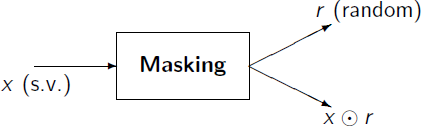


Second-order attack on masked AES implementation

Advanced SCA Training

Goal: How to attack a masked implementation of AES

* Applying masking countermeasure to AES
* Measurement setup
* Using second-order attack to defeat masking
* AES is a NIST standardized symmetric key cryptographic algorithm
* Straightforward implementations of AES can be attacked easily using Side-channel attacks (Ex: power, EM)
* Several countermeasures exist to thwart SCA
  + Random delays
  + Shuffling
  + Masking
* Masking is one of the most widely used countermeasure
* Choose an intermediate variable that is function of the secret key (Ex: S-box output)
* Collect the power/EM traces
* Compute the hypothetical intermediate values for each key candidate
* Estimate the hypothetical power consumption (or EM) values using suitable power model
* Compare the hypothetical values with the measured traces using a statistical distinguisher (Ex: DOM. CPA)
* In Masking, we divide each sensitive variable into two shares: mask (generated randomly) and the masked value



* All the operations are performed independently on the two shares
* The first-order DPA/CPA attack involving single point on
* The state variable is divided into two shares: mask and the masked value
* We can perform all the linear operations (KeyAddition, ShiftRows, MixColumns) independently on the shares
* However, it’s not straightforward in case of SubBytes (S- box) due to it’s non-linearity
* In software SubBytes operation is implemented using lookup table for performance reasons
* **Solution**: Create a masked LUT
* Let the original S-box LUT be *S* which is of size 256 bytes
* We need two masks: input mask to mask the index, and output mask to mask the S-box output.
* Let *m1* be the input mask and *m2* be the output mask
* We create a new look-up table S1 such that *S1(x* ***⊕*** *m1 ) = S(x)* ***⊕*** *m2*
* In some cases *m1* and *m2* can be same, but we use different masks in our implementation
* Masked implementations can still be attacked by combining two points of the trace by correlating the joint distribution with the hypothetical values
* The two points corresponds to the two shares of the sensitive variable (Ex: S-box output)
* There exists several combination functions to combine the points:
  + Absolute difference
  + Product
  + Square of sum
* In a second-order attack the first step is to identify the points on the trace where the two shares are being manipulated
* We do not know exactly where the shares are being manipulated. So we consider a range
* Assume (ti , tj) is the interval corresponding to one share and (tk , tl) corresponds to the other
* We combine every point in (ti , tj) with every point in (tk , tl) using one of the combination function to create a new set of traces
* Choose two shares of the intermediate variable to be attacked (Ex: S-box output; *S1(x* ***⊕*** *m1 ) = S(x)* ***⊕*** *m2 , m2* where *x=p* ***⊕*** *k*).
* Find the interval where *m2* is generated and *S1(x* ***⊕*** *m1 )* is computed and combine using preprocessing step
* For all possible values of x estimate the hypothetical power values by computing the Hamming weight (or Hamming distance, …) of *S(x)*
* Compute the correlation coefficient between the preprocessed traces and the hypothetical power values
* We can also perform CPA attack directly on 2 bytes thus guessing 216 key candidates at a time
* Consider output of two S-boxes i.e. *S1(x1* ***⊕*** *m1 ) = S(x1)* ***⊕***

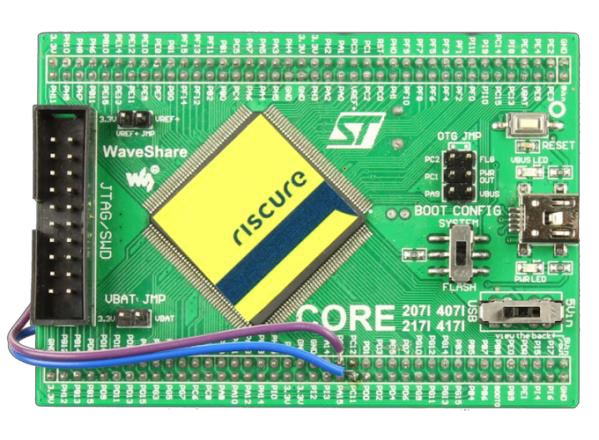
*m2* and *S1(x2* ***⊕*** *m1 ) = S(x2)* ***⊕*** *m2*

* Combine the points corresponding to the leakages from two S-boxes
* The hypothetical values are computed by *S1(x1* ***⊕*** *m1 )* ***⊕***

*S1(x2* ***⊕*** *m1 )* for all the possible values of *x1* and *x2*

* The correct key candidate gives the maximum correlation with preprocessed traces

