SELECTING TIME SAMPLES FOR MULTIVARIATE DPA ATTACKS

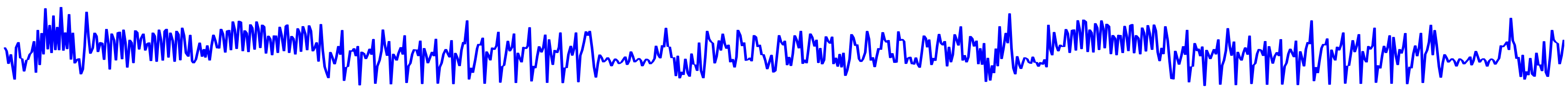
Oscar Reparaz, Benedikt Gierlichs, Ingrid Verbauwhede

KU Leuven, COSIC

CHES 2012 - Leuven, Belgium
2012-09-14

DPA attacks: computational complexity

- Univariate setting: selecting the interesting time sample and key-recovery often done simultaneously (affordable, linear in the trace length)



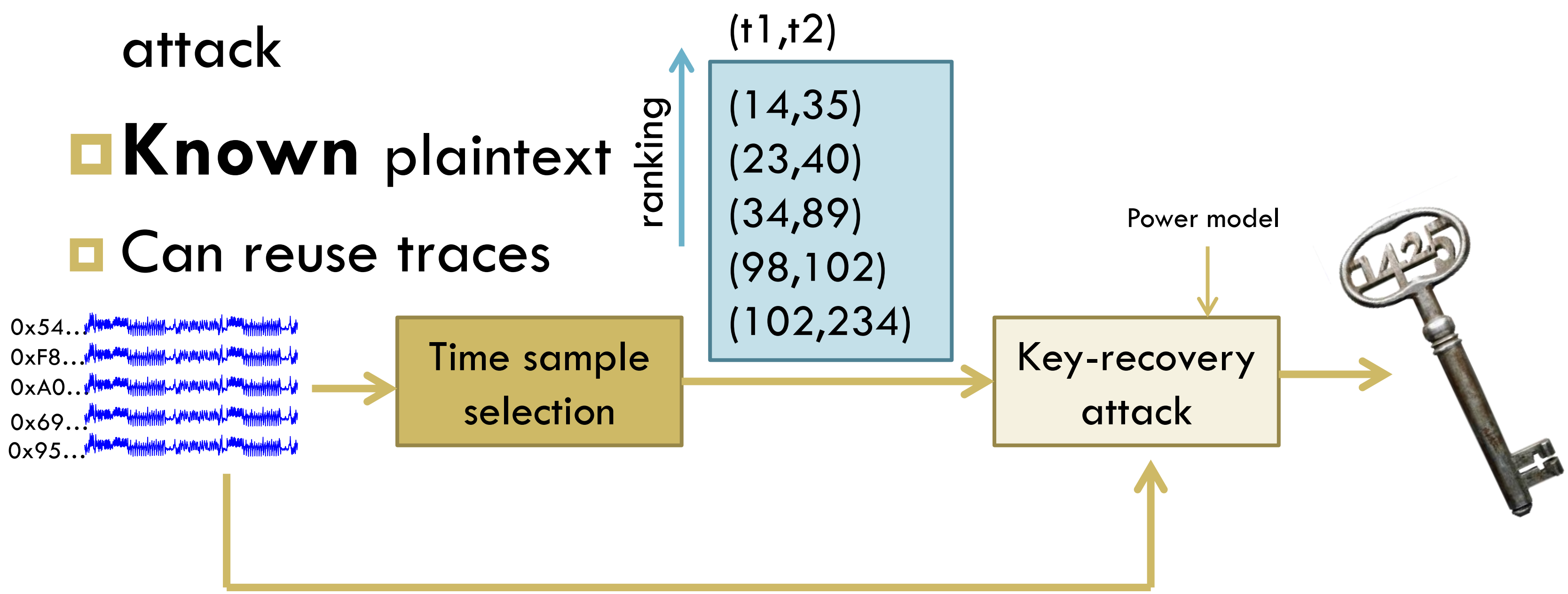
- Multivariate setting: **expensive** to test all tuples for all key-hypothesis (e.g. n x (n-1) / 2 pairs)
  - To speed-up, divide the problem:
    - First find few "interesting" tuples ← **This talk**
    - Then key recovery attack

