A Hardware Implementation of POET 2

Amir Moradi

Horst Görtz Institute for IT-Security
Ruhr-Universität Bochum, Germany
amir.moradi@rub.de

This document briefly addresses the specification of a hardware implementation of version 2.0 of the POET authenticated encryption scheme [2], submitted to the CAESAR authenticated encryption competition [1]. Amongst the variants of POET, only the one with full AES-128 for universal hashing is implemented. Though, the AES modules can easily be adjusted to generate the four-round outputs. Further, in the design expressed below, all operations of the underlying scheme are taken into account. In other words, the design is able to be used for encryption, decryption, tag generation, and tag verification. The block diagram of the implemented design is shown in Figure 1.

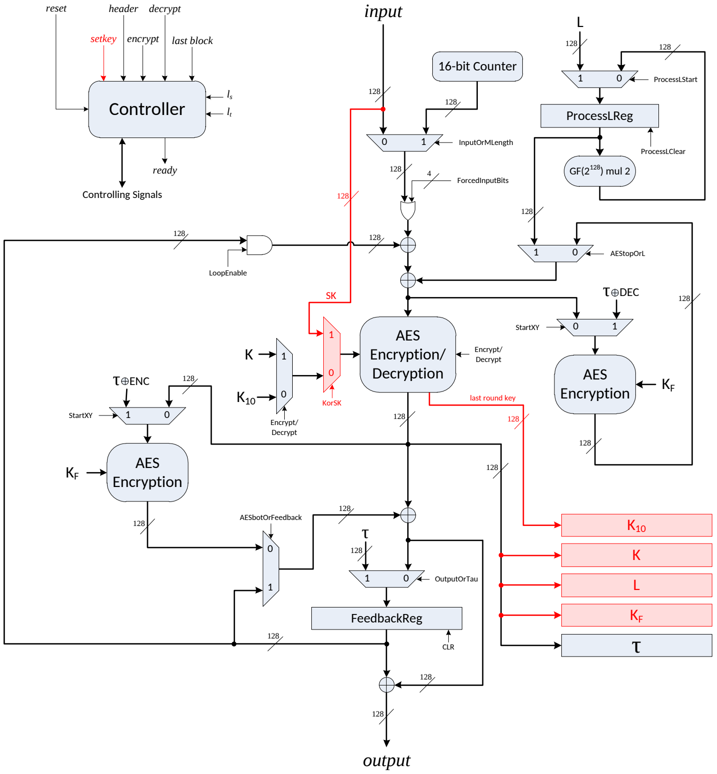

Fig. 1: Architecture of the implemented POETv2 hardware design.

The design has been developed to be independent of the AES modules. The reported performance figures consider round-based AES modules that are capable to finish either an AES encryption or decryption in ten clock cycles. Such modules can be replaced with faster but larger (e.g. unrolled) designs or with smaller but slower (e.g. serialized) designs.

Due to the use of K, L, and K_F, dedicated registers have been considered to store the corresponding values. The current design supports only messages whose length is a multiple of 128 bits (16 bytes).

The user can set the secret key SK, which is used internally to generate the other constants. Header and plaintext (respectively ciphertext) blocks are sent by a 128-bit input port; the ciphertext (respectively plaintext) blocks as well as the tag are returned by a 128-bit output port. A 16-bit counter is used to keep track of the 128-bit block indices. Therefore, the current design supports headers and messages of at most $2^{16} \times 128 = 2^{23} = 8$ MiB each.

In order to support POET decryption, the main AES module (the middle one in Figure 1) should support both AES encryption and decryption while only AES encryption modules suffice for top and bottom hashings. The main AES module requires the last round key K_{10} for decryption. Hence, a dedicated register to store K_{10} has been considered in the design. The design presented in Figure 1 supports both encryption and decryption. If only encryption is required, the main AES module does not need to support decryption; further, a couple of registers and multiplexers can be removed. However, a design which supports only decryption requires both encryption and decryption of the main AES module, and is actually not very different from the full design with all functionalities.

Design	Slice	Reg	LUT	Clock	Throughput
				(MHz)	(Mbps)
AES Encryption only	1,265	267	1,695	91	
AES Encryption and Decryption	2,040	280	2,978	61	
POET Encryption	5,208	1,600	6,139	63	672
POET Decryption	5,727	1,719	6,670	60	640
POET Full	5,870	1,720	6,689	59	629
Key Generation	662	512	187		

Table 1: FPGA performance figures.

The design has been implemented on a Spartan-6 FPGA (XC6SLX75) platform; its correct functionality has been verified by test vectors generated by the reference implementation available at `https://github.com/medsec/poet`. Table 1 depicts the resource and performance figures of different designs on the aforementioned FPGA. Since the main modules of POET are the AES encryp-

Design	Area	Clock	Throughput
	(GE)	(MHz)	(Mbps)
AES Encryption only	8,432	121	
AES Encryption and Decryption	11,008	62	
POET Encryption	33,624	72	768
POET Decryption	33,125	57	608
POET Full	37,164	55	587
Key Generation	4,279		

Table 2: ASIC performance figures.

tion/decryption, the maximum clock frequency of the designs depends heavily on the performance of the employed modules. Therefore, the performance figures of the used AES modules are also reported in Table 1.

It is noteworthy that the reported throughput was calculated for a long stream of data at the maximal frequency. For example, in case of encryption, a ciphertext block is generated every 12 clock cycles. For the ASIC figures depicted in Table 2, we used the Synopsys DesignCompiler version A-2007.12-SP1 for synthesis of the designs to the Virtual Silicon (VST) standard cell library UMCL18G212T3 which is based on the UMC L180 0.18μm 1P 6M logic process with a typical voltage of 1.8V. As a side note, the design has been realized such that l_s and l_t can be selected (each as a single bit) by the user to support different variants of the Intermediate Tag. Further, if the secret key SK is fixed, the constants K, K_{10}, L, and K_F can be hardwired to save the area for their corresponding registers and multiplexers that are marked by red color in Figure 1. In both tables, the row *Key Generation* indicates the corresponding area that could be saved from this approach.

References

1. Farzaneh Abed, Scott Fluhrer, John Foley, Christian Forler, Eik List, Stefan Lucks, David McGrew, and Jakob Wenzel. The POET Family of On-Line Authenticated Encryption Schemes. `http://competitions.cr.yp.to/caesar-submissions.html`, 2014.
2. Farzaneh Abed, Christian Forler, David McGrew, Eik List, Scott Fluhrer, Stefan Lucks, and Jakob Wenzel. Pipelineable On-line encryption. In *FSE*, volume 8540 of *Lecture Notes in Computer Science*, pages 205–223. Springer, 2014.