Noname manuscript No.
(will be inserted by the editor)

Predictive-CSM: Lightweight Fragment Security for 6LoWPAN IoT Networks

Somayeh Sobati-Moghadam

Received: date / Accepted: date

Abstract Fragmentation is a routine part of communication in 6LoWPAN-based IoT networks, designed to accommodate small frame sizes on constrained wireless links. However, this process introduces a critical vulnerability: fragments are typically stored and processed before their legitimacy is confirmed, allowing attackers to exploit this gap with minimal effort.

In this work, we explore a defense strategy that takes a more adaptive, behavior-aware approach to this problem. Our system, called Predictive-CSM, introduces a combination of two lightweight mechanisms. The first tracks how each node behaves over time, rewarding consistent and successful interactions while quickly penalizing suspicious or failing patterns. The second checks the integrity of packet fragments using a chained hash, allowing incomplete or manipulated sequences to be caught early, before they can occupy memory or waste processing time.

We put this system to the test using a set of targeted attack simulations, including early fragment injection, replayed headers, and flooding with fake data. Across all scenarios, Predictive-CSM preserved network delivery and maintained energy efficiency, even under pressure. Rather than relying on heavyweight cryptography or rigid filters, this approach allows constrained devices to adapt their defenses in real time—based on what they observe, not just what they're told. In that way, it offers a step forward for securing fragmented communication in real-world IoT systems.

Keywords IoT, 6LoWPAN, Low Power Networks, Attacks, Performance Analysis

1 Introduction

These days, low-power wireless networks are doing a lot of heavy lifting in the world of IoT. Whether it's managing irrigation systems in agriculture, monitoring machinery in factories, or helping homes run more efficiently, these networks are everywhere. One technology that's made this possible is 6LoWPAN. It allows tiny devices with minimal memory and power to speak IPv6—essentially giving them a seat at the table on the global internet, even if they're running on coin cell batteries and a few kilobytes of RAM.

But as useful as 6LoWPAN is, it comes with a trade-off. Since these small devices can't send big packets all at once, the data has to be broken into fragments. That sounds fine in theory, but

Somayeh Sobati-Moghadam *, **
* Hakim Sabzevari University, Sabzevar, Iran. Email: s.sobati@hsu.ac.ir
** Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

in practice it creates a weak spot. Devices tend to accept these fragments as they arrive and hold them in memory, even before knowing if the whole message makes sense or comes from someone trustworthy. That small gap—between accepting and verifying—gives attackers just enough room to cause trouble. Since fragments are typically accepted and held in memory before the complete packet is reassembled and validated, attackers can exploit this gap. Even without cryptographic keys or full control of the network, a malicious node can cause serious disruption simply by manipulating how fragments are handled.This opens the door to a range of attacks that target the reassembly buffers of constrained devices. These include not only traditional buffer-reservation attacks but also more nuanced behaviors like clone flooding, header replay, and timing-offset injections—many of which have been observed in recent threat modeling studies [11, 31].

For example, in a simple buffer-reservation attack, an adversary sends incomplete packet fragments (often just the initial FRAG1, which refers to the first fragment of a 6LoWPAN packet that carries essential header information and initiates the reassembly process) slightly before a legitimate transmission. Since devices have very limited buffer space—typically only one or two slots—this alone is enough to block legitimate communication. More advanced strategies involve injecting full but meaningless fragment sequences to exhaust reassembly logic, replaying cloned headers to impersonate trusted nodes, or introducing fragment delays to confuse packet sequencing and induce timeouts. These low-cost, high-impact attacks can silently degrade network performance or cause complete denial of service.

Existing mitigation strategies offer only partial solutions. Stateless 6LoWPAN stacks, such as those implemented in Contiki-NG, have no native fragment validation and accept traffic purely based on header structure and timing. Trust-based systems like Chained Secure Mode (CSM) [30] attempt to mitigate this by filtering packets from untrusted routes, but they operate only at the routing layer and are blind to fragment-level anomalies. On the other end of the spectrum, cryptographic approaches such as SecuPAN [15] enforce MAC-based validation of every fragment, but at a cost that is often too heavy for battery-powered or low-RAM devices.

In this paper, we present *Predictive-CSM*, a practical and resource-aware defense framework for 6LoWPAN fragmentation attacks. The system combines two lightweight mechanisms: an adaptive trust model that continuously learns from a neighbor's fragment-level behavior, and a chained hash validator that verifies the integrity of each fragment incrementally. Unlike traditional models, Predictive-CSM operates directly at the adaptation layer, where fragmentation occurs, enabling the system to identify and discard suspicious fragments before they trigger memory exhaustion or communication breakdowns.

We evaluate Predictive-CSM across five carefully designed attack scenarios: early FRAG1 injection (buffer-reservation), full-fragment flooding, header-replay cloning, burst injection (high-rate FRAG1 spam), and late-phase injection (timing-offset spoofing). These scenarios are chosen to reflect both known attacks and realistic adversarial behavior patterns. Simulation results showed that our approach maintains high packet delivery success, rapid adversary detection, and low energy overhead under all attack conditions. By aligning security enforcement with the actual layer where fragmentation occurs, Predictive-CSM closes a longstanding gap in 6LoWPAN security. It offers a deployable, efficient, and adaptive solution that directly addresses the operational constraints of real-world IoT networks while resisting some of the most effective low-layer threats known to date.

Paper Organization. The remainder of this paper is structured as follows: Section 2 reviews existing approaches to 6LoWPAN fragmentation security and highlights their limitations. Section 3 introduces the architecture of the proposed Predictive-CSM framework, including the Predictive Trust Engine (PTE) and Fragment Signature Validator (FSV). Section 4 presents the detailed protocol design, including message formats, trust evaluation logic, and fragment validation work-

flow. Section 5 describes the simulation environment, attack scenarios, and evaluation metrics used to assess performance. Section 6 provides a comprehensive analysis of the experimental results under multiple adversarial conditions. Section 7 develops an analytical model of trust dynamics, buffer utilization, and cryptographic validation. Finally, Section 8 concludes the paper and outlines directions for future work.

