

# **Side Channel Attack (SCA) Toolkit Training Manual**

**For NiCE Hackathon**

(Last Update: 03 September 2024)

## Summary

One of the critical hardware security issues for Integrated Circuits (ICs) and Field Programmable Gate Arrays (FPGAs) is side-channel attack (SCA). By using SCA, an adversary could study the profiles of power dissipation, or electromagnetic (EM) emanation of an Advanced Encryption Standard (AES) device, and then form a correlation to reveal the secret key.

In this training manual, users will go through 6 main hands-on exercises to learn how to use our SCA evaluation tool. The exercises include:

| Exercise | Objective   |
|----------|---|
| 1        | Basic setup –how to use the toolkit to covert a power/EM profile into the required data format for SCAs |
| 2        | Basic data management -how to use the toolkit to manage power/EM traces                                 |
| 3        | How to use the toolkit to attack a microcontroller-based AES in DPA                                     |
| 4        | How to use the toolkit to attack a microcontroller-based AES in CPA                                     |
| 5        | How to use the toolkit to apply a Hamming Weight model to attack a microcontroller-based AES            |
| 6        | How to use the toolkit to apply a Hamming Distance model to attack an FPGA-based AES                    |

Please contact us ([contact@async2secure.com](mailto:contact@async2secure.com)) to get the database of the training data.

### Disclaimer:

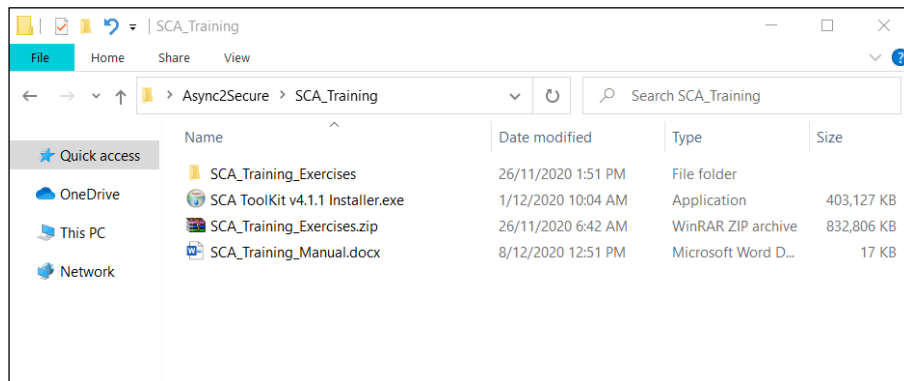
The information contained on this training manual is for general training purposes. The data, graphical user interfaces, setups, symbols, features, labels, wordings, etc. are subject to change without notice. The latest SCA tool version may look different from that contained on this training manual. We assume no responsibility for errors or emissions in the contents of the training manual.

## Table of Contents

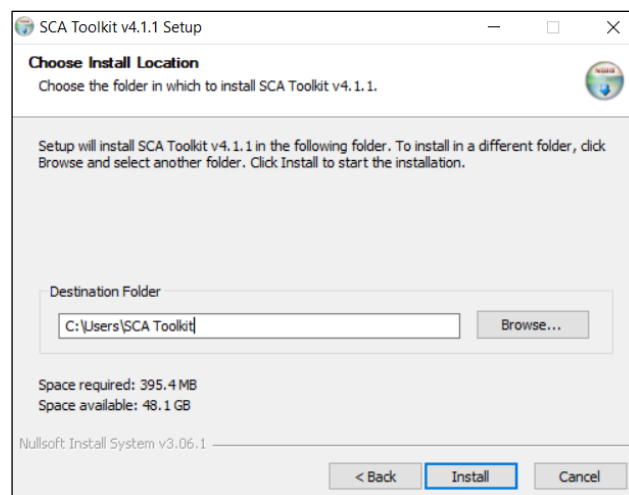
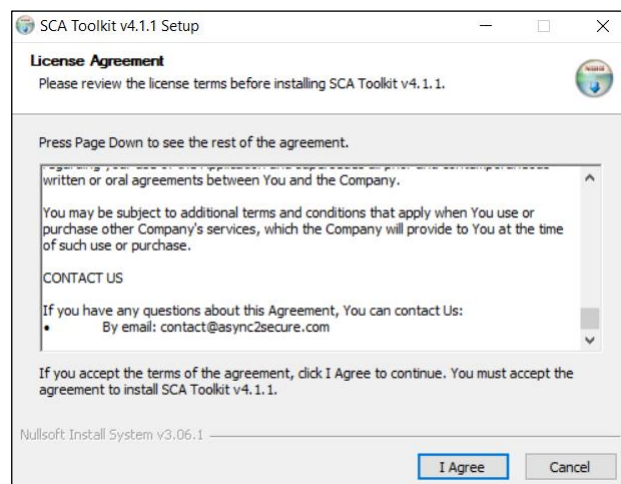
|   |    |
|---|----|
| Summary .....   | 2  |
| Software Setup.....   | 4  |
| Hands-on 1: Basic setup – how to use the toolkit to covert a power/EM profile into the required data format for SCAs..... | 7  |
| Hands-on 2: Basic data management -how to use the toolkit to manage power/EM traces .....                                 | 10 |
| Hands-on 3: How to use the toolkit to attack a microcontroller-based AES in DPA .....                                     | 15 |
| Hands-on 4: How to use the toolkit to attack a microcontroller-based AES in CPA.....                                      | 21 |
| Hands-on 5: How to use the toolkit to apply a Hamming Weight model to attack a microcontroller-based AES.....             | 24 |
| Hands-on 6: How to use the toolkit to apply a Hamming Distance model to attack an FPGA based AES .....                    | 29 |

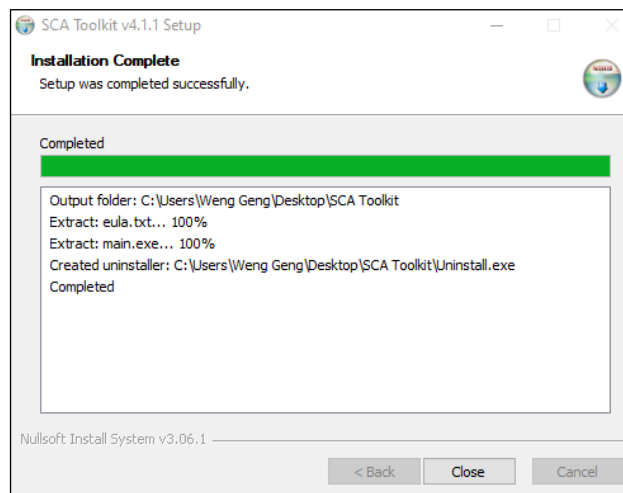
## Software Setup

1. Download the Async2Secure '**SCA Training**' folder into your PC. Double-click on the **EXE** file to install the tool.

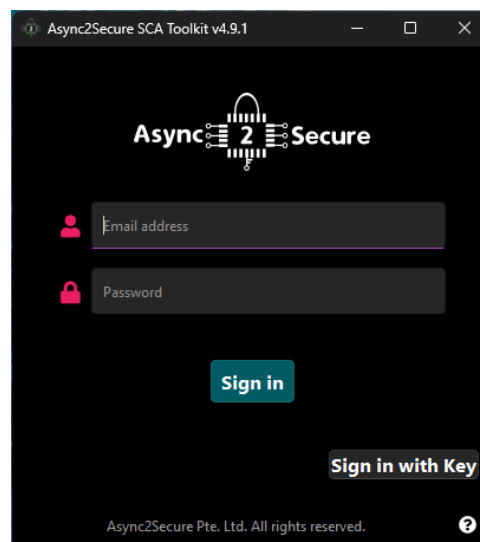


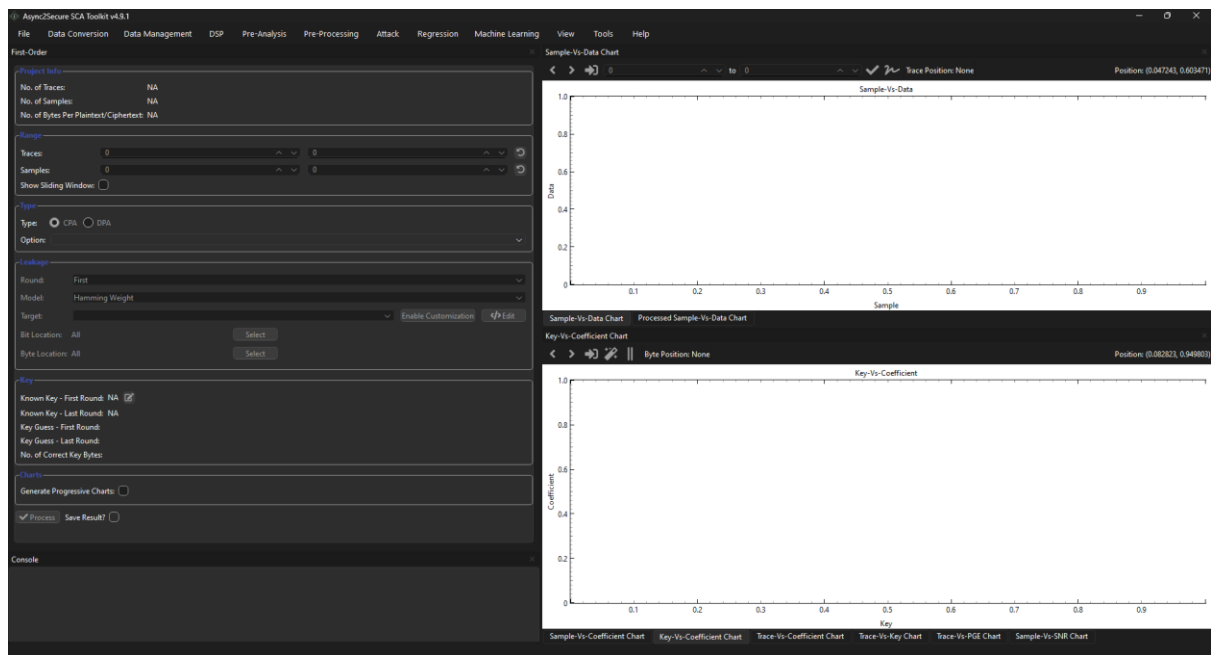
2. Click '**I agree**' to accept the terms of the license agreement. Choose a destination folder for installation, and then click '**Install**'.





