SILK: high level of abstraction leakage simulator for side channel analysis

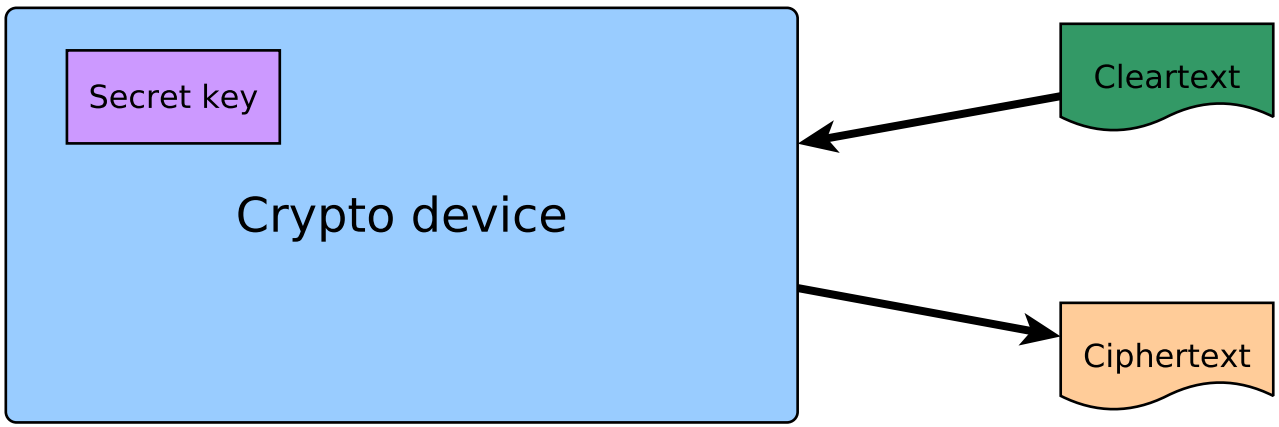
Nikita Veshchikov

QualSec. Université Libre de Bruxelles, Belgium

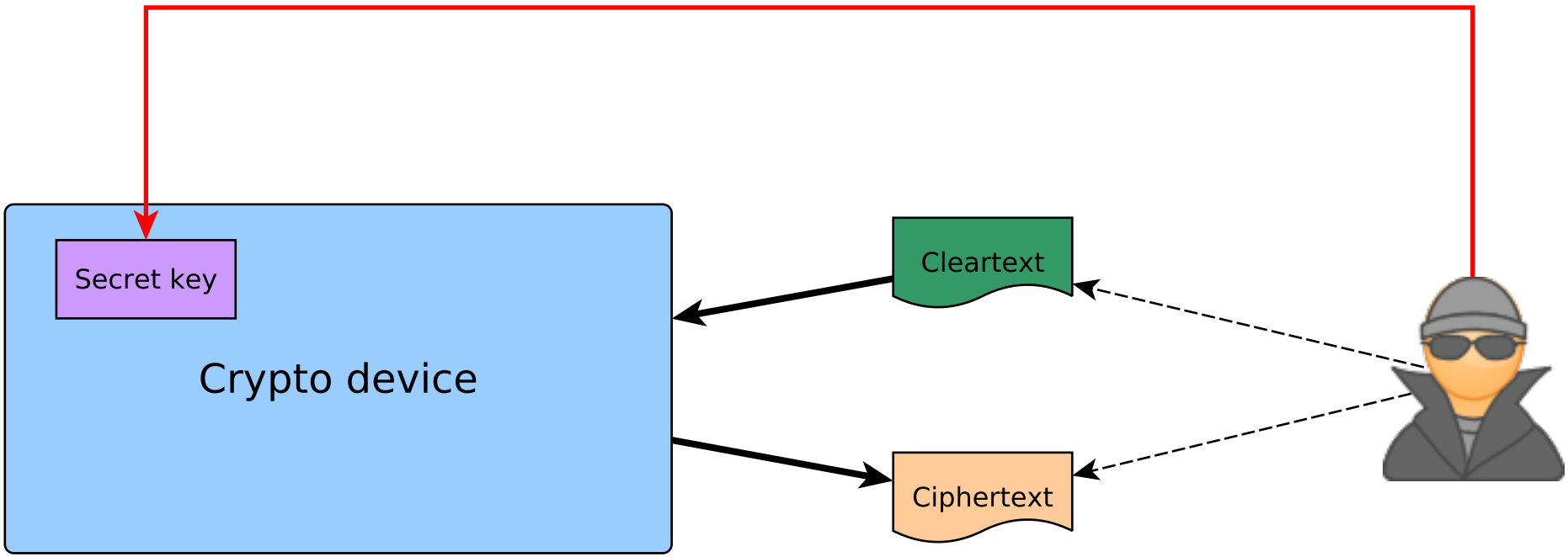
PPREW-4, New Orleans
9 December 2014

