On the Suitability of SCARR for Cross-Device Leakage Analysis

George Crane, **Dean Sullivan**

University of New Hampshire

Resilient Architecture Lab (RAL)

Outline

- Profiled Side Channel Analysis & Cross-Device SCA (3 minute crash course)
- Technical challenges of performing cross-device SCA
- The SCARR SCA framework
- Case Study: Extending SCARR for our use case

Goals for Audience

- See end to end the full process of leakage analysis on an embedded system
- Understand the key challenges faced by researchers performing SCA
- Gain familiarity with the SCARR tool, see if it can meet your challenges
- Learn to make an oracle for leakage analysis

Security in Embedded Devices

Microcontrollers are **everywhere**, including security critical applications

Security in Embedded Devices

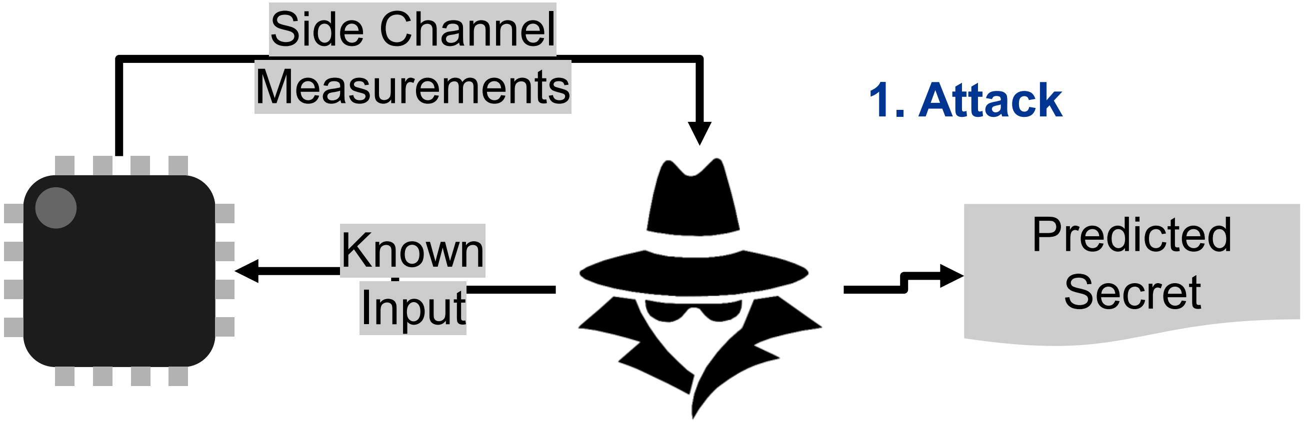
Microcontrollers are **everywhere**, including security critical applications

These applications rely on the device’s ability to securely perform cryptographic operations

- Authentication
- Encryption
- Integrity validation...

Security in Embedded Devices

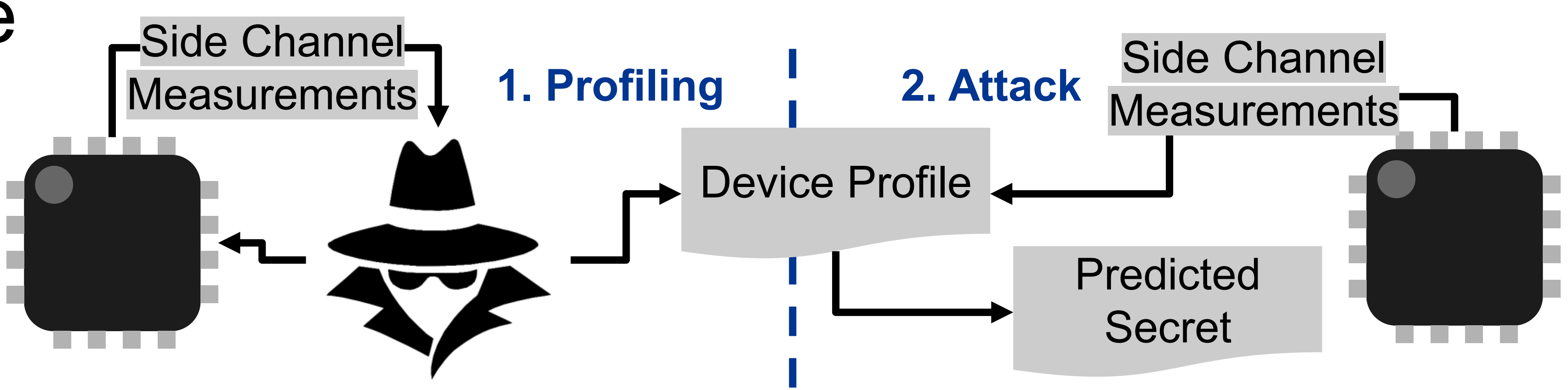
Side-Channels expose data, including secrets like crypto-keys, as it is processed by the device.



Security in Embedded Devices

Side-Channels expose data, including secrets like crypto-keys, as it is processed by the device.

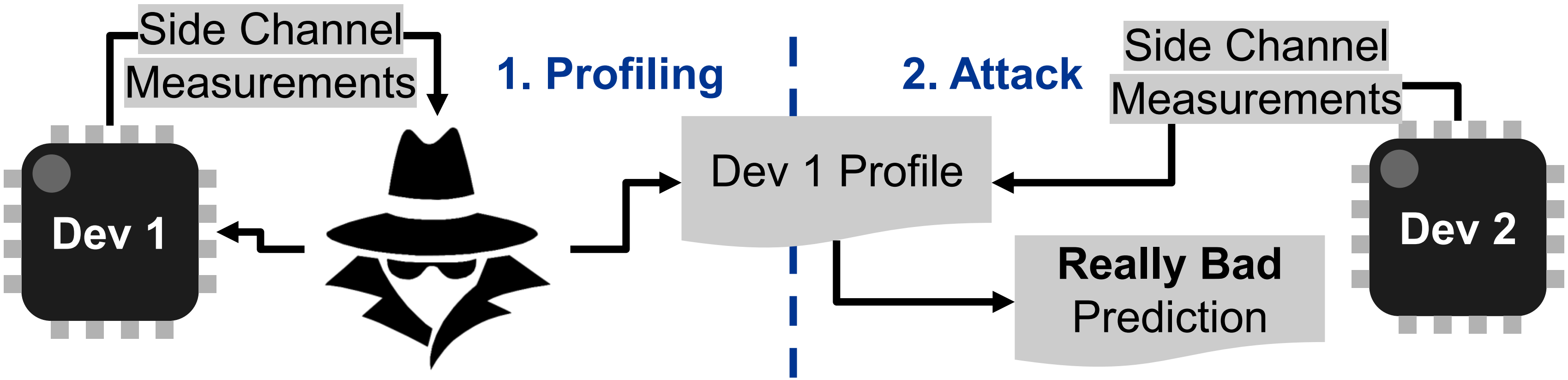
Profiled Side-Channel Analysis (SCA) constructs a statistical model (profile) of a device's leakage



Cross-Device SCA

Theory: Profiles can be applied across devices

Practice: Profiles can be applied across devices **with drastically reduced accuracy**



Cross-Device SCA

Theory: Profiles can be applied across devices

Practice: Profiles can be applied across devices **with drastically reduced accuracy**

Our research in this under-explored field aims to better explain and reduce this discrepency

Cross-Device SCA

We use **leakage quantification** techniques (**LQ**) to assess devices. LQ directly reveals where and how much a device leaks.

