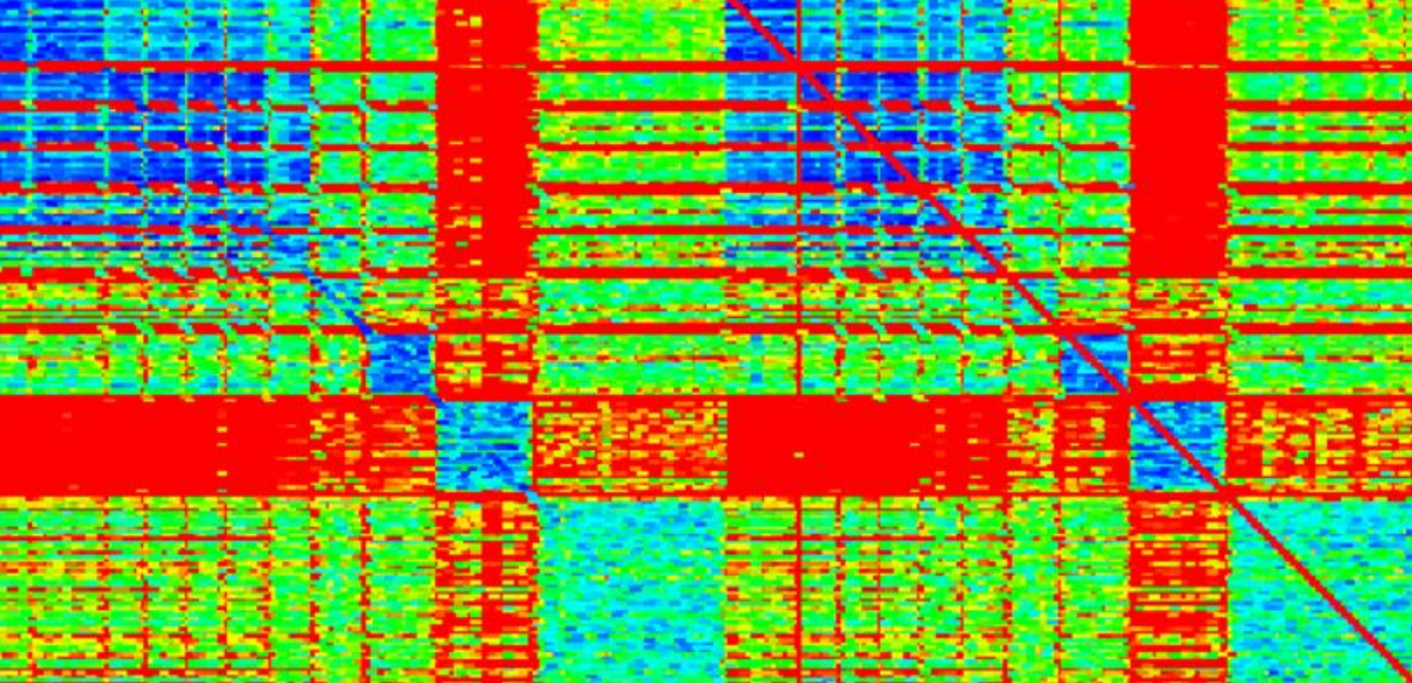
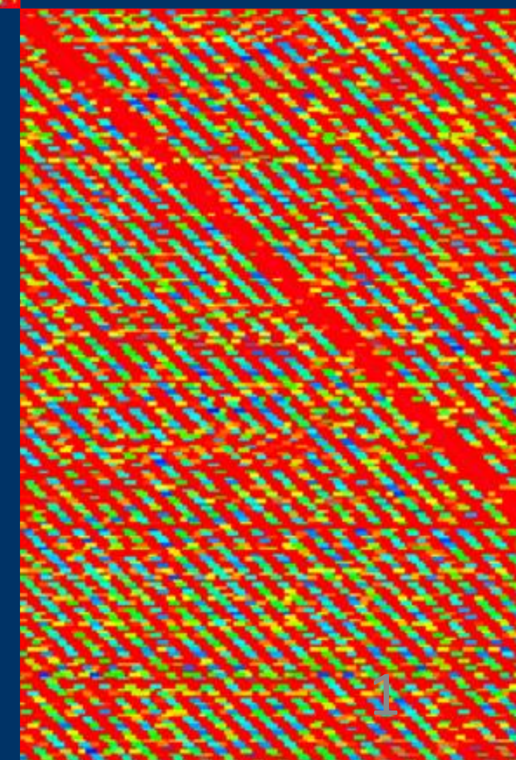


