Metis: An Integrated Morphing Engine CPU to Protect against Side Channel Attacks

Francesco Antognazza, Alessandro Barenghi, Gerardo Pelosi

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milano, Italy

Background and Problem Statement

- Side channel attacks are a concrete threat against the security of computing systems
- Every device leaks confidential information with the environment
- Countermeasures mitigate those attacks through various techniques:
  - hiding: increasing noise in the leaking channel
  - masking: randomly split the secret information into several shares
  - (code) morphing: dynamically varying the leakage model
- Code morphing hinders the ability to model the side channel behavior of the device through randomizing how the sensitive computation itself is performed
- It is still necessary to protect all data transfers to and from memory by means of a masking countermeasure

Metis: A Code Morphing CPU

Metis is the first architectural design providing transparent code morphing support:

- It is implemented on top of the OpenTitan System on Chip (SoC)
- We realize code morphing at ISA level at runtime
- The morphing is performed by the ID-&-EX stage of the pipeline
- Automatized management of sensitive data masked in multiple shares
- Maintains compatibility with OpenOCD and GDB debugging tools

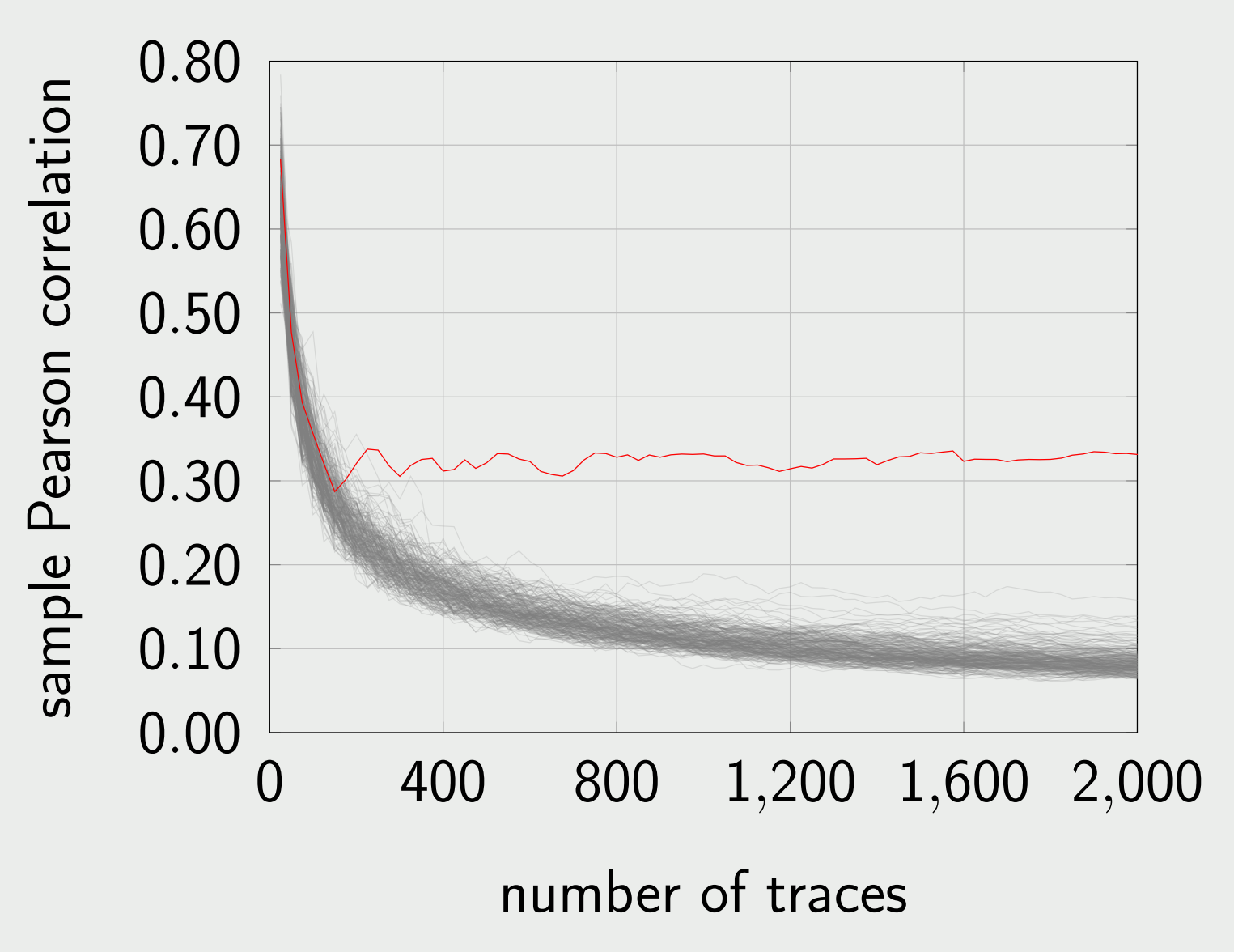
The Modified Microarchitecture

- **Controller** Manages the pipeline stalls required for transparent code morphing due to the processing either the tile of an instruction to be morphed or a masked load/store operation
- **Instruction Fetch** Adds a combinatorial path which allows the Decoder in the ID-&-EX stage to establish if the instruction currently in the IF stage is going to be morphed or not
- **Control Status Register** To allow the code morphing actions to be selectively enabled, the $\mathrm{MSTATUS}$ register in the CSR module includes a MOR enable bit
