Semi-Automated Prototyping of a TPM v2 Software and Hardware Simulation Platform

Martin Pirker, Johannes Winter

{mpirker,jwinter}@iaik.tugraz.at

Institute for Applied Information Processing and Communications (IAIK),
Graz University of Technology, Austria

Trusted Platform Module

- Feb'02 TPM v1.1b specification
- Oct'03 TPM v1.2 rev. 62
- Mar'11 TPM v1.2 rev. 116

Trusted Software Stack

- 2 full open-source implementations of TSS specs

- TrouSerS / C
  - http://trousers.sourceforge.net/

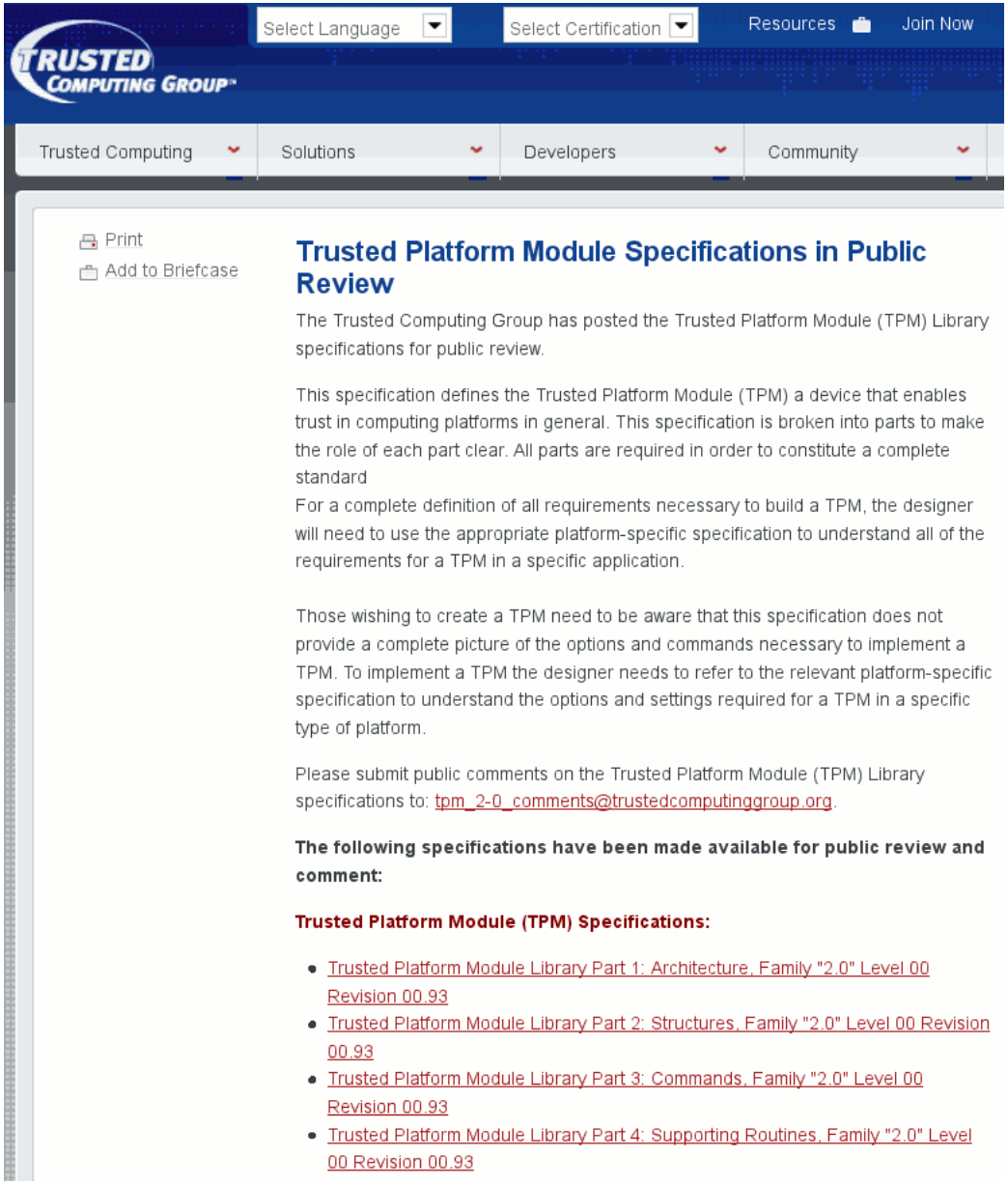
- jTSS / Java
  - http://trustedjava.sourceforge.net/

jTSS

- Developed / started by IAIK within EU OpenTC project
- 100% Java implementation of every TSS detail
- Manual work
- Tried automation of implementation process
  ...but gave up on it

Oct'12

TPM v2 first public draft



v2 Specification Promise

"The information in this document is formatted so that it may be converted to standard computer-language formats by an automated process. The purpose of this automated process is to minimize the transcription errors that often occur during the conversion process […]
In addition, the conventions and notations in this clause describe the representation of various data so that it is both human readable and amenable to automated processing."

