Proposal for Presentation

at *Lightweight Cryptography Workshop 2019*

Will the Future Lightweight Standard be RISC-V Friendly?

Gorkem Nisanci
Eastern Mediterranean University
17330148@students.emu.edu.tr

Dr Tolga Yalcin
Northern Arizona University
tolga.yalcin@nau.edu
(participation confirmed)

Dr Elif Bilge Kavun
The University of Sheffield
e.kavun@sheffield.ac.uk

Creators' motto was "Instruction sets should be free!" when they started development of today's highly popular open source RISC-V instruction set architecture (ISA) [1]. They had both Internet of Things (IoT) and warehouse scale computers (WSC) in mind, and came up with four key requirements:

- *Base-plus-extension ISA:* Flexibility to add application-specific accelerators and supporting instructions is vital in both IoT and WSC.
- *Compact instruction set decoding:* Especially for IoT, smaller the code size leads to smaller memory, lower power, and cost and resource effective systems.
- *Support for multiple precision:* Several WSC already rely on quadruple precision in addition to single and double precision.
- *Multiple addressing modes:* While 32-bit addressing may be sufficient for IoT, WSC rely on 64-bit addressing and expected to be needing more than 64 bits within a decade or so.

It has already been a long way since those days. Within the "half a decade" since its debut, RISC-V has managed to form its own ecosystem in an unprecedented speed [2]. Countless open source implementations already exist with supporting development tools [3]. It will not be too much of an exaggeration to say that the creators have already reached their target. Similarly, it will not be too much to expect even more popularity of it within the future. It is already considered by most people as the "de facto standard" for open-hardware computing platforms for IoT and embedded domains.

We believe that it is only prudent to investigate the suitability of the candidates of the NIST lightweight crypto standardization process. We have therefore started a study where we implement the LW candidate algorithms on RISC-V using optimized assembly coding, and compare their performance in terms of processing speed and memory requirements. In our study, we focus on the standard and embedded 32-bit versions of RISC-V – RV32I and RV32E respectively. For completeness, we will also implement standard encryption and authenticated encryption algorithms and compare their figures with those of LW candidates.

Currently, we are considering “Round 1” candidates in our study. However, in later stages, we will focus on “Round 2” candidates, which are expected to be announced by the end of August 2019. We will also analyze the instruction usage of the implemented algorithms and will present our proposals for possible “lightweight crypto” instruction set extensions.

References:

[1] K. Asanović, D. A. Patterson; Instruction Sets Should Be Free: The Case For RISC-V; Technical Report No. UCB/EECS-2014-146; UC Berkeley; 2014.

[2] J. Sanders; CHIPS Alliance aims to ease RISC-V design and deployment; https://www.techrepublic.com/article/chips-alliance-aims-to-ease-risc-v-design-and-deployment/; 2019.

[3] RISC V Cores and SoCs; https://github.com/riscv/riscv-wiki/wiki/RISC-V-Cores-and-SoCs; 2019.