- **Morphing Register File** To avoid possible register clobbering, we include a Morphing Register File (MRF), which provides temporary registers to de-normalize the tile at hand and a word from a RNG.
- **The Tile Memory** Dedicated storage for the instruction tile which is directly accessed to minimize latencies. The overall organization of the tile memory exhibits n groups of rows, each of which corresponds to a different tile that in turn is composed by at most m RISC-V instructions
- **Physical Memory Protection** Memory areas are augmented with an additional MASKED flag along with RWX standard ones
- **Load Store Unit** When accessing data in masked memory areas the pipeline is stalled and the load of the corresponding mask is emitted, ensuring to flush the bus buffers before the transfer. Masks are refreshed with a configurable probability during load operations.

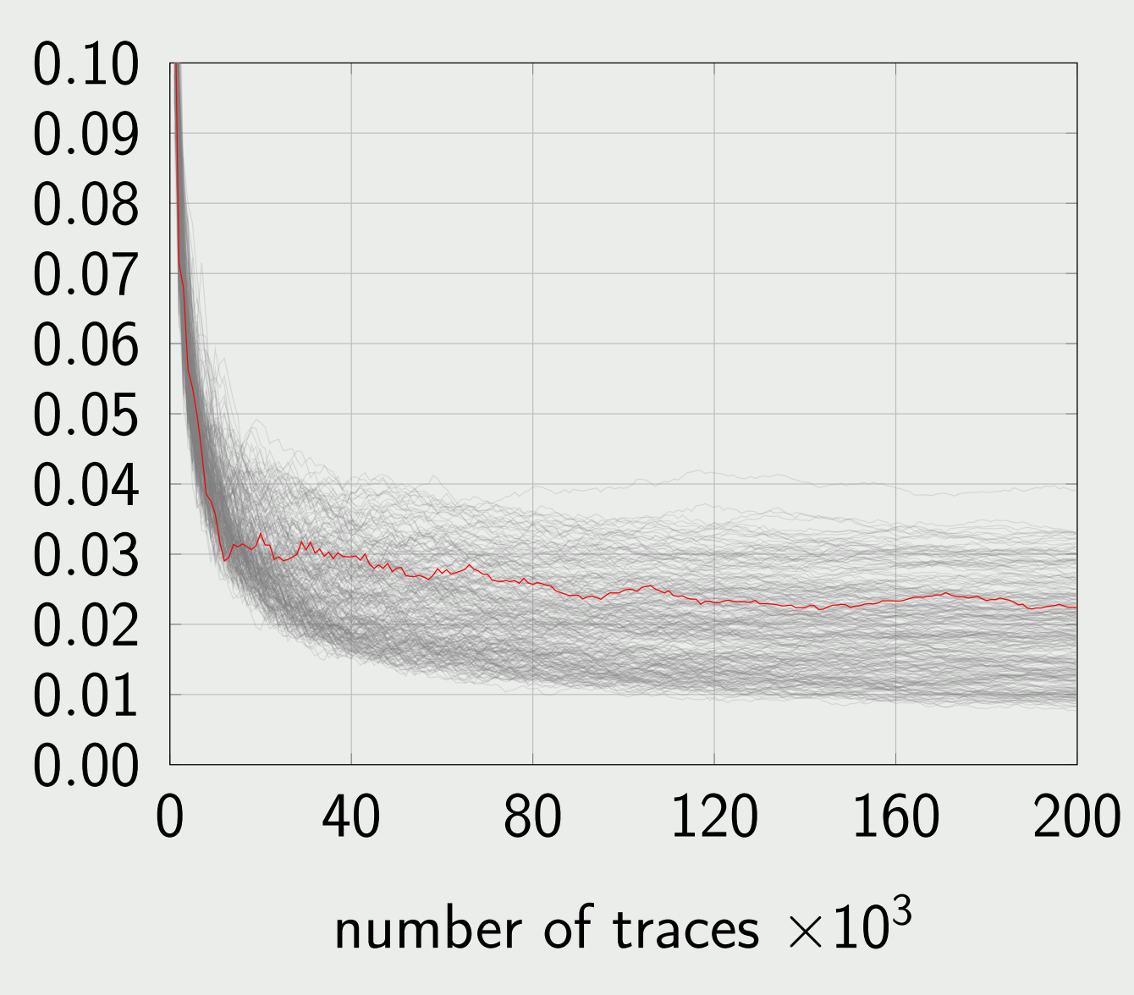
Electromagnetic Side Channel Characterization