2 Related Work

Recent advances in securing 6LoWPAN networks have led to various approaches addressing fragmentation vulnerabilities. For instance, Sharma et al. [32] proposed an adaptive trust management system that leverages machine learning techniques to identify and mitigate attacks on IoT devices, enhancing the resilience of low-power networks.

While the machine learning-based trust management system adapts to new threats, it may require extensive training data to perform effectively. In dynamic IoT environments, obtaining sufficient labeled data can be challenging. The machine learning models may introduce computational overhead that is not suitable for resource-constrained devices commonly found in 6LoWPAN networks. The reliance on statistical methods may lead to false positives, causing legitimate traffic to be incorrectly classified as malicious.

Li et al. [23] introduced a lightweight cryptographic protocol specifically designed for 6LoWPAN, which utilizes elliptic curve cryptography to provide secure communication while minimizing computational overhead. Their approach emphasizes energy efficiency, crucial for battery-operated devices. Although their cryptographic protocol is lightweight, the management of elliptic curve keys is still be complex in large-scale IoT deployments, particularly when devices are mobile or frequently join/leave the network. While elliptic curve cryptography is efficient, it may still impose latency, especially in high-throughput scenarios where rapid fragment handling is required.

Khan et al. [19] developed an anomaly detection system based on statistical models that monitors packet behavior in real time. This system effectively identifies irregular fragment patterns and mitigates potential denial-of-service attacks by dynamically adjusting trust scores.
The anomaly detection system relies heavily on historical traffic patterns, which may not be effective in environments with rapidly changing behaviors or when new types of attacks emerge. As the number of devices increases, the monitoring and analysis process may become cumbersome, leading to scalability challenges in larger networks.

Another notable contribution comes from Zhang et al. [41], who explored hybrid models combining trust-based and cryptographic mechanisms. Their framework demonstrated significant improvements in packet delivery ratios and energy efficiency under various attack scenarios, providing a comprehensive solution for fragmented communication in IoT networks.
Combining trust-based and cryptographic mechanisms can complicate the implementation and require careful tuning of parameters to avoid conflicts between the two approaches. Even though the proposed framework improves delivery ratios, the dual-layer approach may still consume more resources than purely trust-based or cryptographic-only solutions, which can be problematic for low-power devices.

Mansoor et al. [25] focused on the integration of lightweight blockchain technology to secure data integrity in fragmented packets. Their study highlights how decentralized trust models can enhance security without imposing heavy computational burdens on constrained devices.
The integration of blockchain technology, while innovative, introduces significant overhead in terms of data storage and processing, which may not be feasible for constrained IoT devices. The decentralized nature of blockchain can lead to increased latency in transactions, impacting

real-time communication and responsiveness in fragmented packets.
To tackle the lack of fragment validation, SecuPAN [15] introduces a per-fragment MAC scheme using synchronized nonces and shared keys. While this design addresses replay and spoofing threats, it introduces significant overhead in terms of energy, computation, and memory—characteristics that render it impractical for most Class 1 constrained devices (e.g., Tmote Sky or Zolertia nodes). Moreover, SecuPAN depends on synchronized state management across nodes, which is difficult to maintain in lossy or mobile networks.

Predictive-CSM employs a real-time behavior-based trust model that continuously learns from the interactions of neighboring nodes, allowing the system to adapt quickly to new threats without needing extensive historical data [19]. The proposed framework is designed to operate with minimal computational requirements. By using simple arithmetic operations for trust scoring and lightweight cryptographic hash functions for fragment validation, Predictive-CSM is well-suited for resource-constrained devices. Predictive-CSM considers the historical behavior of nodes over time, allowing it to distinguish between transient anomalies and genuine malicious actions, thus reducing the likelihood of false positives. Our proposed framework utilizes a hash-chain approach that does not require complex key management protocols, simplifying implementation in large-scale IoT networks. Predictive-CSM's trust assessment is decentralized and lightweight, allowing the system to scale efficiently as the number of devices increases without overwhelming the network. The combination of adaptive trust management and lightweight cryptographic validation ensures that Predictive-CSM maintains a low resource footprint while enhancing security. Instead of relying on blockchain technology, our method employs a stateless fragment integrity validation mechanism. This reduces data storage requirements and processing overhead. The light weight framework is designed for quick detection of adversarial behavior, typically within 4-7 seconds, minimizing latency issues and ensuring timely mitigation of threats.

3 Proposed Method

In this section, we detail the architecture and operational workflow of the Predictive-CSM framework, which integrates two lightweight security components: the Predictive Trust Engine (PTE) and the Fragment Signature Validator (FSV). These components are specifically designed to address fragment-level attacks in 6LoWPAN networks without compromising energy efficiency or processing capability. What distinguishes Predictive-CSM is its adaptivity. It does not rely on static keys, rigid packet structures, or heavy cryptographic processing. Instead, it builds trust in neighbors the way a human would—with experience. Each node maintains a running trust score for its immediate neighbors, calculated using past success/failure rates, fragment timing consistency, and payload plausibility. These scores directly influence buffer admission decisions for incoming fragments. Moreover, even if a malicious node maintains high trust for a short time, the second protection layer—the Fragment Signature Validator (FSV)—ensures that fragments can be verified end-to-end through a hash sequence. Together, these two layers create a robust security model for severely constrained devices with limited computational and memory resources.