3. Double-click the shortcut on the desktop to launch the application.
4. Log in the application with your account. The GUI of the tool will be appeared.

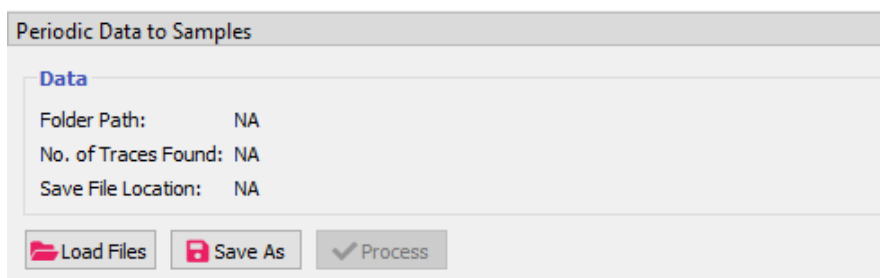
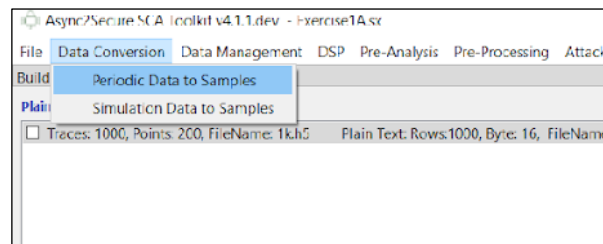




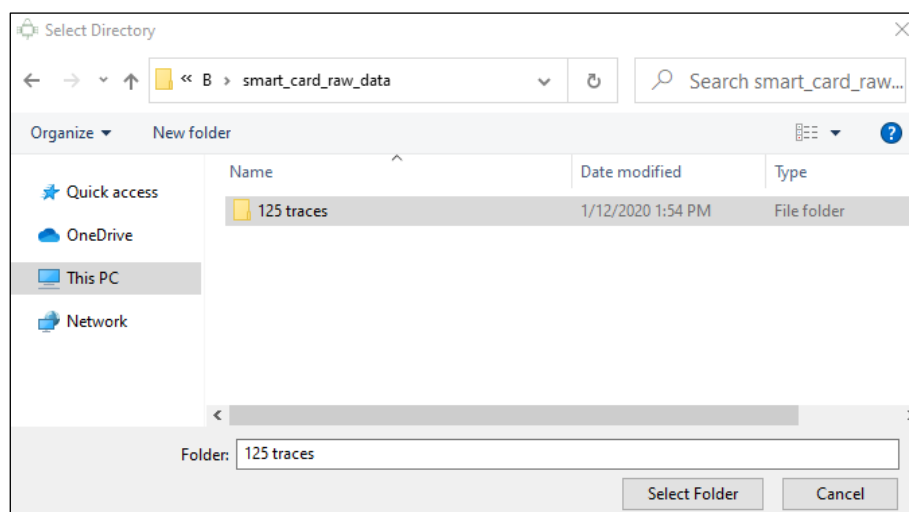
## Hands-on 1: Basic setup – how to use the toolkit to convert a power/EM profile into the required data format for SCAs

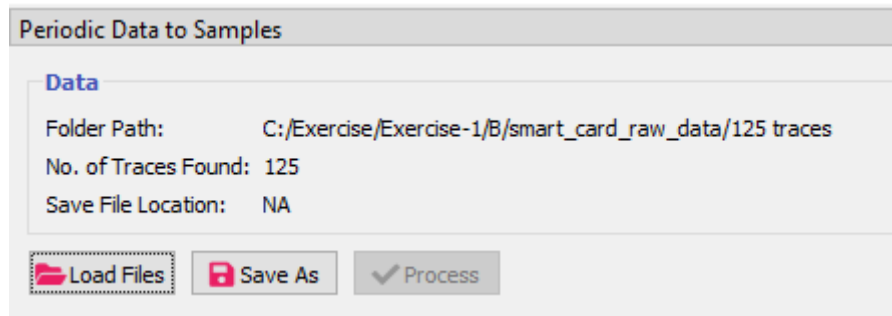
In this exercise, users will learn how to convert a measured power/EM profile (from an oscilloscope) into the required database format. The input is a measured power profiles (.csv) from an oscilloscope. The output is the power trace file (.h5), which will be used to build a project file for SCA evaluation.

1. From the menu bar, click **'Data Conversion' > 'Periodic Data to Samples'**. The **'Periodic Data to Samples'** windows will appear.

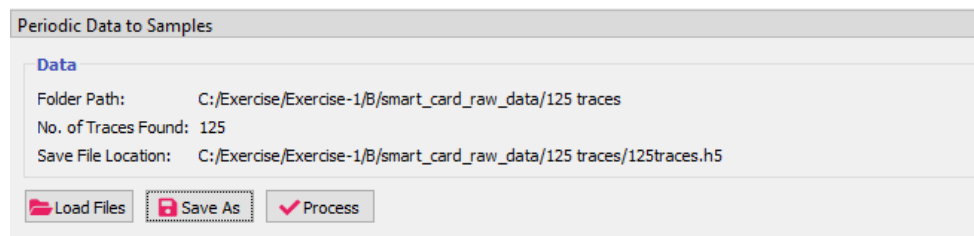
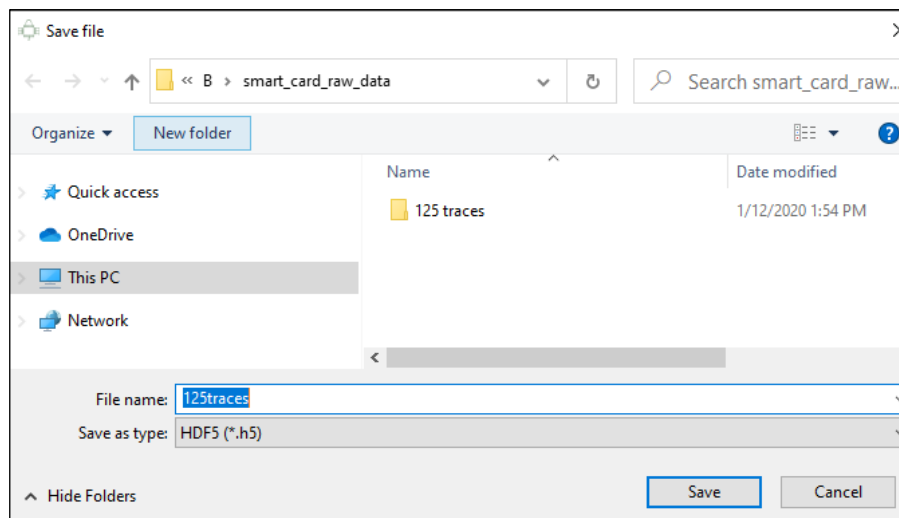


2. Click **'Load Files'**. Select the folder where power measurement files (in CSV format) are stored. The **'Folder Path'** will appear in the window.

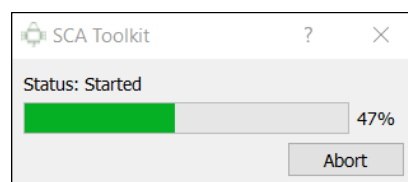




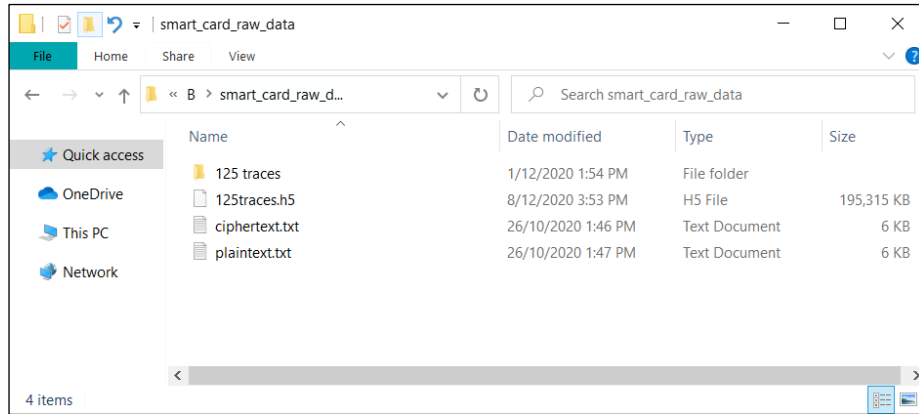
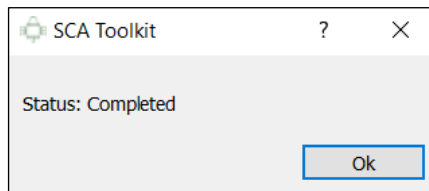
3. Click '**Save As**' and enter the name of the file. The '**Save File Location**' path will appear in the window.



