NOKIA Bell Labs

# Usability and Security of Trusted Platform Module (TPM) Library APIs

Sid Rao[1], Gabriela Limonta[1] and Janne Lindqvist[2]

[1]Aalto University, [2]Nokia Bell Labs


August 7-9, 2022

Boston MA, USA + Virtual

# Trusted Platform Module (TPM)

- Tamper-proof chip
- Unique identity
- Secure storage and operations

- Applications:
  - Boot Security (e.g., UEFI and Google Chromebooks)
  - Disk encryption (e.g., BitLocker, LUKS)
  - Trust and attestation for Cloud, Edge and IoT (e.g., Keylime)
  - VPNs, SSH, SSL or any other applications where keys are needed
  - Recommended by standards/guidelines for NFV and server security

# About

- TPM is an old and widely used technology
- **Motivation**: **TPM not a go-to choice of software developers. Why?**
  - ❑ TPM concepts are complex? And security is even more complex?
  - ❑ Software developers find it hard to realize TPM's potential?
  - ❑ Lacks supporting ecosystem for developers?
  - ❑ All the above?
- **Scope**: TPM library APIs (i.e., standardized high-level APIs for software applications to talk to the TPM chip)
  - **tpm2-tools**, IBMTSS, Microsoft TSS, go-tpm, wolfTPM
- **Goals:**
  - Understand the usability and security pitfalls of TPM developers
  - Review TPM library API implementations
  - Provide concrete design guidelines for usably secure API development

# TPM in a nutshell

## Cryptographic and non-cryptographic security features

- Key creation
- Encryption
- Signing
- Hashing

**Cryptographic capabilities**

- NVRAM
- Platform Configuration Registers
- Sealed data blobs

**Secure storage**

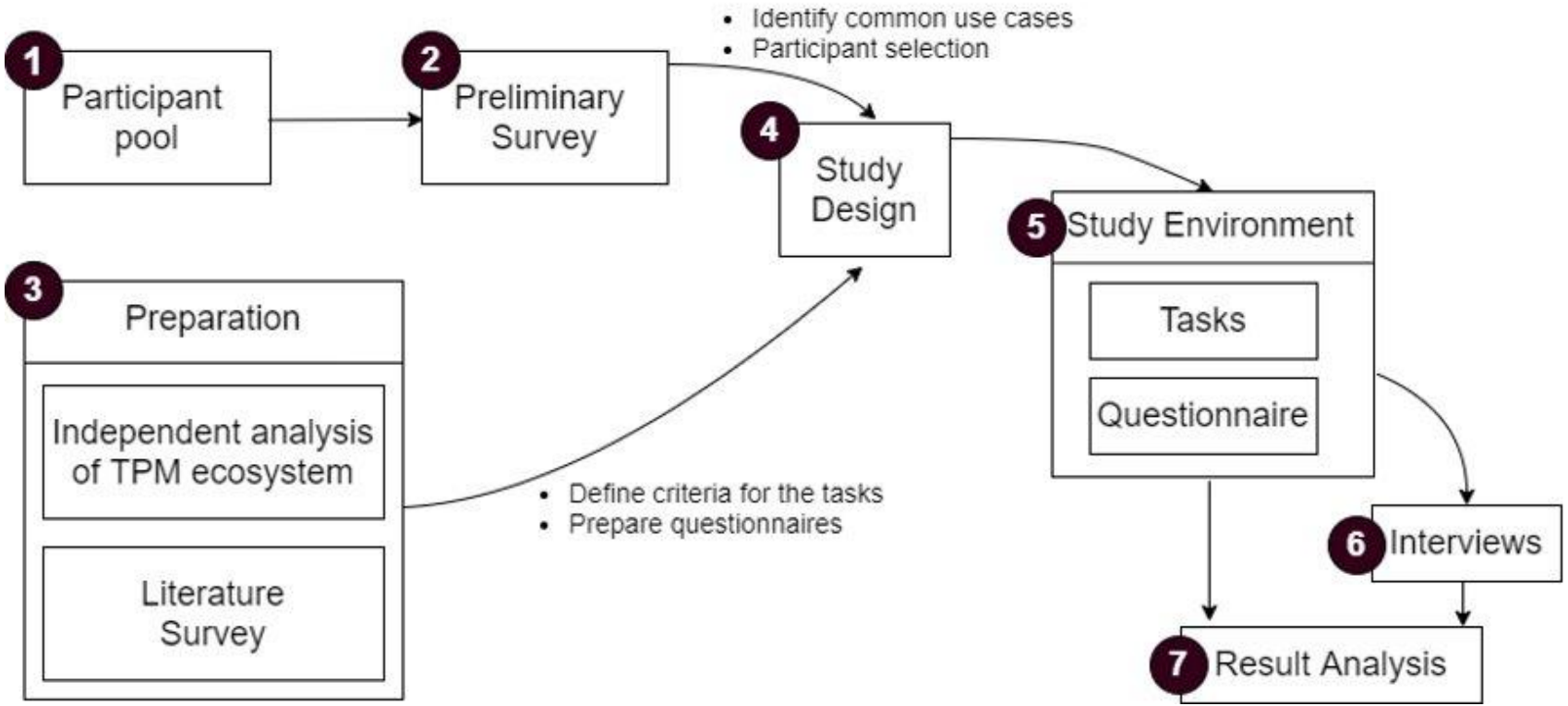
- Key attributes
- TPM-internal states
- TPM-external states
- Authorizations and sessions