A MASKED RING-LWE IMPLEMENTATION

Oscar Reparaz, Sujoy Sinha Roy,
Frederik Vercauteren, Ingrid Verbauwhede

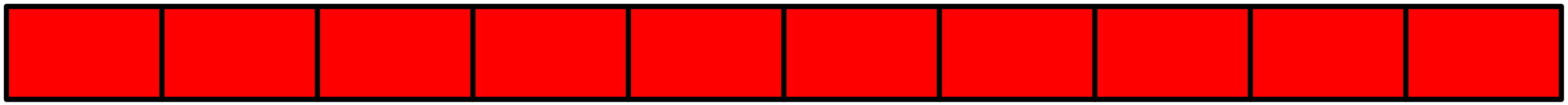
COSIC/KU Leuven

CHES 2015, Saint-Malo, FR



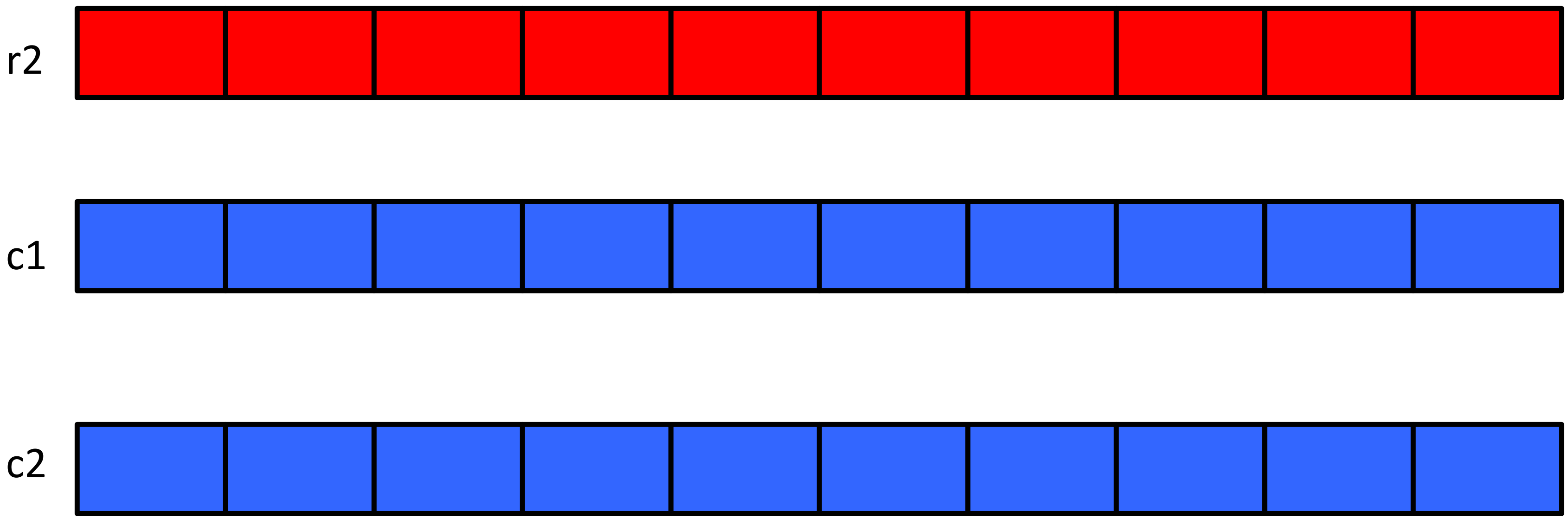
unprotected ring-LWE decryption

r2



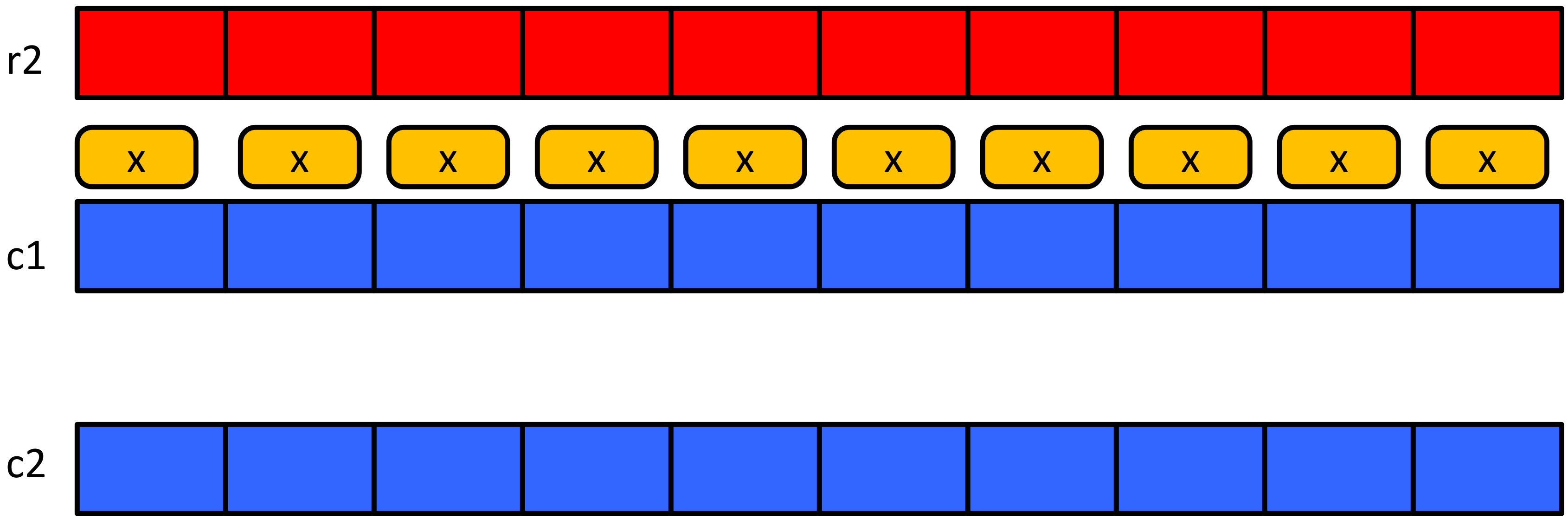
$$m=th[INTT(c_1*r_2+c_2)]$$

unprotected ring-LWE decryption



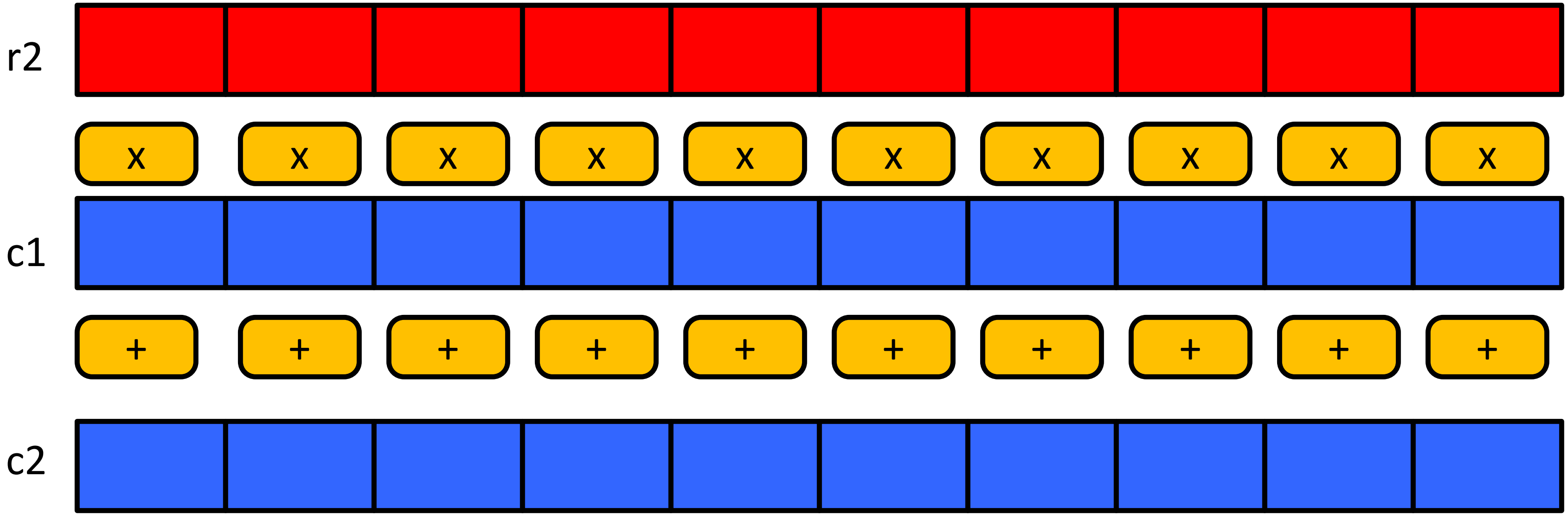
$$m=th[INTT(c_1*r_2+c_2)]$$

<u>**un**</u>protected ring-LWE decryption



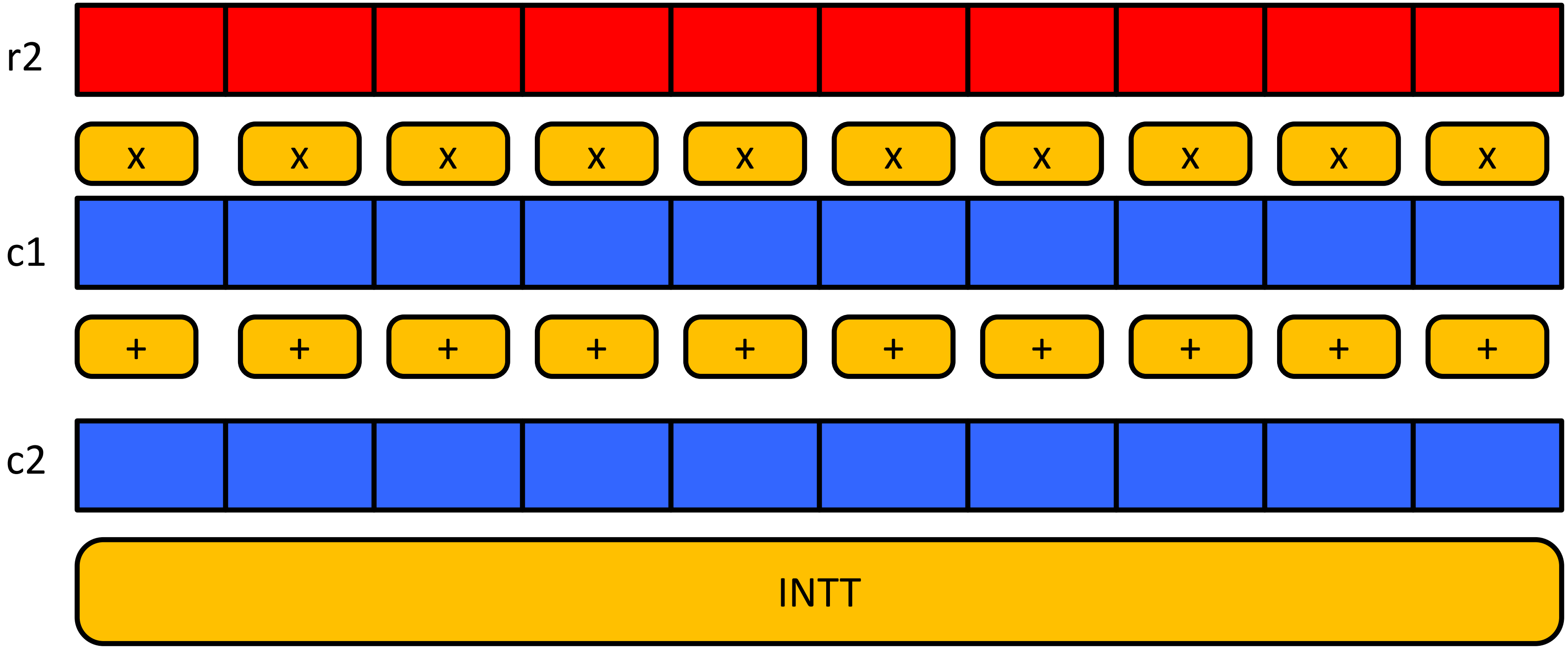
$$m=th[INTT(c_1 * r_2 + c_2)]$$

unprotected ring-LWE decryption



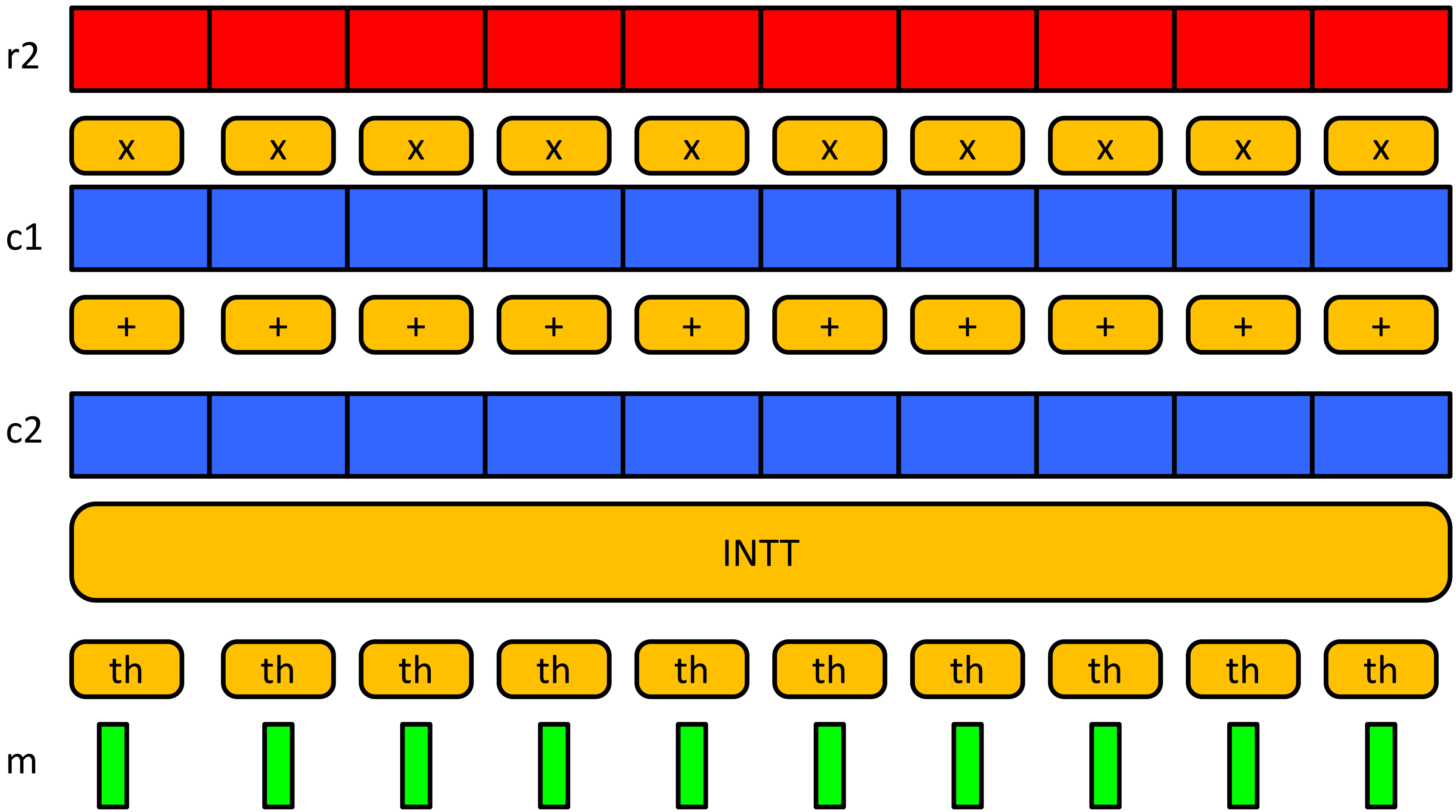
$$m = th[INTT(c_1 * r_2 + c_2)]$$

<u>**un**</u>protected ring-LWE decryption



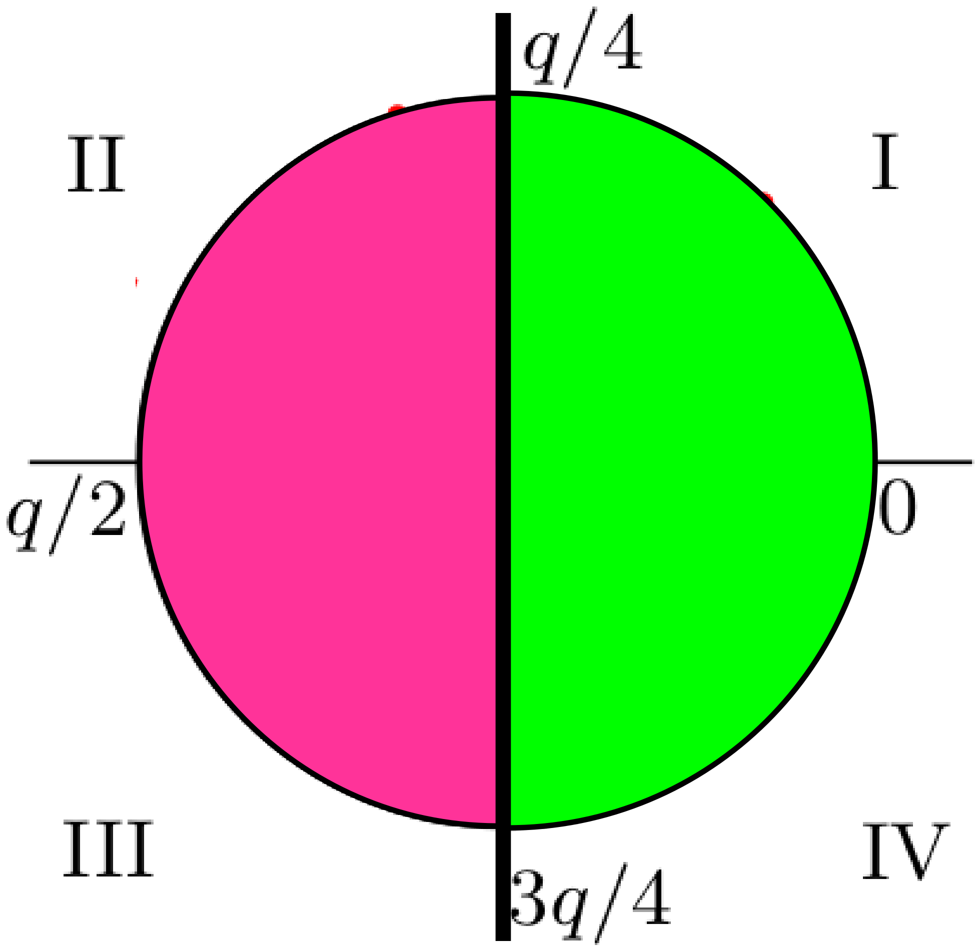
$$m=th[INTT(c_1 * r_2 + c_2)]$$

<u>un</u>protected ring-LWE decryption



$$m=th[INTT(c_1*r_2+c_2)]$$

th operation



masking ring-LWE

- Core idea: split the secret: r=r’+r’’

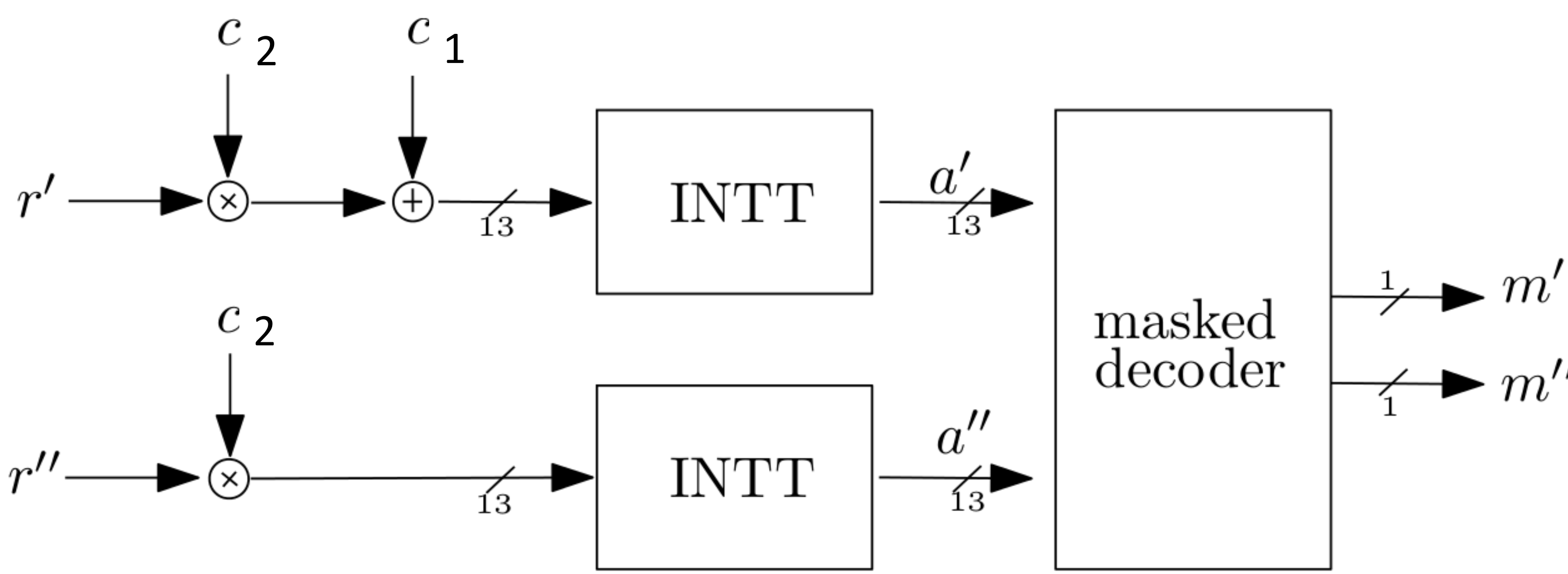
$$\mathrm{INTT}(r \cdot c_2 + c_1) = \mathrm{INTT}(r' \cdot c_2 + c_1) + \mathrm{INTT}(r'' \cdot c_2).$$