4. Click '**Process**' to start the processing. Once completed, click '**OK**'. The file will be generated in H5 format in the directory.







## Hands-on 2: Basic data management -how to use the toolkit to manage power/EM traces

The required files are the converted power trace file, plaintext file, ciphertext file and a secret key.

Supported file extensions are:

- 1) **Power trace file** (HDF5 binary data format (.h5) and NumPy binary format (.npy))
- 2) **Plaintext** and **ciphertext file** (Text file in hex data (.txt) and NumPy binary format (.npy))

The output is the generated project file (.sx) which includes all the input information. Users will be able to view the power waveform when opening the generated project file for SCA evaluation.

1. Click '**Data Management**' > '**Build Project File**'. The '**Build Project File**' window will appear.



2. Click **'Load Files'**. The **'Add Files'** window will appear.

The 'Add Files' window displays the following information:

| Trace File              |    |
|-------------------------|----|
| No. of Traces:          | NA |
| No. of Sample Points:   | NA |
| Loaded Trace File Path: | NA |

| Plaintext File              |    |
|-----------------------------|----|
| No. of Rows:                | NA |
| No. of Bytes:               | NA |
| Loaded Plaintext File Path: | NA |

| Ciphertext File              |    |
|------------------------------|----|
| No. of Rows:                 | NA |
| No. of Bytes:                | NA |
| Loaded Ciphertext File Path: | NA |

| Key           |    |
|---------------|----|
| No. of Bytes: | NA |
| Key:          | NA |

Buttons at the bottom: Cancel, Load Trace File, Load Plaintext File, Load Ciphertext File, Set Key, Add to Project File.

3. Click **'Load Trace File'** and select a power trace file. Click **'Load Plaintext File'** and select a plaintext file. Click **'Load Ciphertext File'** and select a ciphertext file. Click **'Set Key'** and enter the desired key. Click **'Add to Project File'**. The information for the added files will appear in the **'Build Project File'** window.

The 'Add Files' window displays the following information after files are loaded:

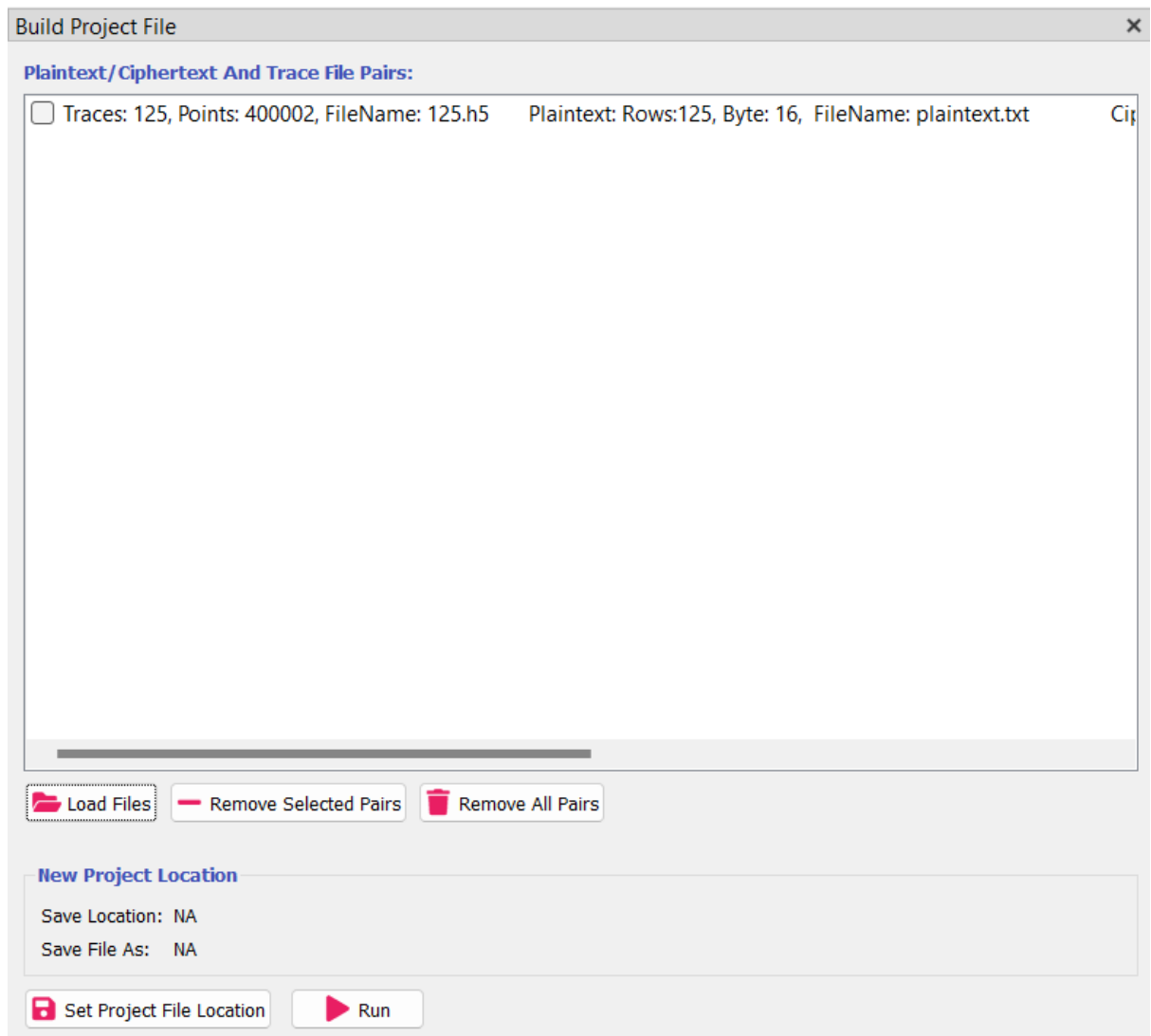
| Trace File              |   |
|-------------------------|---|
| No. of Traces:          | 125   |
| No. of Sample Points:   | 400002  |
| Loaded Trace File Path: | C:/Users/Dell/Desktop/hands-on exercices/raw_data_from_osc/125.h5 |

| Plaintext File              |  |
|-----------------------------|--|
| No. of Rows:                | 125  |
| No. of Bytes:               | 16   |
| Loaded Plaintext File Path: | C:/Users/Dell/Desktop/hands-on exercices/raw_data_from_osc/plaintext.txt |

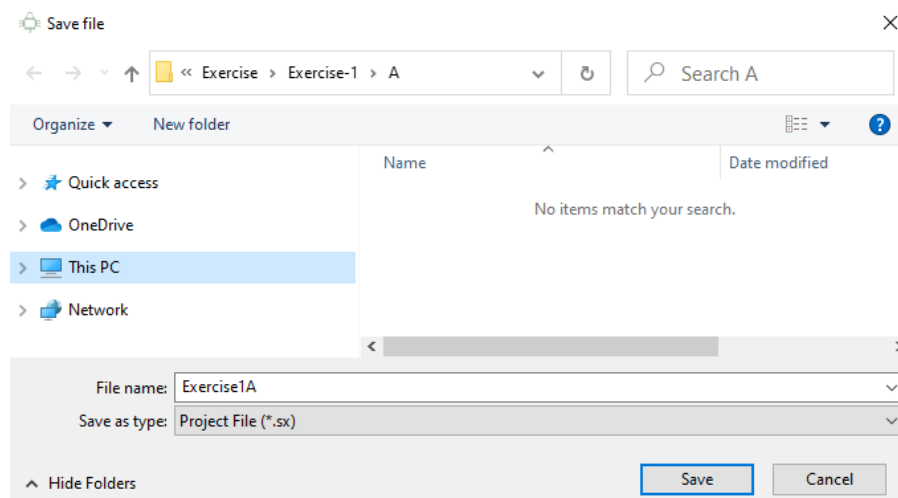
| Ciphertext File              |   |
|------------------------------|---|
| No. of Rows:                 | 125   |
| No. of Bytes:                | 16  |
| Loaded Ciphertext File Path: | C:/Users/Dell/Desktop/hands-on exercices/raw_data_from_osc/ciphertext.txt |

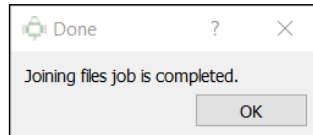
| Single Key/ Key File  |   |
|-----------------------|---|
| No. of Rows:          | NA  |
| No. of Bytes:         | NA  |
| Loaded Key File Path: | NA  |
| Key:                  | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F |

Buttons at the bottom: Cancel, Load Trace File, Load Plaintext File, Load Ciphertext File, Set Key, Add to Project File.

