

# Method of cascaded lightweight hashing for digital twin data streams based on cryptographic message chaining to detect unauthorized data alteration`

Inna Rozlomii<sup>1,\*</sup> and Serhii Naumenko<sup>2,†</sup>

<sup>1</sup> Cherkasy State Technological University, 460, Shevchenko Blvd., Cherkasy, 18006, Ukraine

<sup>2</sup> Bohdan Khmelnytsky National University of Cherkasy, 81, Shevchenko Blvd., Cherkasy, 18031, Ukraine

## Abstract

This paper presents a Cascaded Lightweight Hashing (CLH) method for protecting the integrity of digital twin telemetry streams in resource-constrained embedded systems. The proposed approach introduces the Cryptographic Message Chaining (CMC) mechanism, which establishes cryptographic dependencies between consecutive telemetry packets, enabling continuous integrity verification of the entire data stream rather than independent packet validation. The method incorporates a modified telemetry packet structure containing a previous hash value, timestamp, and sequence number to detect packet modification, insertion, deletion, replay, and reordering attacks. The proposed solution was implemented on an experimental platform based on STM32F407, ESP32-S3, FreeRTOS, and ASCON-Hash256, and evaluated using telemetry streams of 10,000 packets with payload sizes of 64, 128, and 256 bytes. Experimental results demonstrated that the proposed mechanism increased the average packet processing time by only 6.2%, RAM consumption by 7.7%, Flash memory usage by 1.9%, and CPU utilization by 6.2% while preserving linear computational complexity and constant memory consumption. The average Precision, Recall, and F1-score exceeded 99%, confirming reliable detection of telemetry manipulation attacks. Compared with conventional independent hashing, the proposed method additionally detects packet insertion, deletion, replay, and reordering attacks while maintaining acceptable communication overhead, which decreases from 68.8% for 64-byte packets to 17.2% for 256-byte packets. The obtained results demonstrate the suitability of the proposed method for digital twin, industrial IoT, medical IoT, smart grid, and edge computing applications.

## Keywords

Lightweight hashing, cryptographic message chaining, digital twin, telemetry integrity, embedded systems, lightweight cryptography

## 1. Introduction

The rapid adoption of digital twin technology has significantly transformed the operation of cyber-physical systems by enabling continuous synchronization between physical devices and their virtual counterparts [1]. Digital twins receive telemetry from embedded devices, sensors, industrial controllers, and edge nodes in real time, creating dynamic virtual representations that support monitoring, predictive maintenance, process optimization, and autonomous decision-making [2]. Their widespread application in industrial automation, smart manufacturing, intelligent transportation, healthcare, and energy systems has considerably increased the importance of protecting continuously transmitted telemetry streams from cyber threats [3]–[5].

Unlike conventional information systems, digital twins operate with uninterrupted flows of sensor measurements whose integrity directly influences the correctness of virtual models and subsequent control actions [6]. Any unauthorized modification of telemetry data may result in an incorrect representation of the physical process, propagation of erroneous analytical results, degradation of decision-making quality, or unsafe control commands delivered to physical equipment [7], [8]. Consequently, preserving the integrity of streaming telemetry has become one of the key security requirements for trustworthy digital twin infrastructures. Existing approaches mainly rely on conventional cryptographic hash functions, digital signatures, message

ADT-ISMDDC 2026: *Advanced Digital Technologies: Intelligent Systems, Modeling, and Data-Driven Control*, September 22–24, 2026, Odesa, Ukraine

\* Corresponding author.

† These authors contributed equally.

✉ inna-roz.ukr.net (I. Rozlomii); sergey.naumenko94@vu.cdu.edu.ua (S. Naumenko);

ORCID 0000-0001-5065-9004 (I. Rozlomii); 0000-0002-6337-1605 (S. Naumenko);

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

published 2026-10-22

authentication codes, or blockchain-based verification mechanisms [9]–[11]. Although these solutions provide a high level of integrity protection, many of them introduce considerable computational overhead, increased memory consumption, communication latency, or additional storage requirements. Such characteristics limit their applicability in resource-constrained embedded devices that constitute the primary data sources for digital twins [12]. Furthermore, most existing integrity verification mechanisms process telemetry packets independently, making them less effective in detecting packet insertion, deletion, reordering, or selective modification attacks occurring within continuous data streams.

Recent advances in lightweight cryptography have created new opportunities for protecting data generated by embedded systems with limited computational resources [13]. However, current lightweight integrity mechanisms are predominantly designed for independent message authentication and do not explicitly exploit the sequential nature of telemetry transmission between physical devices and digital twins [14]. As a result, they provide limited capabilities for identifying sophisticated attacks targeting the continuity and consistency of streaming data.

An alternative approach is to establish cryptographic dependencies between consecutive telemetry messages, where the integrity of each packet is linked to previously transmitted data through a cascaded hashing mechanism. Such organization transforms an independent sequence of packets into a cryptographically connected stream, enabling rapid detection of unauthorized data alteration while maintaining low computational complexity and minimal memory requirements suitable for embedded platforms.

Despite the growing interest in digital twin security, the development of lightweight mechanisms specifically designed for continuous telemetry streams remains insufficiently investigated. Existing studies primarily focus on communication security, access control, or authentication, whereas considerably less attention has been devoted to efficient integrity verification mechanisms capable of operating under strict resource limitations while preserving real-time performance.

The aim of this paper is to develop a method of cascaded lightweight hashing for digital twin data streams based on cryptographic message chaining that enables efficient detection of unauthorized data alteration while maintaining low computational overhead, reduced memory consumption, and suitability for resource-constrained embedded systems.