We aim to identify differences between identical devices, and develop a methodology to normalize for it during the profiling phase.

Cross-Device SCA

We use **leakage quantification** techniques (**LQ**) to assess devices. LQ directly reveals where and how much a device leaks.

However… Identifying **all** leakage is more difficult than exploiting **some** leakage

- BIG datasets

Enter: SCARR

SCARR is an open source SCA framwork implemented in python

Enter: SCARR

SCARR is an open source SCA framwork implemented in python

- Supports a number of common **LQ** techniques (e.g. SNR, NICV, etc.)
- Streaming algorithms to support out-of-core computing and some parallelism
- Doesn’t natively support some desired functionality, but is extendable

SCARR Example

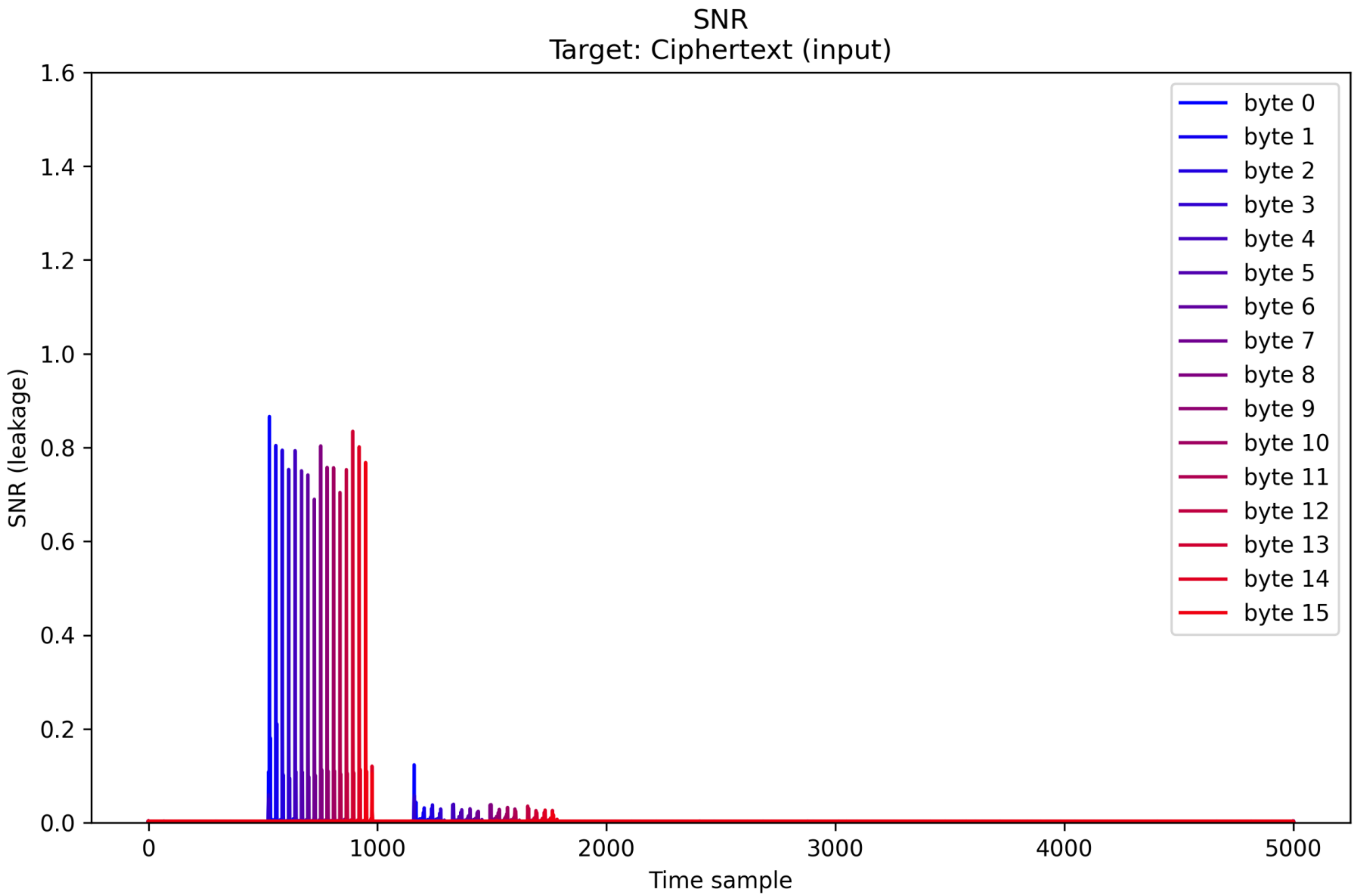
```
# Load the dataset

dataset = TraceHandler(fileName="SAM4S 100000 random Traces
[50c].zarr", batchSize=5000)
```

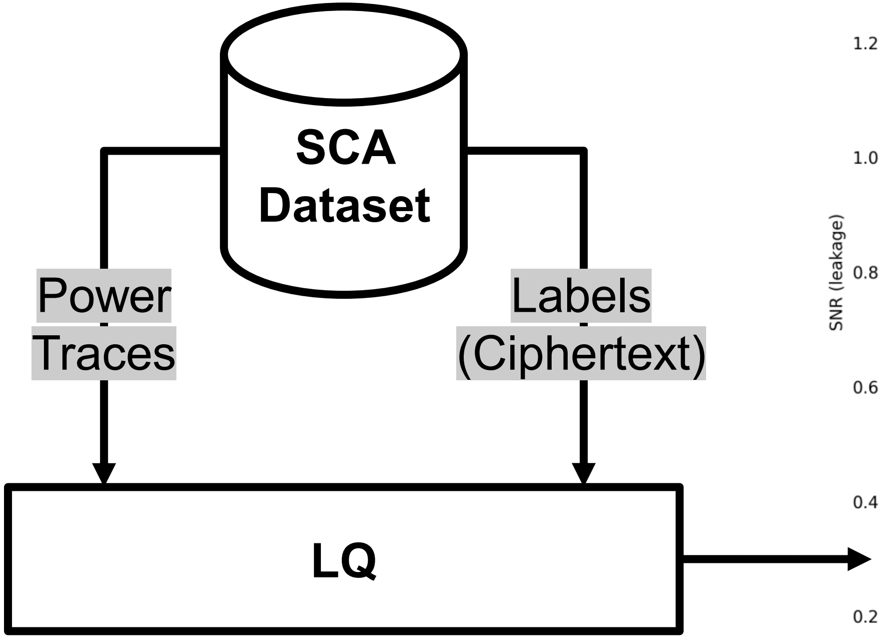
SCARR Example

```
# Load the dataset

dataset = TraceHandler(fileName="SAM4S 100000 random Traces
[50c].zarr", batchSize=5000)


# We will calculate SNR based on value of input ciphertext

engine = SNR(model_value=CipherText())

positions=[i for i in range(16)]  # Target all 16 bytes of text

container = Container(options=ContainerOptions(engine=engine,
handler=dataset), model_positions=positions)
```

