2025

IFN648 Group Project: Lightweight Cryptography

Title:

Hashing the Future of IoT — ASCON-Hash for Secure FOTA

Group9

Pheerawat Senakham

N11916851

Yi-Ting, Chen

N11530430

Yi-Chen, Chen

N11564628

1. Executive Summary

Firmware-over-the-air (FOTA) keeps billions of IoT devices secure and functional, yet today's update pipelines still depend on the 20-year-old SHA-256 hash. Our literature survey (2020 – May 2025) finds **no peer-reviewed work that embeds the newer, NIST-standardised ASCON-Hash inside a full FOTA protocol**—only standalone speed or hardware tests. This report attempts to fills that gap.

We first outline seven practical criteria for a robust FOTA system: cryptographic integrity, rollback protection, fail-safe operation, pause-and-resume support, low resource footprint, secure transport & key management, and certification/auditability. We then measure ASCON-Hash against each criterion and benchmark it against the industry default, SHA-256.

Key findings:

- Efficiency: 4× faster and ~4× less energy on Cortex-M cores, code shrinks from ~3–4 KB to ~1.7 KB, RAM to < 100 B.
- Security: 128-bit collision/pre-image strength, with no attacks on the full 12-round design.
- Integration: One 320-bit permutation powers both ASCON-AEAD (secure channel) and ASCON-Hash, simplifying code and hardware reuse.
- Limitation: FIPS 140-3 validation is not yet available, so highly regulated sectors must wait for certification.

We recommend phased pilots that swap SHA-256 for ASCON-Hash in new or low-risk product lines, reuse the ASCON permutation already present for encrypted channels, and track forthcoming FIPS modules. A full-scale open-source prototype is proposed to accelerate industry adoption.

Bottom line: ASCON-Hash offers an immediate, standards-based path to faster, smaller, and more energy-efficient FOTA without sacrificing security; the remaining hurdle is certification timeline, not technical merit.

2. IoT Security Primer

2.1 CIA Triad in resource-constrained devices

Internet of Things (IoT) devices have been challenged in limited resources since they have been commonly applied in various fields in recent years. With the growing trend of widespread use, relevant vulnerabilities such as software threats, Advanced Network Threats (ANT), and data privacy are being prioritised by IT experts to direct attention to it. To making sure Confidentiality, Integrity, and Availability (CIA), the foundational principles of information security — has become especially difficult under resource-constrained condition. As Jenkins notes, "The CIA triad has since become synonymous with information security," stating its relevance even in the context of lightweight, embedded systems [2].

2.2 Typical threat models for IoT nodes

According to the Open Worldwide Application Security Project (OWASP)'s explanation of threat modeling [5], broadly, threat modeling includes examining concerns for existing systems or applications, predicting potential risks, and feasible approaches to deal with them. In the context of IoT nodes, threat modeling becomes particularly important due to their exposure to both physical and network-based threats. IoT device threat models typically take into account situations like data leaking, firmware tampering, man-in-the-middle (MITM) attacks, denial-of-service (DoS) attacks, and unauthorised access.

3. Firmware-Over-The-Air (FOTA) Updates

3.1 Why firmware integrity matters

Firmware integrity is another perspective that worth our attention when securing IoT network. Most IoT firmware update processes focus on checking digital signatures and making sure the firmware file has not been altered [9]. However, these checks can only confirm who sent the update, not what is actually inside it, which means that even signed firmware could include harmful content without being noticed.

3.2 Generic FOTA workflow

Firmware Over The Air (FOTA) workflow is a suitable solution to remotely organise and update IoT devices as long as they are internet- connected, saving time and preventing physical intervention. For example, to keep cars' firmware up to date, car manufacturers can simply send the required data to vehicles; likewise, mobile manufacturers usually through FOTA running updating programs in the mobile phones' background without any operation from the user, it takes about three to ten minutes [8].

3.3 Attack surface during update delivery

Meanwhile, there are many potential vulnerabilities that can be exploited during the update process. Therefore, to ensure that update data is securely transmitted and only trusted data is installed on target devices, it is suggested to apply end-to-end encryption as well as meticulous digital signature [7].

