Abstractions and tooling for leakage evaluation

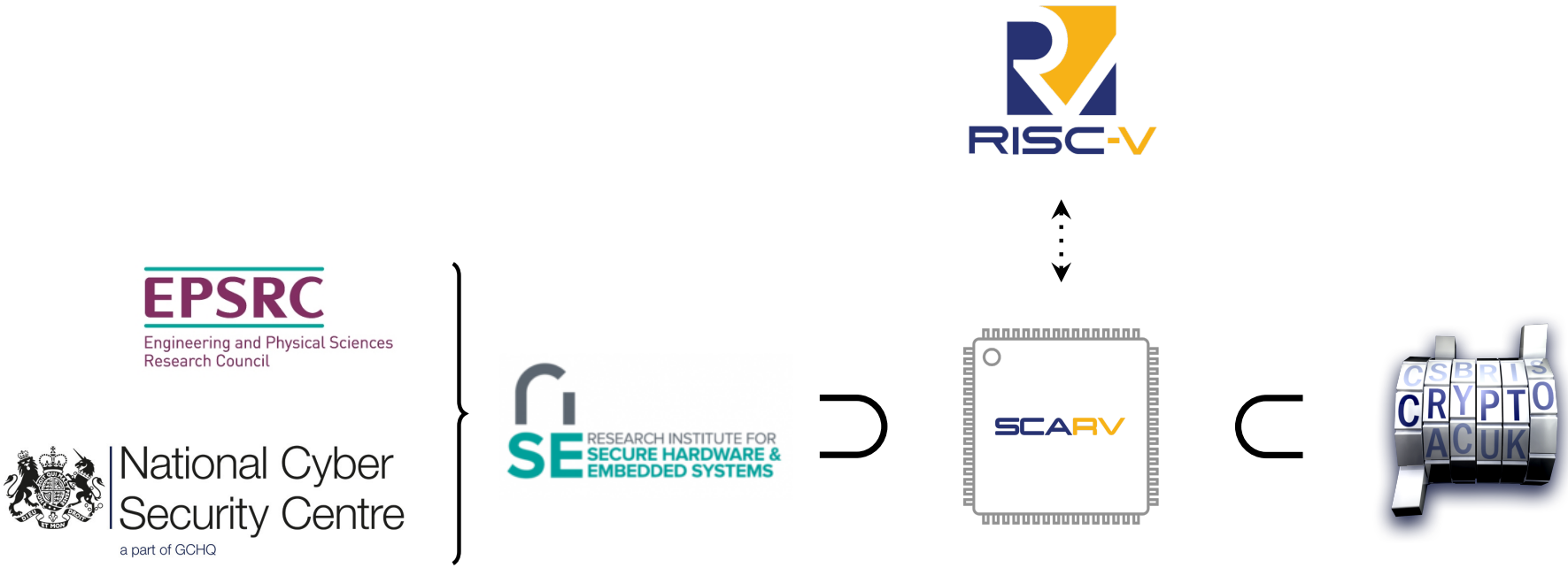
Ben Marshall, **Daniel Page**, Thinh Pham, James Webb
Department of Computer Science, University of Bristol,
Merchant Venturers Building, Woodland Road,
Bristol BS8 1UB, United Kingdom.
daniel.page@bristol.ac.uk

11/04/22 @ COSADE'22

Context

https://www.ukrise.org

Context



SCARV $\simeq$ side-channels + RISC-V
$\simeq$ cryptographic engineering + computer architecture

https://www.ukrise.org https://www.scarv.org

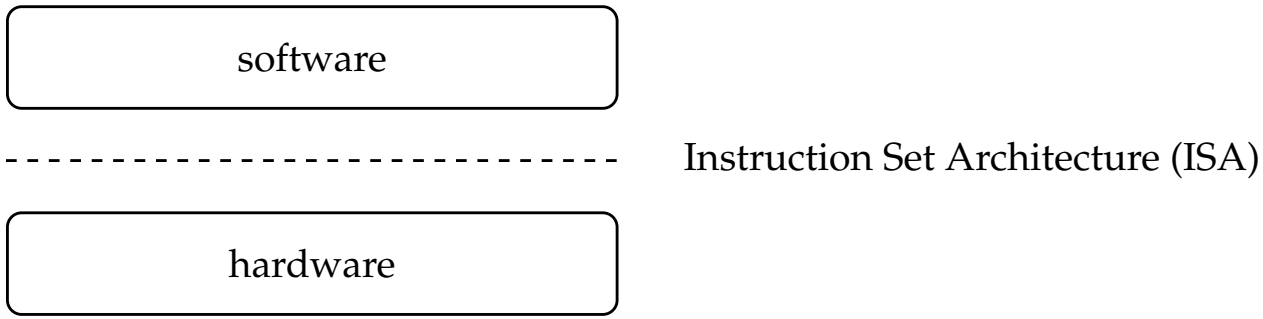
Context

– *Hackers* (https://www.imdb.com/title/tt0113243)

https://imgur.com/t/hackers/YZMw45k

Context

- (A) goal: given

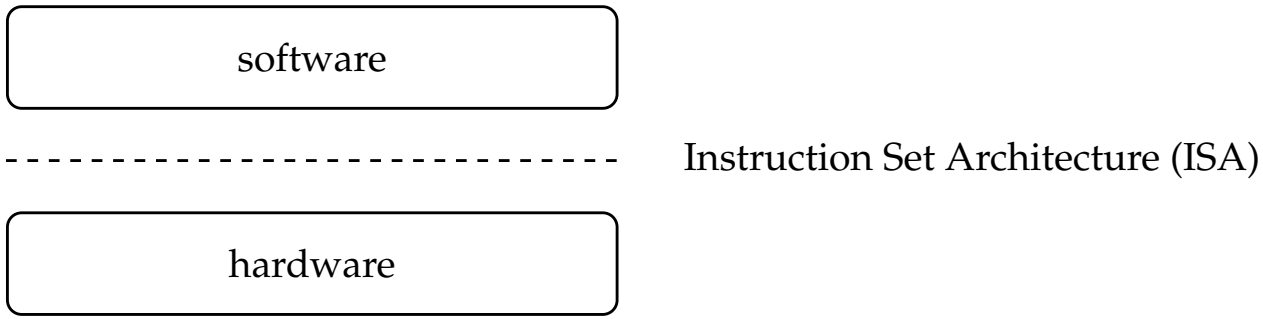


consider the role of an ISA in cryptographic engineering tasks, e.g.,

ISEs for primitives	:	[24]
ISEs for masking	:	[17, 22]
micro-architectural leakage	:	[19, 23, 16]
		⋮

Context

- (A) goal: given

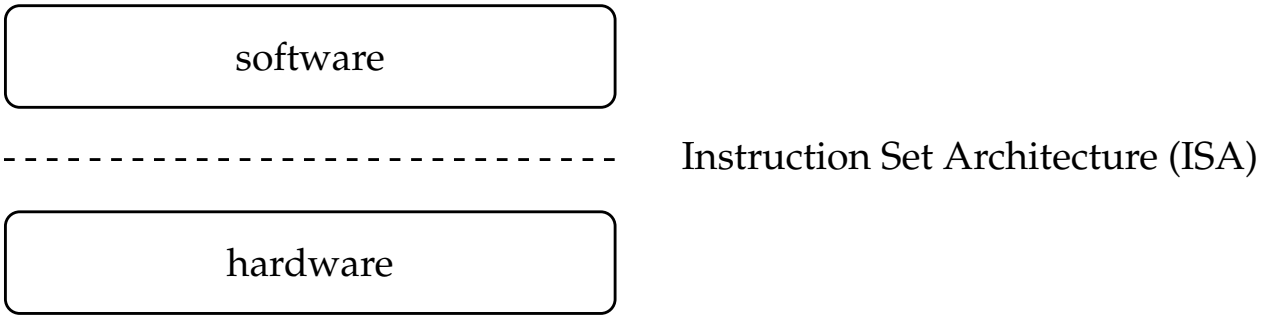


consider the role of an aISA [20] in cryptographic engineering tasks, e.g.,

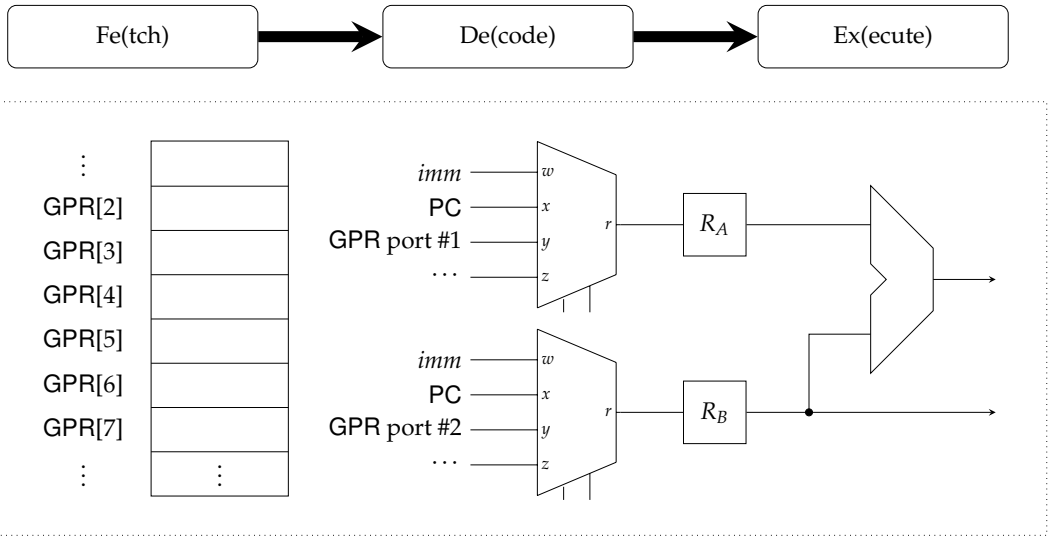
ISEs for primitives	:	[24]
ISEs for masking	:	[17, 22]
micro-architectural leakage	:	[19, 23, 16]
ISEs for leakage	:	[18]
	:	