## 2. Theoretical background and related works

The security of digital twin infrastructures has become a major research area in cyber-physical systems due to the growing reliance of industrial, medical, transportation, and energy applications on continuously synchronized virtual representations of physical assets. Since digital twins process uninterrupted telemetry streams generated by embedded devices, sensors, and edge platforms, maintaining telemetry integrity is essential for reliable monitoring, prediction, diagnostics, and autonomous control. Even minor unauthorized modifications of telemetry data may propagate through the digital model, resulting in inaccurate system state estimation and incorrect operational decisions.

Existing integrity protection mechanisms mainly rely on cryptographic hash functions, message authentication codes (MAC), authenticated encryption, and digital signatures [15], [16]. Cryptographic hash functions efficiently detect data modification, while MAC and digital signatures additionally provide authentication and message authenticity [17]. Authenticated encryption combines confidentiality and integrity protection within a single cryptographic primitive and has therefore become widely adopted in modern IoT communication protocols [18]. However, these mechanisms generally treat every telemetry packet as an independent cryptographic object without

considering its relationship with previously transmitted messages. Their computational overhead also limits practical deployment in resource-constrained embedded systems.

Recent advances in lightweight cryptography have significantly improved the applicability of integrity protection in embedded devices by reducing computational complexity, memory consumption, and energy requirements. Modern lightweight hash functions and authenticated encryption algorithms are specifically designed for microcontrollers with limited processing capabilities and have become important components of secure IoT architectures [19], [20]. Nevertheless, most lightweight integrity mechanisms remain packet-oriented and verify telemetry messages independently. As a result, they provide limited protection against packet insertion, deletion, replay, selective replacement, and reordering attacks, where the integrity of individual packets remains valid despite manipulation of the overall telemetry sequence.

To overcome these limitations, several studies have investigated hash chains, chained authentication mechanisms, sequential message authentication, and blockchain-inspired integrity verification for distributed systems [21], [22]. These approaches establish cryptographic relationships between consecutive messages and therefore improve resistance against stream-oriented attacks. However, classical hash chains frequently require continuous synchronization between communicating entities, while blockchain-based solutions introduce considerable storage requirements, communication latency, and computational overhead due to distributed consensus mechanisms. Such characteristics make these approaches unsuitable for continuous telemetry processing in resource-constrained embedded platforms operating under real-time constraints.

Another limitation of existing approaches is that they have been primarily developed for generic communication systems rather than for digital twin environments. Telemetry generated by digital twins possesses strong temporal continuity, where each measurement represents the next state of the monitored physical process. Existing packet-level integrity verification methods do not exploit this sequential dependency, leaving opportunities for sophisticated stream manipulation attacks that may remain undetected despite successful verification of individual packets.

Consequently, an integrity verification mechanism is still needed that combines the computational efficiency of lightweight cryptography with cryptographic linking of consecutive telemetry packets. Such a solution should preserve real-time operation, require only minimal additional computational and memory resources, and provide reliable detection of unauthorized data alteration, packet insertion, deletion, replay, and reordering attacks. Addressing these challenges forms the basis for the proposed Cascaded Lightweight Hashing (CLH) method based on the Cryptographic Message Chaining (CMC) mechanism, specifically designed for continuous digital twin telemetry streams.

### 3. Cascaded lightweight hashing method for digital twin data streams

#### 3.1. Design principles of cryptographic message chaining

Digital twin platforms continuously receive telemetry generated by distributed embedded devices that periodically transmit measurements describing the current state of physical processes. Unlike conventional file transmission, telemetry streams represent temporally ordered sequences of measurements whose semantic meaning depends not only on individual packets but also on their chronological order. Consequently, preserving the integrity of isolated packets alone cannot fully guarantee the correctness of the reconstructed digital twin state.

Traditional integrity verification mechanisms compute an independent cryptographic hash for every telemetry packet. Although this approach efficiently detects direct modification of packet contents, it cannot reliably identify attacks targeting the sequence of transmitted messages. An

attacker may remove one or several legitimate packets, insert fabricated measurements, replay previously intercepted messages, or modify the transmission order while maintaining valid integrity values for each individual packet. Since every packet remains cryptographically valid, conventional packet-level verification is often unable to recognize violations affecting the continuity of telemetry streams.

The proposed method addresses this limitation by introducing the concept of Cryptographic Message Chaining. Instead of treating telemetry packets as independent entities, consecutive messages become cryptographically interconnected. Each newly generated integrity value depends not only on the current telemetry packet but also on the cryptographic state established during processing of the previous transmission. Consequently, any unauthorized modification propagates through the chain, making inconsistencies immediately observable during integrity verification.

The proposed chaining mechanism provides protection against several common attack scenarios:

- unauthorized modification of packet contents;
- insertion of fabricated telemetry packets;
- deletion of legitimate packets;
- replay of previously transmitted messages;
- reordering of telemetry packets.

Unlike classical hash chains designed primarily for authentication protocols, the proposed CMC mechanism is optimized for continuous telemetry generated by embedded cyber-physical devices. It preserves low computational complexity while requiring only minimal additional memory for maintaining the current cryptographic state.

An important feature of the proposed method is that the integrity of each telemetry message becomes dependent on the integrity of the previous transmission. Therefore, compromising a single packet inevitably affects all subsequent verification results until the communication stream is successfully resynchronized. This property significantly increases resistance against selective packet manipulation while preserving real-time processing capabilities.

The operating principle of the proposed CMC mechanism, in which each telemetry packet is cryptographically linked with the previous transmission to form a continuous integrity chain, is illustrated in Figure 1.

As illustrated in Figure 1, the proposed method establishes a continuous cryptographic relationship between consecutive telemetry packets by incorporating the integrity value of the previous transmission into the hash computation of the current packet. This cascaded organization transforms independent telemetry messages into a cryptographically connected data stream, where each integrity value depends on both the current packet contents and the previously verified cryptographic state.