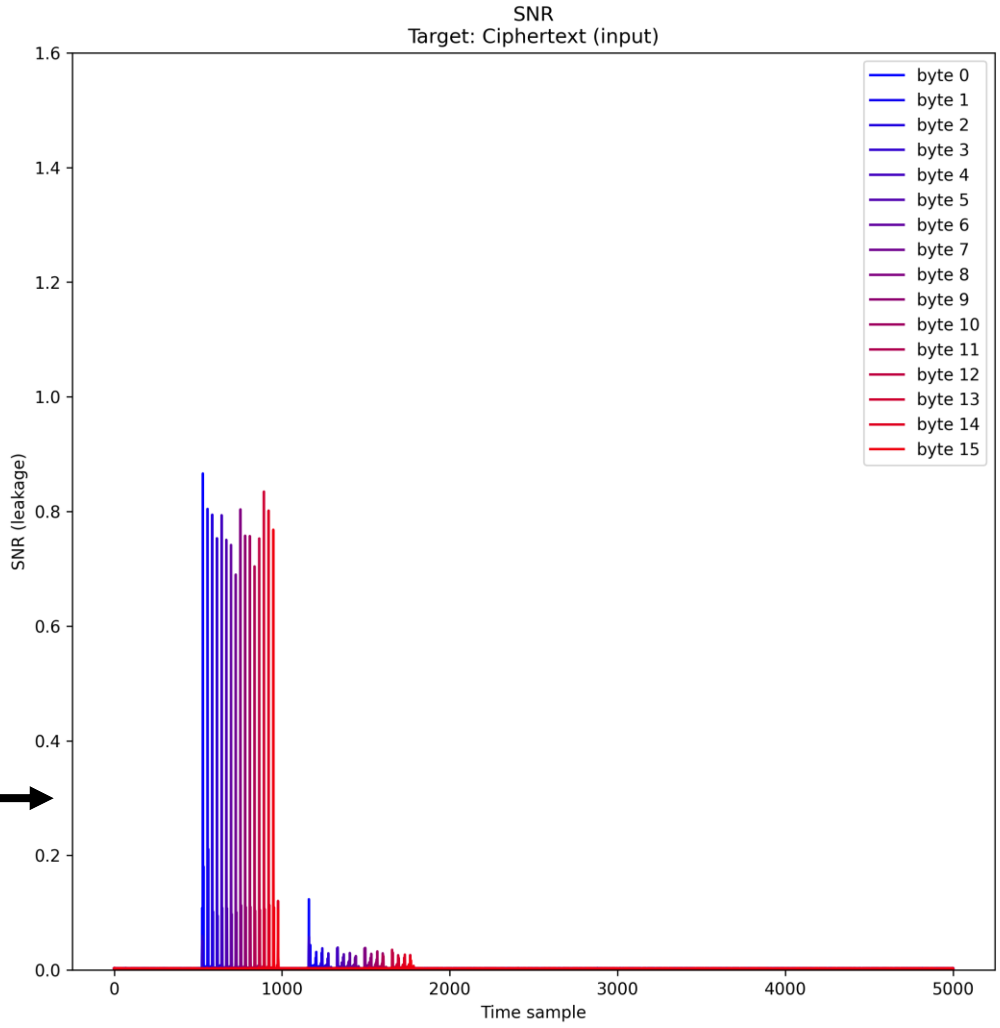
SCARR Example

```
# Load the dataset

dataset = TraceHandler(fileName="SAM4S 100000 random Traces
[50c].zarr", batchSize=5000)


# We will calculate SNR based on value of input ciphertext

engine = SNR(model_value=CipherText())

positions=[i for i in range(16)]  # Target all 16 bytes of text

container = Container(options=ContainerOptions(engine=engine,
handler=dataset), model_positions=positions)


container.run()

results = container.get_result()
```

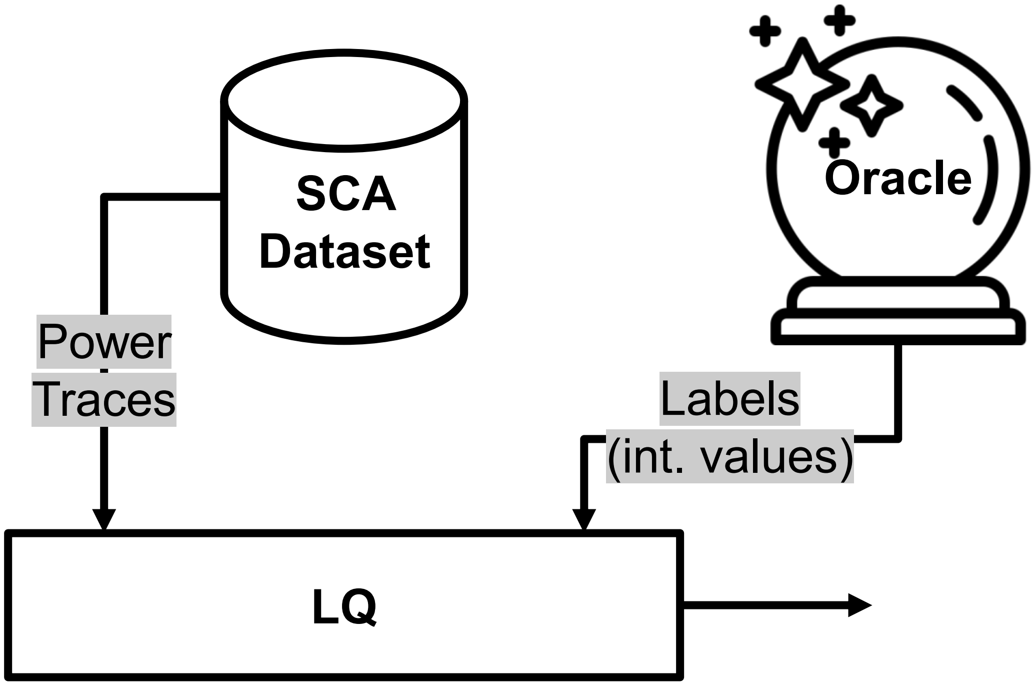

SNR
Target: Ciphertext (input)
SNR (leakage)
Time sample
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
1000
2000
3000
4000
5000
byte 0
byte 1
byte 2
byte 3
byte 4
byte 5
byte 6
byte 7
byte 8
byte 9
byte 10
byte 11
byte 12
byte 13
byte 14
byte 15