Context

- (A) goal: given

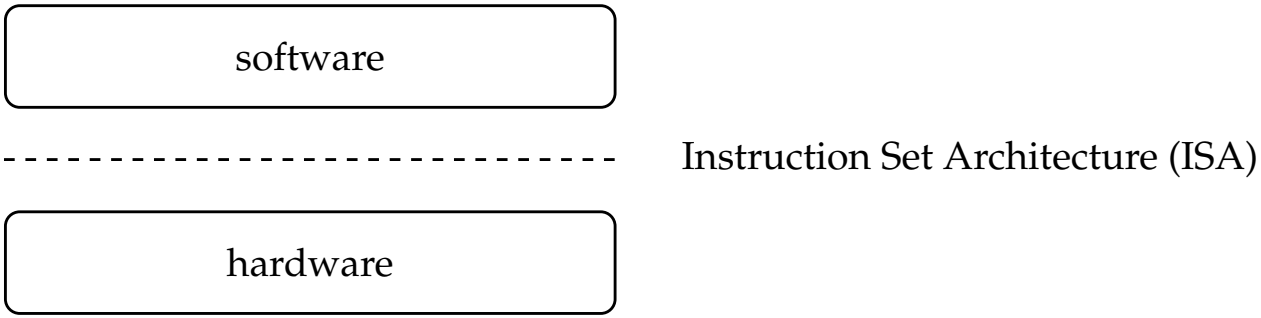


consider the role of an aISA [20] in cryptographic engineering tasks, e.g.,

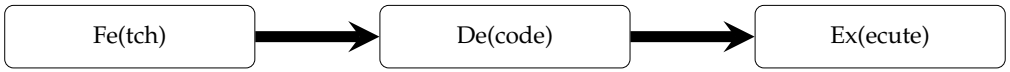


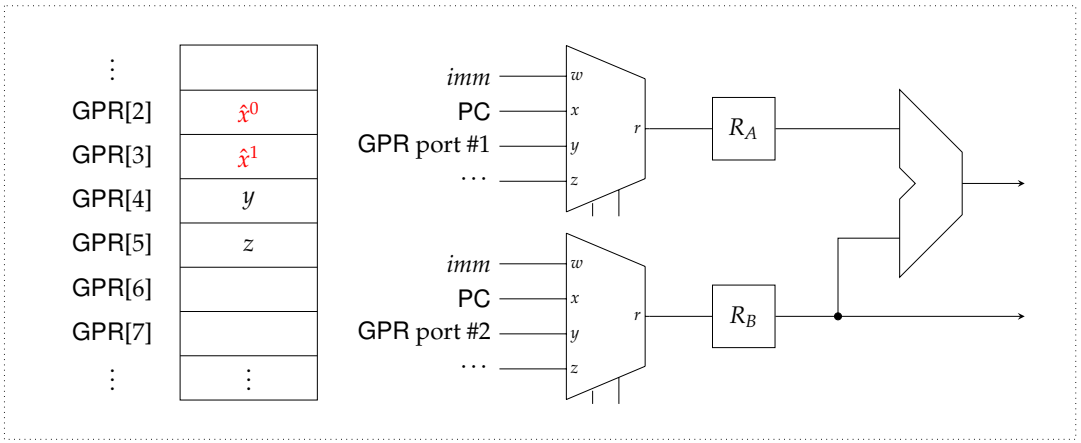
Context

- (A) goal: given



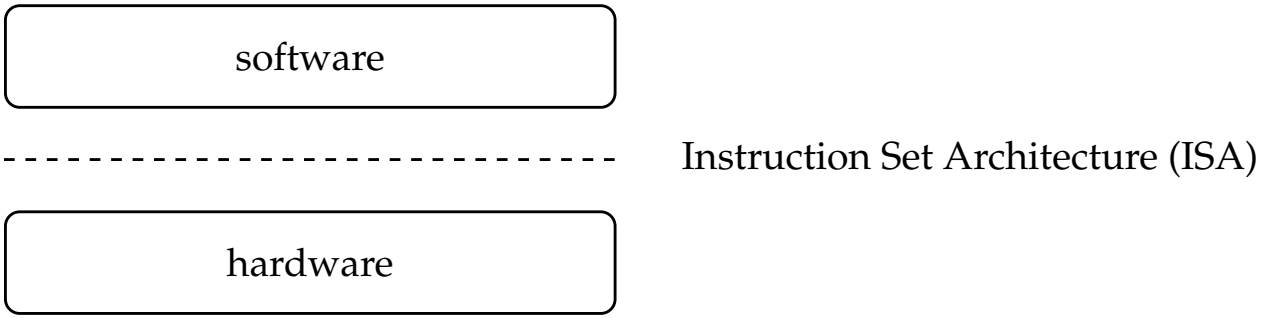
consider the role of an aISA [20] in cryptographic engineering tasks, e.g.,



