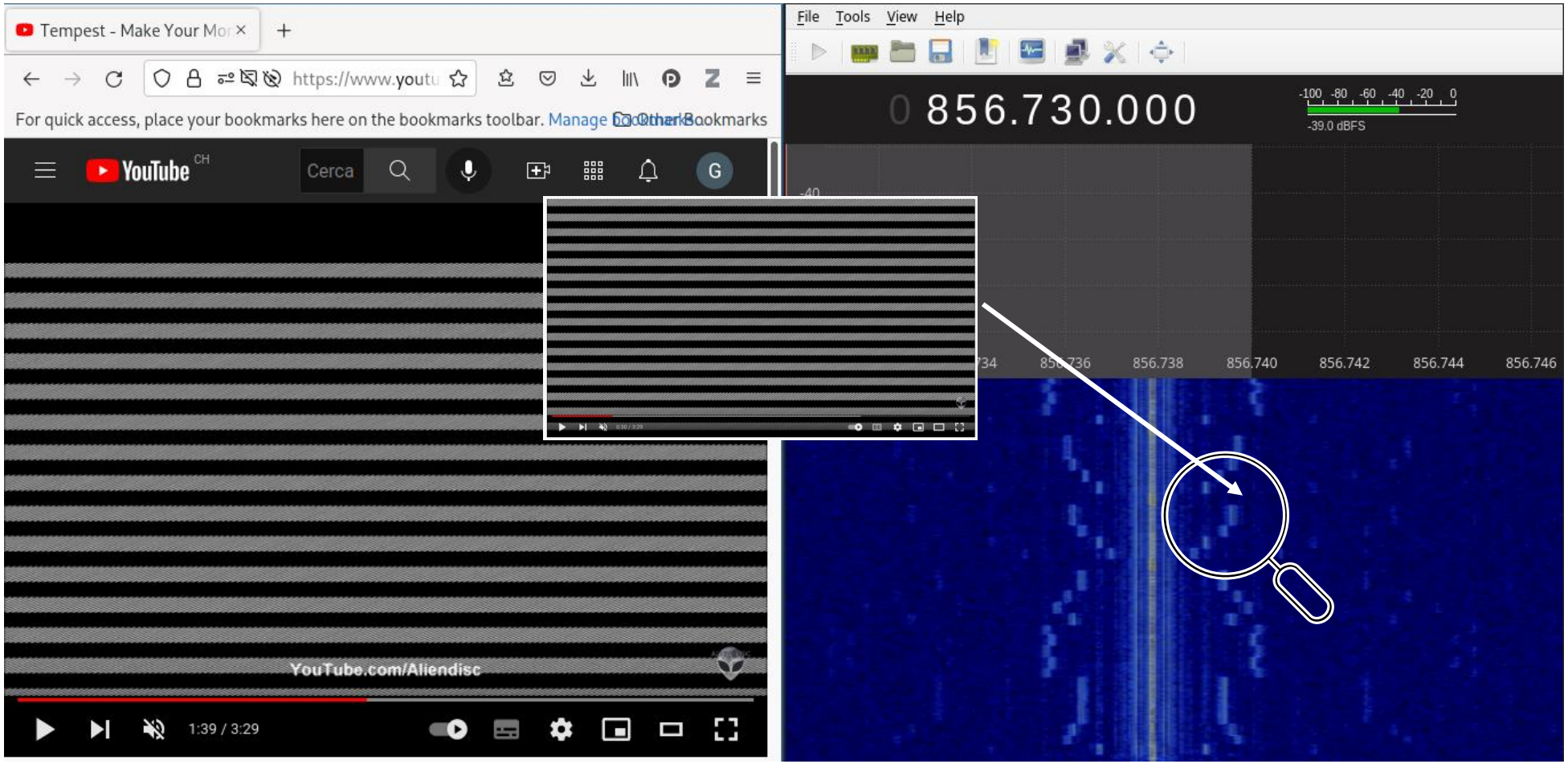
Background: the old classic "Tempest for Eliza" example



https://www.youtube.com/watch?v=DlVM9xqGKx8 **My laptop + HackRF radio receiver**

EURECOM Sophia Antipolis

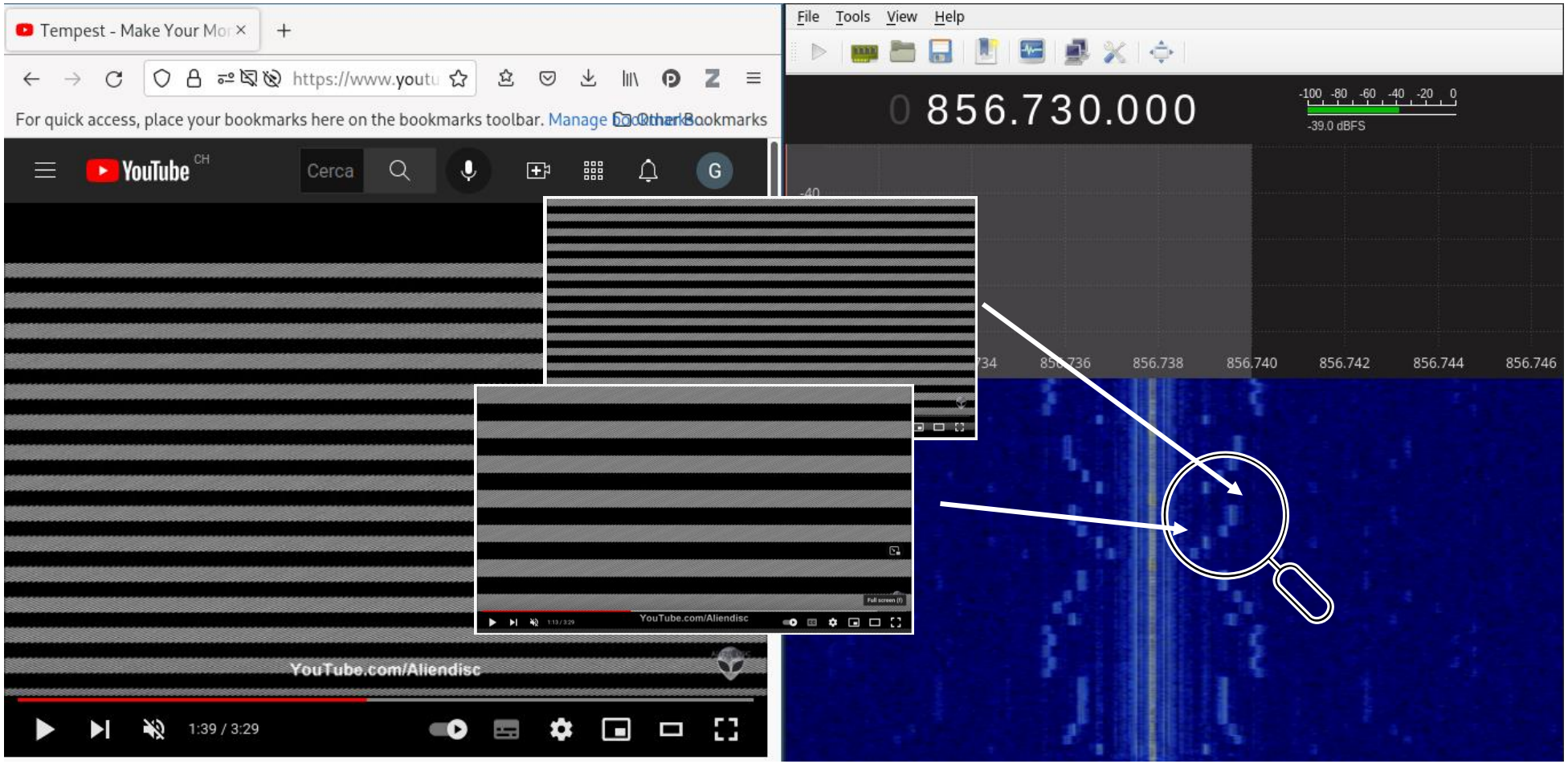
Background: the old classic "Tempest for Eliza" example



https://www.youtube.com/watch?v=DlVM9xqGKx8 **My laptop + HackRF radio receiver**

EURECOM Sophia Antipolis

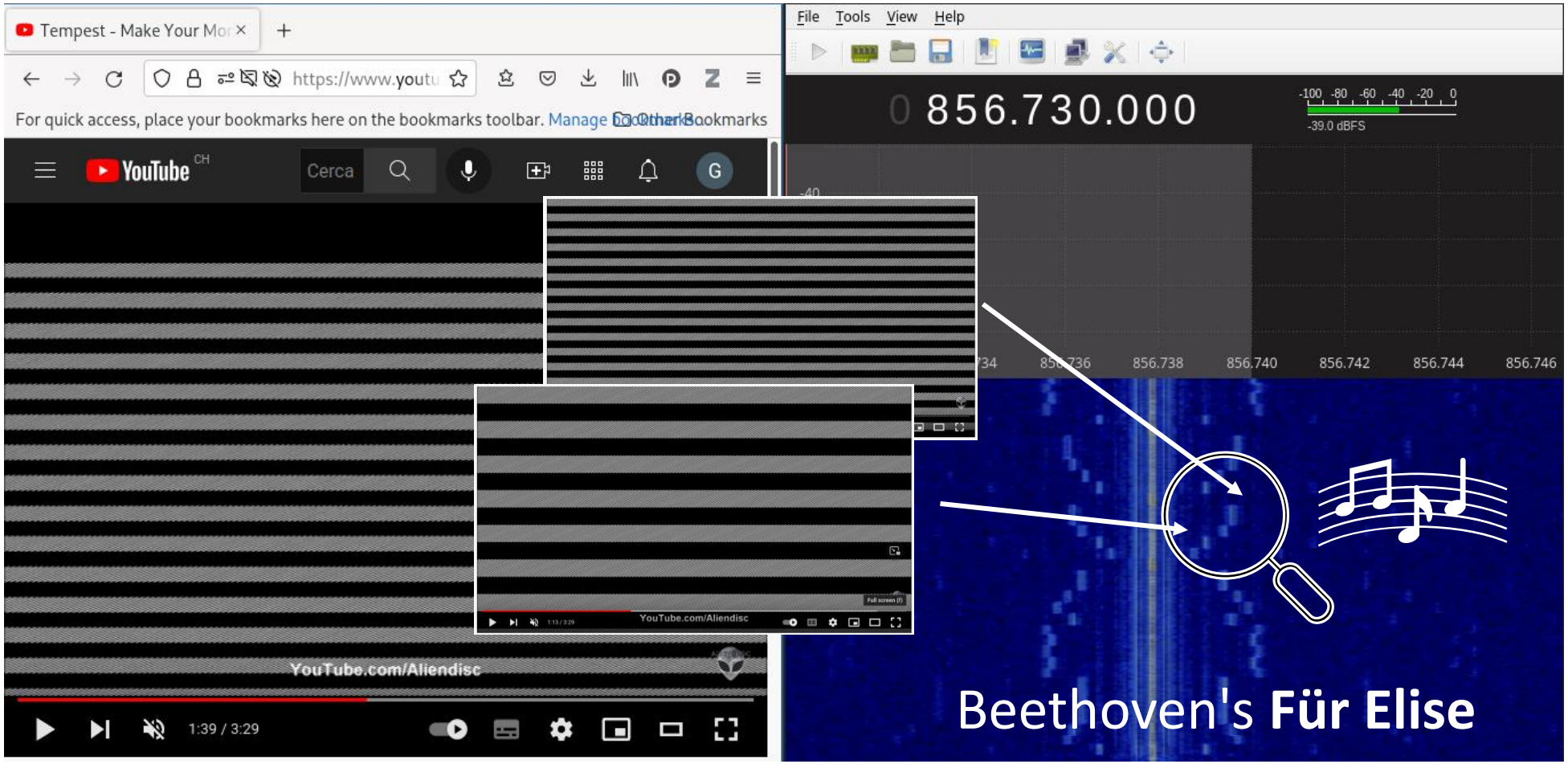
Background: the old classic "Tempest for Eliza" example



https://www.youtube.com/watch?v=DlVM9xqGKx8 **My laptop + HackRF radio receiver**

EURECOM Sophia Antipolis

Background: the old classic "Tempest for Eliza" example



https://www.youtube.com/watch?v=DlVM9xqGKx8 **My laptop + HackRF radio receiver**

EURECOM Sophia Antipolis

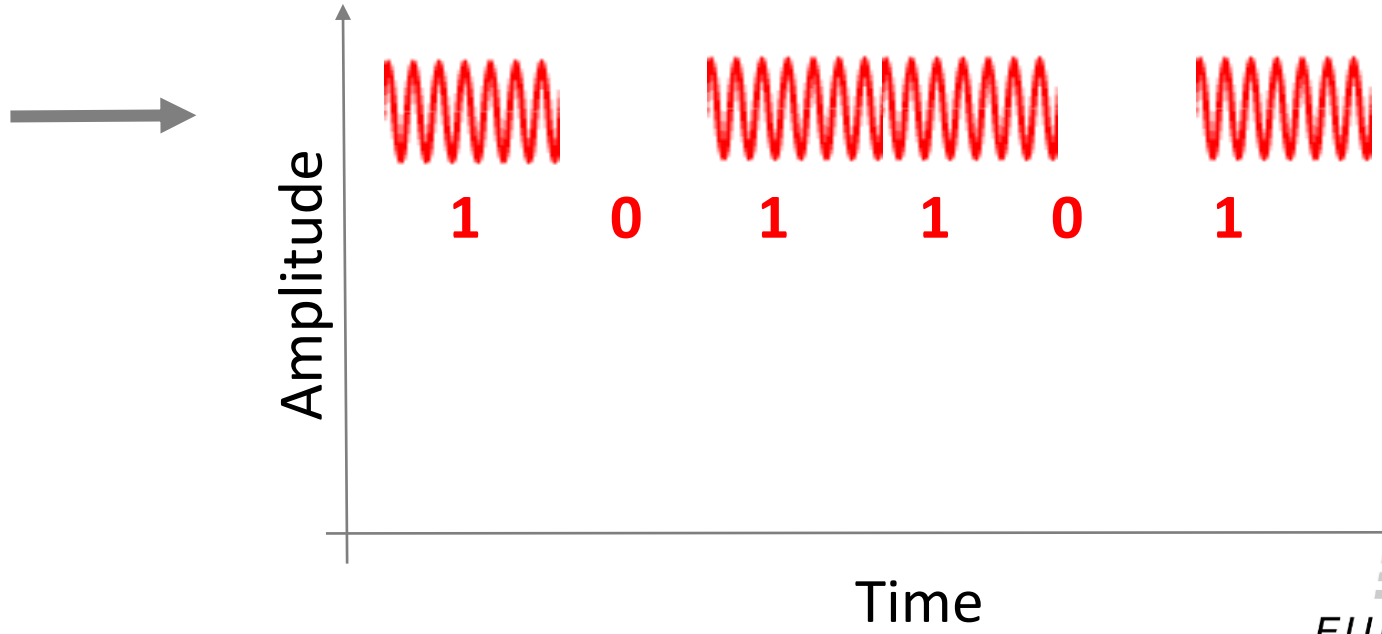
Background: modulation 101

Carrier
Sinusoidal wave at radio frequency

Background: modulation 101

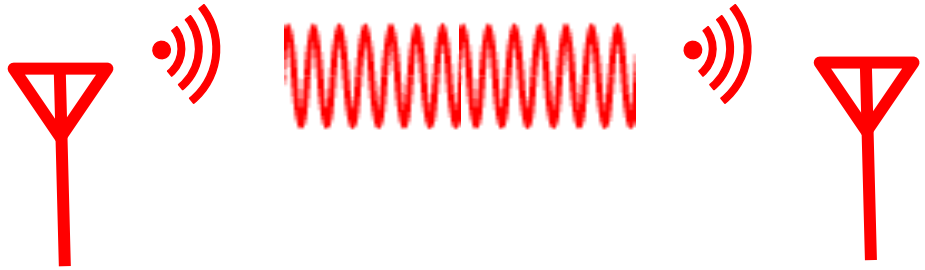
Carrier
Sinusoidal wave at radio frequency

OOK
On-Off Keying



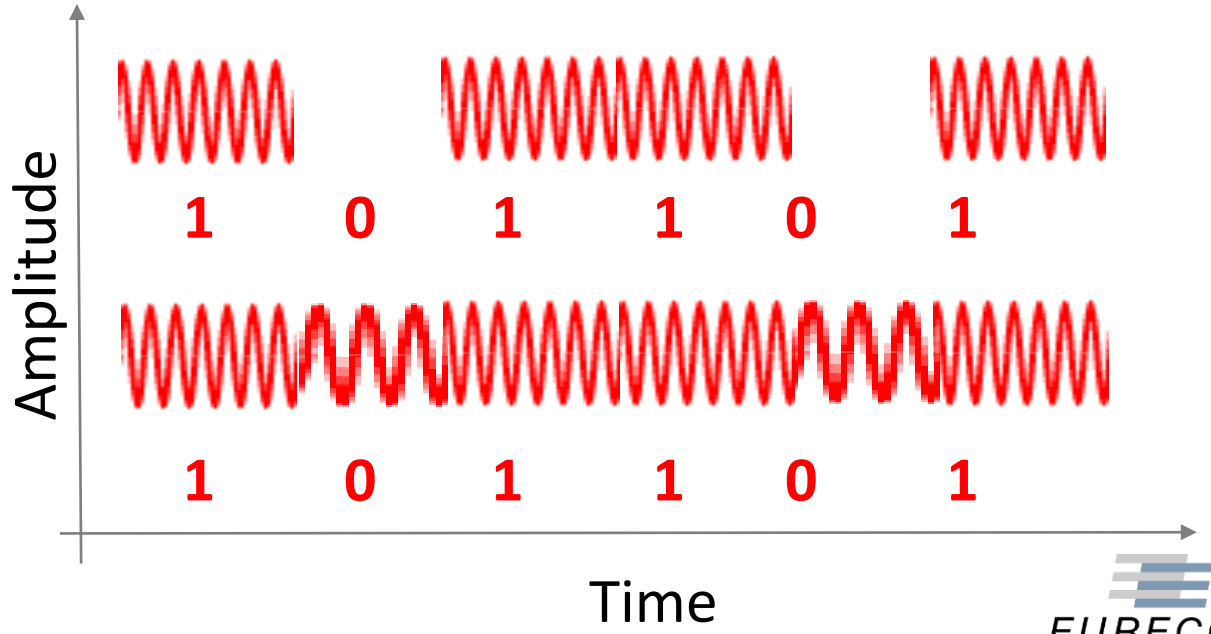
Background: modulation 101

Carrier
Sinusoidal wave at radio frequency



OOK
On-Off Keying

FSK
Frequency-Shift Keying



Background: general primitive in related work

```
start = now()
while( now() – start < T/2 )
        doSomething()
while( now() – start < T )
        doNothing()
```

*M. Guri et al., “GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies,” in USENIX Security 2015.