Intro & Motivations

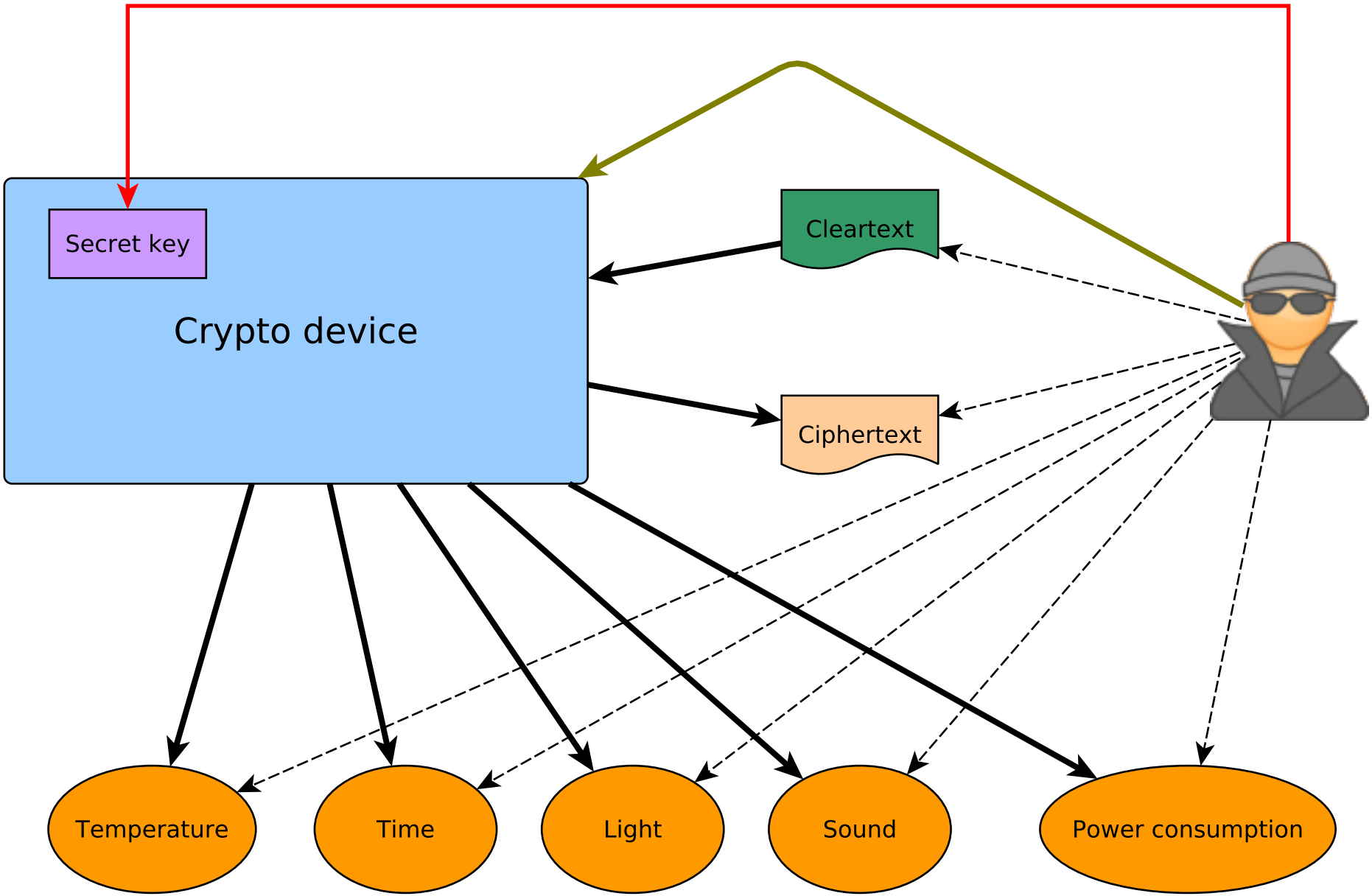
Cryptographic device



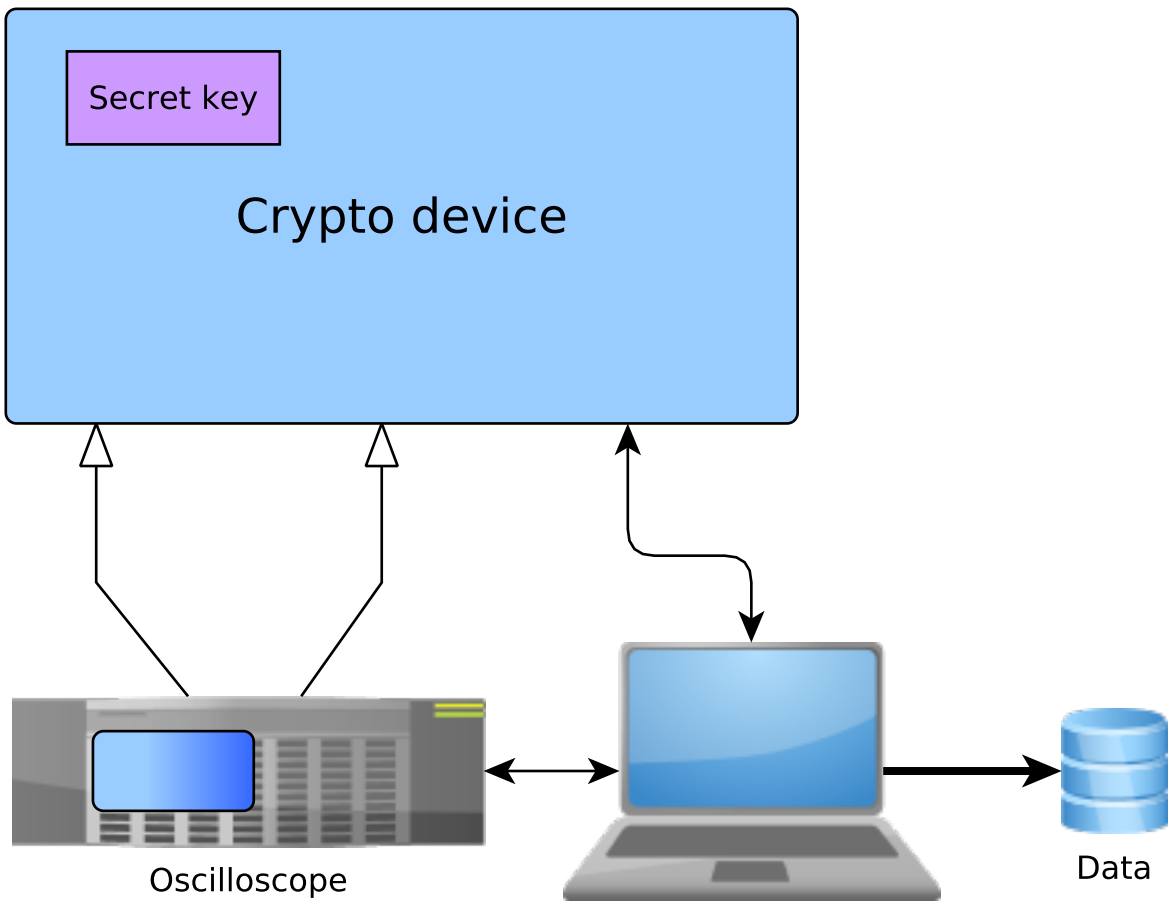
Classical cryptanalysis



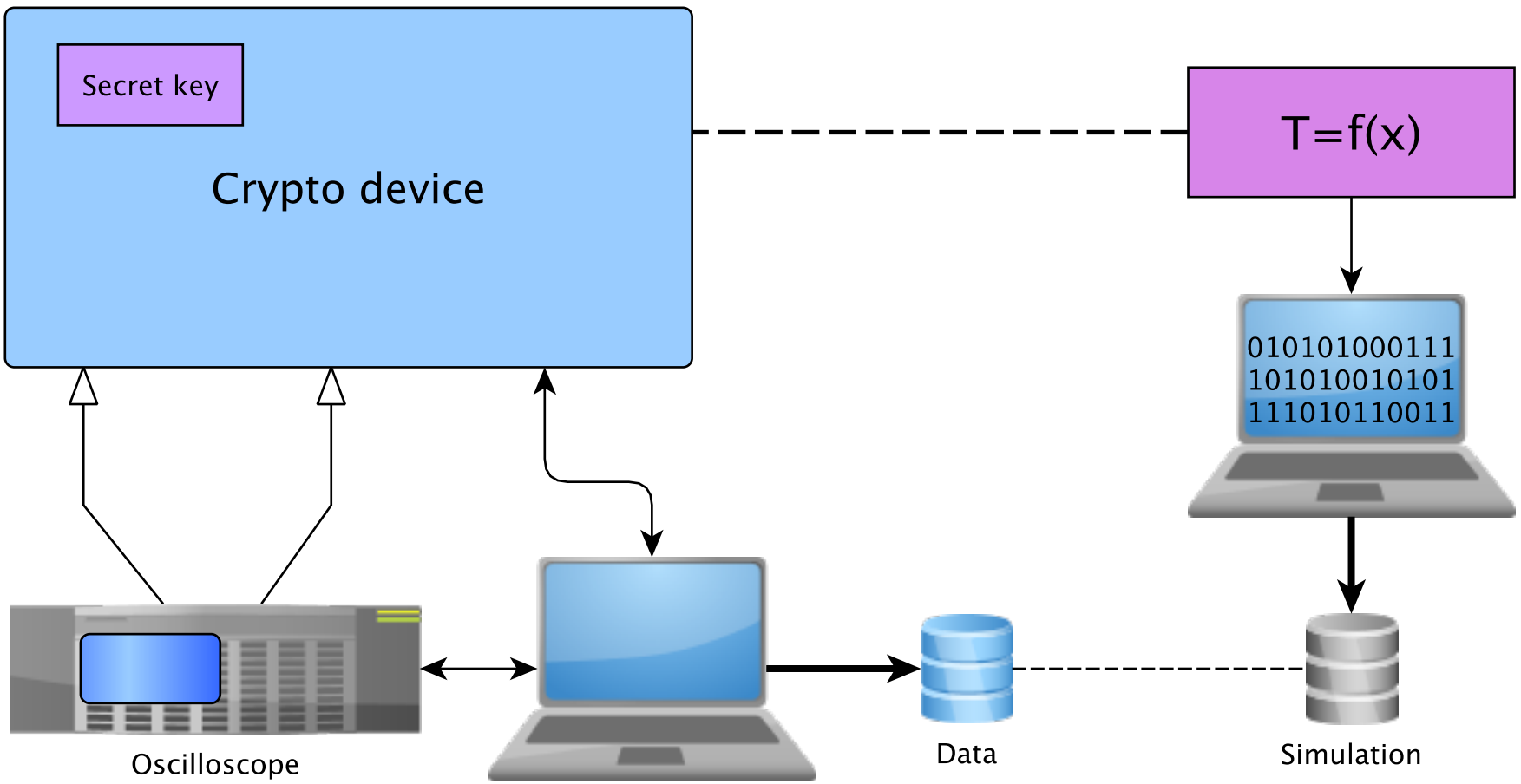
Side channel attacks



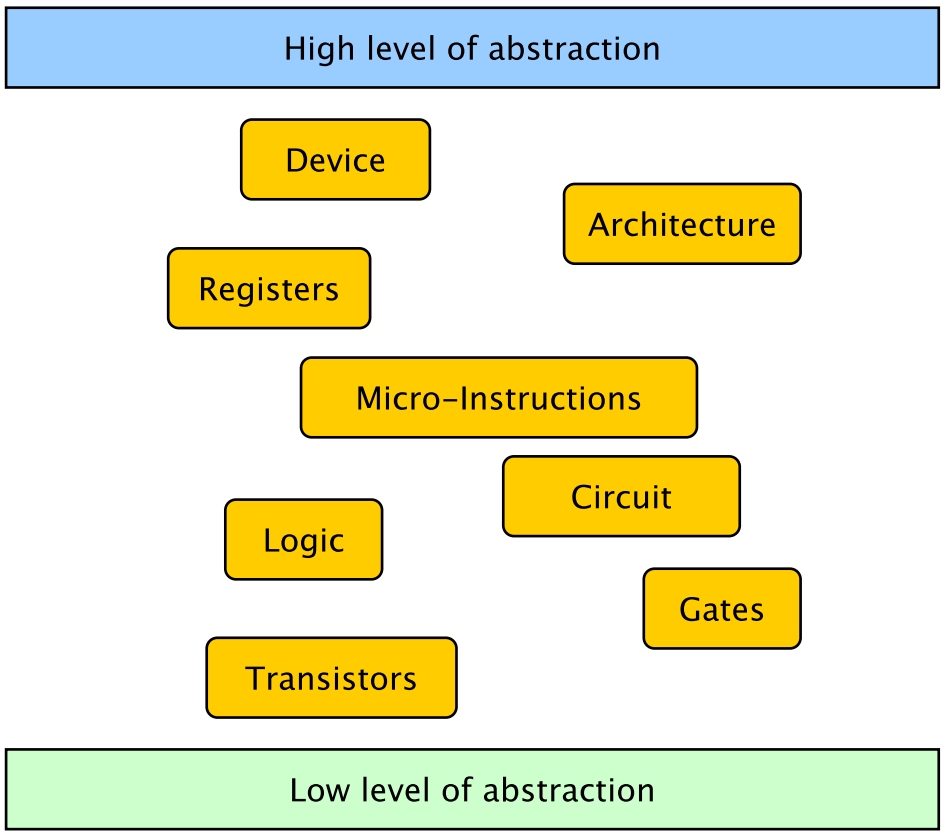