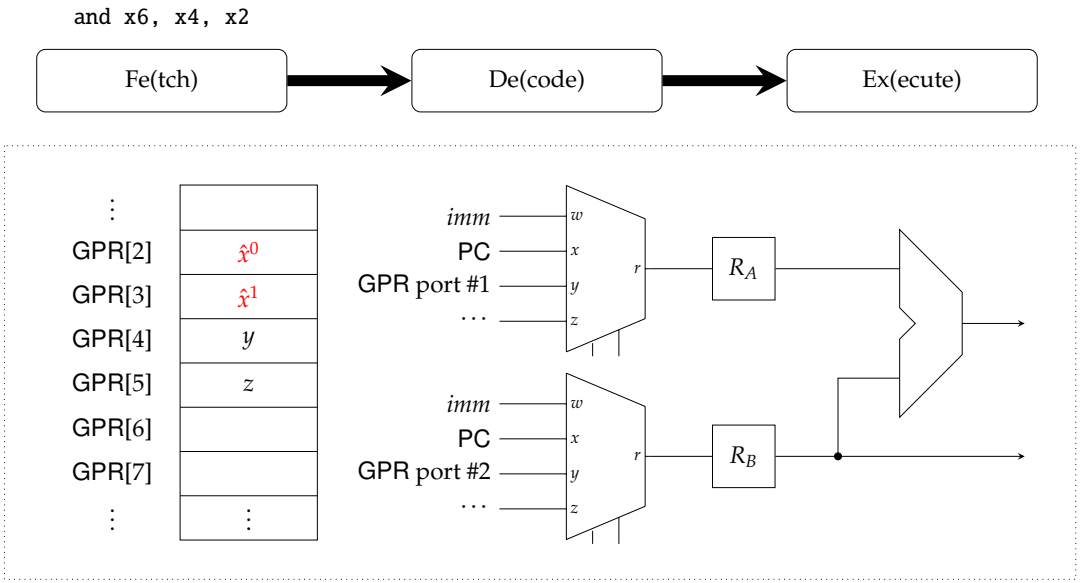


Context

- (A) goal: given

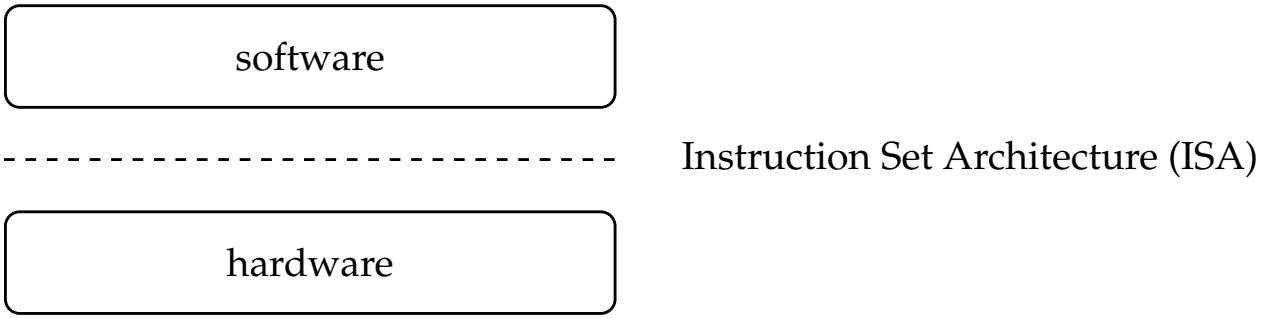


consider the role of an aISA [20] in cryptographic engineering tasks, e.g.,

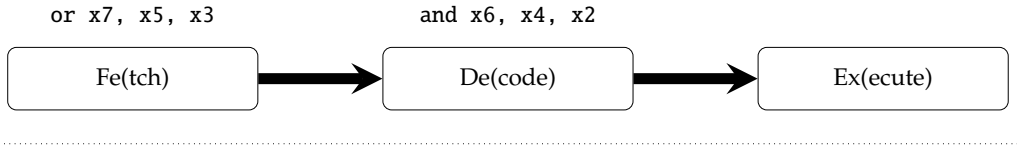


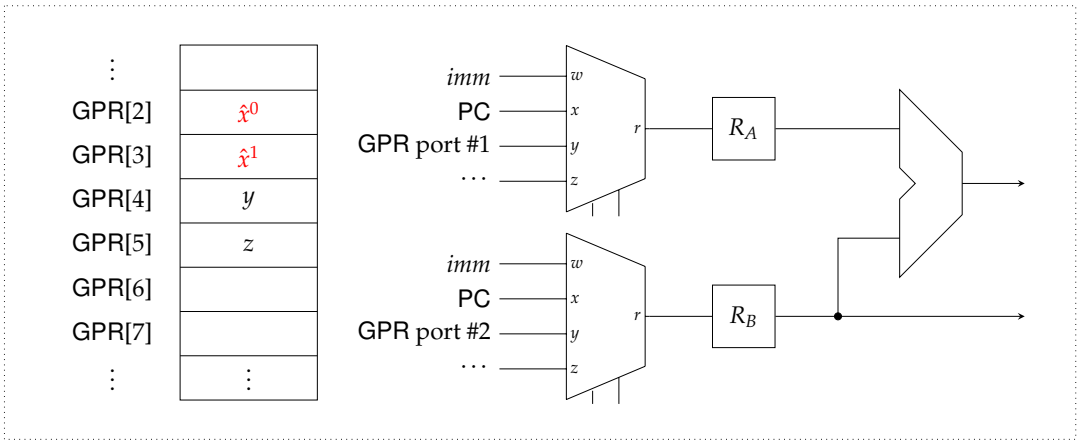
Context

- (A) goal: given



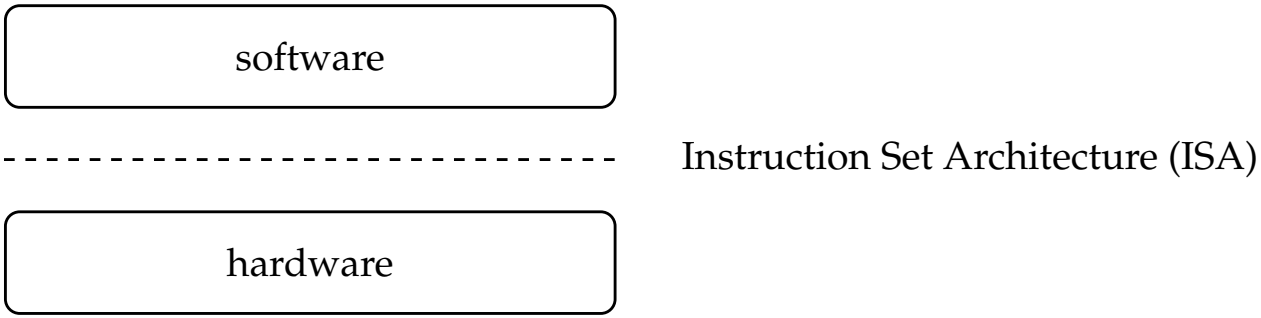
consider the role of an aISA [20] in cryptographic engineering tasks, e.g.,



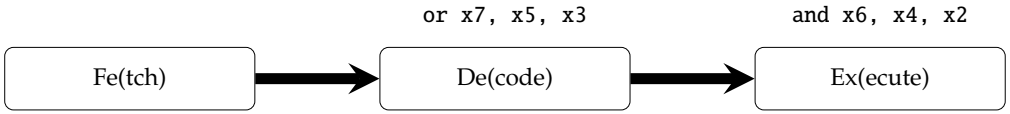


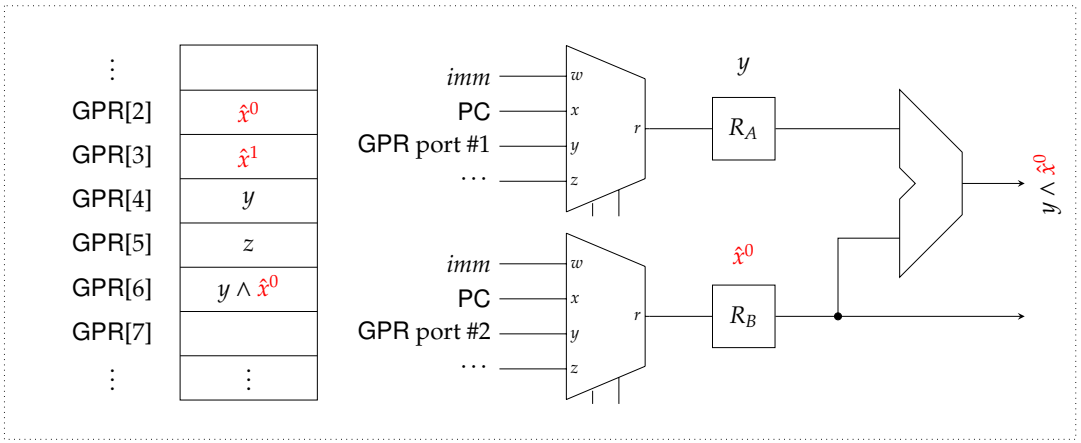
Context

- (A) goal: given



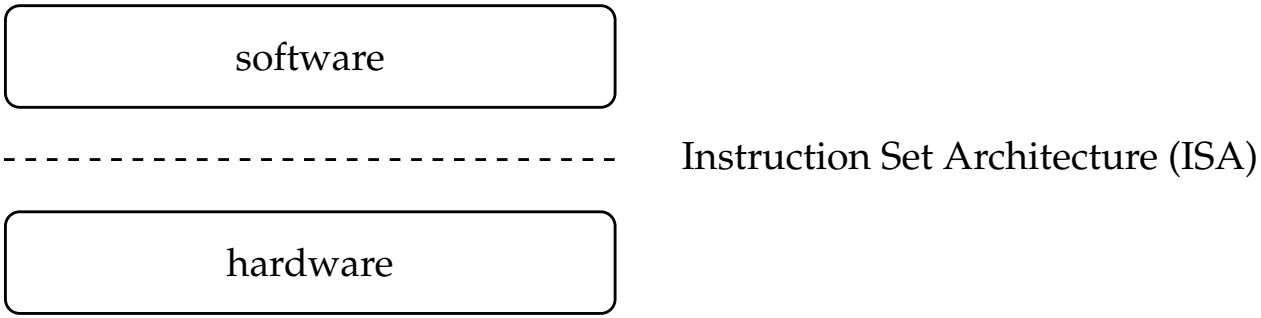
consider the role of an aISA [20] in cryptographic engineering tasks, e.g.,



