Enhancing CAESAR Hardware API Support for Lightweight Architectures

Panasayya Yalla Jens-Peter Kaps

George Mason University, Fairfax, VA, USA

Abstract

The Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR), aimed at developing a portfolio of new-generation authenticated ciphers surpassing the capabilities of current standards, such as AES-GCM, has moved to third round. The selection of authenticated ciphers for the new portfolio is based on the three use cases, namely lightweight applications, high-speed applications, and defense in depth. The use-case for lightweight applications includes performance of hardware implementations on resource constrained devices. The adopted CAESAR Hardware API is supported by a development package which includes VHDL code for universal pre- and post-processors for high-speed implementations. However, no code is provided for lightweight implementations. It was assumed that merging of pre- and post-processor functionality with lightweight cipher cores yields more efficient designs.

Designing lightweight implementations of cipher cores can be quite challenging and time consuming due to constraints on resources. Because of this, there are very few lightweight implementations of round two candidates published. Requiring the designers to comply with the CAESAR API constitutes an additional burden. We noticed that while developing lightweight implementations of Ketje-Jr and Ketje-Sr. Therefore, we developed generic lightweight pre- and post-processors which deal with the protocol aspects of the CAESAR API while leaving all algorithm specific aspects such as padding to the cipher core designers. Special care was taken not to duplicate functionality of a cipher core in the pre- and post-processors. We believe that this makes it simpler for designers of lightweight architectures to comply with the CAESAR API.

We developed lightweight cipher cores for ASCON and SILC and used our generic lightweight pre- and post-processors to make the designs compliant with the CAESAR API. Furthermore, we created new versions of our Ketje-Jr and Ketje-Sr implementations using these generic pre- and post-processors and compared them with our earlier integrated version to evaluate the penalty for using generic pre- and post-processors. All of these designs were developed with a limit of 200 slices and/or 500 LUTS and no embedded resources on Xilinx Spartan 6 FPGAs as the design criteria.