$m=th[\mathrm{INTT}(c_1 * r_2 + c_2)]$

masking ring-LWE

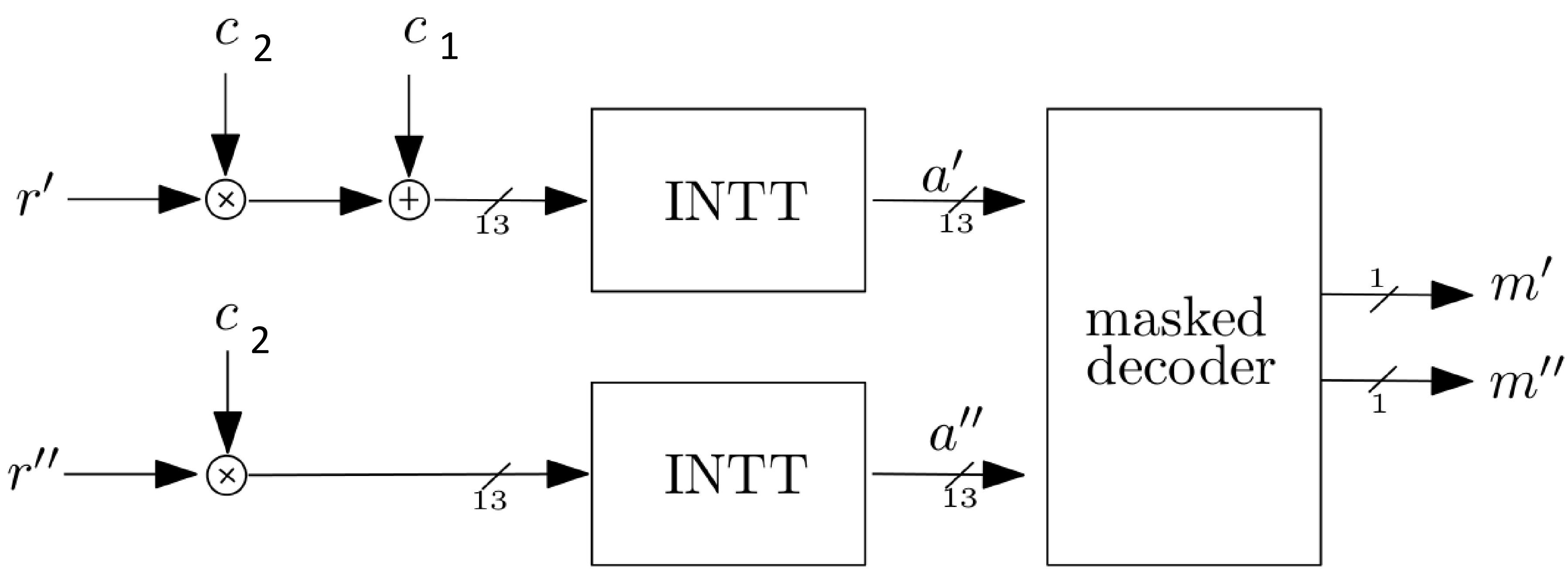
- Core idea: split the secret: r=r'+r''

$$\mathrm{INTT}(r \cdot c_2 + c_1) = \mathrm{INTT}(r' \cdot c_2 + c_1) + \mathrm{INTT}(r'' \cdot c_2).$$

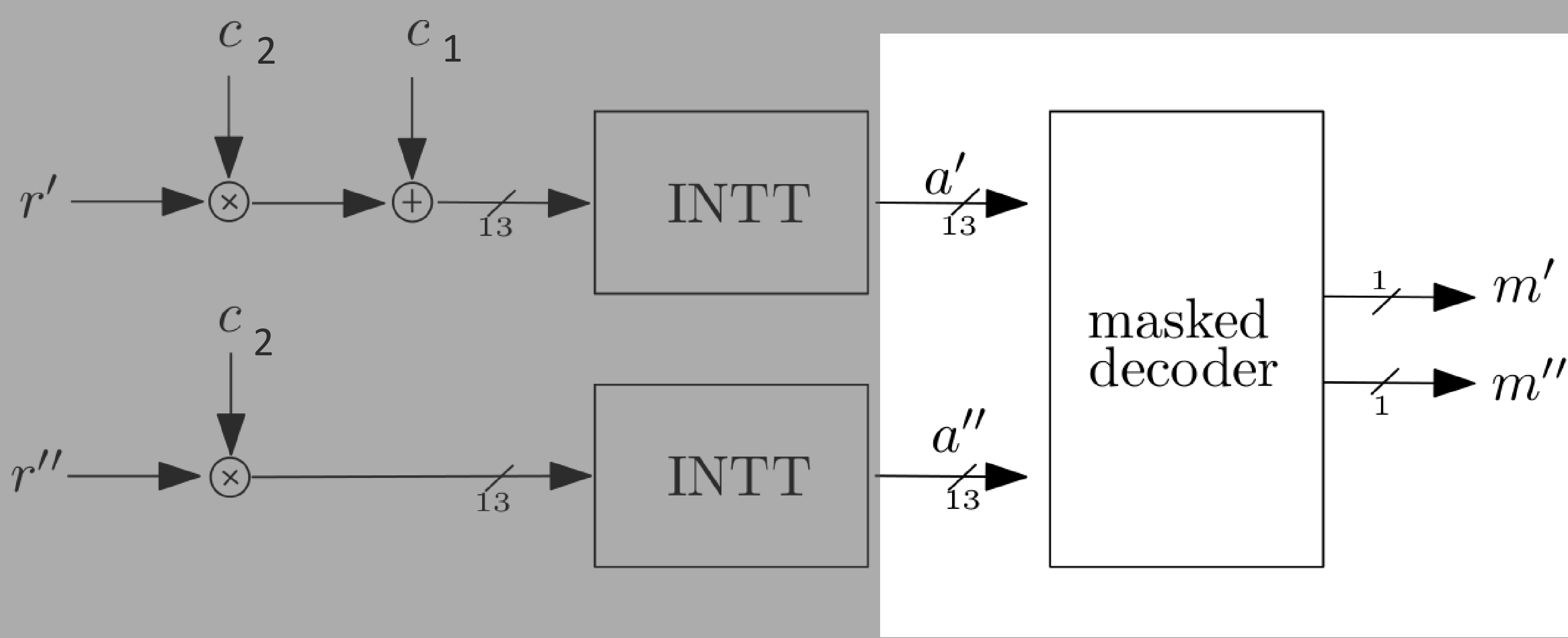


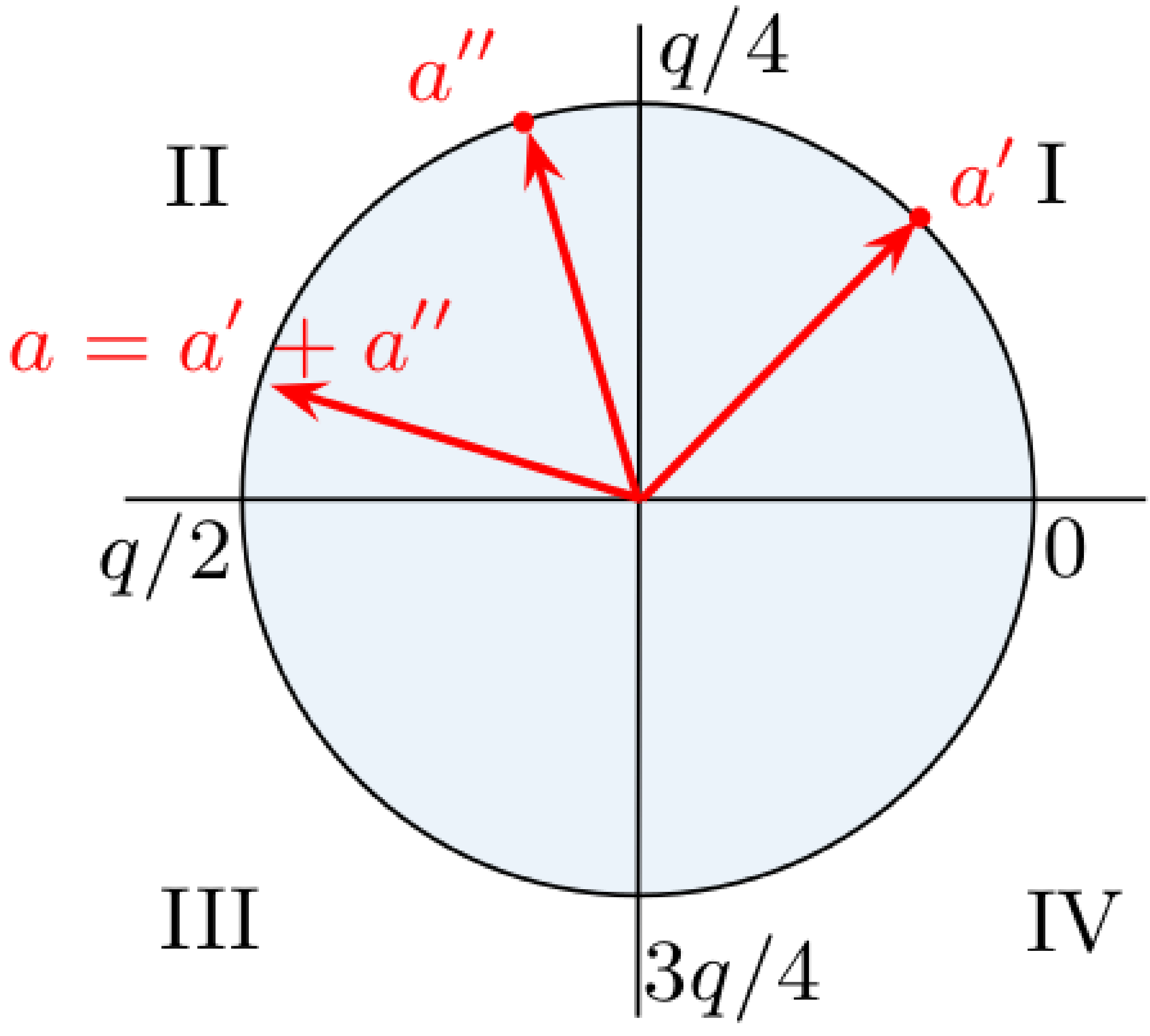
m=th[INTT(c_1*r $_2$+ c $_2$)]

