Aalto University
School of Science

Harnessing Biased Faults in Attacks on ECC-based Signature Schemes

Kimmo Järvinen[1], Céline Blondeau[1], Dan Page[2], Michael Tunstall[2]

[1]Aalto University, Department of Information and Computer Science, Finland
[2]University of Bristol, Department of Computer Science, UK

FDTC 2012, Leuven, Belgium, September 9, 2012

Outline

Background

Existing attacks

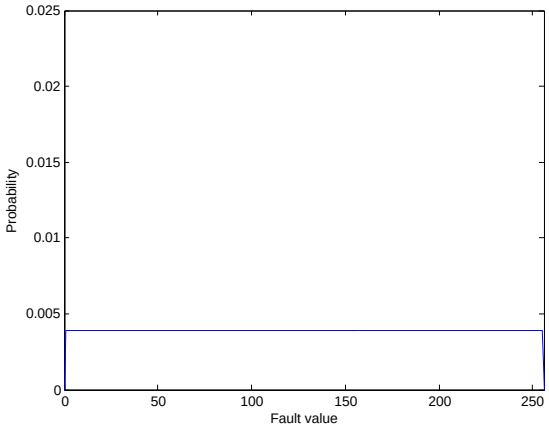
Our attack using biased faults

Results & discussion

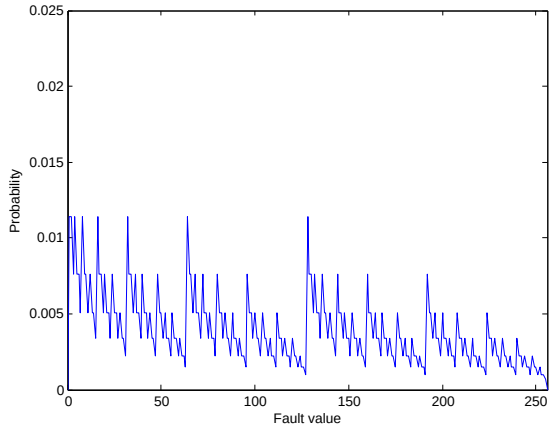
Demo

Introduction

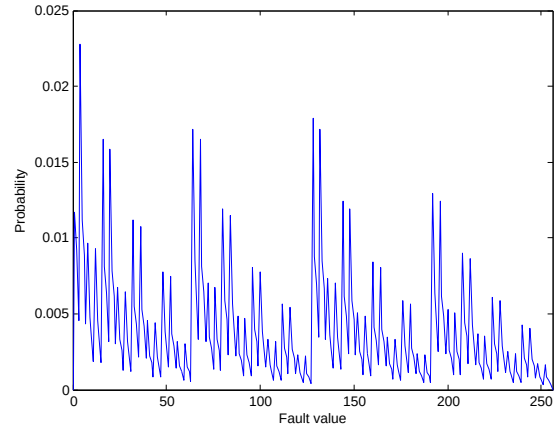
- We build upon the attack presented by Giraud, Knudsen, and Tunstall in ACISP 2004 and CARDIS 2010
- We show that the attack becomes much more powerful if faults are biased (that is, distributed nonuniformly) and the attacker knows or can accurately estimate the biases
- Literature suggests that such phenomena can be produced



$\langle 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5 \rangle$



$\langle 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4 \rangle$



$\langle 0.43, 0.42, 0.32, 0.41, 0.29, 0.49, 0.28, 0.33 \rangle$

Outline of the attack(s)

1. Compute $\mathbf{Q} = d\mathbf{P}$
2. Inject a w-bit fault f into d so $d' = d \oplus (f \cdot 2^m)$
3. Compute $\mathbf{Q}' = d'\mathbf{P}$
4. Calculate $\delta = (d - d')/2^m$ from $\mathbf{Q}$ and $\mathbf{Q}'$ by solving ECDLP $\delta\mathbf{P} = (\mathbf{Q} - \mathbf{Q}')/2^m$
5. Recover information about d using δ (and δ from any previous iterations)
6. Halt if enough information is recovered, otherwise repeat from Step 2

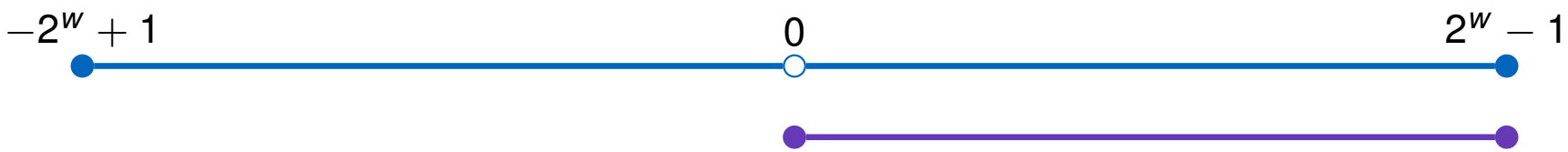
We assume that the attacker has a direct access to $\mathbf{Q}$

The attack of Bao *et al.*

- 1-bit faults
- $\mathbf{Q} - \mathbf{Q}' = (d - d')\mathbf{P} = \begin{cases} -2^m\mathbf{P} & \text{if } d_i = 0 \\ +2^m\mathbf{P} & \text{if } d_i = 1 \end{cases}$
- One fault reveals one key bit
- Difficult fault injection

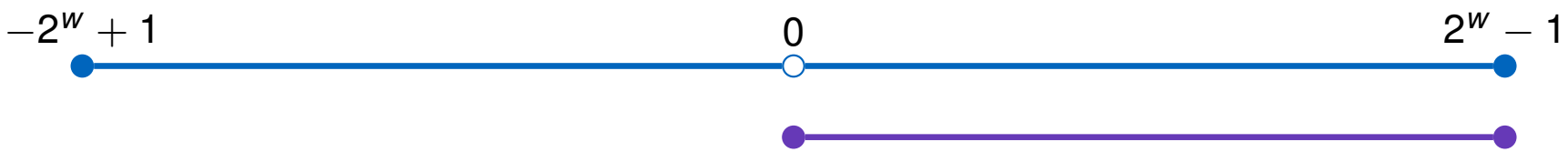
The attack of Giraud *et al.*

- w-bit faults (in their paper: $w = 8$)
- Because $d, d' \in [0, 2^w - 1]$ with $d \neq d'$, for the difference $\delta = d - d'$ we have $\delta \in [-2^w + 1, 2^w - 1] \setminus 0$
- But with a specific fixed d, we have $\delta \in [d - 2^w + 1, d] \setminus 0$



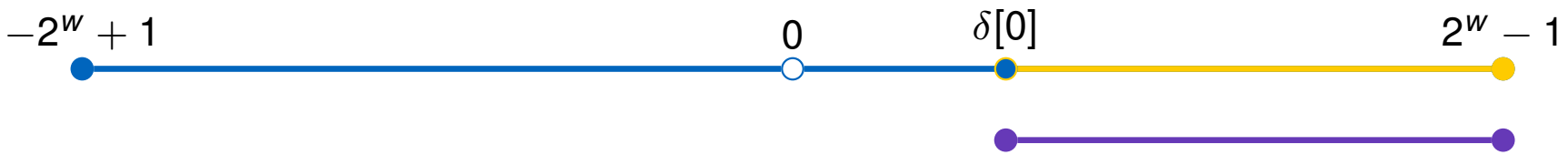
The attack of Giraud *et al.*

- w-bit faults (in their paper: $w = 8$)
- Because $d, d' \in [0, 2^w - 1]$ with $d \neq d'$, for the difference $\delta = d - d'$ we have $\delta \in [-2^w + 1, 2^w - 1] \setminus 0$
- But with a specific fixed d, we have $\delta \in [d - 2^w + 1, d] \setminus 0$
- When we observe δ, we learn information about d: $\max(0, \delta) \leq d \leq \min(2^w - 1, \delta + 2^w - 1)$
- We generate faults until we have enough information