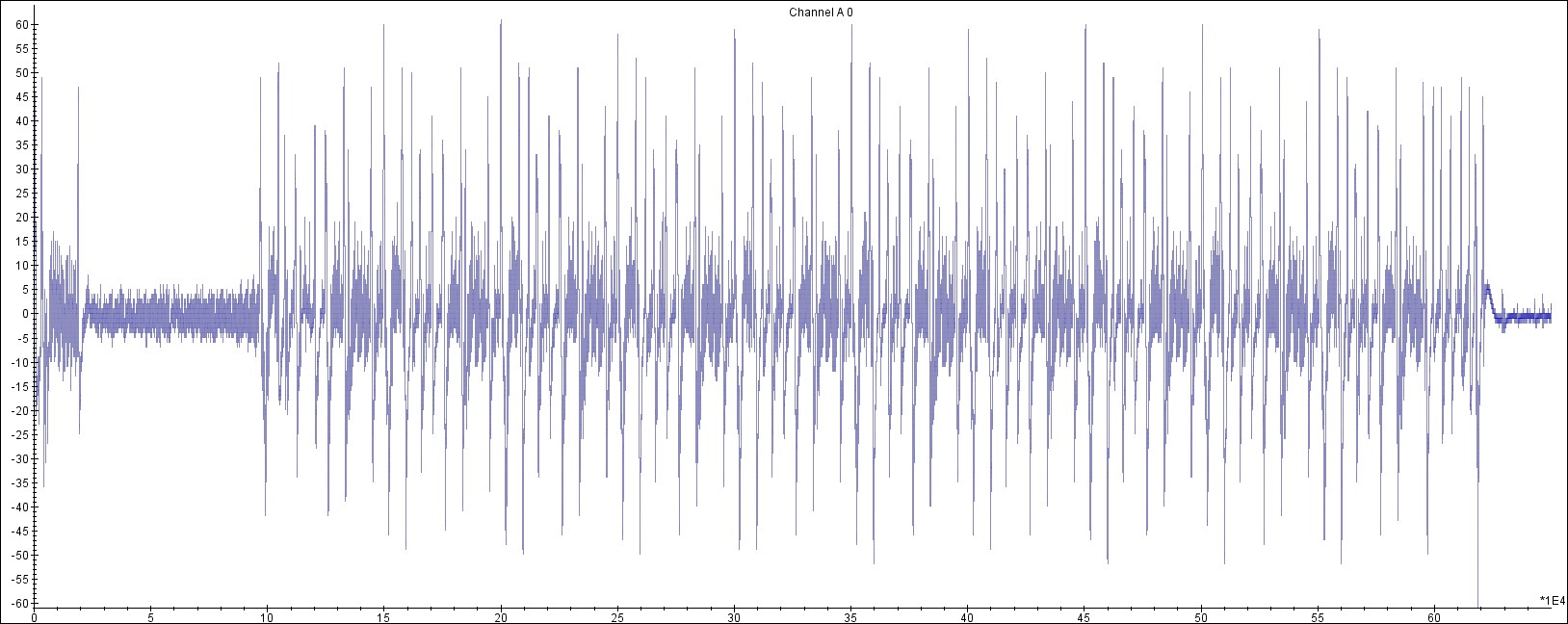
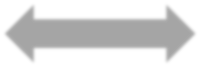
High-end Cortex-m4 CPU which can run at 168MHz Designed as a realistic training target (source code available) We will measure the power consumption at the 3.3V pin

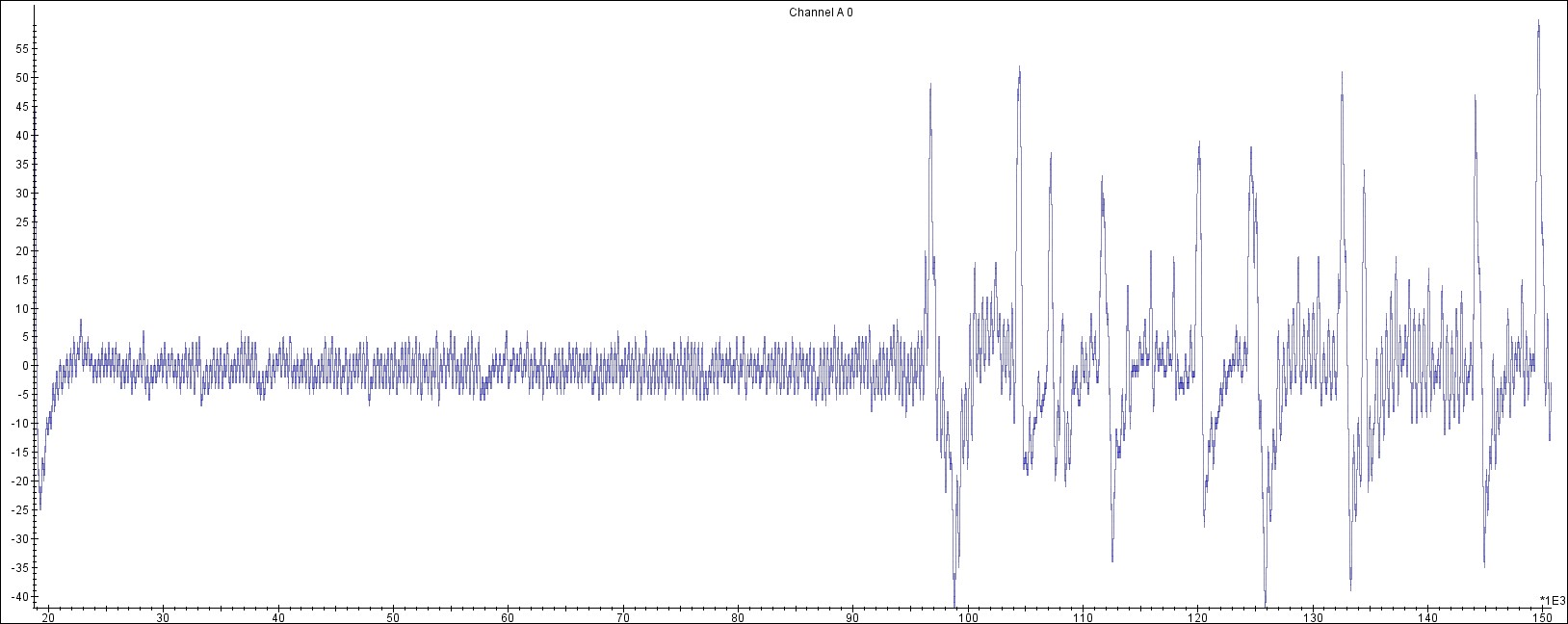


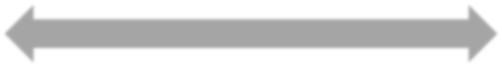
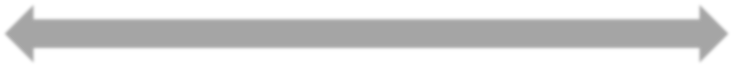
No need for amplifier due to board preparation