Power trace acquisition

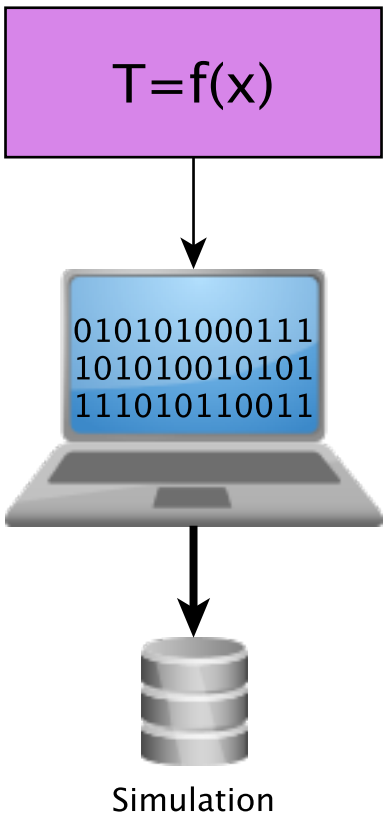


Simulation vs. Real experiment



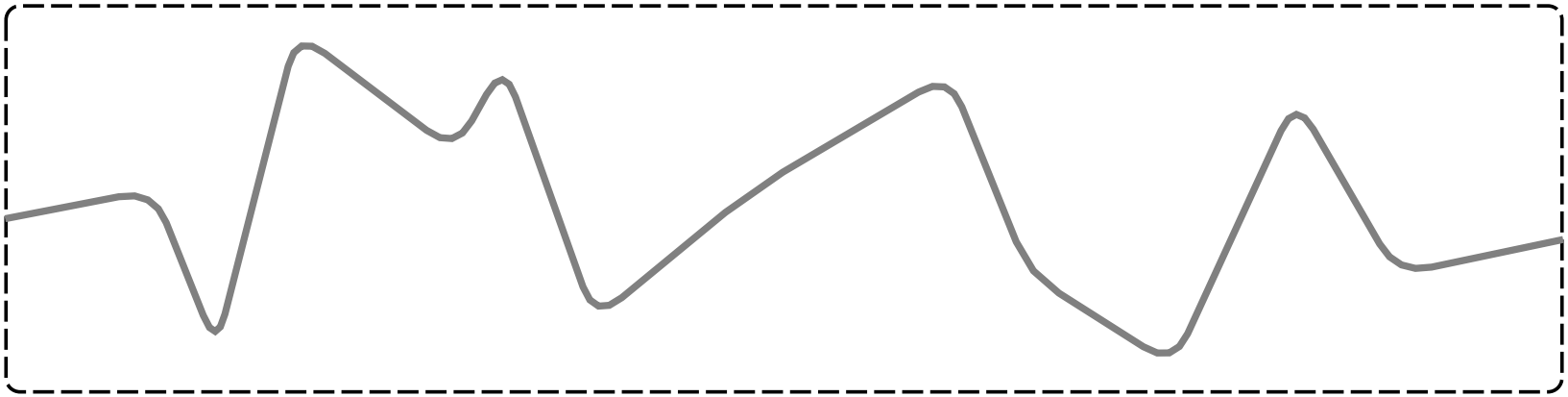
Simulators: levels of abstraction



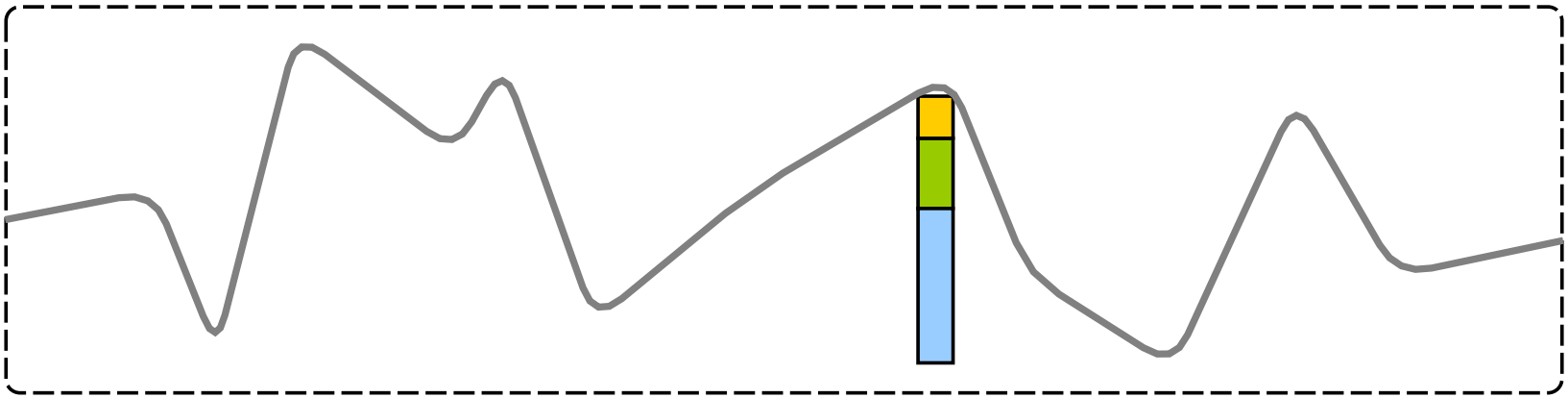


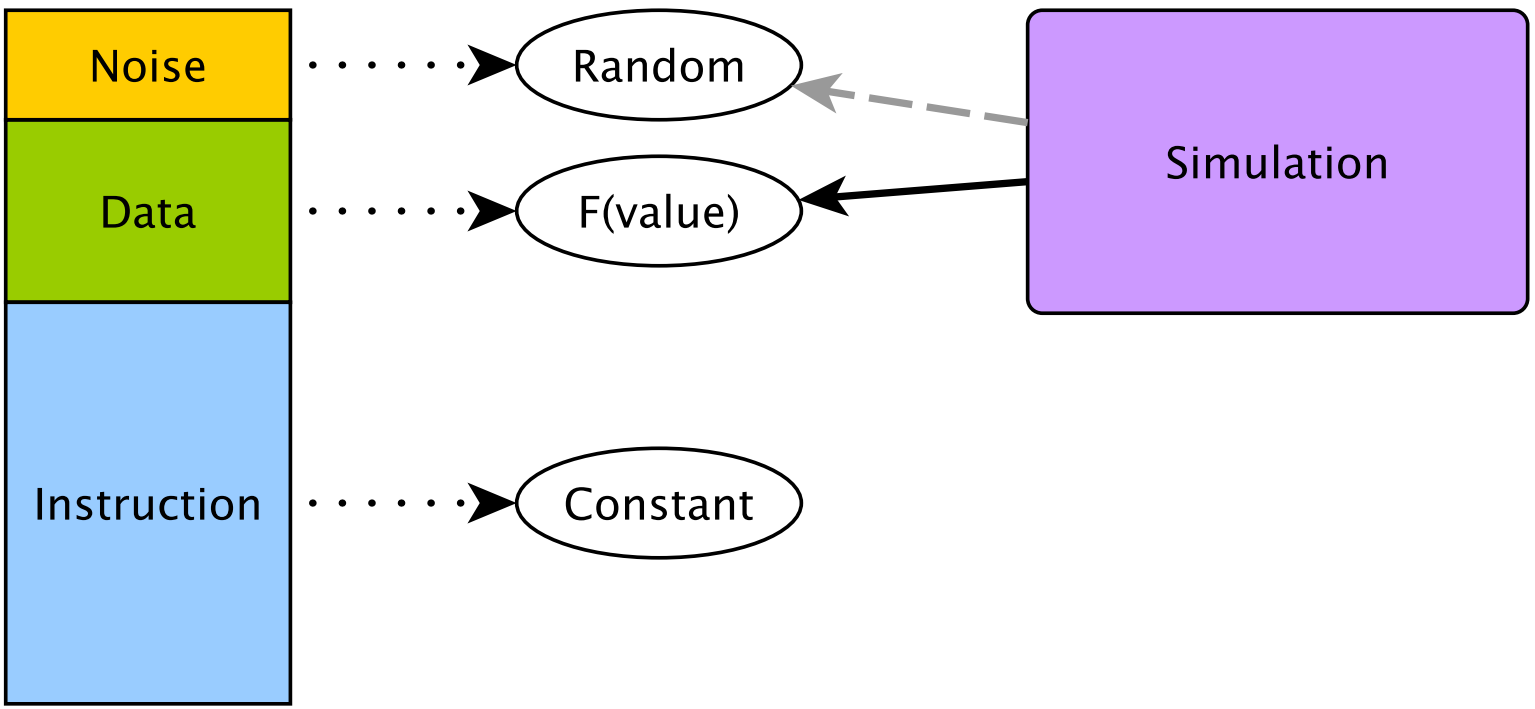