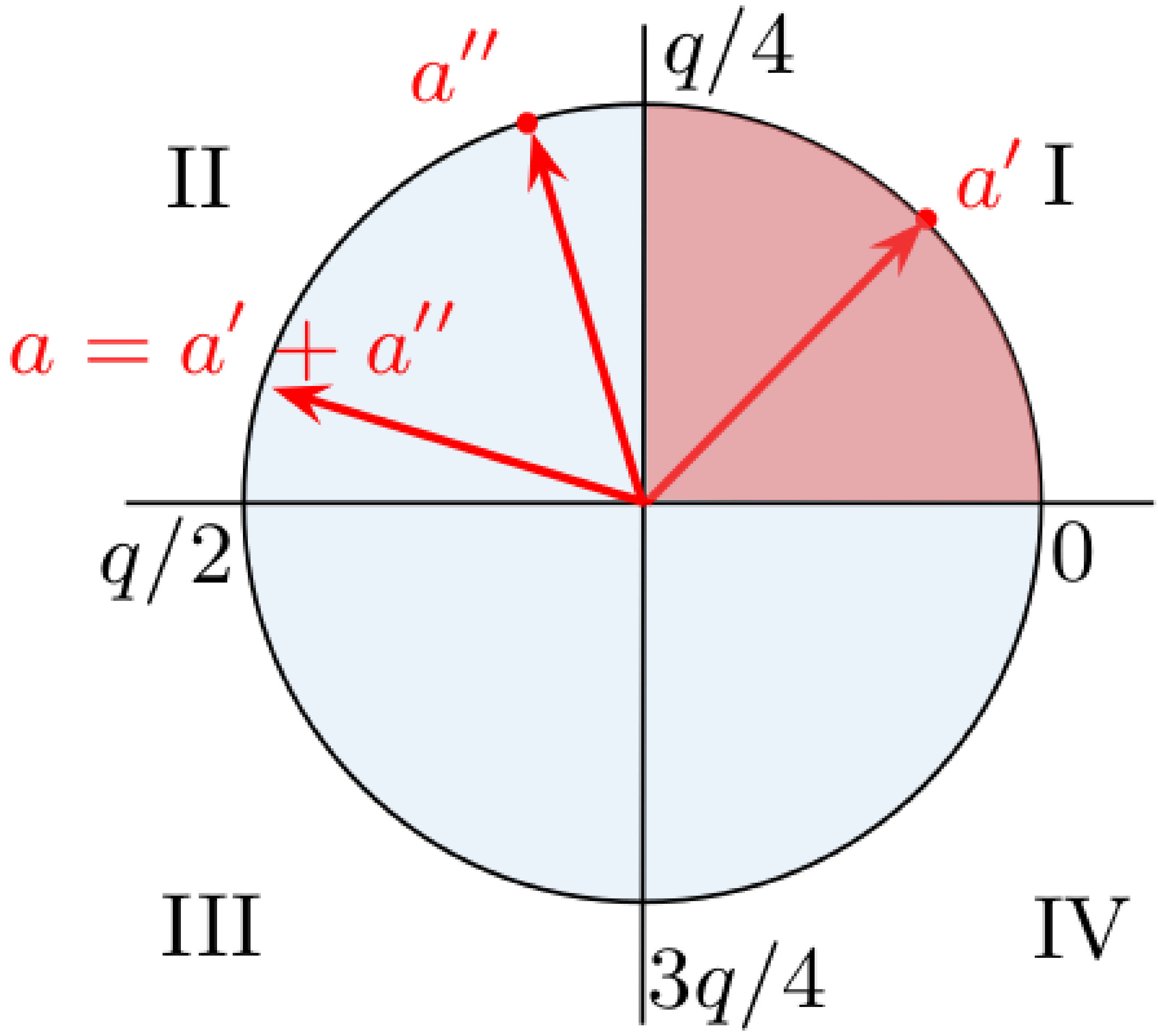
on the masked decoder



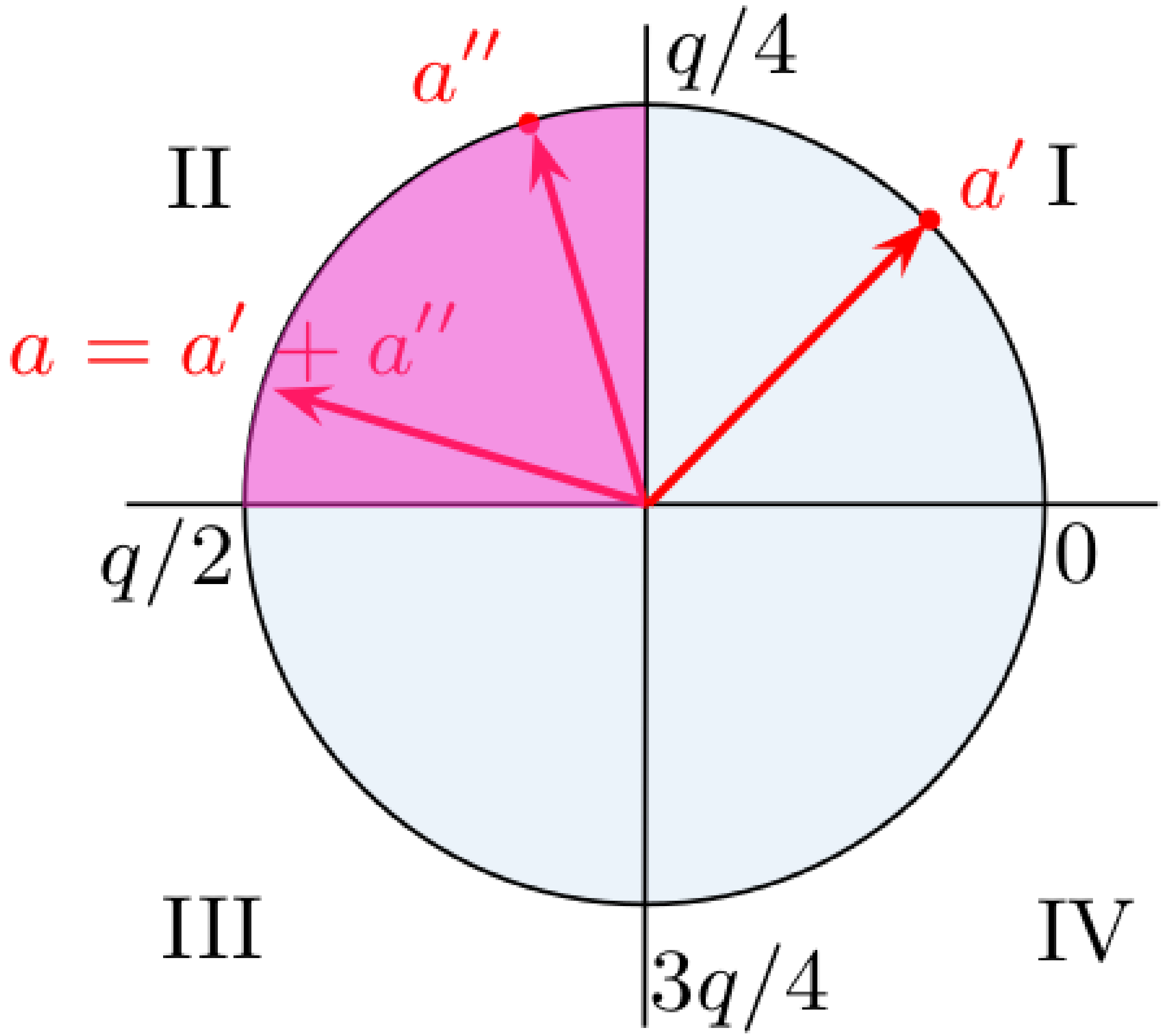
on the masked decoder



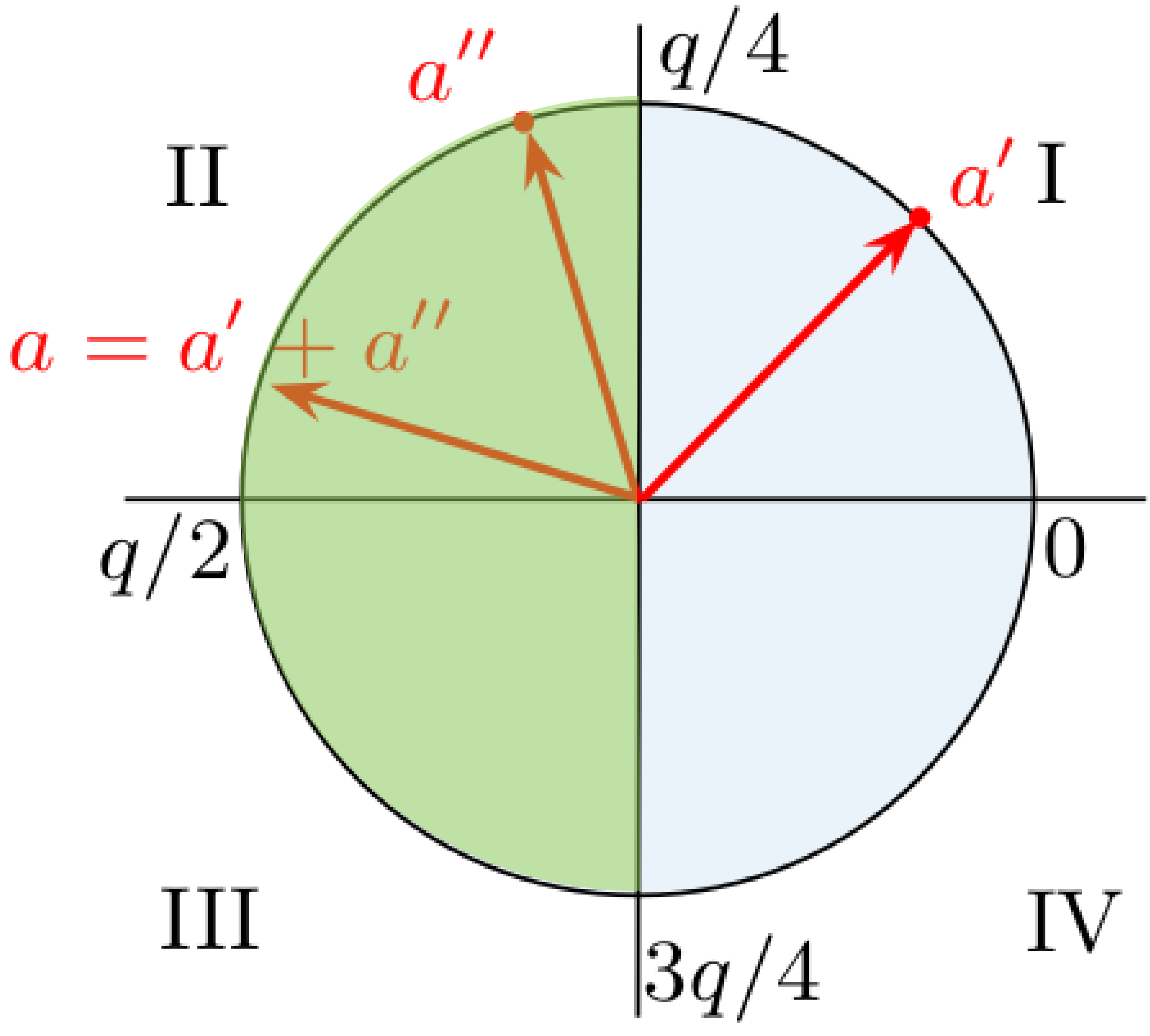

a''
$q/4$
II
a'
I
$a = a' + a''$
$q/2$
0
III
IV
$3q/4$



$q/4$
a''
II
a'
I
$a = a' + a''$
$q/2$
0
III
IV
$3q/4$



$q/4$
a''
II
a'
I
$a = a' + a''$
$q/2$
0
III
IV
$3q/4$

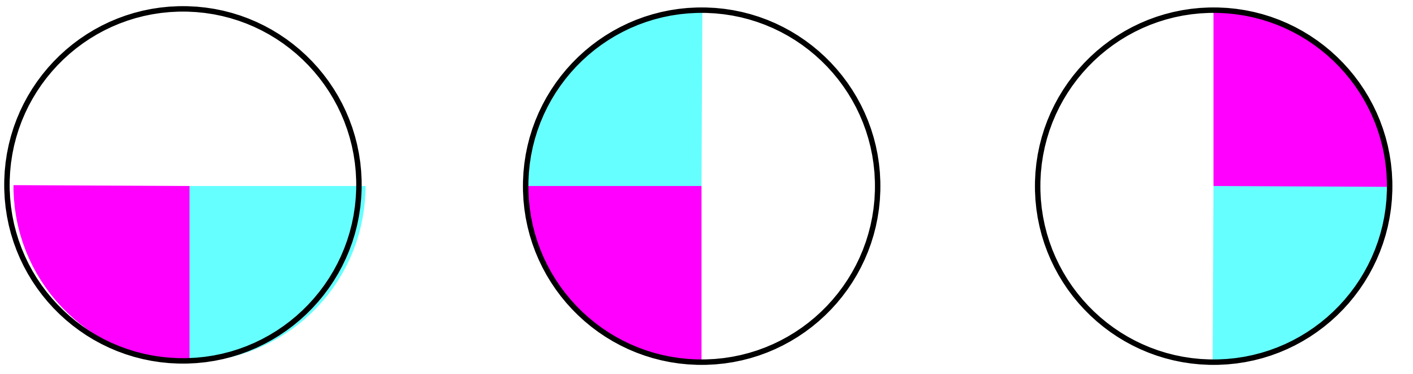

$q/4$
a''
II
a'
I
$a = a' + a''$
$q/2$
0
III
IV
$3q/4$

what happened?