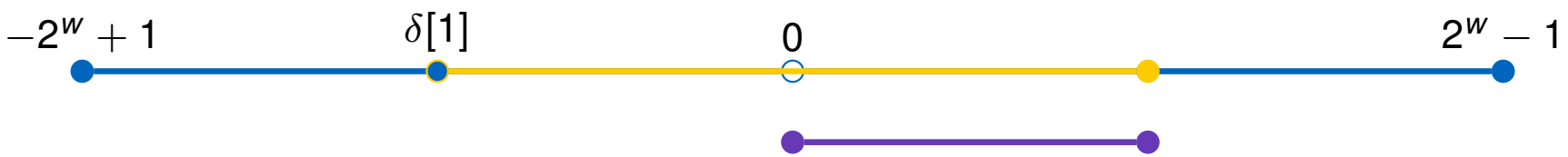
The attack of Giraud *et al.*

- w-bit faults (in their paper: $w = 8$)
- Because $d, d' \in [0, 2^w - 1]$ with $d \neq d'$, for the difference $\delta = d - d'$ we have $\delta \in [-2^w + 1, 2^w - 1] \setminus 0$
- But with a specific fixed d, we have $\delta \in [d - 2^w + 1, d] \setminus 0$
- When we observe δ, we learn information about d: $\max(0, \delta) \leq d \leq \min(2^w - 1, \delta + 2^w - 1)$
- We generate faults until we have enough information



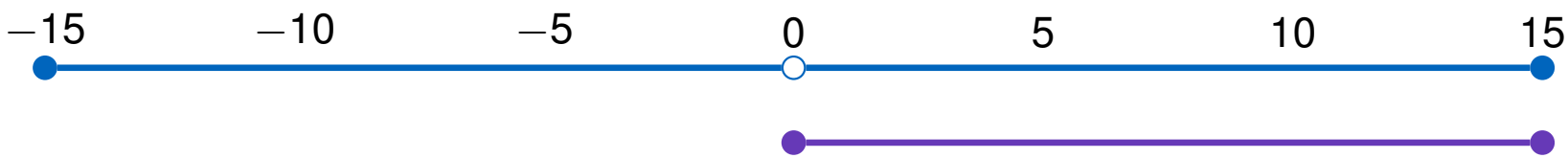
The attack of Giraud *et al.*

- w-bit faults (in their paper: $w = 8$)
- Because $d, d' \in [0, 2^w - 1]$ with $d \neq d'$, for the difference $\delta = d - d'$ we have $\delta \in [-2^w + 1, 2^w - 1] \setminus 0$
- But with a specific fixed d, we have $\delta \in [d - 2^w + 1, d] \setminus 0$
- When we observe δ, we learn information about d: $\max(0, \delta) \leq d \leq \min(2^w - 1, \delta + 2^w - 1)$
- We generate faults until we have enough information