**Restrictions and policies**

- Quoting
- Unique identity

**Attestation**

# Study overview



# Study design

## Participants

**Target**: TPM developers with security background

**Participation count**:

- Preliminary survey: 48
- Interested in the study: 36
- Completed the study: 13
- Interviewed: 9

**Limitations**:

- Small number of participants
- Only tpm2-tools library was used

## Task design

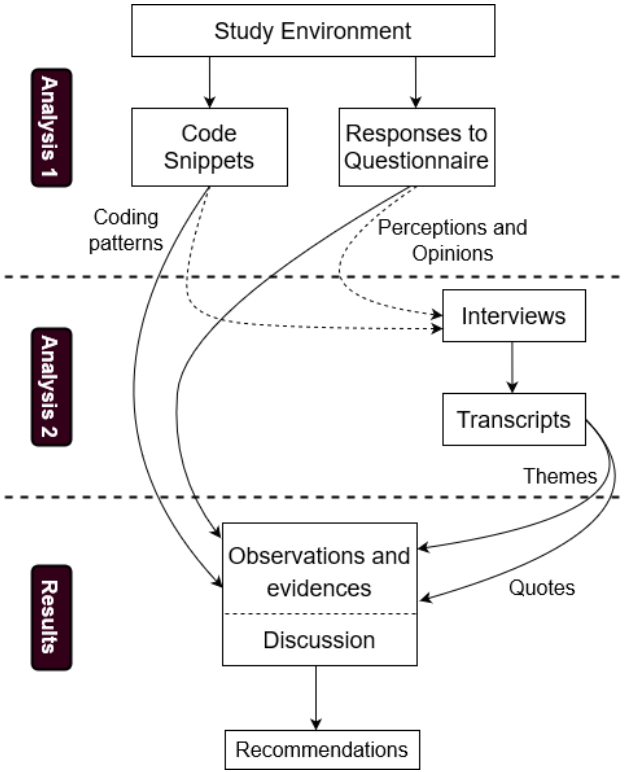
- 4 tasks for evaluating functional correctness and security choices
  - Encryption: either asymmetric or symmetric
  - Storing measurements
  - Securing secrets
  - Remote attestation
- Combination of cryptographic and non-cryptographic features

## Questionnaire design

For evaluating perceptions and opinions

- **At the beginning**: basic demographics
- **After each task**:
  - Familiarity and complexity
  - Security and correctness
  - Reasons for not completing
  - Usefulness of error messages
- **At the end**:
  - Usual choice of supporting materials
  - Reasons for referring to external materials

# Analysis outline




**Analysis Phase 1:**

- Data from the study environment
  - Executed code snippets → prompts
  - Questionnaire responses → probes

**Analysis Phase 2:**

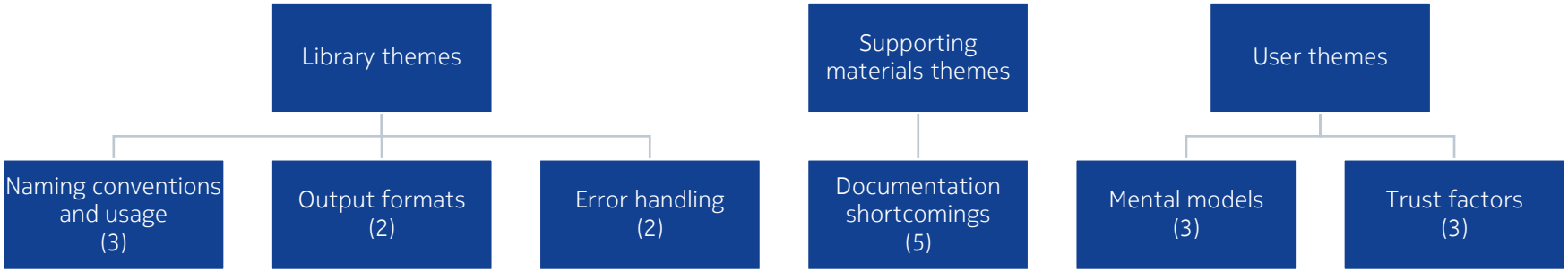
- Interviews transcripts → Thematic analysis

