Memory Authenticated Encryption Engine for a RISC-V processor

KARIM AIT LAHSSAINE
UNIV. GRENOBLE ALPES
CEA, LETI
F-38000 GRENOBLE, FRANCE
KARIM.AITLAHSSAINE@CEA.FR

OLIVIER SAVRY
UNIV. GRENOBLE ALPES
CEA, LETI
F-38000 GRENOBLE, FRANCE
OLIVIER.SAVRY@CEA.FR

Context

Attacks on microprocessors are on the increase, targeting critical modules of the architecture. Among these modules, the memory and its bus are vulnerable to attacks, to secure them, the countermeasure must guarantee :

- **Confidentiality against side-channel attacks**
- **Integrity against Rowhammer and fault injection attacks**

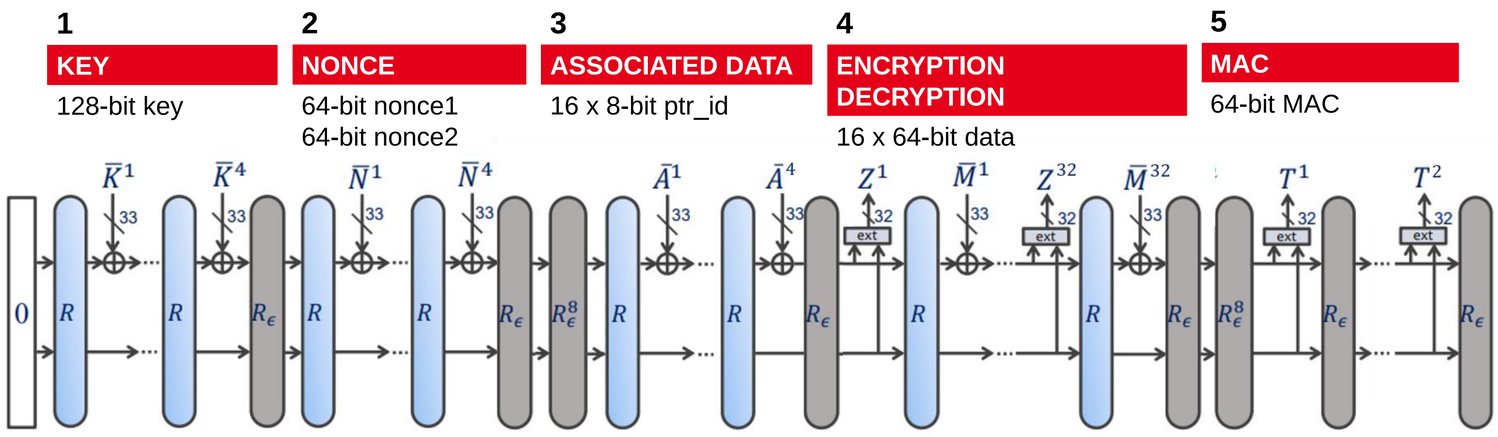
However, this countermeasure must limit the performance degradation of the host system, such as latency or memory and logic footprint.
The hardware countermeasure Memory Authenticated Encryption Engine (MAEE) was designed on the basis of these observations.

Authenticated Encryption

Countermeasures exist, but as shown by [1], most encrypt word by word and associate a MAC (Message Authentication Code) with each word, which is prohibitive in terms of memory footprint. To limit the cost, authenticated encryption algorithms are the preferred solution, because in addition to encrypting the data per chunk, they associate a MAC with the entire chunk.

Subterranean

After studying the algorithms and their benchmark[2], we chose Subterranean 2.0[3], for its small logical footprint, its high data rate and the absence of security holes in the state of the art. The Subterranean algorithm is made up of several rounds, and between them data is absorbed or extracted.