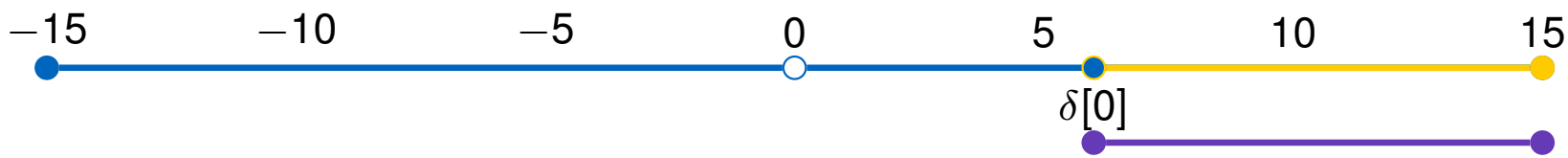
Example: Giraud's attack

N	0
δ	
$d_{\min}$	0
$d_{\max}$	15



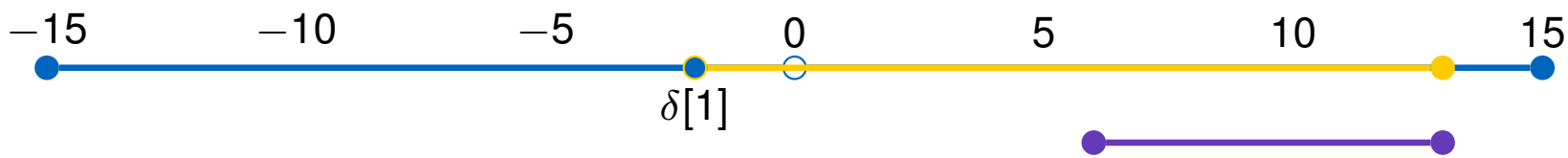
Example: Giraud's attack

N	0	1
δ		6
$d_{\min}$	0	6
$d_{\max}$	15	15



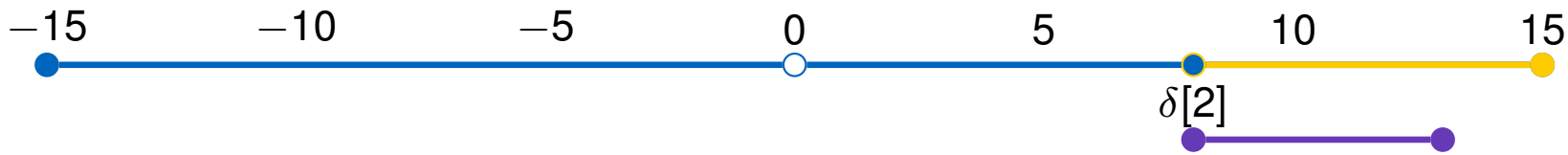
Example: Giraud's attack

N	0	1	2
δ		6	-2
$d_{\min}$	0	6	6
$d_{\max}$	15	15	13



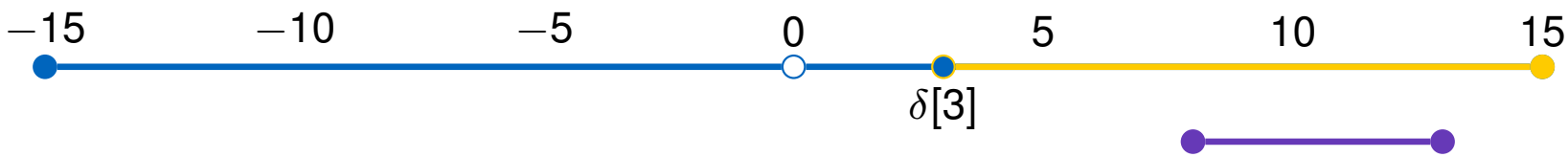
Example: Giraud's attack

N	0	1	2	3
δ		6	-2	8
$d_{\min}$	0	6	6	8
$d_{\max}$	15	15	13	13



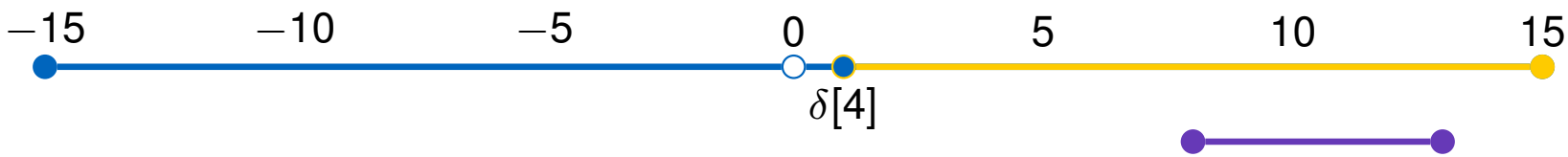
Example: Giraud's attack

N	0	1	2	3	4
δ		6	-2	8	3
$d_{\min}$	0	6	6	8	8
$d_{\max}$	15	15	13	13	13



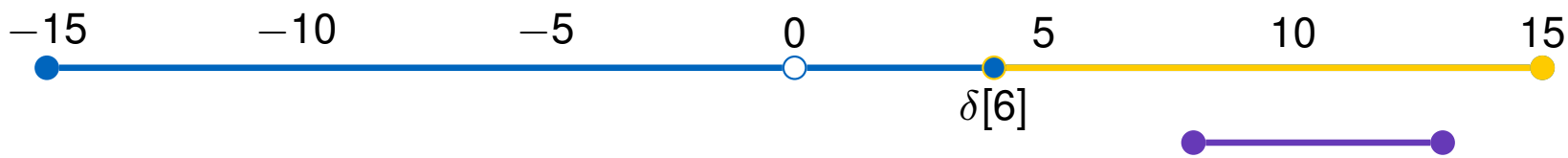
Example: Giraud's attack

N	0	1	2	3	4	5
δ		6	-2	8	3	1
$d_{\min}$	0	6	6	8	8	8
$d_{\max}$	15	15	13	13	13	13



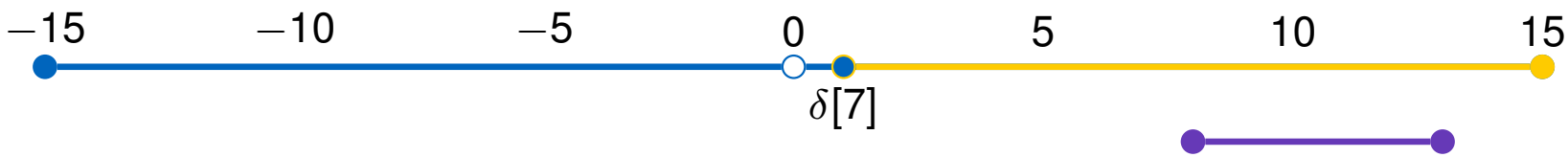
Example: Giraud's attack

N	0	1	2	3	4	5	6
δ		6	-2	8	3	1	4
$d_{\min}$	0	6	6	8	8	8	8
$d_{\max}$	15	15	13	13	13	13	13



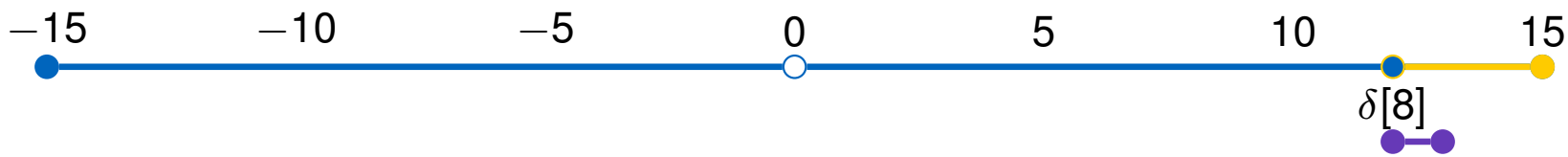
Example: Giraud's attack

N	0	1	2	3	4	5	6	7
δ		6	-2	8	3	1	4	1
$d_{\min}$	0	6	6	8	8	8	8	8
$d_{\max}$	15	15	13	13	13	13	13	13



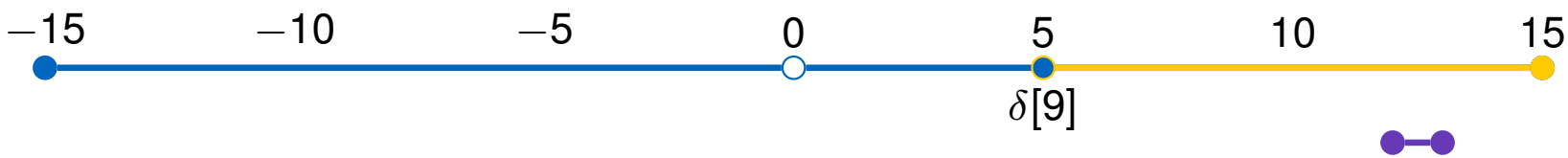
Example: Giraud's attack

N	0	1	2	3	4	5	6	7	8
δ		6	-2	8	3	1	4	1	12
$d_{\min}$	0	6	6	8	8	8	8	8	12
$d_{\max}$	15	15	13	13	13	13	13	13	13



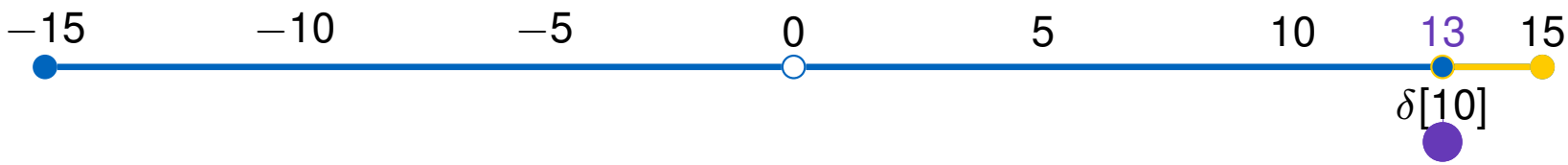
Example: Giraud's attack

N	0	1	2	3	4	5	6	7	8	9
δ		6	-2	8	3	1	4	1	12	5
$d_{\min}$	0	6	6	8	8	8	8	8	12	12
$d_{\max}$	15	15	13	13	13	13	13	13	13	13



Example: Giraud's attack

N	0	1	2	3	4	5	6	7	8	9	10
δ		6	-2	8	3	1	4	1	12	5	13
$d_{\min}$	0	6	6	8	8	8	8	8	12	12	13
$d_{\max}$	15	15	13	13	13	13	13	13	13	13	13



Biased faults

Definition

A fault f is biased iff $\Pr[f = x] \neq |\mathcal{F}|^{-1}$ for some x. That is, some values are more probable than others.

- We consider a bias where the flipping probability of the i^{th} key bit is determined by ϵ_i: $\Pr[f_i = 1] = \frac{1}{2} + \epsilon_i$
- Hence,

$$\Pr[f = x] = \frac{\prod_{i=0}^{w-1} \left(\frac{1}{2} + (-1)^{x_i} \epsilon_i\right)}{1 - \prod_{i=0}^{w-1} \left(\frac{1}{2} - \epsilon_i\right)}$$

- The attack applies also for other kind of biases. For instance, if faults are biased by the values of key bits
- We assume that the attacker knows ϵ_i's

Probability of a key candidate

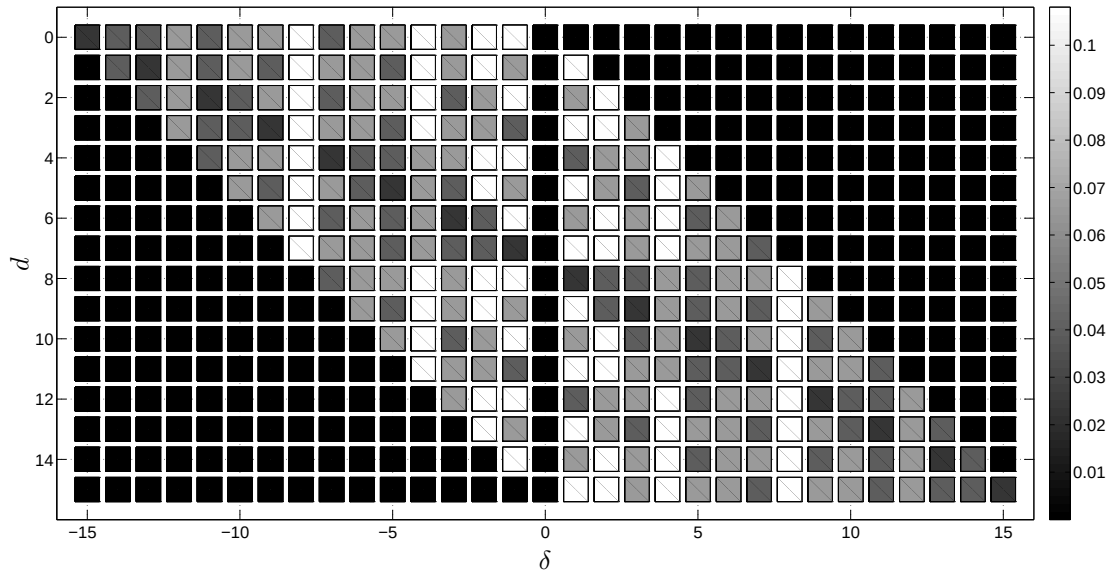
- From $\Pr[f]$'s, we get $\Pr[\delta \mid d]$ for all possible observations and key values
- Observations are collected in $\Delta = \langle \delta[0], \delta[1], \ldots, \delta[N-1] \rangle$
- We can then calculate $\Pr[d \mid \Delta]$ for all key candidates by using Bayesian deduction:

$$\Pr[d \mid \Delta] = \frac{\prod_{i=0}^{N-1} \Pr[\delta[i] \mid d]}{\sum_{j \in \mathcal{K}} \prod_{i=0}^{N-1} \Pr[\delta[i] \mid j]}$$