Our example analysis looks like this

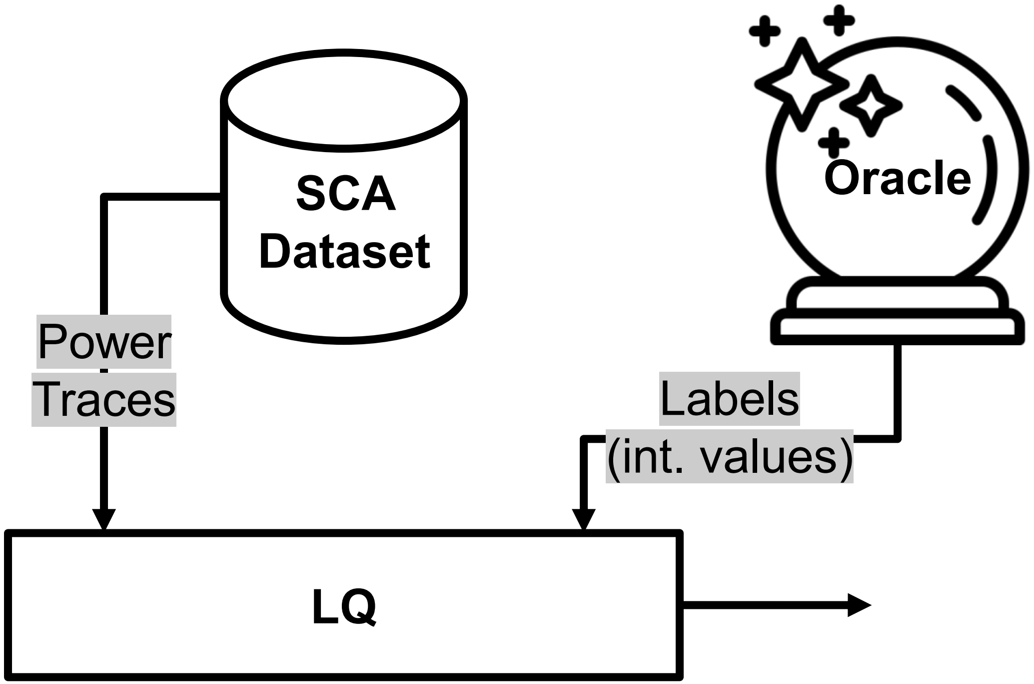


The Oracle



We want to be able to do this, for any intermediate value

The Oracle

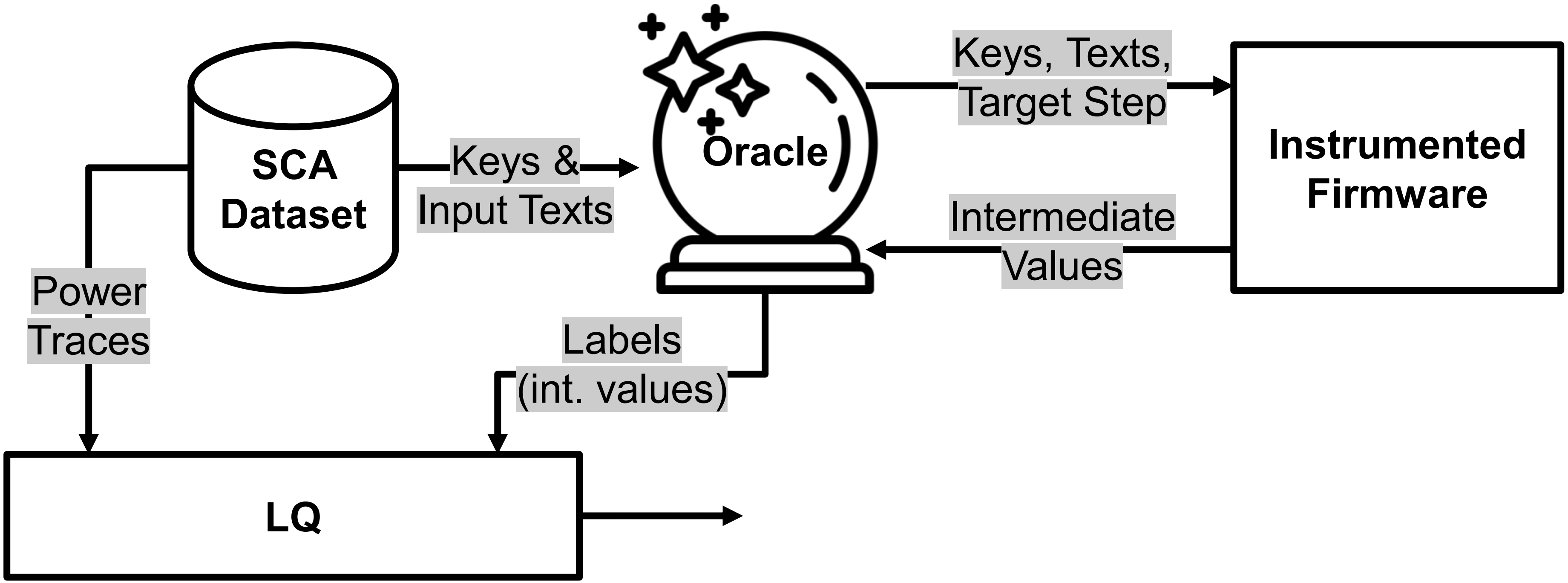


We want to be able to do this, for any intermediate value

SCARR doesn't support this natively, so its up to us to implement.