**Results:**

- Themes of usability pitfalls
- Common coding patterns (i.e., developer habits or mistakes)

# Results

## Thematic analysis: 18 themes identified




## Common coding patterns that affected the security of the participants' code

- Reliance on default values
- Oversights when specifying cryptographic and TPM-specific attributes
- Failure to consider threat models

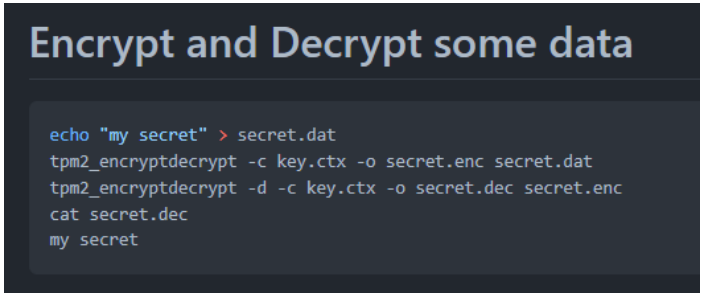
# Example

## Library themes → Error handling → Lack of pointers to resolve

```
$ tpm2_encryptdecrypt -p Gnampf -c task_files/parent.context task_files/file2.txt \
-o task_files/file2.encrypted

WARN: Using a weak IV, try specifying an IV
```

"I find it's kind of destructive criticism when the program just tells me "well, you used the wrong initialization vector", but doesn't make any comments on how to do it better."




tpm2-tools/tpm2_encryptdecrypt.1.md at 5.0 · tpm2-software/tpm2-tools (github.com)

# Recommendations

**For library documentation:**

1. Include background information about TPM concepts
2. Provide code snippets for common use cases
3. Improve entry-level documentation
4. Include guidelines for picking security attributes
5. Fix incoherent aspects

**For library software:**

1. Provide developer-friendly error messages
2. Provide concise output messages
3. Utilize abstractions (e.g., for sequential command execution)
4. Promote secure crypto primitives

# Summary of contributions and results

- **Open-source study platform for TPM-related tasks**
  - Nothing to install and configure --> Works right out of a browser
  - It can be used for hands-on tutorials, hackathons or future studies involving TPMs

- **Qualitative results about the tpm2-tools library**
  - Identified 18 usability and security pitfalls
  - Complex topics + lack of developer-friendly APIs and supporting materials. Developers
    - struggle to use the APIs efficiently
    - are prone to make trivial mistakes that nevertheless undermine security
    - cannot fully utilize TPM's capabilities, and it also discourages newbies
  - Concrete recommendations for the TPM library to immediately address the issues identified

- **Need for usability by design**
  - Usability and security pitfalls in software can be traced back to standard specifications
  - HCI experts should be involved already in the design of specifications

# Thank you!

**Resource materials:**

- **Full paper:** https://www.usenix.org/conference/soups2022/presentation/rao
- **TPM study environment:** https://github.com/nokia/tpm-study-environment

**Contact:**


- **Sid Rao** (sid.rao@nokia-bell-labs.com)
- **Gabriela Limonta** (gabriela.limonta@nokia-bell-labs.com)

# Task and security features mapping

| Task | | Security features | |
|---|---|---|---|
| | | Crypto | Non-crypto |
| Encryption | Asymmetric | C2 | NC1, NC2 |
| | Symmetric | C1 | NC4, NC5 |
| Storing measurements | | C4 | NC6 |
| Securing secrets | | - | NC1, NC3 |
| Remote attestation | | C3 | NC1, NC2, NC6 |

| Cryptographic security features | |
|---|---|
| C1 | Symmetric -> Encryption |
| C2 | Asymmetric -> Encryption |
| C3 | Asymmetric -> Signing |
| C4 | Hashing |
| **Non-cryptographic security features** | |
| NC1 | Use of the TPM hierarchies |
| NC2 | TPM key restrictions |
| NC3 | Restrictions against TPM-internal states |
| NC4 | Restrictions against TPM-external states |
| NC5 | Session-based command or object authorization |
| NC6 | PCR usage |

NOKIA Bell Labs