Specification Text Extraction

- *.PDF files, ~1400 pages TPM v2 spec

  … run through Libreoffice PDF import ...

- → *.FODG files OpenDocument Graphics (XML based)

  … run through custom script ...

- → raw text fragments
  [x,y,style,text], [x,y,style,text], ...

```
---
- - 2.54
  - 14.519
  - T5
  - The information in this document is formatted so that it may be converted to standard
    computer-
---
- - 2.54
  - 14.942
  - T5
  - ! 'language formats '
- - 5.822
  - 14.942
  - T5
  - ! 'by '
- - 6.448
  - 14.942
  - T5
  - ! 'an automated process. '
- - 10.778
  - 14.942
  - T5
  - ! 'The purpose '
- - 13.245
  - 14.942
  - T5
  - ! 'of this '
- - 14.649
  - 14.942
  - T5
  - ! 'automated process '
```

4.1 Introduction

The information in this document is formatted so that it may be converted to standard language formats by an automated process. The purpose of this automated pro minimize the transcription errors that often occur during the conversion process.

Spec Parts

- Part 2
  Data structures
- Part 3
  Commands
- Part 4
  Support / Runtime environment

```
- :title: ! 'Table 17 –Definition of (UINT16) TPM_EO Constants <IN/OUT> '
  :class: :named_constant
  :base: UINT16
  :name: TPM_EO
  :in: true
  :out: true
  :body:
  - - TPM_EO_EQ
    - '0x0000'
  - - TPM_EO_NEQ
    - '0x0001'
  - - TPM_EO_SIGNED_GT
    - '0x0002'
  - - TPM_EO_UNSIGNED_GT
    - '0x0003'
  - - TPM_EO_SIGNED_LT
    - '0x0004'
  - - TPM_EO_UNSIGNED_LT
    - '0x0005'
  - - TPM_EO_SIGNED_GE
    - '0x0006'
  - - TPM_EO_UNSIGNED_GE
    - '0x0007'
  - - TPM_EO_SIGNED_LE
    - '0x0008'
  - - TPM_EO_UNSIGNED_LE
    - '0x0009'
  - - TPM_EO_BITSET
    - '0x000A'
  - - TPM_EO_BITCLEAR
    - '0x000B'
  - - ! '#TPM_RC_VALUE'
    - ''
```

Table 17 — Definition of (UINT16) TPM_EO Constants <IN/OUT>

Operation Name	Value	Comments
TPM_EO_EQ	0x0000	A = B
TPM_EO_NEQ	0x0001	A ≠ B
TPM_EO_SIGNED_GT	0x0002	A > B signed
TPM_EO_UNSIGNED_GT	0x0003	A > B unsigned
TPM_EO_SIGNED_LT	0x0004	A < B signed
TPM_EO_UNSIGNED_LT	0x0005	A < B unsigned
TPM_EO_SIGNED_GE	0x0006	A ≥ B signed
TPM_EO_UNSIGNED_GE	0x0007	A ≥ B unsigned
TPM_EO_SIGNED_LE	0x0008	A ≤ B signed
TPM_EO_UNSIGNED_LE	0x0009	A ≤ B unsigned
TPM_EO_BITSET	0x000A	All bits SET in B are SET in A. ((A&B)=B)
TPM_EO_BITCLEAR	0x000B	All bits SET in B are CLEAR in A. ((A&B)=(
#TPM_RC_VALUE		Error code returned when unmarshaling of

Table 79 — Definition of TPMS_PCR_SELECT Structure

Parameter	Type	Description
sizeofSelect {PCR_SELECT_MIN:}	UINT8	the size in octets of the *pcrSelect* array
pcrSelect [sizeofSelect] {:PCR_SELECT_MAX}	BYTE	the bit map of selected PCR
#TPM_RC_VALUE		

```
// Table 79 –Definition of TPMS_PCR_SELECT Structure
typedef struct {
    UINT8   sizeofSelect;
    BYTE    pcrSelect[PCR_SELECT_MAX];
} TPMS_PCR_SELECT;
```

```
// Table 79 –Definition of TPMS_PCR_SELECT Structure
TPM_RC
TPMS_PCR_SELECT_Unmarshal(TPMS_PCR_SELECT *target, BYTE **buffer, INT32 *size)
{
    TPM_RC  result;
    result = UINT8_Unmarshal((UINT8 *)&(target->sizeofSelect), buffer, size);
    if (result != TPM_RC_SUCCESS) {
        return result;
    }
    if ((target->sizeofSelect) < PCR_SELECT_MIN) {
        return TPM_RC_VALUE; }
    if ((target->sizeofSelect) > PCR_SELECT_MAX) {
        return TPM_RC_VALUE; }
    result = BYTE_Array_Unmarshal((BYTE *)&(target->pcrSelect), buffer, size, (INT32)(target->sizeofSelect);
    if (result != TPM_RC_SUCCESS) {
        return result;
    }
    return TPM_RC_SUCCESS;
}

UINT16
TPMS_PCR_SELECT_Marshal(TPMS_PCR_SELECT *source, BYTE **buffer, INT32 *size) {
    UINT16  result = 0;
    result = (UINT16)(result + UINT8_Marshal((UINT8 *)&(source->sizeofSelect), buffer, size));
    result = (UINT16)(result + BYTE_Array_Marshal((BYTE *)&(source->pcrSelect), buffer, size, (INT32)(source->sizeofSelect)));

    return result;
}
```