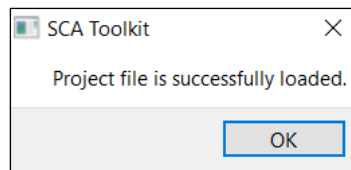
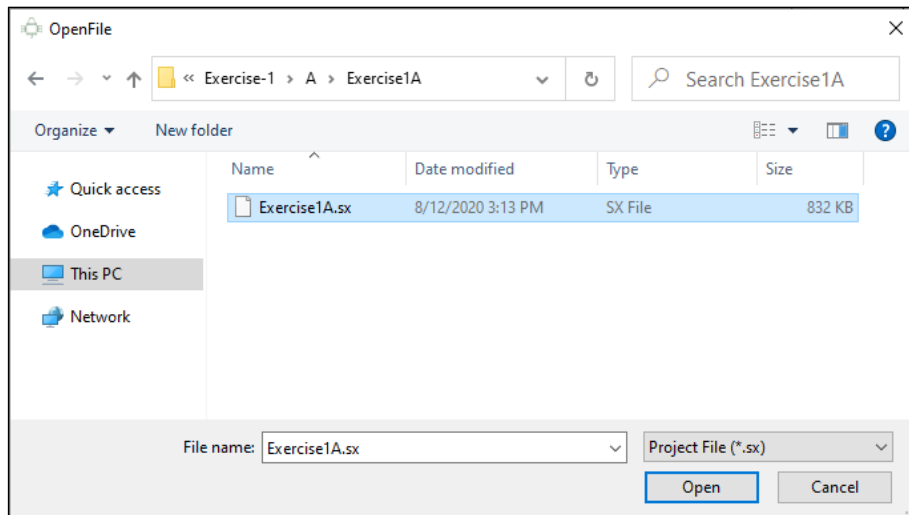
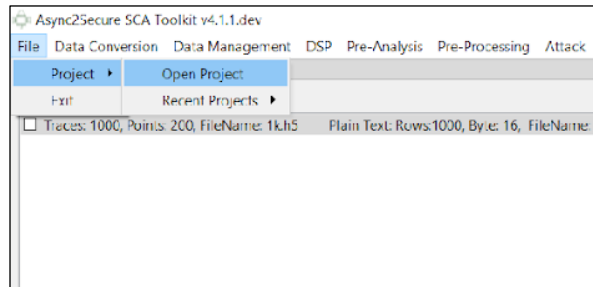


4. Click '**Set Project File Location**', enter the desired project file name, and click '**Save**'. Then click '**Process**' and the '**Done**' window will appear.

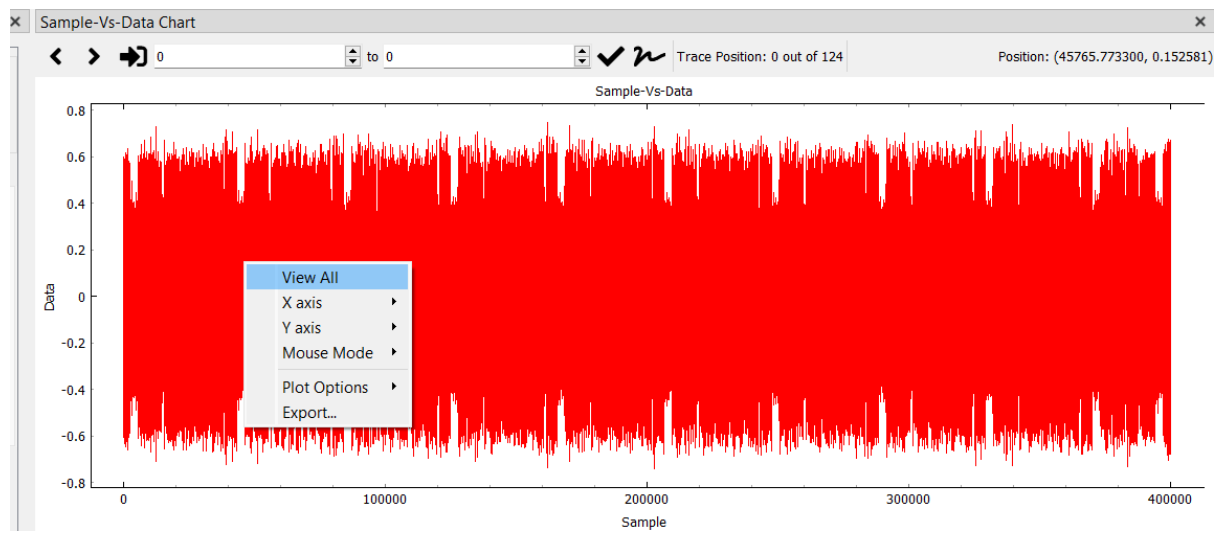




5. From the menu bar, click '**File**' > '**Project**' > '**Open Project**'. Select a project file from the folder created in the previous step, and click '**Open**'.



6. In the '**Traces Chart**' window, right-click the chart and select '**View All**'. The waveform will appear.

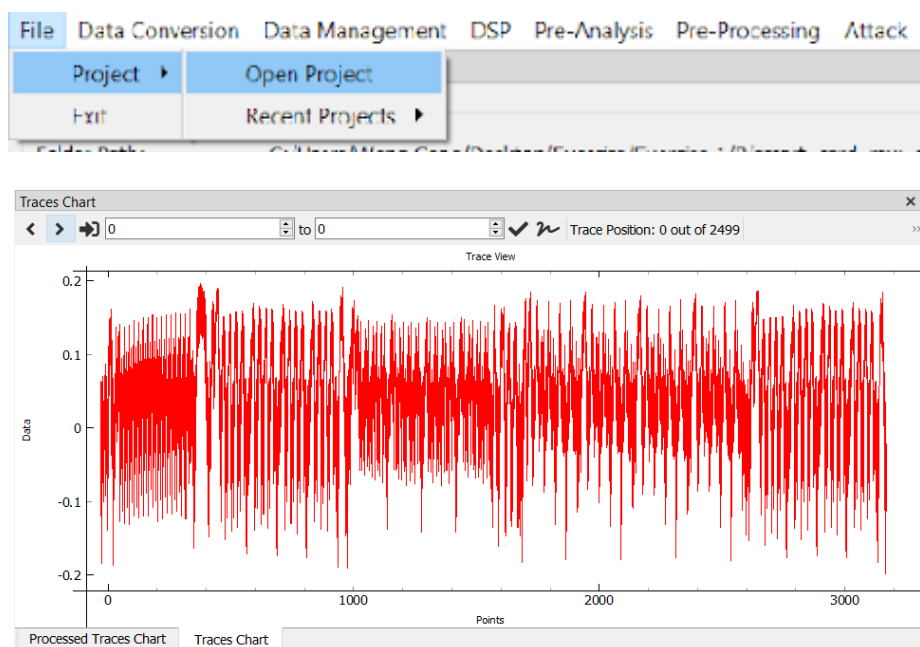


## Hands-on 3: How to use the toolkit to attack a microcontroller-based AES in DPA

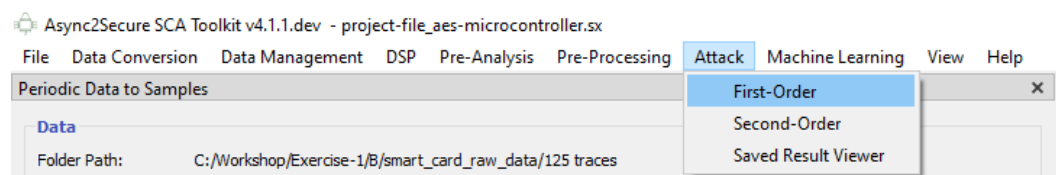
In this exercise, users will learn how to perform a Differential Power Analysis (DPA) on a microcontroller-based AES. This exercise is divided into two parts:

**Attack Part (steps 1 to 6)** – The input is a project file (.sx), which contain the information on power traces, plaintext and ciphertext. A specific key byte, e.g., byte 5 (out of 16 bytes) is selected. The output is the actual key, which can be determined after the attack.

1. From the menu bar, click '**File**' > '**Project**' > '**Open Project**'. Select a project file (\*.sx). The waveform will appear in the '**Trace-Vs-Key Chart**'.



2. From the menu bar, click '**Attack**' > '**First-Order**'. The '**First-Order**' window will appear.



Async2Secure SCA Toolkit v4.1.1.dev - project-file\_aes-microcontroller.sx

File Data Conversion Data Management DSP Pre-Analysis Pre-Processing Attack Machine Learning View Help

First-Order

**Project Info**

No. of Traces: 2500  
 No. of Samples: 4000  
 No. of Bytes Per Plaintext/Ciphertext: 16

**Range**

Traces: 0 2500  
 Samples: 0 4000

**Type**

Type: ☒ CPA ☐ DPA  
 Option:

**Leakage**

Round: First  
 Model: Hamming Weight  
 Target: PT^KEY  
 Bit Location: All   
 Bytes: 5

**Key**

Known Key: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   
 Expected Key: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
 Actual Key: NA  
 No. of Correct Key Bytes: 0 out of 16

**View**

Show Window: ☐  
 Generate Progressive Charts: ☐

3. Select the following parameters for DPA analysis.

- **Traces:** All
- **Samples:** All
- **Type:** DPA
- **Option:** Mean
- **Round:** First
- **Model:** Hamming Weight
- **Target:** SB( $PT^{KEY}$ )
- **Bit Location:** 0
- **Bytes:** 5 (i.e., 6<sup>th</sup> byte)



**Range**

Traces: 0 2500

Samples: 0 4000

Show Sliding Window: ☐

**Type**

Type: ☐ CPA ☒ DPA

Option: Mean

**Leakage**

Round: First

Target: SB(PT^K^KEY)

Bit Location: 0 Select

Bytes: 5 Select

- Click '**Select**' for Bytes and check Byte '**5**' in the Bytes selection window

Byte Selection

Select All Deselect All

|                          |                          |                          |                          |                          |                                     |                          |                          |                          |                          |                          |                          |                          |                          |                          |                          |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 0                        | 1                        | 2                        | 3                        | 4                        | 5                                   | 6                        | 7                        | 8                        | 9                        | 10                       | 11                       | 12                       | 13                       | 14                       | 15                       |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

