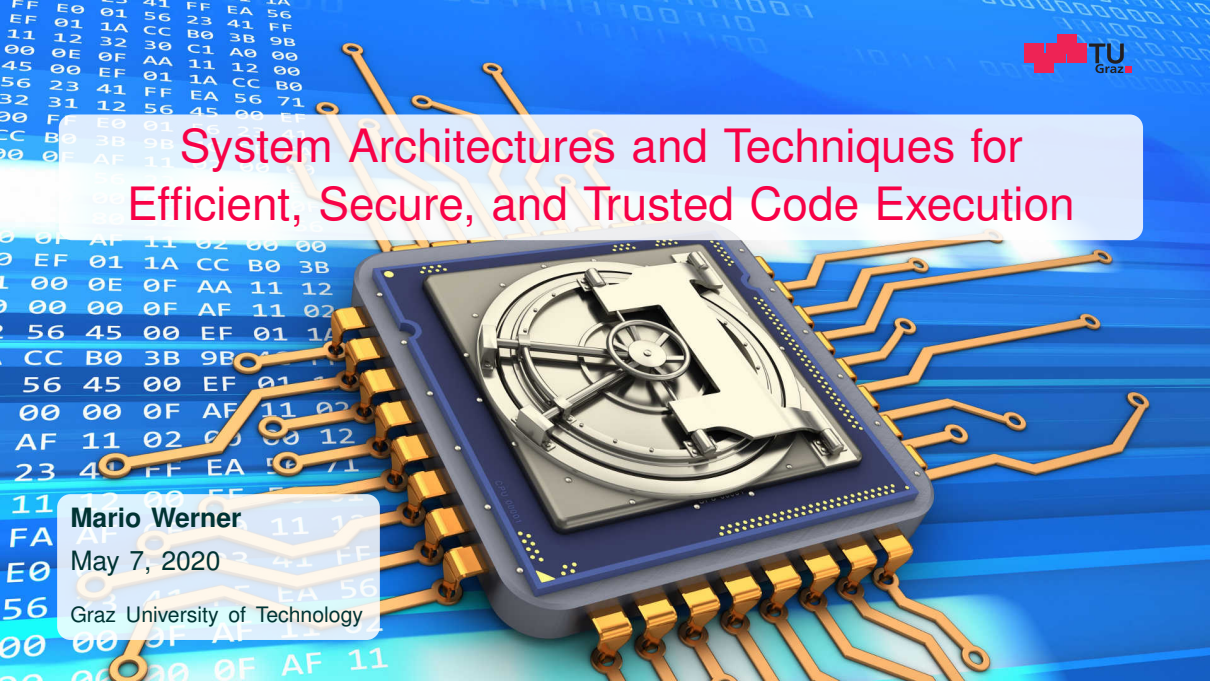
System Architectures and Techniques for Efficient, Secure, and Trusted Code Execution

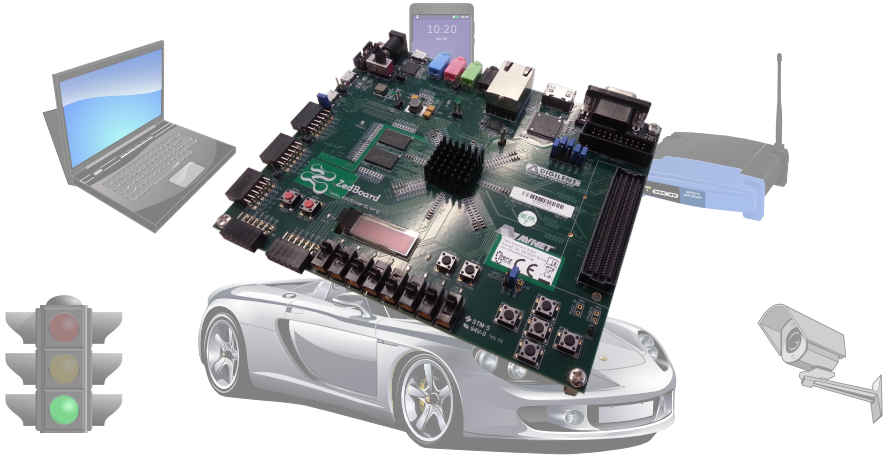


Mario Werner

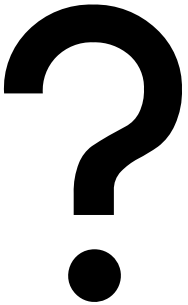
May 7, 2020

Graz University of Technology

Why do we care about code?



Problem Statement

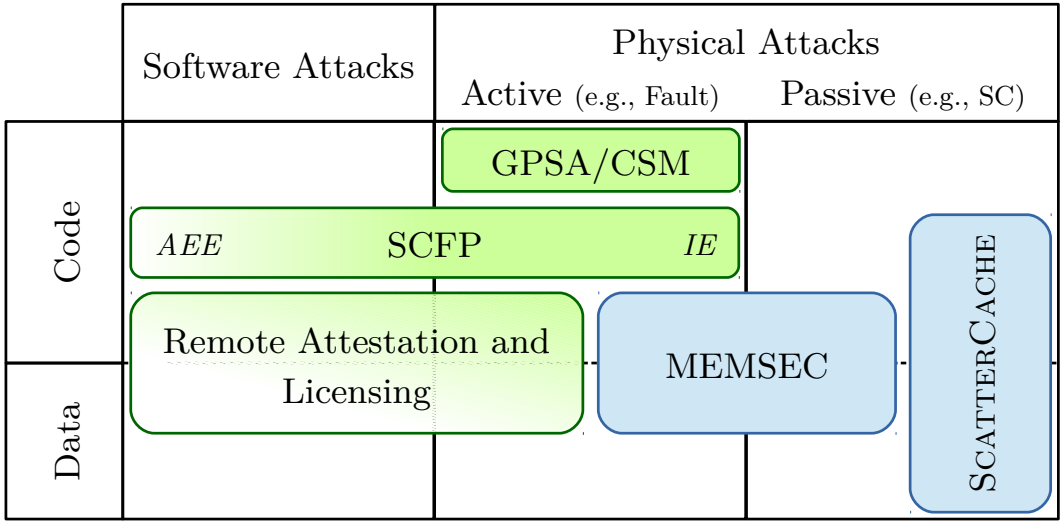


- Most research and vendor solely focus on software security
- Processors lack protection against physical attacks
- Adversary can exploit physical properties of a device
  - Active: e.g., induce faults (voltage/clock glitches, laser) [KSV13]
  - Passive: e.g., extract secrets (power/timing analysis) [MOP07]
- Local and remote attack settings (e.g., cache attacks, Rowhammer [Kim+14], Plundervolt [Mur+20])
- Relevant in mobile, IoT, and cloud settings