Known methods for time sample selection, multivariate setting

- Educated guess [Oswald et al.]
  - Reduces time window, does not output tuples
- Variance method [Lemke-Rust and Paar, Gierlichs et al.]
  - Chosen plaintext, new traces for key recovery, selects time samples, does not output tuples
- Correlation-based [Agrawal et al.]
  - Chosen plaintext, new traces for key recovery, selects tuples
- Fourier-based [Waddle and Wagner]
  - Heuristic

Proposed method

- This paper's method:
  - Selects **tuples** for multivariate DPA attacks
  - Outputs **ranked list** of tuples ⇒ natural order for the attack
  - **Known** plaintext
  - Can reuse traces



Core idea

- Let **M** be masks, **P** plaintexts, **V** masked sensitive variable

$$\mathbf{V}=\mathbf{M} \oplus \mathrm{Sbox}(\mathbf{P}+k)$$

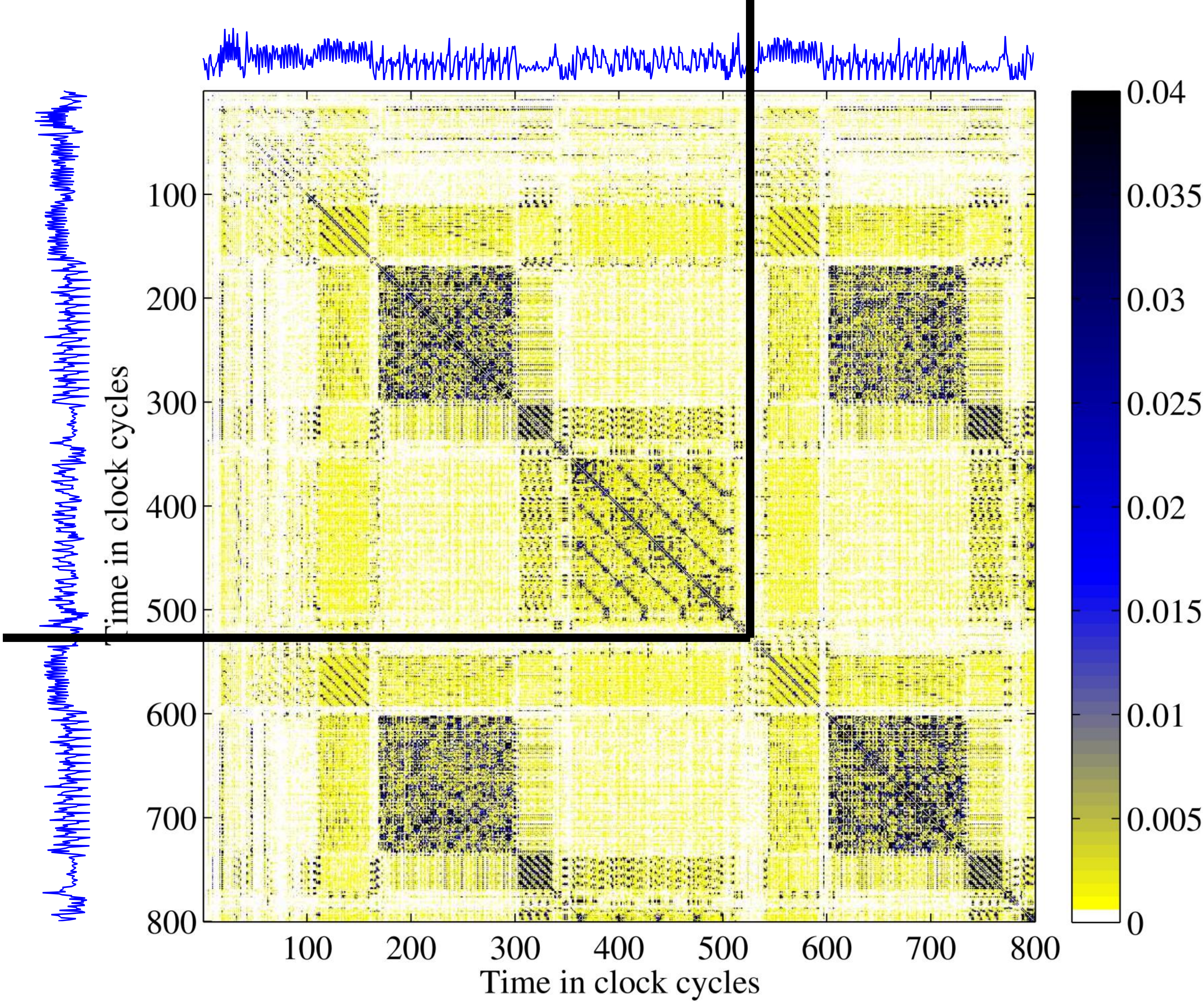
- Suppose the plaintext is fixed **P**=p
- Only **M** varies, implies changing values of **V**

