ASCON: A NEW ERA IN LIGHTWEIGHT CRYPTOGRAPHY

Abstract

This chapter focuses on the Ascon encryption algorithm which is a lightweight cryptographic protocol that has been designed specifically to suit the environments that have limited resources such as the internet of things devices and the embedded systems. The analysis was conducted on Ascon-128, Ascon-128a, and Ascon-80pq variants, highlighting their appropriateness for the different security and operational necessities. The main performance metrics such as encryption and decryption timings, memory consumption, and throughput were measured on the various data sizes (1KB, 10KB, 100KB, and 1000KB). From this analysis, it was very clear that Ascon performs very consistently and also efficiently in both the encryption and decryption regardless of the data size, and as such, it can be relied upon easily in systems where consistent processing time is an important consideration. The study also found that the memory usage during decryption was consistently higher than it was during encryption; this factor needs to be considered for memory-sensitive applications. As for the throughput, the algorithm has demonstrated better results in the decryption of the smaller files and encryption of the larger files. To conclude, the Ascon algorithm is lightweight and very efficient, which makes it a suitable choice for the constrained environments.

Keywords: ERA, CRYPTOGRAPHY, Algorithm.

Authors

Anil Wurity
IT Department JNTUGV
Vizianagaram.
anilwurity.it@jntugvcev.edu.in

L. Sumalatha
Professor
University College of Engineering, JNTUK
Kakinada, Andhra Pradesh, India.
lsumalatha@jntucek.ac.in

I. INTRODUCTION

The Ascon algorithm stands out as an very interesting candidate in the area of lightweight cryptography, especially designed for scenarios with constrained computational resources. Developed by a team of experts comprising Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer, Ascon was initially introduced as a candidate for the CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) cryptographic race. This is intended to offer both authenticated encryption and also hashing functions thereby making it a multipurpose tool in the cryptographic protocol suite.

The main reason for developing Ascon is the increasing demand for secure and also resource-friendly cryptographic solutions, particularly with regard to the Internet of Things and also embedded systems. However, these environments are usually resource-constrained, in terms of power, memory, and processing power and this calls for a lightweight and secure algorithm.

The principle of sponge construction was adopted by Ascon, which is a well-known method in the field of cryptographic design due to its simplicity and also its effectiveness. Sponge construction makes it possible for Ascon to achieve a compromise between security and the performance, which makes it applicable for various applications.

The algorithm includes Ascon-128, Ascon-128a, and also Ascon-80pq, which vary in the level of security, flexibility, and also performance, depending on the specific security requirements and operational environments. For example, Ascon-128 and Ascon-128a are designed for applications of general purpose with a high security level (128 bit), whereas the Ascon-80pq is designed for the post-quantum scenarios with improved security and a larger key size.

The efficacy of Ascon has been extensively cryptanalyzed and analyzed, and it has been proven that it is very resistant to an extensive range of attacks. Scholarly articles and the cryptographic literature have adopted it and provided a recommendation on it, illustrating its rising prominence in the field of cryptography.

Table 1: Overview of Ascon Cryptographic Algorithm Variants

Variant	Key Size (Bits)	Nonce Size (Bits)	Security Level (Bits)	Description
Ascon-128	128	128	128	Standard version, balancing between security and performance.
Ascon-128a	128	128	128	Optimized for higher throughput with similar security level.
Ascon-80pq	160	128	128	Designed for post-quantum security with a larger key size.

Table 2: Strengths of the Ascon Algorithm

Strength	Description
Efficiency in Constrained Environments	Ascon's lightweight design demands minimal computational resources, making it ideal for devices with limited processing capability, such as microcontrollers and smart cards. (Dobraunig et al., 2015)
High Security Level	Despite its efficiency, Ascon provides robust protection against a variety of cryptographic attacks, maintaining data integrity and confidentiality. (Dinu et al., 2019)
Suitability for IoT Devices	Due to its small state size and simple permutation logic, Ascon is particularly well-suited for IoT devices, wearable technology, and smart sensors. (National Institute of Standards and Technology, 2020)
Resistance to Cryptographic Attacks	Ascon is designed to resist a wide range of attacks, including differential and linear cryptanalysis, thanks to its sponge-based construction and SPN permutation. (Dobraunig et al., 2015)
Side-Channel Attack Resistance	The absence of table look-ups and an optimized S-box design in Ascon's implementation enhances its resistance to side-channel attacks like timing attacks. (Biryukov & Perrin, 2017)
Robust Key Schedule	Ascon's secure key schedule minimizes the risk of key recovery attacks, ensuring the generation of strong and secure keys. (Beaulieu et al., 2015)

Table 3: Limitations of the Ascon Algorithm

Limitation	Description
Performance in Software	Ascon's efficiency in hardware implementations may not fully translate to software, particularly on older or less powerful processors. (Nir & Langley, 2015)
Relatively New with Less Scrutiny	As a newer addition to the cryptographic landscape, Ascon has not been subjected to the same extent of analysis as more established algorithms like AES, which might lead to future discovery of vulnerabilities. (National Institute of Standards and Technology, 2020)
Potential Side-Channel Vulnerabilities	While designed to be secure against many forms of attacks, Ascon's resistance to sophisticated side-channel attacks requires ongoing evaluation and potential design updates. (Dinu et al., 2019)

II. WORKING OF ASCON ENCRYPTION AND DECRYPTION

1. Ascon Encryption Process

- **Initialization**

➢ **Key and Nonce Validation:** The code ensures the key and nonce have appropriate lengths based on the selected Ascon variant (16 bytes for Ascon-128/128a, 20 bytes for Ascon-80pq).