*Z. Zhan, Z. Zhang, and X. Koutsoukos, “BitJabber: The World’s Fastest Electromagnetic Covert Channel,” in IEEE ITC 2010

**C. Shen et al., “When LoRa Meets EMR: Electromagnetic Covert Channels Can Be Super Resilient”, IEEE S&P 2021

**W. Entriken, System Bus Radio, 2013, https://github.com/fulldecent/system-bus-radio.

Background: general primitive in related work

```
start = now()
while( now() – start < T/2 )
        doSomething()
while( now() – start < T )
        doNothing()
```

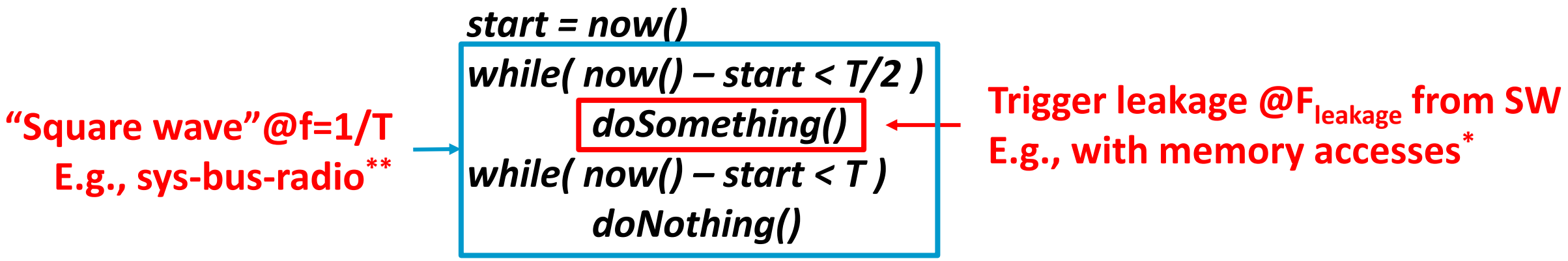
Trigger leakage @$F_{leakage}$ from SW
E.g., with memory accesses*

*M. Guri et al., “GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies,” in USENIX Security 2015.

*Z. Zhan, Z. Zhang, and X. Koutsoukos, “BitJabber: The World’s Fastest Electromagnetic Covert Channel,” in IEEE ITC 2010

**C. Shen et al., “When LoRa Meets EMR: Electromagnetic Covert Channels Can Be Super Resilient”, IEEE S&P 2021
**W. Entriken, System Bus Radio, 2013, https://github.com/fulldecent/system-bus-radio.

Background: general primitive in related work



*M. Guri et al., "GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies," in USENIX Security 2015.

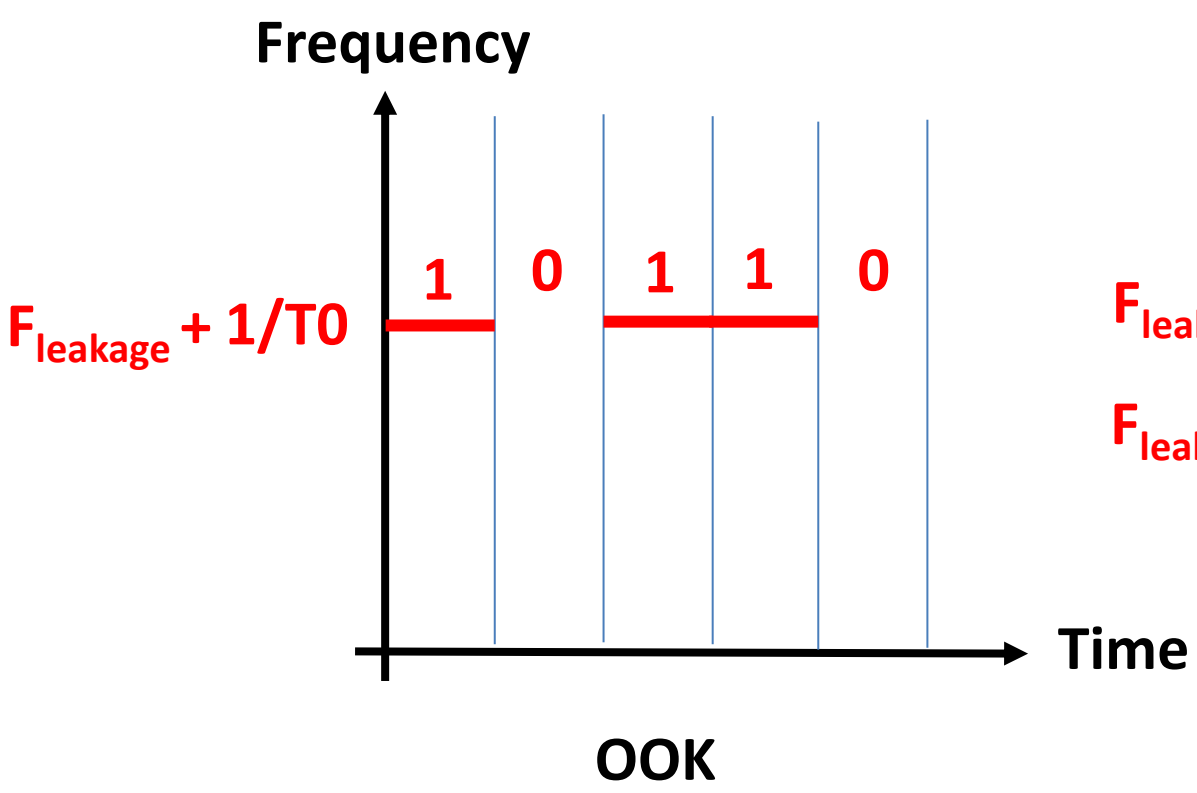
*Z. Zhan, Z. Zhang, and X. Koutsoukos, "BitJabber: The World's Fastest Electromagnetic Covert Channel," in IEEE ITC 2010

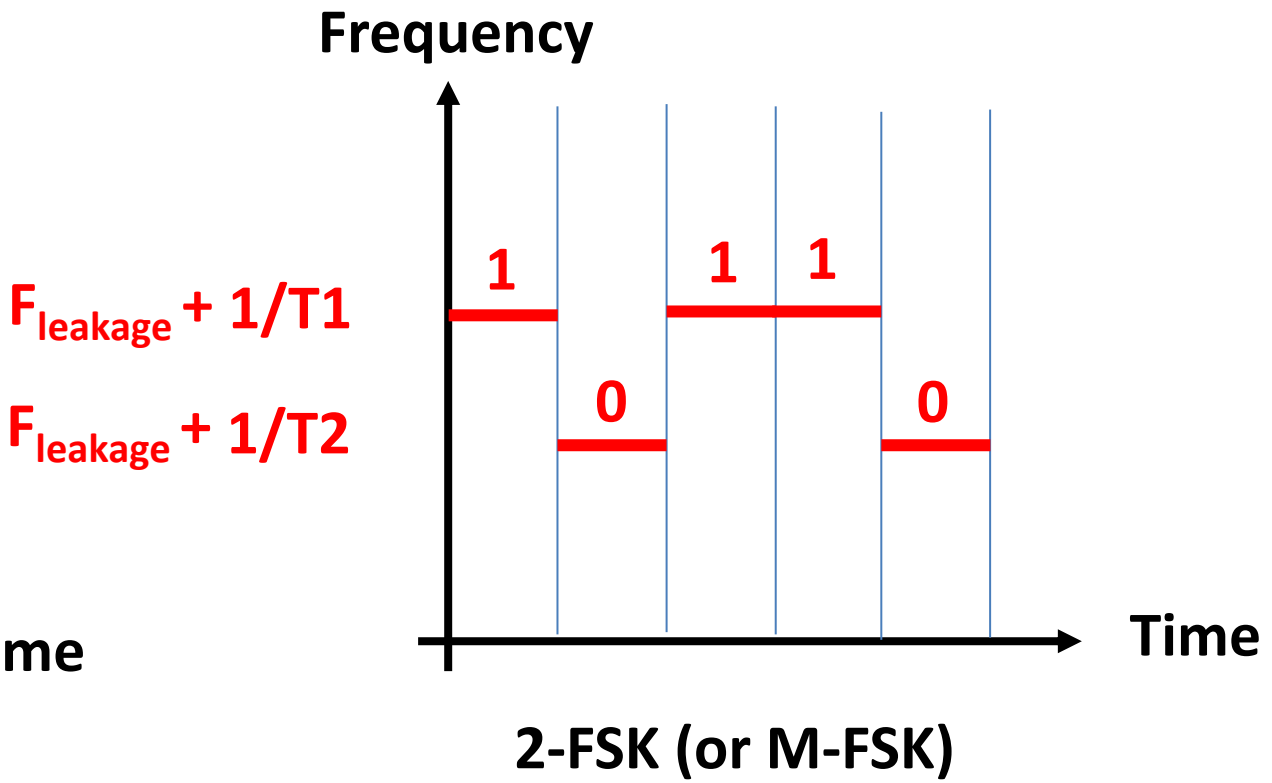
**C. Shen et al., "When LoRa Meets EMR: Electromagnetic Covert Channels Can Be Super Resilient", IEEE S&P 2021

**W. Entriken, System Bus Radio, 2013, https://github.com/fulldecent/system-bus-radio.

EURECOM
Sophia Antipolis

Background: general primitive in related work





EURECOM
Sophia Antipolis

Related work (EM)

Simple custom modulation/protocol

Name	Leakage Type	Modulation Type	Publication Venue
Soft-TEMPEST	Electromagnetic	AM, FSK	Information Hiding 1998
AirHopper	Elecromagnetic	FSK	MALWARE 2014
USBee	Elecromagnetic	FSK	PST 2016
GSMem	Elecromagnetic	OOK	USENIX Security 2015
BitJabber	Elecromagnetic	OOK, FSK	IEEE ITC 2020
MAGNETO	Magnetic	OOK, FSK	ArXiv 2018
ODINI	Magnetic	OOK-(many cores), FSK	IEEE Trans. Inf. Forensics Secur. 2020
Matyunin et. al	Magnetic	OOK, FSK	ASP-DAC 2016
EMLora	Electromagnetic	CSS	IEEE S&P 2021

A first step towards more advanced modulation

EURECOM
Sophia Antipolis

Limitations of previous work

Simple custom modulation

Mostly simple OOK or FSK, generally requires custom receivers

Limitations of previous work

Simple custom modulation
Mostly simple OOK or FSK, generally requires custom receivers

Simple custom protocol
Generally no error correction, etc.

Limitations of previous work

Simple custom modulation
Mostly simple OOK or FSK, generally requires custom receivers

Simple custom protocol
Generally no error correction, etc.

Not flexible
Only one fixed modulation

Limitations of previous work

Simple custom modulation
Mostly simple OOK or FSK, generally requires custom receivers

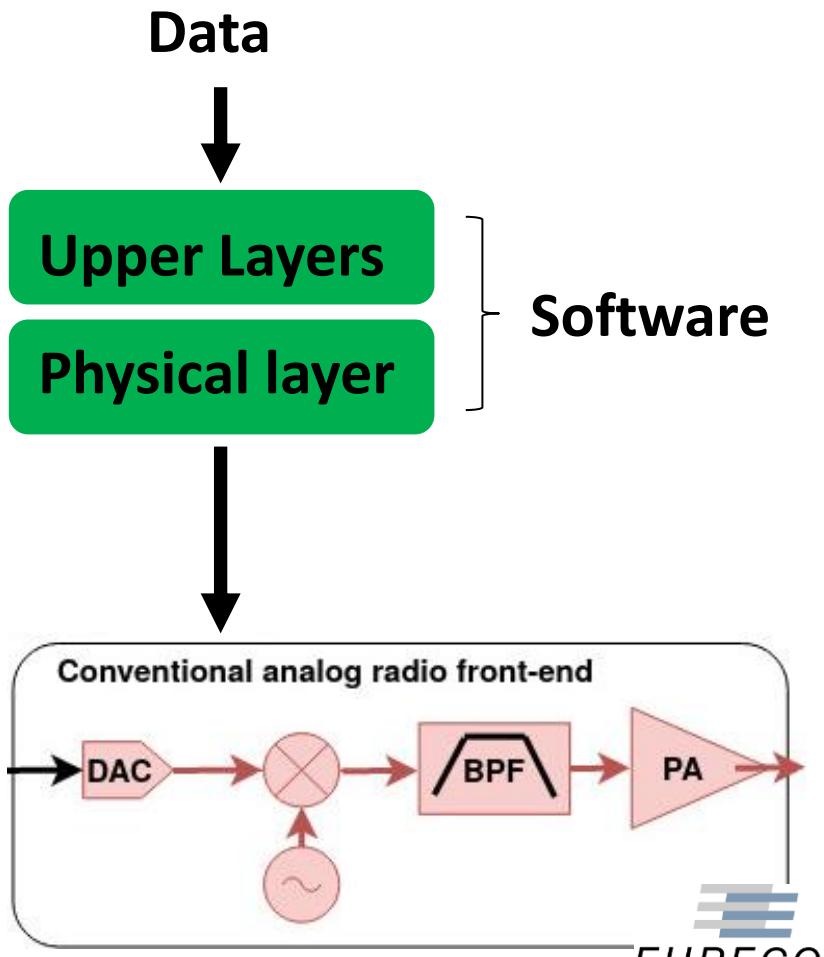
Simple custom protocol
Generally no error correction, etc.

Not flexible
Only one fixed modulation

Single application
Exfiltration from air-gapped devices

Meanwhile the (real) software-defined radios...

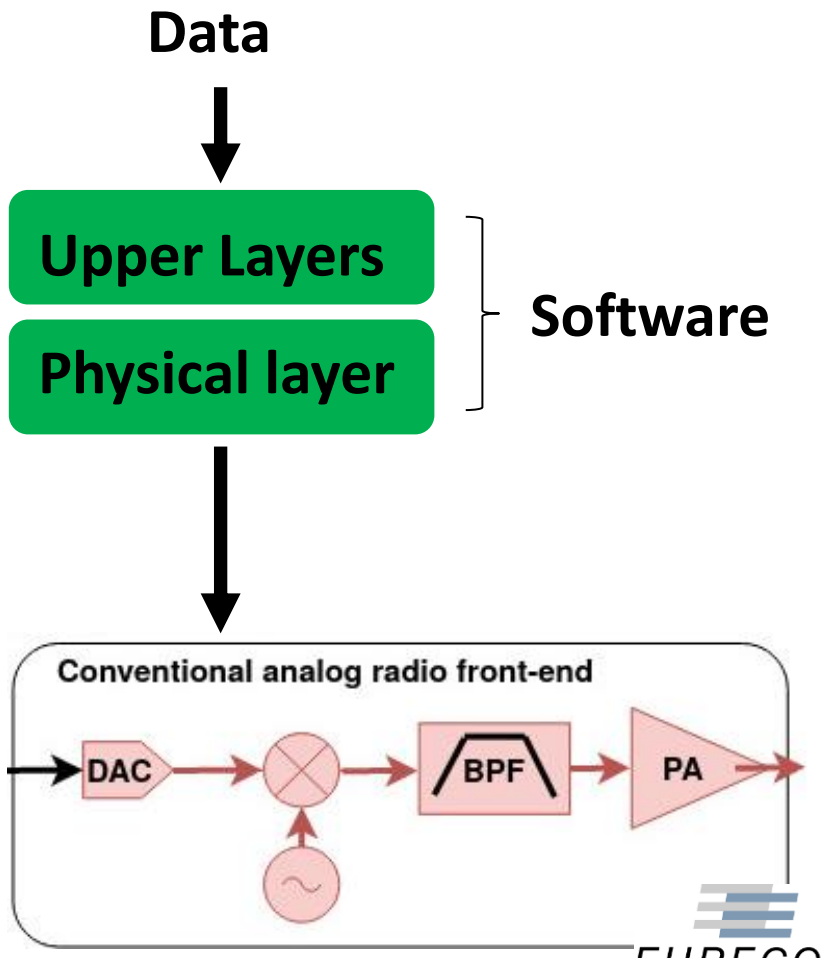
Software-defined
Signals entirely defined in software
Minimal hardware to create the actual waves



Meanwhile the (real) software-defined radios...