24.4.2 Command and Response

Table 99 — TPM2_PCR_Read Command

Type	Name	Description
TPMI_ST_COMMAND_TAG	tag	
UINT32	commandSize	
TPM_CC	commandCode	TPM_CC_PCR_Read
TPML_PCR_SELECTION	pcrSelectionIn	The selection of PCR to read

Table 100 — TPM2_PCR_Read Response

Type	Name	Description
TPM_ST	tag	see clause 8
UINT32	responseSize	
TPM_RC	responseCode	
UINT32	pcrUpdateCounter	the current value of the PCR update counter
TPML_PCR_SELECTION	pcrSelectionOut	the PCR in the returned list
TPML_DIGEST	pcrValues	the contents of the PCR indicated in *pcrSelect* as tagged digests

24.4.3 Detailed Actions

```
1   #include "InternalRoutines.h"
2   #include "PCR_Read_fp.h"
3   TPM_RC
4   TPM2_PCR_Read(
5       PCR_Read_In      *in,                  // IN: input parameter list
6       PCR_Read_Out     *out                  // OUT: output parameter list
7   )
8   {
9   // Command Output
10
11      // Call PCR read function.  input pcrSelectionIn parameter could be changed
12      // to reflect the actual PCR being returned
13      PCRRead(&in->pcrSelectionIn, &out->pcrValues, &out->pcrUpdateCounter);
14
15      out->pcrSelectionOut = in->pcrSelectionIn;
16
17      return TPM_RC_SUCCESS;
18  }
```

```
//
// Do no not edit manually - changes will be destroyed on next run!
// Created  2013-06-11 14:00:06 +0200  by automated extraction script from
//  "TPM Rev 2.0 Part 3 - Commands 00.93p 121009-code.pdf"
//

#ifndef _PCR_READ_H
#define _PCR_READ_H

typedef struct {
    TPML_PCR_SELECTION                  pcrSelectionIn;
} PCR_Read_In;

typedef struct {
    UINT32                              pcrUpdateCounter;
    TPML_PCR_SELECTION                  pcrSelectionOut;
    TPML_DIGEST                         pcrValues;
} PCR_Read_Out;

#define RC_PCR_Read_pcrSelectionIn        (TPM_RC_P + TPM_RC_1)

TPM_RC
TPM2_PCR_Read(
PCR_Read_In *in, // IN: input parameter list
PCR_Read_Out *out // OUT: output parameter list
);

#endif
```

```
#include "InternalRoutines.h"
#include "PCR_Read_fp.h"
TPM_RC
TPM2_PCR_Read(PCR_Read_In * in, // IN: input parameter list
              PCR_Read_Out * out        // OUT: output parameter list
  )
{
// Command Output

// Call PCR read function. input pcrSelectionIn parameter could be changed
// to reflect the actual PCR being returned
    PCRRead(&in->pcrSelectionIn, &out->pcrValues, &out->pcrUpdateCounter);

    out->pcrSelectionOut = in->pcrSelectionIn;

    return TPM_RC_SUCCESS;
}
```

