Appendix H

Trace set coding

Side-Channel Analysis traces include supplementary information that may be stored along with the samples. Inspector supports three formats for the storage of this information:

1. Sequence of bytes (*.bytes); this is the default coding applied to unknown extensions. Each byte in the file represents one sample.
2. Sequence of floats (*.floats); each group of 4 bytes represents one little endian (LSB first) coded sample in float coding (IEEE 754).
3. A structured but flexible format (*.trs); the file starts with a header, followed by a trace block containing a set of traces.

The '.trs' format implements a proprietary encoding in which each object value is preceded by a tag field and optionally a length field (i.e. TLV format). This section explains the coding of this format.

Sequences of recorded traces are stored in a so-called trace set file. Each trace includes the following information:

- The recorded samples
- Optionally, recorded (cryptographic) data related to the analysis process
- Optionally, a local title that represents the meaning of the trace

The global data includes:

- Number of traces
- Number of samples per trace
- Length of cryptographic data included in trace
- Sample coding (e.g. type and length in bytes of each sample)
- Title space reserved for local title in each trace
- Global title representing a general name for each trace
- Description of the trace set
- Offset in X-axis (x value of first sample)
- Label of X-axis (unit, e.g. 'seconds')
- Label of Y-axis (unit, e.g. 'volt')
- Scale value for X-axis
- Scale value for Y-axis
- Preferred representation, linear or logarithmic

Design considerations

Two important requirements are incorporated in the design of this encoding format:

- *Flexibility:* The global information shall be coded in a flexible way, such that new types of information can be added without redefining the trace set file coding. This ensures that existing trace set files can still be read when new global data is added.
- *Performance:* The trace related data shall be randomly accessible to allow quick reading and browsing through the trace set.

The first requirement is met by using a TLV (Tag, Length, Value) structure for the file header. Each object is identified by a tag. Its content is preceded by a length field. Applications can simply generate / process the supported objects and ignore others. The second requirement is met by using a flat structure for all traces containing fixed spaces for titles, data and samples. This allows applications to access individual traces at random in a very fast manner.

Header coding

The trace set file header defines the following objects:

Tag	Name	Mandatory /Optional	Type	Length	Default	Meaning
0x41	NT	M	int	4		Number of traces
0x42	NS	M	int	4		Number of samples per trace
0x43	SC	M	byte	1		Sample Coding (see table)
0x44	DS	O	short	2	0	Length of cryptographic data included in trace
0x45	TS	O	byte	1	0	Title space reserved per trace
0x46	GT	O	byte[]	variable	“trace”	Global trace title
0x47	DC	O	byte[]	variable	None	Description
0x48	XO	O	int	4	0	Offset in X-axis for trace representation
0x49	XL	O	byte[]	variable	None	Label of X-axis
0x4A	YL	O	byte[]	variable	None	Label of Y-axis
0x4B	XS	O	float	4	1	Scale value for X-axis
0x4C	YS	O	float	4	1	Scale value for Y-axis
0x4D	TO	O	int	4	0	Trace offset for displaying trace numbers
0x4E	LS	O	byte	1	0	Logarithmic scale
0x4F-0x5E						Reserved for Future Use
0x5F	TB	M	none	0		Trace block marker: an empty TLV that marks the end of the header

Table H.1: Trace set objects in the trace set header

The object coding always starts with the tag byte. The object length is coded in one or more bytes. If bit 8 (msb) is set to ‘0’, the remaining 7 bits indicate the length of the object. If bit 8 is set to ‘1’, the remaining 7 bits indicate the number of additional bytes in the length field. These additional bytes define the length in little endian coding (LSB first). The content of the object is stored in the subsequent number of bytes, indicated by length. A trace set file contains at least the mandatory objects and may contain any of the optional fields. The TB object is always the last object and marks the end of the header. The value of the numeric objects is coded little endian (LSB first). The float values use the IEEE 754 coding which is commonly supported by modern programming languages. Object SC defines the sample coding:

Figure 5.1 shows the coding of a trace set file:

bit 8-6	**reserved, set to '000'**
bit 5	integer (0) or floating point (1)
bit 4-1	Sample length in bytes (valid values are 1, 2, 4)

Table H.2: Sample coding

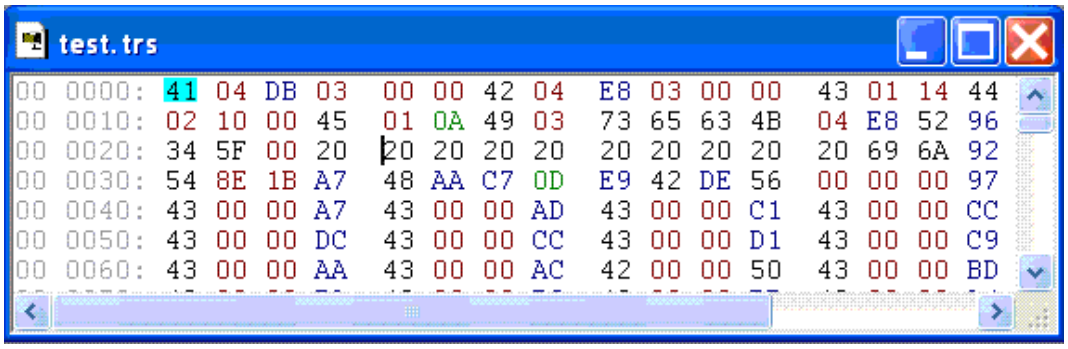

Figure H.1: Trace set example

The test.trs file contains the following objects:

0x41 (NT), length: 4, value: 0x3DB (987) number of traces

0x42 (NS), length: 4, value: 0x3E8 (1000) number of samples

0x43 (SC), length: 1, value: 0x14 (20), float coding, sampleSpace: 4 bytes per sample

0x44 (DS), length: 2, value: 0x10 (16), Data space: number of data bytes included

0x45 (TS), length: 1, value: 0x0A (10), 10 bytes title space per trace

0x49 (XL), length: 3, value: "sec", label X-axis

0x4B (XS), length: 4, value: 0x349652E8 (280E-9), time base of 280ns per sample

0x5F (TB), length: 0, beginning of trace block

987 times: 10 bytes space (title not present) 16 bytes data (e.g. 0x69 0x6A .. 0x56 0x00) 4000 bytes containing 1000 float samples of 4 bytes (e.g. 0x43970000 = 302)

Note that the header length is flexible, but always ends with the Trace Block Marker: 0x5F00.