We considered two variants of the Metis SoC, endowed with different morphing tiles modifying the computation of the `xor`, `and` and `slli` instructions:

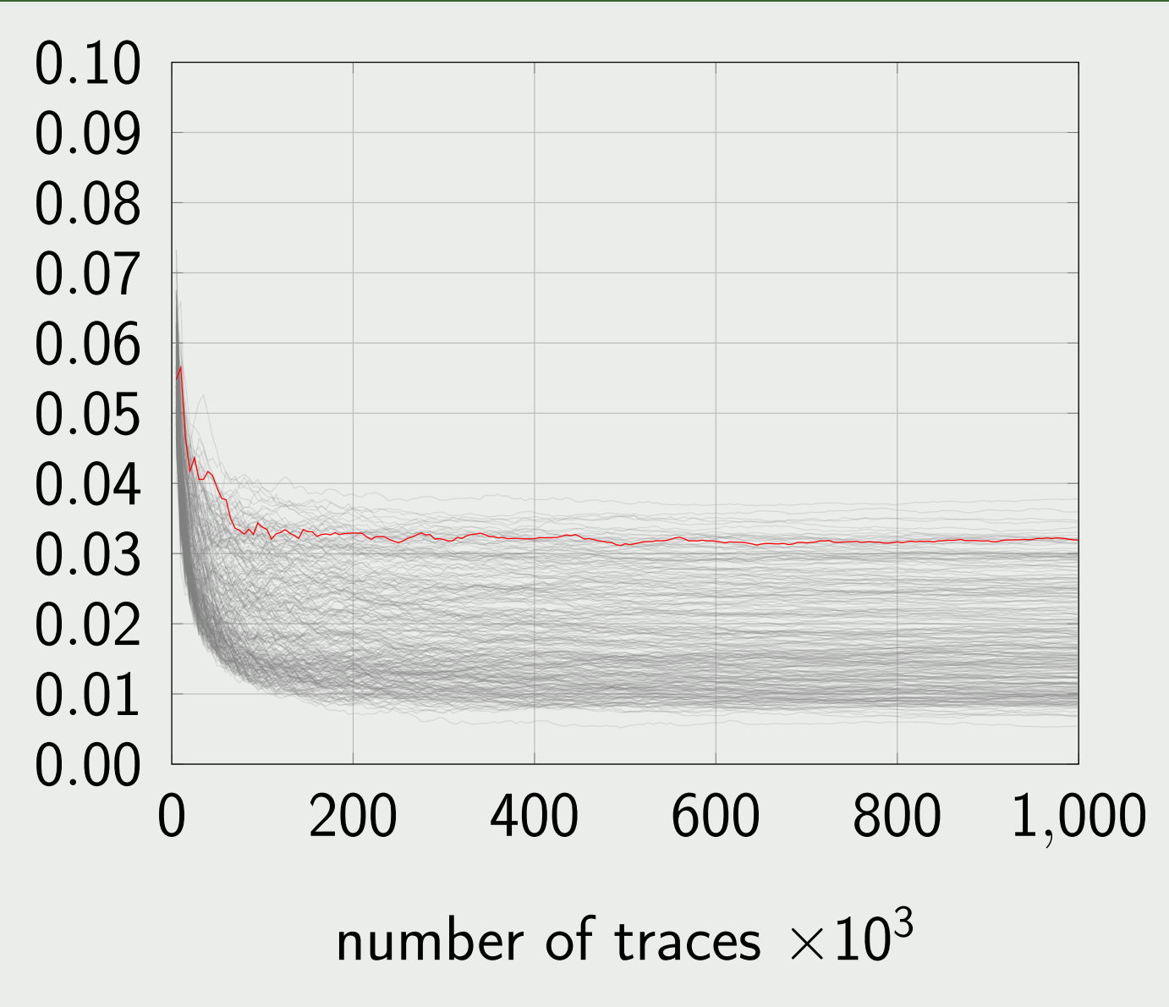
- **Shuffling Tiles (ST)** contains two variants for each of the aforementioned instructions constituted by the original instruction itself and up to 3 dummy operations
- **Morphing Tiles (MT)** contains 8 alternative tiles for each substituted instruction, each alternative tile contains from 0 to 4 dummy normalized instructions with semantically-equivalent outcomes.



