

1001 ways to fail record computations

F. Boudot, P. Gaudry, A. Guillevic,
N. Heninger, E. Thomé, P. Zimmermann



UNIVERSITÉ
DE LORRAINE



Inria

UC San Diego

JACOBS SCHOOL OF ENGINEERING



xlim



Dec. 2nd, 2019

Polynomial Selection

$$p = \text{RSA-240} + 49204$$

$$f = 39x^4 + 126x^3 + x^2 + 62x + 120$$

$$g = 286512172700675411986966846394359924874576536408786368056x^3 \\ + 24908820300715766136475115982439735516581888603817255539890x^2 \\ - 18763697560013016564403953928327121035580409459944854652737x \\ - 236610408827000256250190838220824122997878994595785432202599$$

$$\text{Res}(f, g) = -540p$$

Polynomial Selection

$$p = \text{RSA-240} + 49204$$

$$f = 39x^4 + 126x^3 + x^2 + 62x + 120$$

$$g = 286512172700675411986966846394359924874576536408786368056x^3 \\ + 24908820300715766136475115982439735516581888603817255539890x^2 \\ - 18763697560013016564403953928327121035580409459944854652737x \\ - 236610408827000256250190838220824122997878994595785432202599$$

$$\text{Res}(f, g) = -540p$$

Use the wrong polynomials

Polynomial Selection

$$p = \text{RSA-240} + 49204$$

$$f = 39x^4 + 126x^3 + x^2 + 62x + 120$$

$$g = 286512172700675411986966846394359924874576536408786368056x^3 \\ + 24908820300715766136475115982439735516581888603817255539890x^2 \\ - 18763697560013016564403953928327121035580409459944854652737x \\ - 236610408827000256250190838220824122997878994595785432202599$$

$$\text{Res}(f, g) = -540p$$

Use the wrong polynomials

```
dlp240test.sh:
```

```
if [ "`sha1sum $poly`" != 0423a08c3b518fb5788300caf2cd01c3c4fbda03 ]
then
    echo "You are using the wrong polynomial !!!" >&2
    exit 1
fi
```

Sieving

Collect millions of relations between f -side and g -side with cado-nfs' binary las.

Sieving

Collect millions of relations between f -side and g -side with cado-nfs' binary las.

Use wrong las options and parameters

- about 70 parameters or options
- `-lambda0`, `-lambda1`, `-bkthresh`, `-bkthresh1`, `-bkmult`

Sieving

Collect millions of relations between f -side and g -side with cado-nfs' binary las.

Use wrong las options and parameters

- about 70 parameters or options
- `-lambda0`, `-lambda1`, `-bkthresh`, `-bkthresh1`, `-bkmult`

Too many files

- `ls` impossible
- server crashed
- re-organize files, move `.tgz` relation files, etc...

Core binding

Bind each mono-thread job to a virtual core

Bind each double-thread job `-t 2` to a physical core

```
p=`grep "processor" /proc/cpuinfo | tail -1 | cut -d " " -f 2`  
let N=$((p-1)/2)  
for i in `seq 0 $N`; do  
    b="core:${i}"  
    hwloc-bind --membind $b --cpubind $b $cmd &  
    sleep 0.2  
done
```


Core binding

Bind each mono-thread job to a virtual core

Bind each double-thread job `-t 2` to a physical core

```
p=`grep "processor" /proc/cpuinfo | tail -1 | cut -d " " -f 2`  
let N=$((p-1)/2)  
for i in `seq 0 $N`; do  
    b="core:${i}"  
    hwloc-bind --membind $b --cpubind $b $cmd &  
    sleep 0.2  
done
```

Use a physicists' cluster with heterogeneous CPU

binding error, server crashed

Blindly trust auto CPU binding

can be 30% slower in some cases

Linear algebra

Block Wiedemann stores intermediate computation as big **vectors**.
Gigabytes each.

Block Wiedemann stores intermediate computation as big **vectors**.
Gigabytes each.

Transient storage fault for checkpoints

- Counter-measure: verify the checkpoints.
- Counter-failure: verifier was incomplete!
- Iterate.

Block Wiedemann stores intermediate computation as big **vectors**.
Gigabytes each.

Transient storage fault for checkpoints

- Counter-measure: verify the checkpoints.
- Counter-failure: verifier was incomplete!
- Iterate.

Battle silent failures

MPI-send data.

Hit a silent 4G limitation in message size in some cases.

Finally, everything went fine!

```
 1 [|||||] 100.0%  17 [|||||] 100.0%  33 [|||||] 100.0%  49 [|||||] 100.0%
 2 [|||||] 100.0%  18 [|||||] 100.0%  34 [|||||] 100.0%  50 [|||||] 100.0%
 3 [|||||] 100.0%  19 [|||||] 100.0%  35 [|||||] 100.0%  51 [|||||] 100.0%
 4 [|||||] 100.0%  20 [|||||] 100.0%  36 [|||||] 100.0%  52 [|||||] 100.0%
 5 [|||||] 100.0%  21 [|||||] 100.0%  37 [|||||] 100.0%  53 [|||||] 100.0%
 6 [|||||] 100.0%  22 [|||||] 100.0%  38 [|||||] 100.0%  54 [|||||] 100.0%
 7 [|||||] 100.0%  23 [|||||] 100.0%  39 [|||||] 100.0%  55 [|||||] 100.0%
 8 [|||||] 100.0%  24 [|||||] 100.0%  40 [|||||] 100.0%  56 [|||||] 100.0%
 9 [|||||] 100.0%  25 [|||||] 100.0%  41 [|||||] 100.0%  57 [|||||] 100.0%
10 [|||||] 100.0%  26 [|||||] 100.0%  42 [|||||] 100.0%  58 [|||||] 100.0%
11 [|||||] 100.0%  27 [|||||] 100.0%  43 [|||||] 100.0%  59 [|||||] 100.0%
12 [|||||] 100.0%  28 [|||||] 100.0%  44 [|||||] 100.0%  60 [|||||] 100.0%
13 [|||||] 100.0%  29 [|||||] 100.0%  45 [|||||] 100.0%  61 [|||||] 100.0%
14 [|||||] 100.0%  30 [|||||] 100.0%  46 [|||||] 100.0%  62 [|||||] 100.0%
15 [|||||] 100.0%  31 [|||||] 100.0%  47 [|||||] 100.0%  63 [|||||] 100.0%
16 [|||||] 100.0%  32 [|||||] 100.0%  48 [|||||] 100.0%  64 [|||||] 100.0%
Mem[|||||] 170G/188G  Tasks: 365, 119 thr; 65 running
Swp[|||||] 0K/3.72G    Load average: 65.01 64.26 52.02
Uptime: 00:42:24
```

Finally, everything went fine!

RSA-240 =

509435952285839914555051023580843714132648382024111473186660\
296521821206469746700620316443478873837606252372049619334517
*

244624208838318150567813139024002896653802092578931401452041\
221336558477095178155258218897735030590669041302045908071447

$p = \text{RSA-240} + 49204$

target = hex("The magic words are still Squeamish Ossifrage")

$\log_5(\text{target}) =$

926031359281441953630949553317328555029610991914376116167294\
204758987445623653667881005480990720934875482587528029233264\
473672441500961216292648092075981950622133668898591866811269\
28982506005127728321426751244111412371767375547225045851716