$$\mathbf{V}=\mathbf{M} \oplus \mathrm{Sbox}(p+k)$$

$$\Rightarrow \mathbf{I}(L(\mathbf{M});\, L(\mathbf{V})) \neq 0$$

- On the other hand, for unrelated time samples (t1,t2)

$$\mathbf{I}(L(t1);\, L(t2))=0$$


Time in clock cycles
Time in clock cycles
100
200
300
400
500
600
700
800
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0

Extending the core idea

- Previous method has two drawbacks:
  - Chosen plaintext (undesirable)
  - Not all of the selected tuples are interesting! For example: handling some value twice.
- We can get rid of both drawbacks by extending the core idea to known plaintext:
- **V**, **M** and **P** are not mutually independent

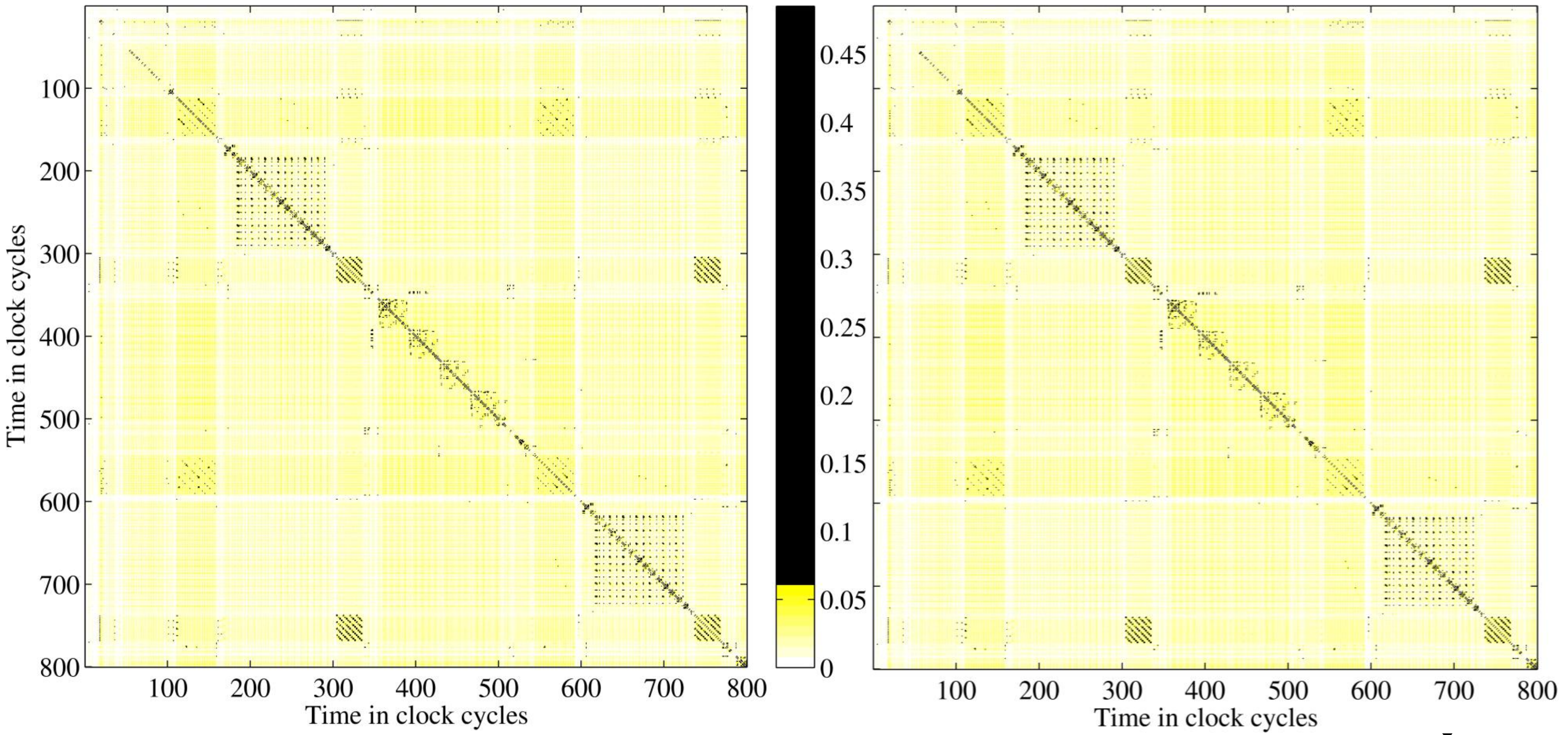
$$\mathbf{V} = \mathbf{M} \oplus \mathrm{Sbox}(\mathbf{P}+k)$$