The core idea behind the Predictive-CSM framework is to combine real-time behavior-based trust assessment with lightweight packet integrity validation in order to secure the 6LoWPAN adaptation layer against fragmentation-based attacks. Traditional solutions, including the Chained Secure Mode (CSM), provide hop-by-hop authentication at the routing layer (RPL), yet leave the data plane—especially fragment handling at the adaptation layer—vulnerable to low-effort attacks. Predictive-CSM addresses this gap by implementing an additional layer of trust intelligence that evolves over time, evaluating the consistency and reliability of each neighbor's

fragment behavior. It also introduces inline validation of packet fragments through hash chaining, allowing malicious fragment sequences to be rejected even if they bypass routing-layer checks. The Predictive-CSM framework is not just a minor enhancement over CSM—it represents a layered shift in how trust and verification are enforced in low-power, fragmented networks. It acknowledges the dual-layer nature of IoT communication—routing trust and data integrity—and addresses both simultaneously. As the next section will showed, this results in not only better security but also in surprisingly improved energy efficiency and lower packet loss under adversarial conditions.

3.1 Component Overview: PTE and FSV

The proposed framework introduces two synergistic components to enhance resilience against fragmentation-based attacks in 6LoWPAN networks: the **Predictive Trust Engine (PTE)** and the **Fragment Signature Validator (FSV)**. The PTE is a lightweight behavior-monitoring module embedded in the adaptation layer that continuously evaluates the trustworthiness of neighboring nodes based on fragment arrival patterns, timing irregularities, and historical delivery success. This dynamic trust score informs buffer allocation decisions and mitigates resource exhaustion caused by malicious FRAG1 flooding or delayed fragment reordering.

On the other hand, FSV serves as the cryptographic layer of defense. It appends chained hash-based tags to fragmented payloads and enables verification of fragment sequences and authenticity before reassembly. Together, PTE and FSV provide both proactive and reactive security—where PTE predicts misbehavior based on behavioral deviations, and FSV cryptographically confirms the fragment chain's integrity.

3.2 Predictive Trust Engine (PTE)

The Trust Scoring Mechanism evaluates each FRAG1 fragment as it arrives. Each source node is associated with a dynamic trust score that is continuously updated based on recent communication behavior. Factors include fragment arrival frequency, sequence order accuracy, and consistency with expected traffic patterns. Nodes with low trust scores may have their fragments dropped or flagged for further inspection.

In order to compute the PTE, the past communication patterns from known devices is stored. Incoming FRAG1 fragments are compared against this historical database to detect anomalies (Algorithm 1). Any deviation, such as sudden traffic bursts or unexpected source IDs, triggers a risk assessment that influences the trust score. A simple time-series based predictor forecasts expected traffic behavior from trusted nodes. It uses minimal memory and computation to maintain energy efficiency. Discrepancies between predicted and actual behavior lower a node's trust score and may initiate protective actions.

To estimate the trustworthiness of neighboring nodes based on their fragment behavior, we define a predictive trust score that evolves over time. The trust score $T_i(t)$ for node i at time t is updated according to the formula:

$$T_i(t) = \lambda \cdot T_i(t-1) + (1-\lambda) \cdot O_i(t)$$

where λ is the forgetting factor (typically between 0.7 and 0.95), $T_i(t-1)$ is the previously computed trust score, and $O_i(t)$ is the latest observed trust event (1 for success, 0 for failure).

Algorithm 1: Evaluating FRAG1 Trust using Predictive Trust Engine (PTE)

Input: $FRAG1$, Node ID n, Historical Pattern H_n, Trust Threshold θ
Output: Trust decision (Accept or Drop)
Initialize $T_n \leftarrow$ current trust score of node n ;
if $n \notin H_n$ **then**
  Add n to H_n with default score $T_n \leftarrow 0.5$;
Extract traffic features from $FRAG1$: frequency, sequence order, timing ;
Compare features to historical pattern in H_n ;
Compute deviation metric δ ;
if δ *is below anomaly threshold* **then**
  Update trust score: $T_n \leftarrow \lambda \cdot T_n + (1 - \lambda) \cdot 1$;
else
  Update trust score: $T_n \leftarrow \lambda \cdot T_n + (1 - \lambda) \cdot 0$;
if $T_n < \theta$ **then**
  return Drop Fragment ;
else
  return Accept Fragment ;

This formulation allows recent behaviors to have more weight while still preserving long-term historical information [6], [33]. Each node also maintains a threshold trust level θ. If $T_n(t) < \theta$, then all future fragments from node n are dropped unless its behavior improves. This creates a sliding window of opportunity for attackers and misbehaving nodes—persistent deviation from normal behavior causes their fragments to be ignored.

3.3 Fragment Signature Validator (FSV)

Each legitimate sender generates a lightweight signature for every fragment using a hashing function over the fragment payload and a shared secret key (Algorithm 2). The signature is appended to the fragment in an extended header field. When fragments are received, the receiver recomputes the hash using the same key and compares it against the received signature. Only fragments that pass this validation are allowed into the reassembly buffer. This process ensures that even if FRAG1 is trusted, malicious fragments can't corrupt the full packet. Fragments that fail validation are discarded immediately. If multiple invalid fragments are detected from the same source within a short time frame, the system flags the source node, updates its trust score, and may block further traffic from it temporarily. The FSV mechanism uses hash chaining for fragment integrity. Each FRAG1 includes a seed hash H_0, and every subsequent fragment f_i carries a chained hash $H_i = H(H_{i-1} \| data_i)$. Upon reassembly, the destination node validates that the final computed hash matches the expected hash stored in the FRAGN fragment. This lightweight method requires negligible CPU overhead on typical IoT hardware. The synergy between PTE and FSV is key. The PTE proactively guards the fragment admission process based on learned trust, while FSV acts as a cryptographic backstop to detect subtle forgery and sequencing anomalies. Together, they offer robust protection against attacks such as buffer-reservation, spoofed fragment flooding, and replayed FRAG1 headers—attacks that are particularly effective against traditional 6LoWPAN setups. Importantly, this scheme does not require any changes to the core 6LoWPAN standards. Instead, it hooks into the decision points of buffer admission and fragment processing, making it easily portable to OSs like Contiki-NG, RIOT, or TinyOS. It also avoids energy-expensive cryptographic primitives like public key encryption, instead using cumulative trust and hash functions to keep computational and memory load minimal.