Script Run

```
processing 221 tables...
...skipping... Table 1 -Name Prefix Convention
...skipping... Table 2 -Unmarshaling Errors
...already in Impl.h - no type def created...  Table 7 -Definition
...already in Impl.h - no type def created...  Table 8 -Definition
...skipping... Table 9 -Description of TPM Command Format Fields
...skipping... Table 10 -Legend for Command Code Tables
...skipping... Table 12 -Format-Zero Error Codes
...skipping... Table 13 -Format-One Error Codes
...skipping... Table 14 -Error Code Groupings
...skipping... Table 81 -Values for proof Used in Tickets
...skipping... Table 82 -General Format of a Ticket
...skipping... Table 193 -Options for space Field of TPM_NV
...skipping... Table 201 -Context Handle Values
writing ./gen/include/TPM_Types.h
writing ./gen/include/BaseTypes.h
writing ./gen/include/OpenBaseTypes.h
writing ./gen/marshal.c
```

```
writing ./gen/cmd/NV_WriteLock.c
writing ./gen/cmdinclude/NV_WriteLock_fp.h
writing ./gen/cmd/NV_GlobalWriteLock.c
writing ./gen/cmdinclude/NV_GlobalWriteLock_fp.h
writing ./gen/cmd/NV_Read.c
writing ./gen/cmdinclude/NV_Read_fp.h
writing ./gen/cmd/NV_ReadLock.c
writing ./gen/cmdinclude/NV_ReadLock_fp.h
writing ./gen/cmd/NV_ChangeAuth.c
writing ./gen/cmdinclude/NV_ChangeAuth_fp.h
writing ./gen/cmd/NV_Certify.c
writing ./gen/cmdinclude/NV_Certify_fp.h

creating ./gen/include/Commands.h
6318 lines raw C source extracted from spec
```

```
writing ./gen/Platfo
writing ./gen/PowerP
writing ./gen/PPPlat
writing ./gen/TpmFai
writing ./gen/includ
writing ./gen/TcpSer
writing ./gen/TPMCmd
writing ./gen/TPMCmds.c

22350 lines raw C source extracted from spec

postprocessing...
writing ./gen/include/Attest_spt_fp.h
writing ./gen/include/Context_spt_fp.h
writing ./gen/include/NV_spt_fp.h
writing ./gen/include/Object_spt_fp.h
writing ./gen/include/Policy_spt_fp.h
writing ./gen/include/AlgorithmCap_fp.h
writing ./gen/include/Bits_fp.h
writing ./gen/include/CommandAudit_fp.h
writing ./gen/include/Commands_fp.h
writing ./gen/include/CpriDataEcc_fp.h
writing ./gen/include/CpriECC_fp.h
writing ./gen/include/CpriHash_fp.h
```

```
===== 8 Subsystem
CommandAudit.c
  174 src lines extracted
DA.c
  131 src lines extracted
Hierarchy.c
  165 src lines extracted
NV.c
  1528 src lines extracted
Object.c
  678 src lines extracted
PCR.c
  977 src lines extracted
PP.c
  123 src lines extracted
Session.c
  744 src lines extracted
Time.c
  192 src lines extracted
```

Table 87 — TPM2_GetTime Command

Type	Name	Description
TPMI_ST_COMMAND_TAG	tag	
UINT32	commandSize	
TPM_CC	commandCode	TPM_CC_GetTime
TPMI_RH_ENDORSEMENT	@privacyAdminHandle	handle of the privacy administrator (TPM_RH_ENDORSEMENT) Auth Index: 1 Auth Role: USER
TPMI_DH_OBJECT+	@signHandle	the *keyHandle* identifier of a loaded key that can perform digital signatures Auth Index: 2 Auth Role: USER
TPM2B_DATA	qualifyingData	data to tick stamp
TPMT_SIG_SCHEME+	inScheme	signing scheme to use if the *scheme* for *signHandle* is TPM_ALG_NULL

Table 88 — TPM2_GetTime Response

Type	Name	Description
TPM_ST	tag	see clause 8
UINT32	responseSize	
TPM_RC	responseCode	.
TPM2B_ATTEST	timeInfo	standard TPM-generated attestation block
TPMT_SIGNATURE	signature	the signature over *timeInfo*

Table 217 — TPM2_NV_Certify Command

Type	Name	Description
TPMI_ST_COMMAND_TAG	tag	
UINT32	commandSize	
TPM_CC	commandCode	TPM_CC_NV_Certify
TPMI_DH_OBJECT+	@signHandle	handle of the key used to sign the attestation structure Auth Index: 1 Auth Role: USER
TPMI_RH_NV_AUTH	@authHandle	handle indicating the source of the authorization value for the Index Auth Index: 2 Auth Role: USER
TPMI_RH_NV_INDEX	nvIndex	Index for the area to be certified Auth Index: None
TPM2B_DATA	qualifyingData	user-provided qualifying data
TPMT_SIG_SCHEME+	inScheme	signing scheme to use if the *scheme* for *signHandle* is TPM_ALG_NULL
UINT16	size	number of octets to certify
UINT16	offset	octet offset into the area This value shall be less than or equal to the size of the *nvIndex* data.

TPM2_Commit Command

Type	Name	Description
TPMI_ST_COMMAND_TAG	tag	
UINT32	paramSize	
TPM_CC	commandCode	TPM_CC_Commit
TPMI_DH_OBJECT	@keyHandle	handle of the key that will be used for anonymous signing using the commit values or an unrestricted decryption key Auth Index: 1 Auth Role: USER
TPM2B_ECC_POINT	P1	a point (M) on the curve used by *signHandle* or an Empty Point
TPM2B_SENSITIVE_DATA	s2	octet array used to derive x-coordinate of a base point or an Empty Buffer
TPM2B_ECC_PARAMETER	y2	y coordinate of the point associated with *s2* or an Empty Buffer is *S2* is an Empty Buffer

TPM2_Commit Response

Type	Name	Description
TPM_ST	tag	see clause 8
UINT32	paramSize	
TPM_RC	responseCode	
TPM2B_ECC_POINT	K	ECC point $K := [d_s](x2, y2)$ or an Empty Point
TPM2B_ECC_POINT	L	ECC point $L := [r](x2, y2)$ or an Empty Point
TPM2B_ECC_POINT	E	ECC point $E := [r]P1$ or an Empty Point
UINT16	counter	least-significant 16 bits of *commitCount*

