# FPGA Implementations of Message Authentication Codes based on Ascon-$p$

Mustafa Khairallah
*Seagate Research Group*
Singapore, Singapore
mustafa.khairallah@seagate.com

Srinivasan Yadhunathan
*Seagate Research Group*
Singapore, Singapore
srinivasan.yadhunathan@seagate.com

***Abstract*—With Ascon being selected as the new NIST lightweight cryptography standard, it is imperative to study how it lends itself to different use-cases that may not be covered directly by the original proposal. In this abstract, we focus on Message Authentication Codes (MACs). We study six ways of instantiating MACs using the Ascon permutations (Ascon-p). We compare these methods over multiple metrics, including throughput, hardware utilization, security margins/claims, side-channel friendliness and energy consumption of short and long messages. The comparison is performed with Xilinx Artix-7 FPGA as a target.**

***Index Terms*—Ascon, Lightweight, MAC, Authentication, FPGA**

## I. Introduction

In 2023, the National Institute for Standardization and Technology (NIST) announced Ascon [1] as the winner of the lightweight cryptography project. The Ascon proposal consists of a hash function and several Authenticated Encryption with Associated Data (AEAD) schemes. It is based on an internal cryptographic permutation known as Ascon-$p$, which operates on 320 bits, while the overall structure is based on a variant of the duplex sponge construction [2].

Both the Ascon schemes and the internal permutation have nice properties in terms hardware implementations. The design is lightweight for implementations not protected against side-channel attacks, in terms of throughput, area and energy efficiency [3], [4]. Simultaneously, the protected implementations are still moderately lightweight [5]. It is expected that many hardware designers may opt for implementing Ascon-$p$hardware accelerators in future devices. For this reason, applications of Ascon-$p$that are not covered by the proposed family of schemes is an interesting area of study.

In this abstract, we study the problem of building FPGA implementations of Message Authentication Codes (MACs) from Ascon-$p$. We consider a non-exhaustive classification of such MACs as:

1) Side-channel-friendly MACs: These are MACs that offer security guarantees against side-channel attacks without having to protect the full implementation. This is known as levelled-implementations [6], where a small part of the computation is heavily protected against side-channel attacks, while the rest of the computations can be lightly protected.
2) Non-side-channel-friendly MACs: These are MACs that do not offer inherent security against side-channel attacks unless the full algorithm is protected up to the required security order.

In this paper, we focus on complete hardware accelerators of the second category. As a proof of concept, we focus on unprotected implementations, while we leave protected implementations and comparing the two categories as future work and part of the full version of this work.

## II. Ascon Permutation

Ascon-$p$is a keyless Substitution-Permutation Network that operates on 320-bit blocks. It is composed of $r$ rounds where each round includes adding round constants, a substitution layer that consists of 64 parallel Sboxes over 5 bits, each, and a linear layer that operates parallelly on five 32-bit words. The details of each of these steps can be found in [1].

## III. Duplex-Sponge Based MACs

The duplex construction was introduced in 2011 by Bertoni et al. [2] as a framework for building encryption, MACs and AEADs from public permutations. The constructions based on the duplex construction have been extensively studied, and the survey by Mennink [7] includes a summary of the duplex-based construction. In this work, we are interested in two MAC constructions; the Full-State Keyed Sponge (FSKS) construction, depicted in Figure 1, and the Ascon-PRF construction, depicted in Figure 2.

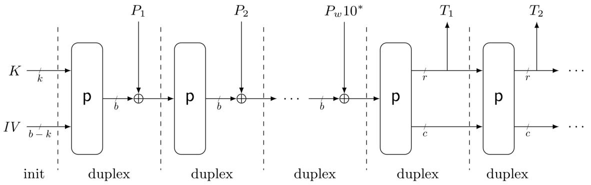

Fig. 1. The Full-State Keyed Sponge (FSKS) construction [7].