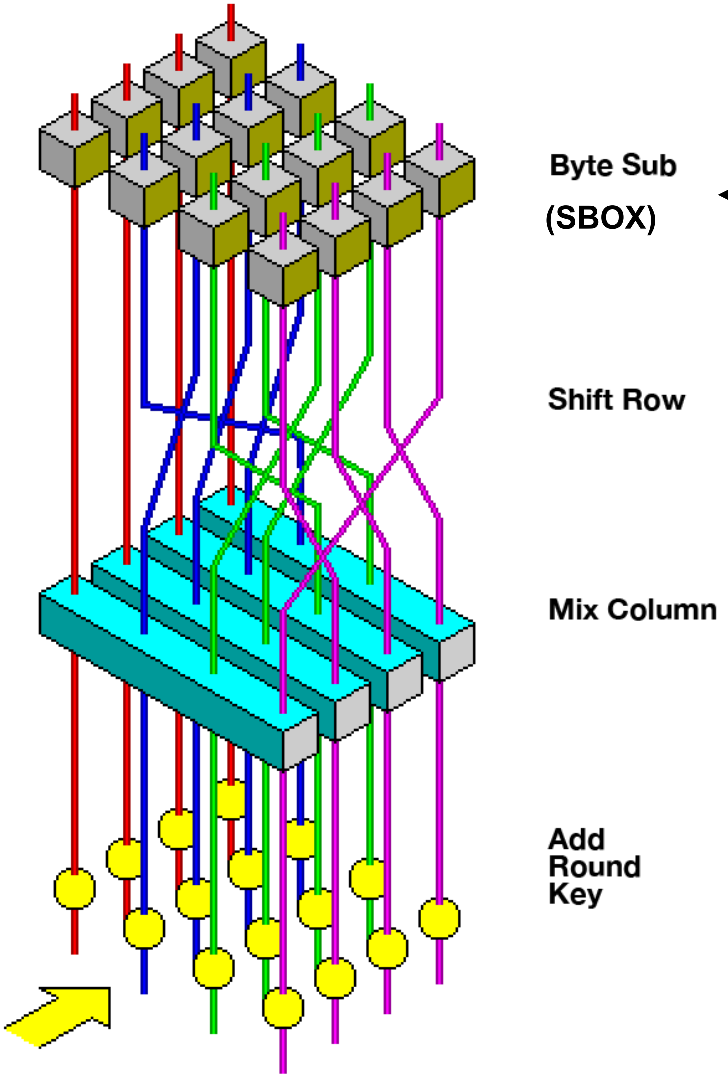
How can we achieve this without making extra work for ourselves?