Algorithm 2: Validating Fragment Signature using FSV

Input: Fragment f_i, Shared key K, Previous hash H_{i-1}
Output: Validation result (Valid or Invalid)

```
Extract payload data d_i from f_i ;
Compute expected signature: H'_i ← HMAC(K, H_{i-1}‖d_i) ;
Retrieve received signature H_i from fragment header ;
if H'_i = H_i then
    Store H_i as H_{i-1} for next fragment ;
    return Valid ;
else
    return Invalid ;
```

3.4 Integration into 6LoWPAN Stack

To support Predictive-CSM, minor extensions are made to the 6LoWPAN fragmentation header. These include additional fields for trust metadata and cryptographic signatures. The protocol remains backward-compatible for nodes that do not support Predictive-CSM. The modified stack first passes incoming fragments through the PTE for trust evaluation. If the fragment passes the PTE threshold, it is then passed to the FSV for signature validation (Algorithm 3). Only fragments that clear both checks are stored in the reassembly buffer. This sequential filtering provides a robust mechanism to defend against fragment-level attacks without adding significant computational overhead.

Algorithm 3: Fragment Processing in Modified 6LoWPAN Stack

Input: Incoming Fragment f_i, Node ID n, Previous Hash H_{i-1}, Trust Threshold θ
Output: Reassembly Buffer Update or Fragment Drop

```
if f_i is FRAG1 then
    Retrieve current trust score T_n for node n ;
    if T_n < θ then
        return Drop fragment ;
    Store trust score for session and initialize reassembly ;
    return Proceed to signature validation ;
Compute expected signature H'_i ← HMAC(K, H_{i-1}‖d_i) ;
Retrieve received signature H_i from fragment header ;
if H'_i = H_i then
    Update hash chain: H_{i-1} ← H_i ;
    Store f_i in reassembly buffer ;
else
    Penalize trust score of node n ;
    if Trust score falls below θ then
        Temporarily block node n ;
    return Drop fragment ;
```

4 Predictive-CSM Protocol Design

To implement Predictive-CSM, the standard 6LoWPAN fragment header is extended with two fields, the **Trust Metadata,** encodes the sender's self-assessed trust score and fragment behavior

flags and the **Fragment Signature,** a truncated hash value computed using a keyed hashing algorithm like HMAC-SHA1. These fields are appended to both FRAG1 and subsequent fragments. The Predictive-CSM protocol is designed to be backward-compatible. Nodes that do not support the trust engine or signature fields will ignore the extended headers and proceed using default 6LoWPAN behavior. Predictive-CSM can be deployed incrementally in heterogeneous IoT environments.

4.1 Sender-Side Operations

The sender's job in this process is to prepare each data fragment so that it carries both the information and a clear sign of its integrity and trustworthiness. Before sending anything, the sender checks how trustworthy the destination node is using a trust evaluation function. This trust score plays an important role in the metadata that gets added to each fragment (Algorithm 4). For the first fragment, a cryptographic hash is created using the packet's payload and a unique nonce. For any fragments that follow, the sender builds a chain by hashing the previous hash along with the current payload, effectively linking them all together.

Once the sender has built the hash correctly, it adds that along with the current trust score to the fragment's header. This extra bit of information is like a proof of identity—it tells the receiver not only who sent the data, but also that the content hasn't been messed with along the way. After that, the sender simply sends off the fragment, wrapping up its side of the process.

4.2 Receiver-Side Operations

When the receiver gets a fragment, first, he/she figures out who the sender is. If the fragment happens to be the start of a new message, the receiver double-checks how much it trusts the sender using a kind of prediction system. If the sender's trust score isn't high enough, the receiver just drops the fragment right away to play it safe (Algorithm 5). But if the sender seems trustworthy, the receiver digs a bit deeper. It re-creates the hash using the shared key and the hash from the last piece, then compares that to what came with the new fragment. If the two don't match, something's probably wrong—maybe the fragment was altered—so the receiver drops it and marks the sender down a notch in the trust system. On the other hand, if everything looks fine, the fragment gets saved. The sender earns a small trust reward, and the chain of hashes continues. If the receiver sees this is the final fragment in the sequence, it puts everything back together into the original packet. That way, only fragments that are both valid and sent by trustworthy sources are accepted.

4.3 Trust Evaluation and Update Rules

The system keeps track of how much each node can be trusted by updating a trust score over time. This score isn't fixed—it changes depending on the behavior of the node. If a node behaves well, like sending fragments in the correct order and at expected intervals, its trust score goes up. But when something suspicious happens—like fragments arriving out of order, too quickly, or appearing tampered with—the trust score drops. The score is always kept within a range from 0 to 1, where 0 means the node is completely untrusted, and 1 means it's fully trusted. If a node's score falls too low, say below 0.3, the system temporarily blacklists it to prevent potential misuse.

Algorithm 4: Sender Operations

Input: Packet *packet*, Destination node *dest_node*
Output: Transmit fragment with trust metadata and signature

```
trust_score ← get_trust_score(dest_node);
if is_first_fragment(packet) then
    H_prev ← HMAC(K, packet.payload ‖ nonce) ;          // Seed hash for FRAG1
else
    H_prev ← get_previous_hash(packet) ;                // Chain from prior fragment
H_i ← HMAC(K, H_prev ‖ packet.payload) ;                // Compute chained hash
;
attach_header(packet, trust_score, H_i) ;               // Add trust + signature
;
transmit(packet);
```

Algorithm 5: Receiver Operations

Input: Incoming fragment $fragment$
Output: Reassembled packet or drop decision

```
sender ← fragment.source;
if is_first_fragment(fragment) then
    if PTE.evaluate(sender) < θ then
        drop(fragment);
        return;
H_received ← fragment.header.signature;
H_expected ← HMAC(K, H_prev ‖ fragment.payload);
if H_received ≠ H_expected then
    PTE.penalize(sender) ;                              // Update trust score
    drop(fragment);
else
    store(fragment) ;                                   // Valid fragment
    PTE.reward(sender);
    H_prev ← H_received ;                               // Update hash chain
    if is_last_fragment(fragment) then
        reassemble_packet();
```

4.4 Security Response to Detected Attacks

When the system notices that a sender is repeatedly transmitting bad or suspicious fragments, it takes action right away. First, the sender's trust score is reduced to reflect the misbehavior. At the same time, alerts are raised so that higher-level components in the network can respond appropriately. If the issue continues, the system may begin to limit how often that sender can transmit data—or block it entirely. These responses are designed to happen quickly and automatically, allowing the network to stay protected and resilient without putting too much strain on system resources.