While the two constructions are fairly similar, they differ in two important points. Both divide the plaintext into blocks. FSKS divides the messages into blocks of the same size as the permutation size $b$ and absorbs one such block after each permutation call. While generating the tag, $r$ bits are extracted

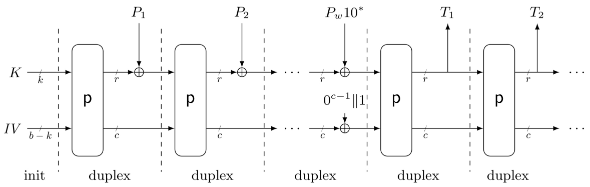

Fig. 2. The Ascon-PRF construction [7].

after each permutation call, where $r$ is known as the rate and $c = b - r$ is known as the capacity. Ascon-PRF, on the other hand, uses $r$-bit blocks for both absorption of the plaintext and extracting the tag. Besides, Ascon-PRF has a domain separation bit that is XORed to the internal state before tag extraction. These differences lead to different security levels of the two constructions. The security of FSKS is dominated by the requirement that any adversary must be bounded by

$$\text{TimeComplexity} \times \text{DataComplexity} \leq 2^c$$

while Ascon-PRF can reach $c$-bit security.

When it comes to selecting concrete parameters, the practical nature of the permutation comes into play. The previous security arguments, which are based on the security proofs in [7]. However, these proofs assume an ideal random permutation. In practice, Ascon-*p*is neither random nor ideal. However, based on the wealth of cryptanalysis efforts, we assume that when it is used with 12 rounds, there are no distinguisher with data or time complexity less than $2^{128}$. Hence, we use 12 rounds for both the initial and final calls in all constructions. We also use 12 rounds for FSKS as the adversary can affect the full state. For Ascon-PRF, the adversary has less control as only $r$ bits are affected during each call. We follow a similar approach to the original Ascon design, by using $r = 64$ with 6 rounds when the adversary is limited to $2^{64}$ data complexity and $r = 192$ with 8 rounds, otherwise.

## IV. Farfalle

The Farfalle construction, depicted in Figure 3, was proposed in 2016 by Bertoni et al. [8]. It is a permutation-based parallelizable Pseudo-Random Function (PRF). It allows extendible outputs, *i.e.*, once a message is absorbed, we can output many corresponding random blocks. By limiting the output size. This construction directly provides a permutation-based MAC. Besides, by limiting the output size to less than to equal to one block, we can ignore the $\texttt{roll}_e$ function. Hence, we need to select the permutation and the $\texttt{roll}_c$ construction. We select all the instances of the permutation to be Ascon-*p*with six rounds. According to the analysis in [8], if the permutation is sufficiently secure, $\texttt{roll}_c$ can be selected as a maximal-length LFSR over GF($2^{320}$). We use the function

$$k_{i+1} = x \cdot k_i \bmod f(x)$$

where, $k_{i+1}$ $k_i$ are polynomials in GF($2^{320}$) and $f(x)$ is the irreducible polynomial

$$f(x) = x^{320} + x^4 + x^3 + x^1 + x^0$$

TABLE I
Summary of the properties of the instantiated MACs

| Scheme | Parallelizable | Rate | Security |
|---|---|---|---|
| FSKS-64 | No | 320/12 | 128 |
| FSKS-128 | No | 320/12 | 64 |
| Ascon-PRF-192 | No | 192/8 | 128 |
| Ascon-PRF-64 | No | 64/6 | 64 |
| Ascon-Farfalle | Yes | 320/6 | 64 |

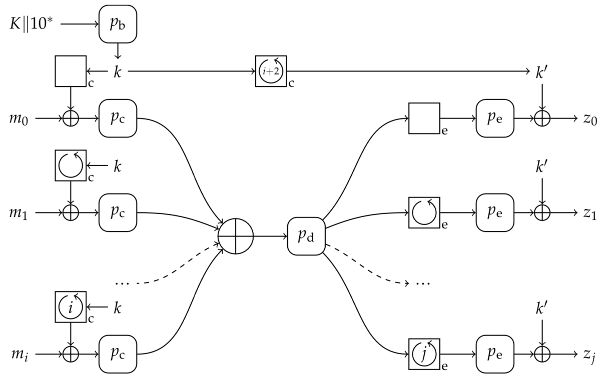

Fig. 3. The Farfalle construction [8].