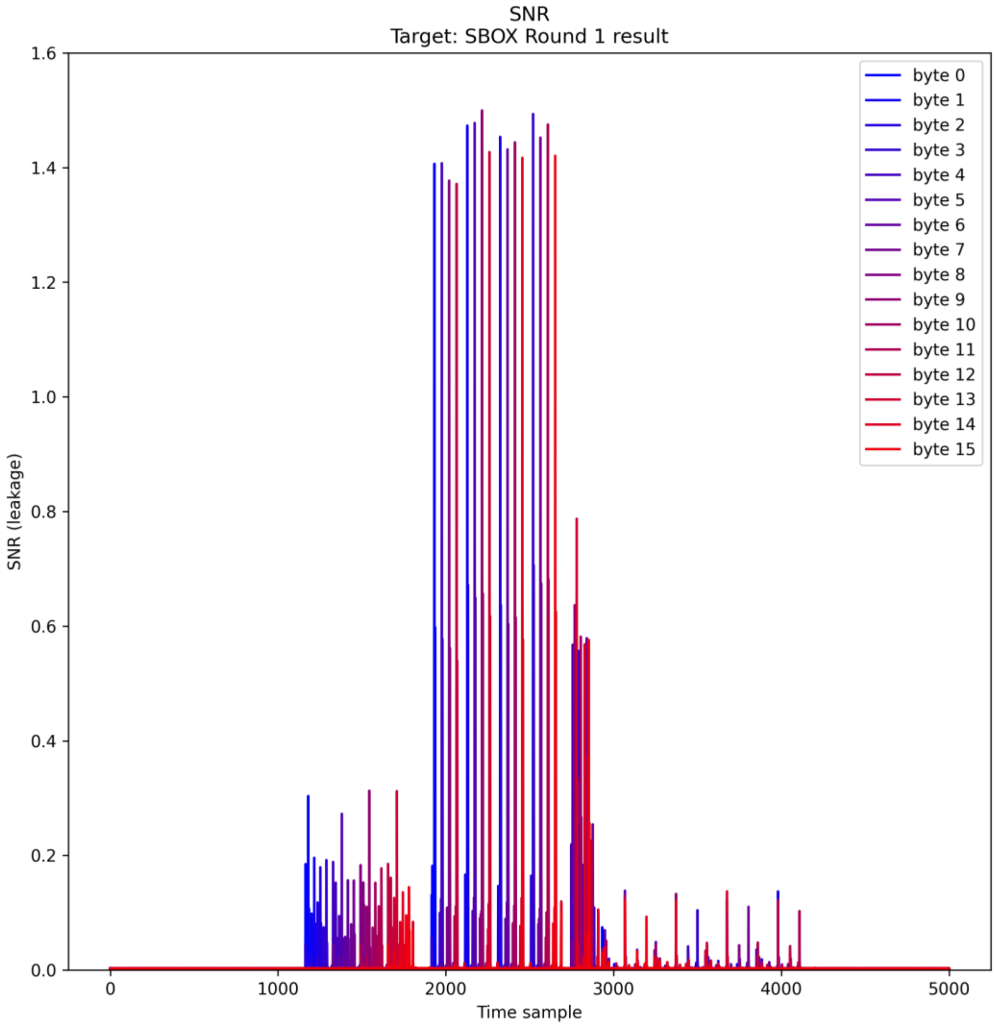
The Oracle: Implemented

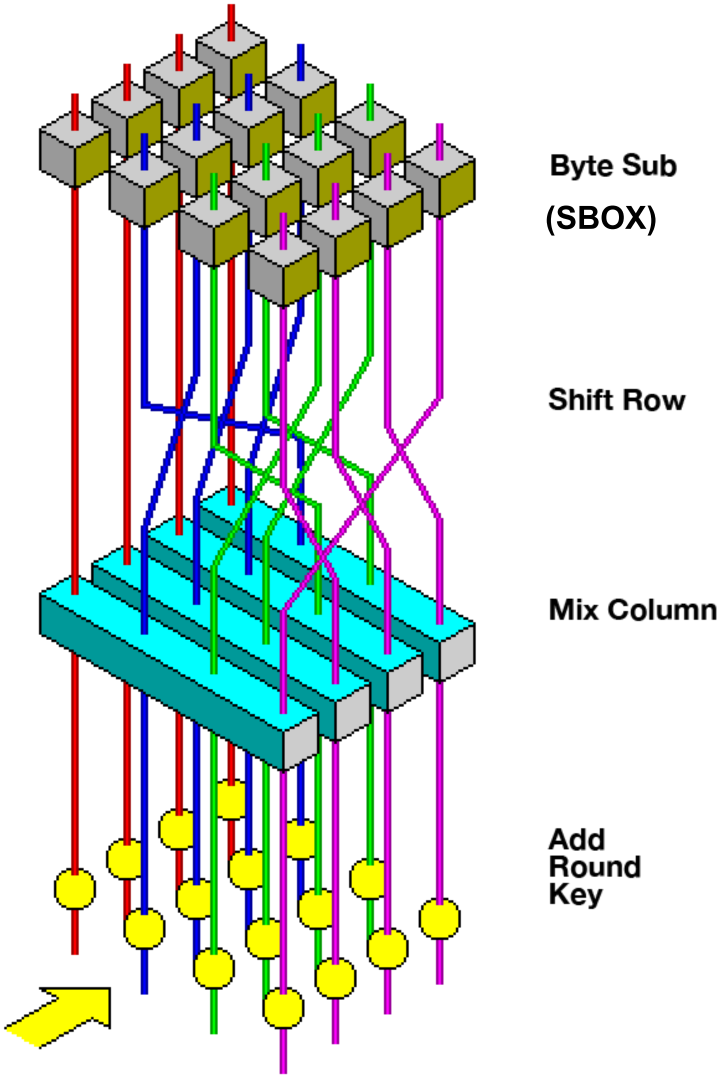


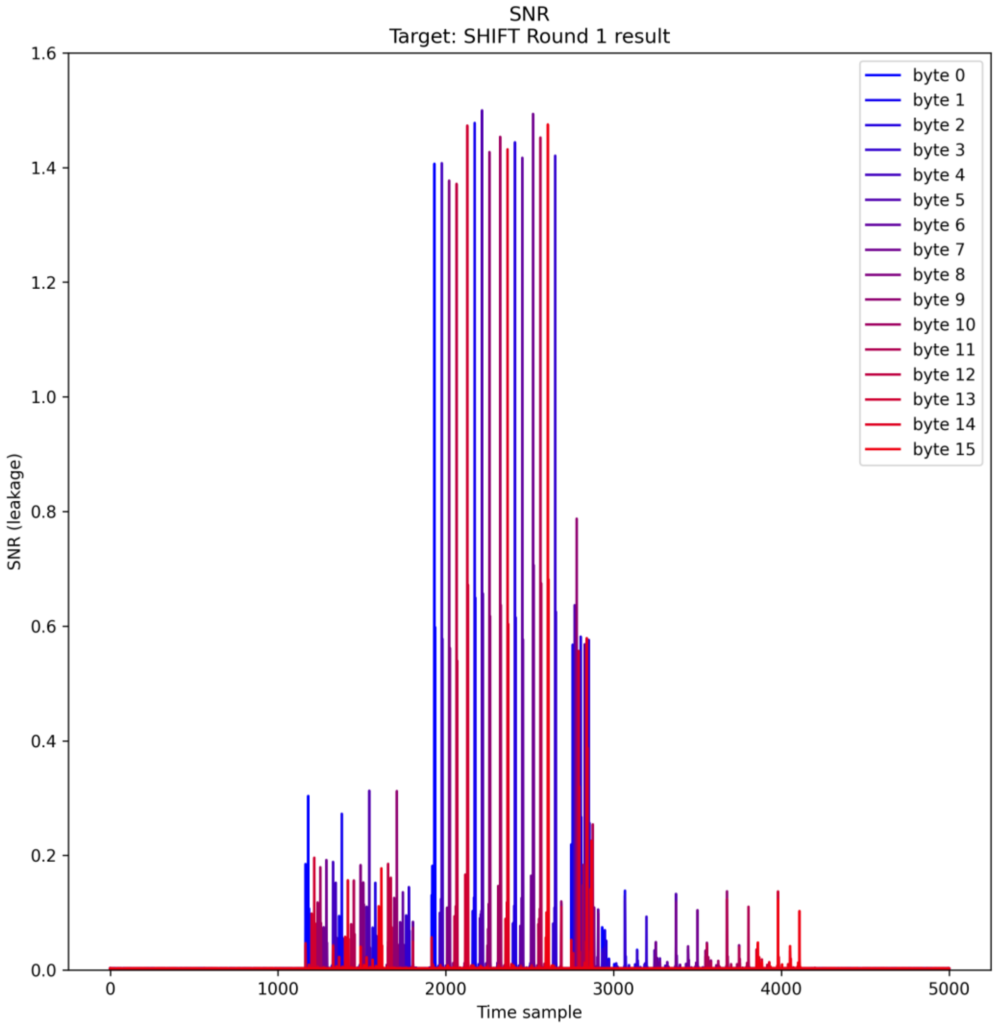
The Oracle: Implemented

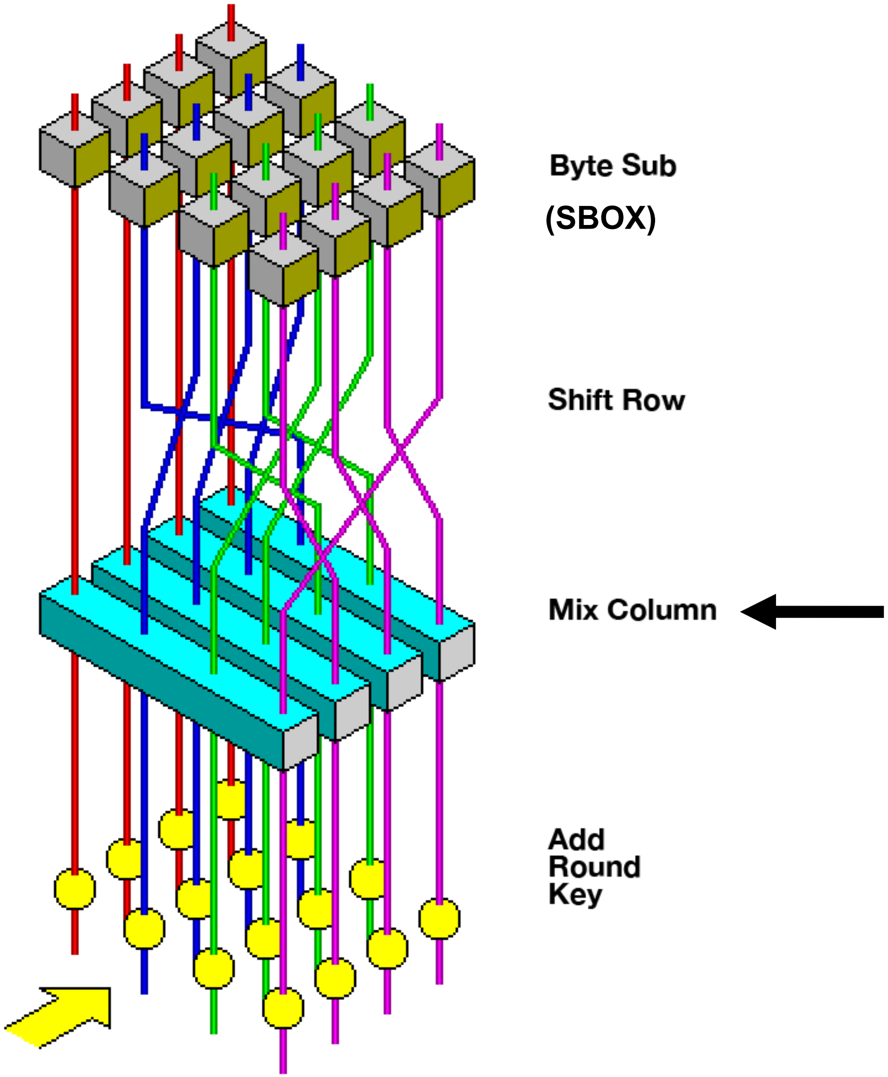
```
# Instead of writing a new ModelValue class for every operation of every cipher we want to target

engine = SNR(model_value=AES256Round1Sbox())

# We instrument the cipher firmware once and get all that for free!

engine = SNR(model_value=AESOracle(

    type=256, round=1, step="SBOX"

))
```