5 Experimental Setup

5.1 Simulation Environment

To evaluate the performance and robustness of the proposed Predictive-CSM framework, we implemented a series of simulations using the Contiki-NG operating system and its Cooja simulator.

Contiki-NG is a widely used operating system for networked embedded systems in the Internet of Things (IoT) domain and offers native support for IPv6, 6LoWPAN, RPL, and lightweight security protocols [36]. Cooja provides a highly configurable environment for simulating wireless sensor networks (WSNs) at both the network and hardware levels, making it suitable for testing both protocol correctness and system performance under adversarial conditions [28].

The simulated network consists of 10 wireless nodes arranged in a star topology. One node functions as the RPL root, another as the adversarial entity, and the remaining nodes operate as legitimate data senders. This configuration allows us to evaluate communication flow in the presence of a centrally positioned attacker. All nodes were configured as Sky motes, which emulate the Tmote Sky platform featuring a TI MSP430 microcontroller, 10 kB of RAM, and IEEE 802.15.4-compliant radio transceivers. These hardware constraints are representative of real-world IoT deployments where computational and memory resources are significantly limited [29].

Each legitimate node transmits one data packet every 90 seconds using UDP over IPv6. The packets are intentionally sized to exceed the IEEE 802.15.4 frame limit, resulting in fragmentation at the 6LoWPAN layer. The fragment size was configured to 96 bytes for payload plus 8 bytes for the 6LoWPAN fragmentation header, consistent with practical deployments [37]. The 6LoWPAN stack uses the default Route-Over forwarding strategy implemented in Contiki-NG, where each intermediate node reassembles and re-fragments packets before forwarding them.

The simulation duration for each scenario was set to 30 minutes, and results were averaged over 15 independent runs to ensure statistical reliability. Each simulation was initialized with a network convergence period of 50 seconds, allowing routing paths to stabilize before any adversarial behavior began. The attacker node mimicked realistic IoT behavior for the initial phase, then launched attacks such as buffer-reservation, full-fragment injection, and header-replay attacks in separate test scenarios.

Power consumption was measured using Contiki-NG's Energest module, which records energy usage across CPU active time, radio transmission, and radio listening modes. This metric was critical for evaluating the resource efficiency of the proposed solution under both benign and adversarial conditions. Packet delivery ratios and fragment-level drop rates were also tracked at the root node using Contiki-NG's packet sniffer and logging utilities.

To ensure relevance and reproducibility, the simulation parameters and methodology align with recent academic studies evaluating IoT security frameworks [8, 35]. The combination of real-time attack scenarios, constrained node emulation, and multi-layered protocol analysis provides a robust testbed for validating both the security guarantees and operational efficiency of Predictive-CSM in adversarial IoT environments.

5.2 Evaluation Metrics

This section outlines the key metrics used to evaluate the system's performance, including Packet Delivery Ratio (PDR), Fragment Drop Rate, Power Consumption, and Detection Latency. These metrics assess reliability, efficiency, energy usage, and responsiveness in identifying and mitigating attacks.

Packet Delivery Ratio (PDR) is the proportion of successfully delivered and reassembled packets.

Fragment Drop Rate is the number of discarded fragments per hundred received, due to trust or hash mismatch.

Power Consumption is the average energy usage per node in milliwatts.

Detection Latency is the time from attack initiation to adversary identification and blocking.

5.3 Adversarial Model

In order to rigorously evaluate the resilience of the proposed Predictive-CSM framework, we simulate a range of realistic attack strategies targeting the 6LoWPAN adaptation layer. The adversarial model described here is informed by well-documented fragmentation vulnerabilities and denial-of-service strategies outlined in recent literature on IoT security [7, 11, 31, 35]. These attacks exploit weaknesses in fragment verification, buffer allocation, and trust assumptions—threat surfaces that remain largely unresolved in default protocol stacks like those in Contiki-NG and RIOT. We assume the attacker is an external node without access to valid cryptographic keys. It behaves passively during the RPL initialization phase to establish perceived legitimacy, and subsequently engages in active disruption once the network reaches routing stability. The following attack scenarios are each designed to expose specific vulnerabilities in the 6LoWPAN fragment reassembly pipeline, and directly correspond to the results presented in Section VII.

1. Early FRAG1 Injection (Buffer-Reservation)

This well-known attack targets buffer exhaustion by sending FRAG1 fragments milliseconds before legitimate transmissions [17]. Since reassembly is initiated upon receipt of FRAG1, the receiver allocates scarce memory to unauthenticated fragments, blocking future reassembly of genuine packets. Prior studies confirm this is among the most effective low-resource denial-of-service strategies in constrained wireless sensor networks [1].

2. Complete Fragment Flooding

In this variant, the adversary injects complete sequences of syntactically correct but semantically invalid fragments. These cause full reassembly attempts, wasting CPU cycles and radio resources [14]. The goal is to maximize energy drain and buffer turnover without raising alarms based on simple traffic volume heuristics.

3. Header-Replay Cloning

This attack uses previously captured FRAG1 headers from trusted nodes and replays them at later intervals, exploiting the absence of per-fragment origin authentication. Such replay-based impersonation attacks are increasingly relevant in IoT systems where trust is static or context-unaware [13, 31].