* PicoScope 3207B USB Oscilloscope
* Used side-channel: power consumption
* External power supply
* PC2 pin for trigger (before and after decryption)
* FTDI cable for communication
* Riscure Current Probe
* Riscure Inspector Software for acquisition and analysis
* Sampling Frequency: 1 GSa/s
* Number of samples: ~650k for the full operation
* Range: 500mV
* Number of traces acquired: 5000

Masked S-box 10 rounds

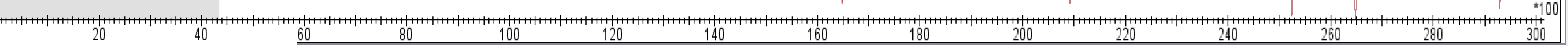
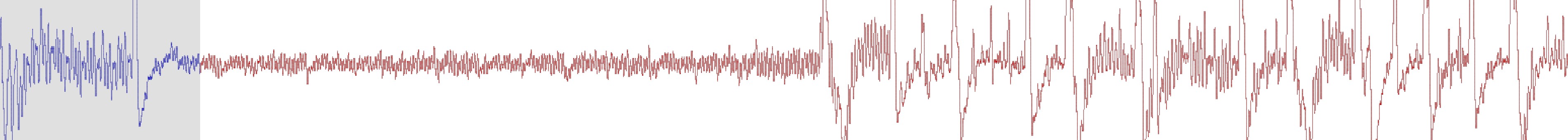
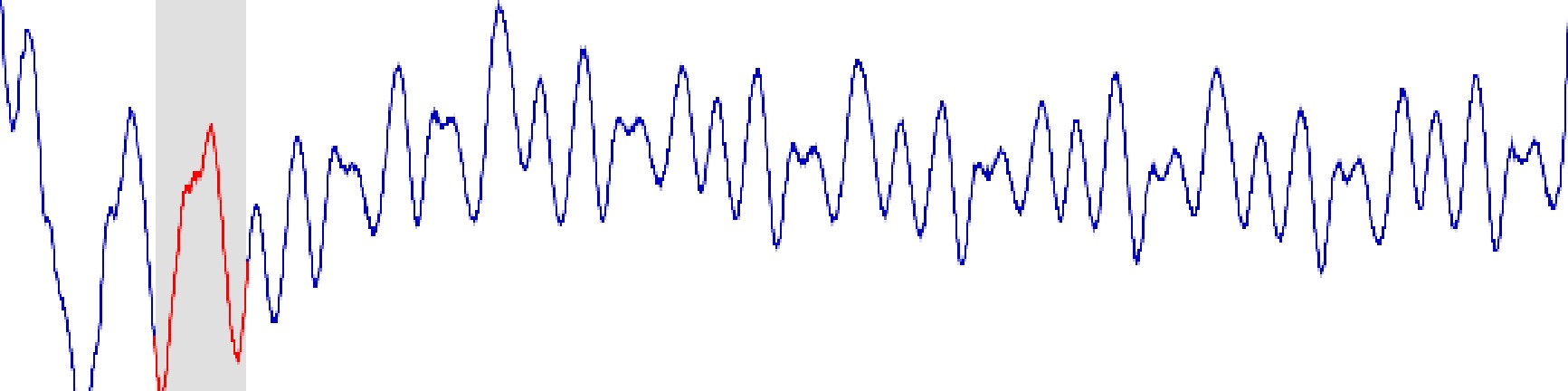


Masked S-box 1 round



* Extract the point where output mask *m2* is generated
* Extract the points where the Subbytes operation for round 1 is happening *(S(xi* ***⊕*** *ki )* ***⊕*** *m2 )*
* Once we have extracted these points we perform second- order attack
* Combine the points where the mask is generated with 16 points corresponding to 16 S-box outputs

~ ES\_Mas eaJ UUK\_IGs + Resamp rea:=rs- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 7 ° l.§\_



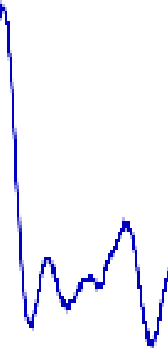
5 \*10

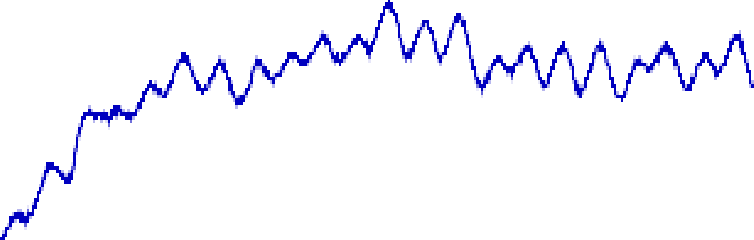
A

4

trace 4999 I A

□ 3

2

0

-1

-2

-3 \*1 00 D

,. 0 5 10 1 5 20

◄ I *m* I

25 30 35 40 ,.