## V. Hardware Architectures

We adopt two hardware architectures:

1) For duplex-based implementations, we use an iterative implementation with different number of unrolled rounds per clock cycles.
2) For Ascon-Farfalle, we use a six-stream pipelined implementation of the permutation. This mean that for long messages, the implementation processes six 320-bit blocks concurrently. To achieve this, we implement a six-round pipeline of the permutation, where the output is accumulated during absorption. The architecture resembles the PMAC architecture proposed by Khairallah et al. [9].

## VI. Results

We have implemented five different MACs based on both the duplex-sponge and Farfalle constructions. These MACs are summarized in Table I. All MACs are implemented with 128-bit keys and tags and target adversaries that can run attacks with $2^{128}$ time complexity. The security level in the table refers to the logarithmic maximum amount of data that can be processed under one key. This can be determined by different factors. Both FSKS-64 and FSKS-128 use 12 rounds per Ascon-*p*call. Such instance of Ascon-*p*is expected to be secure against all distinguishers with data complexity up to $2^{128}$. Hence, the security claims follows directly from the security proof given in [**?**]. FSKS-64 outputs only 64 bits per call during the squeeze phase, with 256-bit capacity, while FSKS-128 outputs 128 bits per call with 192-bit capacity. This

TABLE II
IMPLEMENTATION RESULTS ON XILINX ARTIX-7 FPGA

| | **Rounds/ Cycle** | **LUTs** | **FFs** | **Power** (Watts) | **Period** (ns) | **Cycles** (1600 Bytes) | **Cycles** (16640 Bytes) | **Throughput** (Mbps, short) | **Throughput** (Mbps, long) | **Energy** (nJ, short) | **Energy** (nJ, long) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FSKS-64 | 1 | 1007 | 391 | 0.15 | 5 | 516 | 5028 | 77.52 | 82.74 | 387.00 | 3771.00 |
| | 2 | 1658 | 390 | 0.189 | 5 | 258 | 2514 | 155.04 | 165.47 | 243.81 | 2375.73 |
| | 3 | 2167 | 391 | 0.249 | 7 | 172 | 1676 | 166.11 | 177.29 | 299.80 | 2921.27 |
| | 4 | 2376 | 289 | 0.3 | 7.5 | 129 | 1257 | 206.72 | 220.63 | 290.25 | 2828.25 |
| | 6 | 3334 | 390 | 0.39 | 10 | 86 | 838 | 232.56 | 248.21 | 335.40 | 3268.20 |
| | 12 | 6170 | 389 | 0.464 | 20 | 43 | 419 | 232.56 | 248.21 | 399.04 | 3888.32 |
| FSKS-128 | 1 | 1082 | 391 | 0.152 | 5 | 504 | 5016 | 79.37 | 82.93 | 383.04 | 3812.16 |
| | 2 | 1728 | 390 | 0.189 | 5 | 252 | 2508 | 158.73 | 165.87 | 238.14 | 2370.06 |
| | 3 | 2232 | 391 | 0.25 | 7 | 168 | 1672 | 170.07 | 177.72 | 294.00 | 2926.00 |
| | 4 | 2751 | 389 | 0.3 | 7.5 | 126 | 1254 | 211.64 | 221.16 | 283.50 | 2821.50 |
| | 6 | 3370 | 390 | 0.396 | 10 | 84 | 836 | 238.10 | 248.80 | 332.64 | 3310.56 |
| | 12 | 6207 | 389 | 0.47 | 20 | 42 | 418 | 238.10 | 248.80 | 394.80 | 3929.20 |
| Ascon-PRF-192 | 1 | 1031 | 391 | 0.149 | 5 | 558 | 5571 | 71.77 | 74.68 | 415.21 | 4150.15 |
| | 2 | 1693 | 390 | 0.19 | 5 | 279 | 2786 | 143.54 | 149.35 | 264.73 | 2646.07 |
| | 4 | 2421 | 389 | 0.308 | 7.5 | 140 | 1393 | 191.39 | 199.14 | 321.86 | 3217.06 |
| | 8 | 4312 | 388 | 0.44 | 13 | 70 | 697 | 220.83 | 229.78 | 398.49 | 3983.03 |
| Ascon-PRF-64 | 1 | 826 | 391 | 0.144 | 5 | 1224 | 12504 | 32.68 | 33.27 | 881.28 | 9002.88 |
| | 2 | 1521 | 390 | 0.187 | 5 | 612 | 6252 | 65.36 | 66.54 | 572.22 | 5845.62 |
| | 3 | 2004 | 391 | 0.247 | 7 | 408 | 4168 | 70.03 | 71.29 | 705.43 | 7206.47 |
| | 6 | 3185 | 390 | 0.376 | 10 | 204 | 2084 | 98.04 | 99.81 | 767.04 | 7835.84 |
| Ascon-Farfalle | 6 | 3863 | 2634 | 0.43 | 4 | 58 | 434 | 862.07 | 1198.16 | 99.76 | 746.48 |