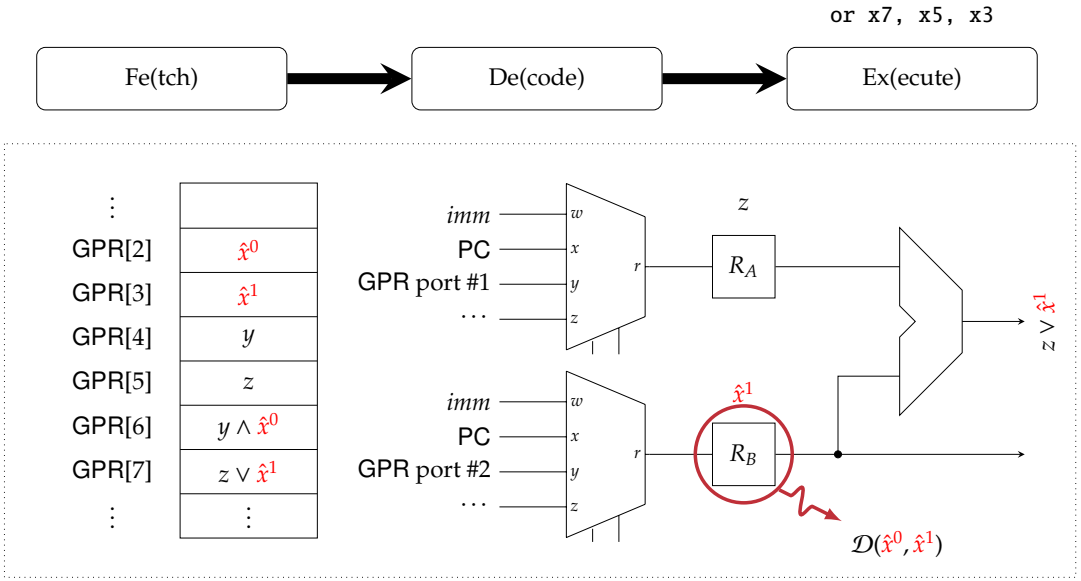


Context

- (A) goal: given

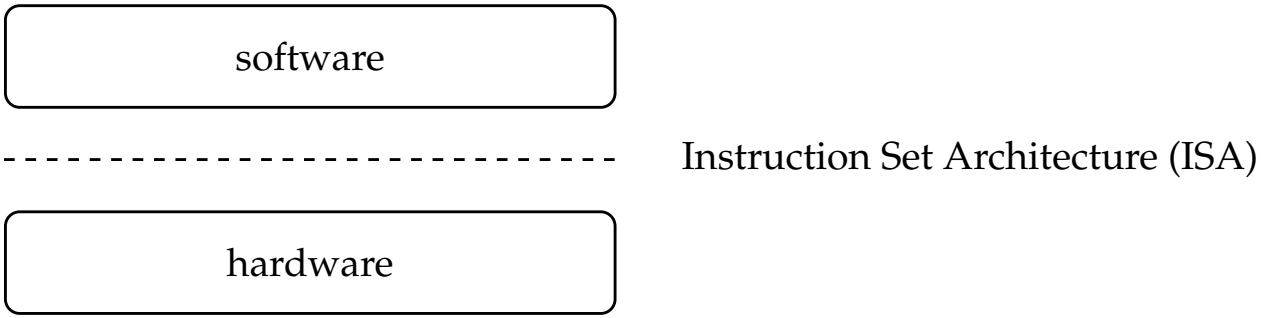


consider the role of an aISA [20] in cryptographic engineering tasks, e.g.,

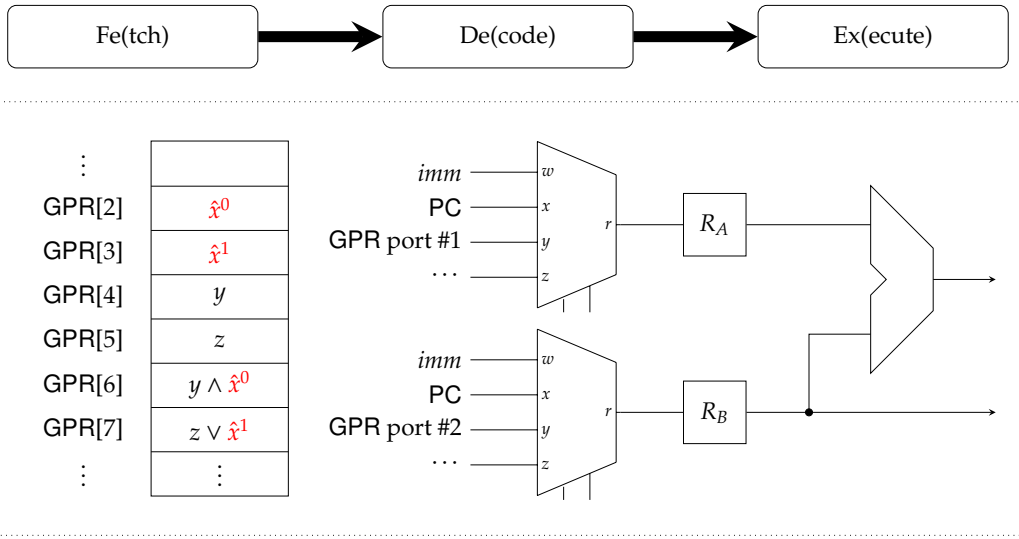


Context

- (A) goal: given



consider the role of an aISA [20] in cryptographic engineering tasks, e.g.,



+

- computational units
- memory interface
- transient execution
- ...

Context

- Problem:
  - we've produced hardware *and* software implementations,
  - we need to assess them using some form of leakage evaluation.
- Solution:
  - from a +ve perspective:
    1. we have hardware infrastructure:
       - SASEBO [2, 21]
       - ChipWhisperer [3, 25]
       - ...
    2. we have software infrastructure:
       - Jlsca [4]
       - SCAred [5]
       - ...
    3. we have data sets:
       - DPA contest [6, 15]
       - ASCAD [7, 13]
       - ...

Context

- Problem:
  - we've produced hardware *and* software implementations,
  - we need to assess them using some form of leakage evaluation.
- Solution:

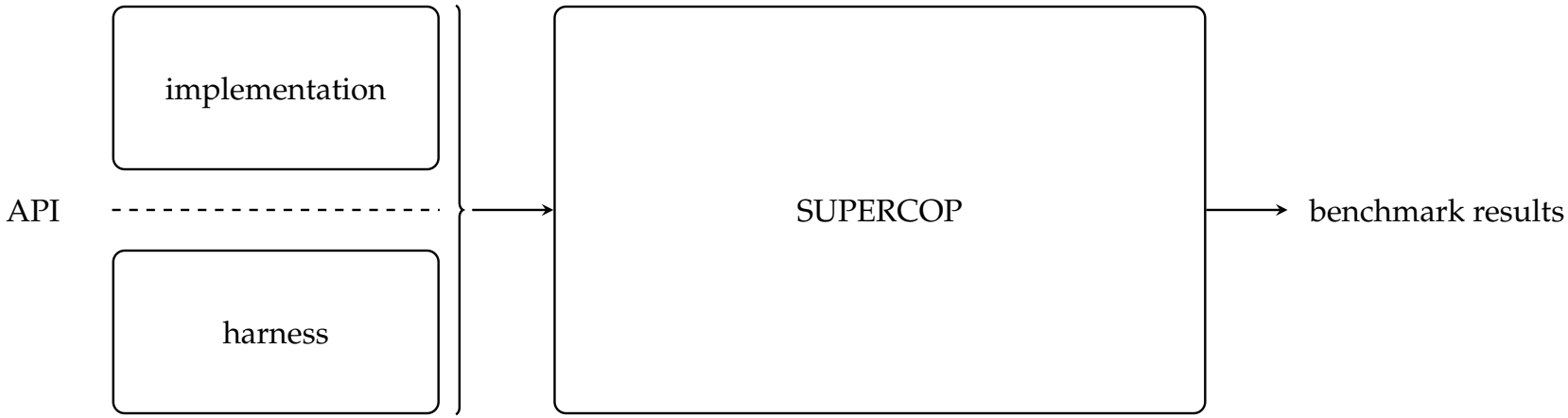
- from a −ve perspective:
  - for *some* users and use-cases, this tooling isn't ideal, e.g.,

1.	abstraction	⇒	research on "fundamentals", CI, ...
2.	reproducibility	⇒	standards, contests, surveys, ...
3.	productivity	⇒	limits on time, space, cost, etc.
	⋮		

  - so what ideas for alternatives and/or additions could make sense?

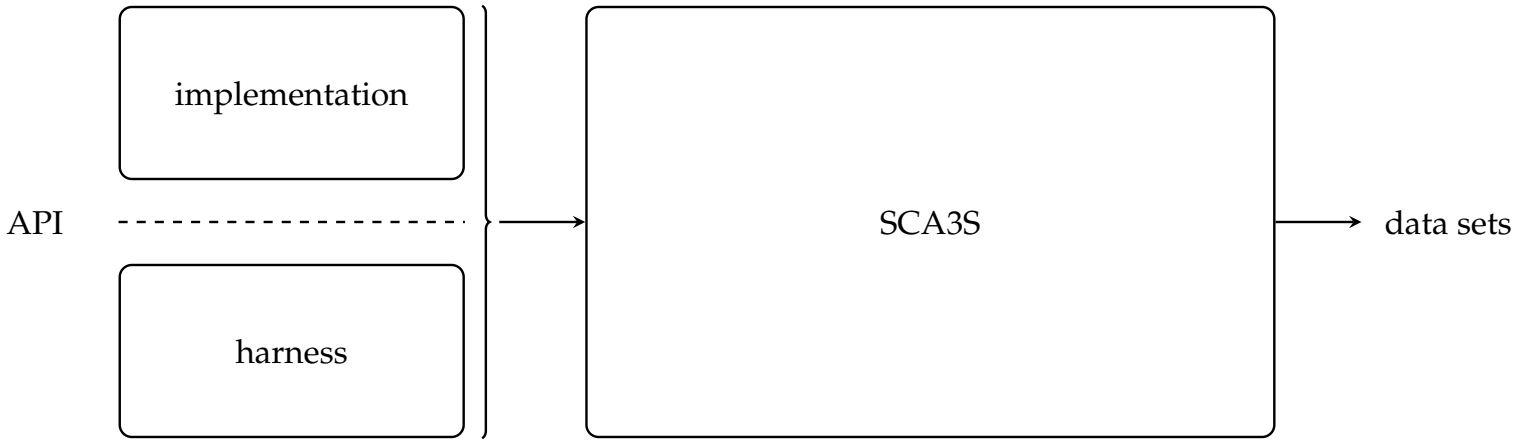
Tool #1: SCA3S ≃ remote acquisition and analysis