- Let $\hat{d}$ be the most probable candidate

Example

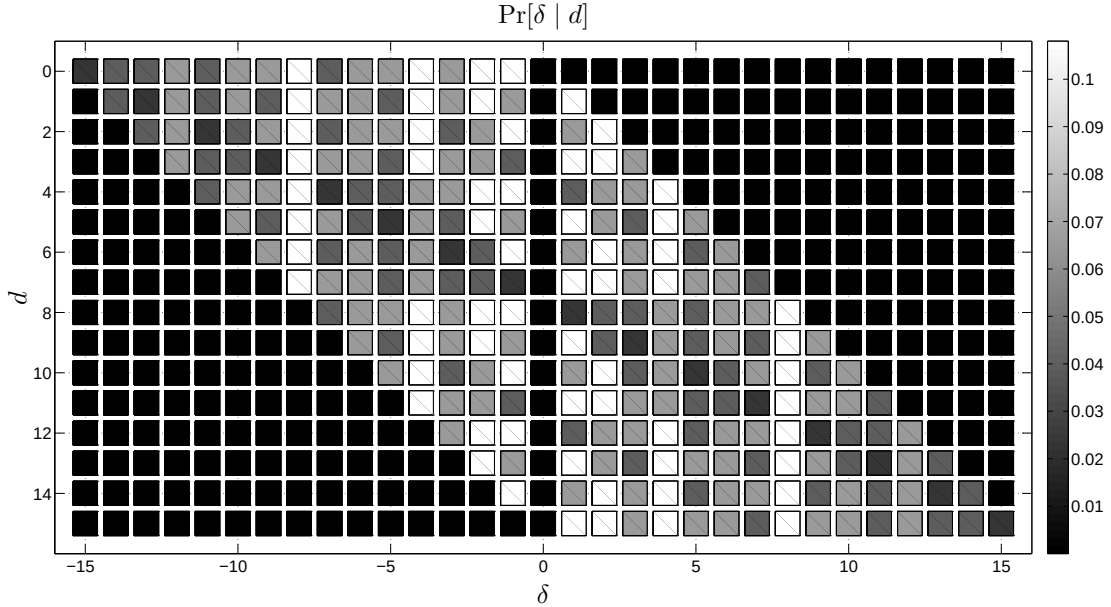
$w = 4$ and $\epsilon = -1/8$



		d												
N	$\delta[i]$	0	...	5	6	7	8	9	10	11	12	13	14	15

Example

$w = 4$ and $\epsilon = -1/8$

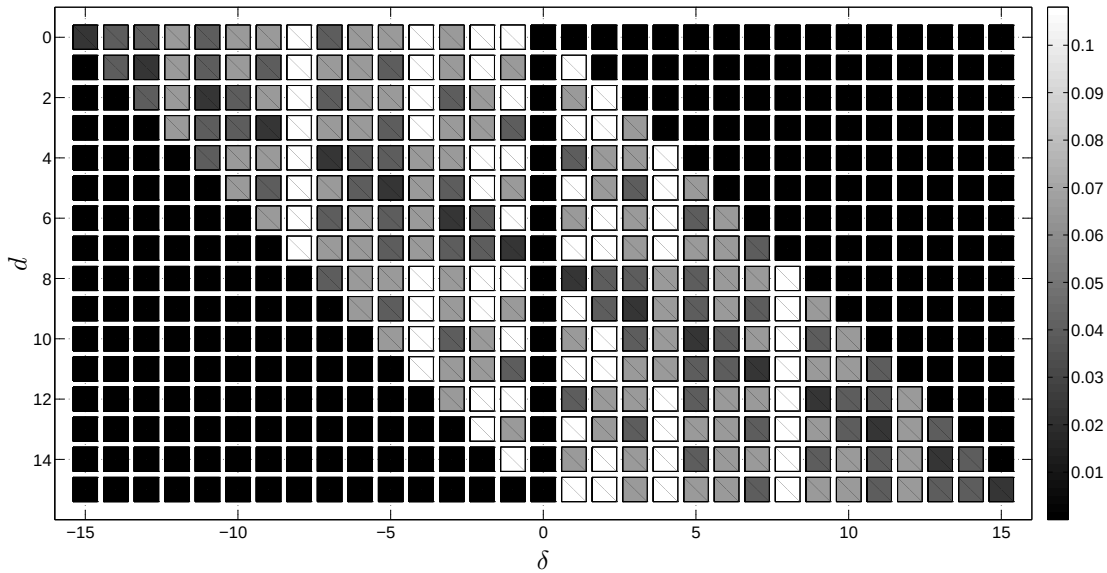


		d												
N	δ[i]	0	...	5	6	7	8	9	10	11	12	13	14	15
1	6	0	...	0	**0.11**	**0.11**	**0.11**	**0.11**	0.07	0.07	**0.11**	**0.11**	**0.11**	**0.11**

Example

$w = 4$ and $\epsilon = -1/8$

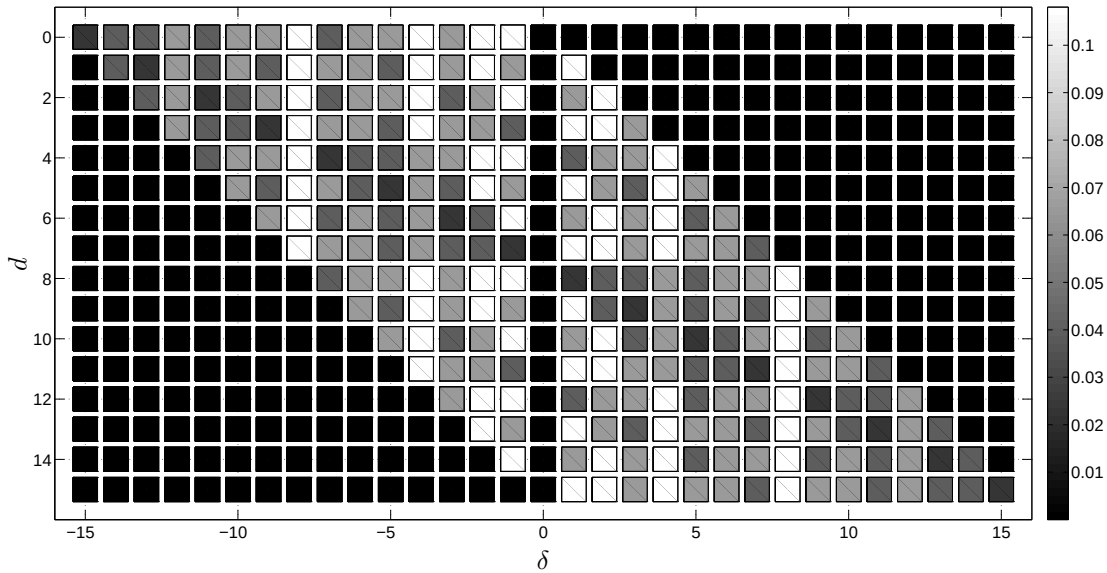
$\Pr[\delta \mid d]$



		d												
N	$\delta[i]$	0	...	5	6	7	8	9	10	11	12	13	14	15
1	6	0	...	0	**0.11**	**0.11**	**0.11**	**0.11**	0.07	0.07	**0.11**	**0.11**	**0.11**	**0.11**
2	−2	0	...	0	0.07	0.07	**0.18**	**0.18**	0.07	0.07	**0.18**	**0.18**	0	0