Software-defined
Signals entirely defined in software
Minimal hardware to create the actual waves

Arbitrary modulation
Can shape generic signals
Advanced modulation techniques possible

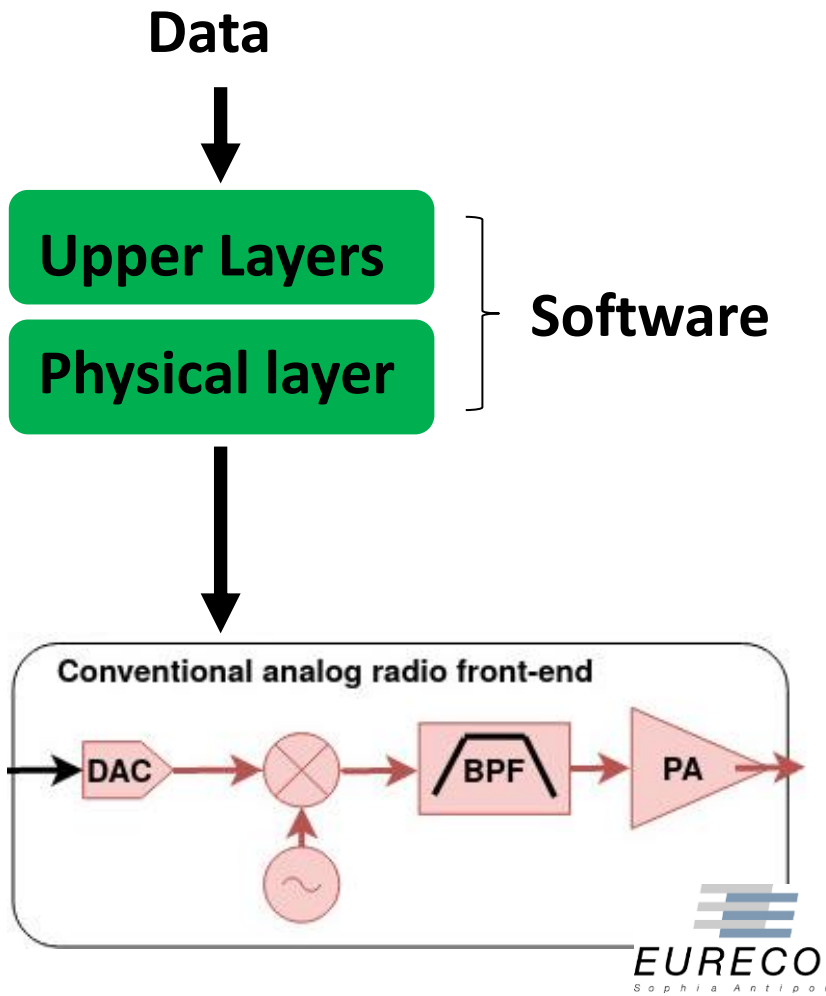


Meanwhile the (real) software-defined radios...

Software-defined
Signals entirely defined in software
Minimal hardware to create the actual waves

Arbitrary modulation
Can shape generic signals
Advanced modulation techniques possible

Advanced protocols
E.g., error correction



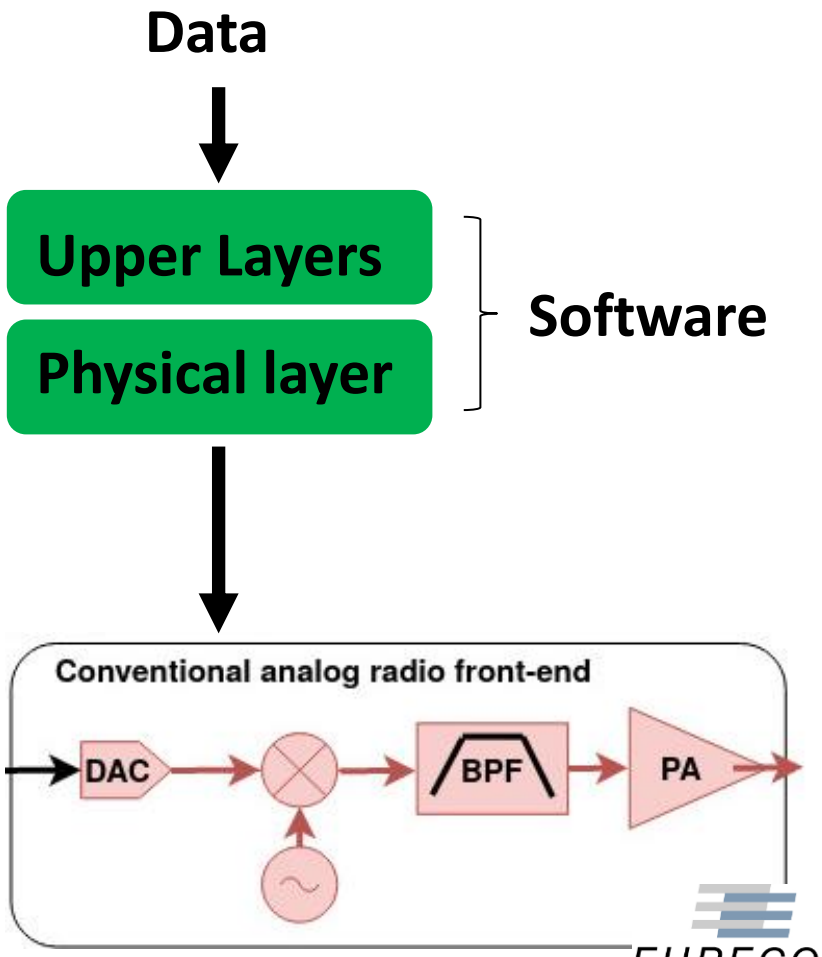
Meanwhile the (real) software-defined radios...

Software-defined
Signals entirely defined in software
Minimal hardware to create the actual waves

Arbitrary modulation
Can shape generic signals
Advanced modulation techniques possible

Advanced protocols
E.g., error correction

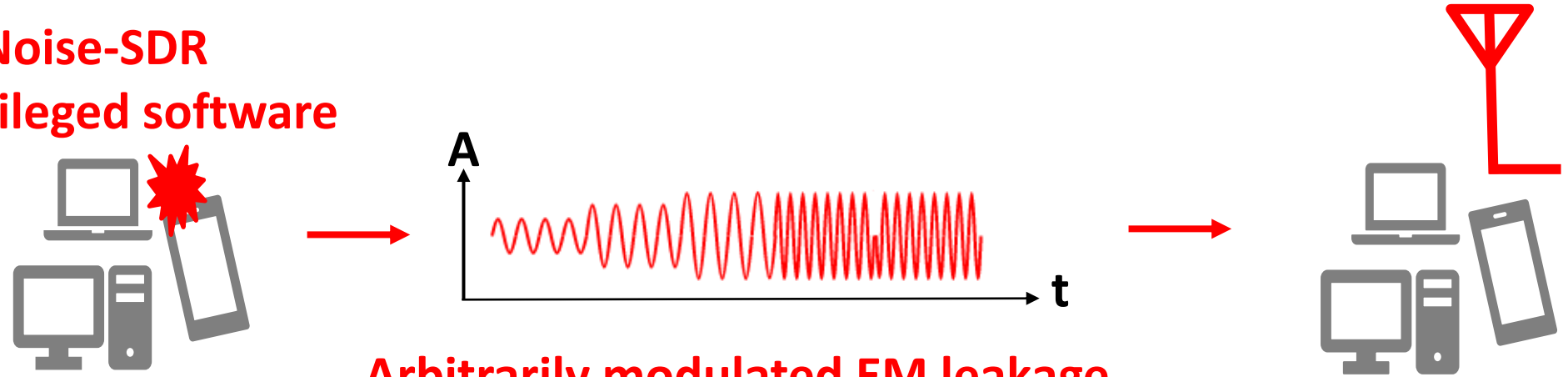
Flexible
Can handle virtually any protocol / application



Can we do more with Soft-TEMPEST?

Goal: Can we do more?

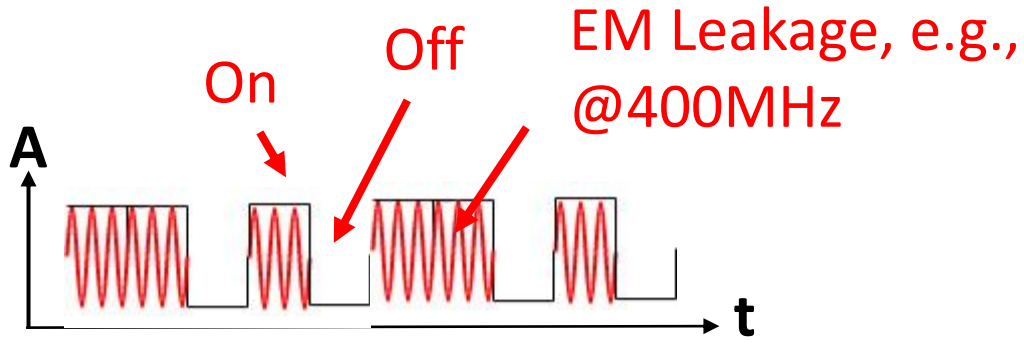
Noise-SDR
Unprivileged software



Arbitrarily modulated EM leakage

- **+ Software-defined:** flexibility, existing protocols
- **+ Advanced PHY layer:** performance
- **+ More applications:** exfiltration, tracking, injection, …

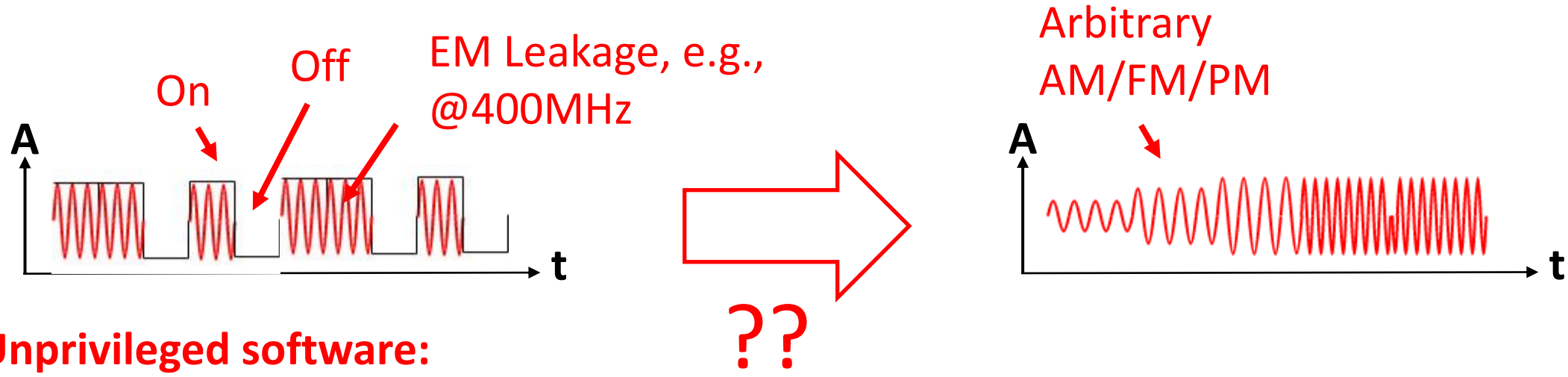
Challenge: from square wave to generic passband signal



Unprivileged software:

1. **DRAM access:** “EM leakage ON”
2. **Do nothing:** “EM leakage OFF”

Challenge: from square wave to generic passband signal



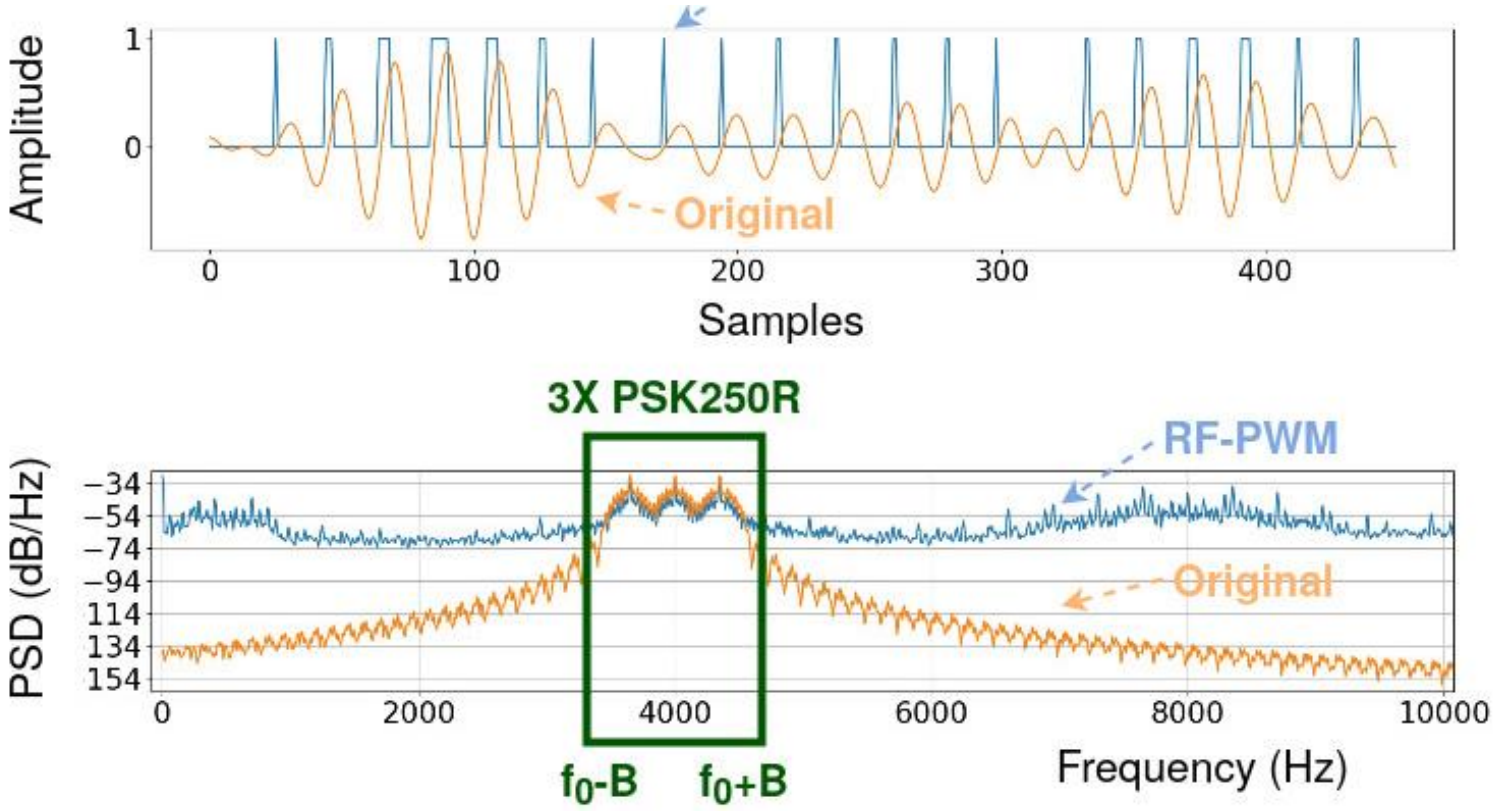
Unprivileged software:

1. DRAM access: “EM leakage ON”

2. Do nothing: “EM leakage OFF”

Solution: leverage pass-band one-bit coding (RF-PWM)

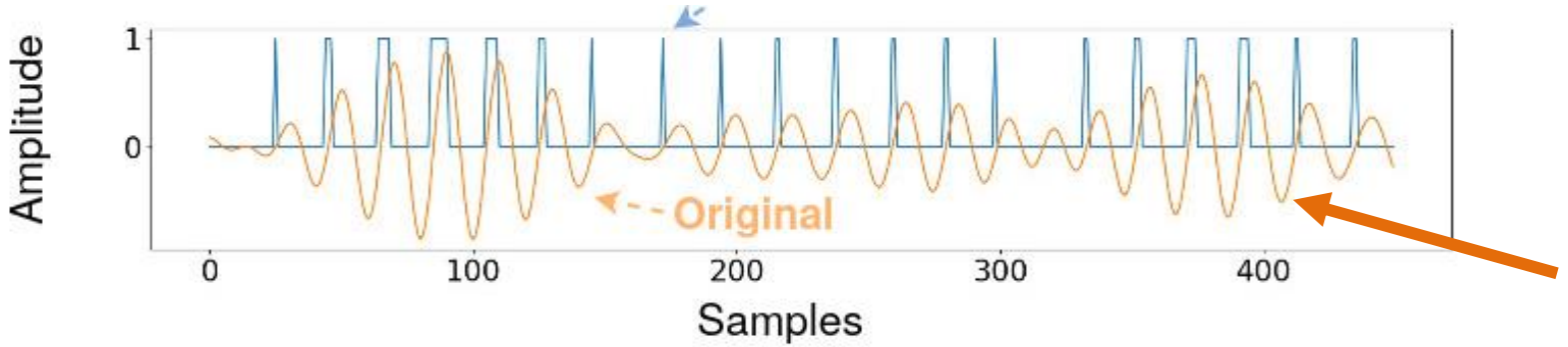
Long story short: approximate a modulated sine-wave with a square wave



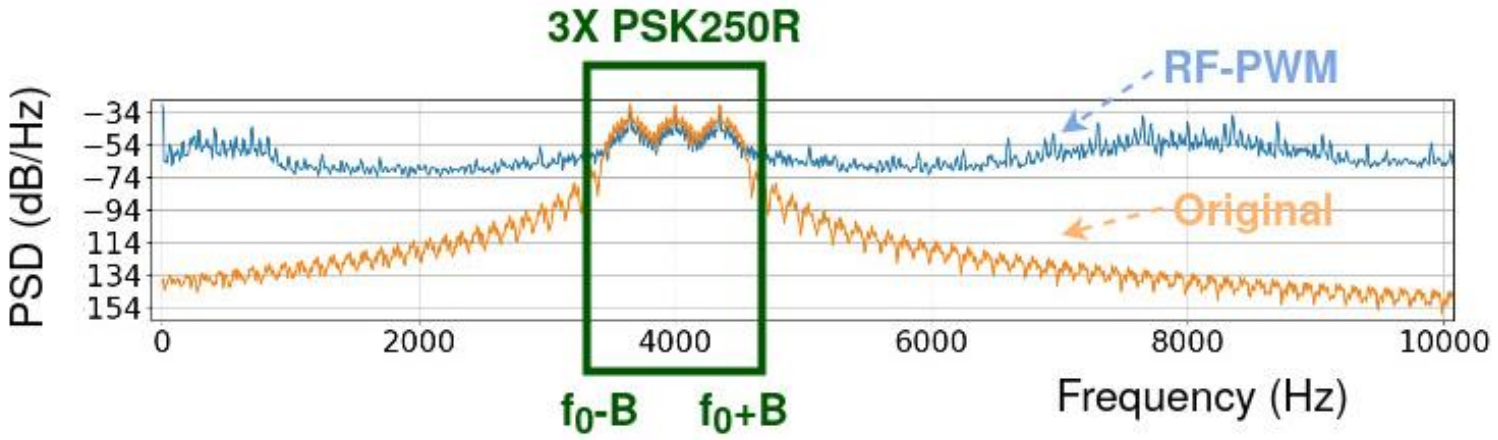
P. AJ Nuyts, P. Reynaert, and W. Dehaene, “Continuous-Time Digital Front-Ends for Multistandard Wireless Transmission” (Springer, 2014).