Contribution

- Techniques to protect code execution against physical attacks
- **Providing Control-Flow Integrity and Attestation**
  - GPSA: adapt the old scheme as fault-attack countermeasure
  - SCFP: extends GPSA using sponge-based AE techniques
  - Remote attestation and licensing using SCFP-like approaches
- **Counteracting Physical Attacks on the Memory System**
  - MEMSEC: hardware framework for memory encryption
  - SCATTERCACHE: cache design hardened against timing attacks
- Concepts tested in simulation and/or on real hardware
- Fully integrated into the toolchain for good usability
- Combined or individual adoption possible

Paper Relations



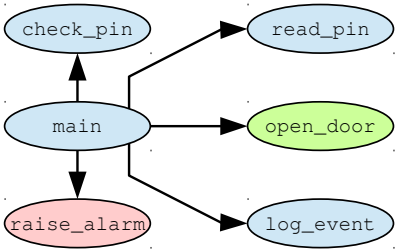
The PhD in Numbers

- 16 (co-)authored papers in total
- 4 papers in this thesis
- Contributed to 2 lectures
  - System-on-Chip Architectures and Modeling (since WT2014)
  - Computer Organization (ST2019) $\rightarrow$ Computer Organization and Networks (WT2019)
- Supervised 18 student projects
  - 4 bachelor theses
  - 4 master projects
  - 8 master theses
  - 2 internships

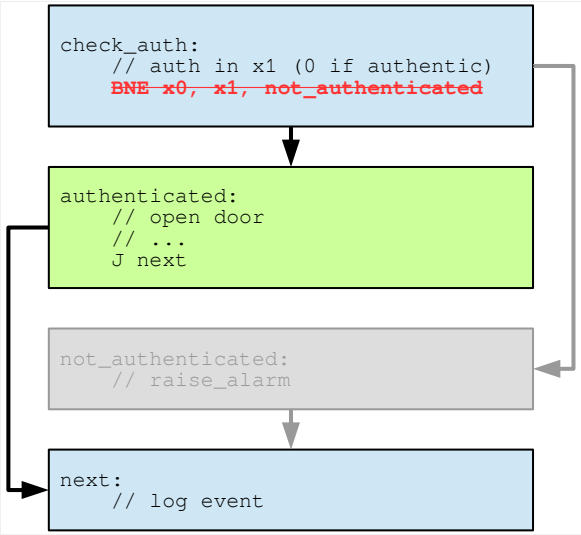
Part I: Providing Control-Flow Integrity and Attestation

Why Control-Flow Integrity?

```
void main() {
  for (;;) {
    unsigned pin = read_pin();
    int auth = check_pin(pin);
    if( auth ) {
        open_door();
    } else {
        raise_alarm();
    }
    log_event();
  }
}
```



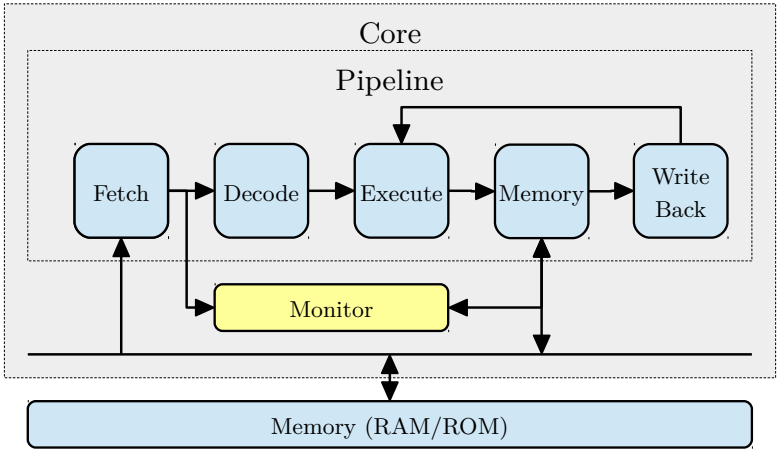
Why Control-Flow Integrity?



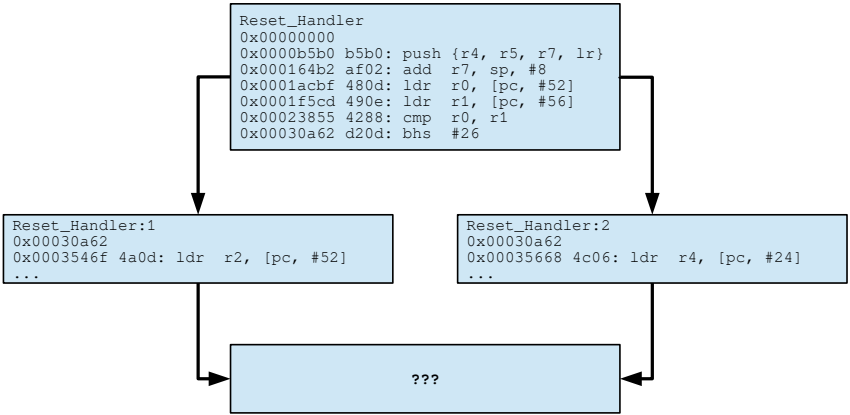
Protecting the Control Flow of Embedded Processors against Fault Attacks

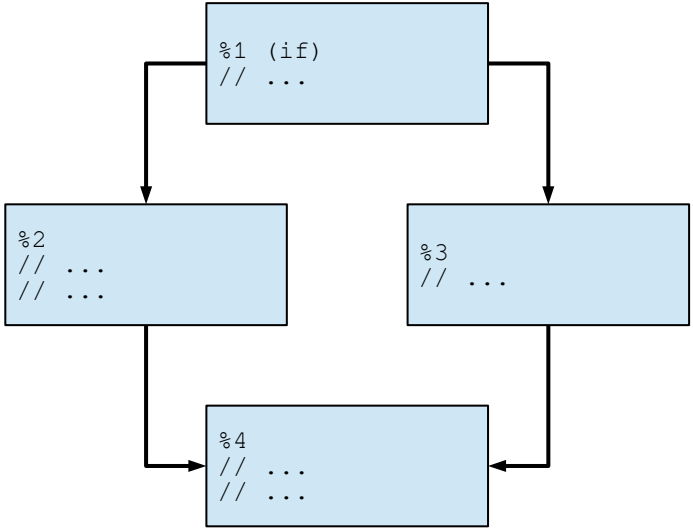
Mario Werner, Erich Wenger, and Stefan Mangard. “Protecting the Control Flow of Embedded Processors against Fault Attacks”. In: *Smart Card Research and Advanced Applications – CARDIS*. 2015, pp. 161–176. DOI: 10.1007/978-3-319-31271-2_10