$$\Rightarrow I(L(\mathbf{V}); L(\mathbf{M}); L(\mathbf{P})) \neq 0$$

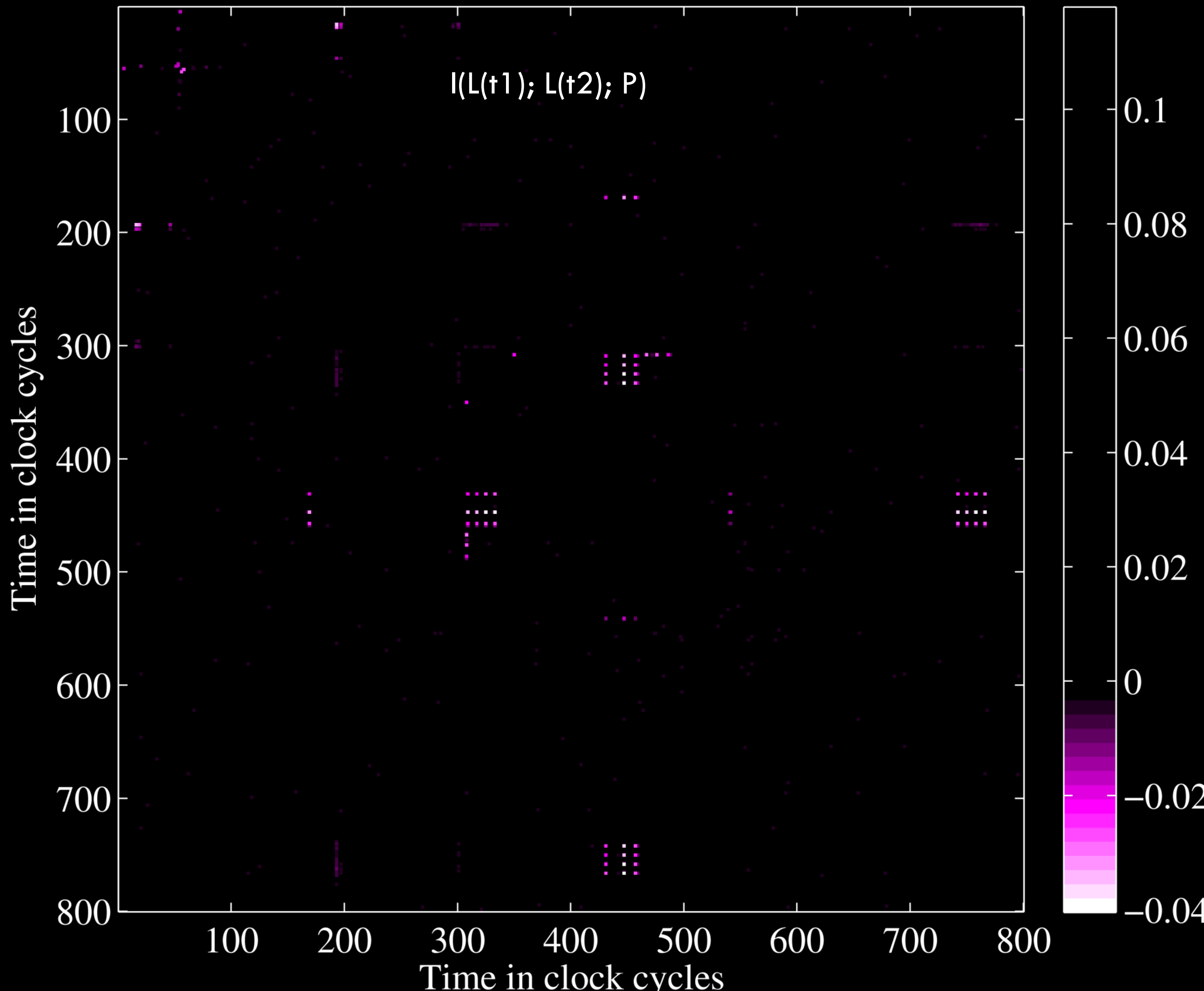
- For unrelated (t1,t2,t3) ⇒ **I**(t1; t2; t3) = 0
- No need to search for L(**P**), **P** is known, apply some **L**()
- The method: compute **I**(**L**(t1); **L**(t2); **L**(**P**))