Unlike conventional packet-level hashing, the proposed CMC mechanism enables the digital twin verification module to validate not only the integrity of individual telemetry packets but also the consistency of the entire transmission sequence. Consequently, any unauthorized packet modification, insertion, deletion, replay, or reordering disrupts the cryptographic chain and is immediately detected during the verification process.

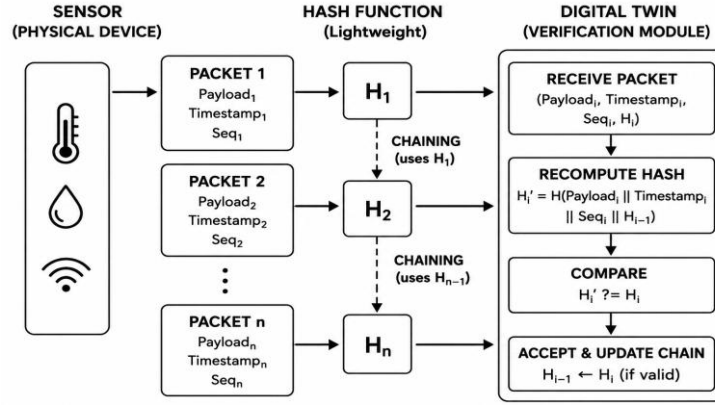


Figure 1: General architecture of cascaded lightweight hashing for Digital Twin telemetry streams

The proposed architecture preserves the computational simplicity of lightweight hash functions while significantly improving the ability to detect stream-oriented manipulation attacks without introducing additional cryptographic primitives or substantial communication overhead. This makes the method particularly suitable for resource-constrained embedded devices that continuously transmit telemetry data to digital twin platforms.

### 3.2. Proposed cascaded hash construction

The proposed cascaded hashing method introduces a new telemetry packet structure that extends conventional payload transmission with cryptographic chaining information. Each transmitted packet contains both application data and a minimal set of integrity-related parameters required to maintain the continuity of the cryptographic chain. Each telemetry packet is represented as (1).

$$P_i = \{Payload_i, TS_i, SN_i, H_{i-1}, H_i\}, \quad (1)$$

where  $Payload_i$  represents sensor measurements;  $TS_i$  denotes the timestamp;  $SN_i$  is the sequential packet identifier;  $H_{i-1}$  is the integrity value of the previous packet;  $H_i$  is the newly generated cascaded hash. Unlike independent hashing approaches, the proposed method generates each integrity value using both the current telemetry information and the cryptographic context inherited from the previous transmission. The cascaded integrity function is defined as (2).

$$H_i = \mathcal{H}(Payload_i || TS_i || SN_i || H_{i-1}), \quad (2)$$

where  $\mathcal{H}(\cdot)$  denotes the selected lightweight cryptographic hash function and  $||$  represents concatenation. The inclusion of timestamps and sequence numbers provides additional protection against replay and packet reordering attacks while preserving deterministic verification procedures. The key innovation of the proposed construction is not the sequential application of a hash function itself, but the integration of the previous cryptographic state with the payload, timestamp, and sequence number within each new integrity value. This creates two complementary dependencies: a content dependency between the telemetry data and its hash, and a temporal dependency between consecutive packets. Therefore, verification is performed at both the packet and stream levels. Unlike independent hashing, manipulation of an otherwise valid packet sequence changes the expected cryptographic state and can be detected without storing or reprocessing the complete telemetry history. The proposed integrity verification process consists of three consecutive stages. During the first stage, the transmitting embedded device forms the telemetry packet, generates the cascaded integrity value, and appends it to the outgoing message. During the second stage, the digital twin platform recalculates the expected integrity value using the received telemetry data together with the previously verified cryptographic state. During the third stage, the calculated and received integrity values are compared. If they coincide, the telemetry packet is accepted and the cryptographic chain is updated. Otherwise, the packet is marked as compromised and the synchronization recovery procedure is initiated. To maintain reliable operation in unstable

communication environments, the proposed method additionally introduces a lightweight resynchronization mechanism. When packet loss is detected, the receiver requests the latest valid cryptographic state instead of rebuilding the entire integrity chain. This significantly reduces recovery latency and minimizes additional communication overhead. The internal organization of the telemetry packet supporting the proposed CMC mechanism is presented in Table 1.

Table 1  
Structure of the proposed telemetry packet

| Field                       | Size (bytes) | Description                                                                                                                     |
|-----------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------|
| Payload                     | Variable     | Sensor measurements or telemetry data generated by the embedded device.                                                         |
| Timestamp                   | 8            | Time identifier indicating the generation moment of the telemetry packet.                                                       |
| Sequence Number             | 4            | Incremental packet identifier used to preserve the correct transmission order and detect missing or replayed packets.           |
| Previous Hash ( $H_{i-1}$ ) | 32           | Cascaded hash value calculated for the previous telemetry packet, providing cryptographic linkage between consecutive messages. |
| Current Hash ( $H_i$ )      | 32           | Lightweight cryptographic hash computed for the current packet using the proposed Cryptographic Message Chaining mechanism.     |

The proposed packet structure extends a conventional telemetry message with only three lightweight service fields while preserving compatibility with existing telemetry transmission architectures. Among these fields, the Previous Hash serves as the key element that establishes the cryptographic dependency between consecutive packets, whereas the Timestamp and Sequence Number strengthen protection against replay attacks and packet reordering. Despite the inclusion of additional integrity information, the proposed packet format introduces only a modest increase in communication overhead compared with conventional telemetry messages. At the same time, it enables continuous integrity verification throughout the entire telemetry stream rather than limiting verification to isolated packets. This design significantly enhances the ability to detect sequential manipulation attacks while remaining suitable for resource-constrained embedded platforms. From a theoretical perspective, the proposed construction transforms integrity verification from a set of independent packet checks into verification of an ordered cryptographic state sequence. Consequently, integrity becomes dependent not only on message content but also on its position and predecessor within the telemetry stream. This property extends the detectable manipulation space from direct packet modification to structural violations of the stream, including insertion, deletion, replay, and reordering. At the same time, maintaining only the latest valid hash state avoids storing the complete chain, which preserves constant state-memory requirements regardless of the telemetry stream length.