- Concept [8]:



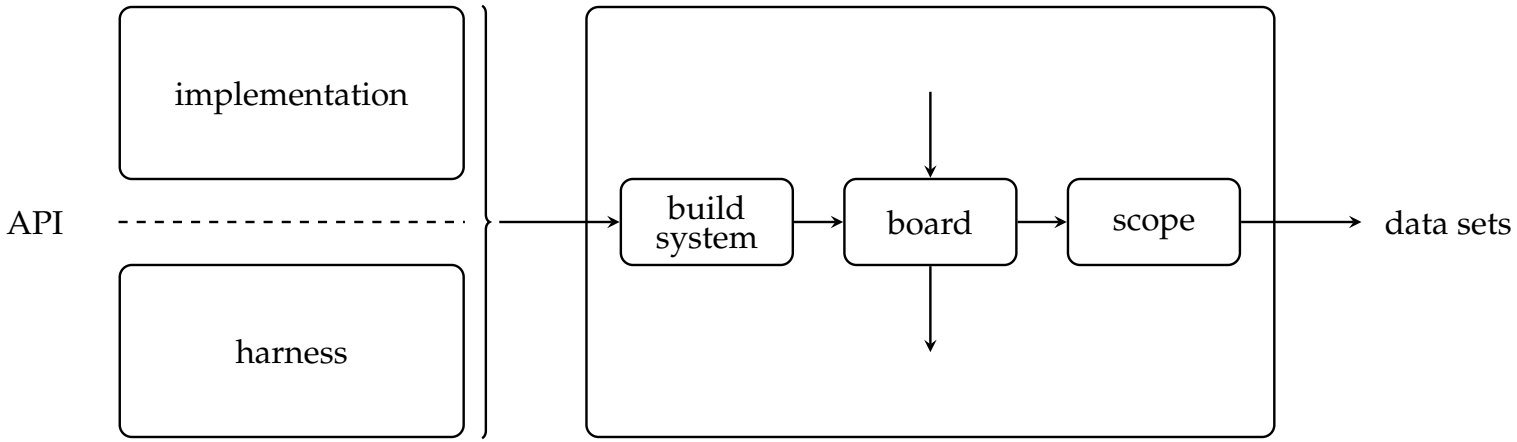
Tool #1: SCA3S ≃ remote acquisition and analysis

- Concept:



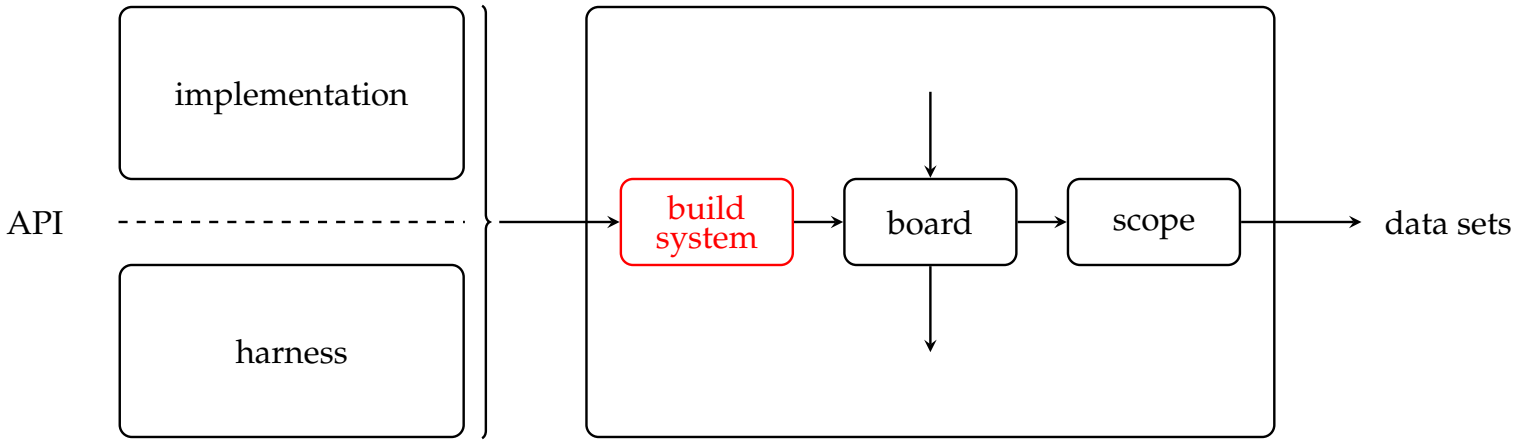
Tool #1: SCA3S ≃ remote acquisition and analysis

- Concept:



Tool #1: SCA3S ≃ remote acquisition and analysis

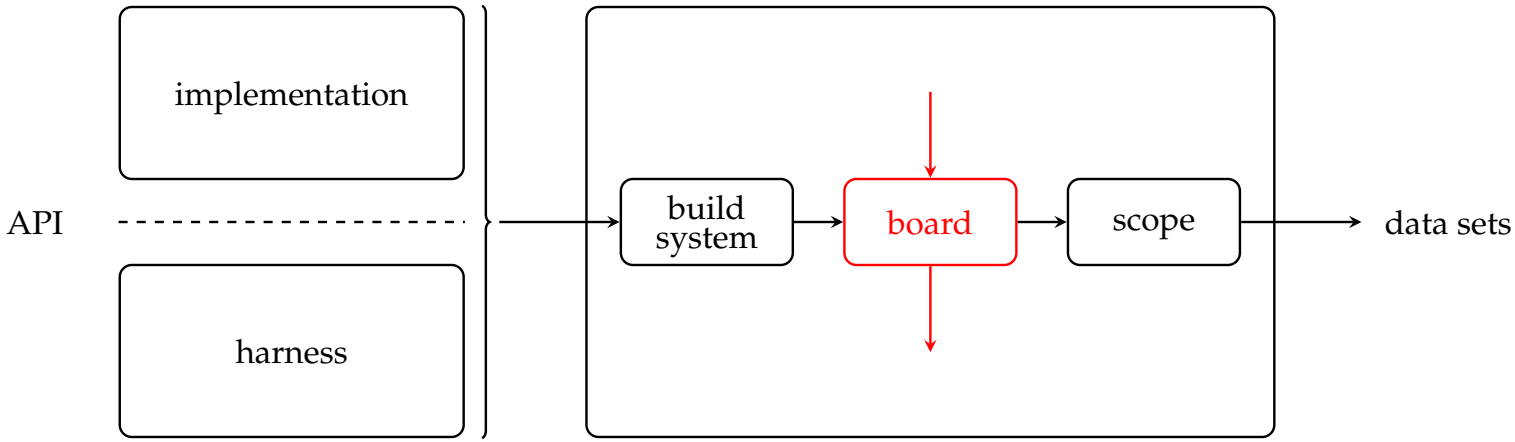
- Concept:



1. build system:
   - esp. with embedded targets, deps. and reproducibility are tricky,
   - ∴ *containerise* tool-chain, HAL, etc. using Docker.

Tool #1: SCA3S ≃ remote acquisition and analysis

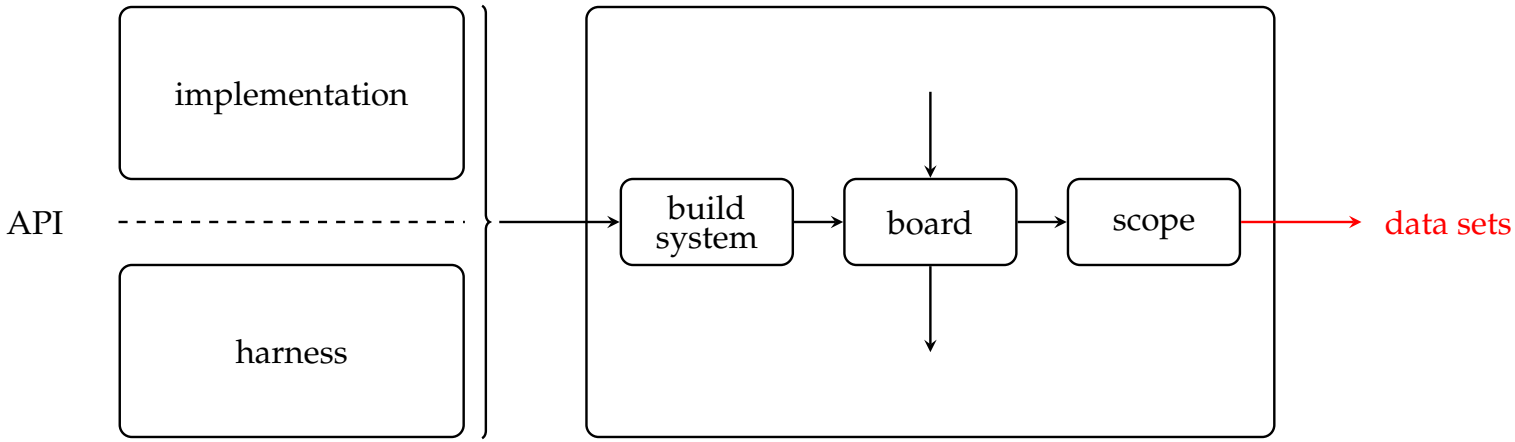
- Concept:



2. I/O protocol:
   - there *are* existing (e.g., SimpleSerial), but *no* standard choices,
   - want higher level of abstraction, ∴
     - implementation defines set of registers (with type, plus fixed or variable length content),
     - implementation defines set of kernels,
     - harness provides a mechanism to interact and introspect.