SILK

Power consumption

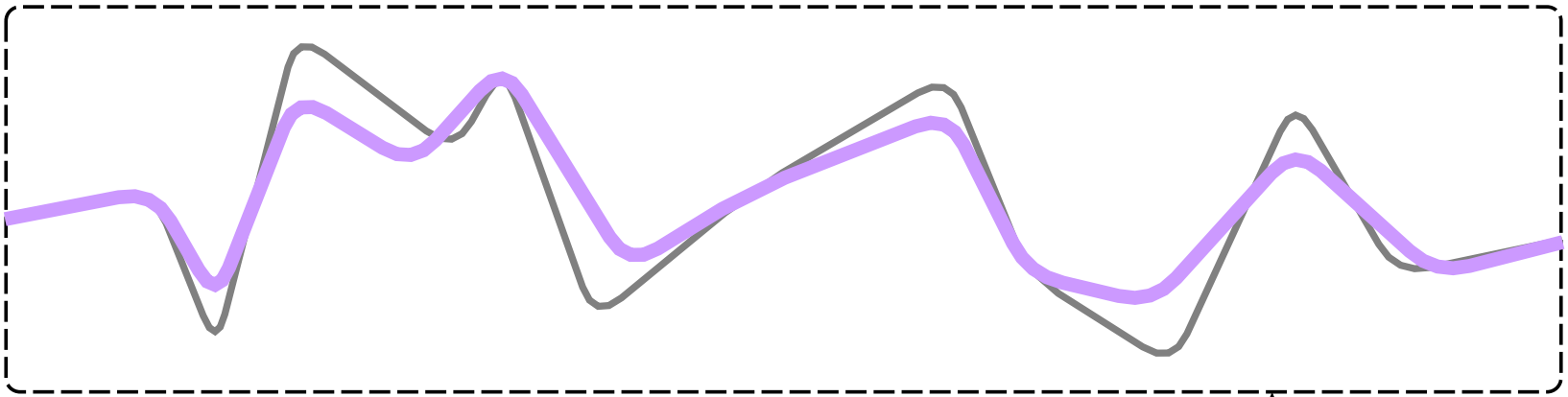


Power consumption: decomposition





Simulator



C++ code
```
#include <silk.h>
main(){
  encrypt(msg);
}
```

Simulation

Leakage distribution

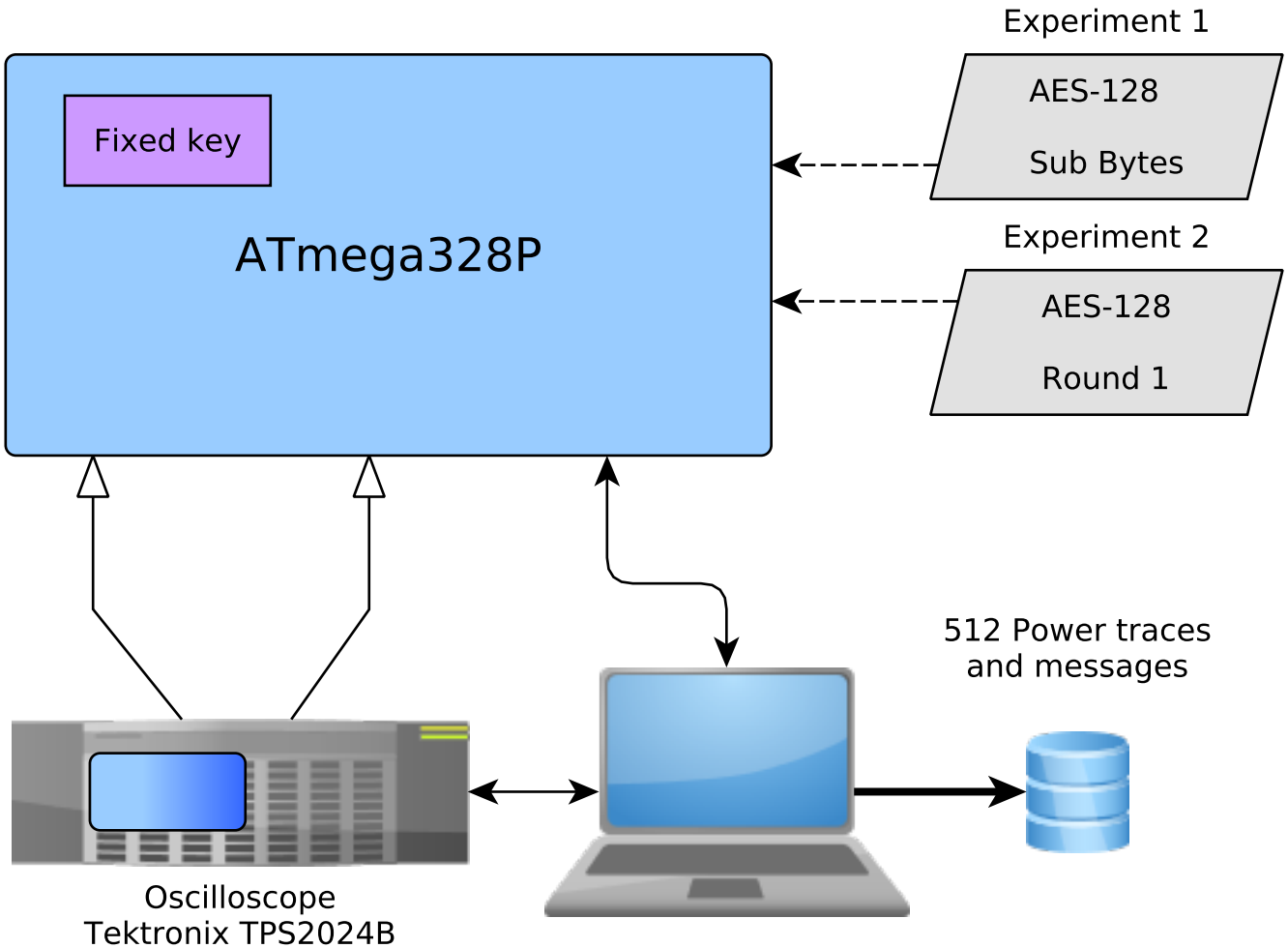