### 3.3. Computational complexity of the proposed method

The proposed cascaded hashing method was designed to satisfy the computational limitations of embedded platforms commonly used as telemetry sources for digital twin systems. The introduction of cryptographic message chaining increases neither the number of executed hash operations nor

the asymptotic computational complexity of integrity verification. Each telemetry packet still requires a single lightweight hash computation, while additional processing is limited to concatenation of several short metadata fields. Assuming a telemetry packet containing  $n$  bytes of application data, the computational complexity of the proposed method can be expressed as  $T(n) = O(n)$ , which is identical to the complexity of conventional independent hashing methods. Memory consumption remains constant throughout the transmission process because only the latest verified integrity value must be stored for subsequent computations. Consequently, the additional RAM requirement is independent of the telemetry stream length and can be expressed as  $M = O(1)$ . Similarly, the Flash memory required for implementing the proposed method differs only negligibly from a conventional lightweight hash implementation because no additional cryptographic primitives are introduced. The communication overhead is also minimal, consisting only of the previous integrity value together with several service fields required for synchronization. Compared with independent hashing, the proposed approach introduces a small increase in transmitted packet size while significantly improving resistance against sequential packet manipulation attacks. The additional metadata enables reliable detection of insertion, deletion, replay, and packet reordering without increasing the number of cryptographic hash evaluations. A theoretical comparison between the proposed cascaded hashing method and conventional independent hashing is presented in Table 2 considering computational complexity, memory requirements, communication overhead, and resistance to typical telemetry manipulation attacks.

Table 2  
Theoretical comparison of independent and cascaded hashing methods

| Criterion                                    | Independent Hashing | Proposed Cascaded Hashing      |
|----------------------------------------------|---------------------|--------------------------------|
| Computational complexity                     | $O(n)$              | $O(n)$                         |
| RAM requirement                              | $O(1)$              | $O(1)$                         |
| Flash memory requirement                     | Low                 | Low                            |
| Hash computations per packet                 | 1                   | 1                              |
| Additional communication overhead            | None                | Previous hash + service fields |
| Detection of packet modification             | Yes                 | Yes                            |
| Detection of packet insertion                | None                | Yes                            |
| Detection of packet deletion                 | None                | Yes                            |
| Detection of packet replay                   | Limited             | Yes                            |
| Detection of packet reordering               | Limited             | Yes                            |
| Integrity verification scope                 | Individual packet   | Entire telemetry stream        |
| Suitability for resource-constrained devices | High                | High                           |

The comparison demonstrates that the proposed cascaded hashing method preserves the computational efficiency of conventional independent hashing while substantially extending its security capabilities. By introducing cryptographic dependencies between consecutive telemetry

packets, the proposed approach enables reliable detection of sequential stream manipulation attacks with only minimal additional communication overhead, making it well suited for embedded digital twin applications.

## 4. Experimental implementation

### 4.1. Experimental setup

The experimental platform consisted of three interconnected components: an embedded telemetry generator, a wireless communication gateway, and a digital twin verification environment. The telemetry source was implemented on an STM32F407 microcontroller responsible for generating periodic sensor measurements and constructing telemetry packets according to the proposed packet format. An ESP32-S3 development board served as the communication gateway, forwarding telemetry packets to the digital twin platform through a Wi-Fi connection. The digital twin itself was implemented as a software emulator running on a personal computer, where integrity verification and cryptographic chain maintenance were performed. The embedded software was developed using FreeRTOS, enabling periodic telemetry generation and concurrent execution of communication and integrity-related tasks. Separate FreeRTOS tasks were allocated for sensor acquisition, packet formation, cryptographic hash computation, wireless transmission, and synchronization management. The proposed CMC mechanism was implemented using ASCON-Hash256, selected due to its lightweight design, efficient software implementation, and suitability for embedded devices with limited computational resources. The algorithm produces a fixed-length 256-bit digest while maintaining low memory consumption and computational overhead. Experimental evaluation was performed using telemetry streams consisting of 10,000 consecutive packets. Three packet sizes were considered in order to simulate different application scenarios: 64 bytes; 128 bytes; 256 bytes. Each telemetry packet contained sensor data, timestamp, sequence number, previous hash value, and the newly generated cascaded hash. During transmission, several attack scenarios were artificially introduced, including packet modification, packet insertion, packet deletion, replay attacks, and packet reordering. The overall architecture of the experimental platform used for telemetry transmission and integrity verification is illustrated in Figure 2.