4. Burst Injection (High-Rate FRAG1 Flooding)

Here, the attacker sends multiple FRAG1s per second (up to 6), with the goal of rapidly overwhelming the limited reassembly buffers. Burst injection represents a brute-force version of buffer-reservation, testing whether a system can reject high-volume malicious traffic in real time without disrupting legitimate flows [4].

5. Late-Phase Injection

This timing-sensitive scenario involves inserting malicious fragments slightly after legitimate FRAG1 transmissions, with the intent to disrupt fragment sequencing or trigger premature timeouts. This technique is increasingly relevant as attackers leverage traffic analysis and jitter modeling to bypass fixed trust rules [7].

Table 1 Mapping Between Modeled Attacks and Evaluation Scenarios

Scenario in Results	Modeled Adversarial Behavior
Early FRAG1 Injection	Preemptive buffer-reservation attack
Complete Fragment Flooding	Full packet flooding with malformed content
Header-Replay Cloning	Reused legitimate fragment headers to spoof trust
Burst Injection (6/sec)	High-rate FRAG1 spamming to saturate buffers
Late-Phase Injection	Timing-offset fragment spoofing post-legitimate traffic

Scenario-to-Result Mapping

Table 1 explicitly links each adversarial behavior with its evaluation label in the results section, ensuring clarity and reproducibility.

By simulating these five targeted and diverse adversarial behaviors, we ensure that Predictive-CSM is tested against both brute-force and context-aware attacks. This approach reflects the evolving nature of IoT threats and aligns with best practices in security testing as outlined in recent surveys and threat modeling frameworks [11, 35].

6 Results

This section presents and analyzes the experimental findings derived from our simulation scenarios, designed to rigorously test the performance of the proposed Predictive-CSM framework against both conventional and advanced 6LoWPAN attacks. We compare its behavior to two baseline protocols: unmodified (vanilla) 6LoWPAN and CSM-integrated 6LoWPAN. The metrics we focus on include Packet Delivery Ratio (PDR), average node power consumption, fragment drop rate, and adversarial detection latency.

6.1 Node Power Consumption

Energy efficiency is critical for battery-operated IoT devices. We measured average power consumption across protocols under varying attack conditions, using Contiki-NG's Energest module. Results are normalized to baseline (no attack) operation.

Table 2 Average Power Consumption (mW) Under Attack Scenarios

Scenario	Vanilla	CSM	SecuPAN	Predictive-CSM	Delta vs SecuPAN
No Attack	0.29	0.32	0.41	0.34	-17.1%
Early FRAG1 Inj.	0.35	0.36	0.52	0.33	-36.5%
Complete Flooding	0.39	0.40	0.58	0.34	-41.4%
Burst Injection	0.43	0.42	0.61	0.35	-42.6%

SecuPAN's cryptographic overhead exhibits 26–42% higher power consumption than Predictive-CSM due to per-fragment MAC computations. This aligns with energy analyses showing that AES-128 MAC operations increase MSP430 CPU active time by 31%. Predictive-CSM's efficiency maintains near-baseline consumption (0.33–0.35 mW) even under attack through early fragment rejection via trust scores, saving 18–22% radio RX energy, and lightweight HMAC-SHA1 hashing (0.01 mW per fragment vs. SecuPAN's 0.08 mW). Vanilla 6LoWPAN paradox shows higher

attack-phase consumption (0.43 mW) than Predictive-CSM despite no security checks, due to buffer overflow-induced retransmissions. Predictive-CSM reduces energy waste by 41.4% versus SecuPAN in flooding attacks while maintaining security, addressing the energy-security trade-off identified in previous studies.

6.2 Packet Delivery Ratio (PDR)

PDR measures network reliability under attack. We evaluate successful reassembly of legitimate packets at the root node.

Table 3 Packet Delivery Ratio (%) Across Protocols

Scenario	Vanilla	CSM	SecuPAN	Predictive-CSM	Gain vs CSM
No Attack	97.4	98.9	99.1	99.2	+0.3%
Early FRAG1 Inj.	54.2	85.3	94.7	99.0	+13.7%
Header Replay	41.6	82.4	96.3	98.9	+16.5%
Burst Injection	39.7	77.2	89.5	97.4	+20.2%

SecuPAN's cryptographic assurance achieves 94.7–96.3% packet delivery ratio (PDR) in attacks through mandatory fragment authentication, but struggles with high-rate bursts (89.5%) due to verification delays. Predictive-CSM's dual-layer advantage matches SecuPAN's PDR in replay attacks (98.9% vs. 96.3%) and excels in burst scenarios (97.4% vs. 89.5%) via adaptive trust thresholds that prevent buffer saturation. CSM's routing-layer limitation shows 13.7–20.2% lower PDR than Predictive-CSM, confirming that routing-layer trust alone cannot prevent fragment-level attacks. Our results validate the hybrid trust-cryptography model, demonstrating that lightweight hashing (approximately 8 bytes per fragment) combined with behavioral analysis can achieve 97–99% PDR without SecuPAN's energy costs.

6.3 Fragment Drop Rate

The fragment drop rate quantifies the system's ability to discriminate malicious fragments while preserving legitimate traffic. We evaluate this metric as the ratio of dropped fragments per 100 received, comparing Predictive-CSM against CSM-6LoWPAN, SecuPAN, and vanilla 6LoWPAN under identical attack conditions.

Table 4 Fragment Drop Rate Across Protocols (per 100 fragments)

Scenario	Vanilla	CSM-6LoWPAN	SecuPAN	Predictive-CSM
Normal Conditions	0.2	0.1	**0.3**	0.1
Early FRAG1 Inj.	8.3	2.4	1.8	**0.6**
Header Replay	12.5	3.7	2.1	**0.8**
Burst Injection	16.9	4.6	3.5	**1.1**

SecuPAN's cryptographic rigor exhibits marginally higher drop rates (1.8–3.5) than Predictive-CSM in attack scenarios due to its strict MAC-based validation, which discards fragments with even minor integrity violations. While effective against spoofing, this approach proves overly aggressive in lossy environments where bit errors may corrupt legitimate fragments.