6 \*1 0 trace 4999

5

4

3

2

0

-1

-2

-3

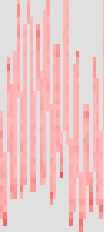
-4

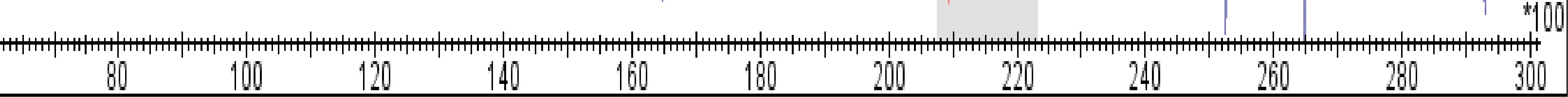
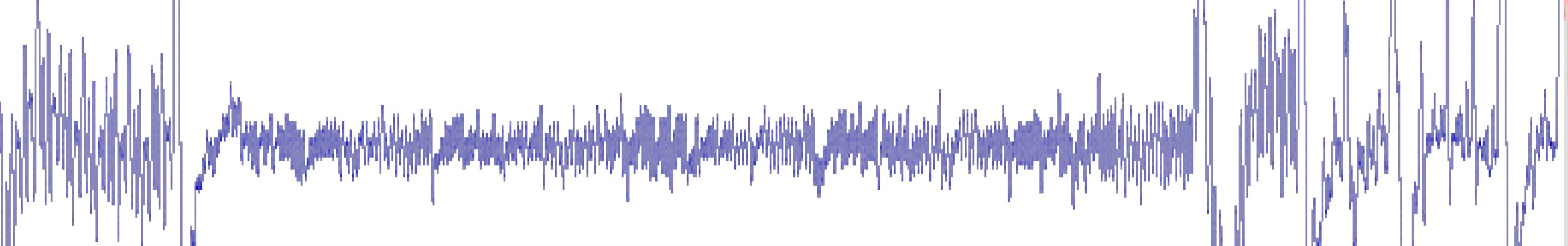
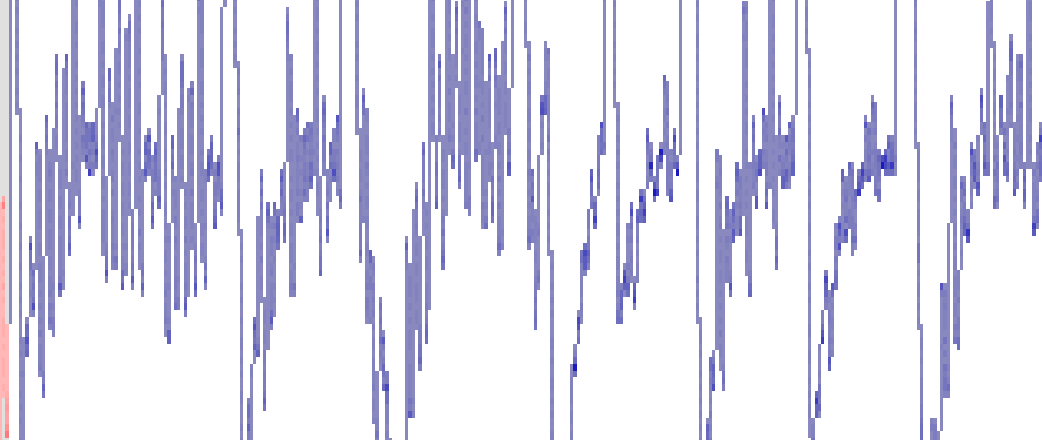
0

Out

Trace

Status Traces Sma ples Data





o \*rn

trace 4999

4

j

*L*

1

0

-1

* *L*
* j

-4

0

*LO*

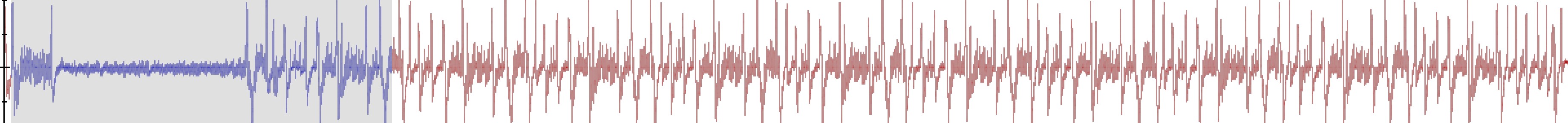
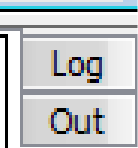
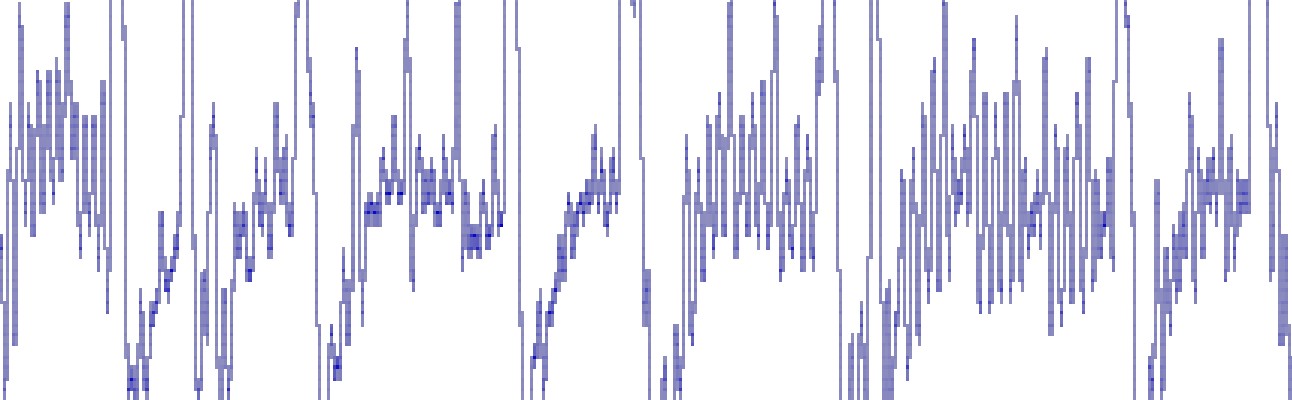
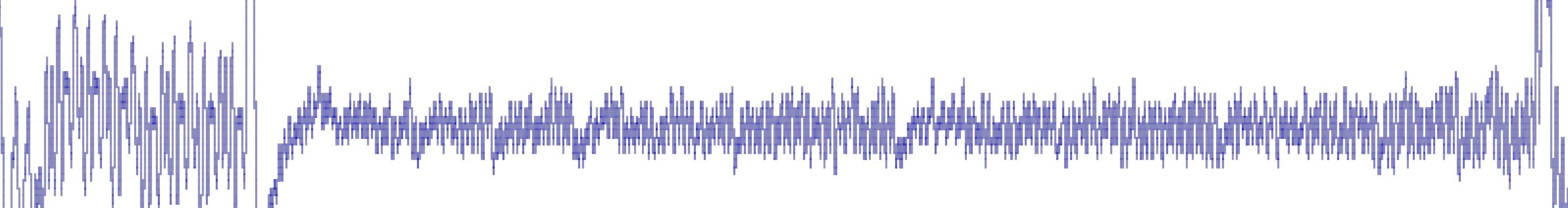
40