(a) Ibex (Unprotected)



(b) Metis-ST (Protected)



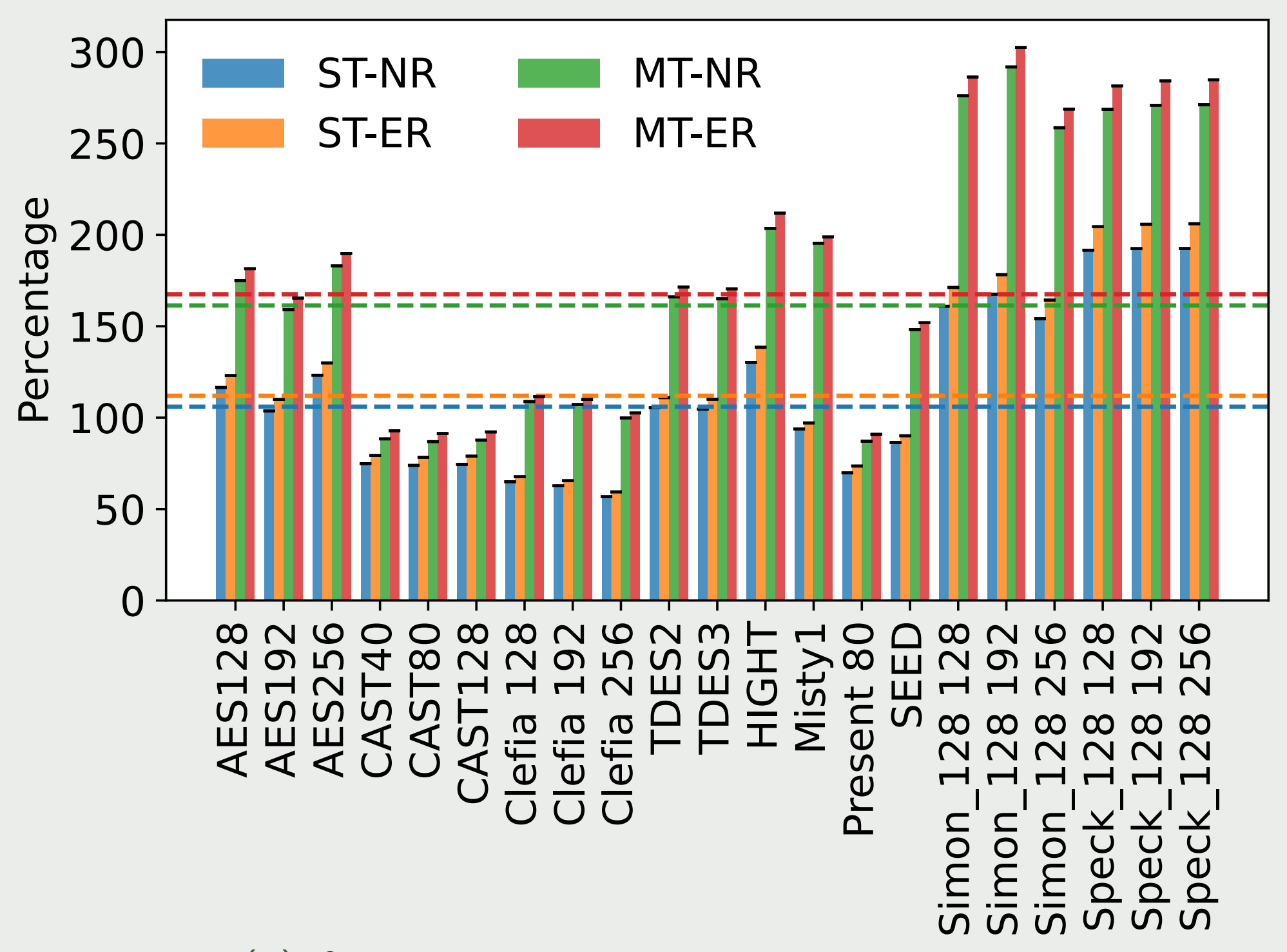
(c) Metis-MT (Protected)