- could decode th(a) from quad(a’) and quad(a’’)
  - quad() return only 2 bits, so it will be easy to perform masked computation.
- Idea: decode th(a) only from quad(a’) and quad(a’’)
  - large compression

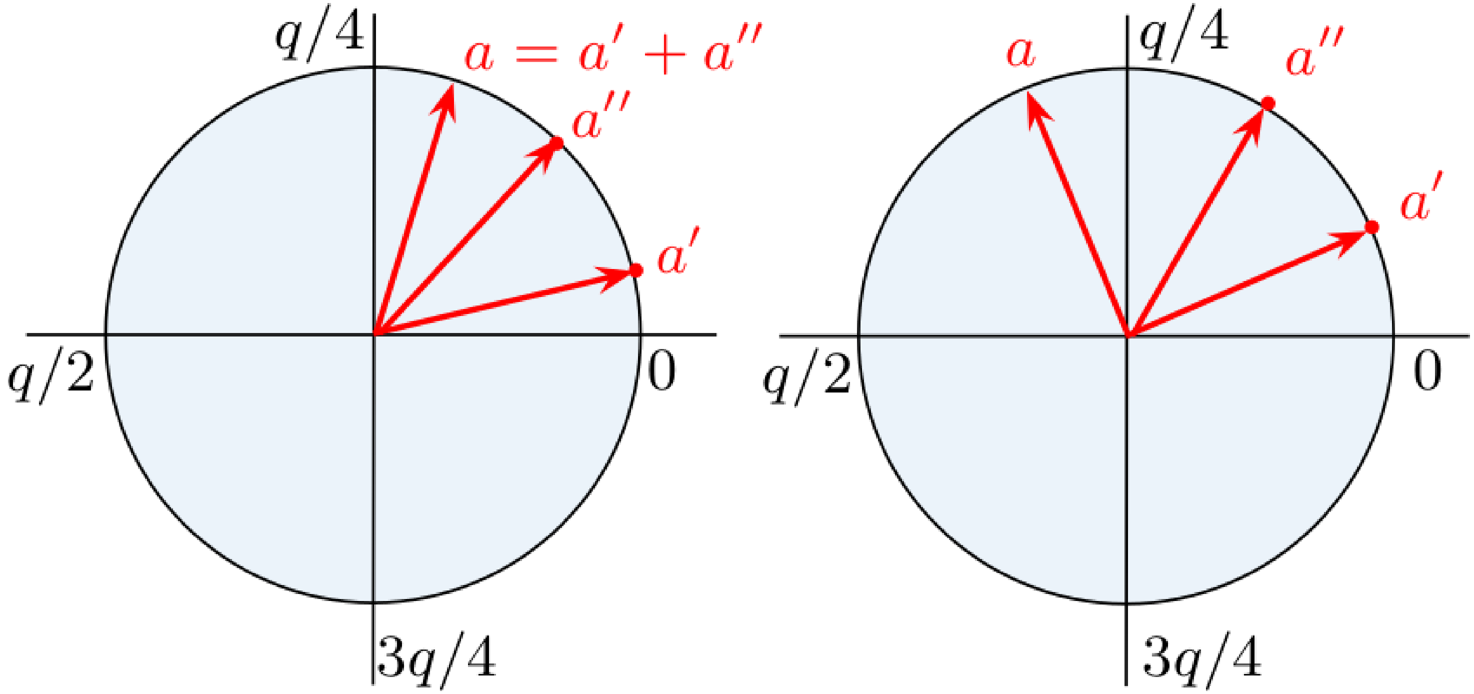
decoding rules

- There are 7 other more cases (“rules”)



- There are 8 cases that don’t allow inferring th(a)!

Cases where it fails



solution: refresh

- Refresh the sharing:

  a'   := a' + D

  a'' := a'' – D

And try again

- Do not draw D from random, compute nice ones.

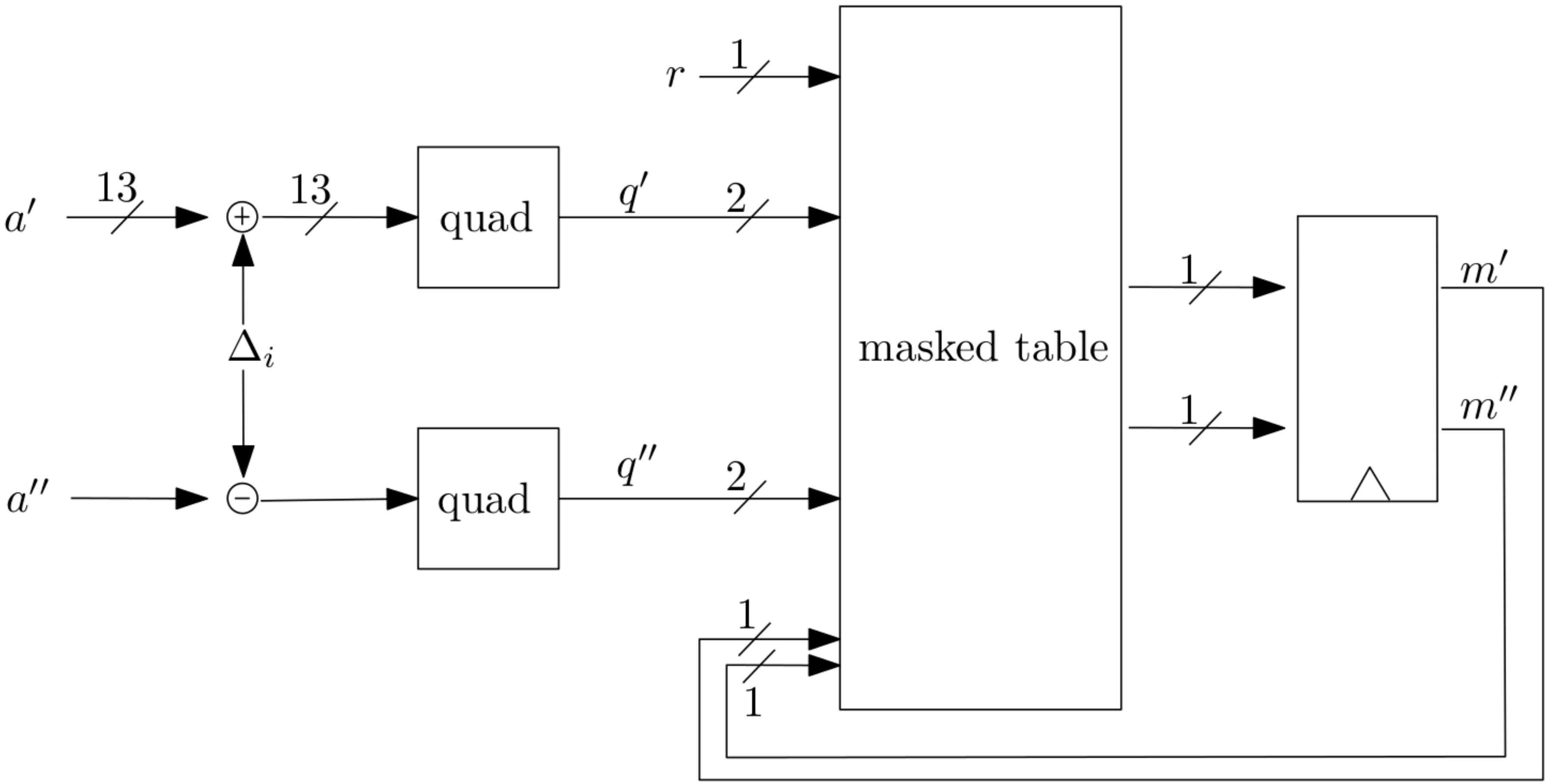


Fig. 3: The masked decoder.

implementation costs

unprotected (CHES2014*)

- 1713 LUTs / 830 FFs / 1 DSP
- Fmax = 120 MHz

protected (this work)

- 2014 LUTs / 959 FFs / 1 DSP
- 100 MHz

Parameter set: (n,q,s)=(256,7681,11.32)
Xilinx Virtex-II xc2vp7 FPGA

* Synthetized on Virtex-II

implementation costs

unprotected (CHES2014*)

- 1713 LUTs / 830 FFs / 1 DSP
- Fmax = 120 MHz
- 2.8 k cycles (23.5 us)

protected (this work)

- 2014 LUTs / 959 FFs / 1 DSP
- 100 MHz
- 7.5 k cycles (75.2 us)

Parameter set: (n,q,s)=(256,7681,11.32)
Xilinx Virtex-II xc2vp7 FPGA

* Synthetized on Virtex-II

implementation costs

unprotected (CHES2014*)

- 1713 LUTs / 830 FFs / 1 DSP
- Fmax = 120 MHz
- 2.8 k cycles (23.5 us)

protected (this work)

- 2014 LUTs / 959 FFs / 1 DSP
- 100 MHz
- 7.5 k cycles (75.2 us)

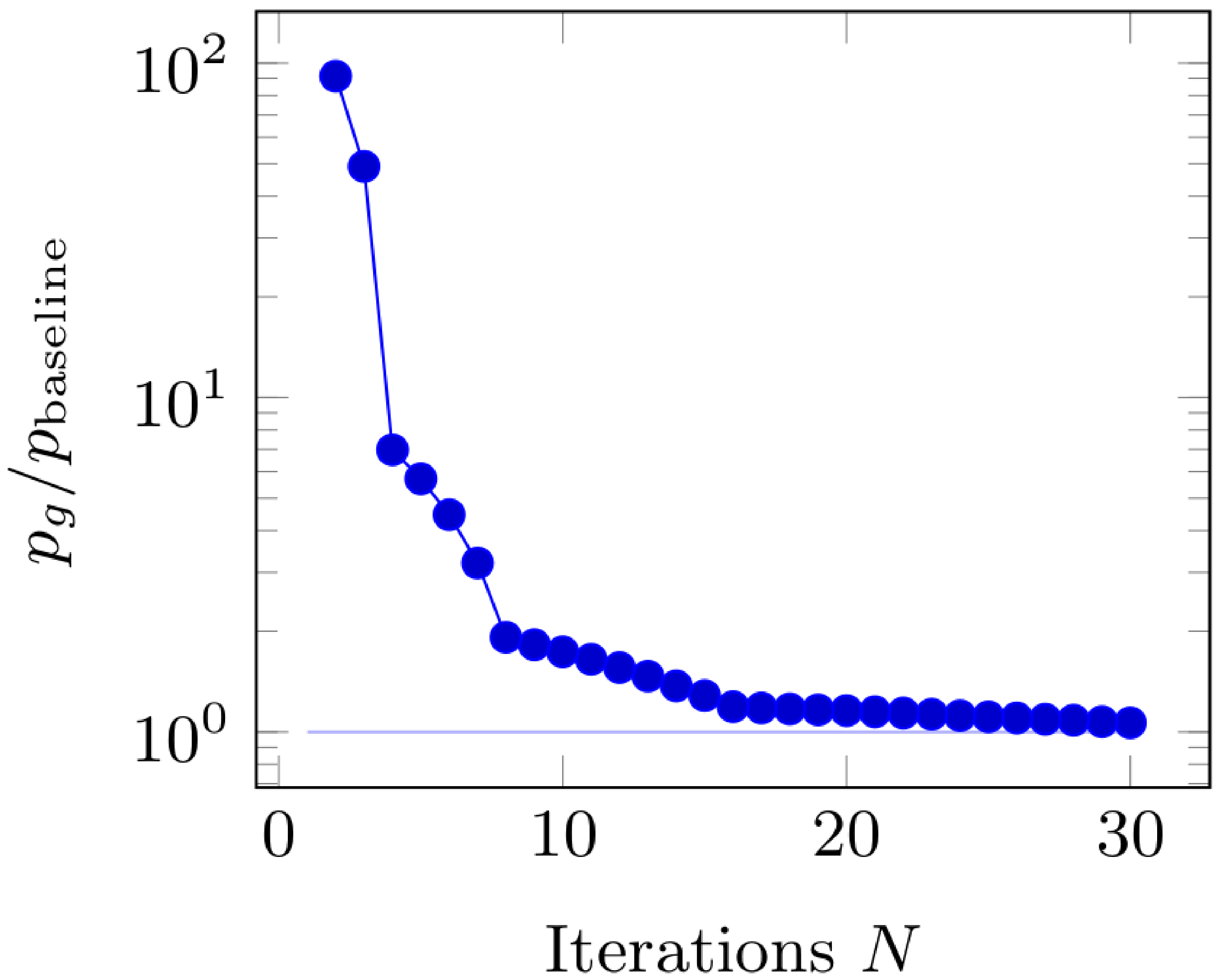
Parameter set: (n,q,s)=(256,7681,11.32)
Xilinx Virtex-II xc2vp7 FPGA