oO

Insped or 4.8.3-SABuild vl. O SCA+FI CJ@

Fil e Ed it Pertmhati on Acq1Uisi tion Comp ress Ali gn Fil t er Ana l'ysis Cry pto Samp l es Trac es User Tools Wi n d ow H el p P rof il e s

'( **J T** ... \_1j Line s: I 1 1 Traces: I 1 1



**AES\_Mask ed\_lOOK\_l Gs (Sk traces only).trs**

\*1 0 mV Channel A 0

20

1 5

10

5

0

-5

-10

-15

..,.

.......--+-t-.......-.--+-t----+-t---+-t------t-t---t-1---t-t---t-t---t-<---t-t---t-<---t-t---t-<t-+----t-t---t-<t-+-.........,, t-+-t-t-t-+-t-t-t-t-t-t-......-t-+-......-----t-+---t-+-+-t--t-+--t-+--t-t--t-+--+-t--t-t--+-t--t-t--+-t--+-t--t-1--+-t---t-t--t-1---t-t---t-t---t-+--t-t---t-+--t-t---t-t---t-+--t-t---t-t-................... , ......... ,

---t-1\_ ,..........,....,.\_.---,µ s

..,.

1 0 20 30 40 50 60 70 80 90 1 00 110 1 20 1 30 14 0 1 50

◄ m ►

\*1 00 mV Channel A 0

2

a

-1

-2 \*1 0 µs

0 5 10 1 5 20 25 30 35 40 45 50 55 60

Trace

Status

Reaclly

Traces

available: 50 00 displayed: 0 selected: 0

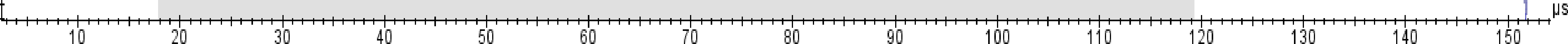
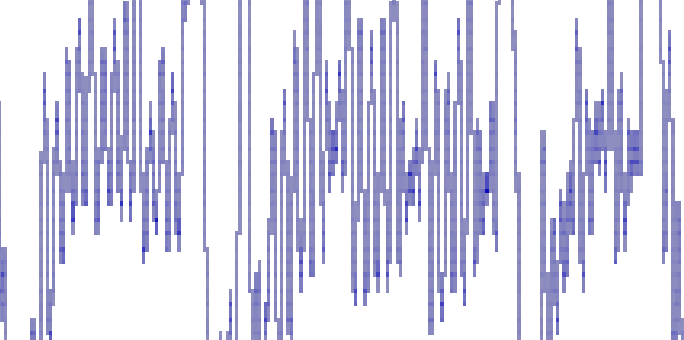
Samp le s Value Data

available:65000 0 (650 µs ) d isplayed:151617 (151.62 µs ) at 2573 (2.573 µs ) se lected: 0 - 0.0,511311 at 112327 (112.83 µs ) 11 15 6B 9B DC 1A 68 5C 85 3C OD FID 32 9D BS 48 OB 52 -41E 26 OF 24 02 EA B5 68 03 05 85 F...



20

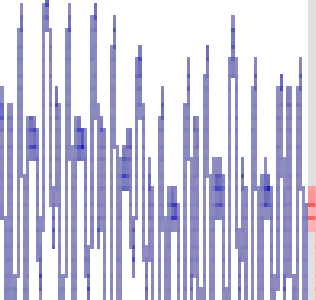
\*1a mv Channel AO

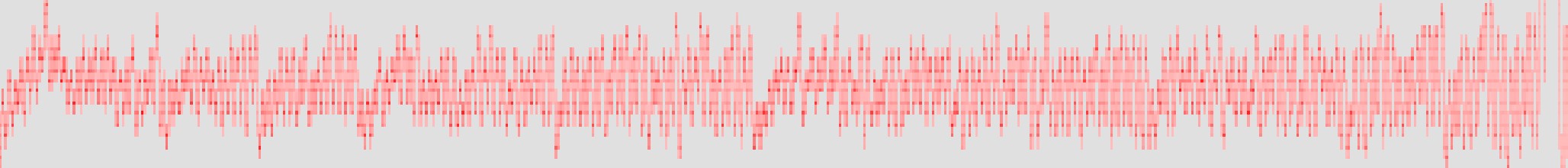


20

1 5

10

5

. !