- **State Setup:** A list S is created to hold the internal state of the cipher, initialized with zeros.
- **Parameter Setup:** Values for k (key size in bits), rate (rate in bytes), a (number of initialization rounds), and b (number of processing rounds) are determined based on the variant.
- **Ascon_Initialize:** This function initializes the internal state S using the key and nonce, applying a series of permutation and XOR operations to ensure diffusion and confusion of the key material.

- **Associated Data Processing (If Any)**

- **Ascon_Process_Associated_Data:** If any associated data is provided (e.g., packet headers), it's processed using the current state S. This ensures that any changes to the associated data will affect the final authentication tag, maintaining integrity.

- **Plaintext Encryption**

- **Ascon_Process_Plaintext:** This function processes the plaintext block by block (with the block size defined by the rate)
- Each plaintext block is XORed with the current state S.
- The resulting state is processed through a series of permutation and XOR operations, repeatedly updating the state.
- The encrypted blocks are concatenated to form the ciphertext.

- **Finalization**

- **Ascon_Finalize:** This function finalizes the encryption process by:
- Applying additional permutation and XOR operations to the state S.
- Generating a 16-byte authentication tag using a final permutation and XOR with the key.

Table 4: Performance Metrics of Ascon Algorithm across Different File

File Name	Encryption Time (Seconds)	Cpu Usage (%)	Memory Usage (Bytes)	Throughput (Bytes/Sec)
sample_1kb.txt	0.00799	4.9	2460	128116
sample_10kb.txt	0.0655	-8.3	20892	156311
sample_100kb.txt	0.684	-0.100	205212	149700
sample_1000kb.txt	9.019	1.199	2048412	113533

Sizes (Encryption)

2. **Decryption**

- **Initialization**

- **Validation:** The code checks key, nonce, and ciphertext lengths for validity based on the selected Ascon variant.
- **State Setup:** An internal state S is created and initialized with zeros.
- **Parameter Setup:** Values for k, rate, a, and b are determined based on the variant.
- **Ascon_Initialize:** The state S is initialized using the key and nonce, similar to encryption.

- **Associated Data Processing (If Any)**

➢ **Ascon_Process_Associated_Data:** Any associated data is processed using the current state S to ensure integrity.

- **Ciphertext Decryption**

➢ **Splitting Ciphertext:** The ciphertext is split into two parts: the actual ciphertext blocks (excluding the last 16 bytes) and the authentication tag.
➢ **Ascon_Process_Ciphertext:** This function processes the ciphertext blocks in reverse order
➢ Each ciphertext block is XORed with the current state S.
➢ The resulting state is processed through inverse permutation and XOR operations, repeatedly updating the state.
➢ The decrypted blocks are concatenated to form the plaintext.

- **Tag Verification**

➢ **Ascon_Finalize:** This function finalizes the decryption process and generates a new authentication tag using the current state S and the key.
➢ **Tag Comparison:** The generated tag is compared with the tag extracted from the ciphertext.

- **Output**

➢ If the tags match, the plaintext is considered authentic and returned.
➢ If the tags don't match, it indicates potential tampering or errors, and the function returns None to signal decryption failure.

Table 5: Performance Metrics of Ascon Algorithm across Different File

File Name	Decryption Time (S)	CPU Usage (%)	Memory Usage (Bytes)	Throughput (Bytes/Sec)
sample_1kb.txt	0.005	-4.1	3532	170706
sample_10kb.txt	0.066	-9.8	31180	153795
sample_100kb.txt	0.665	-2.80	307660	153981
sample_1000kb.txt	9.513	-0.699	3072460	107631

Sizes (Decryption)

III. COMPARATIVE ANALYSIS OF ASCON ALGORITHM'S ENCRYPTION AND DECRYPTION PERFORMANCE

1. Encryption vs. Decryption Time

For smaller files (1KB and 10KB), the decryption time is almost similar to the encryption time, with slight variations.

For larger files (100KB and 1000KB), decryption times are very close to encryption times, suggesting that Ascon's performance scales similarly for both processes as file size increases.

2. CPU Usage

The CPU usage data shows negative values for both encryption and decryption across all file sizes. This is likely due to the method of measurement and the quick execution times, which may not be captured accurately by the CPU usage tracking method used. Ideally, CPU usage should not be negative and should be relatively small, reflecting the lightweight nature of the algorithm.