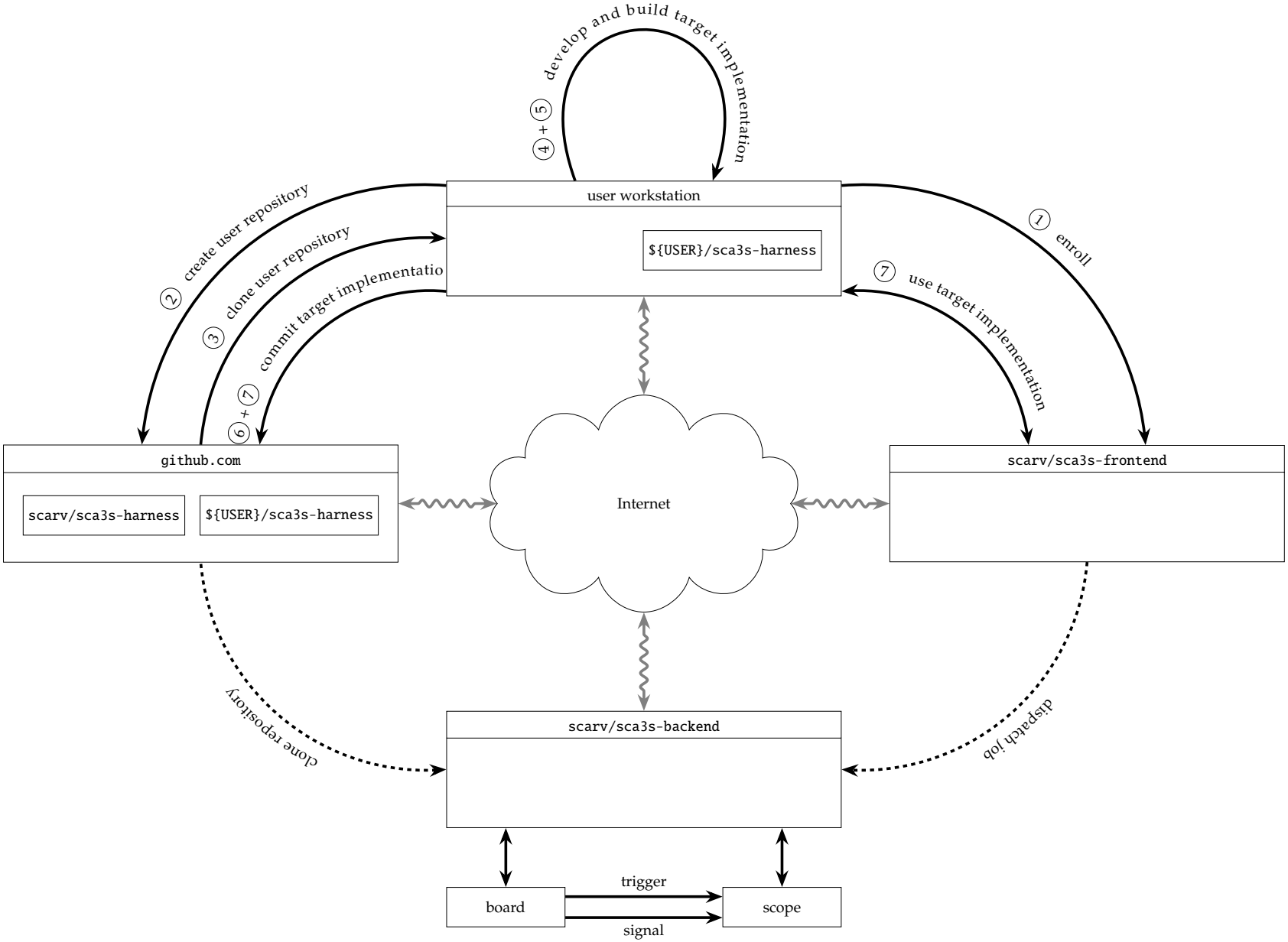
Tool #1: SCA3S ≃ remote acquisition and analysis

- Concept:



3. data set format:
   - there *are* existing (e.g., TRS), but *no* standard instances,
   - ∴ leverage existing technology, namely HDF5.

Tool #1: SCA3S ≃ remote acquisition and analysis



Tool #1: SCA3S ≃ remote acquisition and analysis

- Advert:
  - the prototype stems from a UG dissertation (James Webb)

    https://github.com/scarv/theses/blob/master/meng/jw15520.pdf
  - there's an umbrella repo. at

    https://github.com/scarv/sca3s

    captures all related components,
  - the user-facing repo. is at

    https://github.com/scarv/sca3s-harness

    under which the wiki

    https://github.com/scarv/sca3s-harness/wiki

    documents the workflow,
  - the user-facing web interface is live at

    https://sca3s.scarv.org

Tool #1: SCA3S ≃ remote acquisition and analysis

- Outlook:
  - what it *can* do:
    - support somewhat plug-and-play equipment: ours supports

∋

- CW308-based Cortex-M0 + PicoScope 5444b
- CW308-based Cortex-M3 + PicoScope 5444b
- CW305-based RISC-V + PicoScope 5444b
- SASEBO-based RISC-V + PicoScope 5444b
- GILES [9] (a derivative of ELMO [1])
- ...

    but the front- and back-ends are decoupled, so others could co-exist,
    - parameterised data set acquisition,
    - trigger a TVLA-based analysis of commit into repository,
    - ...

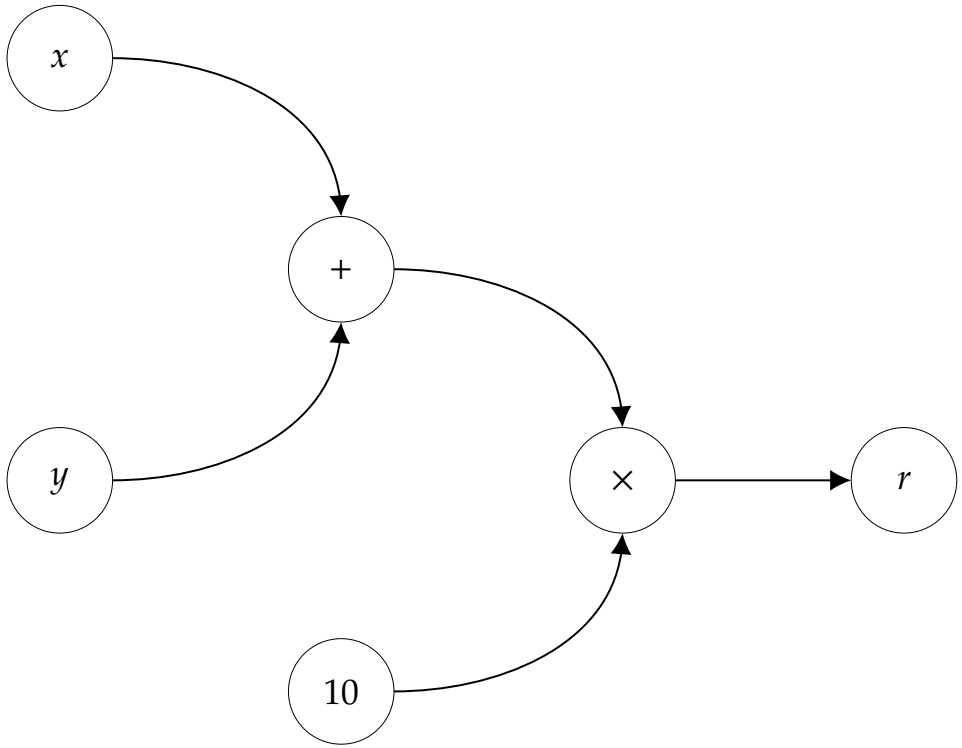
Tool #1: SCA3S ≃ remote acquisition and analysis

- Outlook:

- what it *could* do:
  - act as a corpus to study cryptographic implementation,
  - run automated analysis for $X \in \{\text{constant-time'ness, fault attack, } \ldots\}$,
  - support artefact evaluation processes,
  - support standardisation processes (cf. [14]),
  - support surveys (cf. [23]),
  - ...