a

J

I

.\

-5

-1 a

-15

1. · 1 . I

I

50

45

40

35

30

25

20

1 5

1 0

5

0

-5

-10

-15

-20

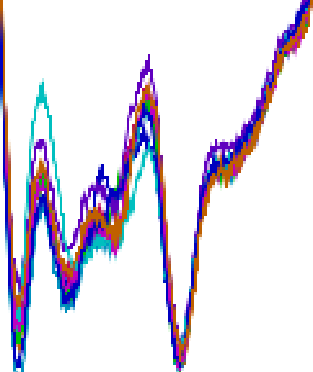
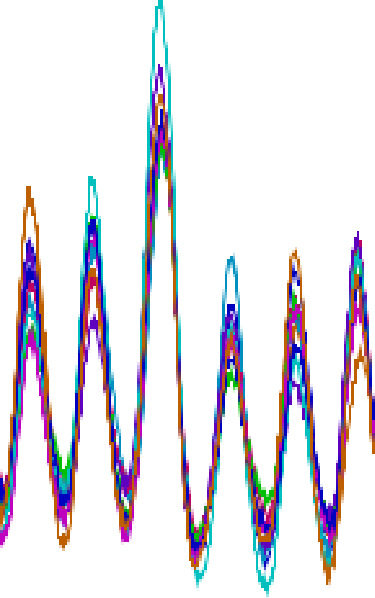
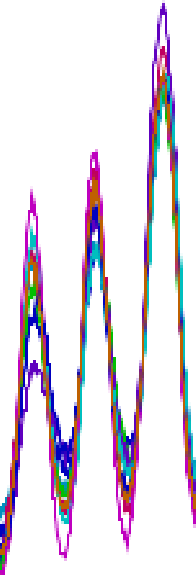
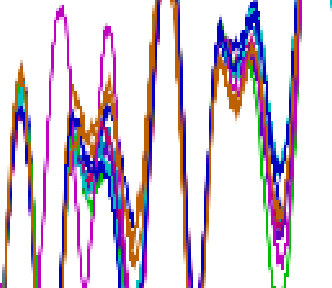
-25