Solution: leverage pass-band one-bit coding (RF-PWM)

Long story short: approximate a modulated sine-wave with a square wave



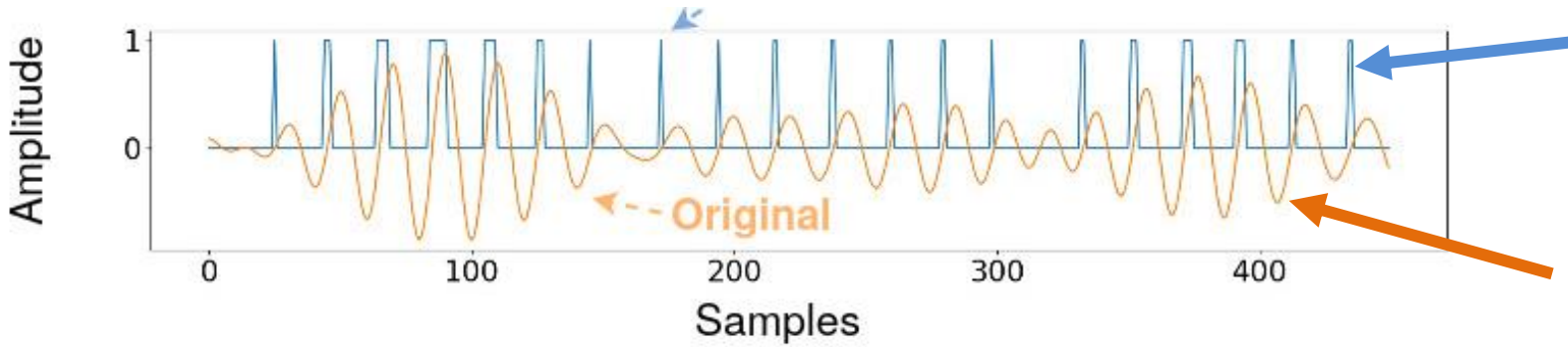
Goal:
3xPSK250R



P. AJ Nuyts, P. Reynaert, and W. Dehaene, “Continuous-Time Digital Front-Ends for Multistandard Wireless Transmission” (Springer, 2014).

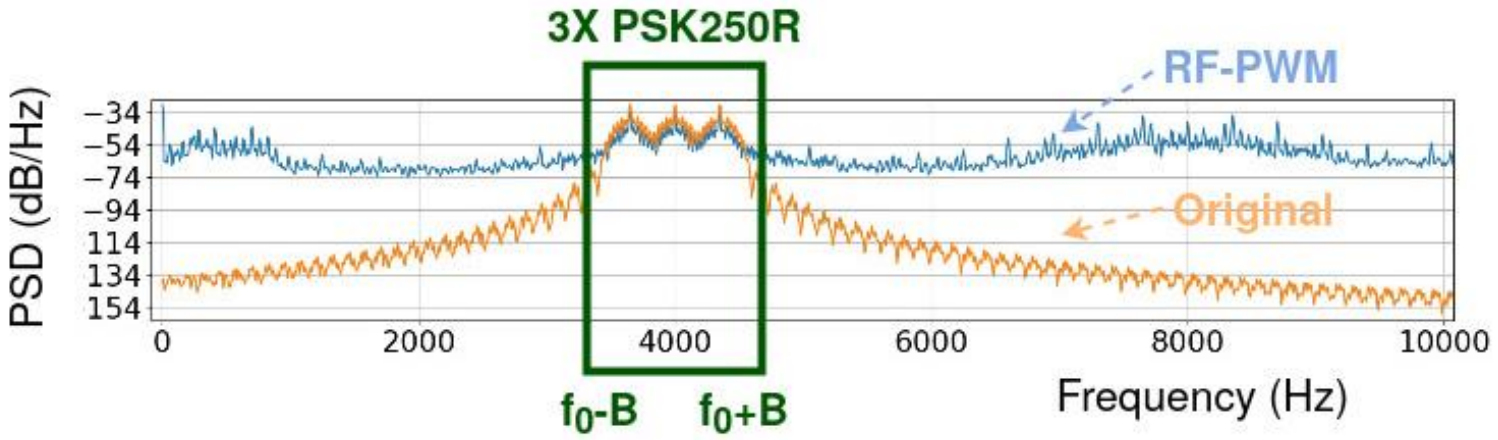
Solution: leverage pass-band one-bit coding (RF-PWM)

Long story short: approximate a modulated sine-wave with a square wave



Approximation:
RF-PWM square wave

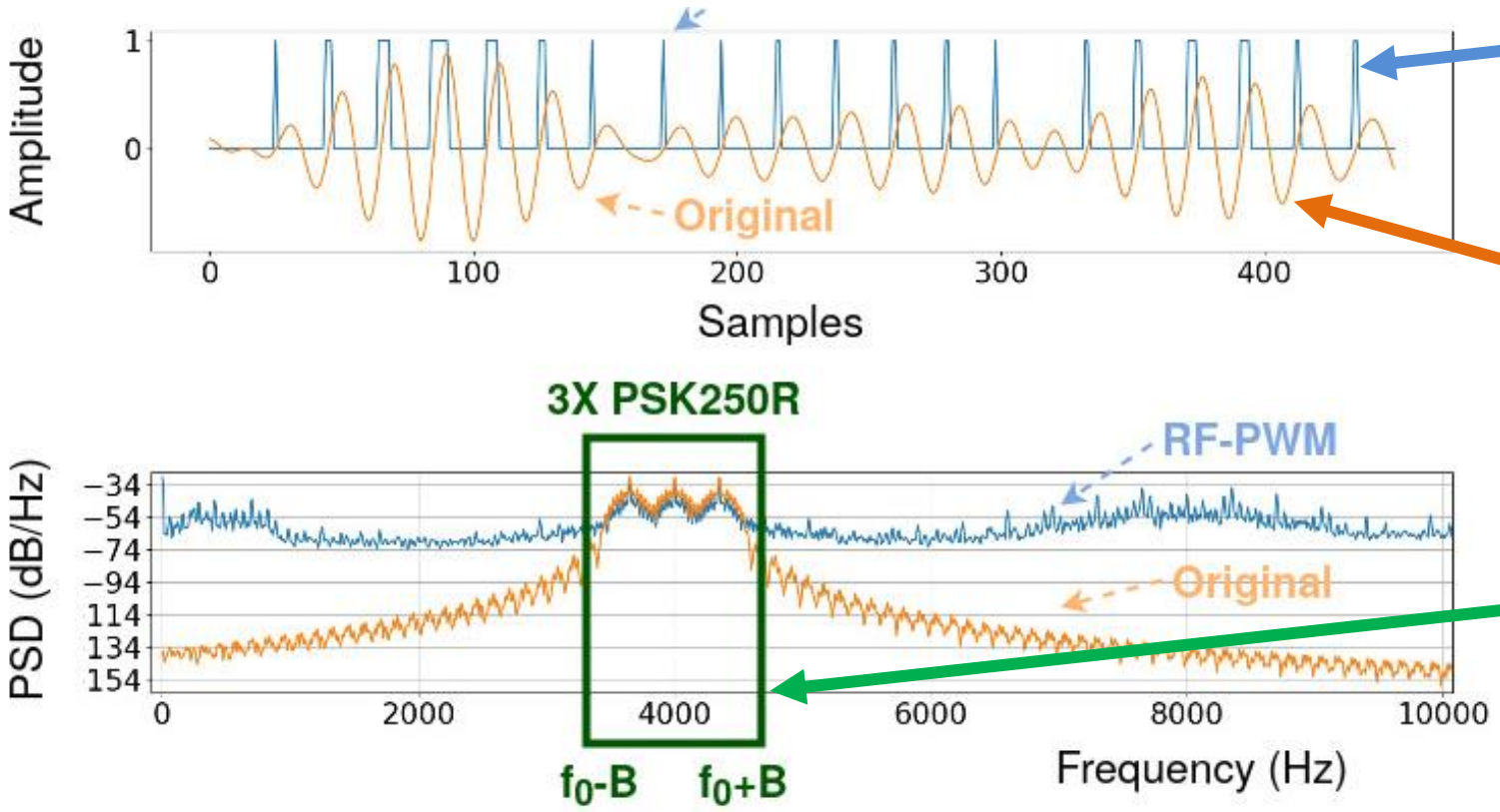
Goal:
3xPSK250R



P. AJ Nuyts, P. Reynaert, and W. Dehaene, “Continuous-Time Digital Front-Ends for Multistandard Wireless Transmission” (Springer, 2014).

Solution: leverage pass-band one-bit coding (RF-PWM)

Long story short: approximate a modulated sine-wave with a square wave



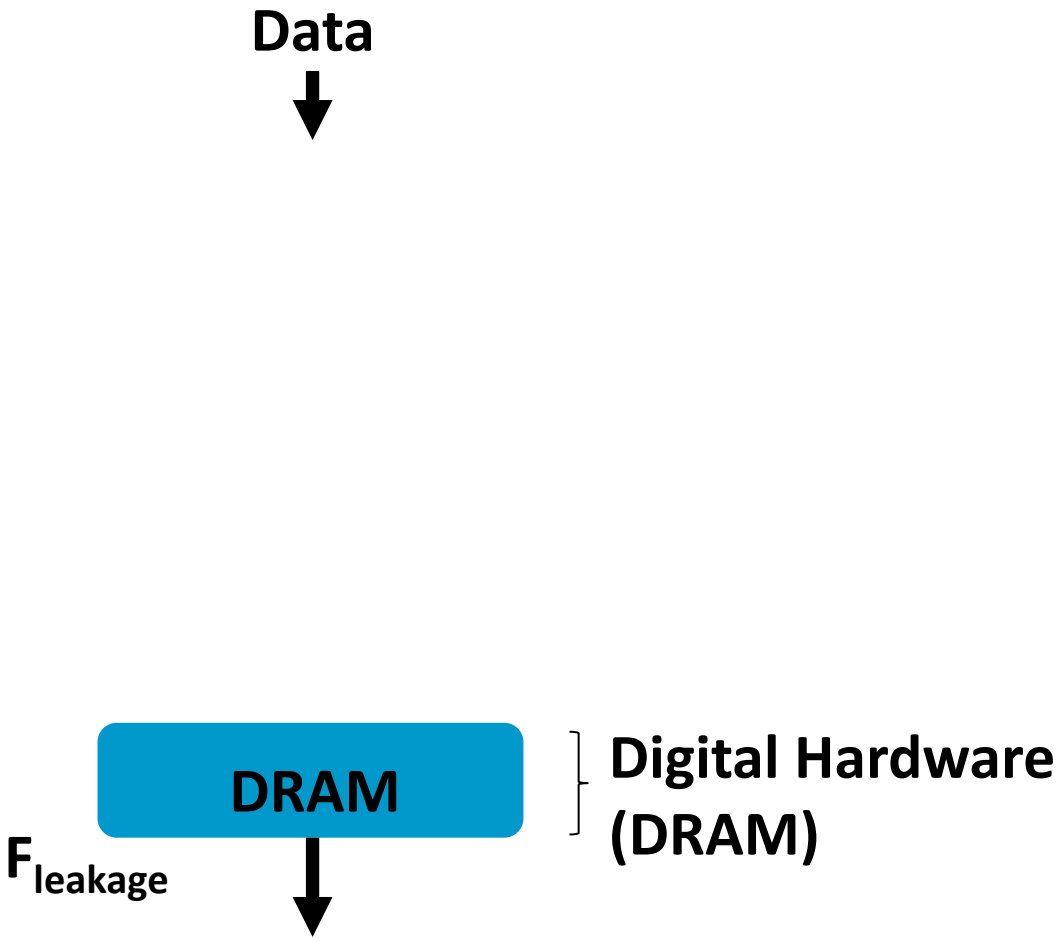
Approximation:
RF-PWM square wave

Goal:
3xPSK250R

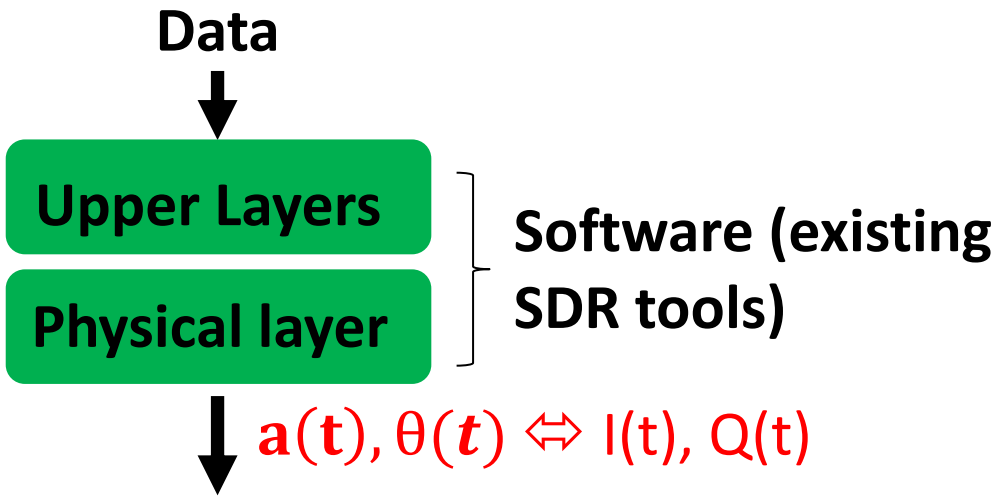
Band of interest:
Good approximation

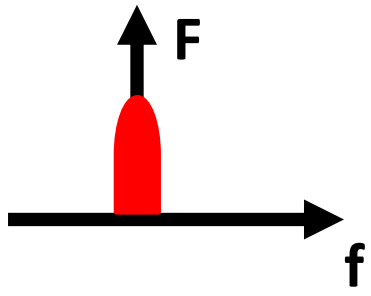
P. AJ Nuyts, P. Reynaert, and W. Dehaene, “Continuous-Time Digital Front-Ends for Multistandard Wireless Transmission” (Springer, 2014).