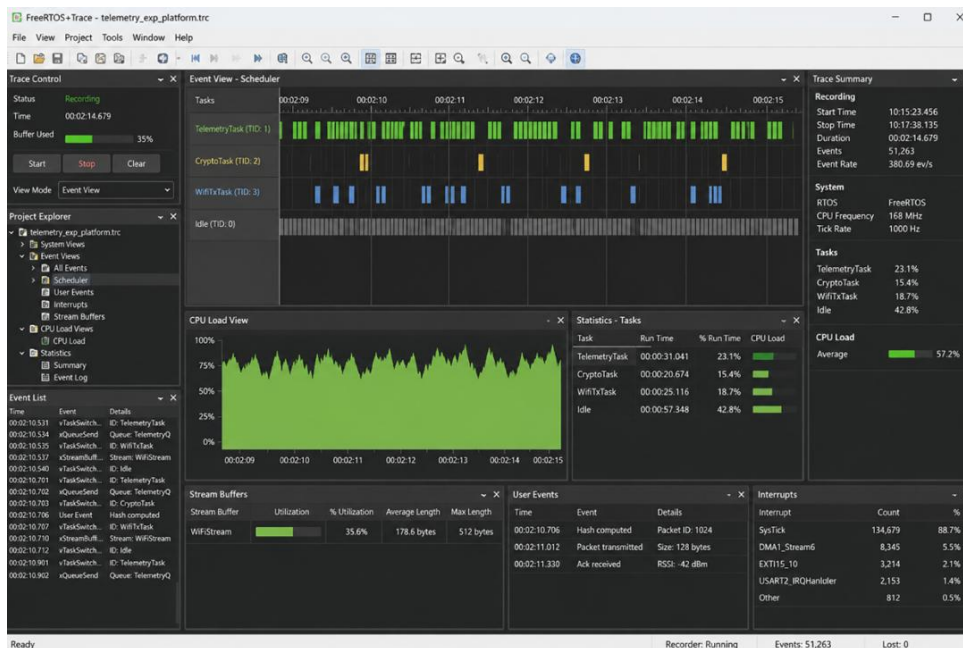


Figure 2: Experimental platform for telemetry transmission and integrity verification

After transmission, every received telemetry packet was processed by the digital twin verification module, where the proposed cryptographic message chaining mechanism recalculated the expected

integrity value and compared it with the received cryptographic hash before updating the internal chain state.

## 4.2. Implementation of cascaded hash verification

The proposed verification procedure was implemented as a sequential integrity validation process executed immediately after receiving every telemetry packet. Unlike conventional independent packet verification, the proposed method continuously maintains the cryptographic state of the telemetry stream, allowing integrity violations to propagate through subsequent packets until synchronization is restored. Upon reception of a telemetry packet, the digital twin verification module first extracts the application payload together with the associated timestamp, sequence number, previous hash, and current hash fields. The extracted information is then forwarded to the lightweight hashing module, where the expected integrity value is recalculated using the CMC algorithm described in Section 3. During the next stage, the verification module compares the received previous hash value with the internally stored cryptographic state established after successful verification of the preceding telemetry packet. This operation allows immediate identification of missing packets, duplicated transmissions, replay attempts, or packet reordering before the current integrity value is evaluated.

If the previous cryptographic state is successfully validated, the current cascaded hash is recomputed and compared with the received integrity value. A successful comparison confirms both the integrity of the current telemetry packet and the continuity of the cryptographic chain. Consequently, the internal verification state is updated by replacing the stored previous hash with the newly verified hash value.

Whenever an integrity mismatch is detected, the packet is rejected and the verification process enters the synchronization stage. Instead of reconstructing the complete telemetry history, the proposed method performs lightweight resynchronization by requesting the latest valid cryptographic state from the transmitting embedded device. This approach substantially decreases synchronization latency while avoiding unnecessary retransmission of previously verified telemetry packets. The sequence of operations performed during telemetry integrity verification using the proposed cascaded lightweight hashing method is presented in Figure 3.

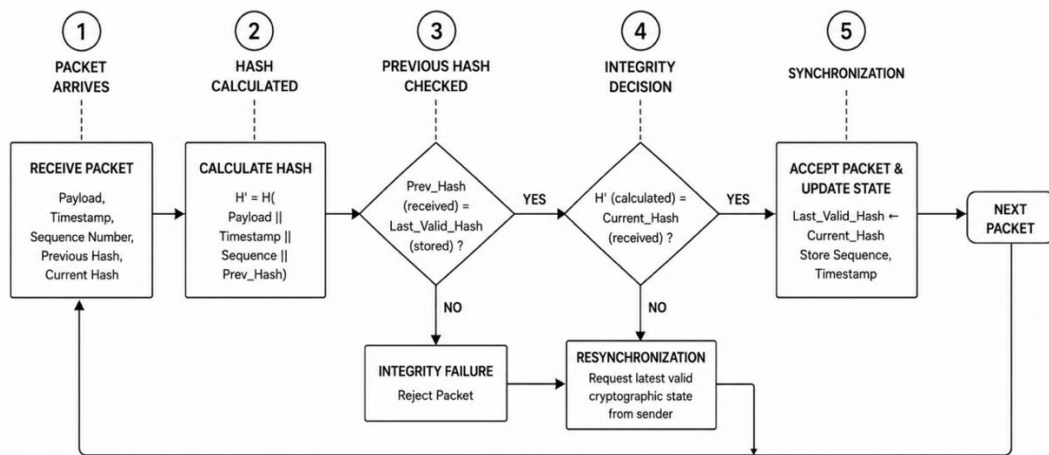


Figure 3: Flowchart of the proposed verification procedure

The proposed implementation requires only a single lightweight hash computation for each telemetry packet while maintaining a constant-size cryptographic state throughout the transmission session. Consequently, the verification procedure remains computationally efficient and suitable for continuous real-time telemetry processing in embedded digital twin environments.

## 5. Results and performance evaluation

### 5.1. Computational performance

The computational performance of the proposed Cascaded Lightweight Hashing (CLH) method was evaluated to determine its suitability for continuous telemetry processing in resource-constrained embedded systems. The evaluation considered four primary performance indicators: processing time, RAM consumption, Flash memory usage, and processor utilization. The obtained results were compared with a conventional independent hashing approach implemented using the same lightweight cryptographic hash function under identical experimental conditions. Each reported value represents the average obtained from processing 10,000 consecutive telemetry packets. The total processing time of the proposed method can be expressed as  $T_p = T_h + T_a$ , where  $T_p$  is the total processing time,  $T_h$  is the execution time of the lightweight hash function, and  $T_a$  is the additional time required for cryptographic message chaining, including previous hash verification and chain state updating. For the conventional independent hashing method,  $T_i = T_h$ . The relative computational overhead introduced by the proposed method is calculated as (3).

$$O = T_p - T_i \times 100\% \quad (3)$$

The total RAM consumption is determined by  $M = M_b + M_c$ , where  $M_b$  denotes the memory allocated for telemetry buffers and  $M_c$  represents the memory required to maintain the cryptographic chain state. Similarly, the processor utilization is calculated as  $U = T_p P \times 100\%$ , where  $P$  is the telemetry transmission period. The computational characteristics obtained during the experiments are summarized in Table 3.