High-Level Concept

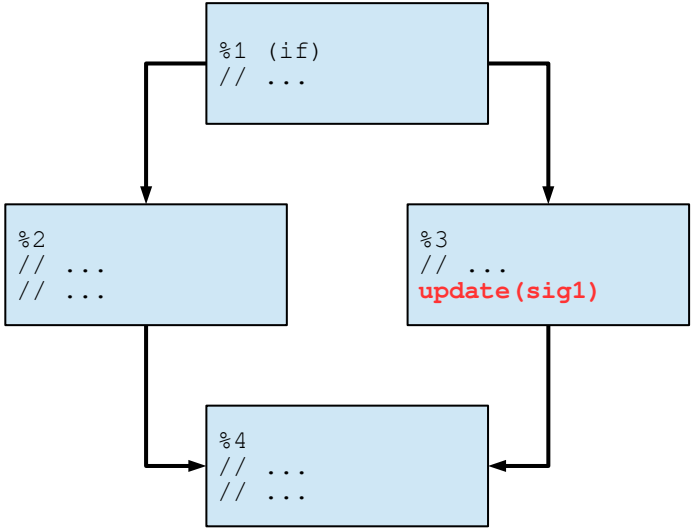


Derived Signatures [MM88]

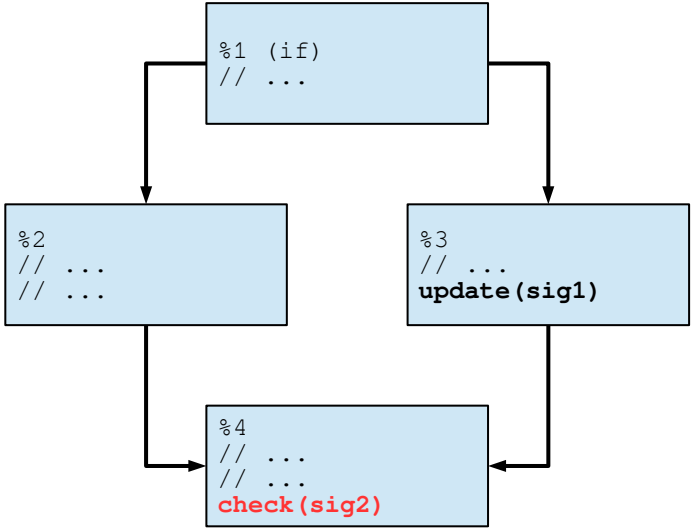



%1 (if)
// ...
%2
// ...
// ...
%3
// ...
%4
// ...
// ...

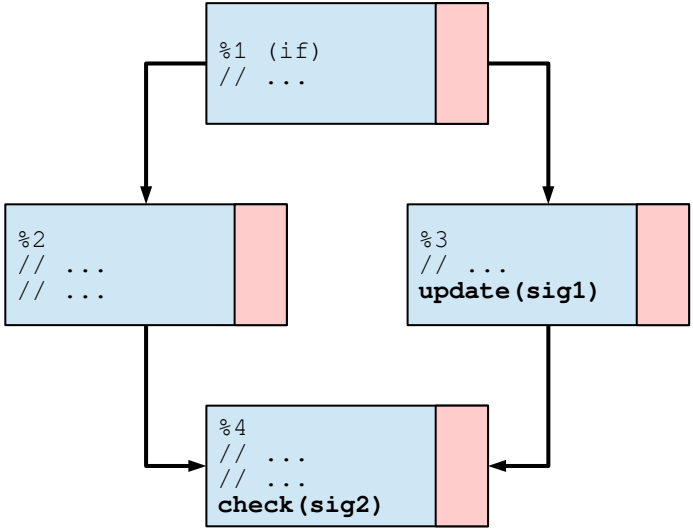
Continuous Signature Monitoring (CSM) [WS88]



Continuous Signature Monitoring (CSM) [WS88]



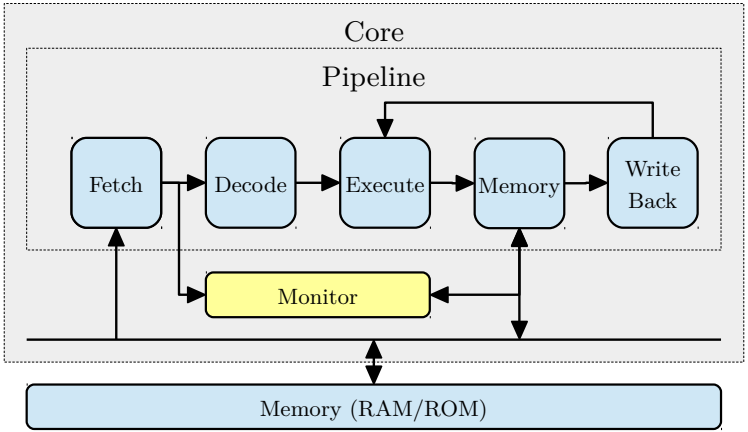
Continuous Signature Monitoring (CSM) [WS88]