Example

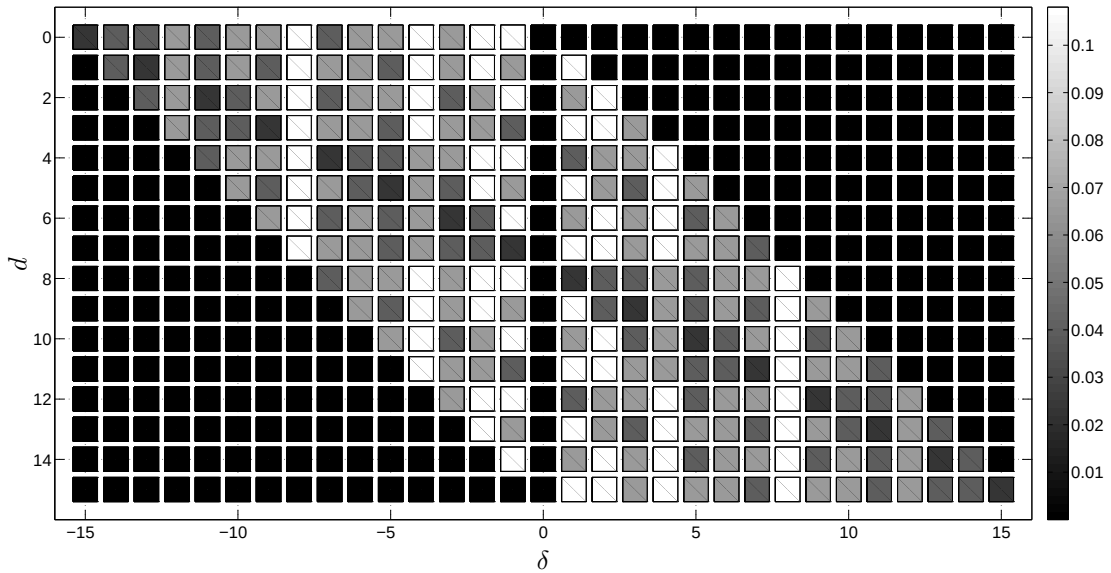
$w = 4$ and $\epsilon = -1/8$



		d												
N	δ[i]	0	...	5	6	7	8	9	10	11	12	13	14	15
1	6	0	...	0	**0.11**	**0.11**	**0.11**	**0.11**	0.07	0.07	**0.11**	**0.11**	**0.11**	**0.11**
2	−2	0	...	0	0.07	0.07	**0.18**	**0.18**	0.07	0.07	**0.18**	**0.18**	0	0
3	8	0	...	0	0	0	**0.21**	**0.21**	0.08	0.08	**0.21**	**0.21**	0	0

Example

$w = 4$ and $\epsilon = -1/8$

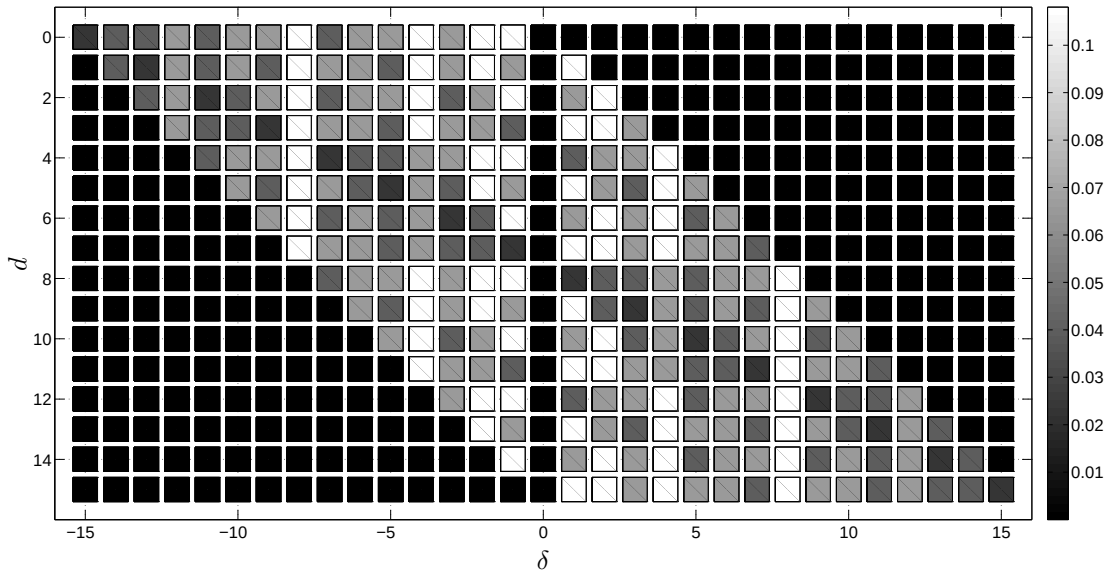


		d												
N	$\delta[i]$	0	...	5	6	7	8	9	10	11	12	13	14	15
1	6	0	...	0	**0.11**	**0.11**	**0.11**	**0.11**	0.07	0.07	**0.11**	**0.11**	**0.11**	**0.11**
2	−2	0	...	0	0.07	0.07	**0.18**	**0.18**	0.07	0.07	**0.18**	**0.18**	0	0
3	8	0	...	0	0	0	**0.21**	**0.21**	0.08	0.08	**0.21**	**0.21**	0	0
4	3	0	...	0	0	0	0.19	0.11	0.07	0.11	**0.32**	0.19	0	0

Example

$w = 4$ and $\epsilon = -1/8$

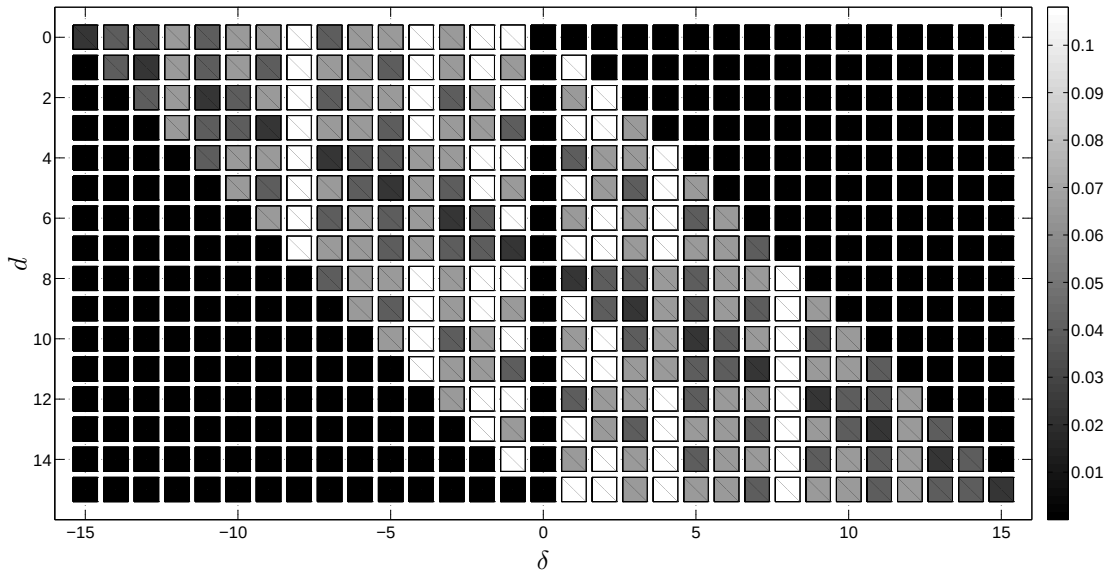
$\Pr[\delta \mid d]$



		d												
N	$\delta[i]$	0	...	5	6	7	8	9	10	11	12	13	14	15
1	6	0	...	0	**0.11**	**0.11**	**0.11**	**0.11**	0.07	0.07	**0.11**	**0.11**	**0.11**	**0.11**
2	−2	0	...	0	0.07	0.07	**0.18**	**0.18**	0.07	0.07	**0.18**	**0.18**	0	0
3	8	0	...	0	0	0	**0.21**	**0.21**	0.08	0.08	**0.21**	**0.21**	0	0
4	3	0	...	0	0	0	0.19	0.11	0.07	0.11	**0.32**	0.19	0	0
5	1	0	...	0	0	0	0.07	0.19	0.07	0.19	0.19	**0.31**	0	0