Table 3  
Computational performance comparison

| Performance metric                 | Independent Hashing | Proposed Cascaded Hashing | Overhead (%) |
|------------------------------------|---------------------|---------------------------|--------------|
| Average processing time ( $\mu$ s) | 142.8               | 151.6                     | 6.2          |
| RAM usage (bytes)                  | 416                 | 448                       | 7.7          |
| Flash memory (kB)                  | 31.4                | 32.0                      | 1.9          |
| CPU utilization (%)                | 17.8                | 18.9                      | 6.2          |
| Hash computations per packet       | 1                   | 1                         | 0            |
| Computational complexity           | $O(n)$              | $O(n)$                    | 0            |

The experimental results indicate that the proposed CLH method introduces only a modest computational overhead while preserving the same asymptotic computational complexity as conventional independent hashing. The increase in processing time and memory consumption results exclusively from maintaining the cryptographic chain state rather than from additional hash evaluations. Consequently, the proposed method remains well suited for real-time telemetry protection in embedded digital twin environments.

### 5.2. Integrity verification performance

The effectiveness of the proposed CLH method was evaluated by simulating four common telemetry manipulation attacks, namely packet modification, packet insertion, packet deletion, and

replay attacks. A total of 10,000 telemetry packets were transmitted during each experiment, with approximately 5% of the packets intentionally manipulated to emulate malicious activity. The remaining packets represented legitimate telemetry transmissions.

The primary evaluation metric was the Detection Rate (DR), defined as (4).

$$DR = \frac{N_d}{N_a} \times 100\%, \quad (4)$$

where  $N_d$  denotes the number of correctly detected attacks and  $N_a$  represents the total number of simulated attacks. To provide a more comprehensive assessment of the verification procedure, the Precision (P), Recall (R), and F1-score ( $F_1$ ) were additionally calculated as (5).

$$\begin{aligned} P &= \frac{TP}{TP+FP} \\ R &= \frac{TP}{TP+FN}, \\ F_1 &= \frac{2PR}{P+R} \end{aligned} \quad (5)$$

where TP denotes true positives, FP false positives, and FN false negatives.

The experimental evaluation demonstrated that the proposed CMC mechanism reliably detected all considered telemetry manipulation scenarios while maintaining a very low false alarm rate. The highest detection accuracy was observed for packet modification and replay attacks because any alteration immediately disrupted the continuity of the cryptographic chain. Packet insertion and deletion attacks were also successfully identified due to inconsistencies between the received and expected cryptographic states. The detection rates obtained for different telemetry manipulation attacks are presented in Figure 4.

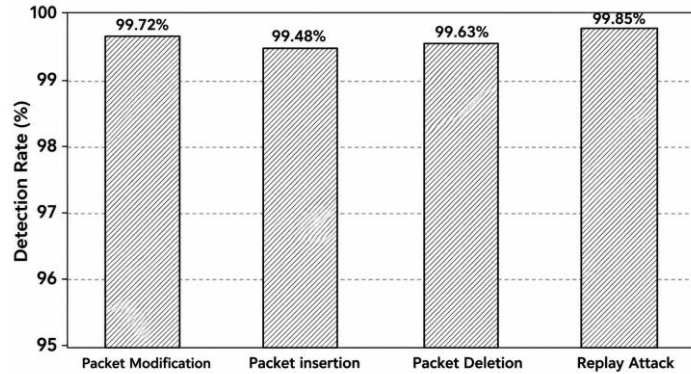


Figure 4: Detection rate of different telemetry manipulation attacks

The average Precision, Recall, and F1-score exceeded 99%, confirming that the proposed method provides highly reliable integrity verification while introducing only a modest computational overhead. These results demonstrate that incorporating cryptographic dependencies between consecutive telemetry packets substantially improves the capability to detect stream-oriented attacks compared with conventional independent hashing.

### 5.3. Communication overhead analysis

An important characteristic of any integrity verification mechanism for embedded systems is the amount of additional communication overhead introduced into the telemetry stream. Since wireless embedded devices typically operate under strict bandwidth and energy constraints, minimizing the size of transmitted service information is essential for maintaining efficient communication.

The communication overhead of the proposed method is determined by the additional service fields appended to each telemetry packet. The total overhead is calculated as (6).

$$S_o = S_h + S_t + S_q, \quad (6)$$

where  $S_h$  denotes the size of the previous hash field,  $S_t$  is the timestamp size, and  $S_q$  represents the sequence number field. The relative communication overhead is expressed as (7).

$$O_c = \frac{S_o}{S_p} \times 100\%, \quad (7)$$

where  $S_p$  denotes the payload size of the original telemetry packet.

For the implemented prototype  $S_h = 32$  bytes,  $S_t = 8$  bytes,  $S_q = 4$  bytes resulting in  $S_o = 44$  bytes.

Accordingly, the communication overhead decreases as the payload size increases. For example, telemetry packets containing 64, 128, and 256 bytes of application data introduce overheads of 68.8%, 34.4%, and 17.2%, respectively.

To evaluate the efficiency of the proposed approach, the communication overhead was compared with two commonly used integrity protection mechanisms based on conventional SHA-256 hashing and MAC. While SHA-256 requires transmission of only the hash value and MAC additionally requires an authentication tag, the proposed CLH method incorporates cryptographic chaining information that enables reliable detection of sequential telemetry manipulation attacks.

The relationship between communication overhead and telemetry packet size for different integrity protection mechanisms is illustrated in Figure 5.