Predictive-CSM's adaptive advantage achieves superior drop rates (0.6–1.1) by combining lightweight hash validation with behavioral trust. The trust engine reduces false positives by tolerating transient errors from historically reliable nodes, aligning with findings in previous studies. This dual-layer approach addresses a key limitation of pure cryptographic methods: their inability to distinguish between malicious intent and channel-induced errors. Comparative performance shows that vanilla 6LoWPAN suffers catastrophic drop rates (8.3–16.9) due to buffer exhaustion. CSM-6LoWPAN improves upon vanilla but remains vulnerable to fragment-level attacks (2.4–4.6). Predictive-CSM reduces drops by four times versus CSM and fifteen times versus vanilla in burst scenarios, validating its efficacy as a DoS mitigation tool.

6.4 Detection Latency

Detection latency measures the time elapsed from attack initiation until Predictive-CSM consistently blocks malicious fragments. This metric is critical for real-time IoT systems where delayed responses can lead to resource exhaustion or service disruption.

Table 5 Detection Latency Across Attack Scenarios

Attack Type	**Predictive-CSM (s)**	**CSM-6LoWPAN (s)**	**SecuPAN (s)**	**Vanilla 6LoWPAN**
Early FRAG1 Injection	5.1	8.3	**4.9**	No detection
Header-Replay Cloning	6.8	12.5	**5.2**	No detection
Burst Injection (6/sec)	**4.4**	16.9	7.1	No detection
Late-Phase Injection	7.0	14.2	**6.8**	No detection

Predictive-CSM outperforms CSM-6LoWPAN in all scenarios, reducing latency by 48–74% due to its per-fragment behavioral analysis. SecuPAN achieves marginally faster detection (e.g., 4.9 seconds vs. 5.1 seconds for FRAG1 injection) through cryptographic validation, but at higher energy costs. The worst-case latency (7.0 seconds) for Predictive-CSM occurs in late-phase injection attacks, where subtle timing anomalies require longer observation windows.

While SecuPAN offers lower latency for some attacks, Predictive-CSM provides a balanced approach by combining near-real-time detection (less than 7.0 seconds) with minimal resource overhead. This makes it suitable for deployments where energy efficiency and computational constraints are prioritized over nanosecond-level response times.

6.5 Parameter Sensitivity Analysis

To evaluate the robustness of Predictive-CSM's trust model, we conducted a systematic analysis of its key parameters: the *forgetting factor* (λ) and *trust threshold* (θ). The goal was to quantify their impact on security performance and operational efficiency.

6.5.1 Forgetting Factor (λ)

The forgetting factor controls how rapidly the trust model adapts to recent behavior. We tested four values:

$$\lambda \in \{0.7, 0.8, 0.9, 0.95\} \tag{1}$$

- **Lower values** ($\lambda = 0.7$) prioritized recent events, reducing attack detection latency to **3.2 seconds** for burst injection but increasing false positives (**12%**) during transient interference.
- **Higher values** ($\lambda = 0.95$) improved stability, with false positives below **3%** in benign conditions but delayed attack response by **1.5–2 seconds**.
- The default $\lambda = 0.9$ balanced these trade-offs, maintaining detection latency below **7 seconds** while limiting false drops to **¡5%**.

6.5.2 Trust Threshold (θ)

The trust threshold determines when a node is blacklisted. We evaluated three configurations:

$$\theta \in \{0.2, 0.3, 0.4\} \tag{2}$$

- **Lower thresholds** ($\theta = 0.2$) reduced legitimate fragment drops by **4%** but allowed attackers **1–2 additional malicious fragments** before mitigation.
- **Stricter thresholds** ($\theta = 0.4$) improved PDR by **2%** under sustained attacks but increased false blocking during intermittent packet loss.
- The chosen $\theta = 0.3$ optimized both security and tolerance, with **98.9% PDR** and **5.1-second median detection latency**.

Table 6 Impact of Parameter Variations on Performance

Configuration	λ	θ	Detection Latency (s)	False Positives (%)	PDR Under Attack (%)
Aggressive	0.7	0.2	3.2	12.1	96.8
Default	0.9	0.3	5.1	4.7	98.9
Conservative	0.95	0.4	7.8	2.9	97.3

Key Insight: As shown in Table 6, the default configuration ($\lambda = 0.9$, $\theta = 0.3$) achieved optimal balance across all metrics, validating our design choices for real-world IoT deployments where transient network issues and persistent attacks coexist.

6.6 Summary of Insights

These experimental outcomes demonstrate that the Predictive-CSM approach offers a compelling balance of precision, performance, and energy efficiency. By combining long-term behavioral learning with inline fragment verification, it effectively addresses both structural and behavioral attack vectors. Notably, its response is both proactive (in lowering trust values) and reactive (in fragment hash validation), unlike prior systems which often depend solely on predefined thresholds or rate-limiting policies [11].

This dual-mode strategy is particularly crucial for environments where computational resources are sparse and false positives can cripple application functionality. It confirms emerging academic consensus that multi-layered, adaptive trust and lightweight cryptography are key pillars of next-generation IoT security architectures [1].

7 Analytical Model of Predictive-CSM Framework

This section presents a formal analysis of the Predictive-CSM framework, covering the evolution of trust scores, cryptographic fragment verification, and resource usage under constrained conditions. These models complement our simulation results and offer deeper insight into system behavior under attack and in regular operation. All notations are used in this section is shown in Table 7.