4. Cryptographic Hash Functions in Constrained Environments

4.1 Role of hashing in authenticity & integrity

Generally, hashing plays a key role in ensuring both the authenticity and integrity of data, especially in firmware updates and secure communication. Hashing is the procedure of converting input data into a different size of string of characters, representing the original input data.

4.2 Modification-Detection Codes (MDCs) vs MAC-based verification

Modification-Detection Code (MDC) uses to detect changes in data by generating a different size hash value from the original message. However, MDCs do not provide authenticity, since anyone can compute the hash without needing a key. In contrast, Message Authentication Code (MAC) not only ensure integrity but also authenticate the sender. In fact, MAC is the most popular method to provide both integrity and authenticity in modern IoT network [4]. Since the MAC is generated using a secret key shared between the sender and receiver. This means only someone with the correct key can create a valid MAC.

4.3 Limitations of classical hashes (SHA-2/SHA-3) on MCUs

Nowadays, although SHA-2 and SHA-3 are popular hash functions used on MCUs, their implementation on MCUs with limited resources presents certain difficulties. First of all, they require a lot of processing power and time, especially SHA-3, which performs in software almost twice as long as SHA-2, even though it operates in hardware around a quarter of the time [3]. Additionally, for internal buffers and code storage, both algorithms need a significant amount of flash memory. This can be a serious limitation for ultra low power MCUs, which often operate with as little as 32 KB of RAM and minimal flash storage.

5. Lightweight Cryptographic Hash Functions (LWCHF)

5.1 Design goals and metrics

LWCHFs are specifically designed to meet the constraints of resource-limited environments, such as MCUs used in IoT devices world widely. In fact, by 2025, there will likely be 30.90 billion connected IoT devices [16].

Therefore, the design of LWCHFs aims to minimise demands on hardware size, computing cycles, and energy consumption.

Performance metrics:

- Area: Refers to the amount of physical space or gate equivalents (GE) required on a chip.
- Cycles per Byte: shows how many clock cycles are required to process each input byte. Faster performance, which is essential for real-time or energy-sensitive applications, is indicated by lower values.
- Energy Efficiency: LWCHFs are designed to consume less power, enabling longer battery life in IoT devices.

6. ASCON Family Overview

NIST LWC standard outcome

In February 2023, the National Institute of Standards and Technology (NIST) selected the ASCON family as the standard for lightweight cryptography. Built on a 320-bit permutation, ASCON supports AEAD, hashing, and XOF, and is tailored for resource-constrained environments like IoT devices [10].

NIST's decision was based on three main factors:

- Mature Security: ASCON had been previously selected for the CAESAR portfolio and required no major changes during the LWC process. It demonstrated strong resistance in both nonce-respecting and nonce-misuse scenarios.
- Strong Performance: ASCON ranked highly in NIST's software and hardware benchmarks, offering efficient protection even with side-channel countermeasures.
- Design Flexibility: With variants such as Ascon-128, Ascon-128a, and Ascon-80pq, ASCON supports different trade-offs between performance and security, including hash and XOF functionality for broader use cases.

Although it lacks a 256-bit key version, NIST clarified that this standard focuses on lightweight applications, not post-quantum cryptography. For such cases, AES-GCM remains suitable.

7. Design Internals of ASCON-Hash

7.1 Sponge construction & 320-bit permutation

ASCON-Hash adopts a sponge construction, dividing a 320-bit internal state into a public rate (r) and a hidden capacity ($c = 320 - r$). Inputs are absorbed into the rate part, processed through permutation rounds, and finalized to output a digest [11].

Two permutations are used:

- p^a (12 rounds): for initialization and finalization
- p^b (6/8 rounds): for message absorption and squeezing

Each round includes:

1. Constant Addition (pC): breaks symmetry using round constants
2. Substitution Layer (pS): 64 parallel 5-bit S-boxes (low-degree, bitslice-friendly)
3. Linear Layer (pL): XORs and rotations for diffusion

This structure enables online processing, reuse across AEAD/hash modes, and compact implementation on 8/16/32/64-bit platforms.