Overlap

Leaked points

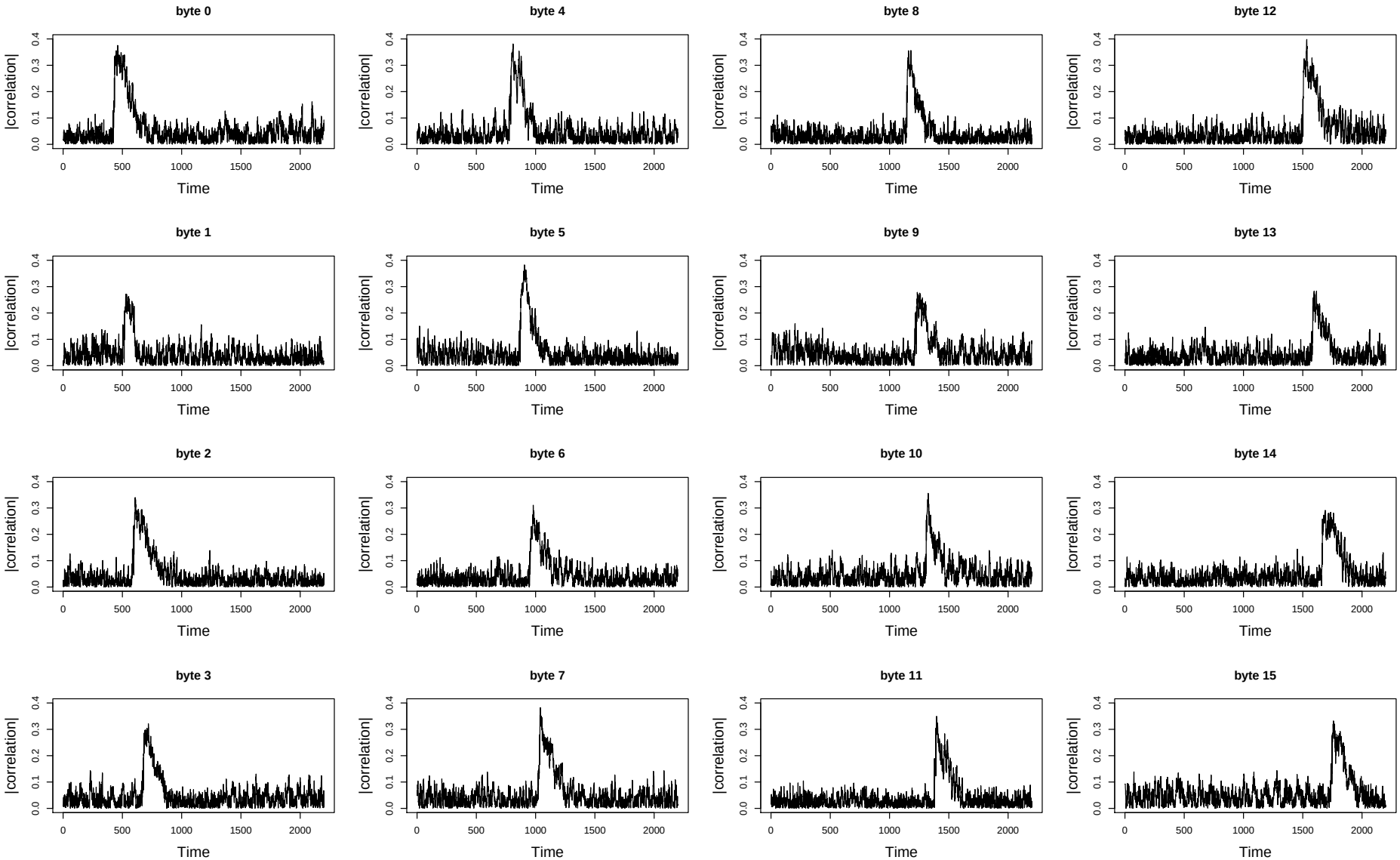
Leakage function

Experiments

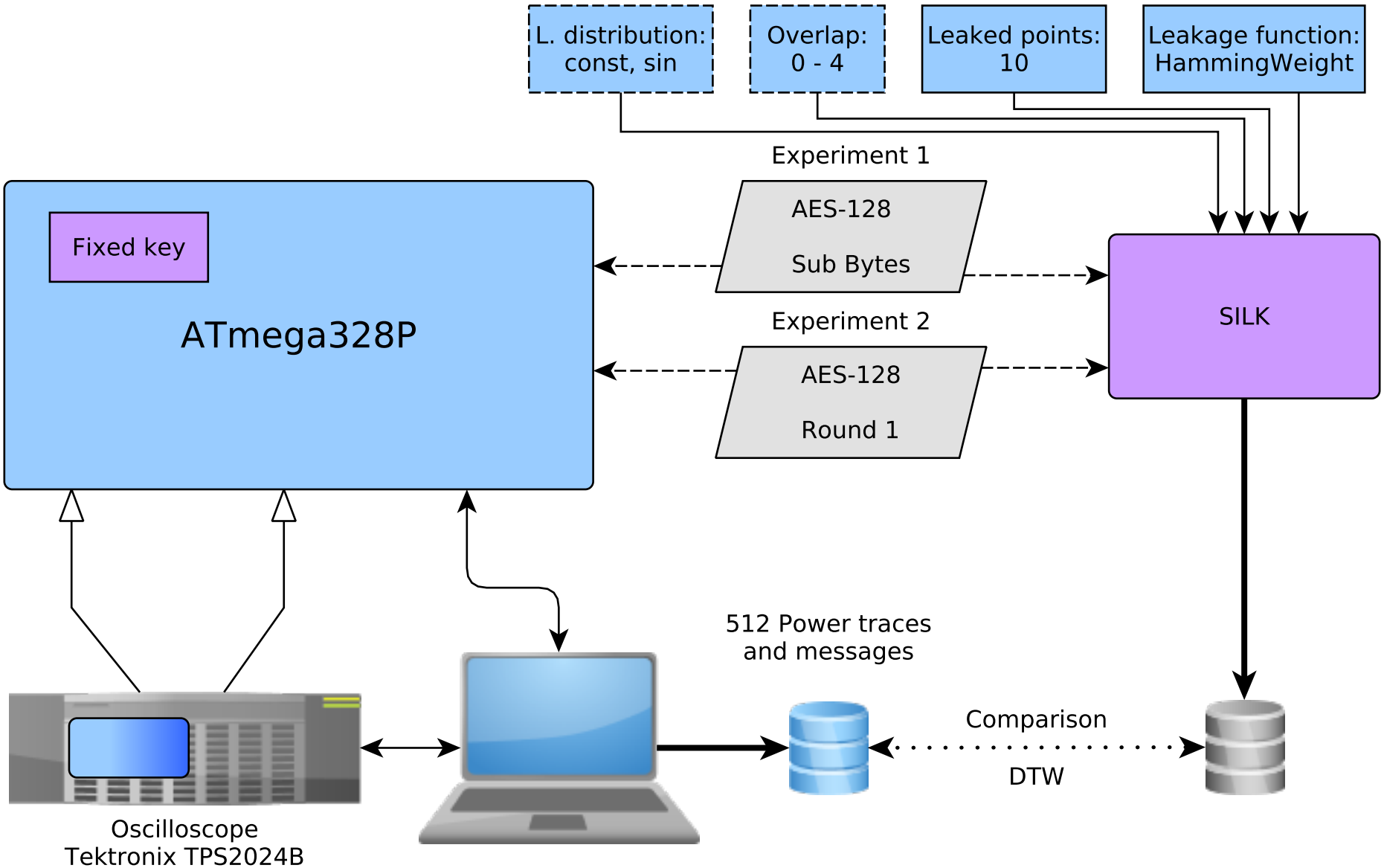
Target: data acquisition



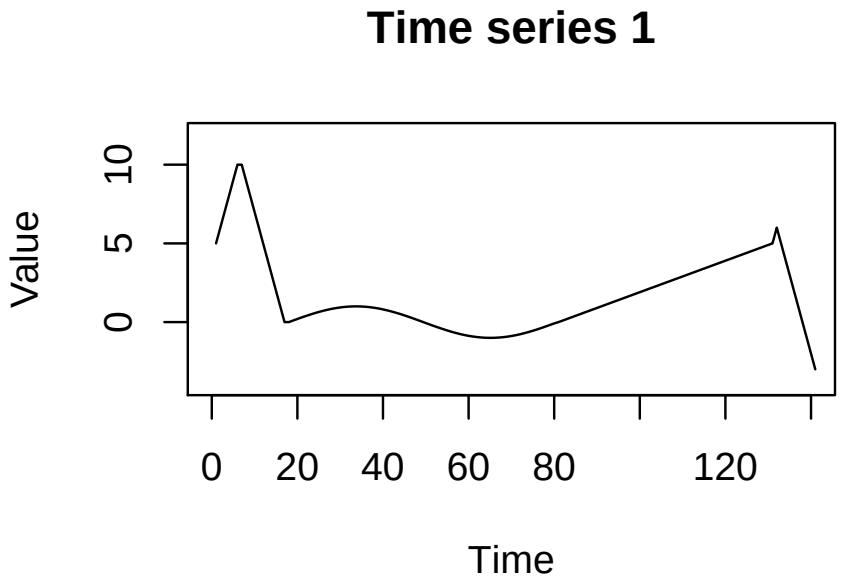
Correlation Power Analysis (with Hamming Weight)