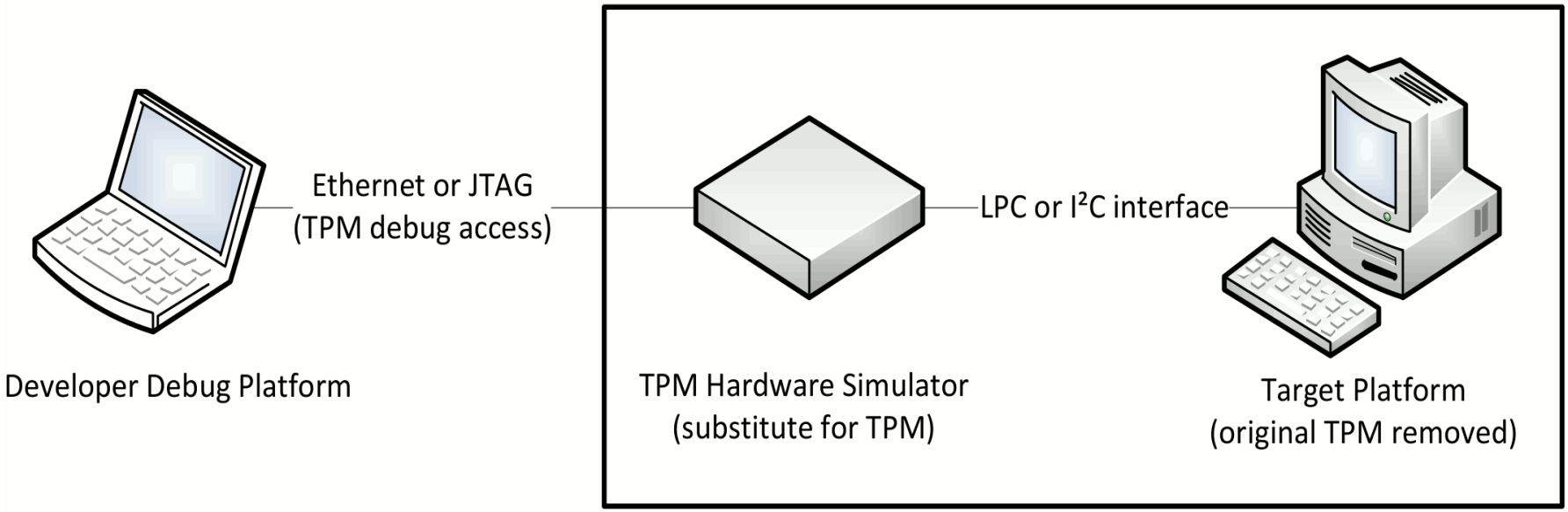
Towards a TPM v2 Simulator

- write Makefile / OpenSSL inclusion
- remove winsock.h / windows.h references
  - socket interface instead of RPC
- remove MS-stuff, e.g. fopen_s
- CFLAGS += -std=C99 -pedantic
- case sensitive header includes (e.g. Tpm.h vs tpm.h)
- duplicate / inconsistent s_NvIsAvailable declaration
- startup / init / self-test code
- … … ...

  → and obtain a Linux build :-)