7.2 Round function, S-box, linear layer

The ASCON round function is optimized for both security and lightweight performance. Key design elements include:

- Constant Addition: XORs a round constant into one state word to ensure round distinction.
- S-box (5-bit): Applied bitsliced to enable masking, with algebraic degree 2 and branch number 3. Inspired by Keccak's χ but optimized for lightweight diffusion.
- Linear Layer: Word-level mixing using rotations and XORs, with branch number 4 to strengthen resistance against differential and linear attacks.

Design Advantages:

- No inverse needed: forward-only permutation simplifies code and saves memory.
- Parallelizable: 5 state words allow up to 5 independent operations.
- Side-channel resistant: Boolean logic supports threshold masking.

7.3 Lightweight implementation results (gate-equivalent, RAM/ROM, cycles)

ASCON is highly efficient in both hardware and software implementations:

- Gate Count:
    - ASCON-Hash/XOF: ~3.0k GE
- Memory Usage:
    - RAM: 0 bits (internal-state only)
    - ROM: ~1.2k bits (round constants/config)

In RECO-ASCON trials, reconfigurable hardware was tested with varying round counts (e.g., 6, 8, 12), balancing energy cost and security level. On ASIC, 8 rounds offered secure hashing with <3K GE and ultra-low power operation, while 12-rounds improved security margins with minimal area increase [12].

These results confirm ASCON's suitability for SoCs and low-power cryptographic accelerators in embedded applications.

8. Security Properties

8.1 Cryptanalytic Results

ASCON-Hash and ASCON-XOF offer [13]:

- 128-bit resistance to preimage, second-preimage, and collision attacks (v1.2 spec)
- Adjustable XOF output affects bounds:
    - Preimage ~ min(128, ℓ)
    - Collision ~ min(128, ℓ⁄2)

Reduced-round attacks (e.g., 2-round collision with complexity ~$2^{32.5}$) have been demonstrated, but they do not threaten full-round versions (12/6/8 rounds). These results aid confirm ASCON's security margins.

No known preimage or forgery attacks exist against the full-round ASCON variants.

8.2 Side-Channel and Fault Injection

Threat	ASCON Features
Side-Channel Attack	Bitsliced S-boxes, no lookup tables, threshold-friendly
Fault Injection Attack	Stateless design, constant-time logic, supports duplication and fault detection

RECO-ASCON shows that ASCON can be tuned for security vs performance, making it suitable for embedded platforms such as smartcards, medical devices, and IoT sensors [14].

9. Performance in Real-World MCUs

9.1 Benchmarks on Cortex-M0/M3 & AVR

ASCON-Hash has been benchmarked on typical IoT microcontrollers including AVR ATmega128, ARM Cortex-M0, and Cortex-M3. Results show it achieves [15]:

- Execution Time:
  - ~650 cycles on Cortex-M3
  - ~800 cycles on M0
  - ~1600 cycles on 8-bit AVR
- Memory Usage:
  - ROM: <2 KB
  - RAM: ~140–150 B

ASCON-Hash offers low latency and minimal footprint, making it practical even on 8-bit platforms, and highly efficient on 32-bit embedded systems.

10. Integrating ASCON-Hash into FOTA Workflows

10.1 Secure-boot + signature + hash chain

A secure FOTA system integrates [17]:

- Hash validation at boot (e.g., ASCON-HASH) to prevent tampered firmware
- Digital signatures to verify vendor authenticity (e.g., ECDSA)
- Hash chains to ensure correct packet ordering and block integrity

Together, these provide layered security while maintaining a lightweight implementation profile.

10.2 Update package structure

A firmware packet typically includes:

Field	Description
Sequence ID	Tracks order
Payload	Firmware block
Digest	Hash of payload
Signature	Verifies authenticity

In this structure, the digest field can be computed using ASCON-Hash to provide lightweight yet secure block-level integrity checks in resource-constrained embedded systems.

10.3 Verification steps on-device

Each packet is processed as follows:

1. Check Sequence ID for ordering
2. Recompute and compare Digest
3. Validate Signature
4. Store verified packets in secure flash
5. Final integrity check before commit

Enables secure updates with minimal RAM, even on devices lacking full-image buffering.