-30

-35

trace 1 0-19

-40



0 30 60 90 1 20 1 50

18 0 21 0 240 270 300 330 360 390 420

\*1 0

8

7

6

5

4

3

2

1

0

-1

-2

-3

-4

-5

-6

-7

-8

-9

-10

-11

-1 2

-13

-14

-15

-16

-17

-18

-19

-20

-21

-22

-23

-24

-25

-26

-27

-28

-29

-30

-31

-32

-33

-34

-35

-36

-37

-38

-39

-40

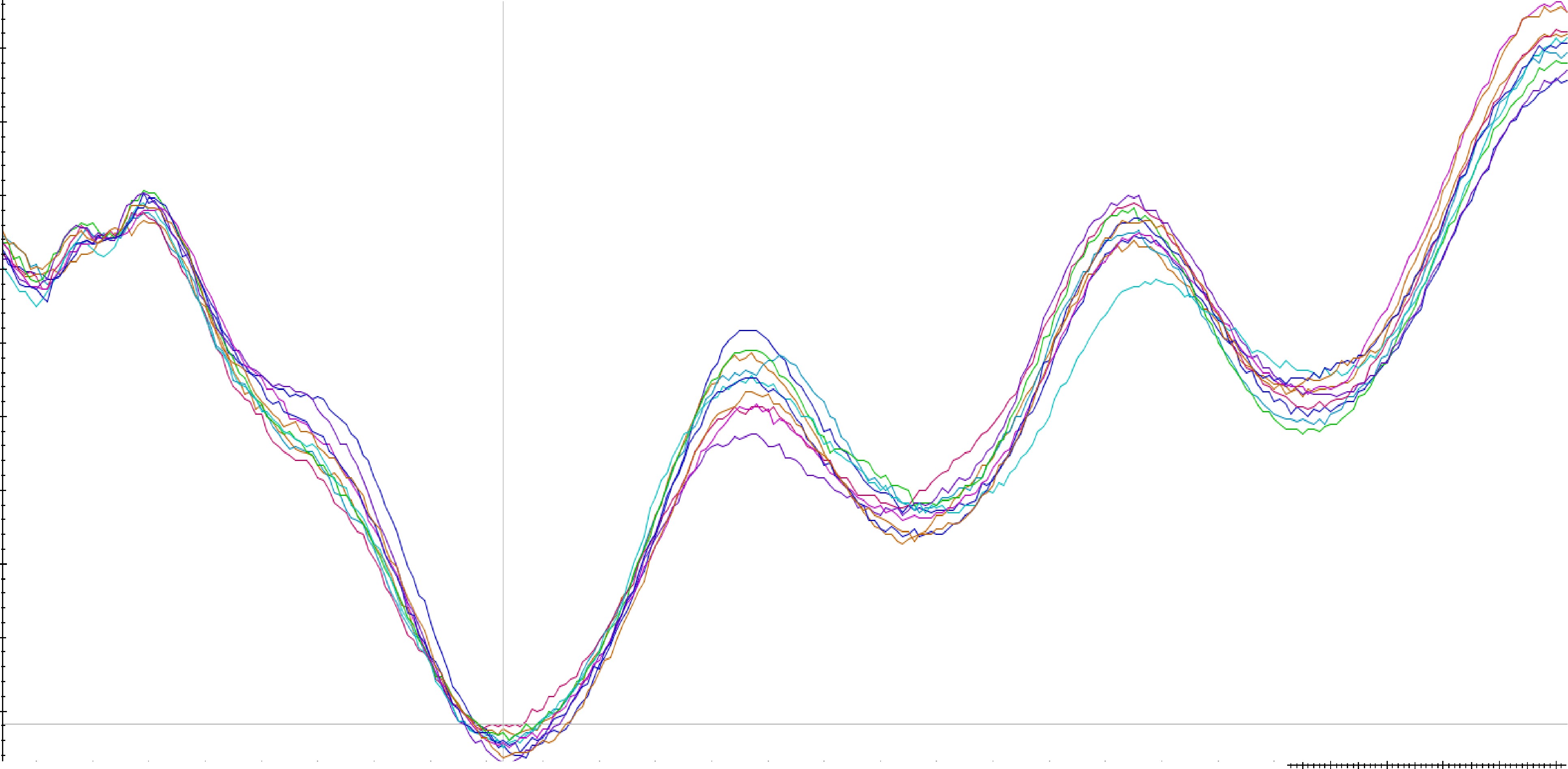
-41

-4 2

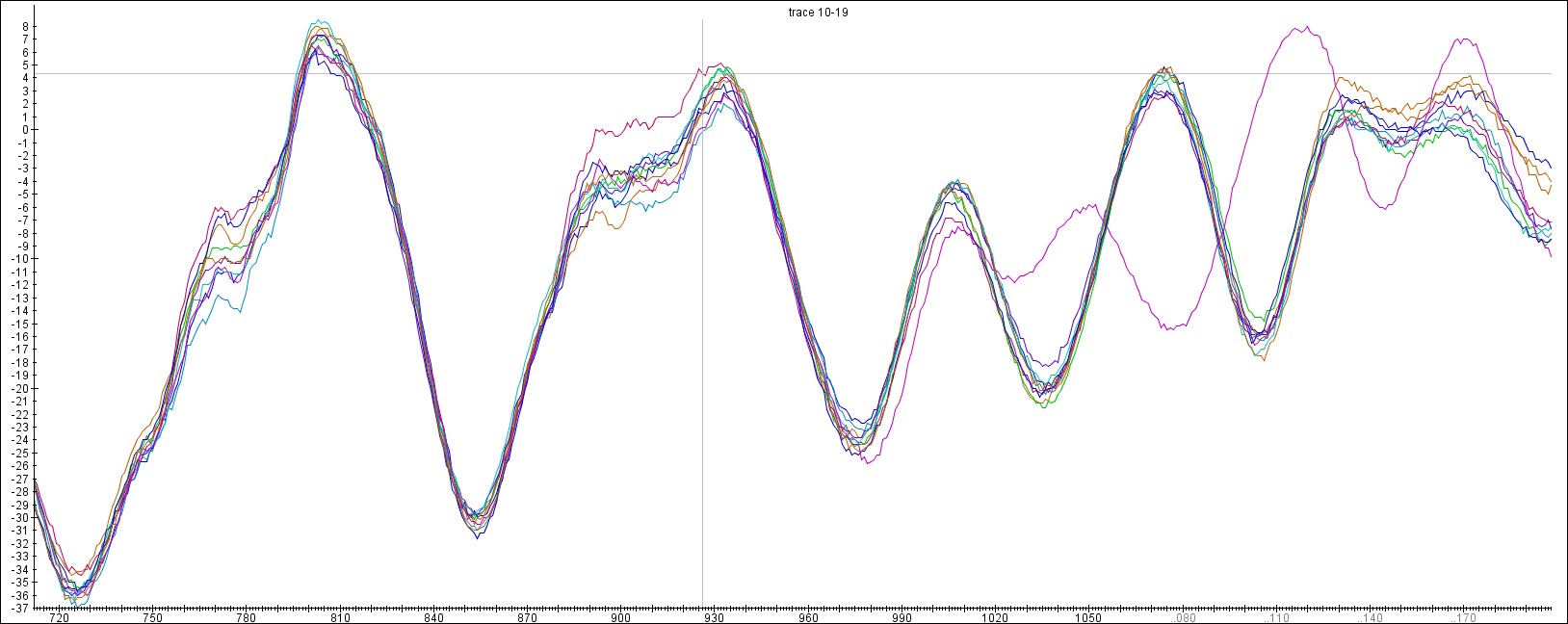
-43

trace 0-9

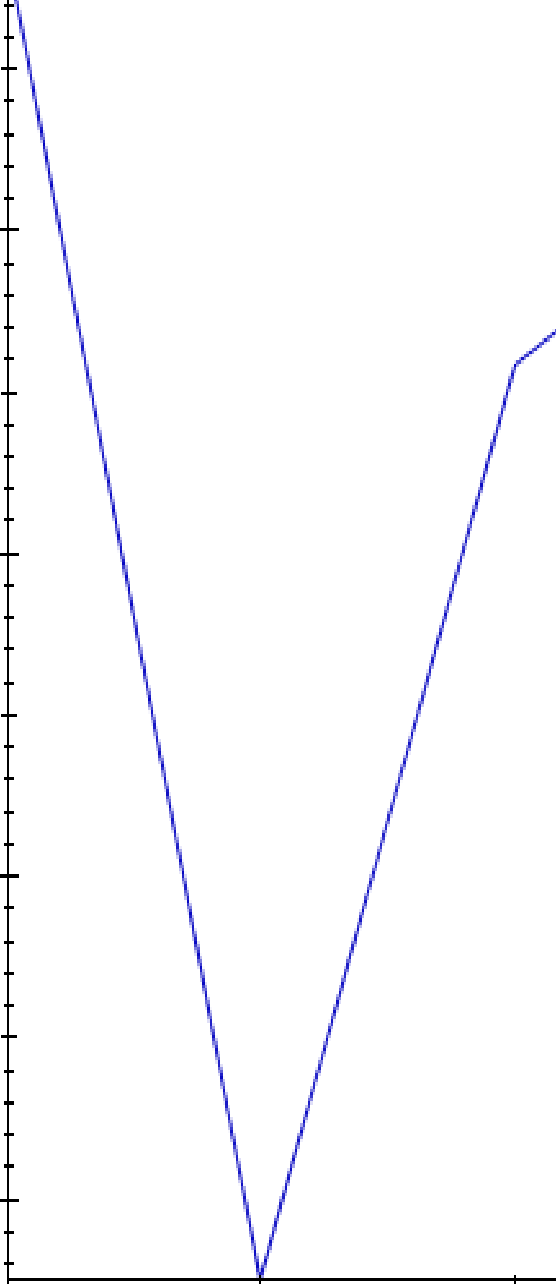
3390 ..410 ..430 ..450 ..470 ..490 ..510 ..530 ..550 ..570 ..590 ..610 ..630 ..650