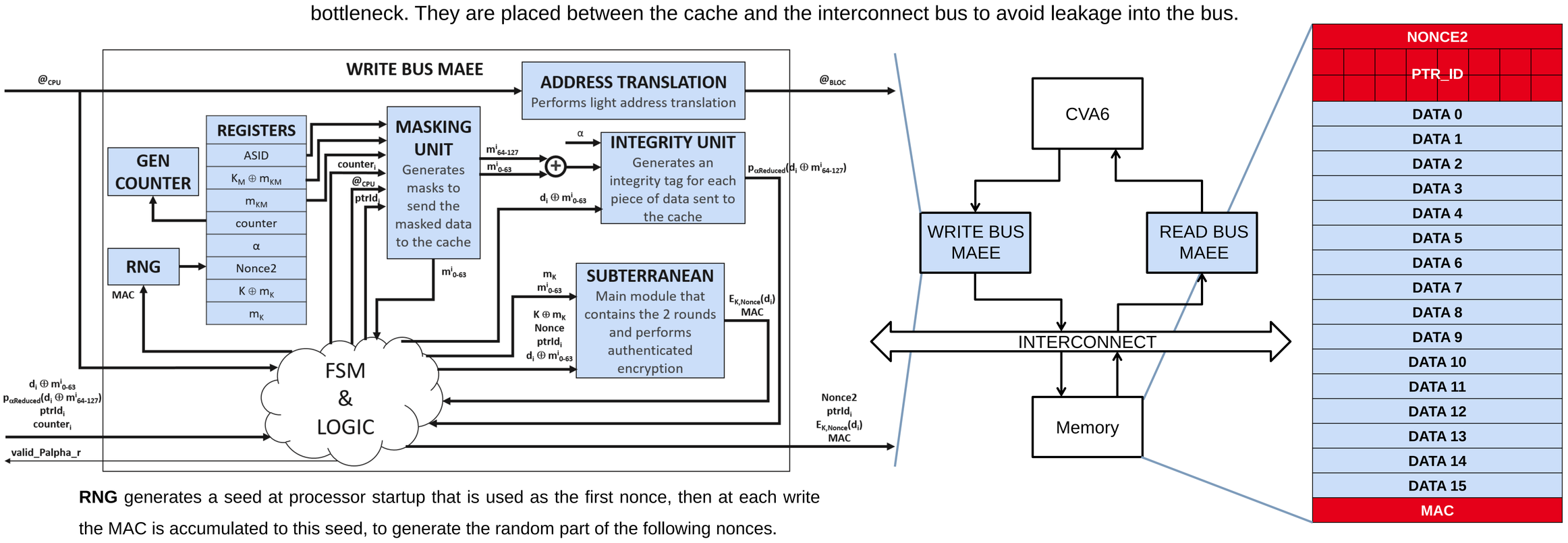


The nonce is composed of the address, nonce1, and a random part updated at each write, nonce2. Nonce1 prevents block swapping.

The encryption takes 40 cycles, and as we choose a chunk size of 160 bytes, with 128 bytes of data (16x64 bits), we have a throughput of 3.2 bytes per cycle.

Implementation

Two MAEEs are implemented on Genesys2 FPGA board, associated with the RISC-V CVA6 core. One for the write bus and one for the read bus, to avoid introducing a bottleneck. They are placed between the cache and the interconnect bus to avoid leakage into the bus.



RNG generates a seed at processor startup that is used as the first nonce, then at each write the MAC is accumulated to this seed, to generate the random part of the following nonces.

Results

After emulating the SoC (CVA6 + MAEE + Peripherals including memory) on FPGA, and running a Linux kernel, we obtain the following performances.

For throughput, the results were obtained with the RAMspeed benchmark on Linux (INTmark with 1Gbytes per pass)[4].

	SoC LUTs	SoC FFs	DRAM size	Linux Boot Time	Read Rate	Write Rate
Without MAEEs	86743	53593	819 MB	2 min 57 s	56.26 MB/s	46.87 MB/s
With MAEEs	90889	54840	1 GB	3 min 17 s	50.68 MB/s	37.70 MB/s
Overhead %	**4.78**	**2.34**	**25**	**9.7**	**-9.92**	**-19.56**

Références

1 Pascal Nasahl et al. “CrypTag: Thwarting Physical and Logical Memory Vulnerabilities Using Cryptographically Colored Memory”.
2 Kamyar Mohajerani et al. “FPGA Benchmarking of Round 2 Candidates in the NIST Lightweight Cryptography Standardization Process: Methodology, Metrics, Tools, and Results”.
3 Joan Daemen et al. “The Subterranean 2.0 Cipher Suite”. In: IACR Transactions on Symmetric Cryptology 2020.S1 (June 2020), pp. 262–294.
4 Alasir. RAMspeed, a cache and memory benchmarking tool. v2.6.0. 2009. url: https://github.com/cruvolo/ramspeed.

OUR ABSTRACT HERE !