Noise-SDR: Arbitrary Modulation of EM noise

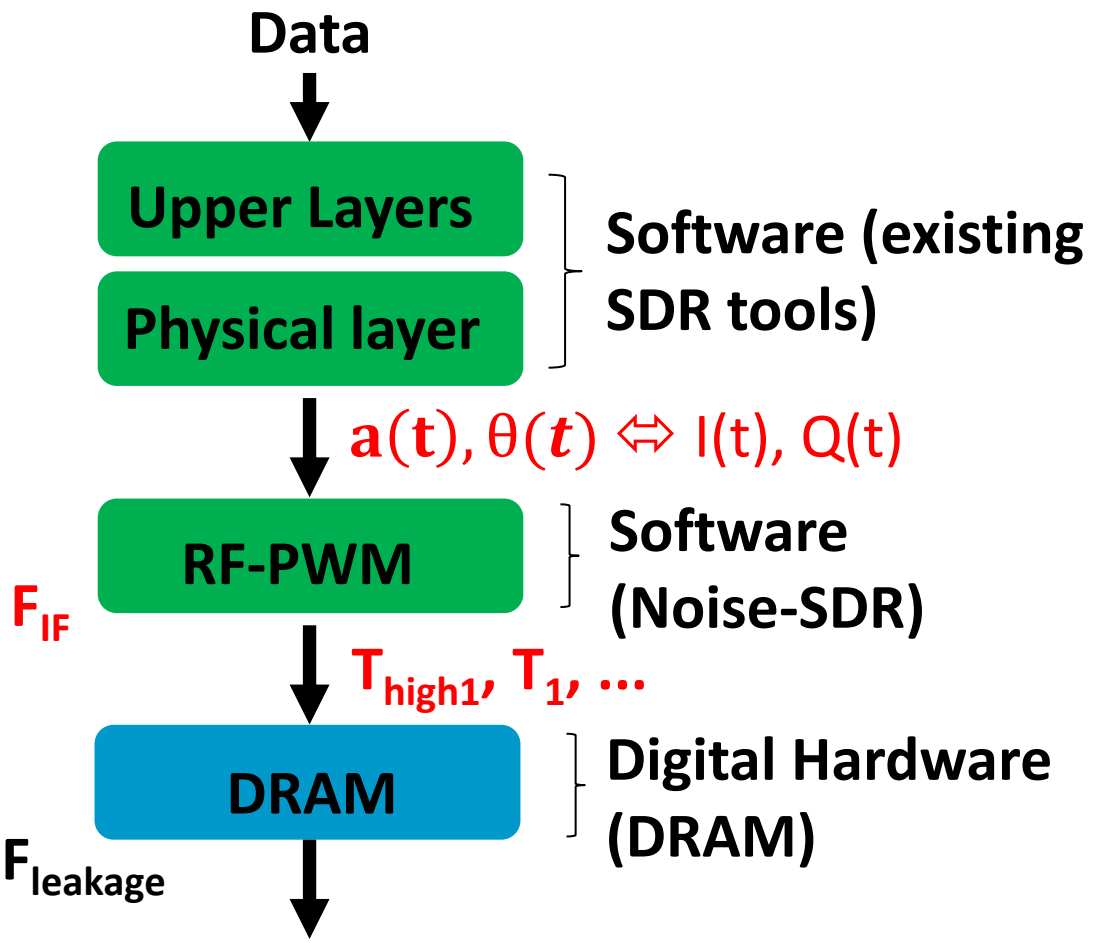


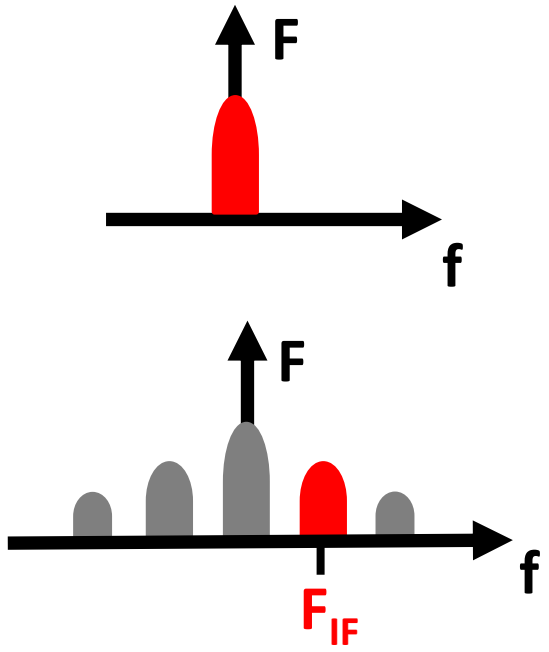
Noise-SDR: Arbitrary Modulation of EM noise



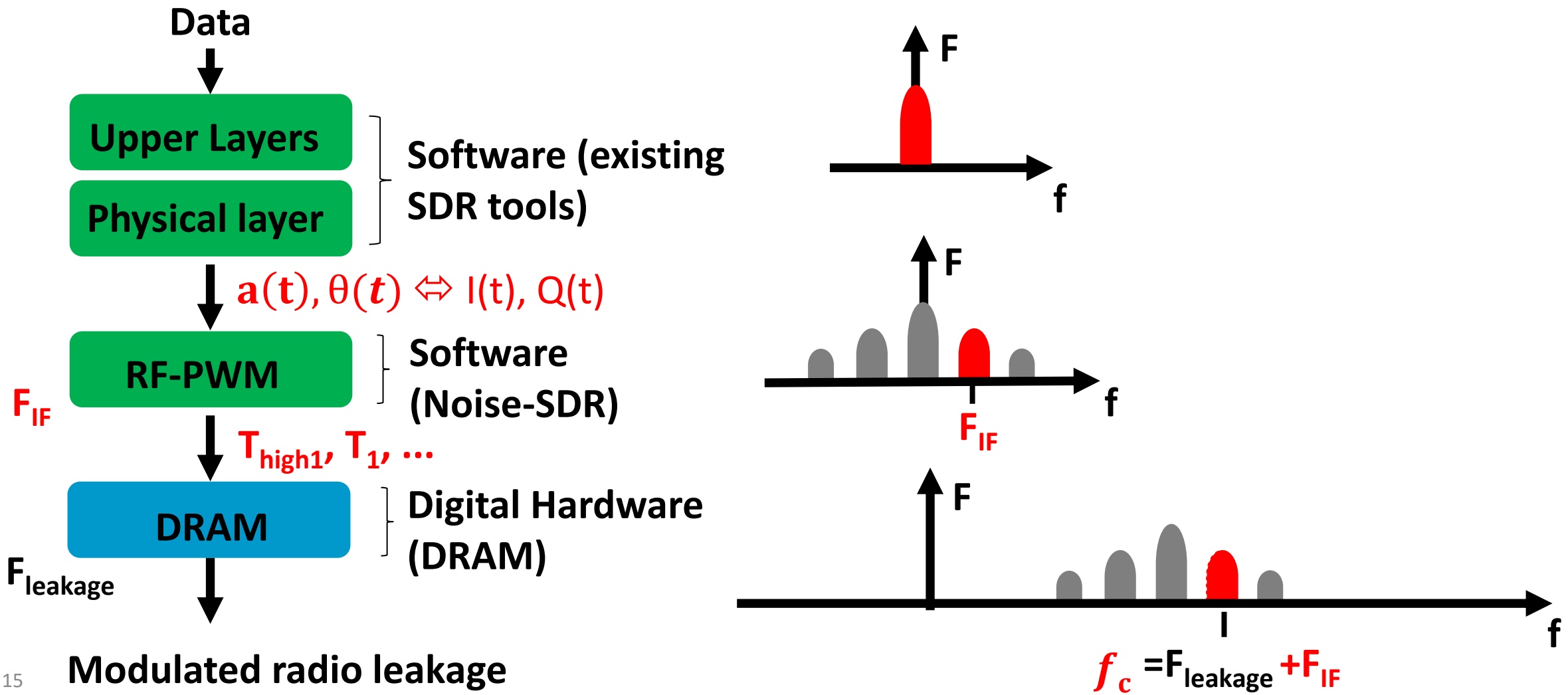


Noise-SDR: Arbitrary Modulation of EM noise

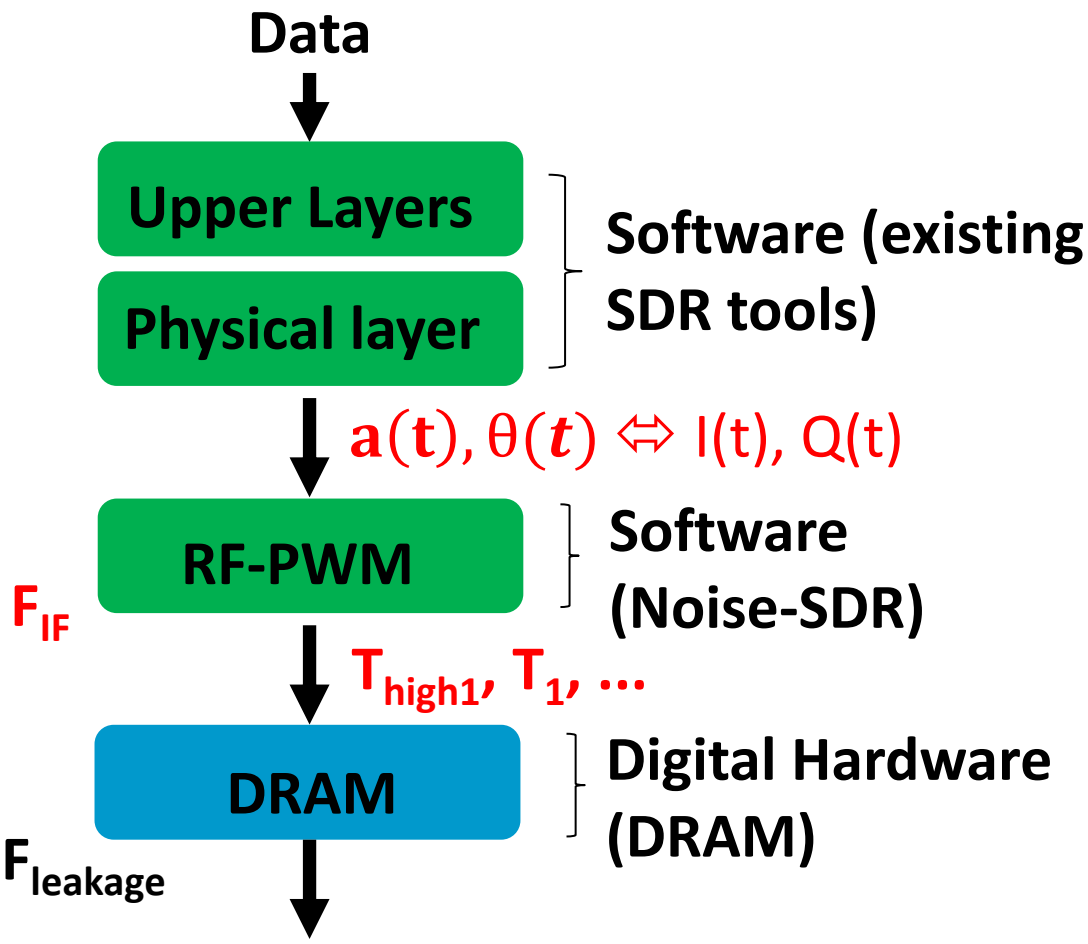


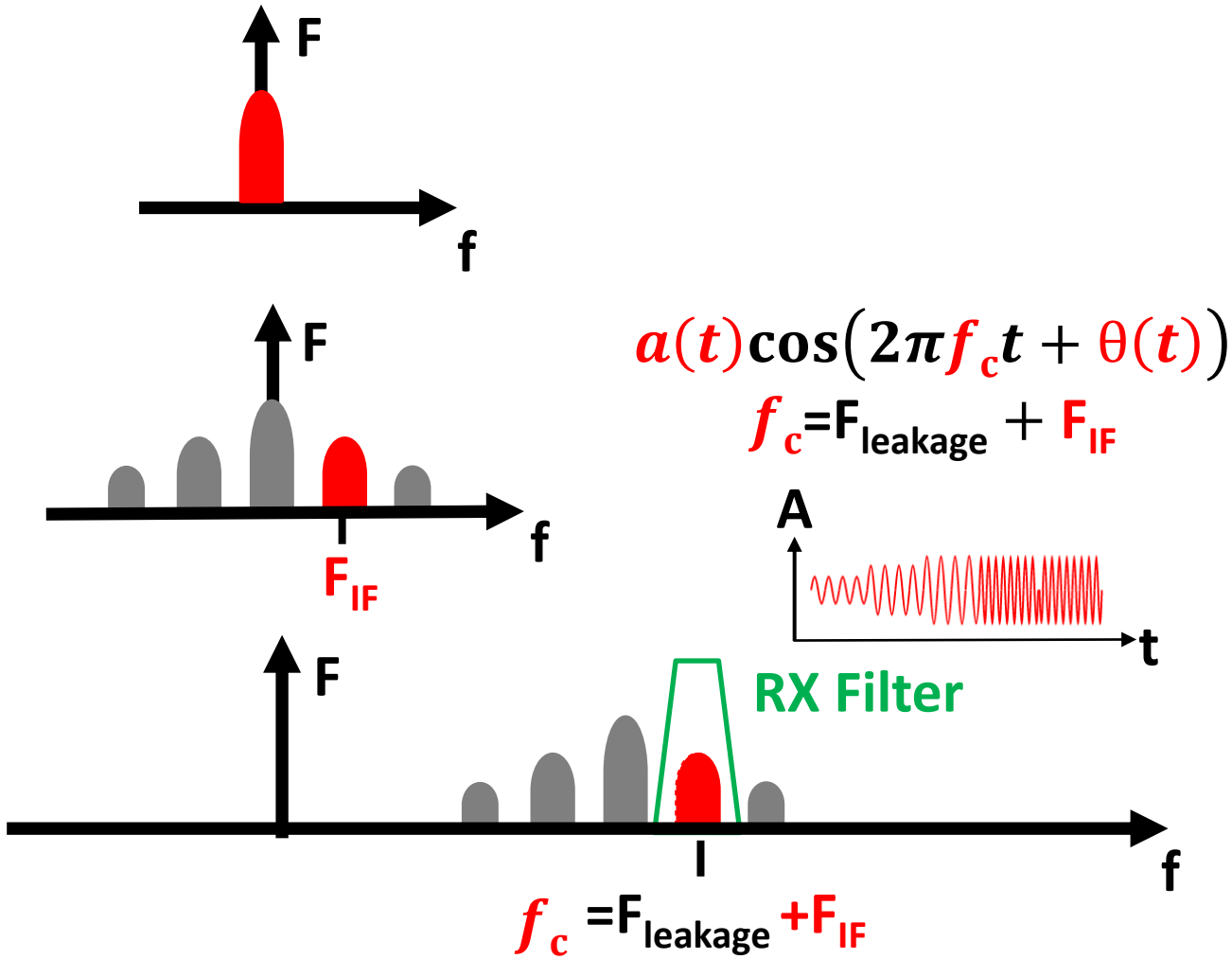


Noise-SDR: Arbitrary Modulation of EM noise



Noise-SDR: Arbitrary Modulation of EM noise



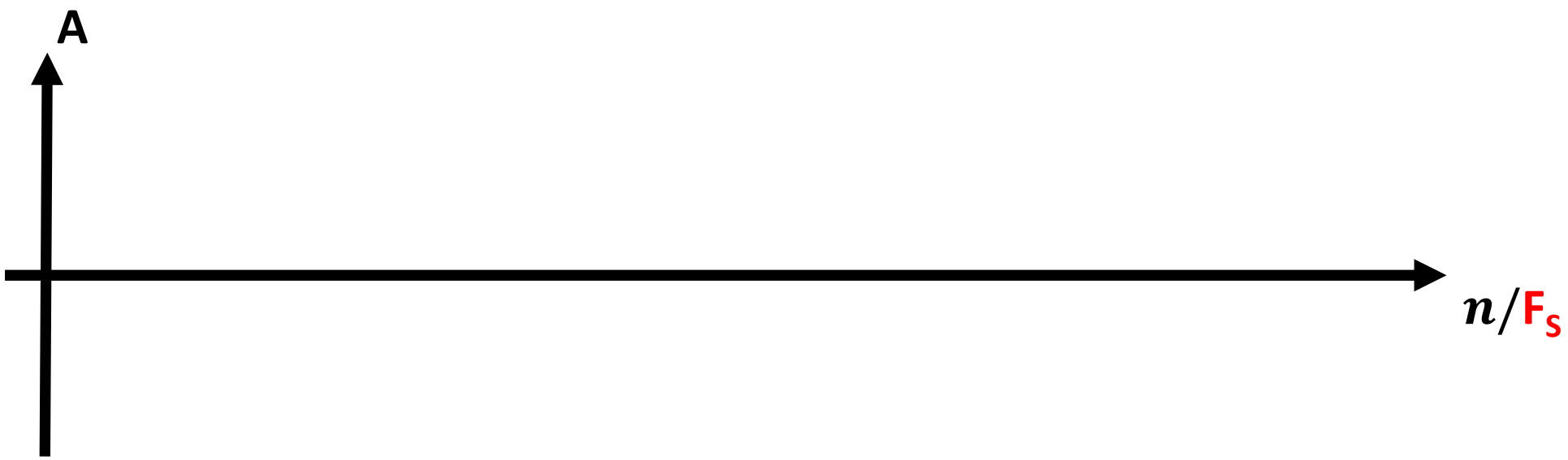


How do we implement it in practice?