# Extract the point corresponding to output mask



trace O



1 5 

1 0

5

0

-5

-10

-15

-20

-25

-30

-35

-40

0 2 3 4 5 6 7 8 g 1 0 11 1 2 1 3 14 1 5 16

•I

|  |  |
| --- | --- |
| I  Ae sSe rnndO rderAnalys is o n AES\_ as!ked \_10 0K\_1fi s + Res arn ple d Tri mmed El eginnin gAnd Sbo}j)es + Po intsExtracted + .., •  Samples Traces  F i rs t : l o F i rs t : l o Nu mbe.r: LS I Nu mb er•: '1196 I Masking type Track bit Key b.. ,  I C) Simp le \_ q\_ y r!.9?.9... 1?.J C) Adva nced step 2 0 1 0 2 C) 3, C) 4 C) 5 C) 6 C) 7 C) 8 @ HW  XOR dis tance Roun d Key bits Block bits  11- 15 I @ 1 *2* 9 C) 10 @ 128 C) 192 C) 256 1@ 128 0 192 C) 256  11 Pre vious rnun d key  l l I  I I •  @[R) [§] [!] |  |
|  |
| 11 |
|  |
|  |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| 5t.ep 1 - Advanced ma s ki ng seaond rd·er A:1&5 an a l ys i s - i n i s hed Ke ybyte O :: OX! A at p Bi t i n :: 1  Ke ybyte 4 :: OxDE  Ra t i : 0 . 3500296 | | | |
| Status  Readly | Traces  av ailall le: | 2 displayed: 0 s elected: 0 | Samples Data  av ailallle: 17 displayed:17 at O s elected: 0 38 BA BO DF E 76 8E D3 95 J B, A 7 36 83 E6 26 AC 7 92 11 6B 10 96 F9 I) 30 FIE 1 F 27 2 EF C2 E8 |





.\_I ,. **.&-1. '- ,l .&.'l ,l** ..,,.\_ **'I.I I.A.li .&.'1,,;1, ..1. 1.:1.1.& I.I ' '-** [I I

**t**

**.aL ,1, *.&. J*** 1 f

**'I' '!.& .&. !'.:& '-** Il l

, [I I .\_. .\_ **,1, l p.**

**'!.& I.I ' .f:"'-' 1-'1 ..l. l.l'..&. l..l .li.&. 111**

'-" if

3 ,I

I<iey bytie candidate ::

4 (0x 04} ,, val ue :: 0 ., 051 8,, at p s i t i on::

7,I

**' I.A .li ,l, ' l,,;I, 1.:1. ..1. 1..'1 1.1"1.A.li .&.!li:.I' "-** Il l .

1m d di st anoe:: 1

Ana l yzed ke y bytie 1 5

o,, Ke y bytie can dida.te :: 6 (0x 06} ,, val ue :: 0 ., 2326 ,, a.t p

siti n111

11 ,, and di s t ano e:: 1



Ae sSe rn:nidOrd erAnalysis on AES\_Mas!ked \_HJOKJ .:Gs + Re sam, p le d · Tri mmed Eegi1rmingA ndSbo )>l)es + PoimsErtrad ed ,tr:s

Samples

Fri s t: -1....1 . ,

Nmu ber: 17\_

M asking type

0 mSi ple O Ad vanced step 1

Track bit

Traces

First: l o

Number:: 1: =1t1=2================================

Key b..,

XOR distance

Ro und

Key bits

*2*

g O 10

@ 128 0 192 0 2"56

Block bits

(j;) 128 0 192 0 256

Previous round key

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 ,, Ke y | bytie | c.andida.te | :: | 1 | 5 9 | (0x 9F} ,, | val ue :: | 0 ., 0579,, | a.t | p | siti n | ., a,, | | and di s ta nce :: 1 |  |
| **2,,** I<iey | bybe | C•and ida.te | :: |  | 1 | (0x 0l ) ,1 | val ue :: | 0 ., 055 2,, | a.t | p | siti n | ..,, 11 | ,, | and di s ta noe:: | 1 |
| 3,I I<iey | bytie | candidate | :: |  | 4 | (0x 04} ,1 | val ue :: | 0 ., 0473 ,, | at | p | siti n | ., 11 | ,, | and di s ta noe:: | 1 |
| Anal y:zed | k,ey | bybe 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| o,, I<iey bytie candidate :: 7 (0x 07 } ,, val ue :: 0 ., 1 5 72,, at p siti n., 1 5 ,, and di s t ano e:: 1 1 ,, I<iey byte C•and idate :: 148 (0x 94} ,, val ue :: - 0 ., 0567 ,, at. p s i t .i n..,, a,, and di s ta noe :: 1 **2,,** I<iey bytie candidate :: 1 (0x 0l } ,, val ue :: - 0 ., 0564,, at p s i t i o n:: **9,,** and di st ance :: 1  3 ,, Ke y byte C•and idate :: 4 (0x 04} ,1 val ue :: 0 ., 043 2,, at . p s i t .i n..,, 1 5 ,, and di s t ano e:: 1  *M.S* ke y:: CA IT BA BE !E BE EF 00 01 02 03 04 05 06 07 | | | | | | | | | | | | | | | |

Are first-order analyses (e.g. CPA) successful on a masked implementation?

* + No (unless implementation is not proper)

How can we combine two samples for a high-order attack?

* + Mathematical function: absolute difference, product, square of sum…

Which operations need to be masked in a cipher?

* + Non-linear operations (e.g. S-boxes)

Given the trace set from masked AES implementation obtained from Piñata, perform second-order attack to obtain the used secret key.

Try to do it in backwards incremental steps:

* + Try first the attack with the extracted points traceset
  + Afterwards, create a module to extract the points from the trimmed traceset and try repeating the attack
  + Finally, post-process the original traceset to create an aligned trimmed traceset that allows point extraction

Questions or remarks? ,1scura