The method

I(L(t1); L(t2); P) = I(L(t1); L(t2)) – I(L(t1); L(t2) | P)



Difference between terms is tiny, invisible here. Next slide: only I(L(t1); L(t2); P)



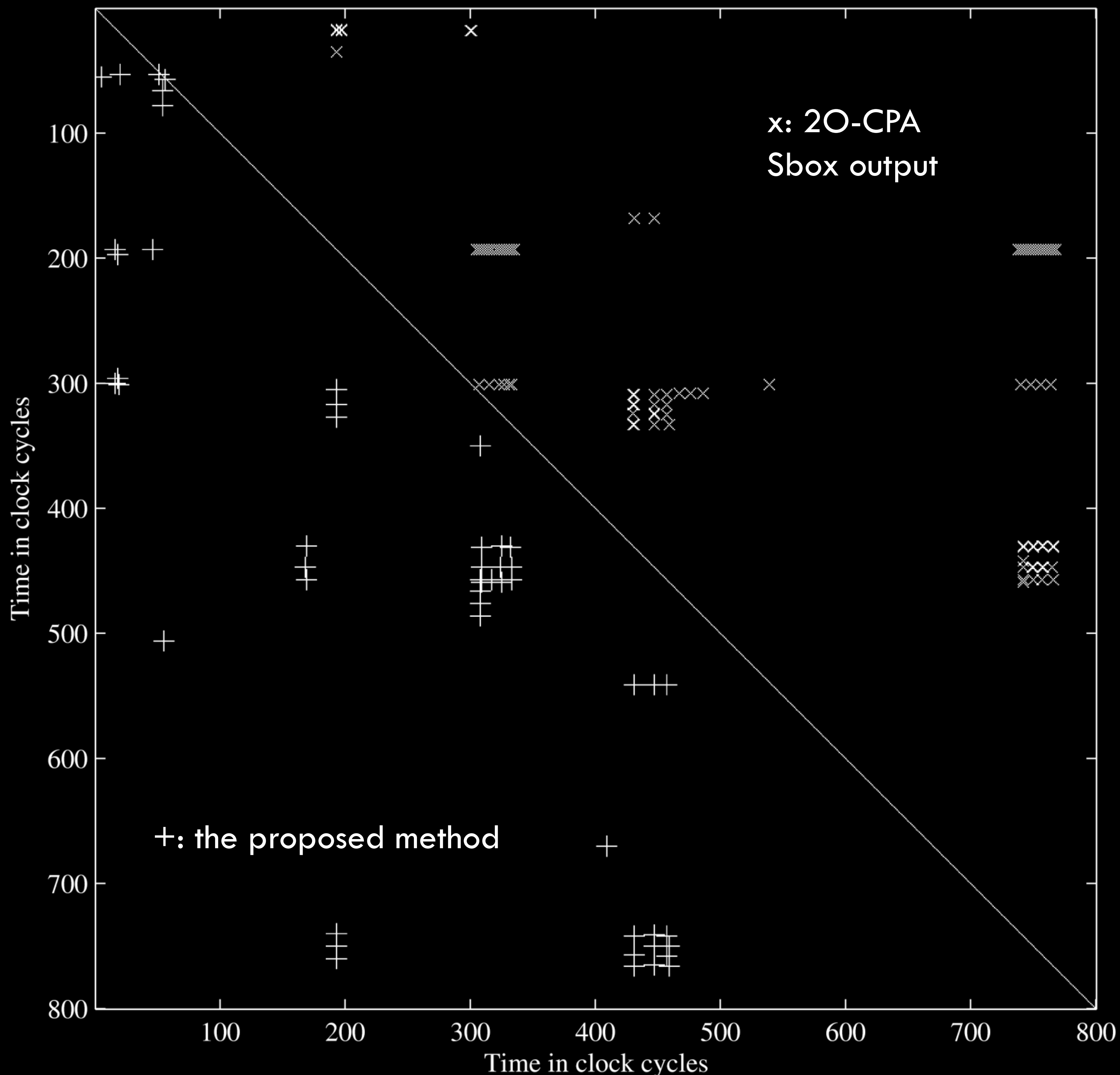
I(L(t1); L(t2); P)
Time in clock cycles
100
200
300
400
500
600
700
800
Time in clock cycles
100
200
300
400
500
600
700
800
0.1
0.08
0.06
0.04
0.02
0
−0.02
−0.04

Which tuples are identified?