OK Cancel

- Select the following parameters for the progressive analysis:
  - Generate Progressive Charts:** Select
  - Batch Size:** 100 (Click the '**Change**' button to adjust the size)
  - Trace-Vs-Coefficient Chart:** Select
  - Trace-Vs-Key Chart:** Select

**View**

Show Window: ☒

Generate Progressive Charts: ☒

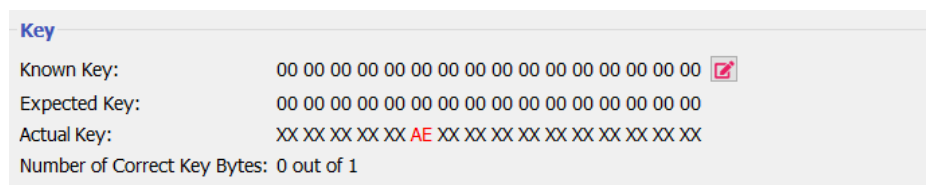
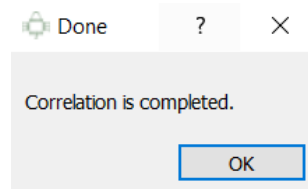
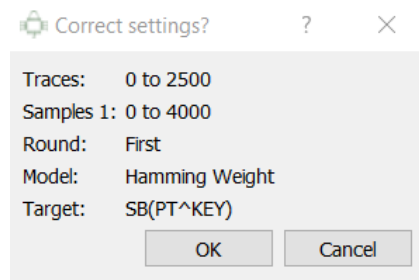
Batch Size: 100 Change

Trace-Vs-Coefficient Chart: ☒

Trace-Vs-Key Chart: ☒

☒ Process Save Result? ☐

- Click '**Process**'. The processing will run and complete. The correct key for Byte 5 is found: **AE** (in hex).

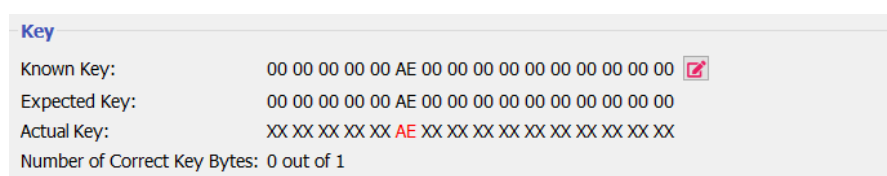
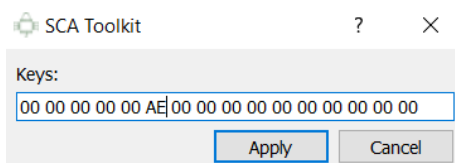


**Analysis Part (steps 7 to 8)** – The input is the user-defined known key (from previous steps), which is used to perform analysis for DPA evaluation. The outputs are

- **Trace-Vs-Key** chart (to determine the minimum number of traces needed to reveal the correct key)
- **Sample-Vs-Coefficient** chart (to identify the sample point which has the highest coefficient value),
- **Key-Vs-Coefficient** chart (to find the correct key with the highest coefficient value)
- **Trace-Vs-Coefficient** chart (to analyze how the coefficient value of the correct key changes with an increasing number of traces).

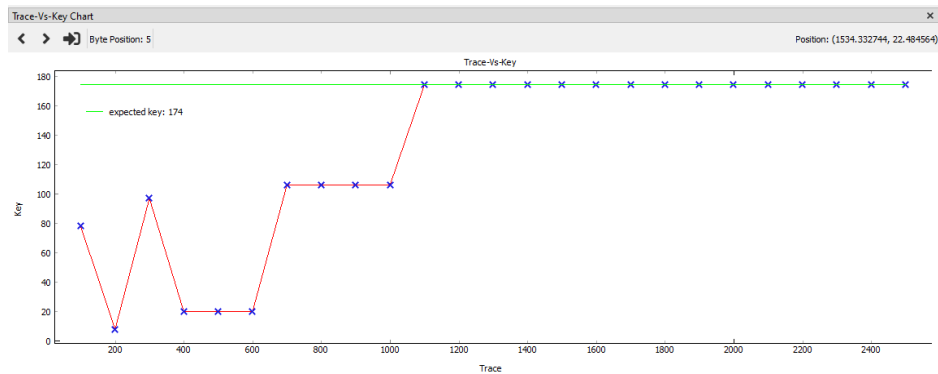
In this example, the DPA requires 1,100 traces to reveal a correct key.

- At the '**Key**' part, set the Known Key of Byte 5 to be '**AE**'. More analysis could be done.

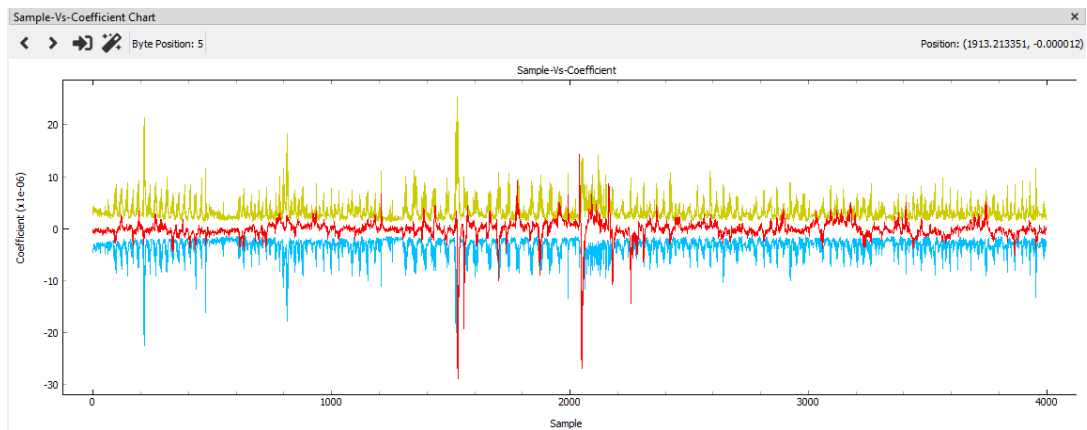


- Click '**Process**'. The analysis will run and complete. Right-click on the chart and select '**View All**' to see the full view.

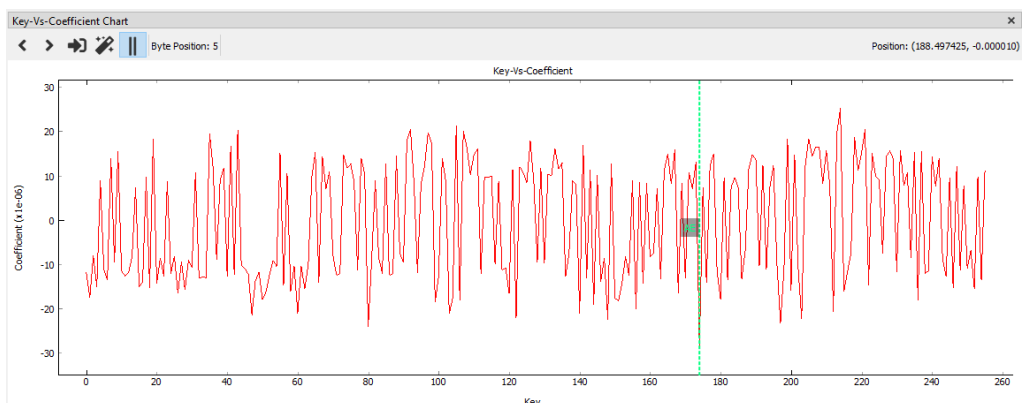
- (a) In the **'Trace-Vs-Key Chart'**, the minimum number of traces needed to reveal the correct key can be observed.



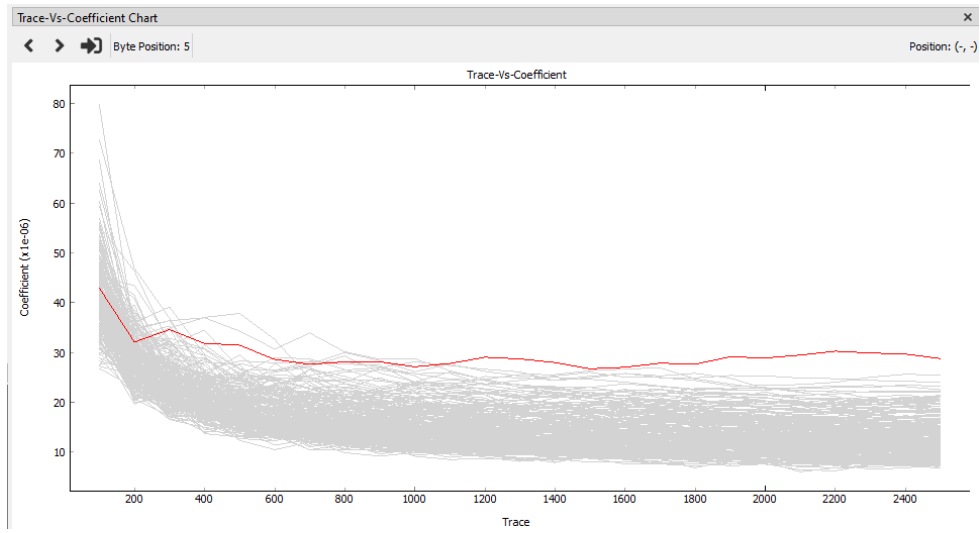
- (b) In the **'Sample-Vs-Coefficient Chart'**, the highest coefficient value is found.



- (c) In **'Key-Vs-Coefficient Chart'**, the correct key 'AE' is found at the highest coefficient value by clicking 'II'.