Example

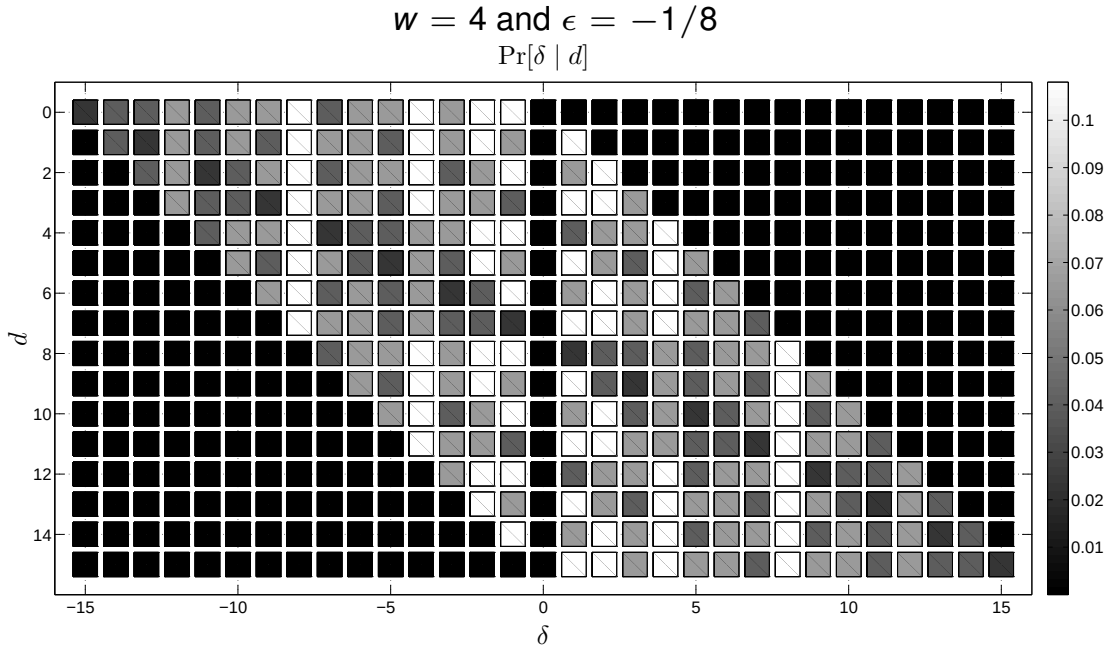
$w = 4$ and $\epsilon = -1/8$



		d												
N	$\delta[i]$	0	...	5	6	7	8	9	10	11	12	13	14	15
1	6	0	...	0	**0.11**	**0.11**	**0.11**	**0.11**	0.07	0.07	**0.11**	**0.11**	**0.11**	**0.11**
2	−2	0	...	0	0.07	0.07	**0.18**	**0.18**	0.07	0.07	**0.18**	**0.18**	0	0
3	8	0	...	0	0	0	**0.21**	**0.21**	0.08	0.08	**0.21**	**0.21**	0	0
4	3	0	...	0	0	0	0.19	0.11	0.07	0.11	**0.32**	0.19	0	0
5	1	0	...	0	0	0	0.07	0.19	0.07	0.19	0.19	**0.31**	0	0
6	4	0	...	0	0	0	0.05	0.14	0.05	0.14	0.23	**0.39**	0	0

Example

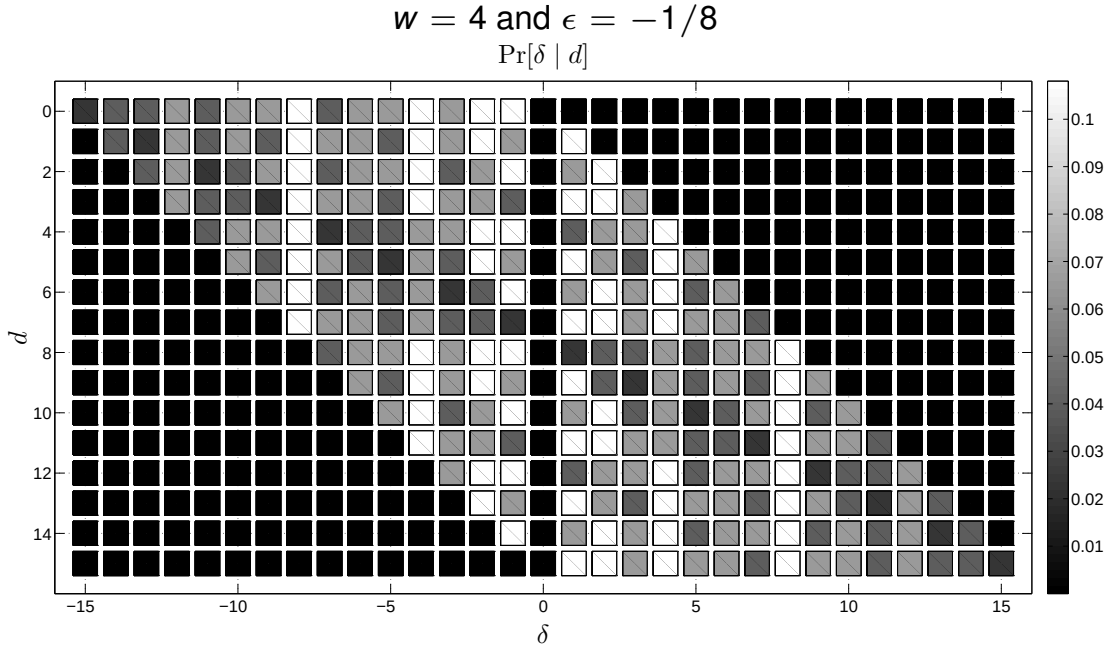
$w = 4$ and $\epsilon = -1/8$



		d												
N	$\delta[i]$	0	...	5	6	7	8	9	10	11	12	13	14	15
1	6	0	...	0	**0.11**	**0.11**	**0.11**	**0.11**	0.07	0.07	**0.11**	**0.11**	**0.11**	**0.11**
2	−2	0	...	0	0.07	0.07	**0.18**	**0.18**	0.07	0.07	**0.18**	**0.18**	0	0
3	8	0	...	0	0	0	**0.21**	**0.21**	0.08	0.08	**0.21**	**0.21**	0	0
4	3	0	...	0	0	0	0.19	0.11	0.07	0.11	**0.32**	0.19	0	0
5	1	0	...	0	0	0	0.07	0.19	0.07	0.19	0.19	**0.31**	0	0
6	4	0	...	0	0	0	0.05	0.14	0.05	0.14	0.23	**0.39**	0	0
7	1	0	...	0	0	0	0.01	0.18	0.04	0.18	0.11	**0.49**	0	0