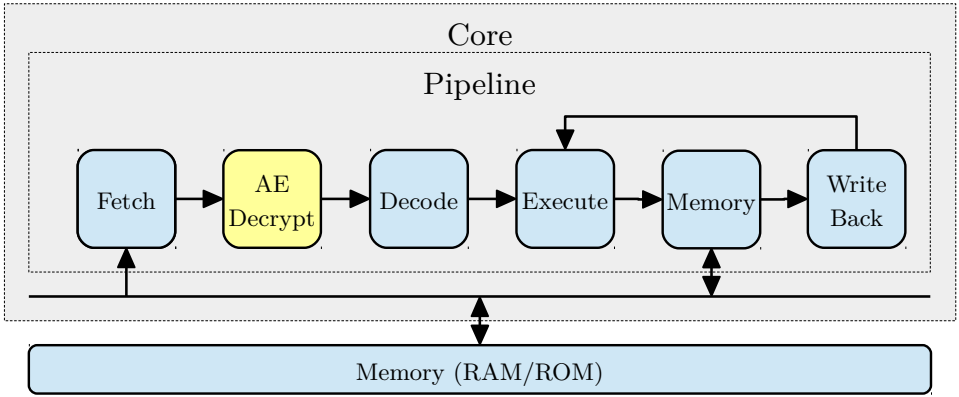
- First known GPSA/CSM implementation
  - Hybrid (HW+SW) scheme
  - LLVM-based toolchain
  - HDL implementation for an ARMv7-M compatible processor
  - 6.4 % hardware overhead
  - 2 % to 71 % runtime overhead
- Analysis and evaluation of signature functions (CRCs, MISRs)
  - CRC-32C: Resistant against at least 7 precise bit flips (injected across two encodings)
  - CSM $h = 4$: Detect a faulty instruction with 99.9 % within 3 cycles (arbitrary fault within encoding)

Sponge-Based Control-Flow Protection for IoT-Devices

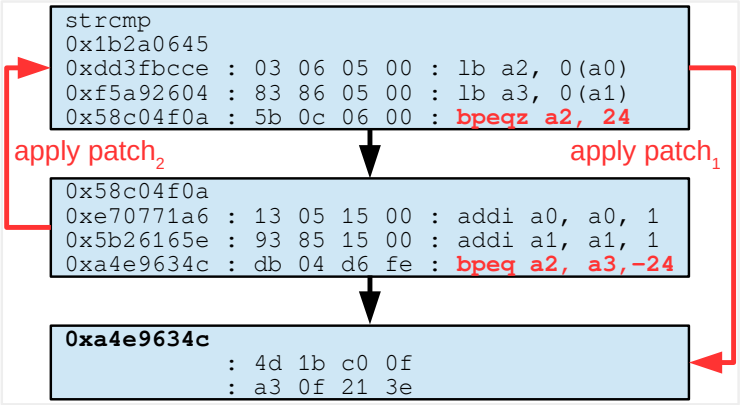
Mario Werner, Thomas Unterluggauer, David Schaffenrath, and Stefan Mangard. “Sponge-Based Control-Flow Protection for IoT Devices”. In: *European Symposium on Security and Privacy – EuroS&P*. **Best Paper Award**. 2018, pp. 214–226. DOI: 10.1109/EuroSP.2018.00023



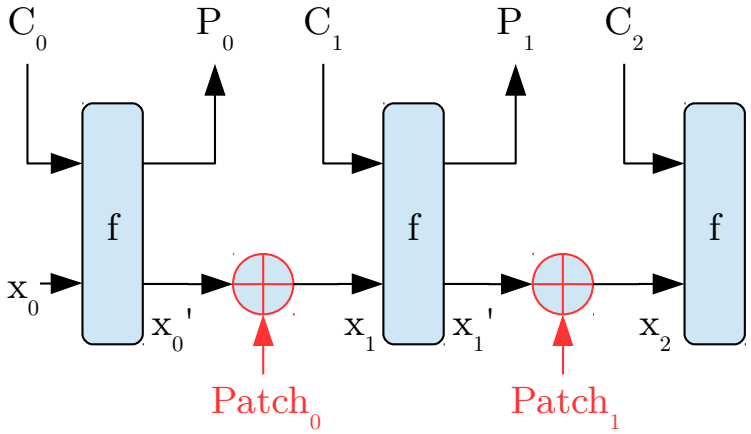
- How can we be sure that the monitor is working?



- Make correctness a requirement for functionality


strcmp
0x1b2a0645
0xdd3fbcce : 03 06 05 00 : lb a2, 0(a0)
0xf5a92604 : 83 86 05 00 : lb a3, 0(a1)
0x58c04f0a : 5b 0c 06 00 : bpeqz a2, 24
apply patch2
apply patch1
0x58c04f0a
0xe70771a6 : 13 05 15 00 : addi a0, a0, 1
0x5b26165e : 93 85 15 00 : addi a1, a1, 1
0xa4e9634c : db 04 d6 fe : bpeq a2, a3,-24
0xa4e9634c
: 4d 1b c0 0f
: a3 0f 21 3e

- Original construction by Andreeva et al. [And+14]



Summary

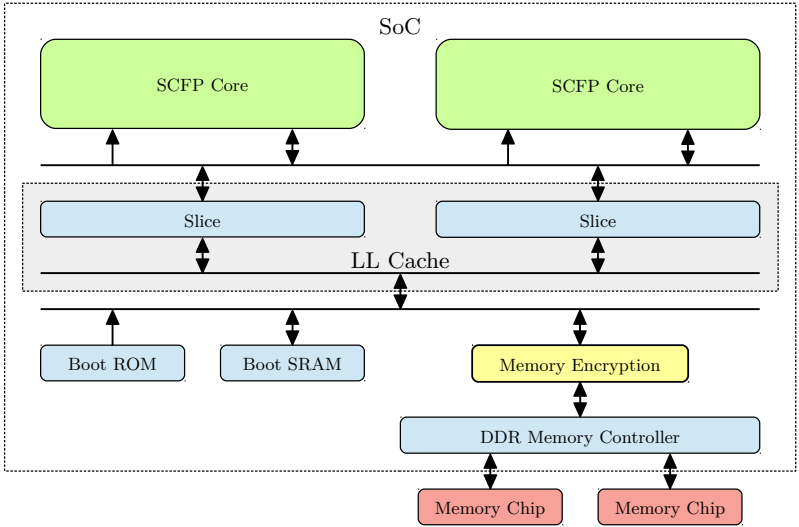
- Sponge-based Control-Flow Protection (SCFP)
  - Hardware-supported CFI scheme
  - Decrypts the instruction stream with instruction granularity
  - Evolution of GPSA and CSM with added confidentiality/authenticity
  - Maintains protection across indirect calls
  - Supports exception/interrupt handling (enables preemptive OSs)
- Presented and analyzed two suitable sponge constructions
- Discussed three SCFP instantiations (IE, AEE, AEE-Light)
- Implemented AEE-Light into a RISC-V processor
  - 9.1 % runtime overhead
  - 19.8 % code size overhead
  - Manufactured in cooperation with ETH-Zurich

