CPA Tool for DPA Contest

[Version 0.91]

June 16, 2011

Research Center for Information Security,
National Institute of Advanced Industrial Science and Technology

History

Version 0.91	Section 1.1 and Chapter 3 have added.

1. OVERVIEW

The Correlation Power Analysis (CPA) tool was designed to analyze power waveforms of an AES circuit on the SASEBO-GII board (http://staff.aist.go.jp/akashi.satoh/SASEBO/en/board/sasebo-g2.html) for the third DPA contest (http://www.dpacontest.org/) organized by TLELECOM ParisTech University. In order to control the AES circuit and to capture a data set of the power waveforms, please use a waveform acquisition tool, which is published on the SASEBO Website (http://staff.aist.go.jp/akashi.satoh/SASEBO/en/DPAcontest/index.html). The programs and the source codes can be freely used and modified for academic research use only.

1.1 NOTE

The CPA tool focuses on 10th round and reports the 10th round key and their correlation values. If user needs initial key, they should reverse the reported key to initial key.

An round key checking tool "RoundKey" is also provided. See section 3.

2. USAGE

2.1 Syntax

The syntax of the command line CPA tool is described below. Only "acquisition_data_dir" must be specified, and the other parameters are optional.

```
CPA.EXE [-d=DBFILE] [-f=FILE] [-s=START] [-e=END] [-t=THREADS] [-i=INTERVAL] [-q]
aquisition_data_dir
```

acquisition_data_dir

This parameter specifies the directory where the data set of the power waveforms "info.xml", "text_out.txt" and "wave.data" are stored.

-d=DBFILE

This parameter specifies the SQLlite database file to record correlation values of all partial key candidates. No file is generated when this parameter is not specified. Details are described in Section 2.2.

-f=FILE

This parameter specifies a CSV file to record the maximum correlation values of all the key candidates. No file is generated when this parameter is not specified. Detail of the format is shown in Table 1.

-s=START

This parameter specifies a start position (time) of the waveform to be analyzed. The default value is 0.

-e=END

This parameter specifies an end position (time) of the waveform to be analyzed. The default value is the last position.

-t=THREADS

This parameter specifies a number of invoked threads for multi-core CPUs. One of the numbers 2, 4, 8 and 16 can be selected, and the default value is 2.

-i=INTERVAL

This parameter specifies an interval number of the power waveforms to display or to store intermediate results. For example, "–i=2000" is specified and the number of waveforms is 10000, the program outputs the intermediate results for 2000, 4000, 6000, 8000 and 10000 waveforms. The default value is the maximum number of the waveforms, and thus the result is output only ounce after all the waveforms are processed.

-q

When this parameter is specified, the result is not displayed on screen. If the parameter is not specified, the five key candidates corresponding to the highest five correlation values are displayed.

Table 1 CSV format

reported time				
proceed traces	pk0	pk1	pk14	pk15
0x00	max correlation0	max correlation0	max correlation14	max correlation15
: :				
0xFF	max correlation0	max correlation0	max correlation14	max correlation15

2.2 SQLite Database

User can store the results into SQLlite database files indicated by the "-d" option, while process speed is reduced. A new file is created to store the results at every interval indicated by the "-i" option. Each file name has the number of interval as its suffix. When each waveform contains 10000 data points, and all the points from 0 to 9999 are analized, 40960000 (= 10000 * 256 * 16) rows are generated at each interval. The schema is shown below:

TABLENAME Correlation

ID INTEGER PRIMARY KEY
-- ID is assigned automatically.
KeyId INTEGER
-- KeyId indicates the S-box (0-15) where the partial 8-bit key belongs.
KeyValue INTEGER
-- The value of the 8-bit partial key candidate (0-155).
Position INTEGE
-- Postion is the position in waveform, the range depends on waveform.
Value DOUBLE.
-- Value is the calcrated correlation.

The file shall follow the sqlite3 database format (Please lefere Sqlite3 Homepage (http://www.sqlite.org/) and System.Data.SQLite: C# sqlite library (http://sqlite.phxsoftware.com/)) so that user can process the file using the SQL commands like as the examples shown below. The examples are contained in the source code package of the CPA tool.

```
%sqlite3 file.db < select_max_correlation.sql .
```

<Example 1>

Extract the correlation values of each key candidate along with the number of waveforms for S-box0.

```
SELECT KeyValue, Position, Value
FROM Correlation
WHERE KeyId = 0
ORDER BY KeyValue, Position;
```

<Example 2>

Extract the partial key candidate with the maximum correlation, its position (associated S-box), and other values. If several rows have the same maximum value, all of them are reported.

```
SELECT a.KeyId, a.KeyValue, a.Position, a.Value
FROM Correlation a,
  (SELECT KeyId, max(Value) AS max FROM Correlation GROUP BY KeyId) b
WHERE a.KeyId=b.KeyId AND a.Value=b.max;
```

3. Round key checking tool, "RoundKey"

The "RoundKey" is a program that solves each round key from initial key or 10th round key for checking CPA analysis results.

3.1 USAGE

The usage is following.
Note that key must be separated with space on each partial key.

```
RoundKey.EXE [-r] key
```

-r

When this parameter is specified, the program treats the key as 10th round key.

<Example 1>

Get round keys from initial key.

```
RoundKey.exe "00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f"
00: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
01: D6 AA 74 FD D2 AF 72 FA DA A6 78 F1 D6 AB 76 FE
02: B6 92 CF 0B 64 3D BD F1 BE 9B C5 00 68 30 B3 FE
03: B6 FF 74 4E D2 C2 C9 BF 6C 59 0C BF 04 69 BF 41
04: 47 F7 F7 BC 95 35 3E 03 F9 6C 32 BC FD 05 8D FD
05: 3C AA A3 E8 A9 9F 9D EB 50 F3 AF 57 AD F6 22 AA
06: 5E 39 0F 7D F7 A6 92 96 A7 55 3D C1 0A A3 1F 6B
07: 14 F9 70 1A E3 5F E2 8C 44 0A DF 4D 4E A9 C0 26
08: 47 43 87 35 A4 1C 65 B9 E0 16 BA F4 AE BF 7A D2
09: 54 99 32 D1 F0 85 57 68 10 93 ED 9C BE 2C 97 4E
10: 13 11 1D 7F E3 94 4A 17 F3 07 A7 8B 4D 2B 30 C5
```

<Example 2>

Get round keys from the 10th round key.

```
RoundKey.exe -r "13 11 1D 7F E3 94 4A 17 F3 07 A7 8B 4D 2B 30 C5"
00: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
01: D6 AA 74 FD D2 AF 72 FA DA A6 78 F1 D6 AB 76 FE
02: B6 92 CF 0B 64 3D BD F1 BE 9B C5 00 68 30 B3 FE
03: B6 FF 74 4E D2 C2 C9 BF 6C 59 0C BF 04 69 BF 41
04: 47 F7 F7 BC 95 35 3E 03 F9 6C 32 BC FD 05 8D FD
05: 3C AA A3 E8 A9 9F 9D EB 50 F3 AF 57 AD F6 22 AA
06: 5E 39 0F 7D F7 A6 92 96 A7 55 3D C1 0A A3 1F 6B
07: 14 F9 70 1A E3 5F E2 8C 44 0A DF 4D 4E A9 C0 26
08: 47 43 87 35 A4 1C 65 B9 E0 16 BA F4 AE BF 7A D2
09: 54 99 32 D1 F0 85 57 68 10 93 ED 9C BE 2C 97 4E
10: 13 11 1D 7F E3 94 4A 17 F3 07 A7 8B 4D 2B 30 C5
```