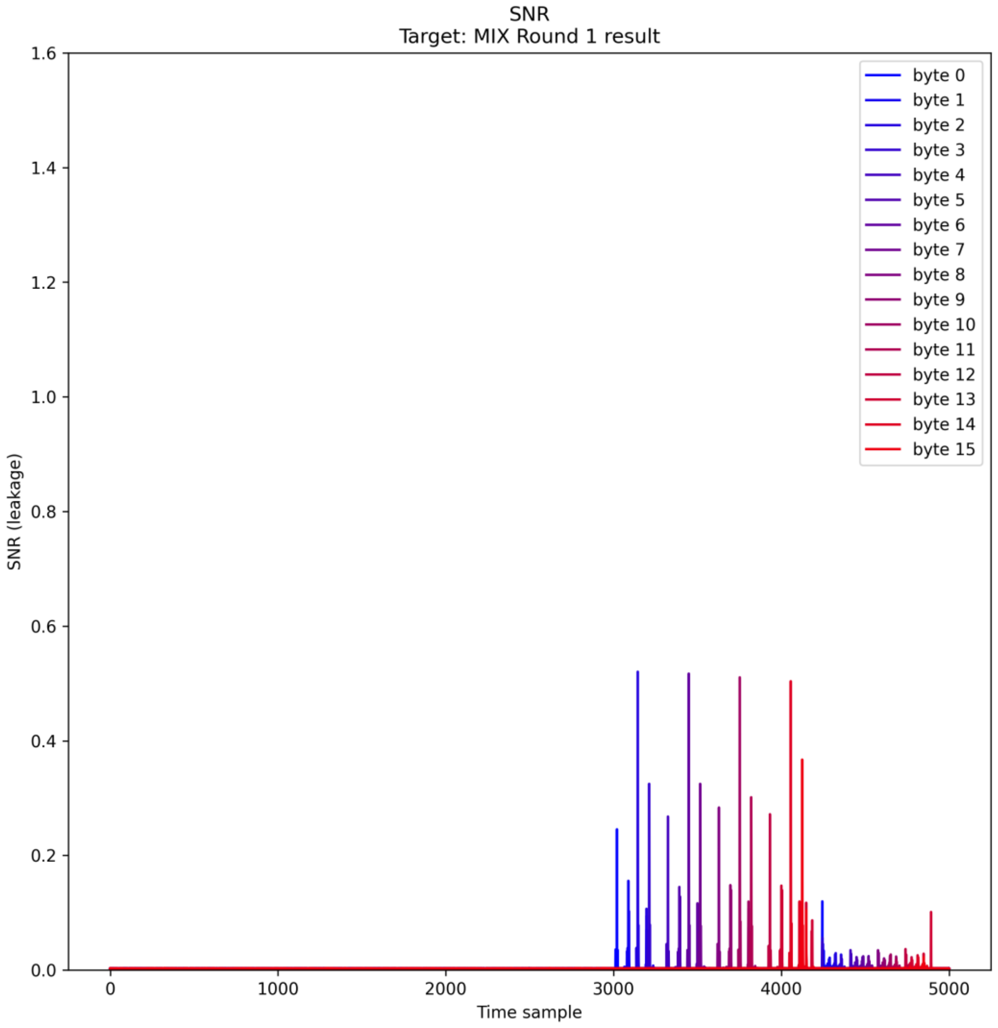

Byte Sub
(SBOX)
Shift Row
Mix Column
Add
Round
Key

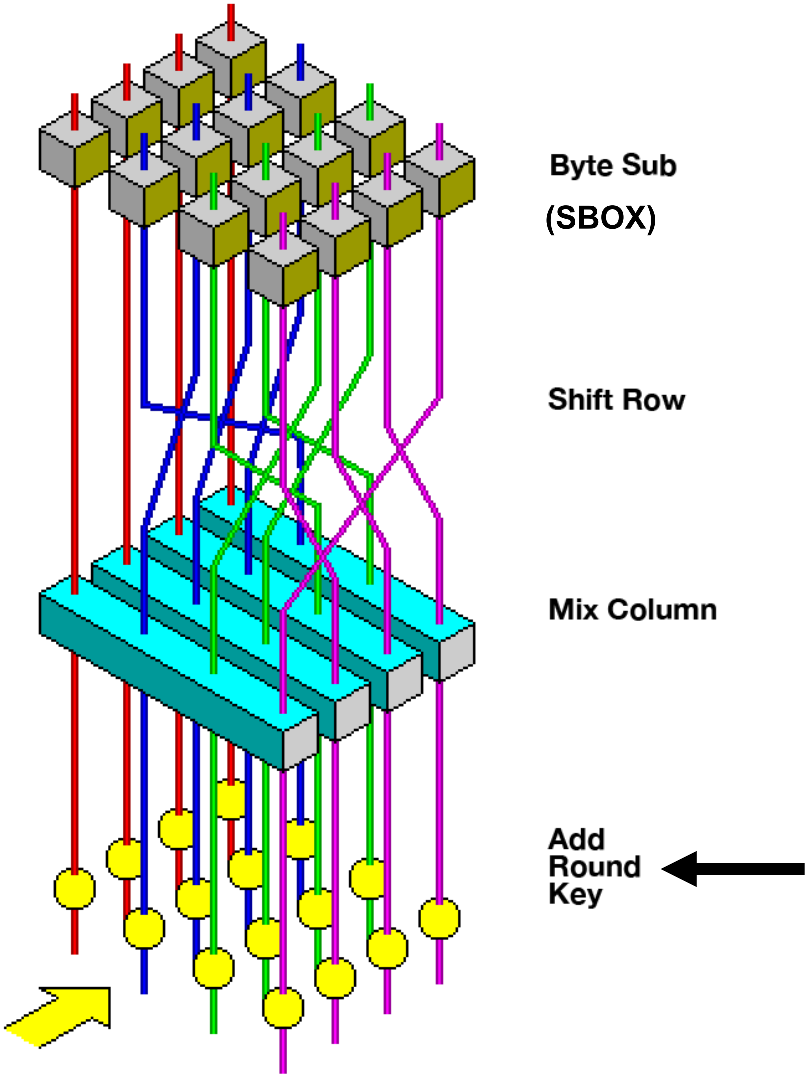

SNR
Target: SBOX Round 1 result
SNR (leakage)
Time sample
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
1000
2000
3000
4000
5000
byte 0
byte 1
byte 2
byte 3
byte 4
byte 5
byte 6
byte 7
byte 8
byte 9
byte 10
byte 11
byte 12
byte 13
byte 14
byte 15