Hardware TPM v2

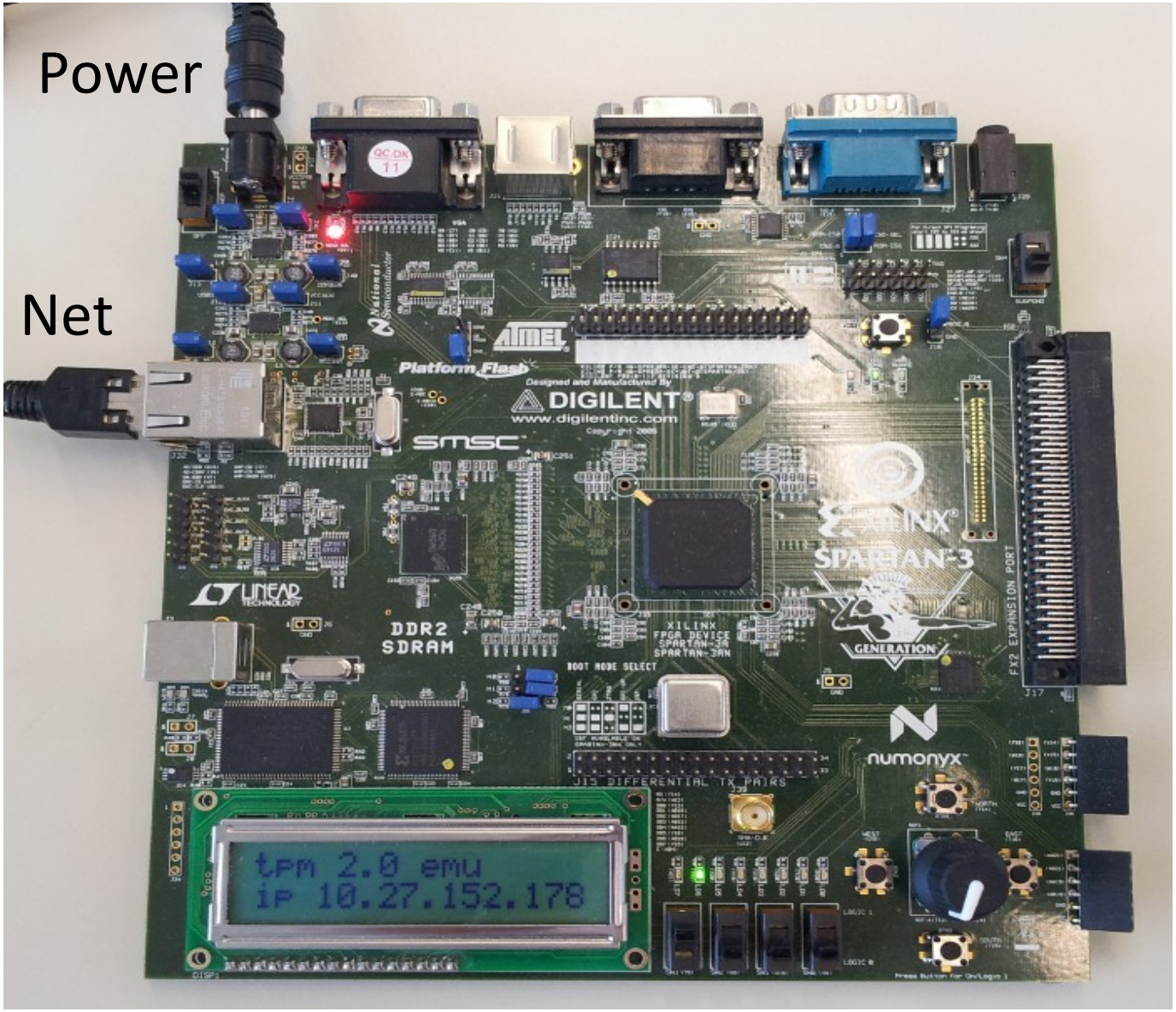
- TPM v2 in software is nice.... and as hardware?
- Idea: run SW Simulator on FGPA platform