11. Discussion

11.1 Evidence of a Research Gap

Literature survey (2020 – May, 2025):

Based on our literature review, there is **currently no peer-reviewed work that implements ASCON-Hash integrated into a full FOTA protocol**. There are some existing studies focus on benchmarking ASCON-Hash in isolation or in comparison with other hash functions [18].

Implication:

This indicates that the **majority of current implementations still rely on SHA-2 or SHA-3, particularly SHA-256**, due to its ubiquity and standardized presence as the default function, despite the introduction of newer NIST lightweight cryptographic hash functions

Purpose of this report:

Thus, lead to our aim of creating this report, **to inform the reader's decision on whether ASCON-Hash is a suitable hashing solution for securing FOTA deployments.**

11.2 What make a robust FOTA

#	Key Criterion	Reason
1	Cryptographic Integrity & Auth.	Blocks malicious or corrupted firmware
2	Rollback / Replay Protection	Prevents or mistake downgrading to vulnerable versions
3	Fail-Safe / Power-Loss Resilience	If power cuts out mid-update, the device still boots. No bricking
4	Pause-and-resume & Delta Capability	Downloads can pick up where they left off Saves bandwidth and energy for large images
5	Low Resource Footprint	Fits bootloaders under 32 kB and won't drain batteries
6	Secure Transport & Key Management	Blocks eavesdroppers, supports long-life key rotation
7	Certification & Auditability	Proves the process meets industry safety/security standards

criteria capture what practitioners routinely score when choosing or designing a FOTA pipeline [19].

Table 1

11.3 Why ASCON-Hash is suited to FOTA

Criterion *(table 1)*	How ASCON-Hash Delivers
1	256-bit digest with 128-bit pre-image & collision strength no practical attacks beyond 4 of 12 rounds of permutation
2	(Partly) Hash itself can't block downgrades, but its low CPU cost lets the bootloader verify version-tagged manifests without inflating flash or battery budget.
3	Short verification time and small code leave space for dual-bank firmware, and also quicker hash means device switches banks sooner, reducing the unsafe window.
4	Low-cost hashing makes per-chunk or Merkle-tree digests practical, enabling interrupted downloads to resume.
5	Complied code size around 1.7KB flash, while running take up less than 100B RAM, ideal for 32 kB bootloaders.
6	Same 320-bit permutation powers both ASCON-AEAD (encrypted channel) and ASCON-Hash (integrity); one firmware core covers both needs, simplify key provisioning and audit.
7	(Partly) NIST-selected LWC standard (2023) and now published as SP 800-232 draft. No FIPS140-3 module yet, use in federal systems and other regulated sectors have to wait

Data from multiple sources [20, 21].

11.4 ASCON-Hash vs. industry default SHA-256

FOTA Criterion	SHA-256	ASCON-Hash (256-bit)
Integrity & authenticity (collision / pre-image strength)	128-bit collision, 256-bit pre-image; no practical attacks.	128-bit collision & pre-image margin
Rollback / replay defence (cost of hashing)	~ 225 cycles/byte on M3	~ 58 cycles/byte on M3
Fail-safe / unsafe window (time to verify 32 KB before A/B swap)	~ 8 ms on M3.	~ 1.9 ms on M3
Pause-and-resume (hash per 4 KB block)	~ 1 ms on M0+	0.24 ms on M0+
Low resource footprint	Flash ~ 3–4 KB, RAM ~ 1 KB	Flash ~ 1.7 KB, RAM < 100 B
transport & key mgmt. (channel + hash)	Needs AES + SHA-256 two separate primitives.	Same 320-bit permutation powers ASCON-AEAD and Hash
Certification & auditability	Widely FIPS-140-2/3 certified; ubiquitous tool-chain support.	NIST LWC standard (SP 800-232 draft

Data from multiple sources [20, 21, 22].

From the table we can see, **ASCON-Hash outperforms SHA-256 on speed, energy, and code/RAM footprint** (criteria 2–6) while matching its 128-bit collision security (criterion 1).

SHA-256 still wins on mature certifications and universal library support (criterion 7).

12. Recommendations

12.1 Recommendation on Start Piloting ASCON-Hash

Our survey shows that **ASCON-Hash now offers a well-tested, NIST-standardised alternative to SHA-256** with dramatic savings in code size, energy, and verification time, while maintaining a 128-bit security margin. The technology is no longer experimental, reference software, open RTL cores, and third-party cryptanalysis are publicly available and stable.