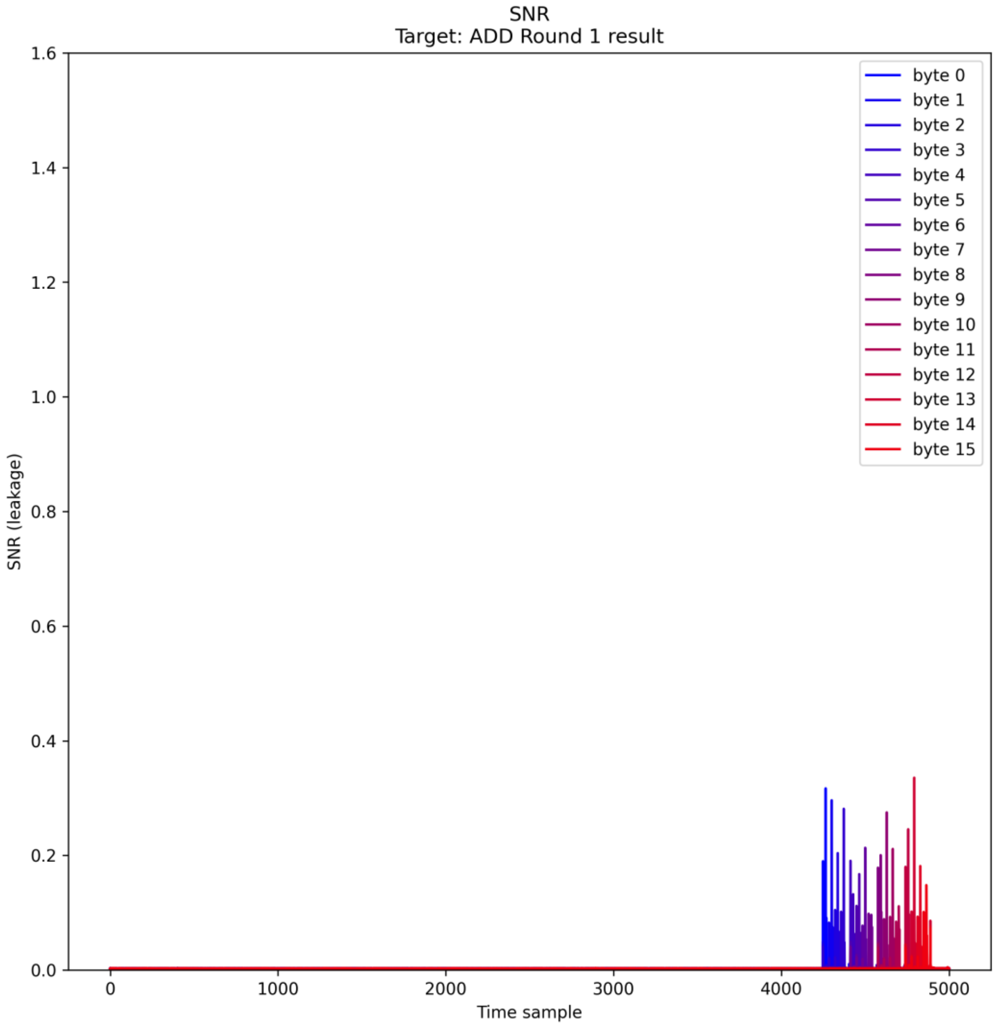

Byte Sub
(SBOX)
Shift Row
Mix Column
Add
Round
Key


SNR
Target: SHIFT Round 1 result
SNR (leakage)
Time sample
byte 0
byte 1
byte 2
byte 3
byte 4
byte 5
byte 6
byte 7
byte 8
byte 9
byte 10
byte 11
byte 12
byte 13
byte 14
byte 15


Byte Sub
(SBOX)
Shift Row
Mix Column
Add
Round
Key


SNR
Target: MIX Round 1 result
SNR (leakage)
Time sample
byte 0
byte 1
byte 2
byte 3
byte 4
byte 5
byte 6
byte 7
byte 8
byte 9
byte 10
byte 11
byte 12
byte 13
byte 14
byte 15
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
1000
2000
3000
4000
5000


Byte Sub
(SBOX)
Shift Row
Mix Column
Add
Round
Key


SNR
Target: ADD Round 1 result
SNR (leakage)
Time sample
byte 0
byte 1
byte 2
byte 3
byte 4
byte 5
byte 6
byte 7
byte 8
byte 9
byte 10
byte 11
byte 12
byte 13
byte 14
byte 15
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
1000
2000
3000
4000
5000

Extending SCARR for Cross-Device Leakage Analysis

How can our new oracle be integrated into SCARR?

Extending SCARR for Cross-Device Leakage Analysis

How can our new oracle be integrated into SCARR? It can’t

Extending SCARR for Cross-Device Leakage Analysis

How can our new oracle be integrated into SCARR? It can't*

*Without extensive modification to the internals of the framework.

SCARR Limitations for Cross-Device Leakage Analysis

- All data is internally managed by SCARR. If an analysis requires a different dataflow, it will be a problem.
- SCARR dataset format forces storage of redundant data, which could otherwise be calculated on demand with an oracle.

Our Homemade Solution

```
dset = ZarrHandler("SAM4S 100000 random Traces [50c].zarr",
chunks=10000)
```

- Datasets are loaded lazily to minimize memory overhead, but can be accessed arbitrarily like an array.

Our Homemade Solution

```
dset = ZarrHandler("SAM4S 100000 random Traces [50c].zarr",
chunks=10000)

oracle = OracleAES(
    aes_type=128, lazy=False,
    keys=dset.get(“key”), texts=dset.get(“ciphertext”),
    round=1, step=”SBOX” )
```

- All operations are self-contained, and can be combined into pipelines like those in the SCARR backend

Our Homemade Solution

```
dset = ZarrHandler("SAM4S 100000 random Traces [50c].zarr",
chunks=10000)


oracle = OracleAES(

    aes_type=128, lazy=False,

    keys=dset.get(“key”), texts=dset.get(“ciphertext”),

    round=1, step=”SBOX” )


snr = SNR(traces=dset.get(“traces”), labels=oracle.result)
```

Our Homemade Solution

Advantages:

- Complexities of streaming algorithms are mostly handled for us
- Enables simultaneous & interdependent analysis of multiple datasets
- Doesn’t require an arbitrary dataset format

Thank You!

Our analysis code and oracle proof of concept are publicly available, check it out!

https://github.com/cooc1501/OPTIMIST-25