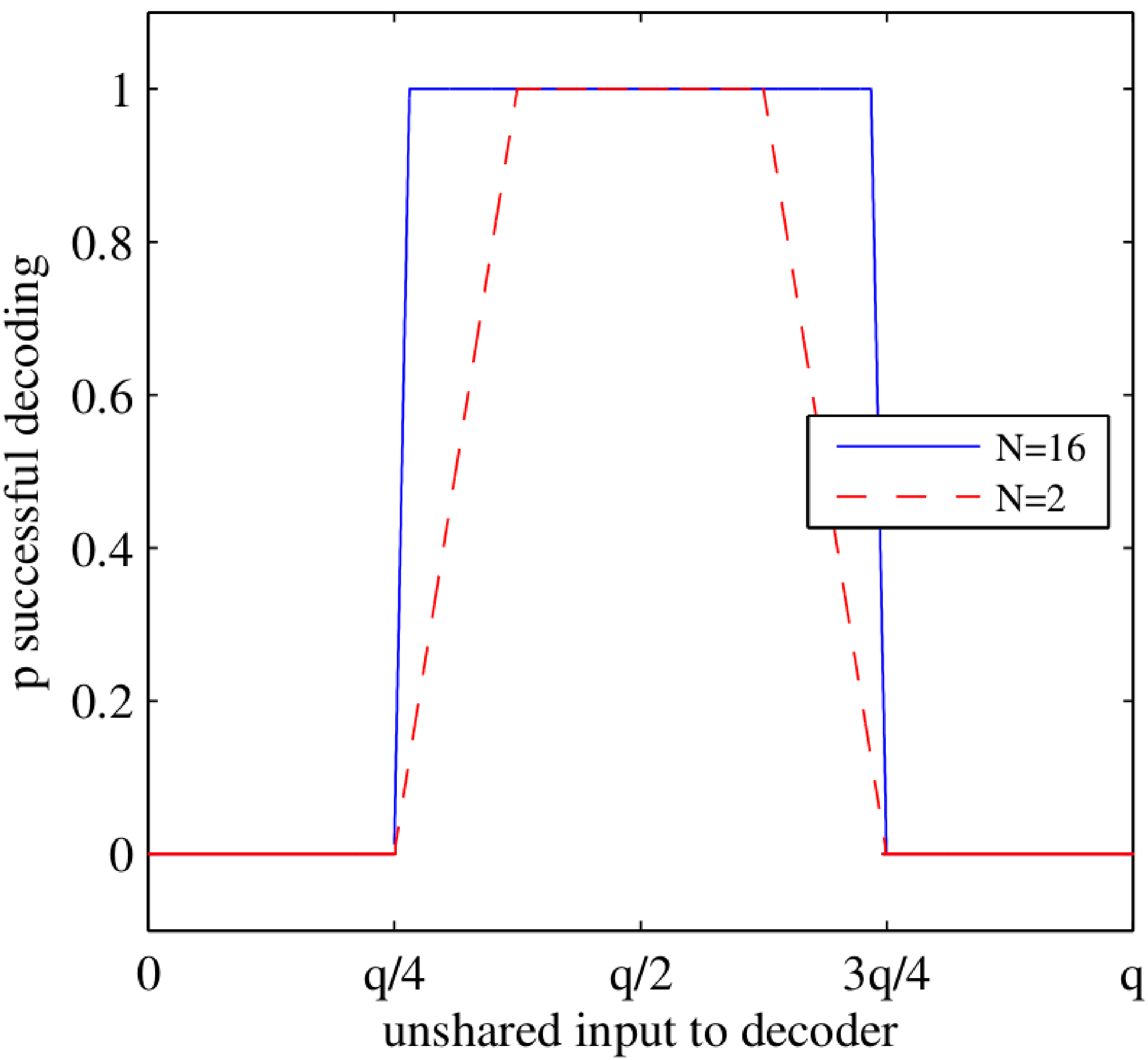
ECC: Rebeiro et.al. (CHES2012): 289 kcycles * LUT
This work: 151 k cycles*LUTs

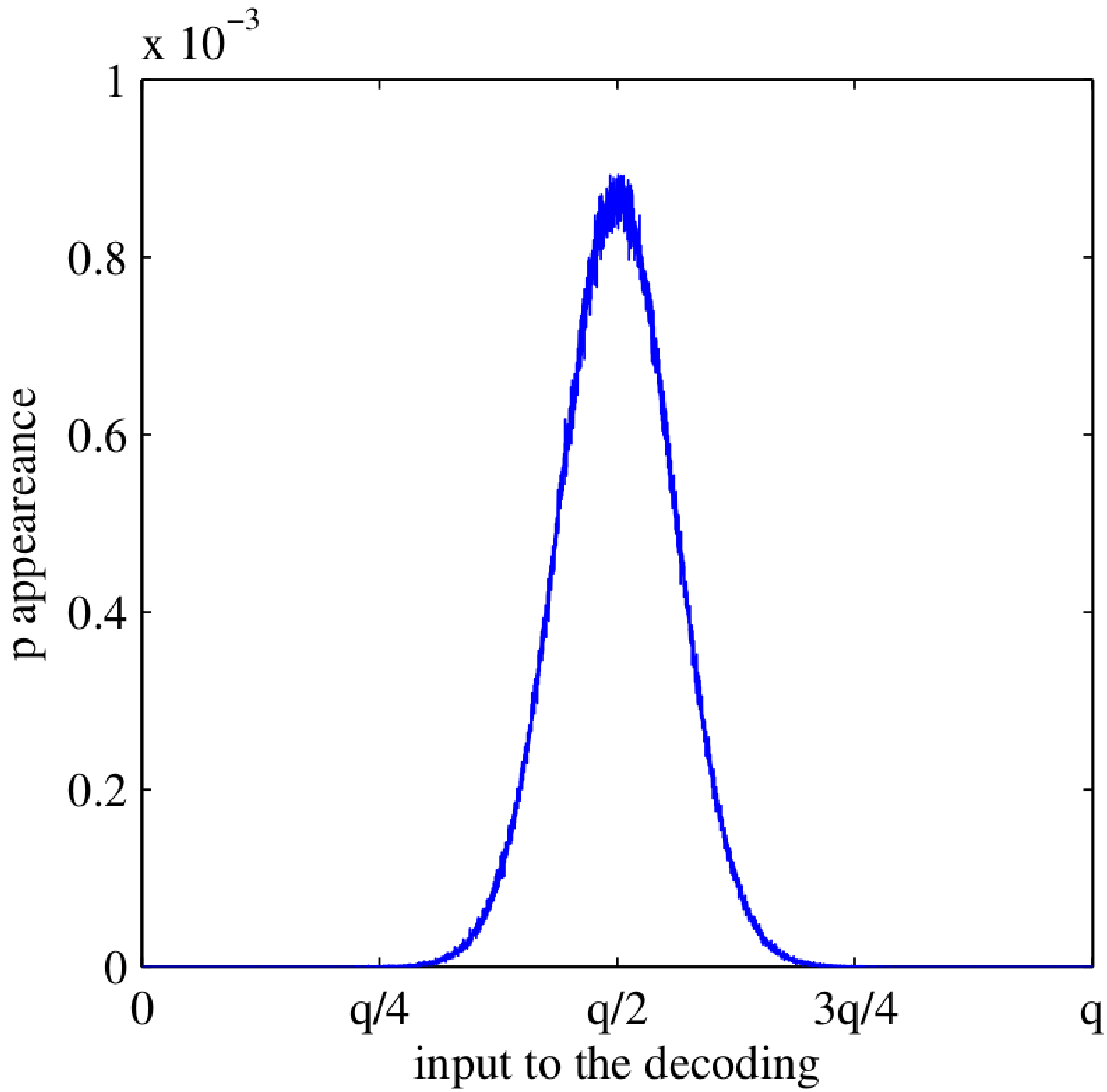
* Synthetized on Virtex-II

error rates

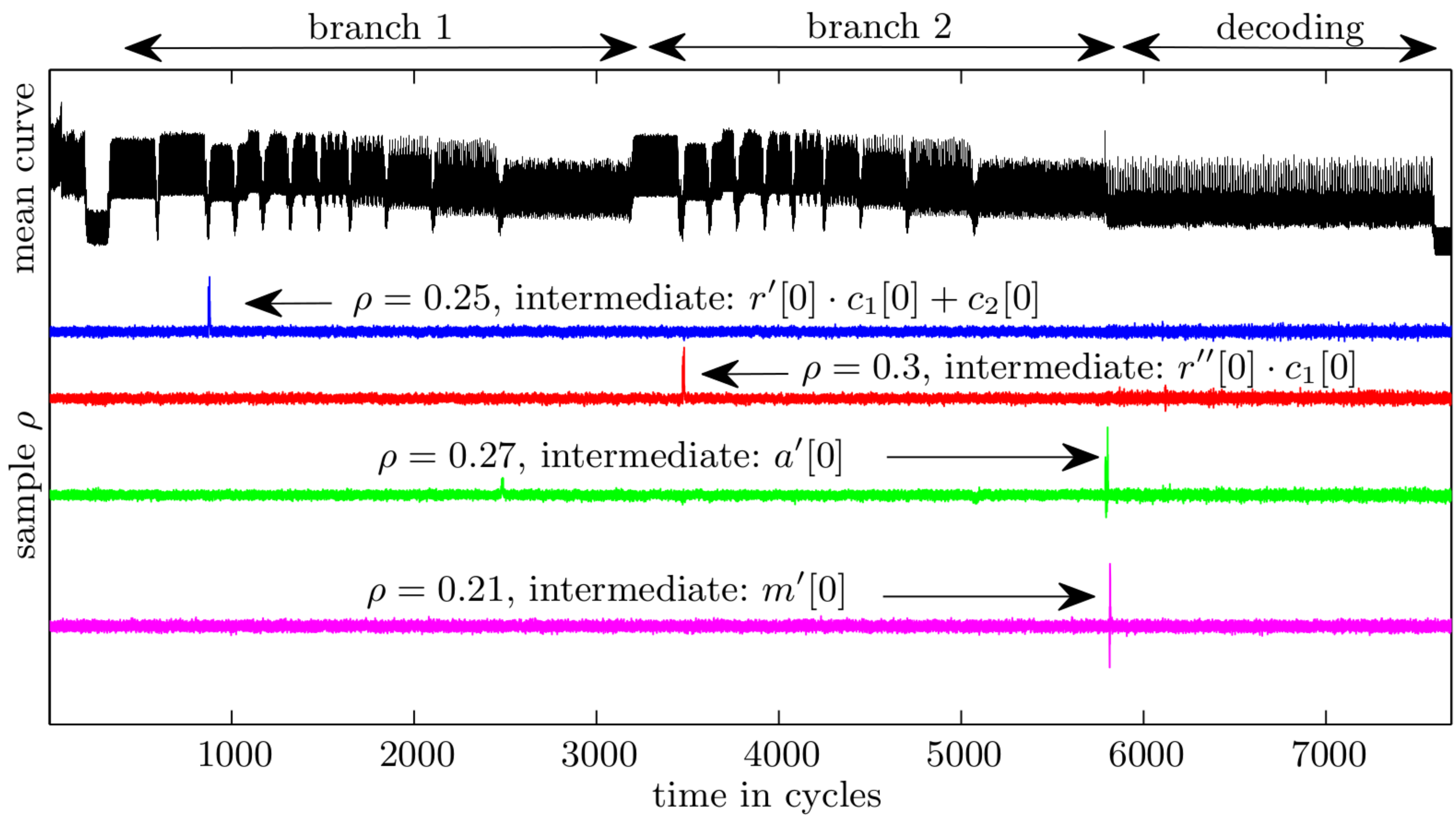
error rates




1
0.8
0.6
0.4
0.2
0
p successful decoding
N=16
N=2
0
q/4
q/2
3q/4
q
unshared input to decoder