means that FSKS-64 requires one extra call per message while being able to achieve double the security.

Ascon-PRF-64 and Ascon-PRF-192 follow the heuristic approach followed by the original Ascon design. From a provable security stand point, both instances achieve 128-bit security. However, instead of using 12 rounds per call, we recover part of the performance drop due to reducing the rate by reducing the number of rounds per call. We use 12 rounds form the first and last call per message. The internal calls are limited to 6 rounds and 8 rounds for Ascon-PRF-64 and Ascon-PRF-192, respectively. Hence, we limit the rate and security of Ascon-PRF-64 to 64 bits, while we use 192-bit rate for Ascon-PRF-192, with 128-bit security.

Finally, the security of Ascon-Farfalle is based on the heuristic analysis performed in [8]. While the design follows Farfalle closely, two details of the original constructions are missing from this use case:

1) We only output a single block. This eliminates the need to design a $\texttt{roll}_e$ function. $\texttt{roll}_e$ is one of the critical points of failure in the original construction.
2) The output block is truncated from 320-bits to 128-bits. This adds an extra layer of protection relying on the truncated permutation construction ( [7], Section 5).

Hence, we conjecture it is sufficient to use 6 rounds per call to achieve at least security against adversaries with $2^{64}$ data complexity.

We have implemented the five designs using Verilog and synthesized them using Xilinx Vivado, targetted for the Xilinx Artix-7 100t FPGA. The results are summed up in Table II. We observe that both FSKS and Ascon-PRF offer interesting trade-offs between speed, utilization and power consumption. For long messages, the difference between FSKS-64 and FSKS-128 is very small. Hence, if the performance over short messages is not an issue, FSKS-64 is to be preferred over FSKS-128. Besides, similar to observations made in earlier work [4], [10], we observe that the optimal instance for energy consumption is not the round based implementation, but the two-round-unrolled implementation.

Another interesting observation is that while Ascon-PRF-64 is significantly slower and less energy-efficient compared to other implementations, Ascon-PRF-192 has performance that is close to that of FSKS-128, which offers an interesting security trade-off between full-state absorption with 12 rounds vs. half-state absorption with 8 rounds.