- (d) In **'Trace-Vs-Coefficient Chart'**, the coefficient value of the correct key (as compared to other key candidates) is plotted with increasing number of traces.



## Hands-on 4: How to use the toolkit to attack a microcontroller-based AES in CPA

In this exercise, users will learn how to perform a Correlation Power Analysis (CPA) on a microcontroller-based AES. The exercise is divided into three parts:

**Attack Part (step 1)** – The input is a project file (.sx) containing information on power traces, plaintext and ciphertext. To solve the long attack duration of DPA in Exercise 4, first-order CPA is used. A specific key byte, e.g. byte 5 (out of 16 bytes) is selected for the attack similarly. The output is the actual key, which could be determined after the attack.


1. From the menu bar, click '**Attack**' > '**First-Order**'.
2. Select the following parameters for CPA analysis.

- **Traces:** 0 to 100
- **Samples:** All
- **Type:** CPA
- **Round:** First
- **Model:** Hamming Weight
- **Target:** SB( $PT^{KEY}$ )
- **Bit Location:** All
- **Bytes:** 5 (i.e., 6<sup>th</sup> byte)

The screenshot shows a configuration window for CPA analysis, divided into three sections: Range, Type, and Leakage.

- Range:** Contains input fields for 'Traces' (0 to 100) and 'Samples' (0 to 4000), each with a refresh button. A 'Show Sliding Window' checkbox is present and unchecked.
- Type:** Contains a 'Type' section with radio buttons for 'CPA' (selected) and 'DPA'. Below it is an 'Option' dropdown menu.
- Leakage:** Contains dropdown menus for 'Round' (First), 'Model' (Hamming Weight), and 'Target' (SB( $PT^{KEY}$ )). Below these are 'Bit Location' (All) and 'Bytes' (5), each with a 'Select' button.

3. Set 6<sup>th</sup> byte of Known Key to 'AE'.

| Key           |  |
|---------------|--|
| Known Key:    | 00 00 00 00 00 AE 00 00 00 00 00 00 00 00 00 00 00  |
| Expected Key: | 00 00 00 00 00 AE 00 00 00 00 00 00 00 00 00 00 00   |

4. Select the following parameters for progressive analysis.
  - **Generate Progressive Charts:** Select
  - **Batch Size:** 10 (Click the 'Change' button to adjust the size)
  - **Trace-Vs-Coefficient Chart:** Select
  - **Trace-Vs-Key Chart:** Select

| Charts                       |                                     |
|------------------------------|-------------------------------------|
| Generate Progressive Charts: | <input checked="" type="checkbox"/> |
| Batch Size:                  | 10 <span>Change</span>              |
| Trace-Vs-Coefficient Chart   | <input checked="" type="checkbox"/> |
| Trace-Vs-Key Chart           | <input checked="" type="checkbox"/> |

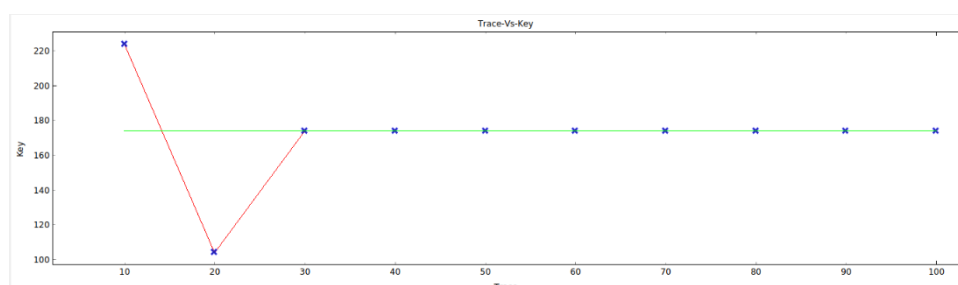
**Analysis Part (step 2)** – The input is the user-defined known key (from the previous attack), which is used to perform analysis for CPA evaluation. The output include:

- **Trace-Vs-Key** chart
- **Sample-Vs-Coefficient** chart
- **Key-Vs-Coefficient** chart
- **Trace-Vs-Coefficient** chart

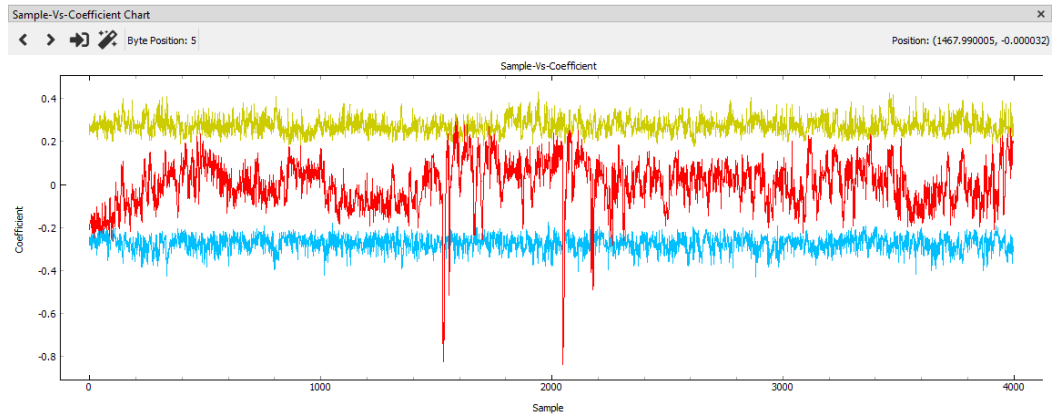
In this example, the CPA requires only 30 traces to reveal the correct key, compared to 1,100 traces in the DPA.

5. Click '**Process**'. Once the analysis is completed, right-click on the chart and click '**View All**' to see the full view.

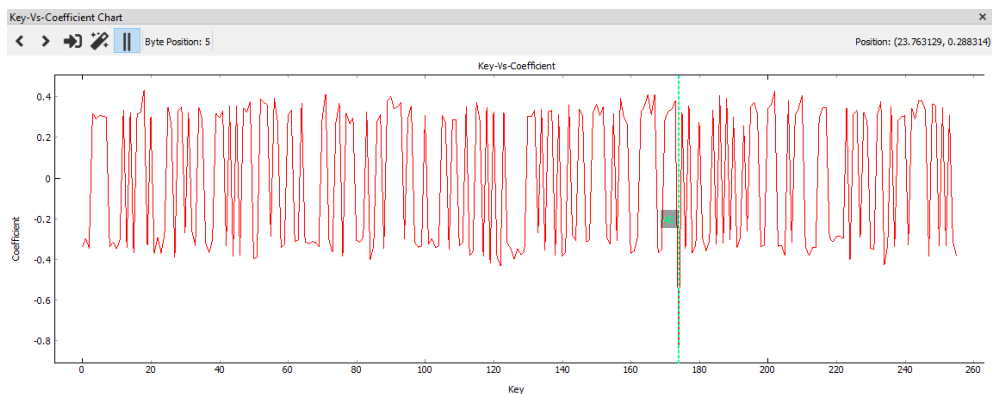
- (a) In '**Trace-Vs-Key Chart**', the minimum number of traces to reveal the correct key is found.



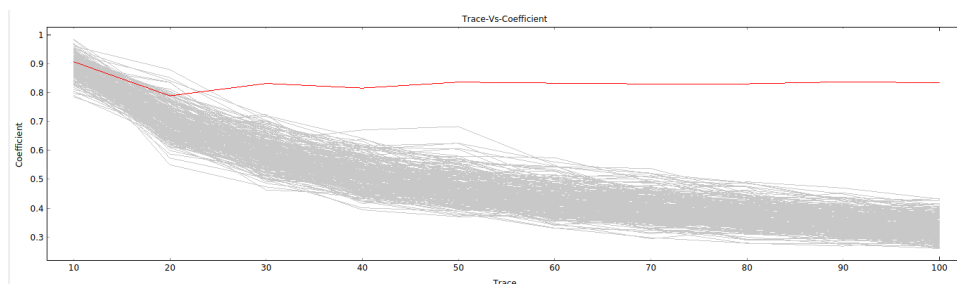
(b) In '**Sample-Vs-Coefficient Chart**', the highest coefficient value is found.



(c) In '**Key-Vs-Coefficient Chart**', the correct key '**AE**' is found at the highest coefficient value by clicking '**II**'.



(d) In '**Trace-Vs-Coefficient Chart**', the correlation coefficient value of the correct key (as compared to other key candidates) is plotted with an increasing number of traces.