Implementation: discrete-time RF-PWM

Input: F_S, $a(n/F_S)$, $\theta(n/F_S)$, F_{IF}

Simplified explanation



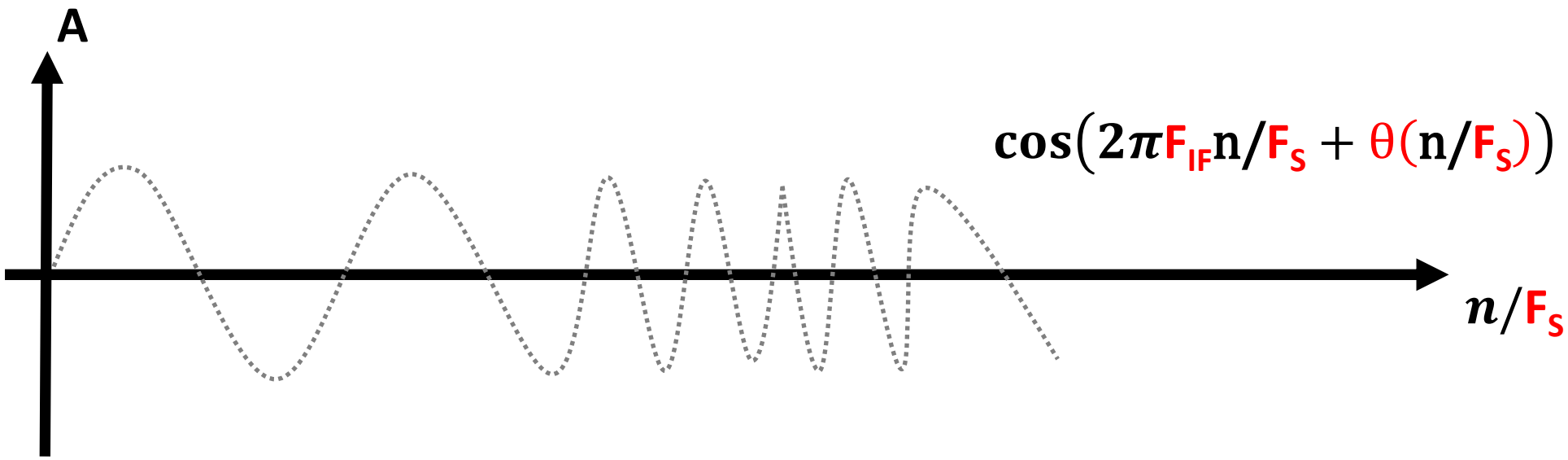
Output:

EURECOM
Sophia Antipolis

Implementation: discrete-time RF-PWM

Input: F_S, $a(n/F_S)$, $\theta(n/F_S)$, F_{IF}

Simplified explanation



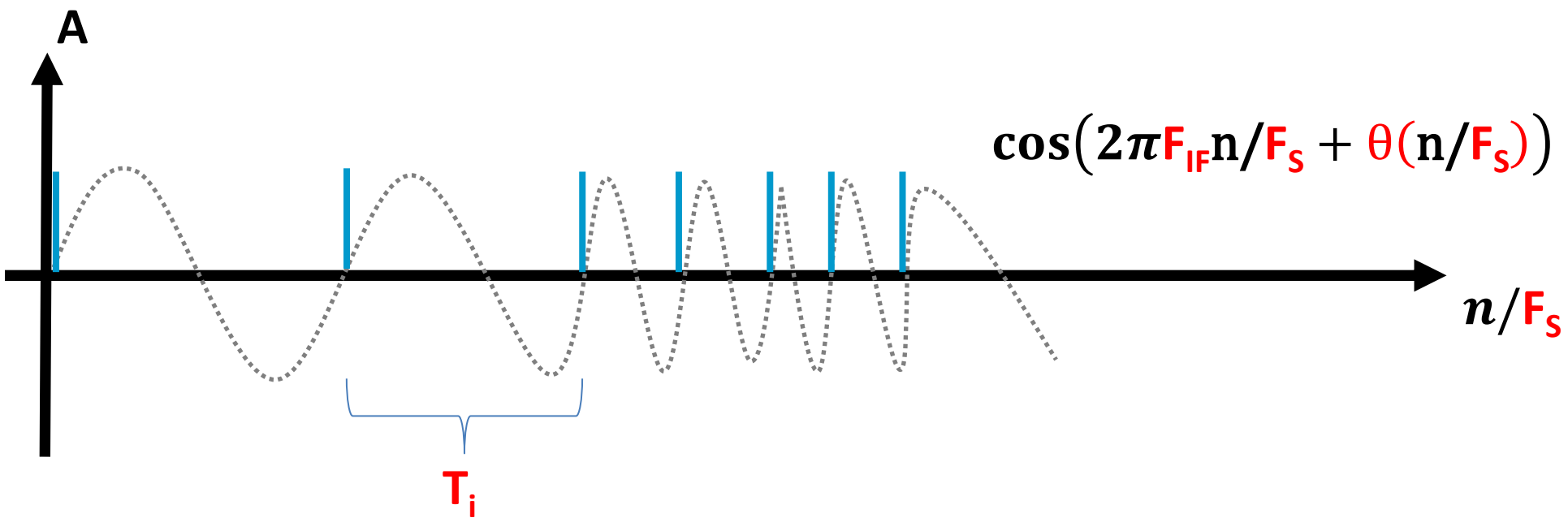
Output:

EURECOM
Sophia Antipolis

Implementation: discrete-time RF-PWM

Input: F_S, $a(n/F_S)$, $\theta(n/F_S)$, F_{IF}

Simplified explanation



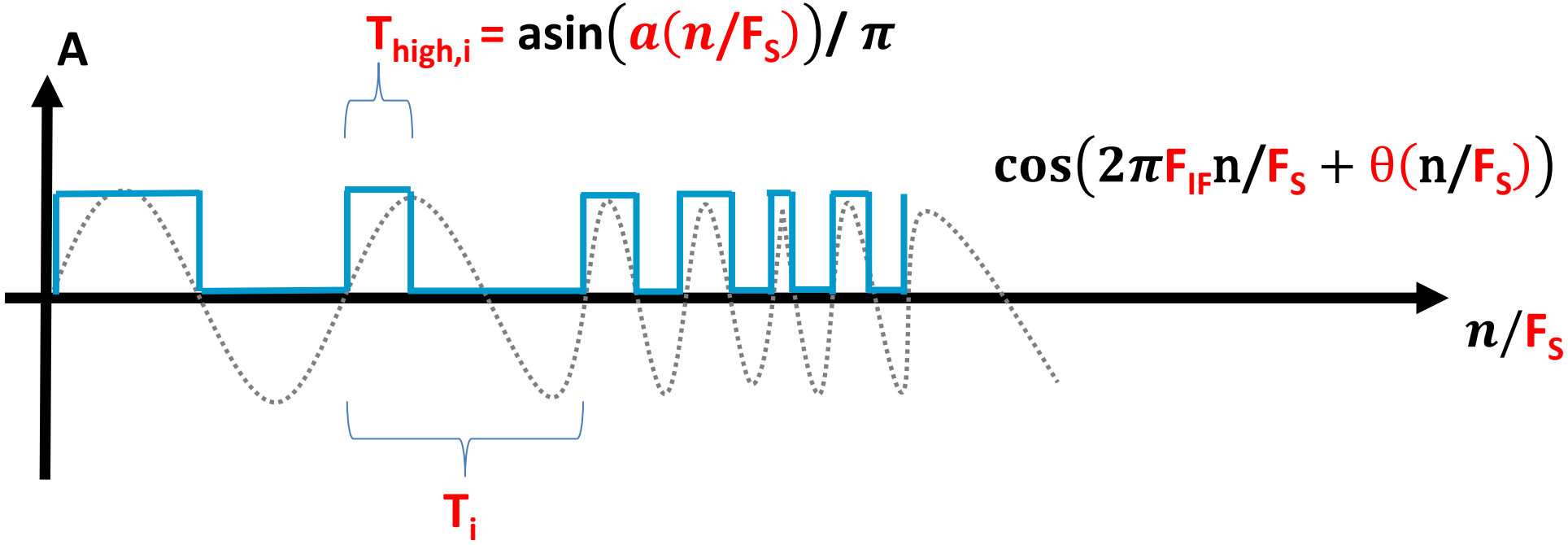
Output: T_1, T_2, T_3, ...

EURECOM
Sophia Antipolis

Implementation: discrete-time RF-PWM

Input: F_S, $a(n/F_S)$, $\theta(n/F_S)$, F_{IF}

Simplified explanation



Output: $T_{high,1}$ T_1, $T_{high,2}$ T_2, $T_{high,3}$ T_3, ...

EURECOM
Sophia Antipolis

Implementation: software-control

```
start = now()
while( now() – start < T_high,i )
        leakyOperation()
while( now() – start < T_i )
        doNothing()
```

Time accuracy is fundamental!
(Bandwidth, am/fm/pm quantization)

*M. Schwarz et al., “Fantastic Timers and Where to Find Them: High-Resolution Microarchitectural Attacks in JavaScript,” in FC 2017.

**Z. Zhang et al., “Leveraging EM Side-Channel Information to Detect Rowhammer Attacks,” in IEEE S&P 2020

***Z. Zhang et al., “Triggering Rowhammer Hardware Faults on ARM: A Revisit,” ASHES@CCS 2018.

Implementation: software-control

```
start = now()
while( now() – start < T_high,i )
        leakyOperation()
while( now() – start < T_i )
        doNothing()
```

Time accuracy is fundamental!
(Bandwidth, am/fm/pm quantization)

Leaky**, fast***

Accurate*, stable

*M. Schwarz et al., “Fantastic Timers and Where to Find Them: High-Resolution Microarchitectural Attacks in JavaScript,” in FC 2017.

**Z. Zhang et al., “Leveraging EM Side-Channel Information to Detect Rowhammer Attacks,” in IEEE S&P 2020

***Z. Zhang et al., “Triggering Rowhammer Hardware Faults on ARM: A Revisit,” ASHES@CCS 2018.

Implementation: software-control

```
start = now()
while( now() – start < T_high,i )
        leakyOperation()
while( now() – start < T_i )
        doNothing()
```

Time accuracy is fundamental!
(Bandwidth, am/fm/pm quantization)

Leaky, fast*****

Accurate*, stable

clock_gettime()

(or µ-arch attacks literature)

*M. Schwarz et al., “Fantastic Timers and Where to Find Them: High-Resolution Microarchitectural Attacks in JavaScript,” in FC 2017.

**Z. Zhang et al., “Leveraging EM Side-Channel Information to Detect Rowhammer Attacks,” in IEEE S&P 2020

***Z. Zhang et al., “Triggering Rowhammer Hardware Faults on ARM: A Revisit,” ASHES@CCS 2018.

Implementation: software-control

```
start = now()
while( now() – start < T_high,i )
        leakyOperation()
while( now() – start < T_i )
        doNothing()
```

Time accuracy is fundamental!
(Bandwidth, am/fm/pm quantization)

Leaky, fast*****

Many in the paper and in general

Accurate*, stable

clock_gettime()

(or µ-arch attacks literature)

*M. Schwarz et al., “Fantastic Timers and Where to Find Them: High-Resolution Microarchitectural Attacks in JavaScript,” in FC 2017.

**Z. Zhang et al., “Leveraging EM Side-Channel Information to Detect Rowhammer Attacks,” in IEEE S&P 2020

***Z. Zhang et al., “Triggering Rowhammer Hardware Faults on ARM: A Revisit,” ASHES@CCS 2018.

Implementation: software-control

```
start = now()
while( now() – start < T_high,i )
        leakyOperation()
while( now() – start < T_i )
        doNothing()
```

Accurate*, stable

clock_gettime()

(or µ-arch attacks literature)

Time accuracy is fundamental!
(Bandwidth, am/fm/pm quantization)

Leaky, fast*****

Many in the paper and in general

E.g., on Arm-v8 (re)use ROWHAMMER

```
__attribute__((naked)) \
void hammer_civac(uint64_t *addr) {
  __asm volatile("LDR X9, [X0]");
  __asm volatile("DC CIVAC, X0");
  __asm volatile("DSB 0xB");
  __asm volatile("RET");
}
```

*M. Schwarz et al., “Fantastic Timers and Where to Find Them: High-Resolution Microarchitectural Attacks in JavaScript,” in FC 2017.

**Z. Zhang et al., “Leveraging EM Side-Channel Information to Detect Rowhammer Attacks,” in IEEE S&P 2020

***Z. Zhang et al., “Triggering Rowhammer Hardware Faults on ARM: A Revisit,” ASHES@CCS 2018.

Implementation: combine Noise-SDR with popular SDR tools

Fldigi-Noise-SDR

> ./fldigi-noise-sdr -I secret.txt -m MODE_3X_PSK250R –c 4000

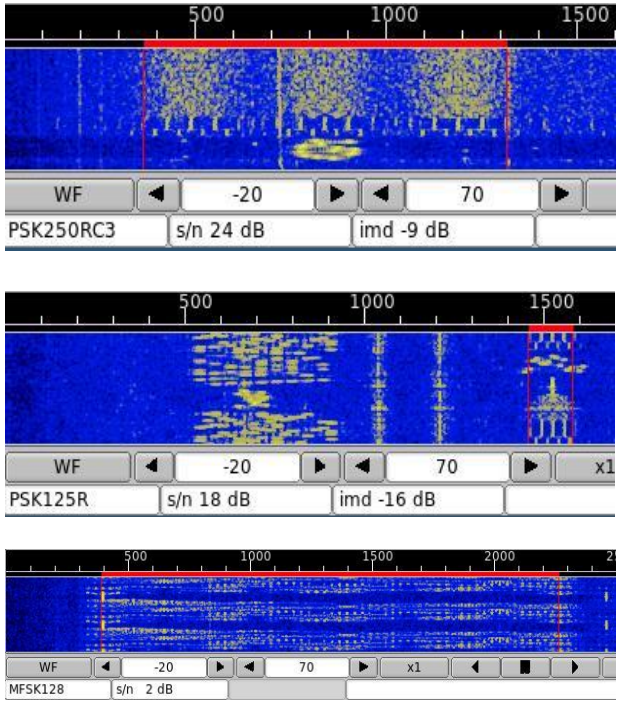
Or **Gnuradio+RF-PWM+Offline-Noise-SDR**

>./rfpwm.py ft4.iq ft4.timings
>./offline-noise-sdr ft4.timings

Evaluation

Noise-SDR in action: a few examples

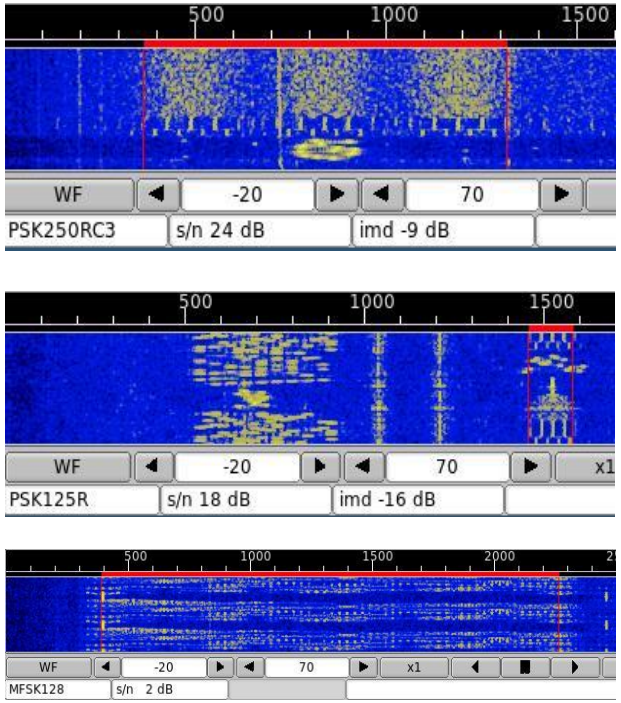
More videos: https://github.com/eurecom-s3/noise-sdr



EURECOM
Sophia Antipolis

Noise-SDR in action: a few examples

More videos: https://github.com/eurecom-s3/noise-sdr



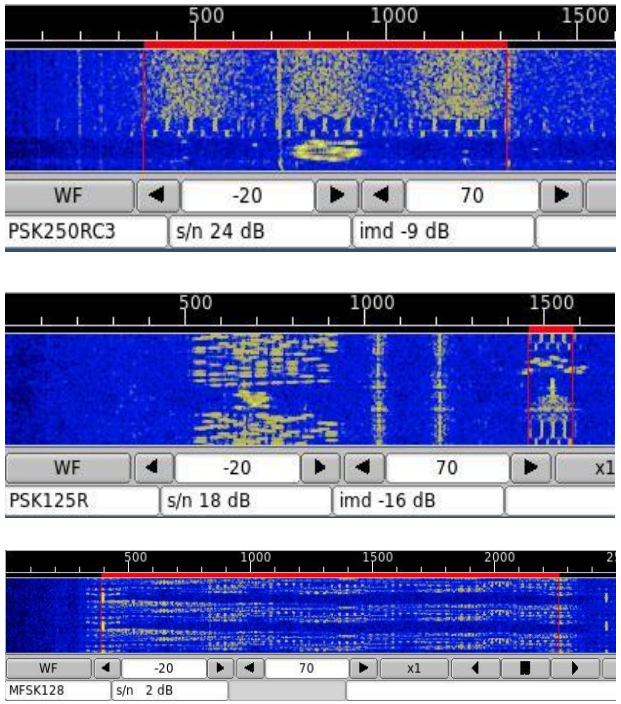
Pros

1. Software-defined, flexible
2. AM, FM, PSK, RTTY, THOR, SSTV, LoRa, GLONASS C/A, etc.
3. ArmV7A, ArmV8A, x86, MIPS
4. Mobile/desktop/laptop/IoT

EURECOM
Sophia Antipolis

Noise-SDR in action: a few examples

More videos: https://github.com/eurecom-s3/noise-sdr



Pros

1. Software-defined, flexible
2. AM, FM, PSK, RTTY, THOR, SSTV, LoRa, GLONASS C/A, etc.
3. ArmV7A, ArmV8A, x86, MIPS
4. Mobile/desktop/laptop/IoT