Example

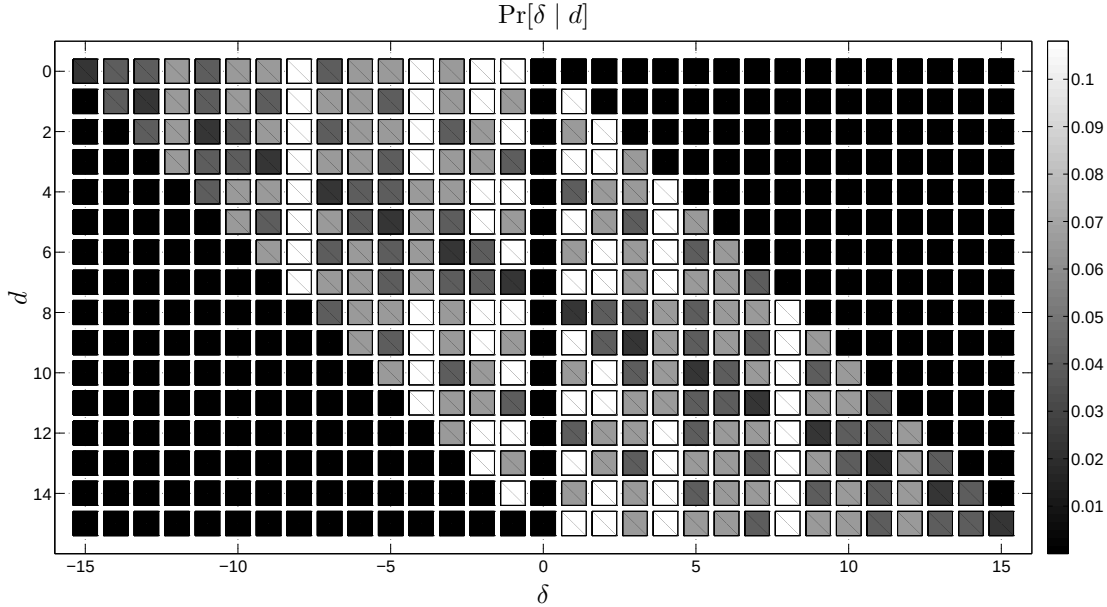
$w = 4$ and $\epsilon = -1/8$



		d												
N	$\delta[i]$	0	...	5	6	7	8	9	10	11	12	13	14	15
1	6	0	...	0	**0.11**	**0.11**	**0.11**	**0.11**	0.07	0.07	**0.11**	**0.11**	**0.11**	**0.11**
2	−2	0	...	0	0.07	0.07	**0.18**	**0.18**	0.07	0.07	**0.18**	**0.18**	0	0
3	8	0	...	0	0	0	**0.21**	**0.21**	0.08	0.08	**0.21**	**0.21**	0	0
4	3	0	...	0	0	0	0.19	0.11	0.07	0.11	**0.32**	0.19	0	0
5	1	0	...	0	0	0	0.07	0.19	0.07	0.19	0.19	**0.31**	0	0
6	4	0	...	0	0	0	0.05	0.14	0.05	0.14	0.23	**0.39**	0	0
7	1	0	...	0	0	0	0.01	0.18	0.04	0.18	0.11	**0.49**	0	0
8	12	0	...	0	0	0	0	0	0	0	0.18	**0.82**	0	0

Example

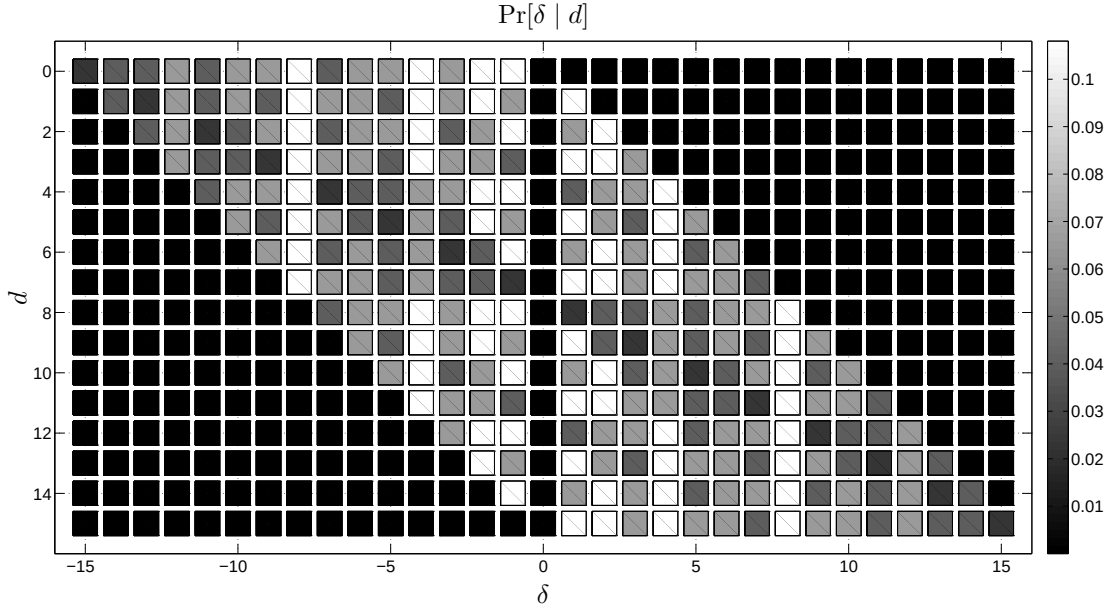
$w = 4$ and $\epsilon = -1/8$



		d												
N	$\delta[i]$	0	...	5	6	7	8	9	10	11	12	13	14	15
1	6	0	...	0	**0.11**	**0.11**	**0.11**	**0.11**	0.07	0.07	**0.11**	**0.11**	**0.11**	**0.11**
2	−2	0	...	0	0.07	0.07	**0.18**	**0.18**	0.07	0.07	**0.18**	**0.18**	0	0
3	8	0	...	0	0	0	**0.21**	**0.21**	0.08	0.08	**0.21**	**0.21**	0	0
4	3	0	...	0	0	0	0.19	0.11	0.07	0.11	**0.32**	0.19	0	0
5	1	0	...	0	0	0	0.07	0.19	0.07	0.19	0.19	**0.31**	0	0
6	4	0	...	0	0	0	0.05	0.14	0.05	0.14	0.23	**0.39**	0	0
7	1	0	...	0	0	0	0.01	0.18	0.04	0.18	0.11	**0.49**	0	0
8	12	0	...	0	0	0	0	0	0	0	0.18	**0.82**	0	0
9	5	0	...	0	0	0	0	0	0	0	0.11	**0.89**	0	0