- Depends on **L**. (The attacker has freedom to choose **L**)

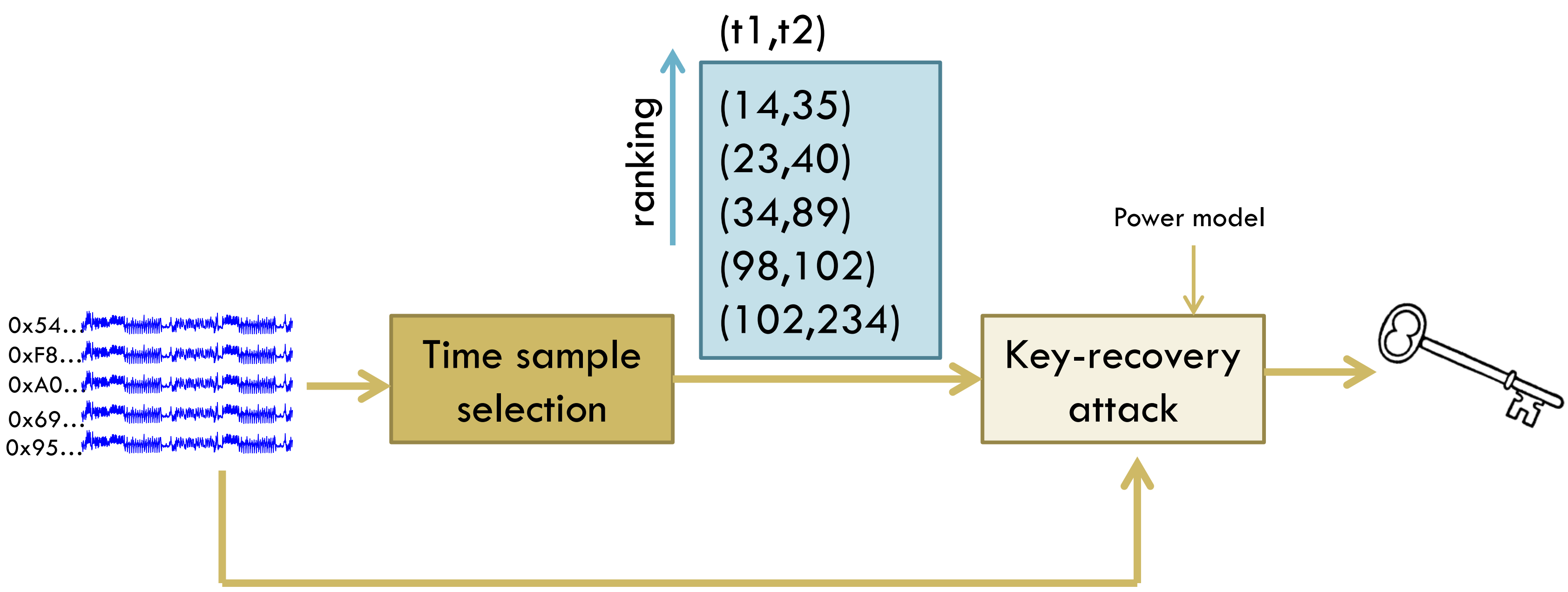
$$I(\mathbf{L}(t1); \mathbf{L}(t2); \mathbf{L}(P))$$

- Different behavior depending on **L**
  - **L = Id** ⇒ I(**L**(t1); **L**(t2); P)
    - Shares of the plaintext
    - Shares of the sbox input
    - **Shares of the sbox output** (works for bijective and non-injective)
      - Normally leakage of sbox output shares is the easiest to attack ⇒ good
      - In our experiments, the method **mostly** selected time samples corresponding to shares of sbox output: see next slide