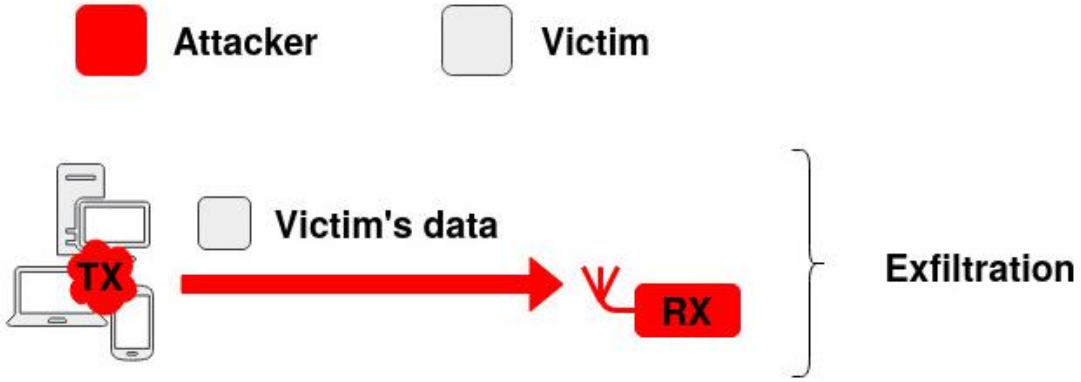
Limitations

1. There must be a leakage (device dependent)
2. Limited bandwidth
3. Limited choice of carrier frequency

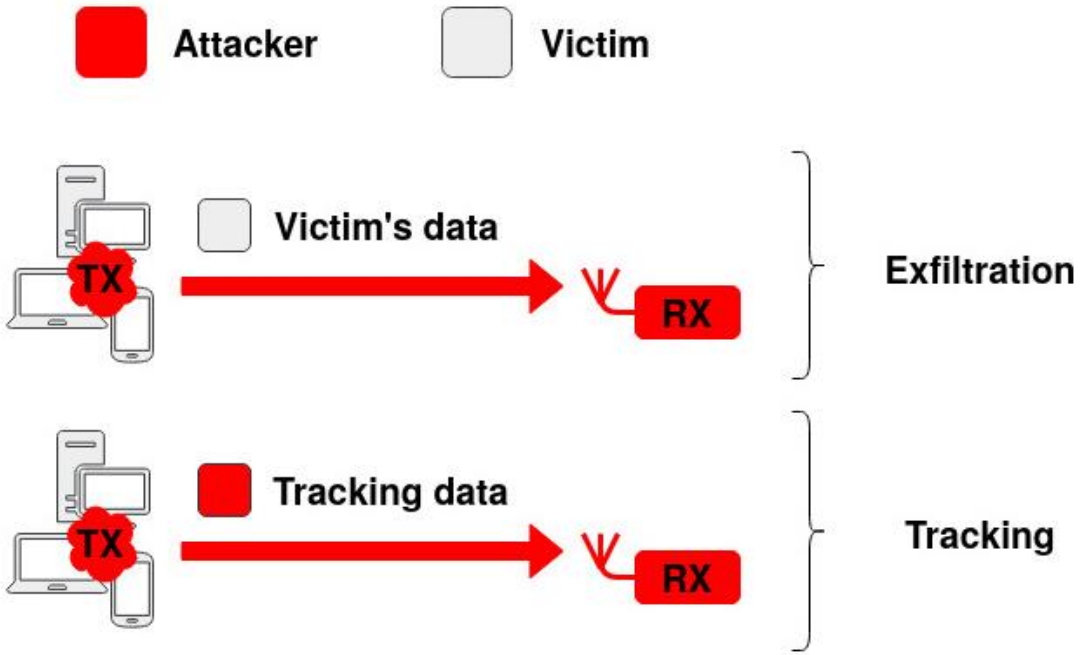
EURECOM
Sophia Antipolis

Security Impact

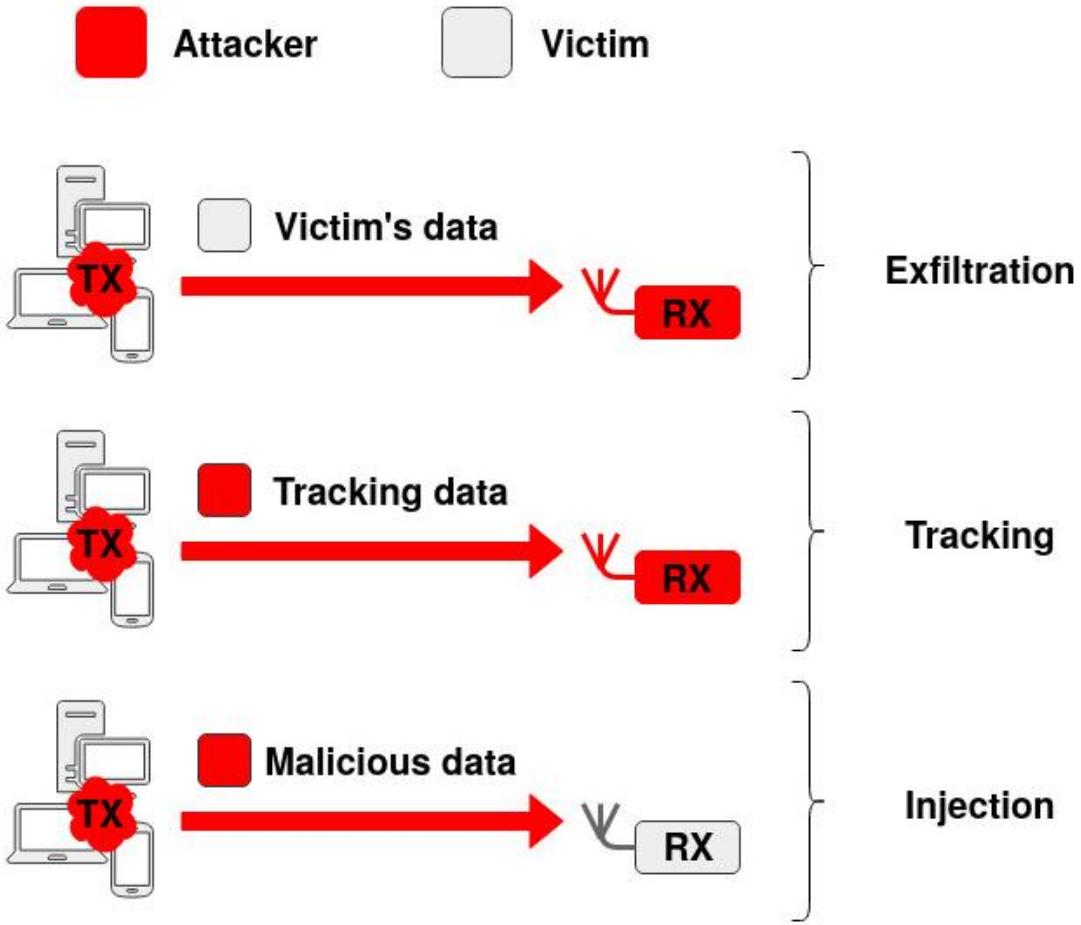
Security impact: exfiltration, tracking, injection



Security impact: exfiltration, tracking, injection

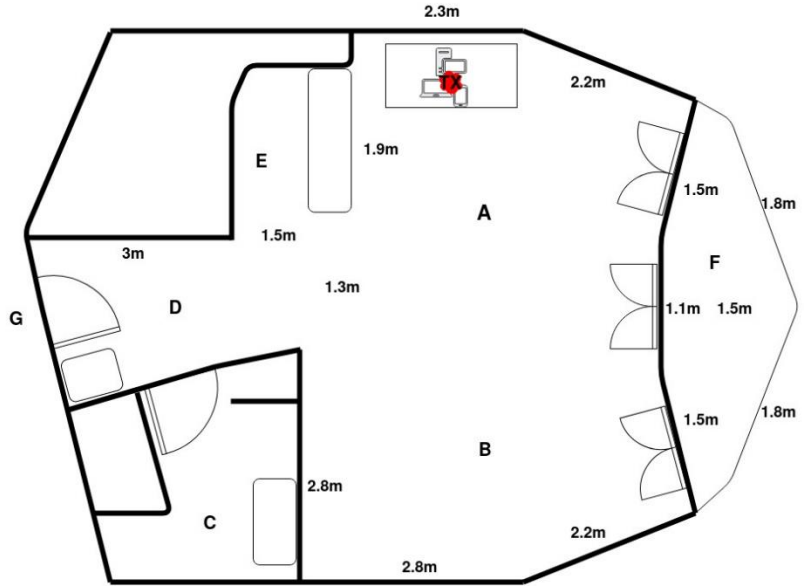


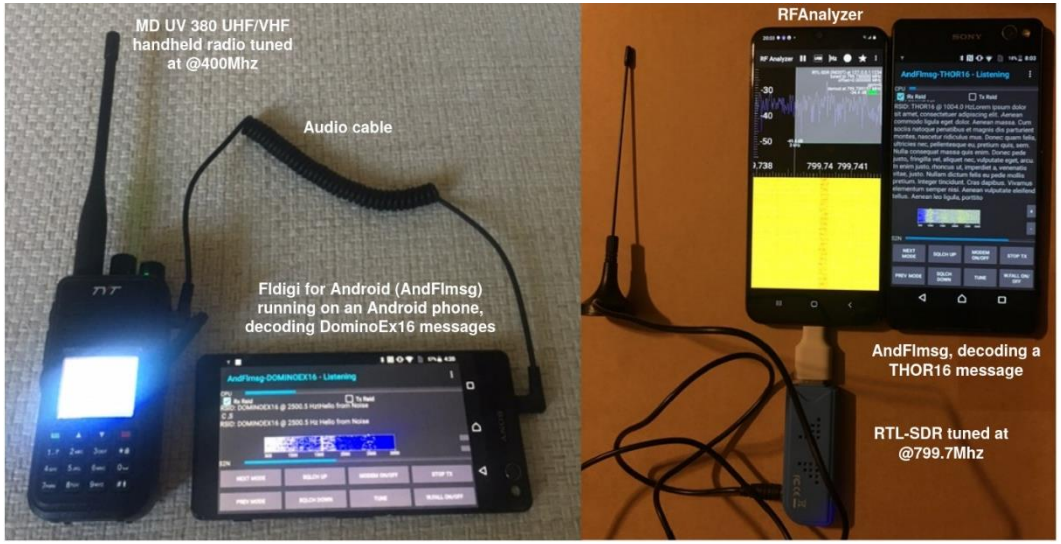
Security impact: exfiltration, tracking, injection



EURECOM
Sophia Antipolis

Examples of Exfiltration





E.g., robust long-range: THOR and RSIDs (e.g., MIPS32 >5m beind wall)

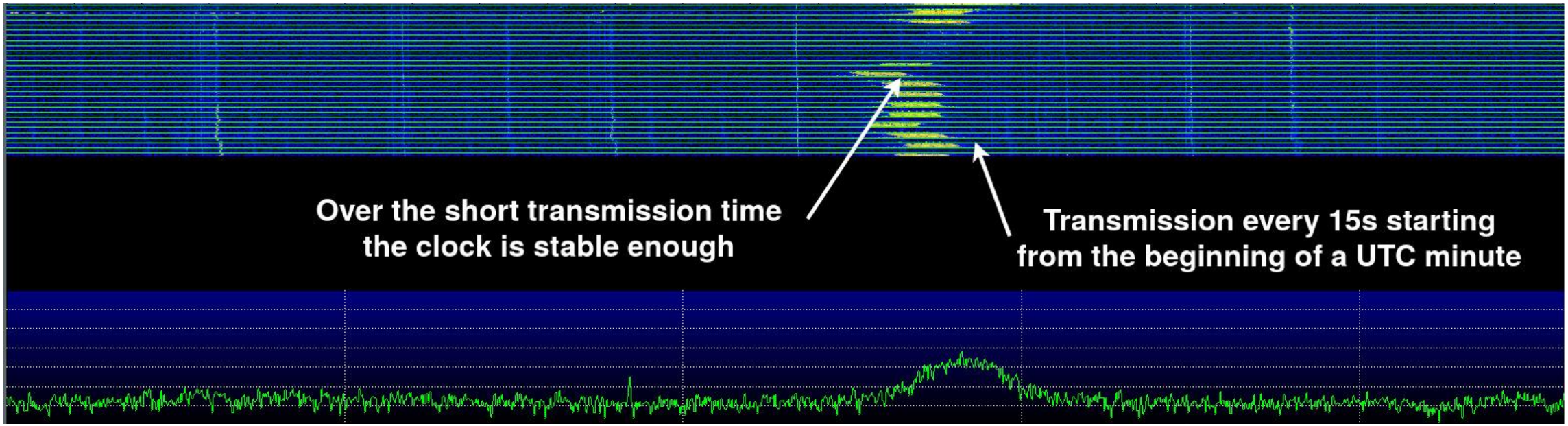
E.g., fast short-range: 3xPSK (e.g., MIPS32 660wpm)

E.g., flexible receivers: Handheld or SDR radio + FlDigi on smartphone

Almost zero implementation effort, not much knowledge/skills required

EURECOM
Sophia Antipolis

Examples of Tracking

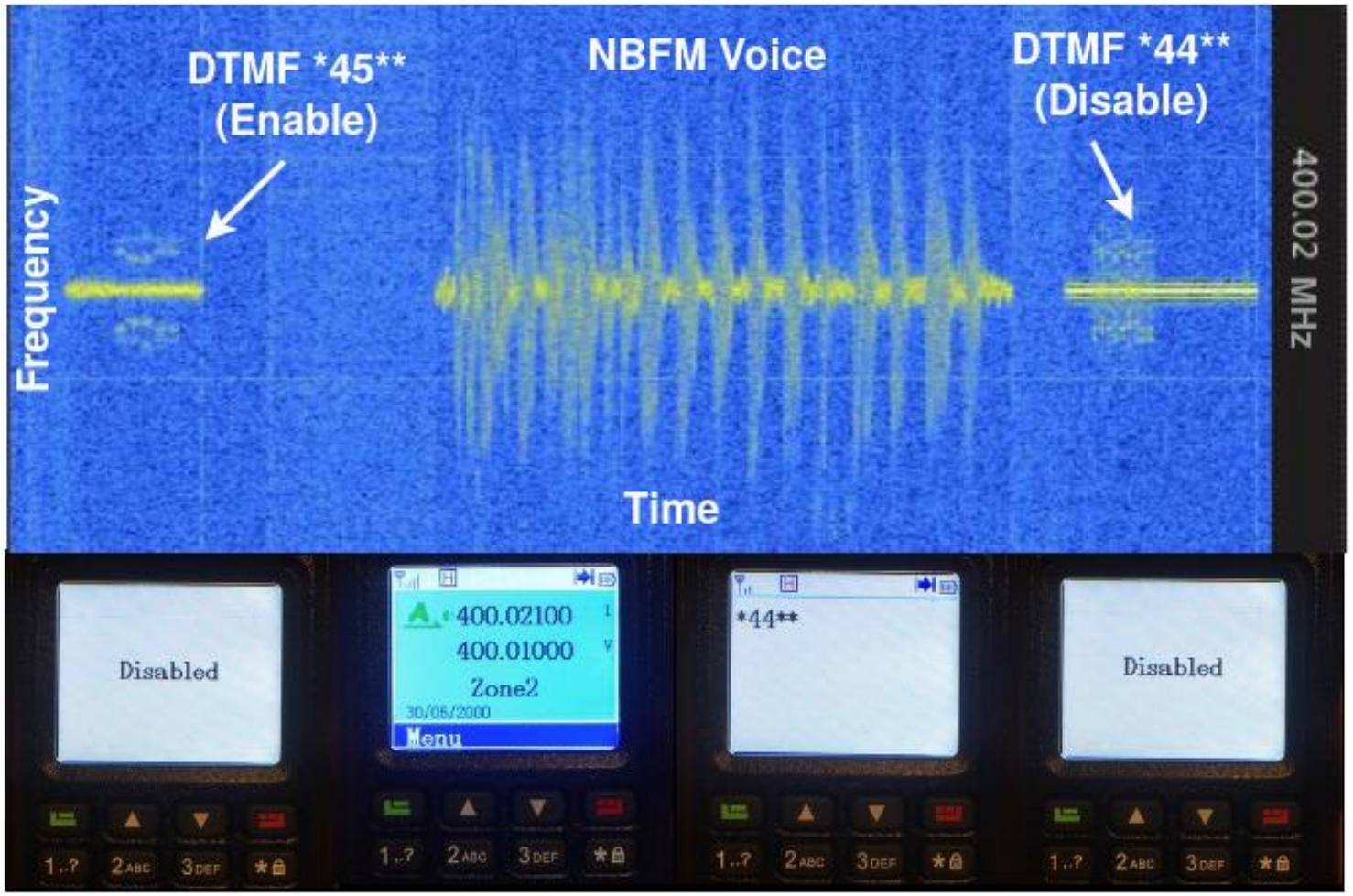


Tracking using FT4 beacons, up to 5m on Galaxy S5 Mini
Using existing reception tools

J. Taylor, FT4, https://physics.princeton.edu/pulsar/k1jt/FT4_Protocol.pdf.

J. Taylor, WSJT, https://physics.princeton.edu/pulsar/K1JT/.

Examples of Injection



IoT to UHF radio injection, a few meters

EURECOM
Sophia Antipolis

Countermeasures

Soft-TEMPEST-specific (HW)
Reduce leakages and coupling

Soft-TEMPEST-specific (SW)
Reduce timing resolution and software control on hardware

Applications specific (SW/HW):
Shield smartphone, spoofing detection, …

Future work and conclusion

Future work

Optimizations

Time resolution, other types of one-bit coding, ...

Future work

Optimizations
Time resolution, other types
of one-bit coding, ...

Other sources / languages
JavaScript, WebAssembly, GPU, …
(some preliminary results)

Future work

Optimizations
Time resolution, other types
of one-bit coding, ...

Other sources / languages
JavaScript, WebAssembly, GPU, …
(some preliminary results)

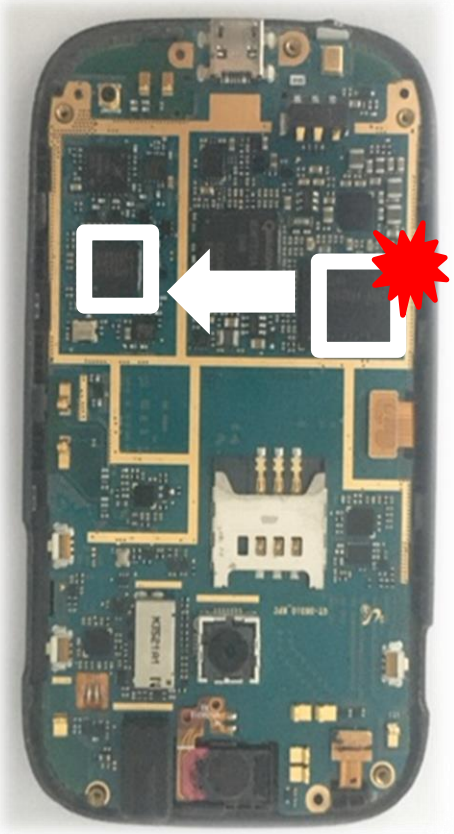
Spoofing and jamming
Radios, sensors, on the same device
(Radio Frequency Interference) ...