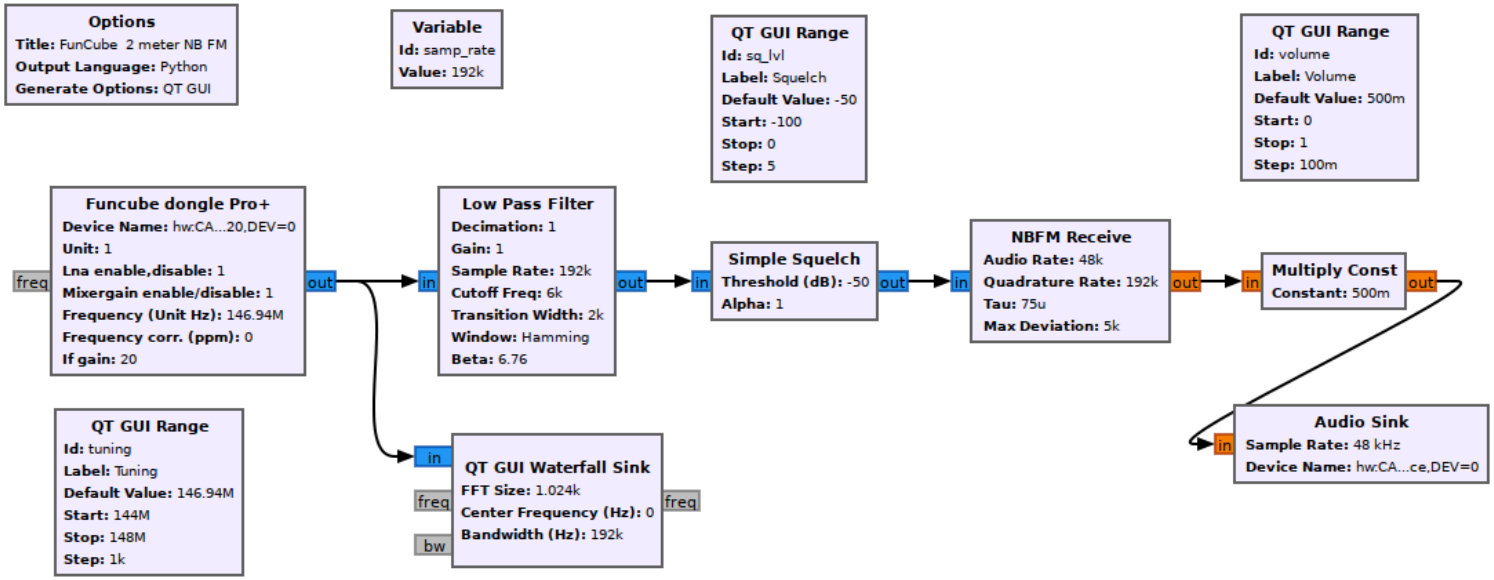
Tool #2: `scaflow` ≃ analysis as data-flow programming

- Concept: model computation as a directed graph.



Tool #2: `scaflow` ≃ analysis as data-flow programming

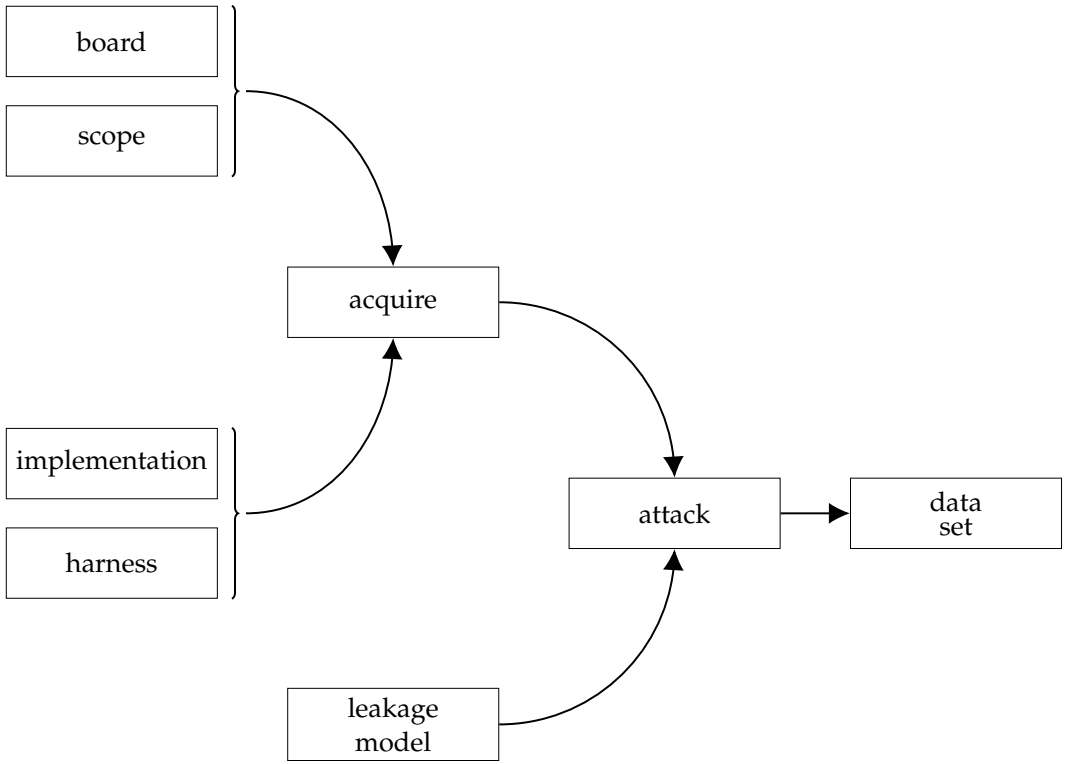
- Concept: model computation as a directed graph.



https://wiki.gnuradio.org/index.php?title=File:FunCube_2_meter_NB_FM_fg.png

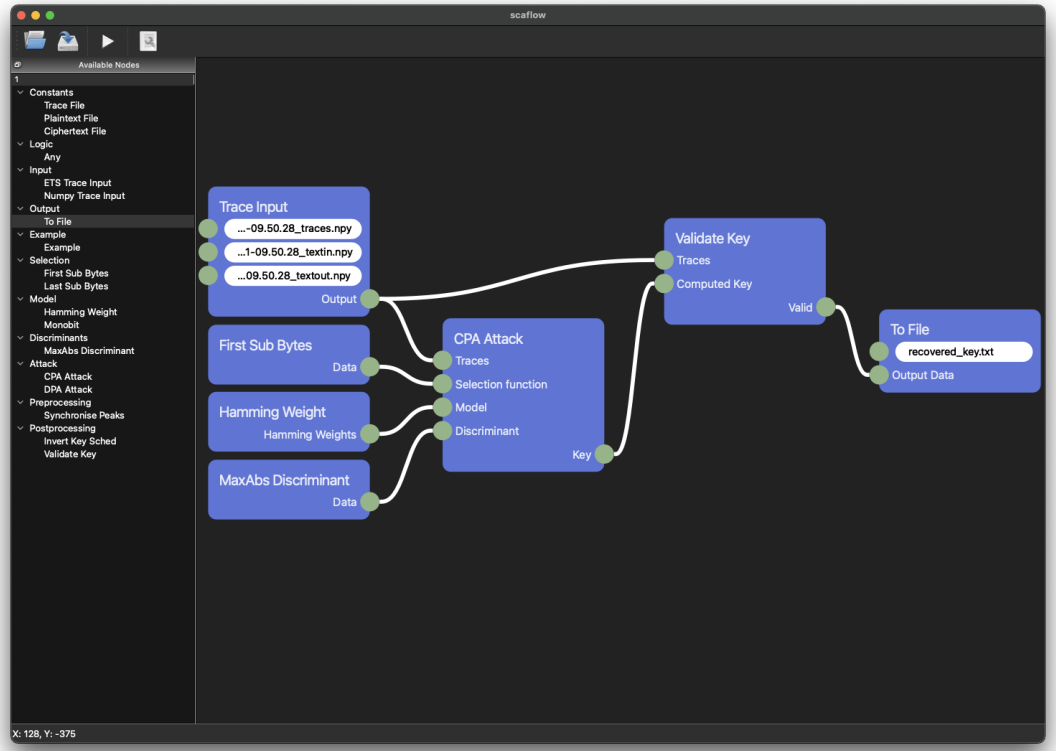
Tool #2: `scaflow` ≃ analysis as data-flow programming

- Concept: model computation as a directed graph.