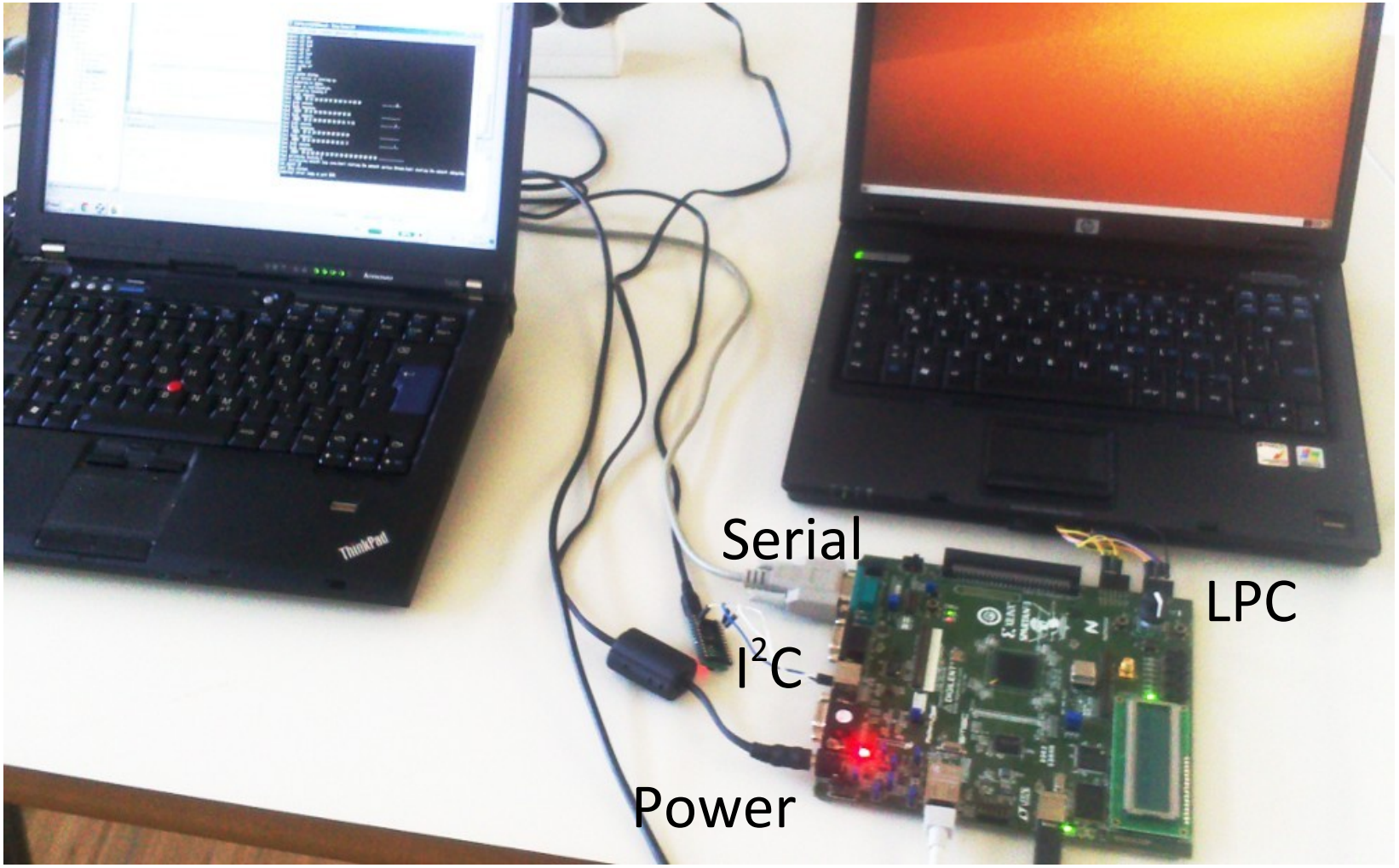


Hardware TPM v2

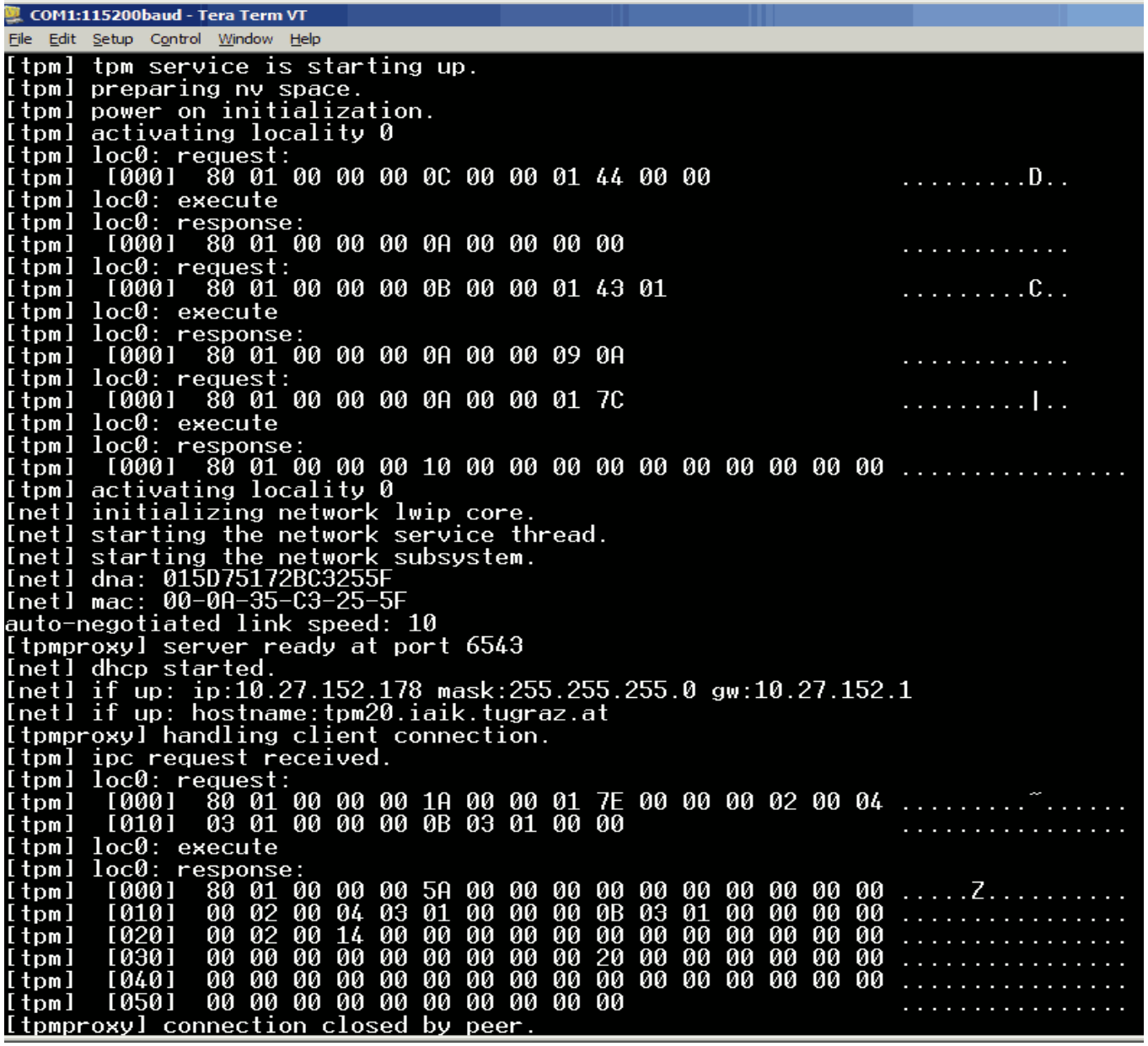
- Xilinx XC3S700ANN FGPA board
- 32-bit Xilinx MicroBlaze soft-core processor
- I^2C+LPC bus slave controllers
- open-source lwIP TCP/IP stack
- stripped OpenSSL cryptography