Figure: Results of performing a CPA against AES-128, targeting the computation of the SubBytes primitive on both the unprotected Ibex and on Metis. The sample correlation of the correct key hypothesis depicted in red, while the sample correlation of all the other key guesses is reported in grey.

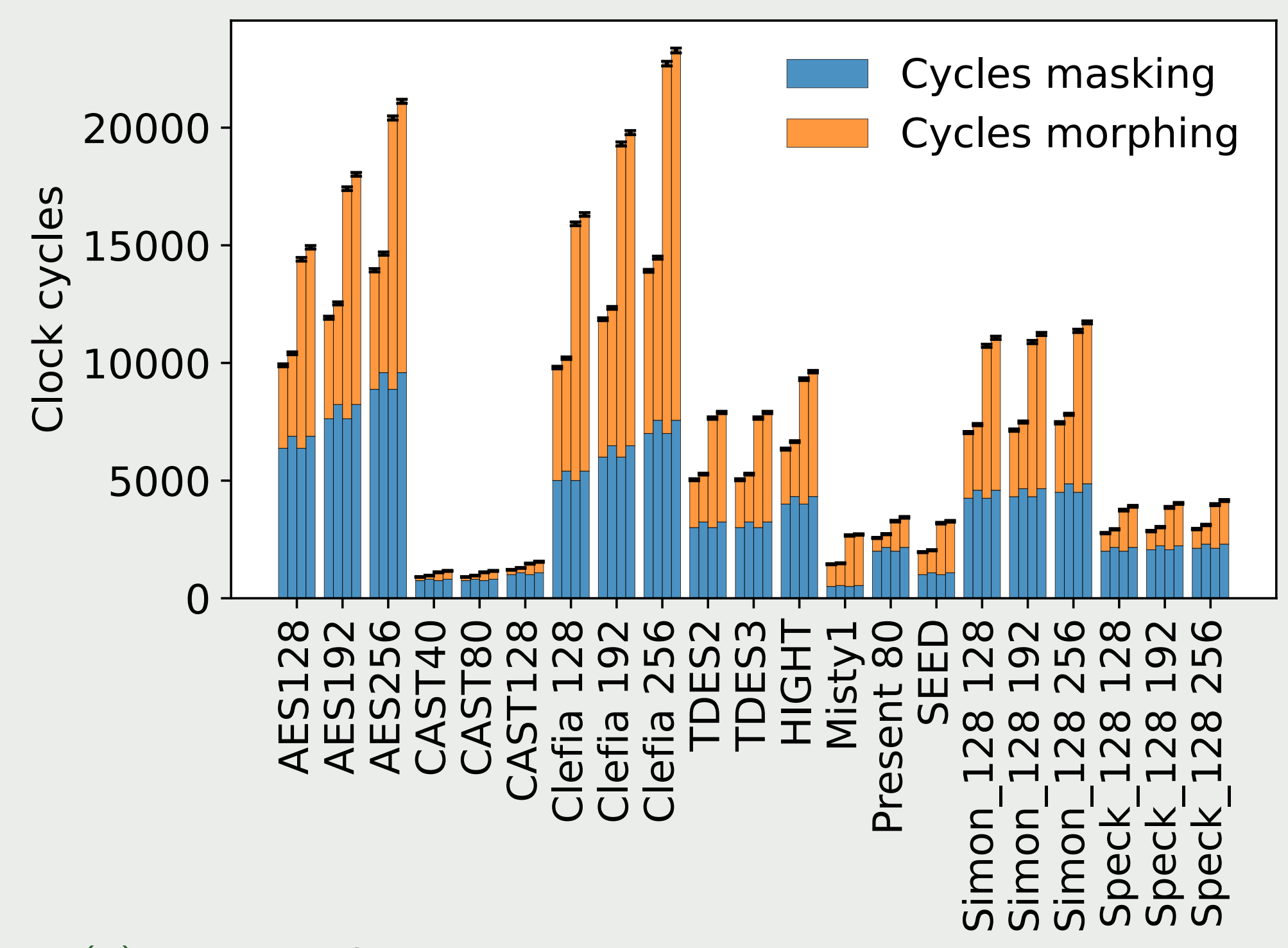
Metis protected solutions successfully prevent the recovery of the correct key value when 200k and 1M traces are employed for the ST and MT tile-sets, respectively. This represents an $>250\times$ and a $>1,250\times$ improvement in the MTD metric w.r.t. a correlation power analysis targeting an unprotected design.