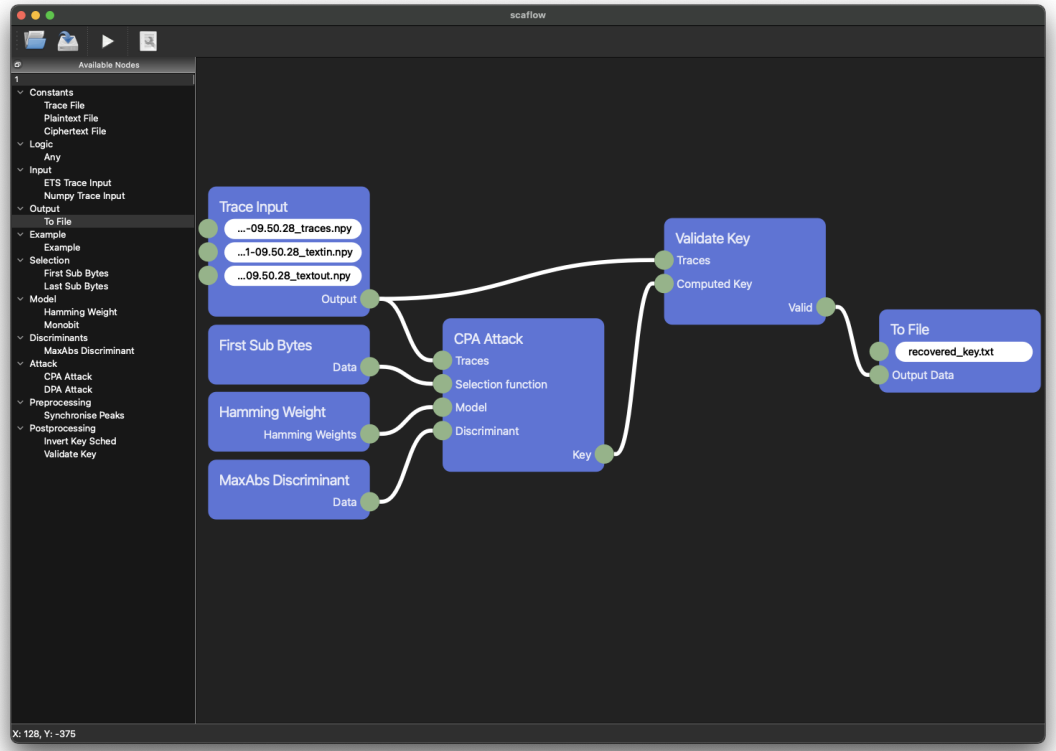
Tool #2: `scaflow` ≃ analysis as data-flow programming

- Concept: model computation as a directed graph.



Tool #2: `scaflow` ≃ analysis as data-flow programming

- Concept: model computation as a directed graph.



but *why* ... similar argument to LabVIEW [10], e.g.,

1. high level of abstraction, e.g., emphasising modular workflow,
2. decouple specification from execution of workflow,
3. ...

Tool #2: `scaflow` ≃ analysis as data-flow programming

- Advert:
  - the prototype stems from a UG dissertation (Fergus Longley)

    `https://github.com/scarv/theses/blob/master/meng/fl17431.pdf`
  - there's an umbrella repo. at

    `https://github.com/scarv/scaflow`

    captures all related components.

Tool #2: `scaflow` ≃ analysis as data-flow programming

- Outlook:
  - what it *can* do: errr, not a lot
    - manipulation of basic nodes via UI (plus some exploration of "advanced" nodes, e.g., choice),
    - execute workflow using underlying library (e.g., to solve CHES challenge),
    - serialise workflow into JSON,
    - ...

    this was very much a PoC only, so a re-write, e.g., using Ryven [11], is important first step.

Tool #2: `scaflow` ≃ analysis as data-flow programming

- Outlook:
  - what it *could* do:
    - enable transparent, intelligent performance decisions (offload, caching, etc.),
    - promote (more) modular, inter-operable analysis "blocks" (e.g., libraries),
    - support artefact evaluation processes,
    - ...

Conclusions

- A −ve take: on one hand,
  - there's nothing particularly *deep* here,
  - none of the tools are particularly mature: *lots* of "what if" and "yeah but" questions,
  - they aren't the right or even a viable approach for every user or use-case.

Conclusions

- A +ve take: on the other hand,
  - related people and artefacts should be valued (cf. US-RSE [12]),
  - community is important (cf. ECRYPT); tools like this help foster the community,
  - maturity of a research field is reflected (to some extent) in available infrastructure,
  - scale and reproducibility (e.g., wrt. ML-based SCA) clearly demand automation,
  - lots of interesting opportunities for impact,

  so the ideas more so than the tools seem of value.

Conclusions

- A +ve take: on the other hand,
  - related people and artefacts should be valued (cf. US-RSE [12]),
  - community is important (cf. ECRYPT); tools like this help foster the community,
  - maturity of a research field is reflected (to some extent) in available infrastructure,
  - scale and reproducibility (e.g., wrt. ML-based SCA) clearly demand automation,
  - lots of interesting opportunities for impact,

  so the ideas more so than the tools seem of value, and (arguably) hint at open challenges, e.g.,
  - usability:
    - what does a CI-like "badge" mean for leakage evaluation?
    - how to usefully communicate TVLA-like output into development cycles?
    - ...
  - performance:
    - what do genuinely efficient on-disk and in-memory data set formats look like?
    - ...

Questions?

References

[1] D. McCann, E. Oswald, and C. Whitnall. "Towards Practical Tools for Side Channel Aware Software Engineering: 'Grey Box' Modelling for Instruction Leakages". In: *USENIX Security Symposium*. 2017, pp. 199–216 (see pp. 24, 25).

[2] URL: https://satoh.cs.uec.ac.jp/SASEBO (see pp. 14, 15).

[3] URL: https://www.newae.com/chipwhisperer (see pp. 14, 15).

[4] URL: https://github.com/Riscure/Jlsca (see pp. 14, 15).

[5] URL: https://github.com/eshard/scared (see pp. 14, 15).

[6] URL: https://www.dpacontest.org (see pp. 14, 15).

[7] URL: https://github.com/ANSSI-FR/ASCAD (see pp. 14, 15).

[8] URL: https://bench.cr.yp.to/supercop.html (see p. 16).

[9] URL: https://github.com/sca-research/GILES (see pp. 24, 25).

[10] URL: https://www.ni.com/en-gb/shop/software/products/labview.html (see pp. 26–30).

[11] URL: https://github.com/leon-thomm/Ryven (see pp. 32, 33).

[12] URL: https://us-rse.org (see pp. 34–36).

[13] R. Benadjila et al. "Deep learning for side-channel analysis and introduction to ASCAD database". In: *Journal of Cryptographic Engineering* 10.2 (2020), pp. 163–188. URL: https://doi.org/10.1007/s13389-019-00220-8 (see pp. 14, 15).

[14] *Call for Side-Channel Security Validation Labs*. Tech. rep. 2022. URL: https://cryptography.gmu.edu/athena/LWC/Call_for_Security_Evaluation_Labs.pdf (see pp. 24, 25).

[15] C. Clavier et al. "Practical improvements of side-channel attacks on AES: feedback from the 2nd DPA contest". In: *Journal of Cryptographic Engineering* 4.4 (2014), pp. 259–274. URL: https://doi.org/10.1007/s13389-014-0075-9 (see pp. 14, 15).

[16] S. Gao, E. Oswald, and D. Page. "Towards Micro-Architectural Leakage Simulators: Reverse Engineering Micro-Architectural Leakage Features is Practical". In: *To appear in Theory and Application of Cryptographic Techniques (EUROCRYPT)*. 2022 (see pp. 5, 6).

References

[17] S. Gao et al. "An Instruction Set Extension to Support Software-Based Masking". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* 2021.4 (2021), pp. 283–325. URL: https://doi.org/10.46586/tches.v2021.i4.283-325 (see pp. 5, 6).

[18] S. Gao et al. "FENL: an ISE to mitigate analogue micro-architectural leakage". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* 2020.2 (2020), pp. 73–98. URL: https://doi.org/10.13154/tches.v2020.i2.73-98 (see pp. 5, 6).

[19] S. Gao et al. "Share slicing: friend or foe?" In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* 2020.1 (2019), pp. 152–174. URL: https://doi.org/10.13154/tches.v2020.i1.152-174 (see pp. 5, 6).

[20] Q. Ge, Y. Yarom, and G. Heiser. "No security without time protection: we need a new hardware-software contract". In: *Asia-Pacific Workshop on Systems (APSys)*. https://doi.org/10.1145/3265723.3265724. 2018 (see pp. 5–13).

[21] Y. Hori et al. "SASEBO-GIII: A hardware security evaluation board equipped with a 28-nm FPGA". In: *IEEE Global Conference on Consumer Electronics*. 2012, pp. 657–660. URL: https://doi.org/10.1109/GCCE.2012.6379944 (see pp. 14, 15).

[22] B. Marshall and D. Page. *SME: Scalable Masking Extensions*. Cryptology ePrint Archive, Report 2021/1416. 2021 (see pp. 5, 6).

[23] B. Marshall, D. Page, and J. Webb. "MIRACLE: MIcRo-ArChitectural Leakage Evaluation". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* 2022.1 (2022), pp. 175–220. URL: https://doi.org/10.46586/tches.v2022.i1.175-220 (see pp. 5, 6, 24, 25).

[24] B. Marshall et al. "The design of scalar AES Instruction Set Extensions for RISC-V". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* 2021.1 (2021), pp. 109–136. URL: https://doi.org/10.46586/tches.v2021.i1.109-136 (see pp. 5, 6).

[25] C. O'Flynn and Z. Chen. "ChipWhisperer: An Open-Source Platform for Hardware Embedded Security Research". In: *Constructive Side-Channel Analysis and Secure Design (COSADE)*. LNCS 8622. Springer-Verlag, 2014, pp. 243–260. URL: https://doi.org/10.1007/978-3-319-10175-0_17 (see pp. 14, 15).