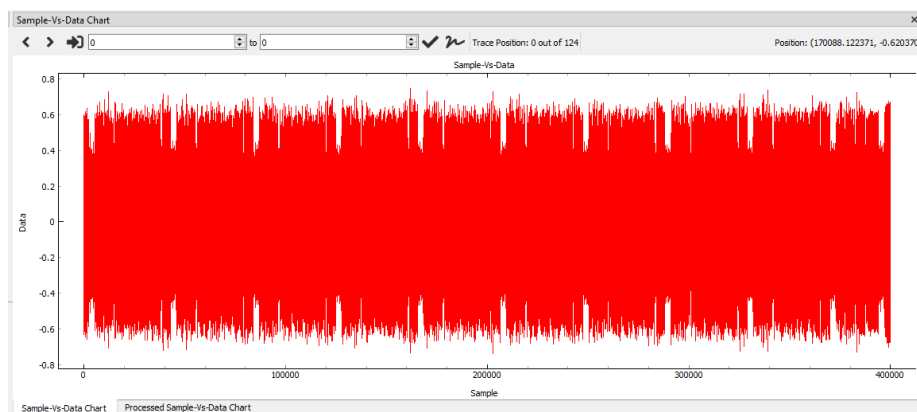
## Hands-on 5: How to use the toolkit to apply a Hamming Weight model to attack a microcontroller-based AES

This exercise involves performing a basic Correlation Power Analysis (CPA) to reveal secret keys in AES implementations on commercial chips. Users will learn how to conduct a CPA using Hamming Weight on a microcontroller-based AES. The exercise is divided into two parts:

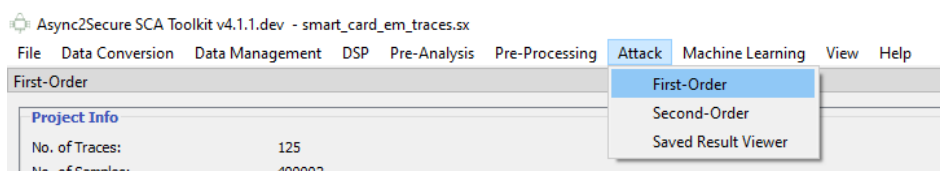
**First Round Hamming Weight CPA (steps 1 to 4)** – The input is a project file (.sx) from a smart card AES. Users will need to set the range of samples where the first-round AES operation occurs. The outputs include:

- The 16 bytes of correct key
- Analysis of coefficient values at different sample points
- Key candidates for each byte of the key.

1. From the menu bar, click **'File' > 'Project' > 'Open Project'**. Select the smart card project file. The waveform will appear in the **'Sample-Vs-Data Chart'**.



2. From the menu bar, click **'Attack' > 'First-Order'**. The **'First-Order'** window will appear.



3. **First Round Hamming Weight Attack:** Select the following parameters

- **Traces:** All
- **Samples:** First Round Peak (14,000 to 4,400)
- **Type:** CPA
- **Round:** First
- **Model:** Hamming Weight



- **Target:** SB(PT^K<sup>KEY</sup>)
- **Bit Location:** All
- **Bytes:** All

**Range**

Traces:

Samples:

Show Sliding Window: ☐

**Type**

Type: ☒ CPA ☐ DPA

Option:

**Leakage**

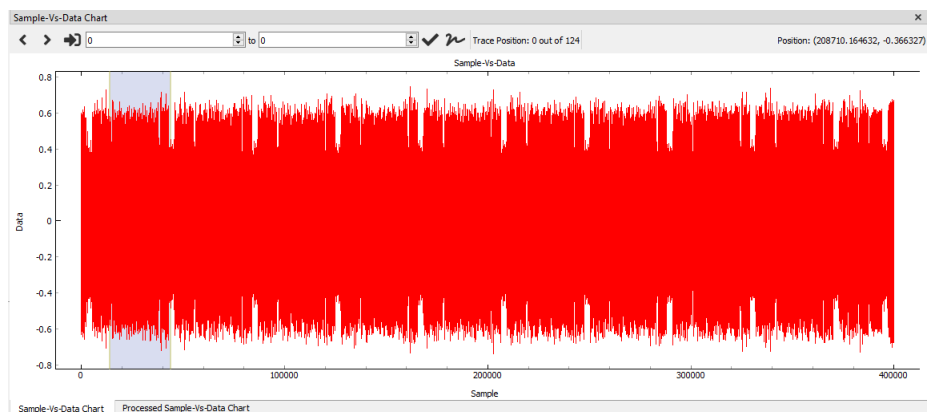
Round:

Model:

Target:


Bit Location: All

Bytes: All

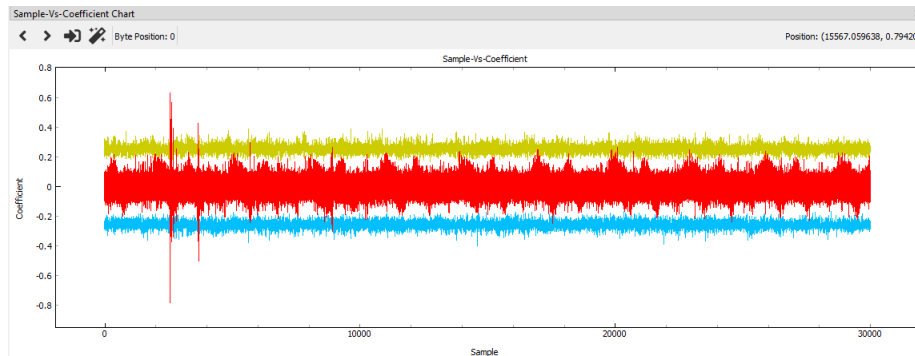


4. Click '**Process**'. Click '**OK**' for processing.

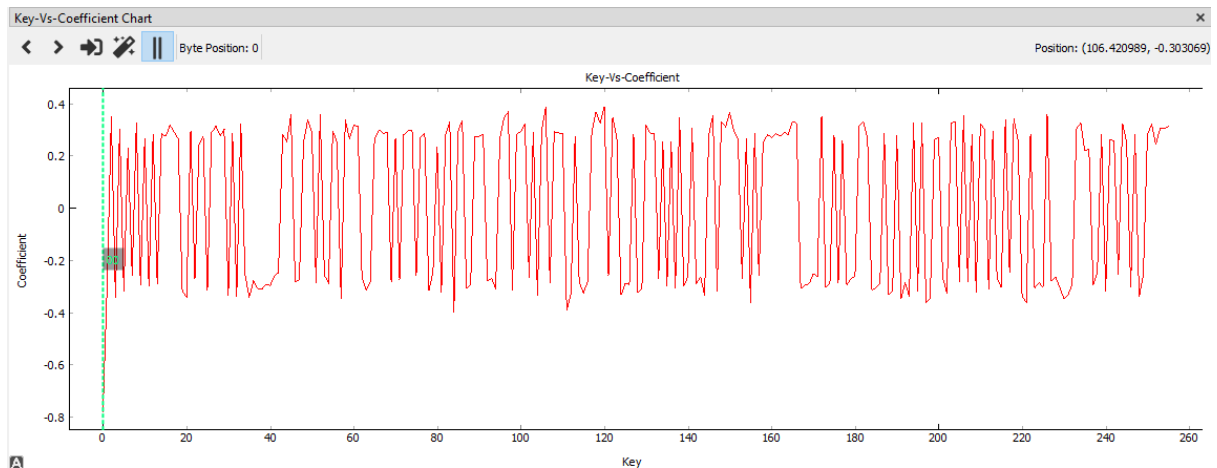
(a) In '**Key**' part, the actual keys for all 16 bytes are found.

|                           |   |
|---------------------------|---|
| <b>Key</b>                |   |
| Known Key:                | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  |
| Expected Key:             | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   |
| Actual Key:               | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   |
| No. of Correct Key Bytes: | 16 out of 16  |

- (b) In '**Sample-Vs-Coefficient Chart**', the highest correlation coefficient value for key byte 0 is found. Click '>' for the analysis of different key bytes.



- (c) In '**Key-Vs-Coefficient Chart**', the correct key '00' for byte 0 is found at the highest correlation value by clicking 'II'. Click '>' for the analysis of different key bytes.



**Last Round Hamming Weight CPA (steps 5 to 6)** – the input is the project file (.sx) of a smart card AES. The users will need to set the range of samples where the last round AES operation is performed. The outputs are similar to the first round Hamming Weight CPA.

5. **Last Round Hamming Weight Attack:** Select the following parameters

- **Traces:** All
- **Samples:** Last Round Peak (370,000 to 384,000)
- **Type:** CPA
- **Round:** Last
- **Model:** Hamming Weight
- **Target:**  $ISB(CT^{KEY})$
- **Bit Location:** All
- **Bytes:** All

**Range**

Traces:

Samples:

Show Sliding Window: ☒

**Type**

Type: ☒ CPA ☐ DPA

Option:

**Leakage**

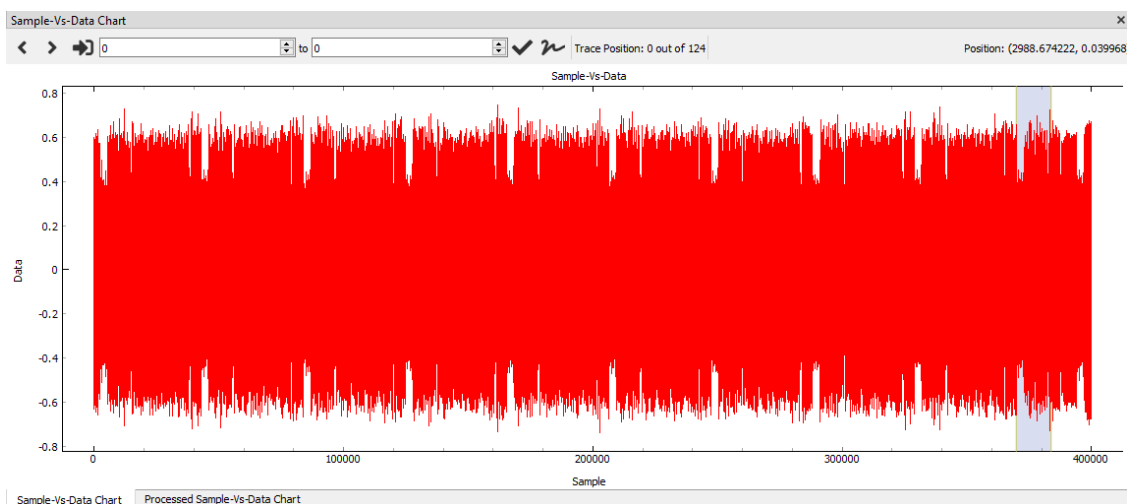
Round:

Model:

Target:

Bit Location: All

Bytes: All



6. Click '**Process**'. Repeat step 4 for the analysis.

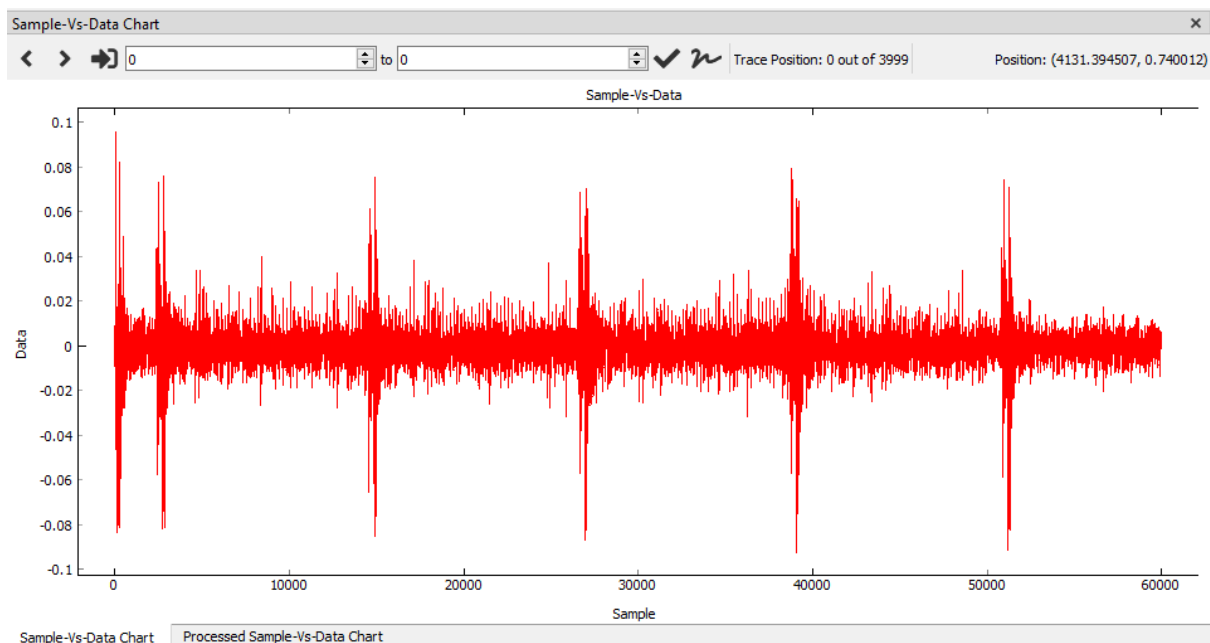
## Hands-on 6: How to use the toolkit to apply a Hamming Distance model to attack an FPGA based AES

In this exercise, users will learn how to perform a Correlation Power Analysis (CPA) using Hamming Distance on a microcontroller-based AES. The exercise is divided into two parts:

**First Round Hamming Distance CPA (Steps 1 to 4)** – The input is a project file (.sx) of a FPGA-based Nano-AES. Users will need to set the range of samples where the first round AES operation occurs. The outputs include:

- The 16 bytes of correct key
- Analysis of coefficient values at different sample points
- Key candidates for each byte of the key.

1. From the menu bar, click '**File**' > '**Project**' > '**Open Project**'. Select the nano AES design project file. The waveform will appear in the '**Sample-Vs-Trace Chart**'.



2. From the menu bar, click '**Attack**' > '**First-Order**'.
3. **First Round Hamming Distance Attack:** Select the following parameters
  - **Traces:** 0 to 2000
  - **Samples:** 1900 to 9000
  - **Type:** CPA
  - **Round:** First
  - **Model:** Hamming Distance
  - **Target:**  $(PT^{KEY}) \wedge (SB(PT^{KEY}))$
  - **Bytes:** All

**Range**

Traces: 0 2000

Samples: 1900 9000

Show Sliding Window: ☒

**Type**

Type: ☒ CPA ☐ DPA

Option:

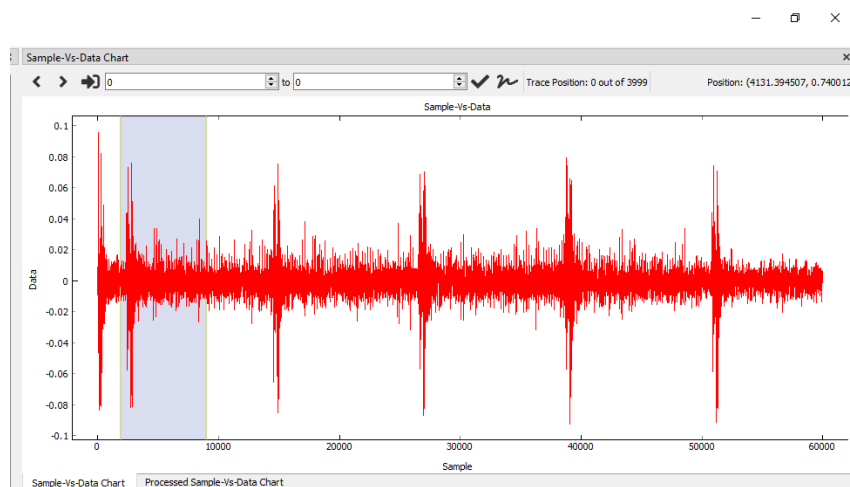
**Leakage**

Round: First

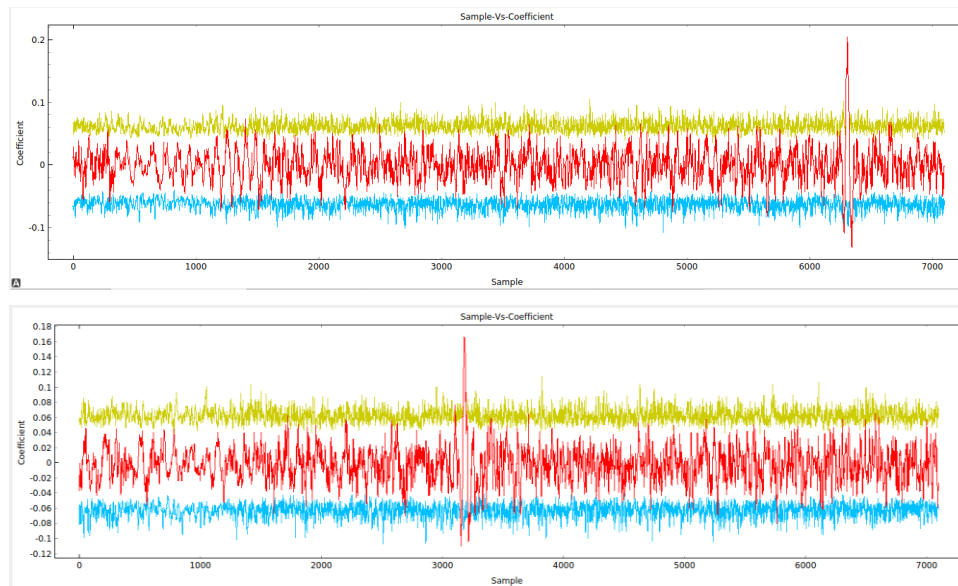
Model: Hamming Distance

Target:  $(PT^{KEY}) \wedge (SB(PT^{KEY}))$

Bytes: All Select



- Click '**Process**' to start the analysis. Once the analysis is completed, note that each leakage region varies for each byte.



**Last Round Hamming Distance CPA (steps 5 to 6)** – The input is a project file (.sx) of a FPGA-based Nano-AES. Users will need to set the range of samples where the last round AES operation occurs. The outputs are similar to those from the first round Hamming Distance CPA.

5. **Last Round Hamming Distance Attack:** Select the following parameters

- **Traces:** All
- **Samples:** 41,000 to 51,000
- **Type:** CPA
- **Round:** Last
- **Model:** Hamming Distance
- **Target:**  $(CT^{KEY}) \wedge (ISB(CT^{KEY}))$
- **Bytes:** All

**Range**

Traces:

Samples:

Show Sliding Window: ☒

**Type**

Type: ☒ CPA ☐ DPA

Option:

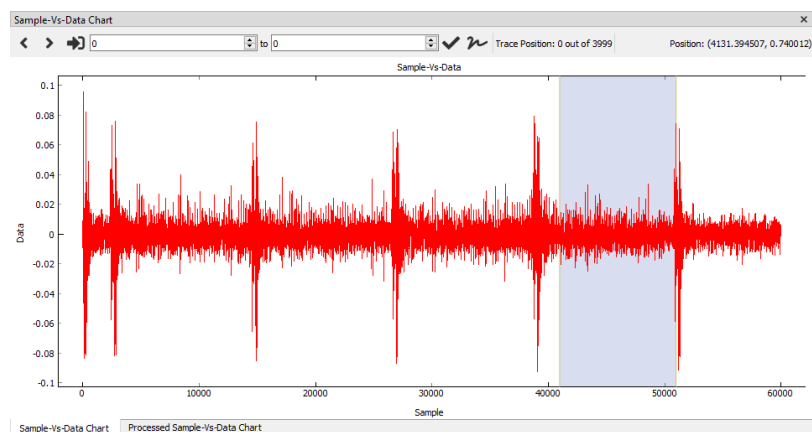
**Leakage**

Round:

Model:

Target:

Bytes: All



- Click '**Process**' to start the analysis.