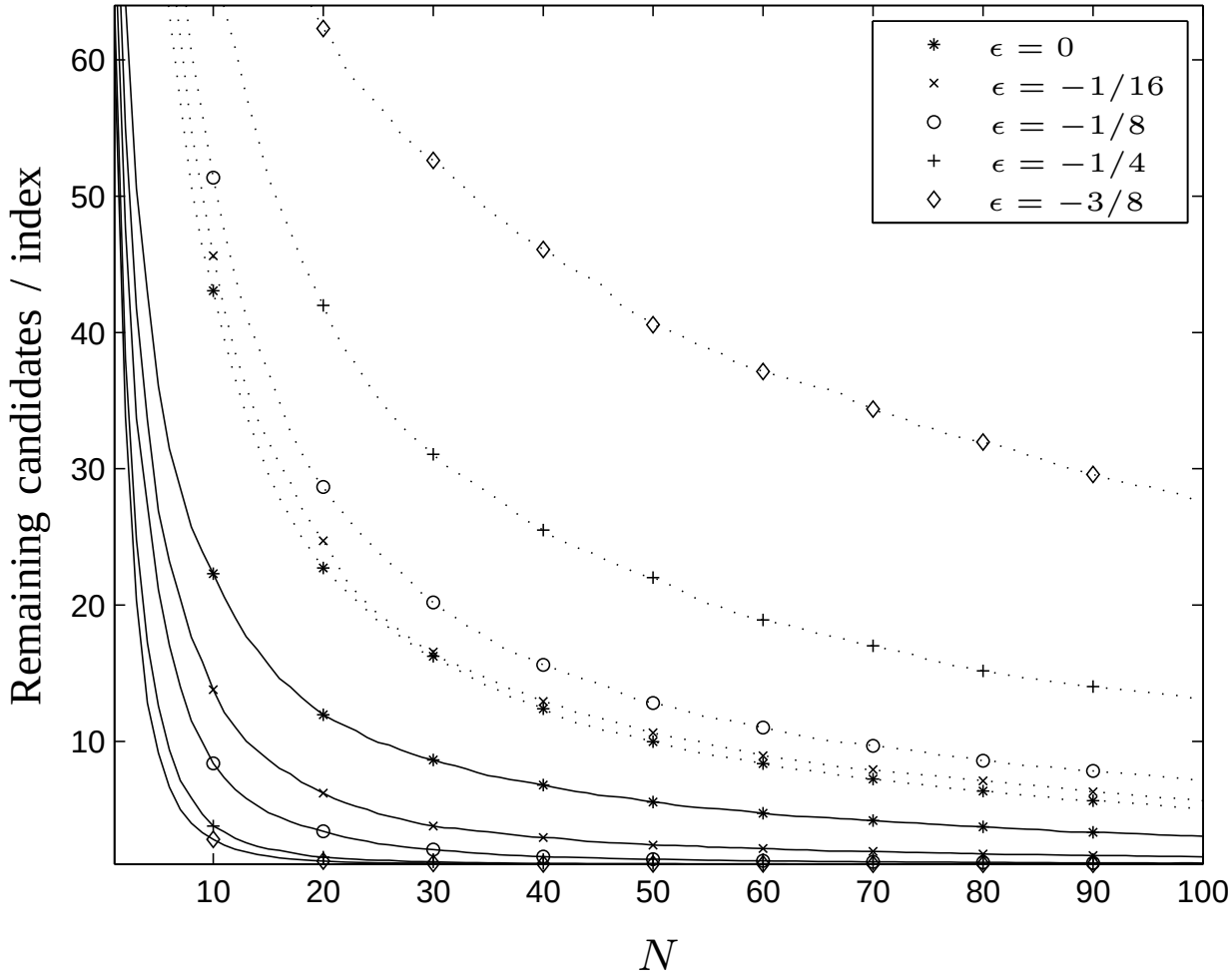
Example

$w = 4$ and $\epsilon = -1/8$



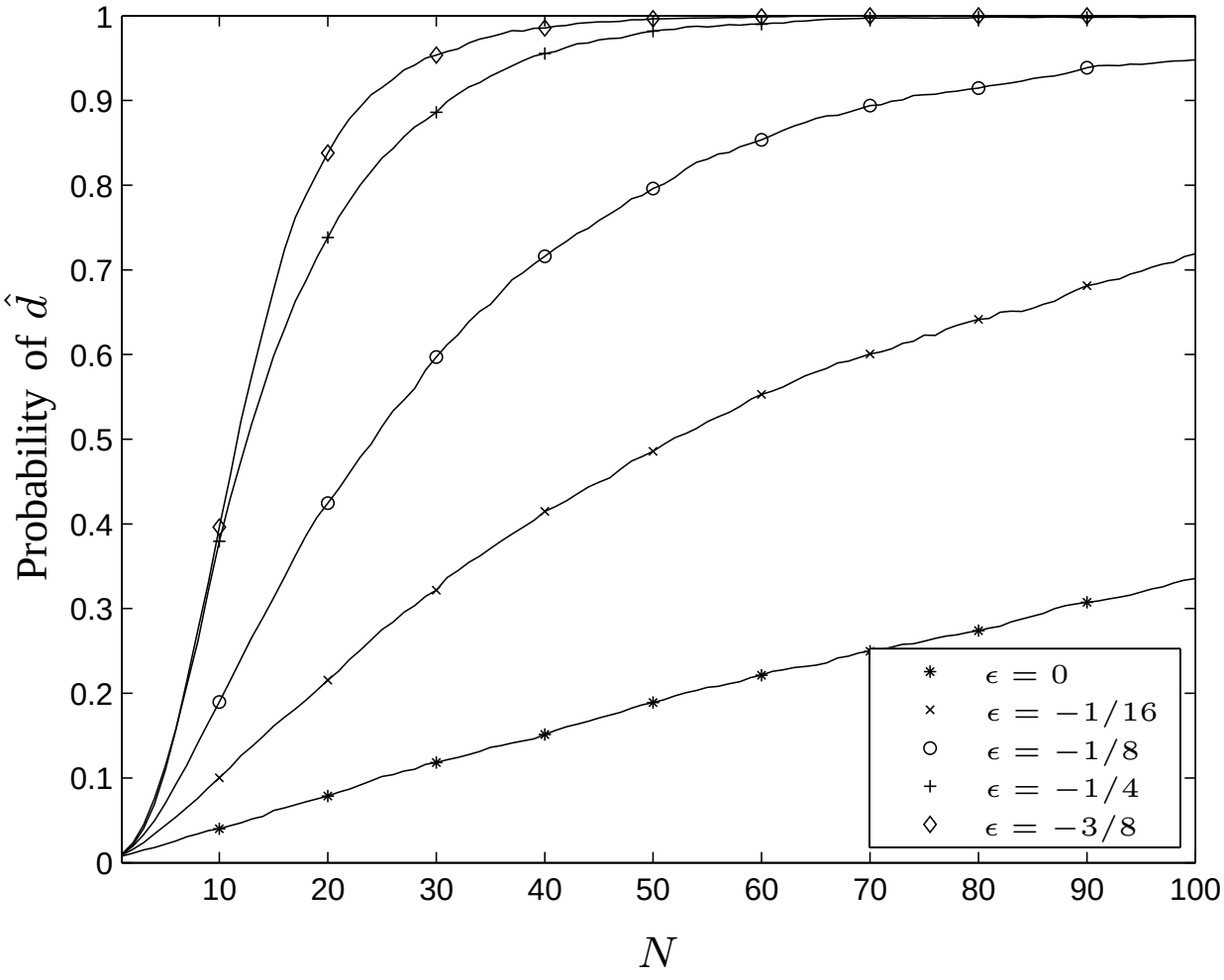
		d												
N	$\delta[i]$	0	...	5	6	7	8	9	10	11	12	13	14	15
1	6	0	...	0	**0.11**	**0.11**	**0.11**	**0.11**	0.07	0.07	**0.11**	**0.11**	**0.11**	**0.11**
2	−2	0	...	0	0.07	0.07	**0.18**	**0.18**	0.07	0.07	**0.18**	**0.18**	0	0
3	8	0	...	0	0	0	**0.21**	**0.21**	0.08	0.08	**0.21**	**0.21**	0	0
4	3	0	...	0	0	0	0.19	0.11	0.07	0.11	**0.32**	0.19	0	0
5	1	0	...	0	0	0	0.07	0.19	0.07	0.19	0.19	**0.31**	0	0
6	4	0	...	0	0	0	0.05	0.14	0.05	0.14	0.23	**0.39**	0	0
7	1	0	...	0	0	0	0.01	0.18	0.04	0.18	0.11	**0.49**	0	0
8	12	0	...	0	0	0	0	0	0	0	0.18	**0.82**	0	0
9	5	0	...	0	0	0	0	0	0	0	0.11	**0.89**	0	0
10	13	0	...	0	0	0	0	0	0	0	0	**1.00**	0	0

Results: Index of the correct key



$w = 8$, averages from 1000 experiments for each bias

Results: Probabability of the best candidate



$w = 8$, averages from 1000 experiments for each bias

How realistic it is to assume biased faults?

Clock glitches (Balasch et al., FDTC 2011)

- Faults on data loaded from memory are biased by the position and value of the data

Voltage depletion (Barenghi et al., IACR ePrint 2010/130)

- Faults on data loaded from memory are biased by the position and value ($1 \rightarrow 0$ flips) of the data

Laser shots (Canivert et al., e.g. IEEE VLSI Test Symp. 2009)

- Faults are biased by the value of data

How to obtain the biases?

Estimates of the biases can be obtained in two ways:

1. Fault a public value on the targeted device (or a similar device) and calculate the biases
   - Critical that faults are similar to those targeted to d
2. Fault d and deduct the biases from the distribution of δ's
   - These faults can be reused in the actual attack

Conclusions & future work

Summary

- We presented a very powerful fault attack on public-key cryptosystems that uses biased faults
- More theoretical analysis, countermeasures, etc. are available in the paper

Suggestions for future work

- Fault injection experiments in order to verify the fault model and to receive information on what kind of biases are obtainable in practice
- Further analysis: at least varying biases for different bits ($\epsilon_i \neq \epsilon_j$) and faults biased by the value

Demo

- Programmed by our summer student Juan Francisco Muñoz Castro