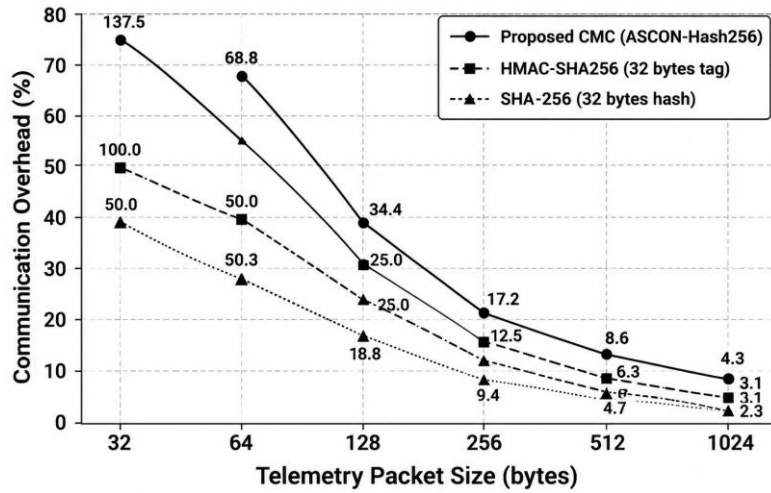


Figure 5: Communication overhead versus telemetry packet size

Although the proposed method introduces a fixed amount of additional integrity information, the relative communication overhead rapidly decreases with increasing telemetry packet size. At the same time, the additional service fields provide significantly stronger protection against packet insertion, deletion, replay, and reordering than conventional independent hashing methods. Consequently, the proposed CMC mechanism offers a favorable balance between communication efficiency and telemetry stream integrity, making it suitable for digital twin applications operating on resource-constrained embedded platforms.

## 6. Discussion

The proposed CLH method enhances conventional packet-level integrity verification by establishing cryptographic dependencies between consecutive telemetry messages. Unlike independent hashing, where each packet is verified separately, the proposed CMC mechanism preserves the integrity of the entire telemetry stream. Consequently, unauthorized modification of a single packet propagates through subsequent integrity values, enabling detection of inconsistencies that remain unnoticed with conventional independent hashing.

The proposed approach is particularly suitable for applications requiring continuous telemetry transmission and real-time operation. Digital twin platforms, autonomous cyber-physical systems, and distributed edge computing environments continuously exchange sequential telemetry data, making them vulnerable to packet insertion, deletion, replay, and reordering attacks. Under these conditions, cryptographic chaining significantly improves integrity protection while introducing only a modest computational overhead. An additional advantage of the proposed method is its independence from a specific lightweight hash algorithm. Although the experimental implementation employed ASCON-Hash256, the CMC mechanism can be integrated with other lightweight hash functions without changing its operating principles. Since only the latest cryptographic state is maintained, memory consumption remains constant and each packet requires a single hash evaluation, allowing efficient processing of high-frequency telemetry streams.

The proposed method requires synchronization between communicating devices; therefore, unstable communication channels may increase recovery overhead despite the lightweight resynchronization mechanism. Furthermore, the method focuses exclusively on telemetry integrity and should be combined with complementary mechanisms providing confidentiality and authentication. The proposed approach is applicable to IIoT, MIIoT, smart grid, digital twin, and edge computing environments, where trustworthy telemetry directly affects monitoring and autonomous decision-making. Future work will investigate hardware acceleration using FPGA, integration with post-quantum lightweight hash algorithms, and support for multichannel telemetry verification.

## 7. Conclusions

This paper proposed a CLH integrity of digital twin telemetry streams based on the newly introduced CMC mechanism. Unlike conventional independent packet hashing, the proposed approach establishes cryptographic dependencies between consecutive telemetry packets, enabling verification of both individual packet integrity and the consistency of the entire telemetry stream.

The proposed method was implemented on an experimental platform based on STM32F407, ESP32-S3, FreeRTOS, and ASCON-Hash256. Experimental evaluation demonstrated that the additional cryptographic chaining increased the average packet processing time by only 6.2%, RAM consumption by 7.7%, Flash memory usage by 1.9%, and CPU utilization by 6.2%, while preserving linear computational complexity  $O(n)$  and constant memory consumption  $O(1)$ . Compared with conventional independent hashing, the proposed method additionally detects packet insertion, deletion, replay, and reordering attacks that cannot be reliably identified when packets are verified independently. Experimental results obtained from 10,000 telemetry packets confirmed a detection accuracy exceeding 99% with only a modest communication overhead, which decreases from 68.8% for 64-byte packets to 17.2% for 256-byte packets as the telemetry payload increases. The obtained results demonstrate that the proposed CMC mechanism provides a favorable balance between computational efficiency and telemetry integrity protection, making it suitable for digital twin, IIoT, MIIoT, smart grid, and edge computing applications. The proposed method also establishes a flexible foundation for future integration with hardware cryptographic accelerators, FPGA implementations, post-quantum lightweight hash algorithms, and multichannel telemetry verification.

## Acknowledgements

This research was funded by the Ministry of Education and Science of Ukraine under grant 0125U000637.

## Declaration on Generative AI

Generative AI tools were partially used for English language refinement, grammar correction, and preparation of the graphical materials presented in Figures 4 and 5. The charts were generated using AI-assisted visualization tools based exclusively on the experimental results obtained by the authors during the conducted experiments. The authors take full responsibility for the scientific content, methodology, interpretation of the results, and accuracy of all information presented in this paper.