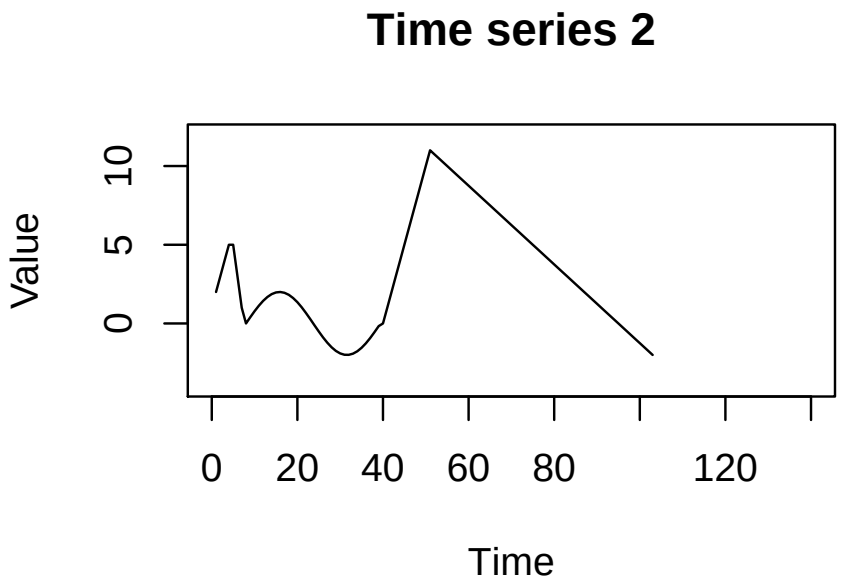


Target: data simulation

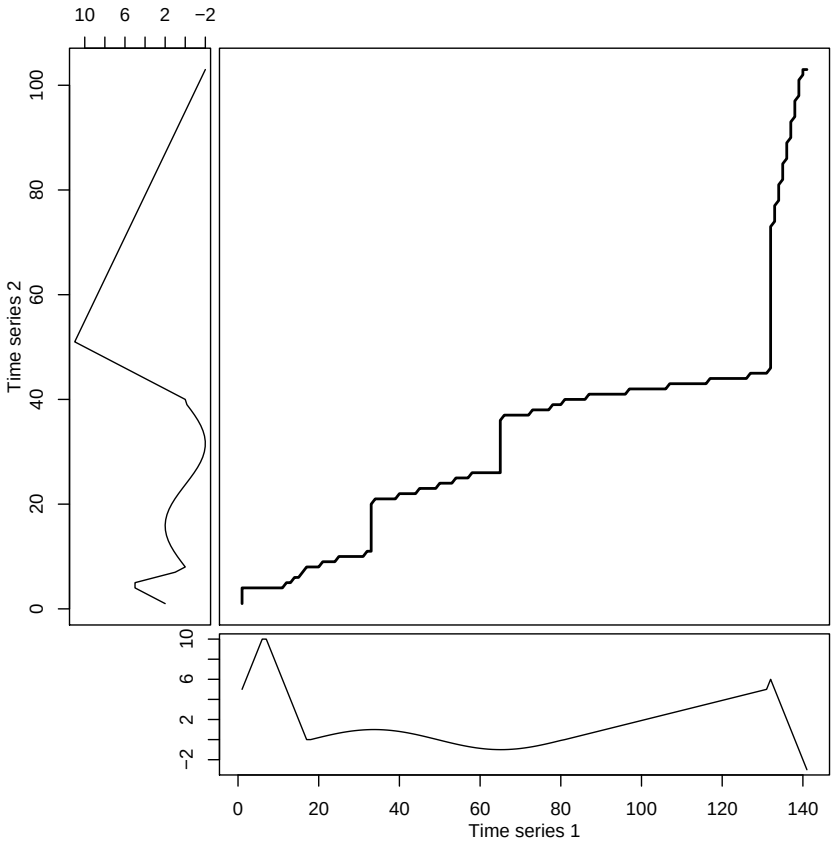


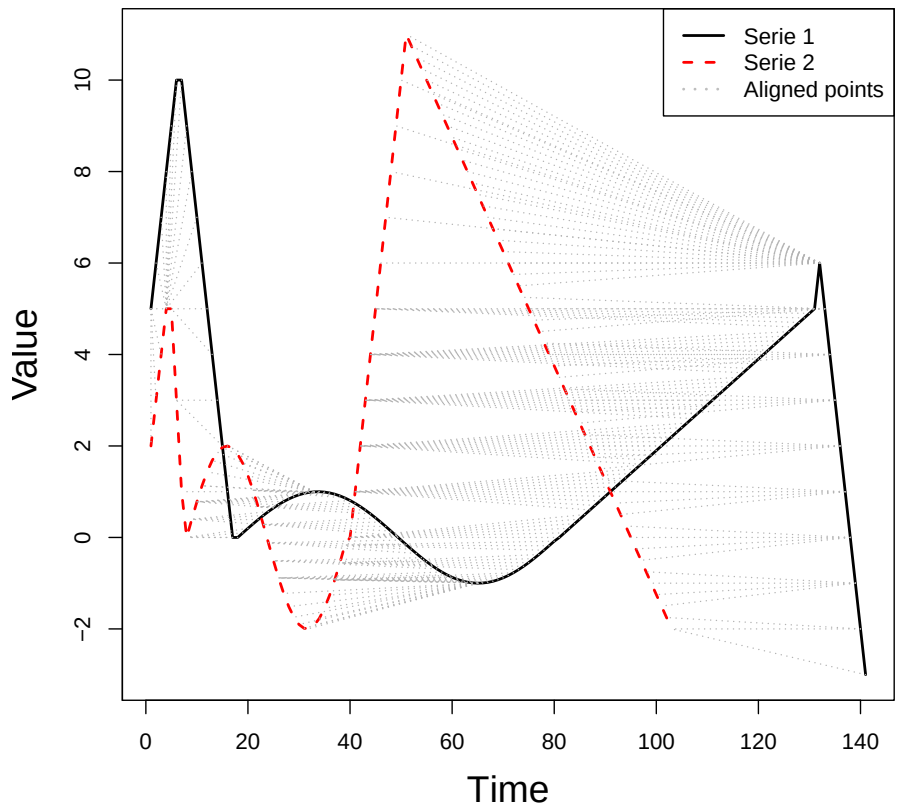
Dynamic Time Warping: Example





Dynamic Time Warping: Result





Evaluation with DTW

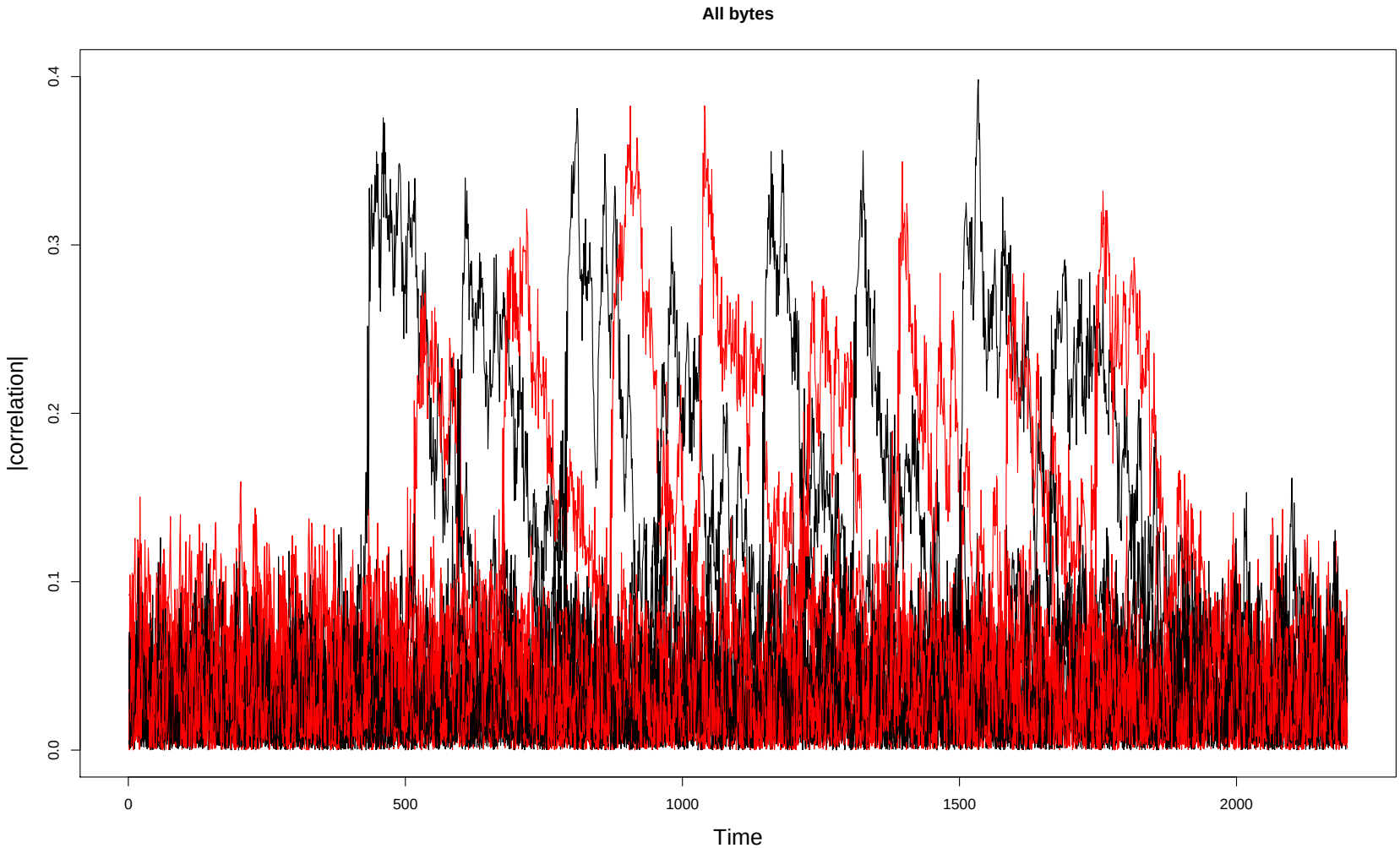
- R implementation: `dtw package`
- Distance metrics:
  - Euclidean
  - Correlation based: $Dist(P_1, P_2) = 1 - |correlation(P_1, P_2)|$

Results: SubBytes

	Euclidean				
Leakage	Overlap				
Distribution	0	1	2	3	4
const	1.419	1.444	1.470	1.497	1.526
$0.5 \times sin$	0.341	0.347	0.353	0.359	0.365
sin	0.656	0.668	0.680	0.692	0.705
$2 \times sin$	1.287	1.310	1.334	1.359	1.384

CPA on real traces & DTW (correlation based)

DTW Correlation based distance	Overlap				
	0	1	2	3	4
	0.793	0.788	0.785	0.780	0.777