Finally, the multi-stream implementation of Ascon-Farfalle has significantly larger area and power consumption compared to most implementations, but it also has much higher frequency and throughput, and is much more energy efficient. In fact, it has the best throughput and and energy efficiency among all scheme. We define energy efficiency as energy/LUT/bit for long messages and throughput efficiency as throughput/LUT. These metrics are summed up in Table III.

While Ascon-Farfalle has a high register count due to pipelining, it is explained in [9] that most of these registers come for free in FPGA as they are already part of the logic slices used to implement the circuit.

TABLE III
COMPOUND EFFICIENCY METRIC OF THE DIFFERENT SCHEMES.

| **Scheme** | **Throughput/LUT** | **Energy/LUT.bit** |
|---|---|---|
| FSKS-64 | 0.099 (2 rounds) | $8.94 \times 10^{-6}$ (4 rounds) |
| FSKS-128 | 0.096 (2 rounds) | $9.84 \times 10^{-6}$ (4 rounds) |
| Ascon-PRF-192 | 0.088 (2 rounds) | $9.98 \times 10^{-6}$ (4 rounds) |
| Ascon-PRF-64 | 0.043 (2 rounds) | $1.85 \times 10^{-5}$ (3 rounds) |
| Ascon-Farfalle | 0.31 | $1.45 \times 10^{-6}$ |

## REFERENCES

[1] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, “Ascon v1. 2,” *Submission to the CAESAR Competition*, vol. 5, no. 6, p. 7, 2016.

[2] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, “Duplexing the sponge: single-pass authenticated encryption and other applications,” in *Selected Areas in Cryptography: 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers 18*. Springer, 2012, pp. 320–337.

[3] K. Mohajerani, R. Haeussler, R. Nagpal, F. Farahmand, A. Abdulgadir, J.-P. Kaps, and K. Gaj, “Fpga benchmarking of round 2 candidates in the nist lightweight cryptography standardization process: methodology, metrics, tools, and results,” *Cryptology ePrint Archive*, 2020.

[4] M. Khairallah, T. Peyrin, and A. Chattopadhyay, “Preliminary hardware benchmarking of a group of round 2 nist lightweight aead candidates,” *Cryptology ePrint Archive*, 2020.

[5] K. Mohajerani, L. Beckwith, A. Abdulgadir, E. Ferrufino, J.-P. Kaps, and K. Gaj, “Sca evaluation and benchmarking of finalists in the nist lightweight cryptography standardization process,” Cryptology ePrint Archive, Paper 2023/484, 2023, https://eprint.iacr.org/2023/484. [Online]. Available: https://eprint.iacr.org/2023/484

[6] D. Bellizia, O. Bronchain, G. Cassiers, V. Grosso, C. Guo, C. Momin, O. Pereira, T. Peters, and F.-X. Standaert, “Mode-level vs. implementation-level physical security in symmetric cryptography: a practical guide through the leakage-resistance jungle,” in *Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part I 40*. Springer, 2020, pp. 369–400.

[7] B. Mennink, “Understanding the duplex and its security,” *Cryptology ePrint Archive*, 2022.

[8] G. Bertoni, J. Daemen, S. Hoffert, M. Peeters, G. Van Assche, and R. Van Keer, “Farfalle: parallel permutation-based cryptography,” *IACR Transactions on Symmetric Cryptology*, pp. 1–38, 2017.

[9] M. Khairallah, A. Chattopadhyay, and T. Peyrin, “Looting the luts: Fpga optimization of aes and aes-like ciphers for authenticated encryption,” in *Progress in Cryptology–INDOCRYPT 2017: 18th International Conference on Cryptology in India, Chennai, India, December 10-13, 2017, Proceedings 18*. Springer, 2017, pp. 282–301.

[10] A. Caforio, F. Balli, and S. Banik, “Energy analysis of lightweight aead circuits,” in *Cryptology and Network Security: 19th International Conference, CANS 2020, Vienna, Austria, December 14–16, 2020, Proceedings 19*. Springer, 2020, pp. 23–42.