- SCFP-like technique with added challenge response protocol
- Idea: Reuse the balancing of the internal crypto state
- Supports all common + a novel attestation mode:
  - Static (arbitrary data/code)
  - Graph (novel approach, attests crypto state)
  - Path (repeated graph attestation)
  - Hybrid (any combination of above the modes)
- Software-only prototype on top of SCFP
- Online licensing extension
  - Verifier provides patch values for prover via license tokens
  - Each token is a patch encrypted with the attestation state

Part II: Counteracting Physical Attacks on the Memory System

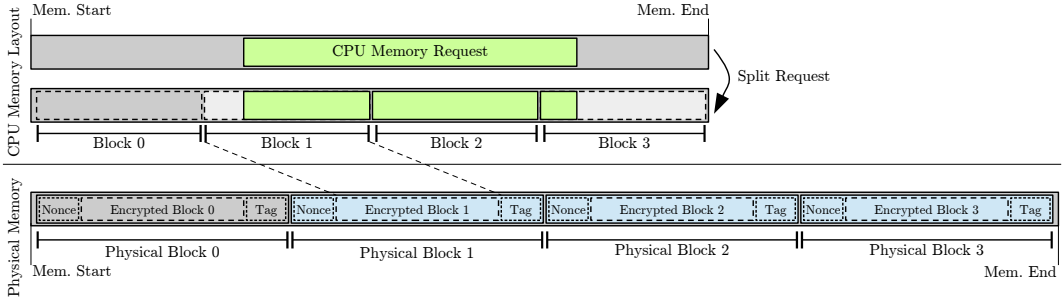
Transparent Memory Encryption and Authentication

Mario Werner, Thomas Unterluggauer, Robert Schilling, David Schaffenrath, and Stefan Mangard. “Transparent memory encryption and authentication”. In: *Field Programmable Logic and Applications – FPL*. 2017, pp. 1–6.
DOI: 10.23919/FPL.2017.8056797

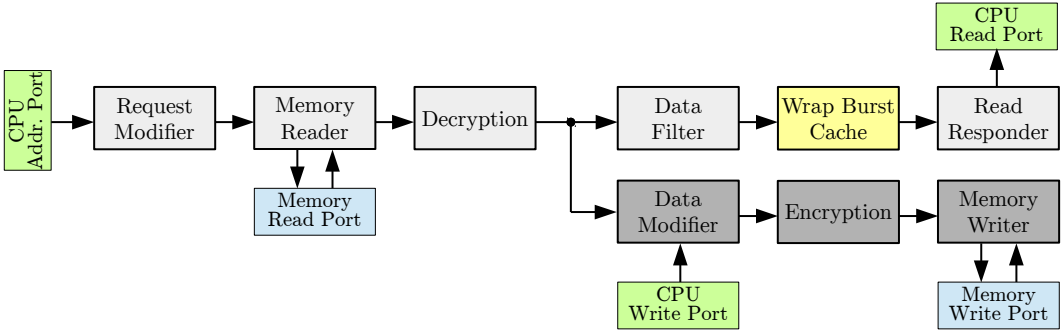

SoC
SCFP Core
SCFP Core
Slice
Slice
LL Cache
Boot ROM
Boot SRAM
Memory Encryption
DDR Memory Controller
Memory Chip
Memory Chip

Main Challenges

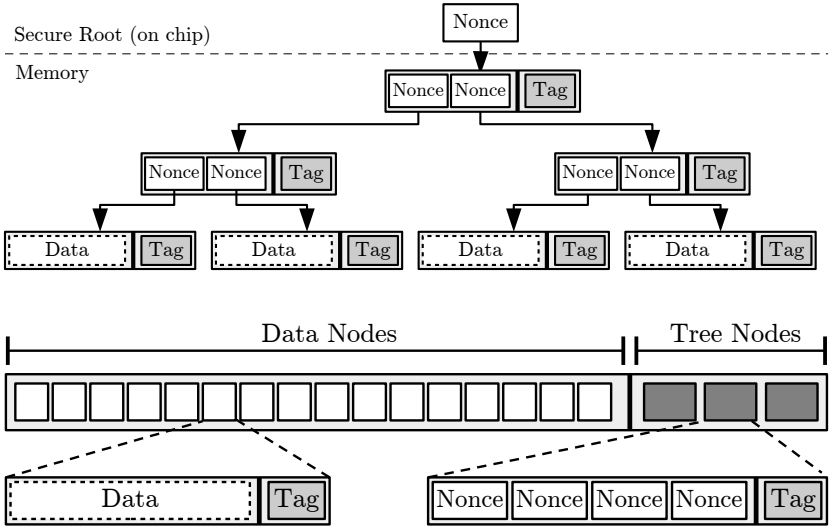
- Transparent translation between two address spaces
- Aligning requests with cipher block/mode sizes
- Extracting/inserting data according to the original request
- Support for interconnect peculiarities (strobes, bursts, . . .)



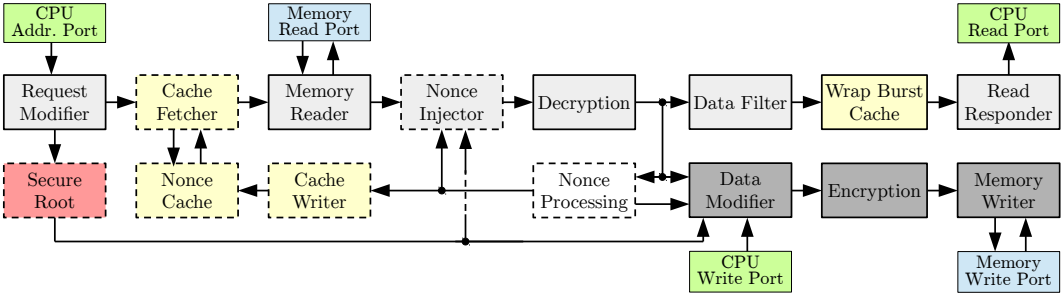
AXI4 Encryption/Authentication Pipeline



- Writes are always RMW
- Supports block wise cipher modes (incl. metadata)
- Optimizes wrapping bursts when reading


Secure Root (on chip)
Memory
Nonce
Nonce
Nonce
Tag
Nonce
Nonce
Tag
Nonce
Nonce
Tag
Data
Tag
Data
Tag
Data
Tag
Data
Tag
Data Nodes
Tree Nodes
Data
Tag
Nonce
Nonce
Nonce
Nonce
Tag