Future work

Optimizations
Time resolution, other types of one-bit coding, ...

Other sources / languages
JavaScript, WebAssembly, GPU, …
(some preliminary results)

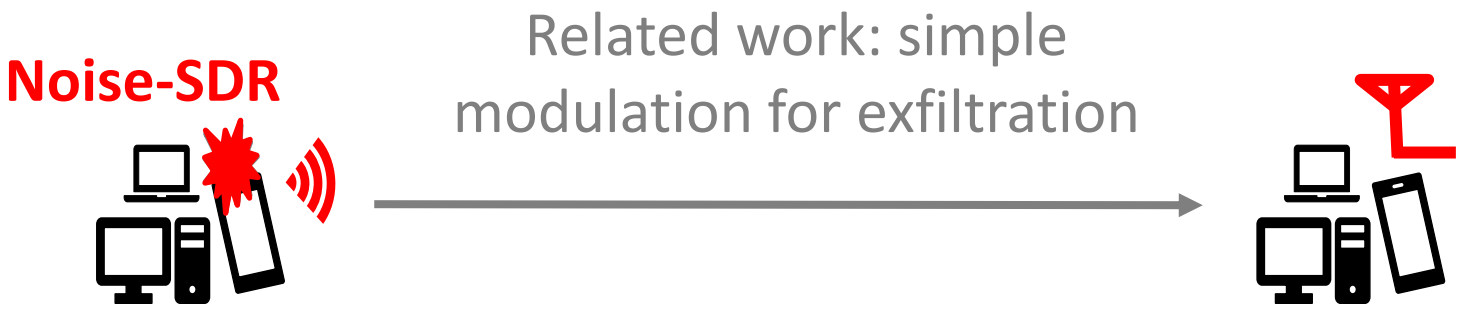
Spoofing and jamming
Radios, sensors, on the same device (Radio Frequency Interference) ...



Noise-SDR
+ RF Interference

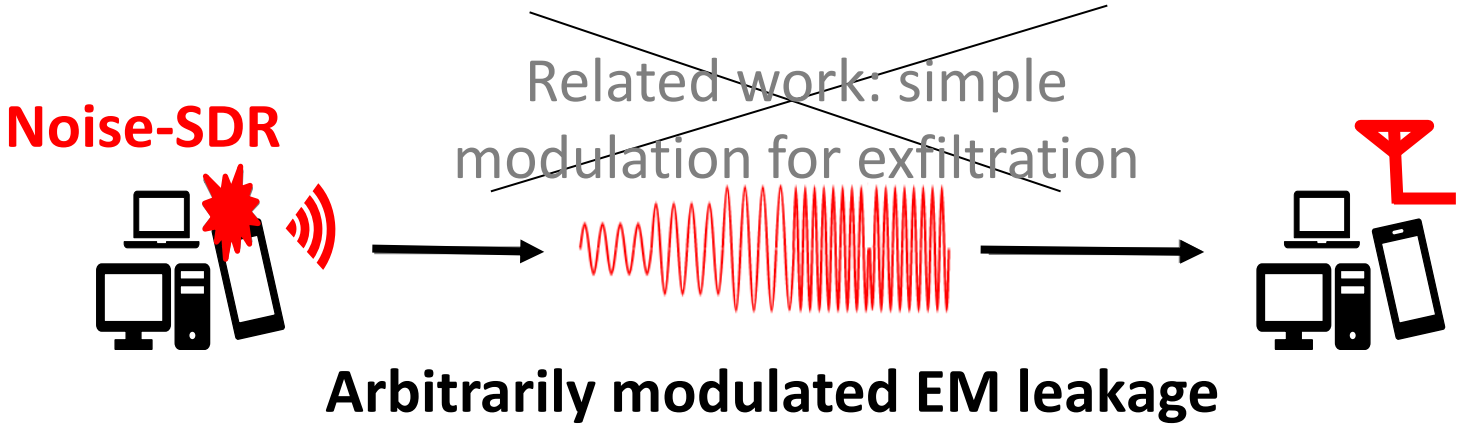
https://github.com/eurecom-s3/noise-sdr

Noise-SDR: Arbitrary Modulation of Electromagnetic Noise from Unprivileged Software and Its Impact on Emission Security



https://github.com/eurecom-s3/noise-sdr

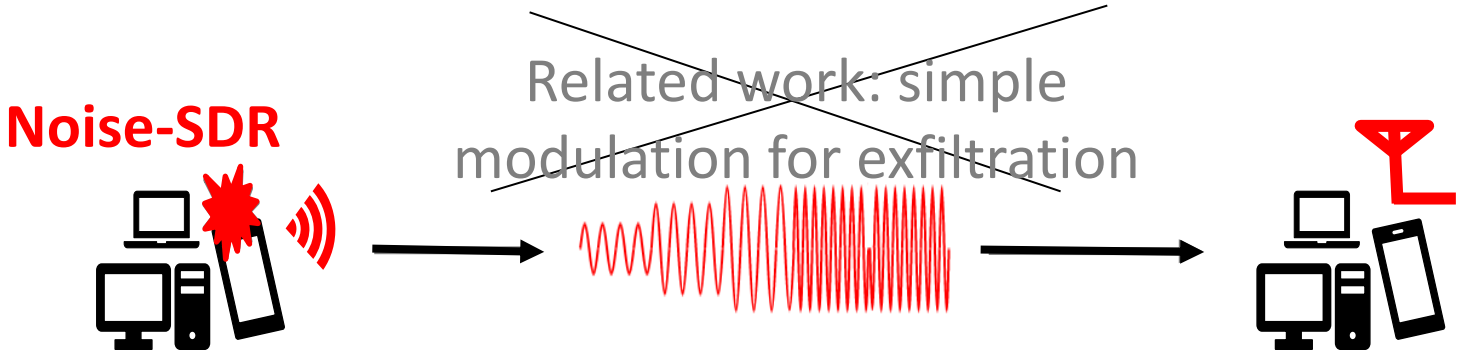
Noise-SDR: Arbitrary Modulation of Electromagnetic Noise from Unprivileged Software and Its Impact on Emission Security



EURECOM
Sophia Antipolis

https://github.com/eurecom-s3/noise-sdr

Noise-SDR: Arbitrary Modulation of Electromagnetic Noise from Unprivileged Software and Its Impact on Emission Security

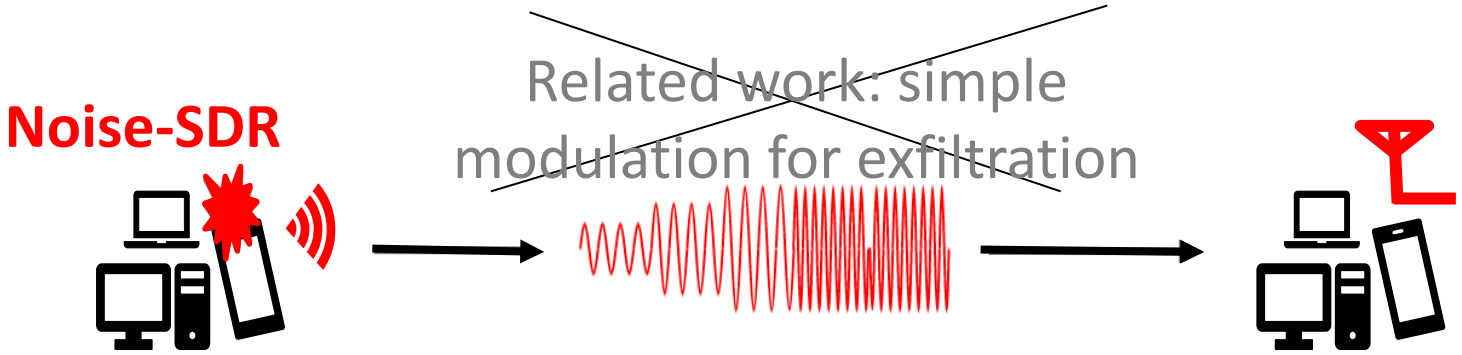


Arbitrarily modulated EM leakage

How: DRAM accesses, pass-band one-bit coding, software-defined

https://github.com/eurecom-s3/noise-sdr

Noise-SDR: Arbitrary Modulation of Electromagnetic Noise from Unprivileged Software and Its Impact on Emission Security



Arbitrarily modulated EM leakage

How: DRAM accesses, pass-band one-bit coding, software-defined

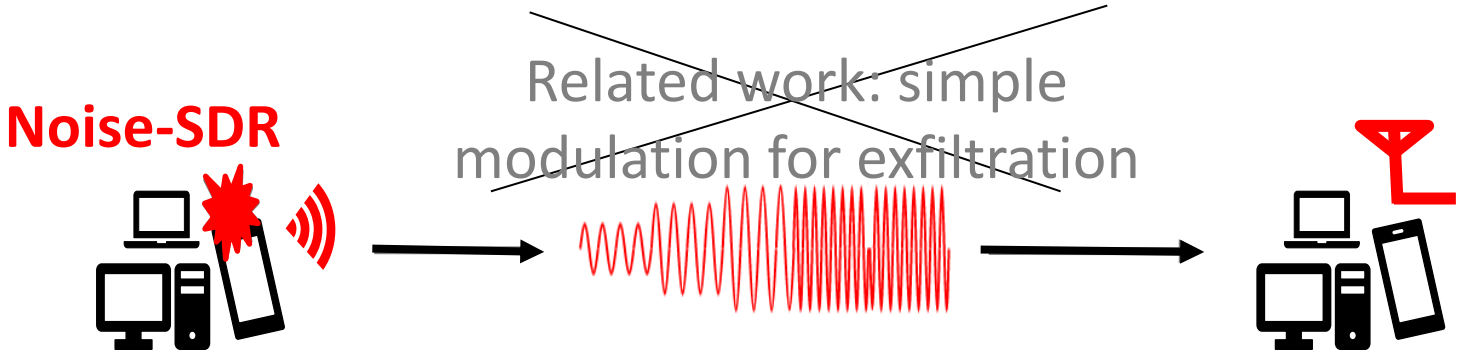
Pros: flexibility, performance, reuse of existing protocols

Cons: limited bandwidth, center frequency

EURECOM Sophia Antipolis

https://github.com/eurecom-s3/noise-sdr

Noise-SDR: Arbitrary Modulation of Electromagnetic Noise from Unprivileged Software and Its Impact on Emission Security



Arbitrarily modulated EM leakage

How: DRAM accesses, pass-band one-bit coding, software-defined

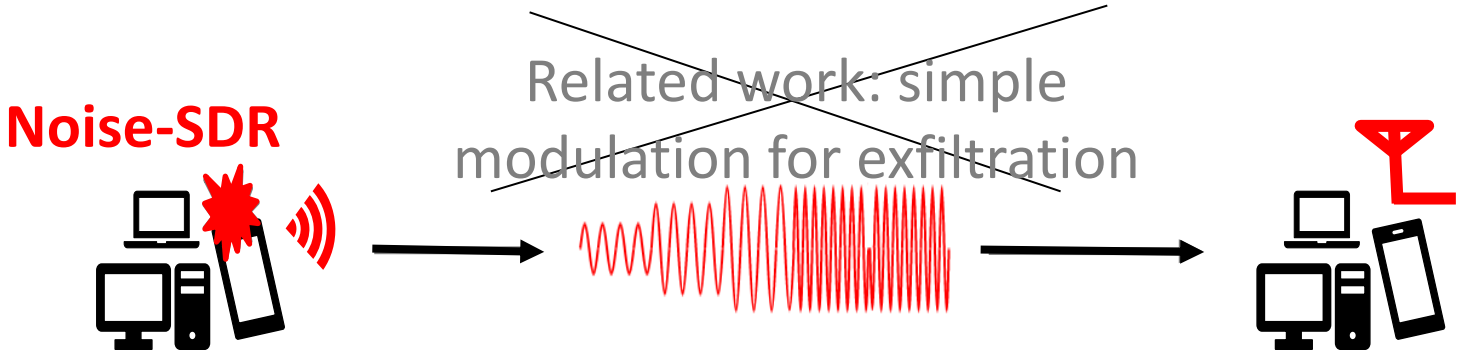
Pros: flexibility, performance, reuse of existing protocols

Cons: limited bandwidth, center frequency

Implementation: ArmV7A, ArmV8A, x86, MIPS

https://github.com/eurecom-s3/noise-sdr

Noise-SDR: Arbitrary Modulation of Electromagnetic Noise from Unprivileged Software and Its Impact on Emission Security



Arbitrarily modulated EM leakage

How: DRAM accesses, pass-band one-bit coding, software-defined

Pros: flexibility, performance, reuse of existing protocols

Cons: limited bandwidth, center frequency

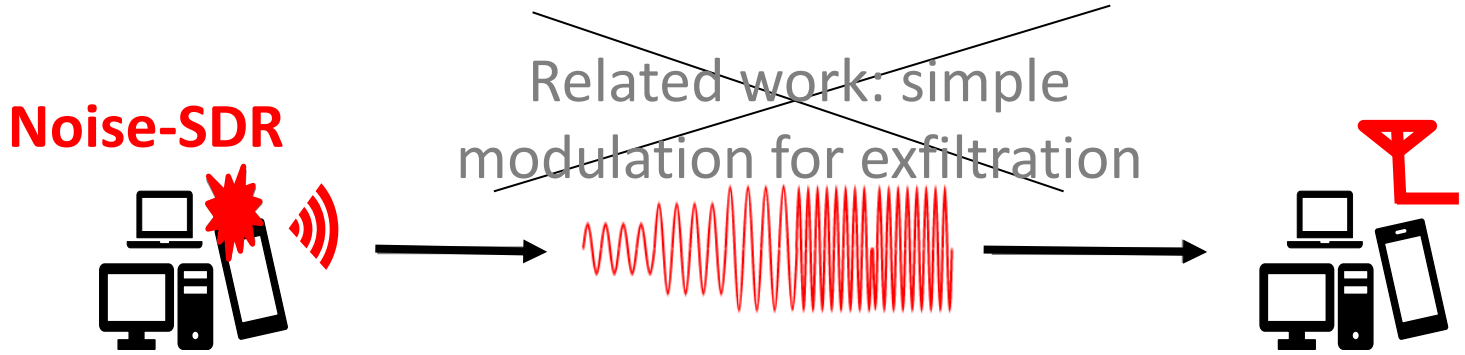
Implementation: ArmV7A, ArmV8A, x86, MIPS

Security impact: exfiltration, tracking, injection, …

Open source

https://github.com/eurecom-s3/noise-sdr

Noise-SDR: Arbitrary Modulation of Electromagnetic Noise from Unprivileged Software and Its Impact on Emission Security



Arbitrarily modulated EM leakage

How: DRAM accesses, pass-band one-bit coding, software-defined

Pros: flexibility, performance, reuse of existing protocols

Cons: limited bandwidth, center frequency

Implementation: ArmV7A, ArmV8A, x86, MIPS

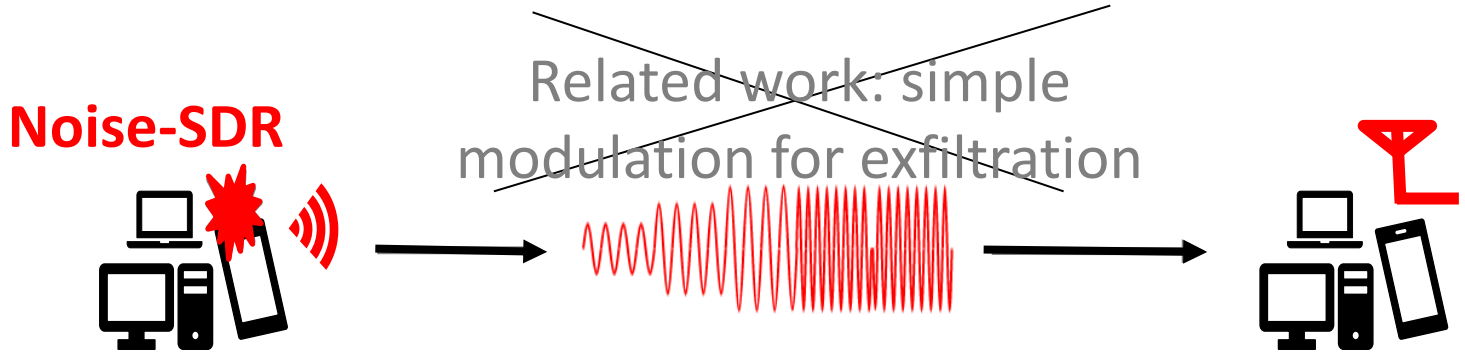
Security impact: exfiltration, tracking, injection, …

Open source

https://github.com/eurecom-s3/noise-sdr

Noise-SDR: Arbitrary Modulation of Electromagnetic Noise from Unprivileged Software and Its Impact on Emission Security

Thank you!
Questions?



Arbitrarily modulated EM leakage

How: DRAM accesses, pass-band one-bit coding, software-defined

Pros: flexibility, performance, reuse of existing protocols

Cons: limited bandwidth, center frequency

Implementation: ArmV7A, ArmV8A, x86, MIPS

Security impact: exfiltration, tracking, injection, …

EURECOM
Sophia Antipolis