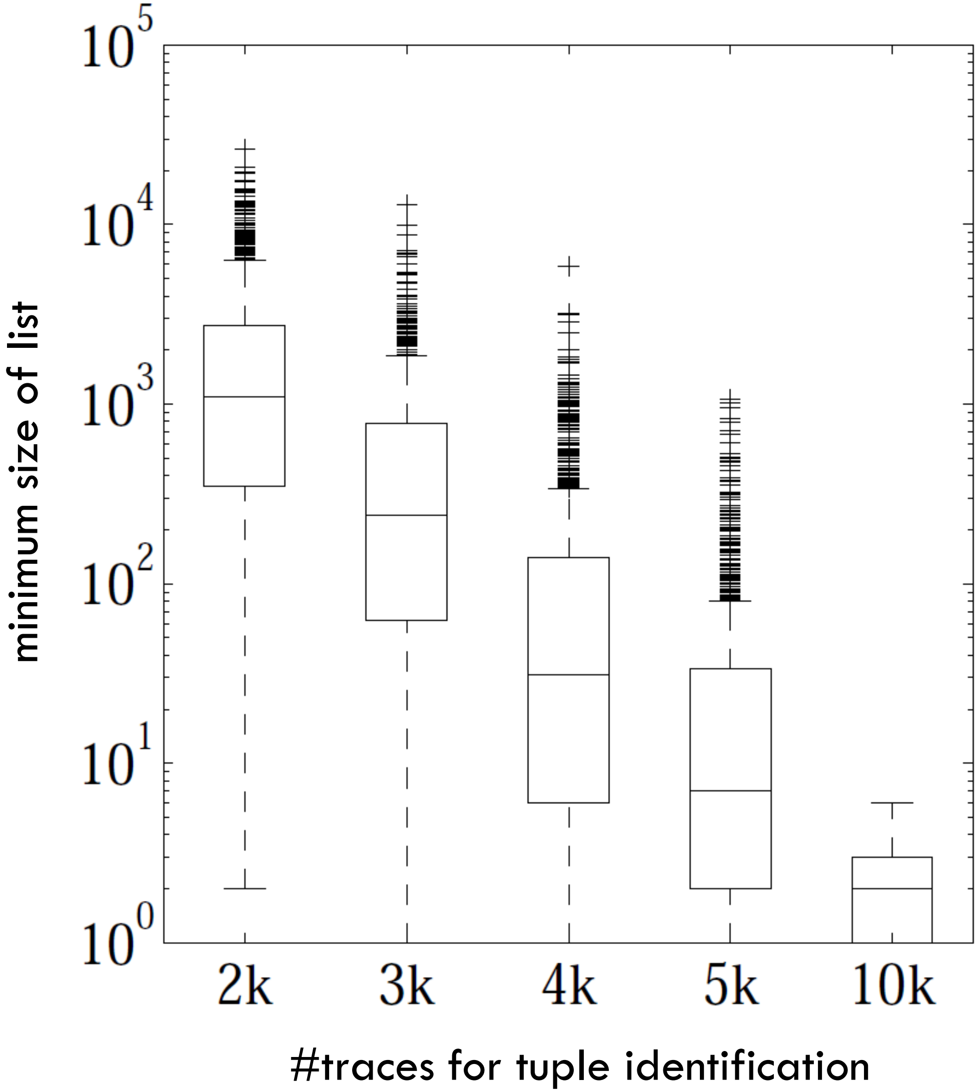

x: 2O-CPA
Sbox output
+: the proposed method
Time in clock cycles
Time in clock cycles
100
200
300
400
500
600
700
800

Evaluation

- Isolate performance of phases



Evaluation: min size of list



mMIA attack
Others: see paper

Running time improvement

- Theoretical improvement factor in running time
  - speed-up ≤ |subkey space|
  - in our example, bytewise key recovery: ≤ 256
- Empirical: 40...100 times faster

- Numbers are for one byte of an AES key, speed-up can apply to other bytes

- Trade-off: running time vs. number traces
  - Empirical: < 5 times number of traces

Conclusion

- A method to identify relevant tuples of time samples suitable for multivariate DPA attacks
  - No key guess, requires known (not chosen) plaintext, traces can be used for key recovery, traverses Sboxes
  - Does not place any hypothesis on leakage behavior, but knowledge can be used for further speed-up
  - Leads to a speed-up of orders of magnitude in multivariate DPA attacks
  - Cost: more traces (not orders of magnitude)
- Black-box evaluation less complex, can be automated
- Other applications: bit-tracing
  - Animations: http://homes.esat.kuleuven.be/~oreparaz/ches2012/