AXI4 Authentication Tree-Pipeline



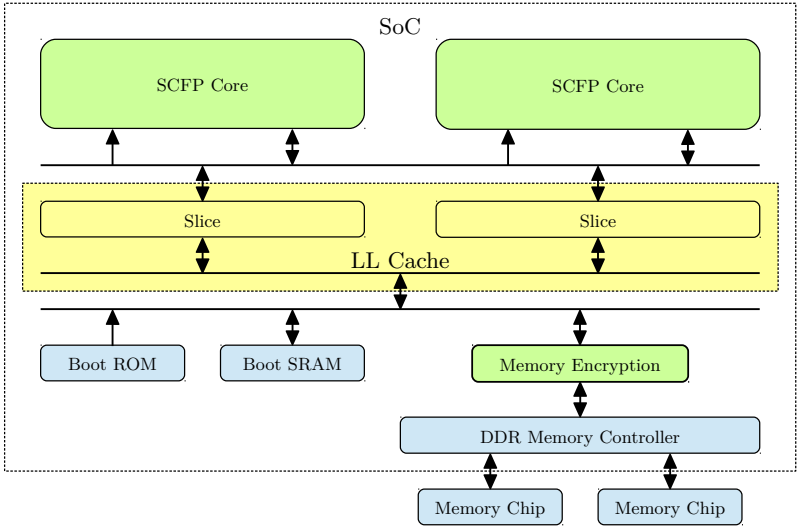
- Writes are always RMW
- Single traversal from the root to the leaf
- Caches for reads, parallel trees for writes
- Arbitrary tree arity

Summary

- Modular VHDL framework for memory encryption/authentication
- Example pipelines for transparent integration
- Supports various ciphers and modes of operation
  - Ciphers: Prince, AES, Ascon
  - Modes: ECB, CBC, XTS, TEC-tree
- AXI4 support (write strobes, narrow transfers, wrapping bursts)
- Tested using an ARM A9 dual-core CPU and Linux as master
- Open source: https://github.com/IAIK/memsec
- Enables follow-up research [UWM19] and projects (https://mitrecyberacademy.org/competitions/ectf-2019/ectf19desc.html)

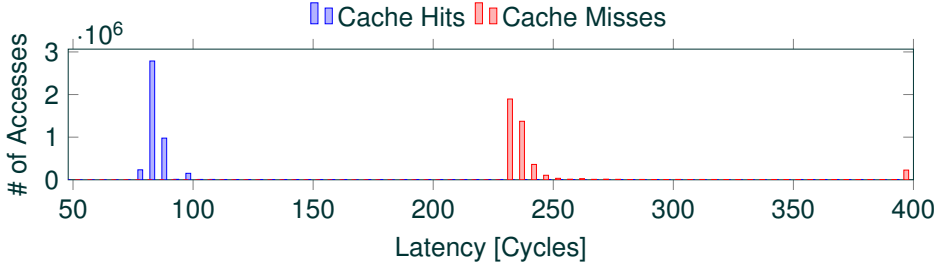
SCATTERCACHE: Thwarting Cache Attacks via Cache Set Randomization

Mario Werner, Thomas Unterluggauer, Lukas Giner, Michael Schwarz, Daniel Gruss, and Stefan Mangard. "ScatterCache: Thwarting Cache Attacks via Cache Set Randomization". In: *USENIX Security Symposium*. 2019, pp. 675–692. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/werner


SoC
SCFP Core
SCFP Core
Slice
Slice
LL Cache
Boot ROM
Boot SRAM
Memory Encryption
DDR Memory Controller
Memory Chip
Memory Chip

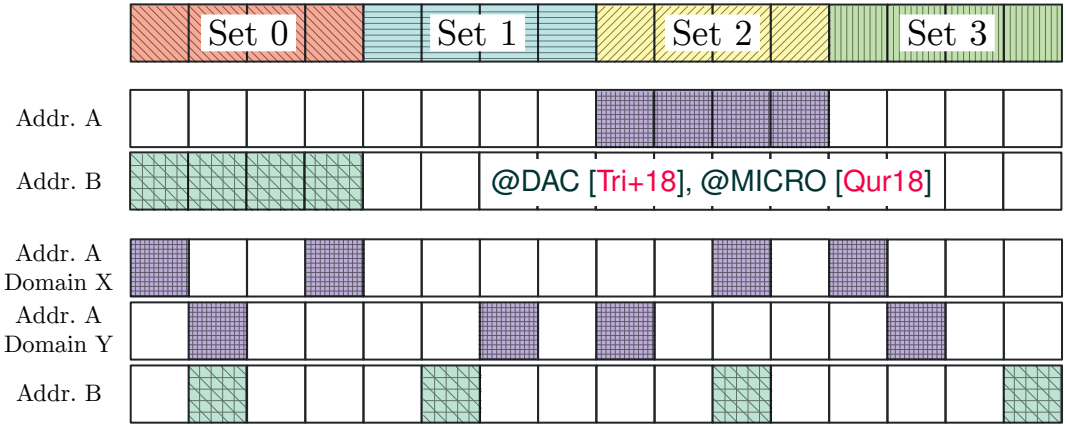
Caches in a Nutshell

- Small but fast memory close to the processor
- Exploit spatial and temporal locality in software to improve performance
- Access to recently used or close-by (*i.e.*, cached) data is faster



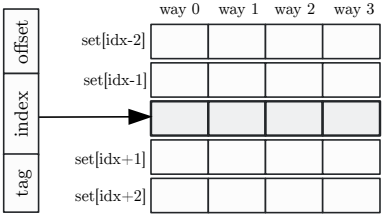
→ Enables cache attacks that infer memory accesses from timing information

generated using the CTA calibration tool [GSM15] on my i5-4200U laptop

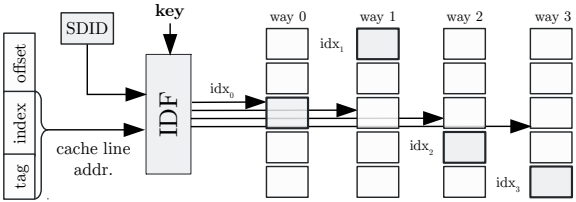

Set 0
Set 1
Set 2
Set 3
Addr. A
Addr. B
@DAC [Tri+18], @MICRO [Qur18]
Addr. A Domain X
Addr. A Domain Y
Addr. B

We want something that is similar in hardware to a traditional cache!

instead of this:



let's do this:



- Low latency cryptographic primitives as index derivation function (IDF)
- Skewed addressing instead of fixed set addressing

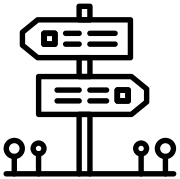
Summary

- ScatterCache combines skewed caches and low latency cryptographic primitives
  - Breaks the fixed link between addresses and cache sets
  - Removes the rigid assignment of cache lines to sets
  - Enables software control over the cache congruencies via SDIDs
- Comparable performance to contemporary caches
- Harder to attack even in strong attack models
- Attacks are probabilistic and demand new approaches
- Still, more analysis is required in more realistic models

Conclusion

Conclusion

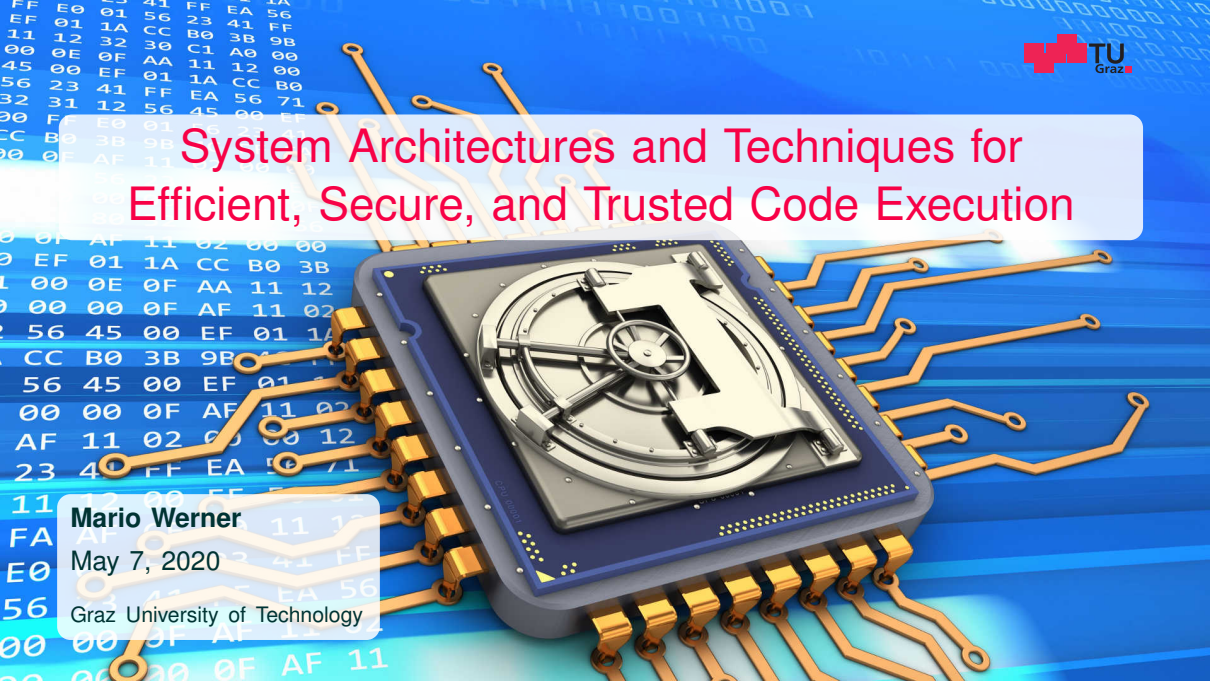
- Techniques to protect code execution against physical attacks
  - Control-Flow Integrity: GPSA, SCFP, remote attestation and licensing
  - Memory System: MEMSEC, SCATTERCACHE
- PoC implementations hint reasonable costs and good performance
- Fully integrated into the toolchain for good usability
- Highly configurable concepts with huge design space
- Foundation for follow-up research (e.g., SCFP, MEMSEC, SCATTERCACHE)



Potential topics for follow-up research:

- Hardware supported protection of on-chip data/addresses
- Side-channel evaluation of SCFP
- Development of custom cryptographic primitives
  (low latency, configurable block size)
- Quantifying the security of “secure” caches against real attacks
- Evaluation of our concepts on larger processors

System Architectures and Techniques for Efficient, Secure, and Trusted Code Execution



Mario Werner

May 7, 2020

Graz University of Technology

[Gro+16] Hannes Gross, Manuel Jelinek, Stefan Mangard, Thomas Unterluggauer, and **Mario Werner**. "Concealing Secrets in Embedded Processors Designs". In: *Smart Card Research and Advanced Applications – CARDIS*. 2016, pp. 89–104. DOI: 10.1007/978-3-319-54669-8_6.

[Kal+20] Daniel Kales, Sebastian Ramacher, Christian Rechberger, Roman Walch, and **Mario Werner**. "Efficient FPGA Implementations of LowMC and Picnic". In: *The Cryptographers' Track at the RSA Conference – CT-RSA*. 2020. URL: https://eprint.iacr.org/2019/1368.

[Sch+18] Robert Schilling, **Mario Werner**, Pascal Nasahl, and Stefan Mangard. "Pointing in the Right Direction - Securing Memory Accesses in a Faulty World". In: *Annual Computer Security Applications Conference – ACSAC*. 2018, pp. 595–604. DOI: 10.1145/3274694.3274728.

[SWM18] Robert Schilling, **Mario Werner**, and Stefan Mangard. "Securing conditional branches in the presence of fault attacks". In: *Design, Automation & Test in Europe – DATE*. 2018, pp. 1586–1591. DOI: 10.23919/DATE.2018.8342268.

[UWM17a] Thomas Unterluggauer, **Mario Werner**, and Stefan Mangard. "Securing Memory Encryption and Authentication Against Side-Channel Attacks Using Unprotected Primitives". In: *Conference on Computer and Communications Security – CCS*. 2017, pp. 690–702. DOI: 10.1145/3052973.3052985.

[UWM17b] Thomas Unterluggauer, **Mario Werner**, and Stefan Mangard. "Side-channel plaintext-recovery attacks on leakage-resilient encryption". In: *Design, Automation & Test in Europe – DATE*. 2017, pp. 1318–1323. DOI: 10.23919/DATE.2017.7927197.

[UWM19] Thomas Unterluggauer, **Mario Werner**, and Stefan Mangard. "MEAS: memory encryption and authentication secure against side-channel attacks". In: *J. Cryptographic Engineering* 9 (2019), pp. 137–158. DOI: 10.1007/s13389-018-0180-2.