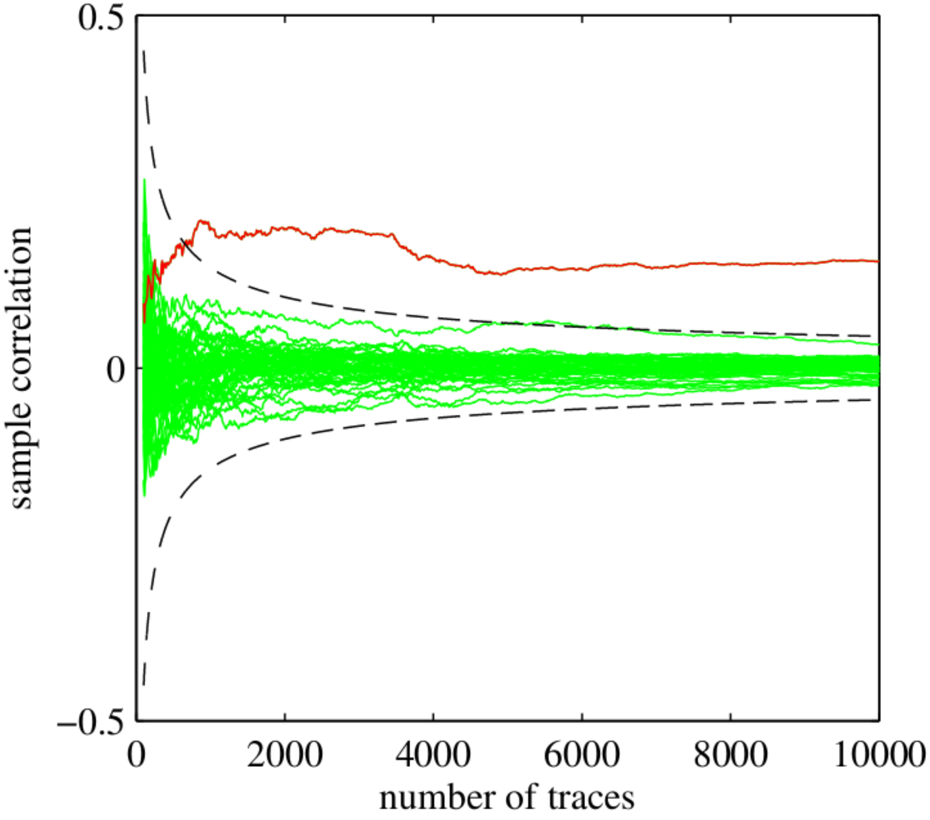

x 10^-3
1
0.8
0.6
0.4
0.2
0
p appeareance
0
q/4
q/2
3q/4
q
input to the decoding

evaluation

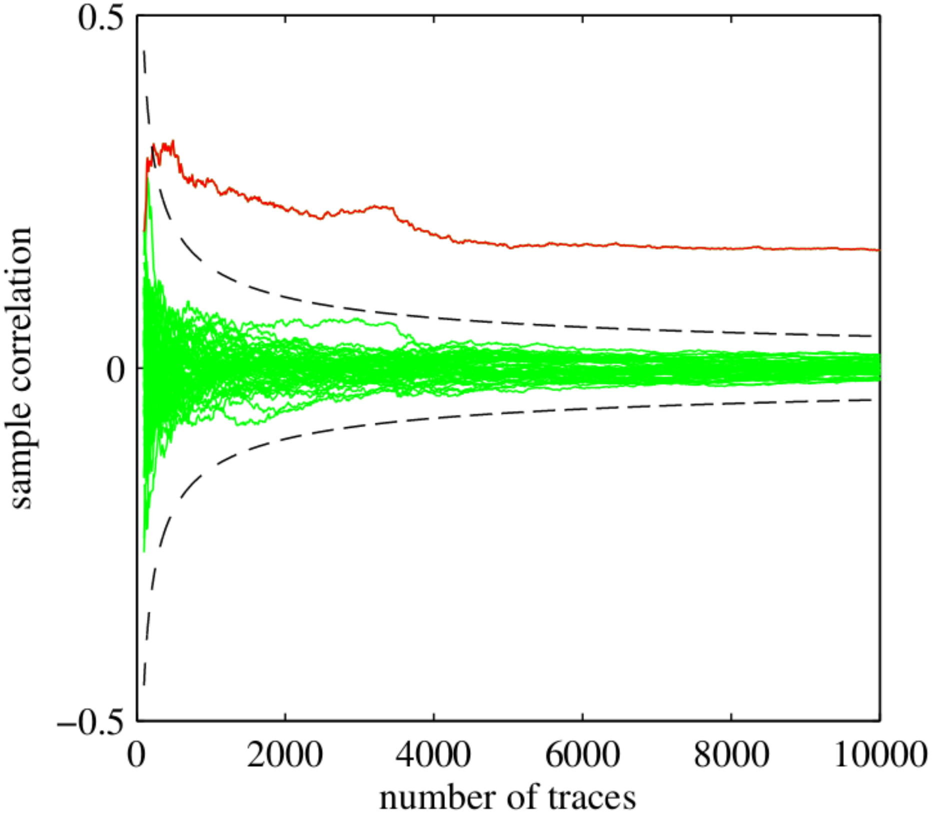


branch 1
branch 2
decoding
mean curve
sample ρ
$\rho = 0.25$, intermediate: $r'[0] \cdot c_1[0] + c_2[0]$
$\rho = 0.3$, intermediate: $r''[0] \cdot c_1[0]$
$\rho = 0.27$, intermediate: $a'[0]$
$\rho = 0.21$, intermediate: $m'[0]$
1000
2000
3000
4000
5000
6000
7000
time in cycles