Performance Assessment

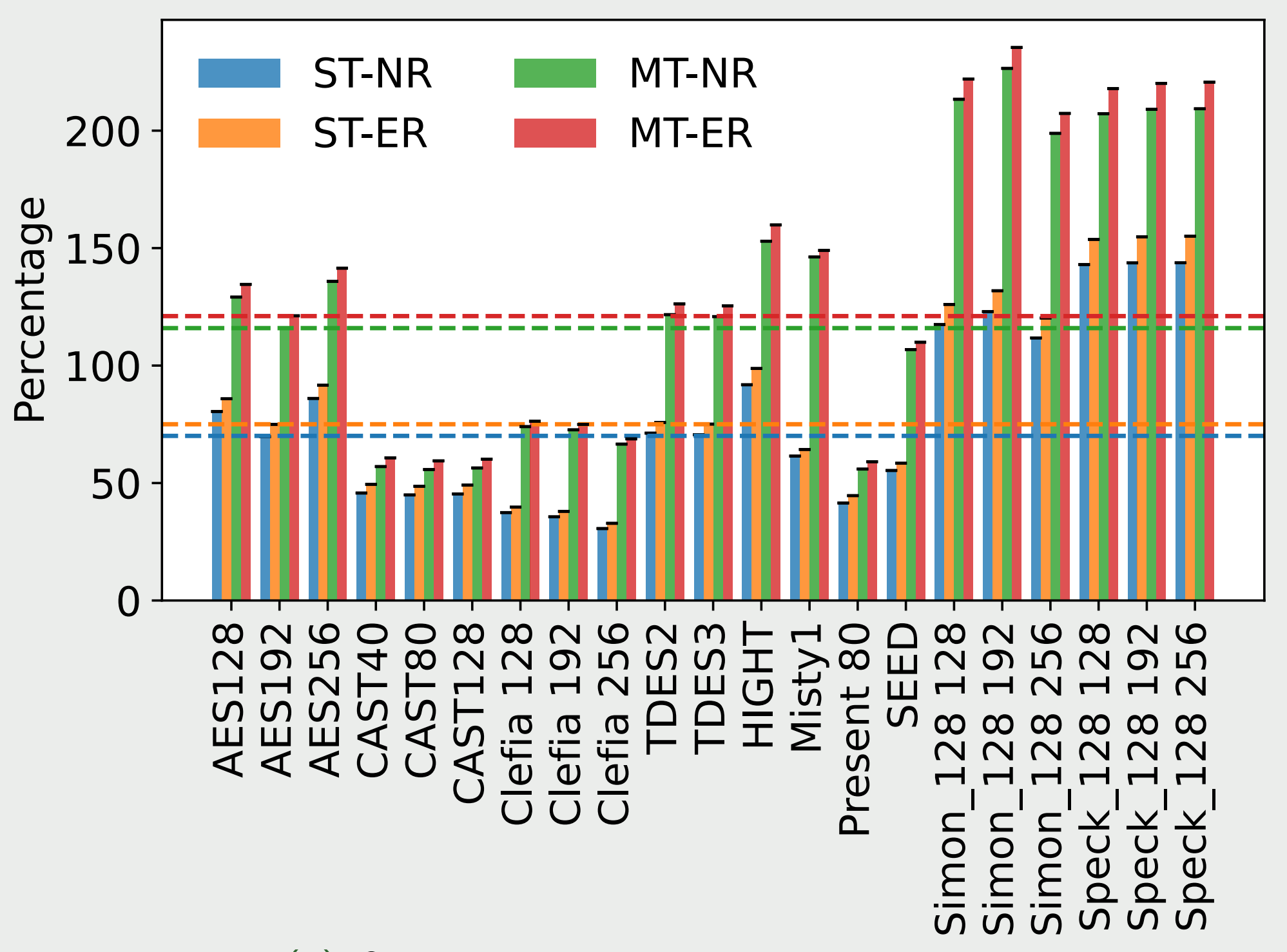
- To provide a fair evaluation of the integrated code morphing overhead on a practically relevant workload, we evaluated Metis on all the available ISO/IEC standard symmetric block ciphers, namely: ISO/IEC 18033-3 wide-block ciphers AES and SEED, and narrow-block ciphers: Triple DES-EDE, CAST, HIGHT and Misty1; ISO/IEC 29192-2 lightweight ciphers: Clefia and Present; ISO/IEC 29167-21 Speck and ISO/IEC 29167-22 Simon lightweight ciphers.
- We enable the code morphing in Metis through setting the morphing enable bit in the $\mathrm{MSTATUS}$ status register, before the start of the cipher execution and we measured performance with two different mask refreshing probabilities: 0.25 for Normal Refreshing (NR), and 0.75 for Extended Refreshing (ER).
- The overall execution times of the ISO standard cipher suite have an overhead of 1.05× when considering ST, and 1.61× when considering MT, and a mask refresh probability of 0.25, retaining the similarly small extra overhead ($\approx 6\%$) for a mask refreshing with probability 0.75.



(a) Overhead measured on execution time



(b) Number of clock cycles spent in morphing and masking



(c) Overhead measured on wall clock

Figure: Performance evaluation of the Metis SoC, comparing the results obtained on the ST and MT tiles, NR and ER refreshing strategies to the performance of the Ibex SoC. All dashed lines represent the geometric means of the results in the corresponding plot. Figure (a) reports the overheads in execution time taking into account the difference in the clock rate of the two SoCs: OpenTitan (48 MHz), Metis (40 MHz). Figure (b) reports the number of clock cycles spent in the morphing and masking computations by Metis. Figure (c) measures the overhead of Metis with respect to the plain OpenTitan when considering the wall clock.

Further details are available in the IEEE Access journal publication - vol. 9 pp.69210-69225 - DOI:10.1109/ACCESS.2021.3077977