Practical Recommendation — what you can do

- Pilot projects: integrate ASCON-Hash in new bootloaders or low-risk product lines first, keeping SHA-256 as a fallback.
- Shared primitive strategy: where devices already plan to use ASCON-AEAD for secure channels, reuse the same permutation for hashing and remove the SHA-2 code, cutting flash usage.
- Certification roadmap: follow NIST SP 800-232 progress and track forthcoming FIPS 140-3 modules so pilots can transition smoothly to certified builds.

Adopting ASCON-Hash incrementally lets organisations capture its performance benefits today without waiting for a complete ecosystem shift.

12.2 Recommendation on Research & Deployment: Full-scale FOTA prototype

No published work yet shows ASCON-Hash running inside a complete, fail-safe firmware-update pipeline. A production-grade prototype would prove its practical gains, demonstrates real-world throughput, power draw, and failure-handling with ASCON-Hash in place of SHA-256. Thus, uncover any integration hurdles before industry-wide adoption.

Delivering this prototype would close the current research gap and give vendors a ready-made blueprint for migrating away from SHA-256 in future FOTA deployments.

13. Conclusion

ASCON-Hash is no longer an experimental curiosity: it is the NIST-selected lightweight hash standard [20] and has already been stress-tested through third-party cryptanalysis, open reference code, and multiple hardware cores. Compared with today default FOTA SHA-256, ASCON-Hash delivers:

- ~ 4 × faster verification and ~ 4 × lower energy on Cortex-M cores,
- about half the flash code size (~ 1.7 KB vs ~ 3-4 KB) and a tiny RAM footprint (< 100 B), and
- a matching 128-bit collision, pre-image security margin, with no practical full-round attacks known.

These gains directly strengthen five of the seven practitioner criteria for a robust firmware-update pipeline. The only areas where SHA-256 still leads are mature certification and universal tool-chain presence. That certification gap is narrowing: NIST has begun the validation pathway for the new LWC portfolio, and vendors can anticipate FIPS-compliant ASCON modules within the next product cycle.

Bottom line:
For commercial IoT and other constrained devices, ASCON-Hash is an immediately deployable upgrade path that cuts verification cost without sacrificing security. Organisations can adopt it incrementally, starting with pilot bootloaders or systems that already use ASCON-AEAD, while keeping SHA-256 as a fallback until FIPS validation arrives. Whether you transition now or later, the evidence shows that lightweight hashing has matured; the decision is no longer can ASCON-Hash secure FOTA, but when you choose to leverage its clear efficiency advantages.

References

[1] Frse, B. B. O. (2023, May 25). *ASCON is a light-weight champion - ASecuritySite: When Bob Met Alice*. Medium. https://medium.com/asecuritysite-when-bob-met-alice/ascon-is-a-light-weight-champion-bfd81853d61a

[2] Jenkins, I. R. (2020). *Defense in depth of resource-constrained devices* (Doctoral dissertation, Dartmouth College). Dartmouth College Ph.D. Dissertations, 59. https://digitalcommons.dartmouth.edu/dissertations/59

[3] Kim, Y. B., Youn, T., & Seo, S. C. (2021). Chaining optimization methodology: A new SHA-3 implementation on low-end microcontrollers. *Sustainability, 13*(8), 4324. https://doi.org/10.3390/su13084324

[4] Li, H., Kumar, V., Park, J., & Yang, Y. (2021). Cumulative message authentication codes for resource-constrained IoT networks. *IEEE Internet of Things Journal, 8*(15), 11847–11859. https://doi.org/10.1109/jiot.2021.3074054

[5] OWASP Foundation. (n.d.). *Threat modeling*. https://owasp.org/www-community/Threat_Modeling

[6] Tewari, R. (2022, January 24). *SHA-3 hash construction*. Secure Machinery. https://securemachinery.com/2017/08/19/keccak-or-sha-3/

[7] The Ultimate Guide to FOTA: Firmware Over-The-Air explained. (n.d.). *Cavli Wireless*. https://www.cavliwireless.com/blog/nerdiest-of-things/fota-why-update-firmware-over-the-air