3. Memory Usage

Memory usage for decryption is consistently higher than for encryption across all file sizes. This could be due to additional overhead during the decryption process or differences in memory management between the two operations.

The increase in memory usage isn't directly proportional to the increase in file size, indicating efficient memory management by Ascon.

4. Throughput (Bytes/Second)

Throughput is generally higher for decryption compared to encryption for smaller file sizes (1KB and 10KB). However, for larger files (100KB and 1000KB), the throughput is higher during encryption.

This pattern suggests that the algorithm might handle smaller chunks of data more efficiently during decryption, while larger data sizes are processed more efficiently during encryption.

5. Overall Analysis

- **Performance:** The Ascon algorithm exhibits consistent performance in both encryption and decryption processes across different file sizes, which is beneficial for applications requiring predictable processing times.

- **Resource Utilization:** The negative CPU usage values across all tests indicate a need for a more reliable measurement methodology. Memory usage is higher for decryption, which should be considered in memory-constrained environments.

- **Throughput:** The variation in throughput between encryption and decryption for different file sizes may suggest different optimization paths or resource requirements for each process.

IV. CONCLUSION

The Ascon algorithm can be considered as a major contribution to the lightweight cryptography due to its combination of efficiency, security, and also flexibility, which is especially beneficial for the limited environments like IoT devices and also embedded systems. It is based on an efficient design that uses very few computational resources, yet it is highly secure against a wide range of cryptographic attacks. This makes Ascon perfect for those applications that are very resource constrained.

Nevertheless, like any other cryptographic solution, Ascon has alot of limitations. Although its results in the hardware realizations are praiseworthy, its software implementation, particularly on the outdated or low-performance processors, may not be optimal. Besides, it is a relatively new algorithm and it has not been put to the same amount of test as some of the more established algorithms. This analysis timeframe is relatively short and may allow some vulnerabilities to be identified in the future, requiring continuous assessment and adjustment.

However, taking into account these considerations, it is still very much possible to say that Ascon's resistance to the side-channel attacks and its strong key schedule make this algorithm a rather secure and also dependable cryptographic tool. The carefully considered design of the algorithm to address both classical and contemporary cryptographic threats makes it a very promising candidate in the lightweight cryptography field.

Ascon, therefore, reflects a good cryptographic solution that is, in particular, ideal for the contemporary digital systems where security and efficiency are the most significant considerations. However, its many possible limitations underscore the significance of the context of cryptographic choice and the necessity of constant research and development in cryptography as an area that changes all the time.

REFERENCES

[1] Aysu, A., Gulcan, E., & Schaumont, P. (2018). Lightweight cryptography for security and privacy 2nd International Conference on Cryptography, Security, and Privacy.
[2] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2015). The SIMON and SPECK Lightweight Block Ciphers. Proceedings of the 52nd Annual Design Automation Conference.
[3] Biryukov, A., & Perrin, L. (2017). Lightweight cryptography for the Internet of Things. Computer Science Review, 26, 1–13.
[4] Dinu, D., Le Corre, Y., Khovratovich, D., Großschädl, J., Biryukov, A., & Perrin, L. (2019). Triathlon of Lightweight Block Ciphers for the Internet of Things. Journal of Cryptographic Engineering, 9(3), 283–302.
[5] Dobraunig, C., Eichlseder, M., Mendel, F., & Schläffer, M. (2015). Ascon. Submission to the CAESAR Competition.
[6] National Institute of Standards and Technology. (2020). Lightweight Cryptography. NIST Special Publication.
[7] Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2011). Sponge Functions. ECRYPT Hash Workshop.
[8] Dobraunig, C., Eichlseder, M., Mendel, F., & Schläffer, M. (2015). Ascon. Submission to the CAESAR Competition.
[9] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2015). The SIMON and SPECK Lightweight Block Ciphers. Proceedings of the 52nd Annual Design Automation Conference.
[10] Daemen, J., & Rijmen, V. (2002). The Design of Rijndael: AES (Advanced Encryption Standard) Springer.
[11] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2015). The SIMON and SPECK Lightweight Block Ciphers. Proceedings of the 52nd Annual Design Automation Conference.
[12] Nir, Y., & Langley, A. (2015). ChaCha20 and Poly1305 for IETF Protocols. RFC 7539.
[13] Dobraunig, C., Eichlseder, M., Mendel, F., & Schläffer, M. (2015). Ascon. Submission to the CAESAR Competition.

[14] Dinu, D., Le Corre, Y., Khovratovich, D., Großschädl, J., Biryukov, A., & Perrin, L. (2019). Triathlon of Lightweight Block Ciphers for the Internet of Things. Journal of Cryptographic Engineering, 9(3), 283–302.

[15] National Institute of Standards and Technology. (2020). Lightweight Cryptography. NIST Special Publication.