PRNG off

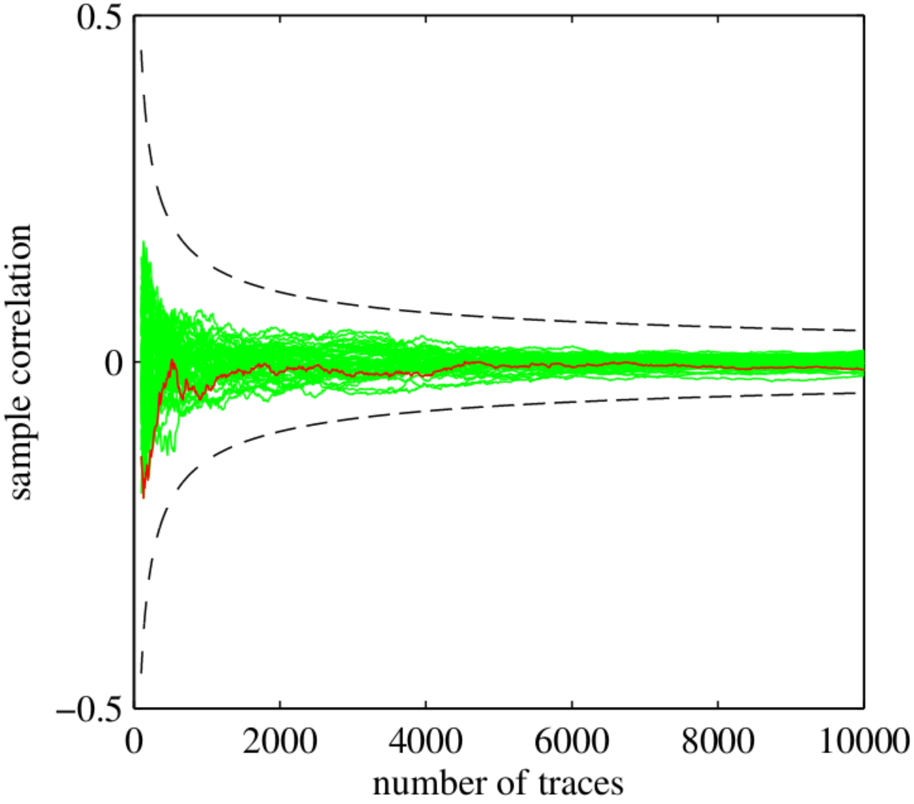


0.5
0
−0.5
sample correlation
0
2000
4000
6000
8000
10000
number of traces

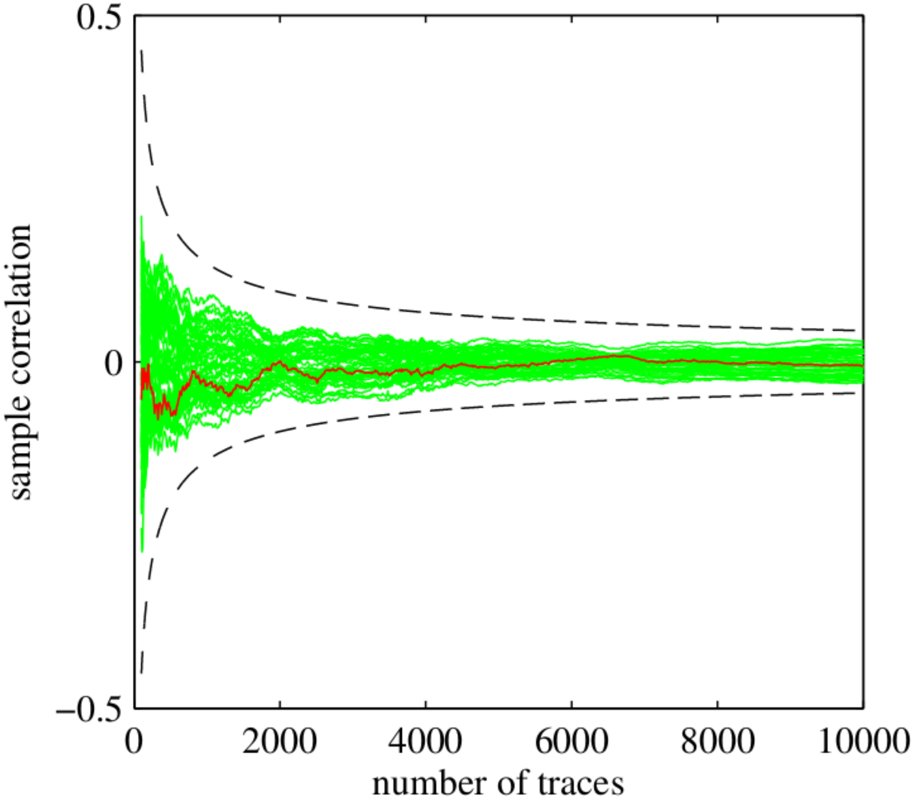


0.5
0
−0.5
sample correlation
0
2000
4000
6000
8000
10000
number of traces

PRNG on

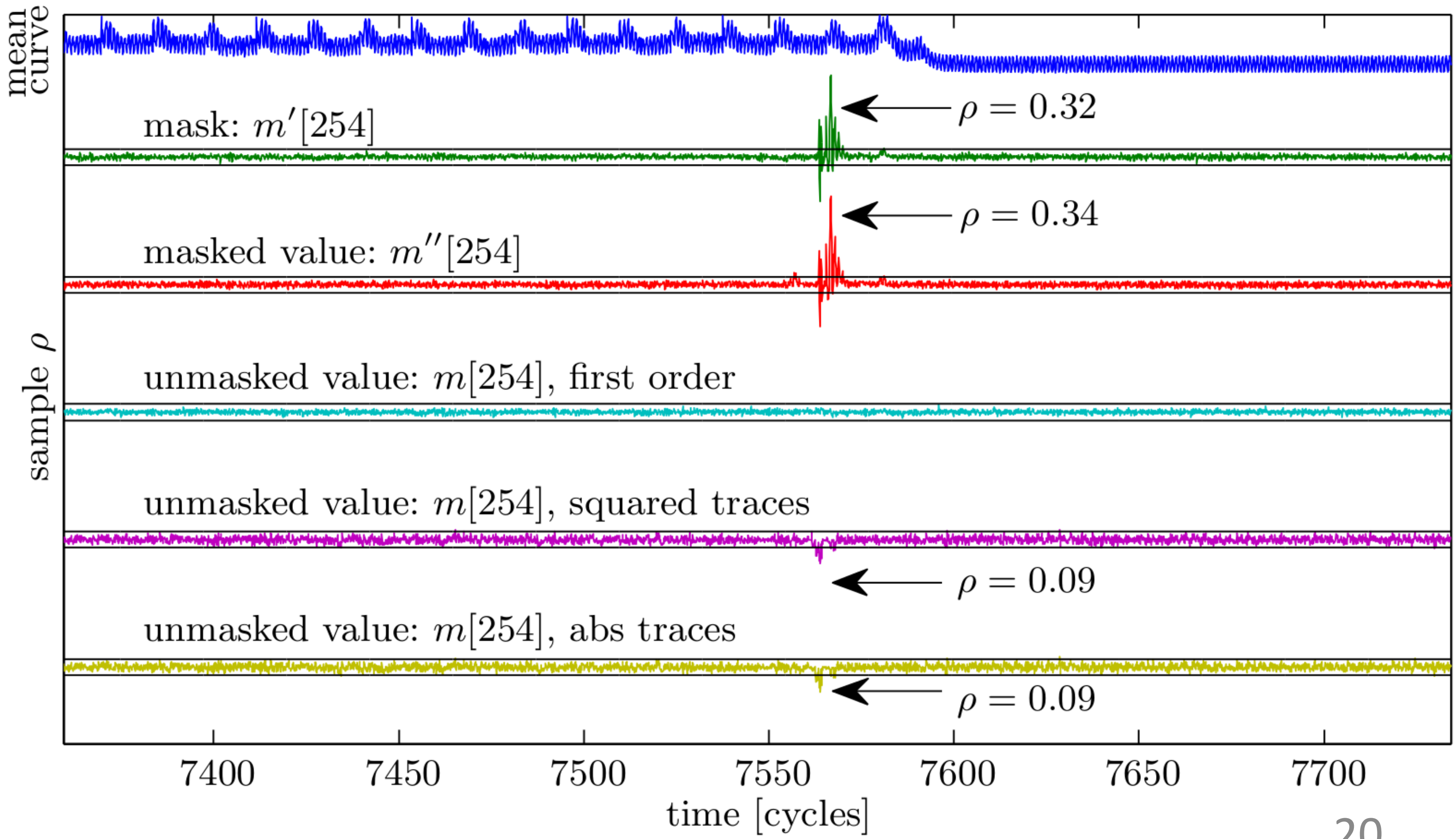


0.5
0
−0.5
sample correlation
0
2000
4000
6000
8000
10000
number of traces

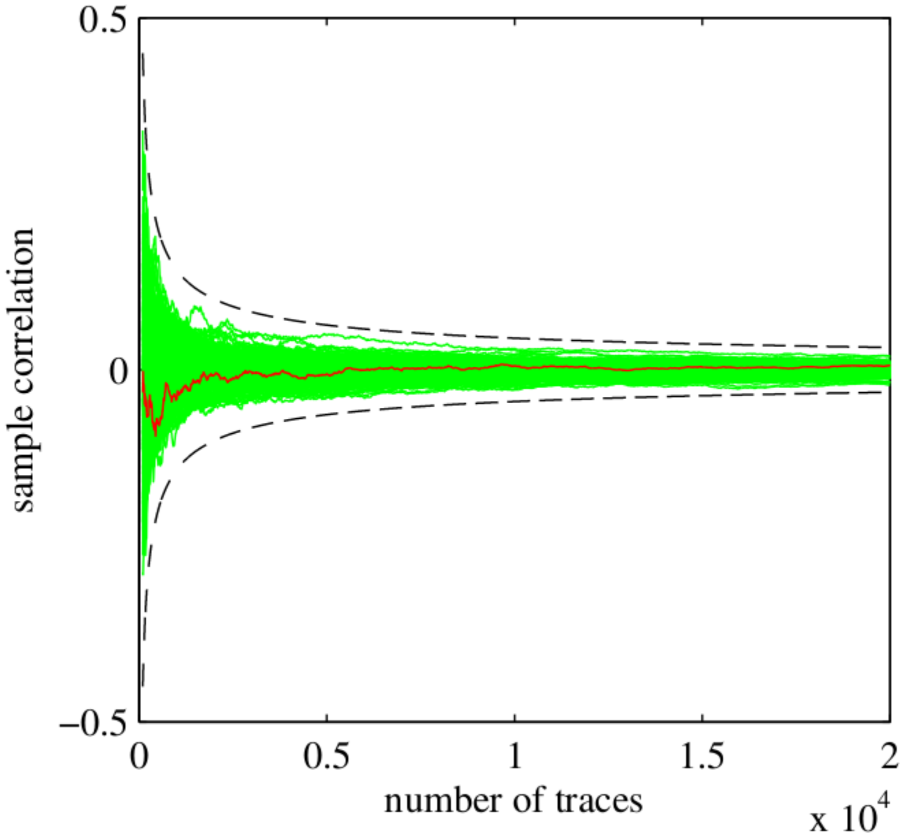


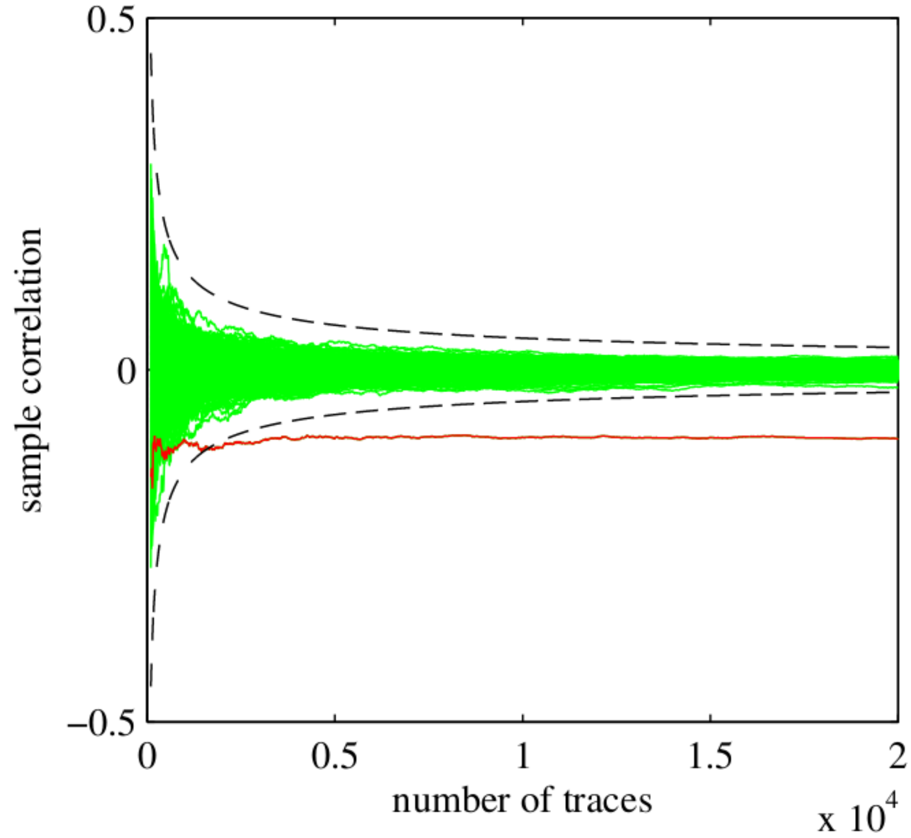
0.5
0
−0.5
sample correlation
0
2000
4000
6000
8000
10000
number of traces

second order



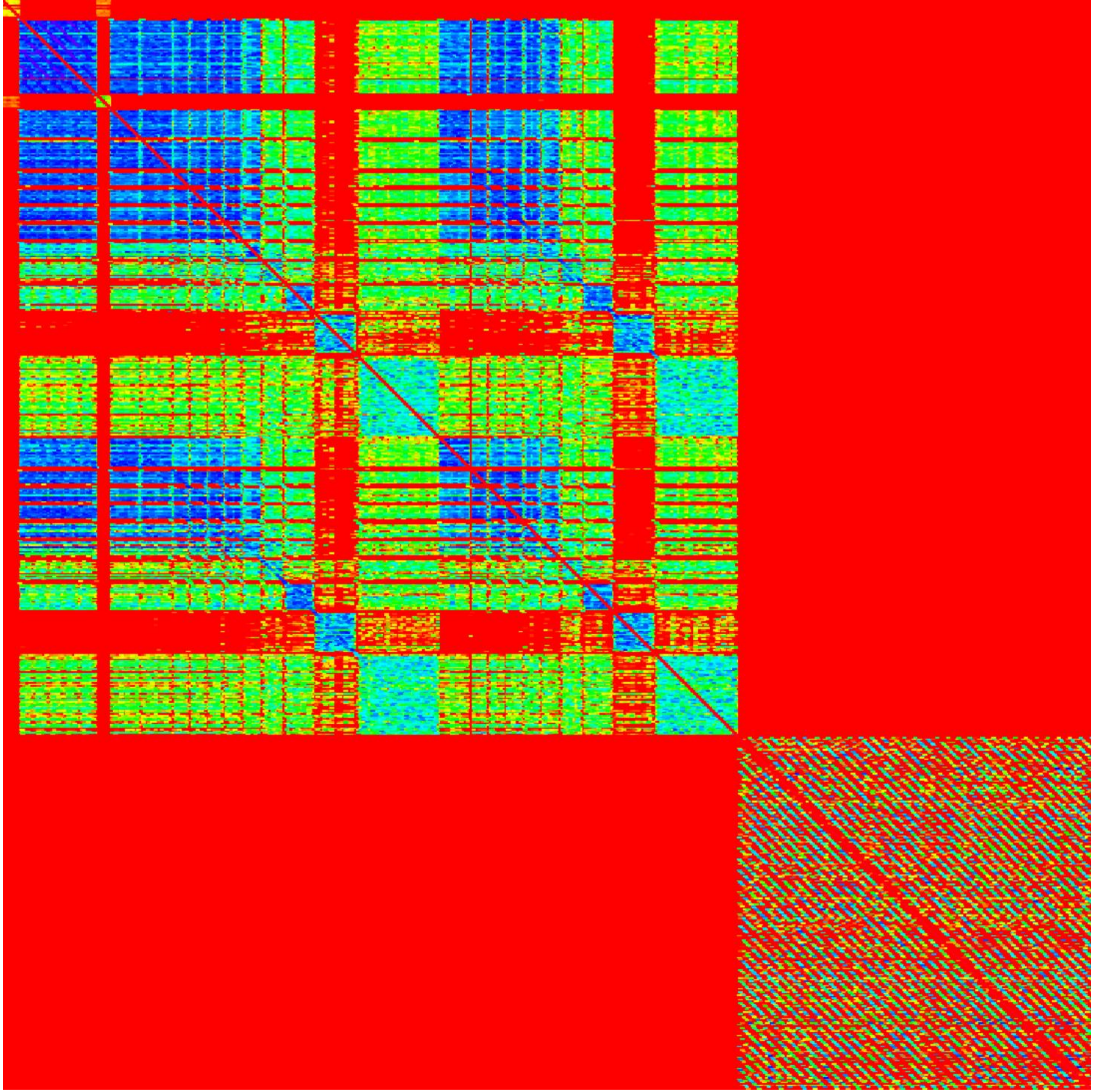
second order

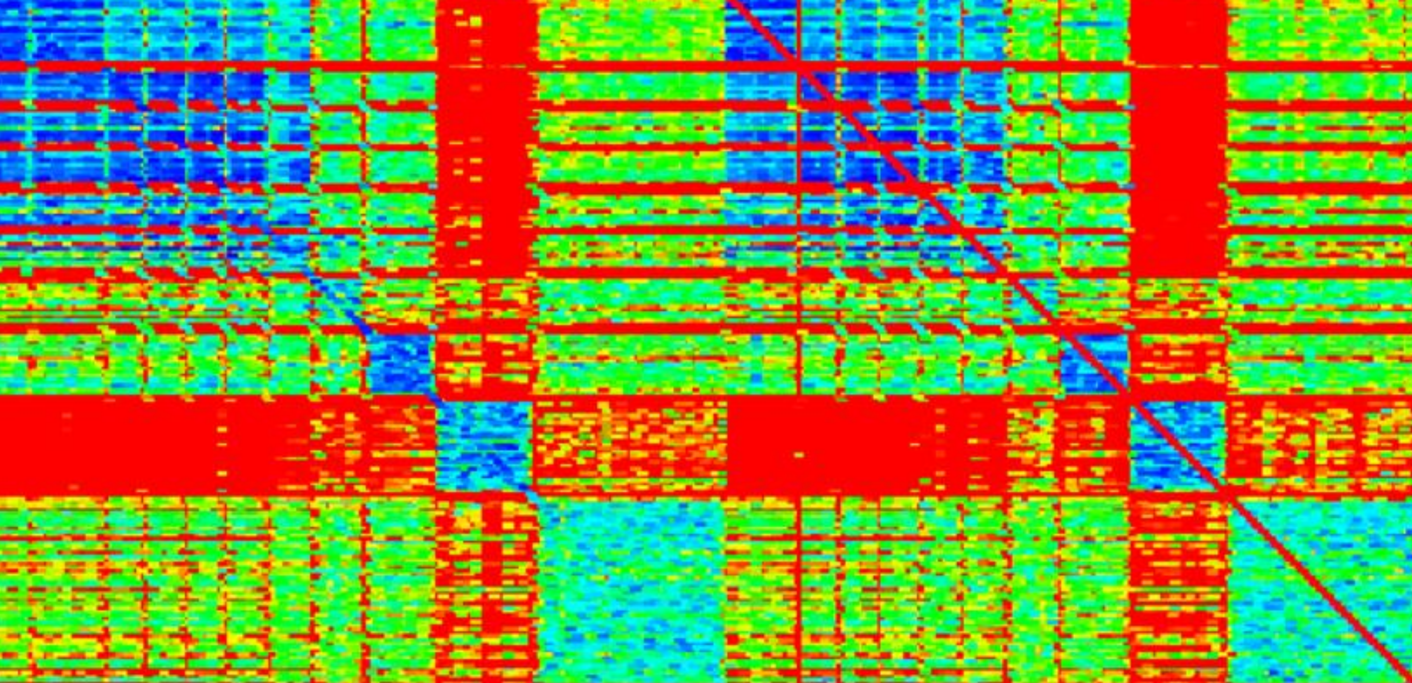




Conclusion

- Fully masked ring-LWE decryption
  - outputs Boolean shares
- Manageable overhead: x2.6 cycles wrt unprotected
- Small!
- Bespoke decoder
  - Error rate controlled
- Practical evaluation





KU LEUVEN

A MASKED RING-LWE IMPLEMENTATION

Oscar Reparaz, Sujoy Sinha Roy,
Frederik Vercauteren, Ingrid Verbauwhede

COSIC/KU Leuven
CHES 2015, Saint-Malo, FR