Conclusions & Perspectives

Future works

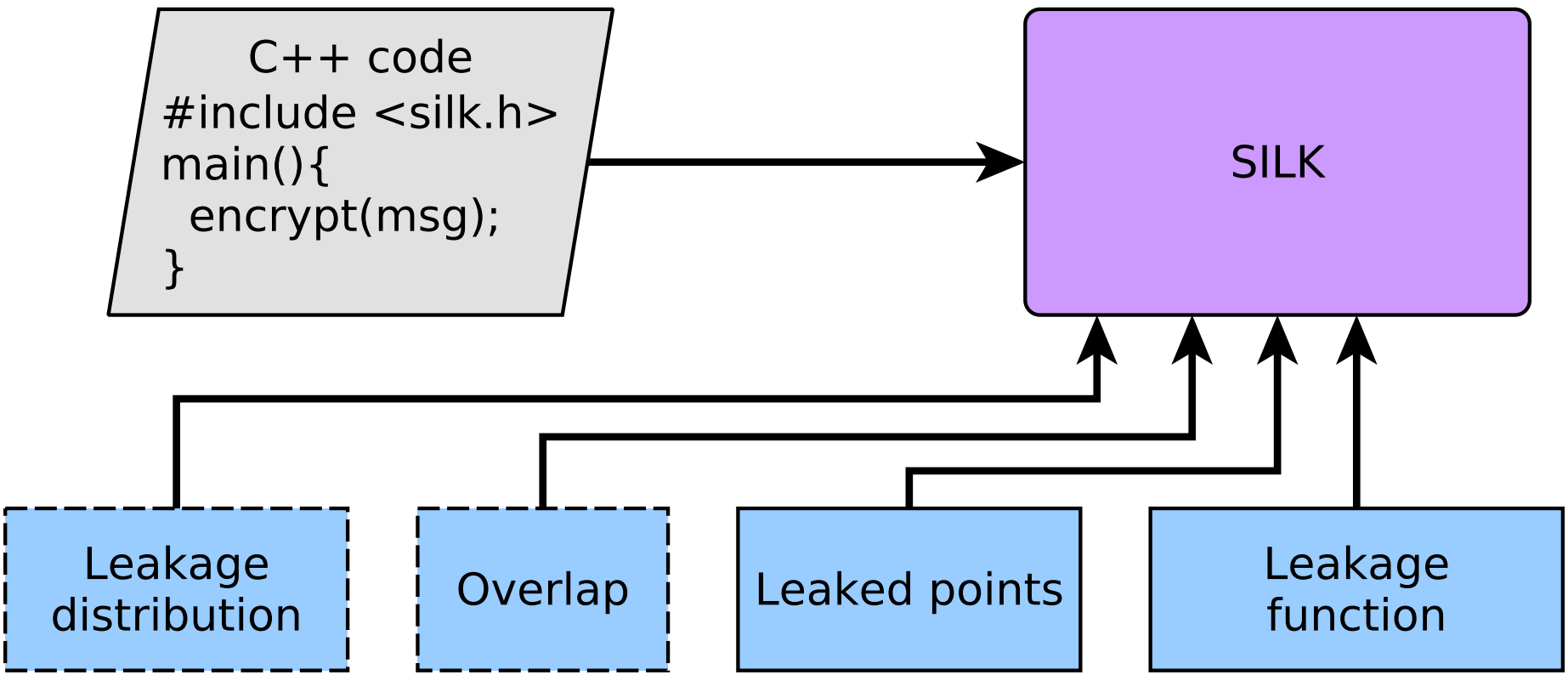
- Choosing best parameters
- Add flexibility
- Lower level of abstraction
- Simulators for ASIC/FPGA (VHDL)

Conclusions

- High level of abstraction leakage simulator – useful tool
- SILK for side channel analysis
- Reasonable results with simple assumptions

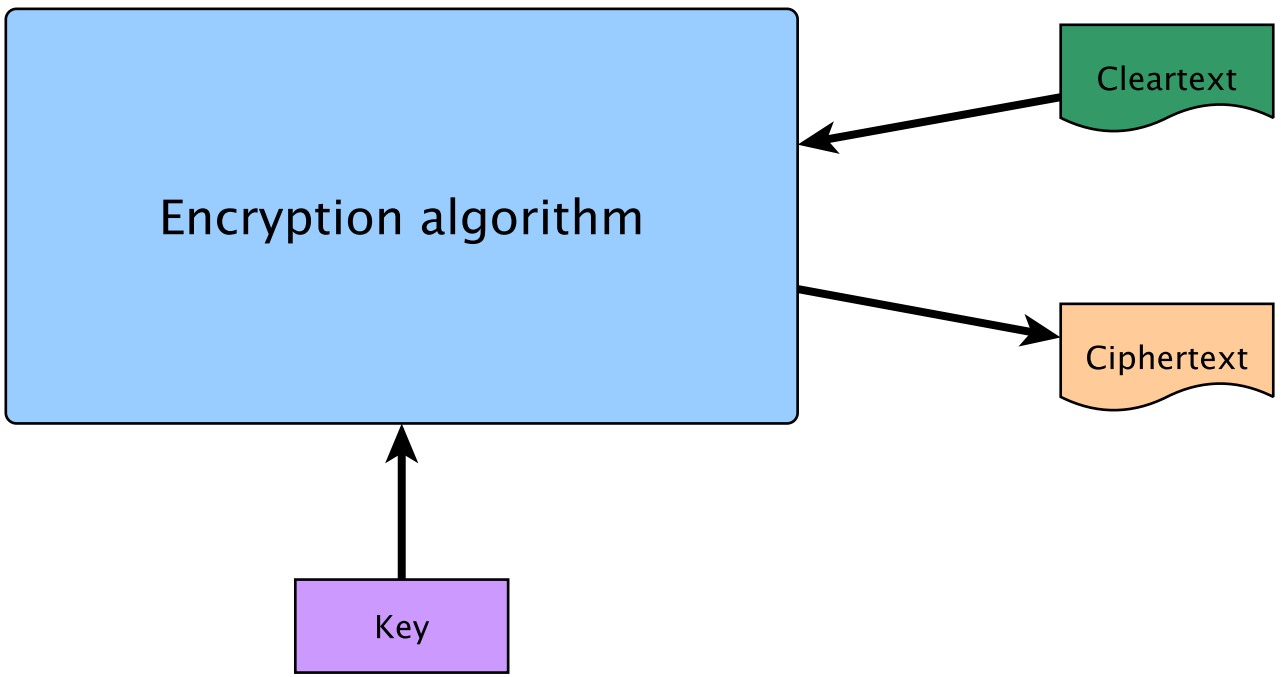
Questions?