## References

- [1] C. Qian, X. Liu, C. Ripley, M. Qian, F. Liang, W. Yu, Digital twin – cyber replica of physical things: Architecture, applications and future research directions, *Future Internet* 14 (2) (2022) 64. DOI: 10.3390/fi14020064
- [2] V. M. Nadella, Digital Twins for predictive network management and system simulation, *International Journal of AI, BigData, Computational and Management Studies* 3 (3) (2022) 100–111. DOI: 10.63282/3050-9416.IJAIBDCMS-V3I3P112
- [3] M. R. Kabir, D. Halder, S. Ray, Digital twins for IoT-driven energy systems: A survey, *IEEE Access* 12 (2024) 177123–177143. DOI: 10.1109/ACCESS.2024.3506660
- [4] E. Faure, I. Rozlomii, S. Naumenko, Hybrid digital twin-driven anomaly detection in IoT telemetry using LSTM autoencoder, in: *Proceedings of the International Workshop on Data Analytics (WDA'26)*, Kyiv, Ukraine, 2026, pp. 76–89. DOI: 10.5281/zenodo.21155695
- [5] W. A. Ali, M. P. Fanti, M. Roccotelli, L. Ranieri, A review of digital twin technology for electric and autonomous vehicles, *Applied Sciences* 13 (10) (2023) 5871. DOI: 10.3390/app13105871
- [6] A. A. S. Alsubaih, K. Sepehrnoori, M. Delshad, A. Alsaedi, A comprehensive review of well integrity challenges and digital twin applications across conventional, unconventional, and storage wells, *Energies* 18 (17) (2025) 4757. DOI: 10.3390/en18174757
- [7] I. Rozlomii, A. Yarmilko, S. Naumenko, Innovative resource-saving security strategies for IoT devices, *Journal of Edge Computing* 4 (1) (2025) 35–56. DOI: 10.55056/jec.748
- [8] K. Kose, N. A. Kose, F. Liang, Securing unmanned devices in critical infrastructure: A survey of hardware, network, and swarm intelligence, *Electronics* 15 (6) (2026) 1204. DOI: 10.3390/electronics15061204
- [9] S. E. Nouma, A. A. Yavuz, Trustworthy and efficient digital twins in post-quantum era with hybrid hardware-assisted signatures, *ACM Transactions on Multimedia Computing, Communications and Applications* 20 (6) (2024) 1–30. DOI: 10.1145/3638250
- [10] D. Gautam, G. Thakur, P. Kumar, A. K. Das, Y. Park, Blockchain assisted intra-twin and inter-twin authentication scheme for vehicular digital twin system, *IEEE Transactions on Intelligent Transportation Systems* 25 (10) (2024) 15002–15015. DOI: 10.1109/TITS.2024.3394438

- [11] R. Suleiman, A. Maradapu Vera Venkata Sai, W. Yu, C. Wang, Blockchain for security in digital twins, *Future Internet* 17 (9) (2025) 385. DOI: 10.3390/fi17090385
- [12] Q. Tang, B. Wu, A digital twin-driven multi-resource constrained location system for resource allocation, *The International Journal of Advanced Manufacturing Technology* 130 (9) (2024) 4359–4385. DOI: 10.1007/s00170-023-12886-w
- [13] H. M. Kamdjou, D. Baudry, V. Havard, S. Ouchani, Resource-constrained extended reality operated with digital twin in industrial internet of things, *IEEE Open Journal of the Communications Society* 5 (2024) 928–950. DOI: 10.1109/OJCOMS.2024.3356508
- [14] L. U. Khan, Z. Han, W. Saad, E. Hossain, M. Guizani, C. S. Hong, Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities, *IEEE Communications Surveys & Tutorials* 24 (4) (2022) 2230–2254. DOI: 10.1109/COMST.2022.3198273
- [15] A. Yarmilko, I. Rozlomii, H. Kosenyuk, Hash method for information stream's safety in dynamic cooperative production system, in: *Mathematical Modeling and Simulation of Systems, Proceedings of the 16th International Scientific-Practical Conference (MODS 2021)*, Springer International Publishing, Cham, 2022, pp. 173–183. DOI: 10.1007/978-3-030-89902-8\_14
- [16] Q. Wang, W. Wu, L. Qian, Y. Cai, J. Qian, L. Meng, Design and implementation of secure and reliable information interaction architecture for digital twins, *China Communications* 20 (2) (2023) 79–93. DOI: 10.23919/JCC.2023.02.006
- [17] S. Mitsenko, S. Naumenko, A. Yarmilko, Information protection and recovery Hamming codes based' hash technique, in: *Proceedings of the 11th International Conference on Information Control Systems & Technologies (ICST 2023)*, CEUR Workshop Proceedings, vol. 3513, 2023, pp. 64–67. DOI: 10.5281/zenodo.21066947
- [18] S. Ali, F. Anwer, Secure IoT framework for authentication and confidentiality using hybrid cryptographic schemes, *International Journal of Information Technology* 16 (4) (2024) 2053–2067. DOI: 10.1007/s41870-024-01753-w
- [19] N. F. Mufidah, H. H. Nuha, Performance and security analysis of lightweight hash functions in IoT, *Jurnal Informatika: Jurnal Pengembangan IT* 9 (3) (2024) 264–270. DOI: 10.30591/jpit.v9i3.7633
- [20] M. Țălu, Securing IoT ecosystems: Experimental evaluation of modern lightweight cryptographic algorithms and their performance, *Journal of Cybersecurity* 7 (1) (2025) 565. DOI: 10.32604/jcs.2025.073690
- [21] B. Chowdhury, H. Jahankhani, S. Subramaniam, Zero-trust blockchain-based digital twin 6G AI-native conceptual framework against cyber attacks for e-healthcare, in: *International Conference on Global Security, Safety, and Sustainability*, Springer Nature Switzerland, Cham, 2023, pp. 453–479. DOI: 10.1007/978-3-031-82031-1\_23
- [22] D. A. Patil, S. G. A comprehensive survey on securing the social internet of things: Protocols, threat mitigation, technological integrations, tools, and performance metrics, *Scientific Reports* 15 (1) (2025) 40190. DOI: 10.1038/s41598-025-23865-4