[Wei+19] Samuel Weiser, **Mario Werner**, Ferdinand Brasser, Maja Malenko, Stefan Mangard, and Ahmad-Reza Sadeghi. "TIMBER-V: Tag-Isolated Memory Bringing Fine-grained Enclaves to RISC-V". In: *Network and Distributed System Security Symposium – NDSS*. 2019. URL: https://www.ndss-symposium.org/ndss-paper/timber-v-tag-isolated-memory-bringing-fine-grained-enclaves-to-risc-v/.

[Wer+17] **Mario Werner**, Thomas Unterluggauer, Robert Schilling, David Schaffenrath, and Stefan Mangard. "Transparent memory encryption and authentication". In: *Field Programmable Logic and Applications – FPL*. 2017, pp. 1–6. DOI: 10.23919/FPL.2017.8056797.

[Wer+18] **Mario Werner**, Thomas Unterluggauer, David Schaffenrath, and Stefan Mangard. "Sponge-Based Control-Flow Protection for IoT Devices". In: *European Symposium on Security and Privacy – EuroS&P*. **Best Paper Award**. 2018, pp. 214–226. DOI: 10.1109/EuroSP.2018.00023.

[Wer+19a] **Mario Werner**, Robert Schilling, Thomas Unterluggauer, and Stefan Mangard. "Protecting RISC-V Processors against Physical Attacks". In: *Design, Automation & Test in Europe – DATE*. 2019, pp. 1136–1141. DOI: 10.23919/DATE.2019.8714811.

[Wer+19b] **Mario Werner**, Thomas Unterluggauer, Lukas Giner, Michael Schwarz, Daniel Gruss, and Stefan Mangard. "ScatterCache: Thwarting Cache Attacks via Cache Set Randomization". In: *USENIX Security Symposium*. 2019, pp. 675–692. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/werner.

[WUW13] Erich Wenger, Thomas Unterluggauer, and **Mario Werner**. "8/16/32 Shades of Elliptic Curve Cryptography on Embedded Processors". In: *Progress in Cryptology – INDOCRYPT*. 2013, pp. 244–261. DOI: 10.1007/978-3-319-03515-4_16.

[WW11] Erich Wenger and **Mario Werner**. "Evaluating 16-Bit Processors for Elliptic Curve Cryptography". In: *Smart Card Research and Advanced Applications – CARDIS*. 2011, pp. 166–181. DOI: 10.1007/978-3-642-27257-8_11.

[WW17] Samuel Weiser and **Mario Werner**. "SGXIO: Generic Trusted I/O Path for Intel SGX". In: *Conference on Data and Application Security and Privacy – CODASPY*. 2017, pp. 261–268. DOI: 10.1145/3029806.3029822.

[WWM15] **Mario Werner**, Erich Wenger, and Stefan Mangard. "Protecting the Control Flow of Embedded Processors against Fault Attacks". In: *Smart Card Research and Advanced Applications – CARDIS*. 2015, pp. 161–176. DOI: 10.1007/978-3-319-31271-2_10.

[And+14] Elena Andreeva, Begül Bilgin, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. “APE: Authenticated Permutation-Based Encryption for Lightweight Cryptography”. In: *Fast Software Encryption – FSE*. 2014, pp. 168–186. DOI: 10.1007/978-3-662-46706-0_9.

[GSM15] Daniel Gruss, Raphael Spreitzer, and Stefan Mangard. *Cache Template Attacks Repository*. 2015. URL: https://github.com/IAIK/cache_template_attacks (visited on 05/05/2020).

[Kim+14] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji-Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. “Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors”. In: *International Symposium on Computer Architecture – ISCA*. 2014, pp. 361–372. DOI: 10.1109/ISCA.2014.6853210.

[KSV13] Dusko Karaklajic, Jörn-Marc Schmidt, and Ingrid Verbauwhede. “Hardware Designer’s Guide to Fault Attacks”. In: *IEEE Trans. Very Large Scale Integr. Syst.* 21 (2013), pp. 2295–2306. DOI: 10.1109/TVLSI.2012.2231707.

[MM88] Aamer Mahmood and Edward J. McCluskey. “Concurrent Error Detection Using Watchdog Processors - A Survey”. In: *IEEE Trans. Computers* 37 (1988), pp. 160–174. DOI: 10.1109/12.2145.

References ii

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007. ISBN: 978-0-387-30857-9.

[Mur+20] Kit Murdock, David Oswald, Flavio D. Garcia, Jo Van Bulck, Daniel Gruss, and Frank Piessens. “Plundervolt: Software-based Fault Injection Attacks against Intel SGX”. In: *IEEE Symposium on Security and Privacy – S&P*. 2020. URL: https://www.plundervolt.com/doc/plundervolt.pdf.

[Qur18] Moinuddin K. Qureshi. “CEASER: Mitigating Conflict-Based Cache Attacks via Encrypted-Address and Remapping”. In: *IEEE/ACM International Symposium on Microarchitecture – MICRO*. 2018, pp. 775–787. DOI: 10.1109/MICRO.2018.00068.

[Tri+18] David Trilla, Carles Hernández, Jaume Abella, and Francisco J. Cazorla. “Cache side-channel attacks and time-predictability in high-performance critical real-time systems”. In: *Design Automation Conference – DAC*. 2018, 98:1–98:6. DOI: 10.1145/3195970.3196003.

[WS88] Kent D. Wilken and John Paul Shen. “Continuous Signature Monitoring: Efficient Concurrent-Detection of Processor Control Errors”. In: *International Test Conference – ITC*. 1988, pp. 914–925. DOI: 10.1109/TEST.1988.207880.

Image/Icon Credits

- “Checklist” icon by Philipp Petzka from the Noun Project.
- “direction” icon by DinosoftLab from the Noun Project.
- “lightbulb” icon by Maxim Kulikov from the Noun Project.
- “presentation” icon by Shmidt Sergey from the Noun Project.
- “Question” icon by andrewcaliber from the Noun Project.
- “car” image by bolala kido from Pixabay.
- “green-light” image by Clker-Free-Vector-Images from Pixabay.
- “laptop” image by Clker-Free-Vector-Images from Pixabay.
- “router” image by Clker-Free-Vector-Images from Pixabay.
- “smartphone” image by 1117826-1117826 from Pixabay.
- “surveillance” image by OpenClipart-Vectors from Pixabay.