nikita.veshchikov@ulb.ac.be
http://www.ulb.ac.be/di/dpalab/

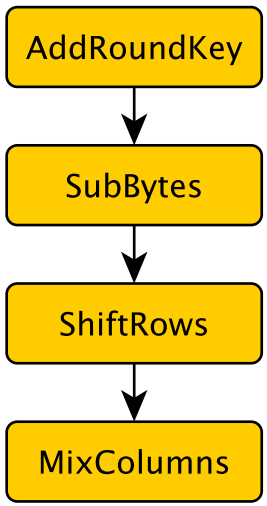


Backup

Encryption



AES round structure

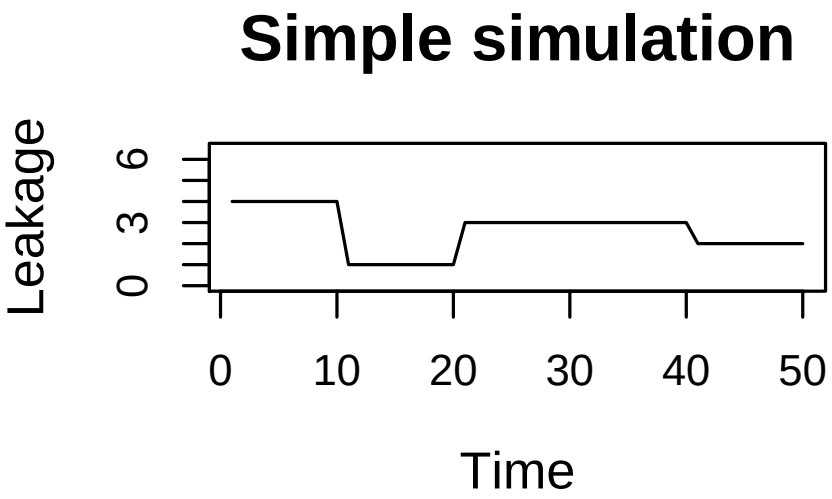


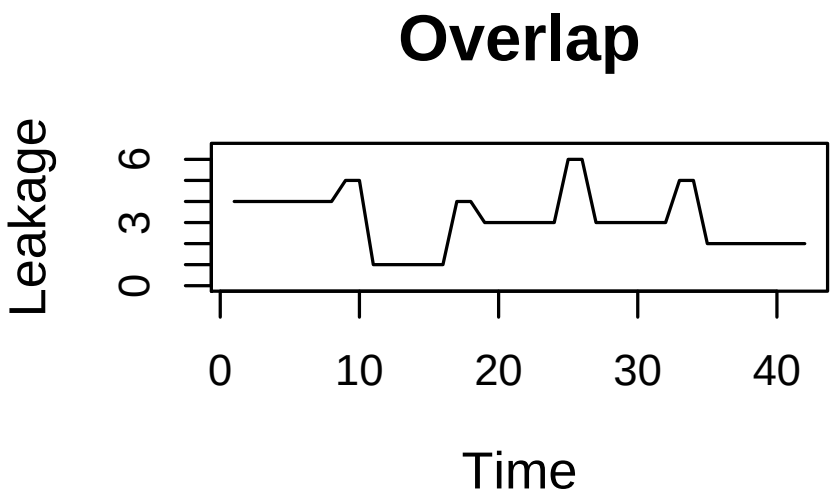
Example: Code

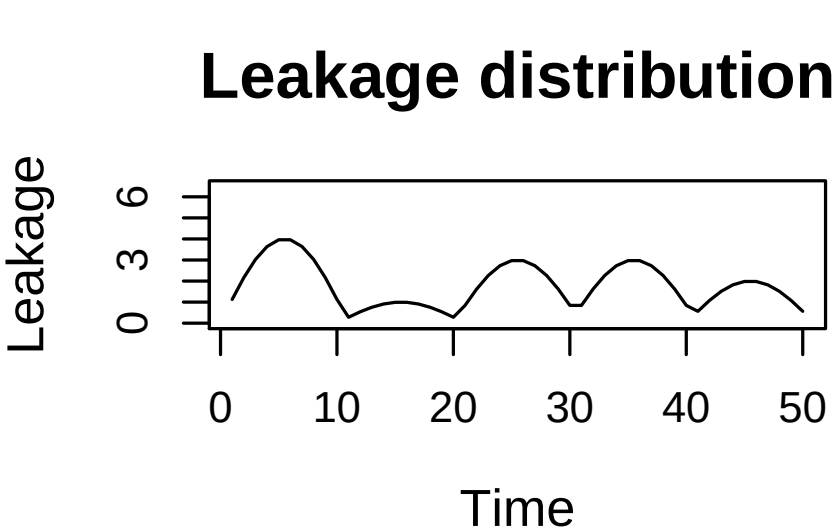
```
#include<iostream>

void bar(uint8_t param);

int main(){
  uint8_t myVar = 5;



  cout << bar(myVar);

  return 0;
}
```

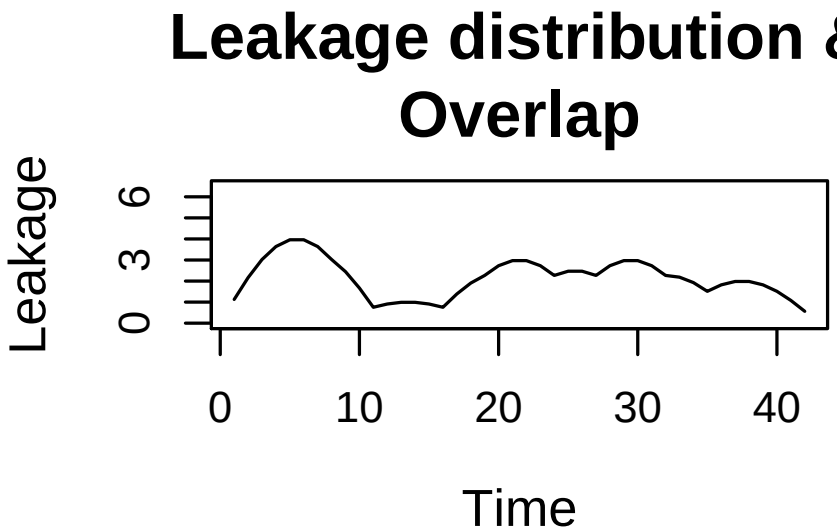
```
#include<iostream>
#include<silk.h>

void bar(U8 param);

int main(){
  U8 myVar = 5;
  U8::setLeakageFunction(myLFunc);
  Tracer::setLeakagePointsNbr(10);
  Tracer::setLeakageOverlap(2);
  cout << bar(myVar);
  Tracer::traceToFile("sim.csv");
  return 0;
}
```

Example: Simulated traces with different parameters

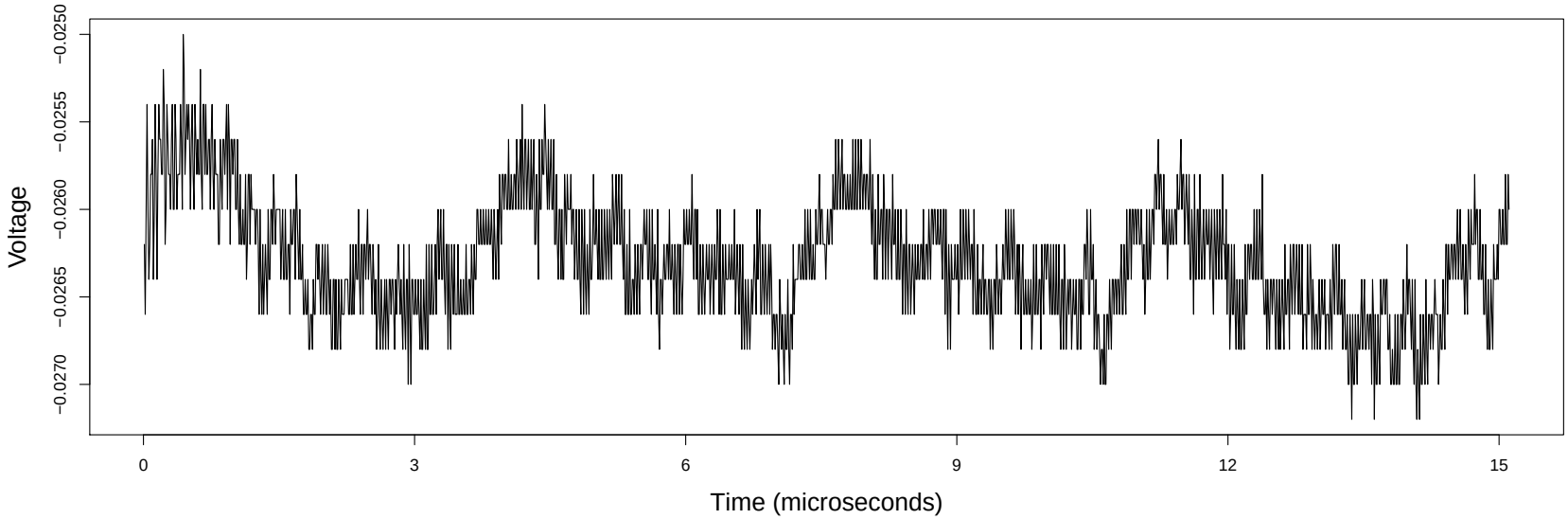


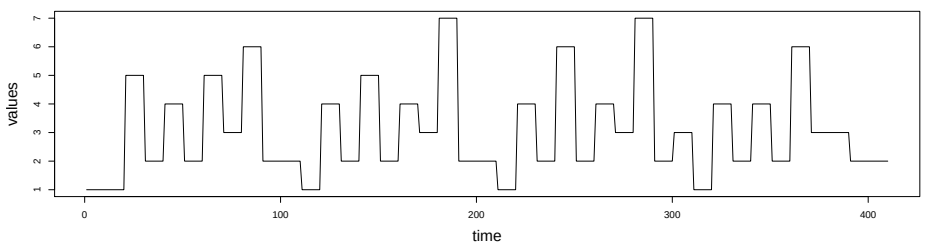


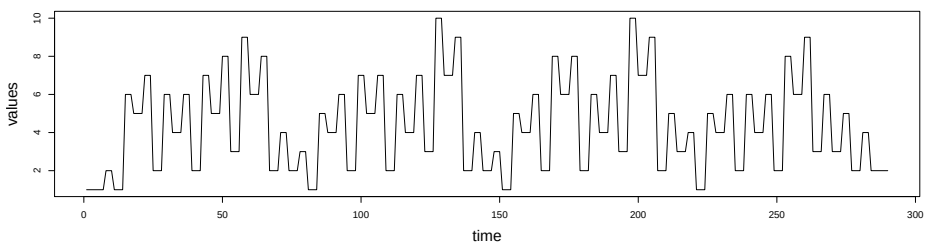


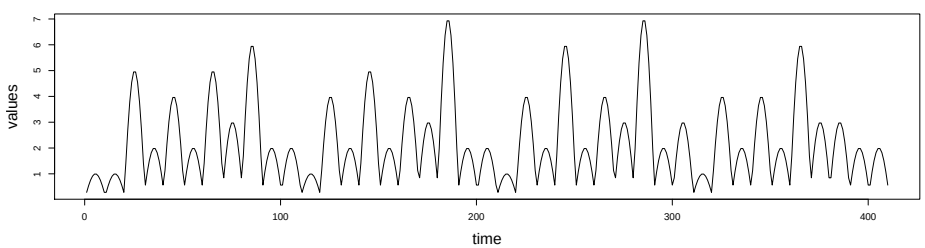


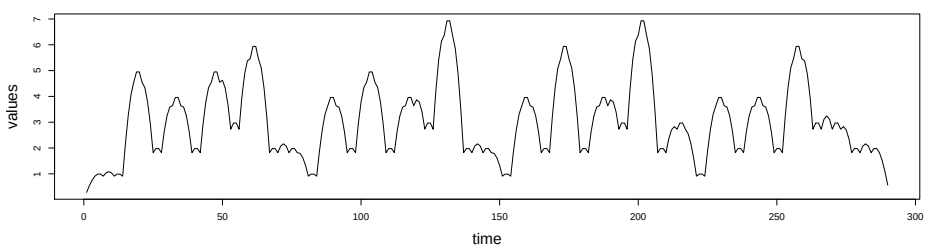
Real power trace vs. Simulations (SubBytes)











CPA on simulated traces (SubBytes)