[8] What is FOTA? (n.d.). *HMD Global*. https://www.hmd.com/en_au/support/topics/software-and-updates/what-is-fota

[9] Why firmware integrity is insufficient for effective threat detection and hunting. (n.d.). *Binarly.io*. https://www.binarly.io/blog/why-firmware-integrity-is-insufficient-for-effective-threat-detection-and-hunting

[10] Turan, M. S., McKay, K., Chang, D., Bassham, L. E., Kang, J., ... & Hong, D. (2023). *Status report on the final round of the NIST lightweight cryptography standardization process*. US Department of Commerce, National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/ir/2023/NIST.IR.8454.pdf

[11] Dobraunig, C., Eichlseder, M., Mendel, F., & Schläffer, M. (2021). Ascon v1.2: Lightweight Authenticated Encryption and Hashing. *Journal of Cryptology*, *34*(3). https://doi.org/10.1007/s00145-021-09398-9

[12] El-Hadedy, M., Guo, X., Yoshii, K., Cai, Y., Herndon, R., Banta, B., & Hwu, W.-M. (2023). RECO-ASCON: Reconfigurable ASCON hash functions for IoT applications. *Integration (Amsterdam)*, *93*, 102061-. https://doi.org/10.1016/j.vlsi.2023.102061

[13] Zhai, D., Bai, W., Fu, J., Gao, H., & Zhu, X. (2024). Improved 2-round collision attack on IoT hash standard ASCON-HASH. *Heliyon*, *10*(5), e26119–e26119. https://doi.org/10.1016/j.heliyon.2024.e26119

[14] Sönmez Turan, M., McKay, K., Chang, D., Kang, J., & Kelsey, J. (2024). *Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions* (No. NIST Special Publication (SP) 800-232 (Draft)). National Institute of Standards and Technology. https://csrc.nist.gov/pubs/sp/800/232/ipd

[15] Sarasa Laborda, V., Hernández-Álvarez, L., Hernández Encinas, L., Sánchez García , J. I., & Queiruga-Dios, A. (2025). Study About the Performance of Ascon in Arduino Devices. *Applied Sciences*, *15*(7), 4071-. https://doi.org/10.3390/app15074071

[16] Windarta, S., Suryadi, S., Ramli, K., Pranggono, B., & Gunawan, T. S. (2022). Lightweight cryptographic hash functions: Design trends, comparative study, and future directions. *Ieee Access*, *10*, 82272-82294. https://ieeexplore.ieee.org/abstract/document/9846993

[17] Park, C. Y., Lee, S. J., & Lee, I. G. (2025). Secure and Lightweight Firmware Over-the-Air Update Mechanism for Internet of Things. *Electronics*, *14*(8), 1583. https://www.mdpi.com/2079-9292/14/8/1583

[18] Neve, R.P., Bansode, R. (2023). Performance Evaluation of Lightweight ASCON-HASH Algorithm for IoT Devices. In: Balas, V.E., Semwal, V.B., Khandare, A. (eds) Intelligent Computing and Networking. IC-ICN 2023. Lecture Notes in Networks and Systems, vol 699. Springer, Singapore. https://doi.org/10.1007/978-981-99-3177-4_25

[19] Saad El Jaouhari, Eric Bouvet, Secure firmware Over-The-Air updates for IoT: Survey, challenges, and discussions, Internet of Things, Volume 18,2022,100508, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2022.100508.

[20] S. Meltem, K. Turan, D. Mckay, J. Chang, J. Kang, and Kelsey, "NIST Special Publication 800 NIST SP 800-232 ipd Ascon-Based Lightweight Cryptography Standards for Constrained Devices Authenticated Encryption, Hash, and Extendable Output Functions Initial Public Draft," doi: https://doi.org/10.6028/NIST.SP.800-232.ipd.

[21] M. S. Turan, "Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process," *Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process*, 2023, doi: https://doi.org/10.6028/nist.ir.8454.

[22] W. Ross, "FIPS PUB 140-3 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION," *Security Requirements for Cryptographic Modules*, 2019, doi: https://doi.org/10.6028/NIST.FIPS.140-3.