.....fits just in on-chip flash (8Mbit) of FPGA
(FPGA bitstream + bootloader + TPM simulator code)


Power
Net
tpm 2.0 emu
ip 10.27.152.178
DIGILENT
www.digilentinc.com
DDR2
SDRAM
SPARTAN-3
GENERATION
numonyx


Serial
LPC
I²C
Power
Net
USB/JTAG

Serial Console



“Hello World”

```
E:\MiscJava\HelloTpm20>java -cp libs/libjtpm20.jar;bin/ at.iaik.tc.tpm20.HelloTpm20 10.27.152.178 6543
--------------------------------------------------------------------------------
PCR_Read[PCR=0] RQU:  80 01 00 00 00 1A 00 00 01 7E 00 00 00 02 00 04 03 01 00 00 00 0B 03 01 00 00 00 00 00 00 00 00
PCR_Read[PCR=0] RSP:  80 01 00 00 00 5A 00 00 00 00 00 00 00 00 00 00 00 02 00 04 03 01 00 00 00 0B 03 01 00 00 00 00
PCR_Read[PCR=0] RSP:  00 02 00 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 00 00 00 00
PCR_Read[PCR=0] RSP:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
--------------------------------------------------------------------------------
PCR_Read[PCR=1] RQU:  80 01 00 00 00 1A 00 00 01 7E 00 00 00 02 00 04 03 02 00 00 00 0B 03 02 00 00 00 00 00 00 00 00
PCR_Read[PCR=1] RSP:  80 01 00 00 00 5A 00 00 00 00 00 00 00 00 00 00 00 02 00 04 03 02 00 00 00 0B 03 02 00 00 00 00
PCR_Read[PCR=1] RSP:  00 02 00 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 00 00 00 00
PCR_Read[PCR=1] RSP:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
--------------------------------------------------------------------------------
PCR_Read[PCR=2] RQU:  80 01 00 00 00 1A 00 00 01 7E 00 00 00 02 00 04 03 04 00 00 00 0B 03 04 00 00 00 00 00 00 00 00
PCR_Read[PCR=2] RSP:  80 01 00 00 00 5A 00 00 00 00 00 00 00 00 00 00 00 02 00 04 03 04 00 00 00 0B 03 04 00 00 00 00
PCR_Read[PCR=2] RSP:  00 02 00 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 00 00 00 00
PCR_Read[PCR=2] RSP:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
--------------------------------------------------------------------------------
PCR_Read[PCR=3] RQU:  80 01 00 00 00 1A 00 00 01 7E 00 00 00 02 00 04 03 08 00 00 00 0B 03 08 00 00 00 00 00 00 00 00
PCR_Read[PCR=3] RSP:  80 01 00 00 00 5A 00 00 00 00 00 00 00 00 00 00 00 02 00 04 03 08 00 00 00 0B 03 08 00 00 00 00
PCR_Read[PCR=3] RSP:  00 02 00 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 00 00 00 00
PCR_Read[PCR=3] RSP:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
--------------------------------------------------------------------------------
PCR_Read[PCR=4] RQU:  80 01 00 00 00 1A 00 00 01 7E 00 00 00 02 00 04 03 10 00 00 00 0B 03 10 00 00 00 00 00 00 00 00
PCR_Read[PCR=4] RSP:  80 01 00 00 00 5A 00 00 00 00 00 00 00 00 00 00 00 02 00 04 03 10 00 00 00 0B 03 10 00 00 00 00
PCR_Read[PCR=4] RSP:  00 02 00 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 00 00 00 00
PCR_Read[PCR=4] RSP:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

TPM v2 in-system Simulation

- Done
  - JTAG debugging of FPGA TPM
  - I^2C interface for embedded (e.g. for Arduino)
  - Network interface (similar to IBM SW TPM v1.2)
- Ongoing
  - FPGA LPC interface based on previous work, FPGA side handling of TIS protocol work-in-progress

Lessons Learned

- Is the TPM v2 spec suited for automated processing?
  Yes, better than v1.2
- Is it possible to synthesize a TPM simulator from it?
  Yes, but quite some work to create/generate missing code
- Can we use a SW simulator to fake a HW TPM v2?
  Port to FPGA board, work-in-progress...
- Outlook
  - Cleanup, debug, document… someone with a TSS v2... ? :-)

Credits

- Martin Pirker (mpirker@iaik.tugraz.at)
  - Spec parser, extractor, code generation
- Johannes Winter (jwinter@iaik.tugraz.at)
  - FPGA port
- Paper:
  Proceedings of 6th International Conference on Trust & Trustworthy Computing (TRUST 2013), 17-19 Jun 2013, London, UK; LNCS 7904, Springer
  http://trust2013.sba-research.org/

INTRUST'13 conference

5/6 Dec 2013 – Graz, Austria

CFP: 8.Jul !

intrust13.iaik.tugraz.at