Table 7 Summary of Analytical Model Notations

Symbol	Description
$T_n(t)$	Trust score of node n at time t
$T_n(t-1)$	Previous trust score of node n
λ	Forgetting factor (trust memory decay), $0 < \lambda < 1$
$O_n(t)$	Outcome of current interaction (1 = valid, 0 = invalid)
θ	Trust threshold for fragment acceptance
K	Shared secret key for HMAC computation
d_i	Payload of the i^{th} fragment
H_i	Hash value for fragment i
H_{i-1}	Hash value from the previous fragment
nonce	Random or time-based seed for initial hash H_0
B	Number of available reassembly buffer slots
λ (buffer)	Arrival rate of valid fragments
A	Arrival rate of malicious/invalid fragments
τ	Reassembly timeout window
ρ	Buffer occupancy ratio
P_{buffer}	Probability that a reassembly buffer is available

1. Trust Dynamics Model

In dynamic and decentralized IoT environments, where devices frequently interact without centralized control, maintaining trust is crucial to ensure reliable communication. Unlike traditional networks that often depend on static credentials or centralized authorities, low-power wireless systems must rely on localized, real-time decisions informed by each node's observable behavior. This has led to the adoption of lightweight, behavior-based trust models, which allow individual nodes to assess their immediate neighbors over time [26, 40].

The Predictive-CSM framework incorporates such a model through its Predictive Trust Engine (PTE), which continuously monitors and updates the trustworthiness of each neighbor. The core idea is simple but effective: nodes that consistently send well-formed, timely, and valid packet fragments see their trust scores increase, while those that cause errors—such as fragment mismatches, malformed content, or suspicious timing—experience a decline in trust. This mimics real-world trust dynamics: gradually earned, but easily lost.

Formally, the trust score of a neighbor node n at time t, denoted $T_n(t)$, is updated using an exponential moving average:

$$T_n(t) = \lambda \cdot T_n(t-1) + (1-\lambda) \cdot O_n(t) \tag{3}$$

Here:

- $\lambda \in (0, 1)$ is the forgetting factor, controlling how much recent behavior influences the score,
- $T_n(t-1)$ is the previously computed trust score,

– $O_n(t)$ represents the outcome of the current interaction: 1 for success (valid fragment), 0 for failure (e.g., invalid signature or malformed sequence).

The trust score is bounded in the range $[0, 1]$. When a node's score falls below a threshold θ, it is considered untrustworthy, and its fragments are dropped without further processing. This allows the system to dynamically adjust to both rapid attacks and slow-degrading behavior, making it resilient to diverse threat patterns [9, 10].

To illustrate how trust declines under repeated malicious behavior, consider a scenario in which $O_n(t) = 0$ for several consecutive time windows. Assuming an initial trust score of $T_n(0) = 0.8$, a threshold of $\theta = 0.3$, and a forgetting factor $\lambda = 0.9$, the node would be blacklisted after just 3–4 invalid fragment events. This level of responsiveness is especially important in real-time systems where buffer exhaustion or flooding attacks can escalate within seconds.

One of the strengths of this model lies in its adaptability. It does not rely on fixed rules about what constitutes "malicious" activity. Instead, it infers patterns from observed behavior over time. Moreover, the trust calculation involves only basic arithmetic operations, making it computationally lightweight and ideal for deployment on resource-constrained microcontrollers commonly used in IoT applications [19].

It is important to note that the trust model does not operate in isolation. It serves as the first layer of defense in the Predictive-CSM architecture. When used alongside fragment-level cryptographic checks provided by the Fragment Signature Validator (FSV), the trust score becomes a powerful tool for early attacker detection and fragment filtering before significant damage occurs.

What sets this approach apart from traditional binary or rule-based systems is its ability to reflect behavioral nuance. Rather than making rigid decisions based on single events, it tracks consistency over time. This means that short-lived disruptions—like packet jitter, signal interference, or temporary congestion—will not result in immediate penalties. Instead, the trust score degrades gradually, providing room for recovery and avoiding false positives. In contrast, persistent suspicious patterns quickly trigger trust erosion and isolation of the misbehaving node.

This continuous trust evaluation aligns with recent research advocating for adaptive security mechanisms in IoT networks. Furthermore, because the trust score is updated based on direct fragment-level observations within the 6LoWPAN adaptation layer, it offers a highly accurate and timely reflection of node behavior. There is no need for centralized monitoring or computationally expensive anomaly detection. The result is a robust and scalable trust system that significantly enhances security without imposing unnecessary burdens on constrained devices.

2. Fragment Integrity Verification

While the Predictive Trust Engine (PTE) provides a behavior-based mechanism to assess node reliability over time, it cannot on its own guarantee the integrity or authenticity of individual fragments. To address this limitation, the Predictive-CSM framework incorporates a second line of defense: the Fragment Signature Validator (FSV). This component provides per-fragment cryptographic validation that ensures both the authenticity and sequence integrity of fragments, even when sent by seemingly trustworthy nodes.

The core mechanism used by the FSV is chained hashing, a lightweight cryptographic approach suitable for low-power and memory-constrained devices. Chained hash schemes have proven effective in securing data streams in IoT and 6LoWPAN networks by enabling incremental, verifiable linkage between sequential packets or fragments [38].

In Predictive-CSM, the sender constructs a hash chain by first generating a seed hash for the initial fragment:

$$H_0 = \text{HMAC}(K, d_0 \parallel \text{nonce}) \tag{4}$$

Here, K is a shared secret key, d_0 is the payload of the first fragment, and the nonce provides randomness to prevent replay attacks. For each subsequent fragment f_i, a chained hash is computed:

$$H_i = \text{HMAC}(K, H_{i-1} \parallel d_i) \tag{5}$$

Each fragment thus carries a hash value that depends not only on its own content, but also on the hash of the previous fragment, ensuring that any tampering or reordering will be immediately detectable.

On the receiver side, this hash is recomputed and compared to the one embedded in the fragment header. If a mismatch is found, the fragment is dropped and the sender is penalized via the trust engine. This early rejection mechanism is more efficient than full-packet validation schemes, which require the receiver to hold and assemble all fragments before verification. Studies confirm that incremental hash-based authentication is highly compatible with 6LoWPAN's fragment handling mechanisms and can reduce energy and memory usage significantly [21].

The FSV offers security benefits at a low cost. Computational requirements are also light, involving only a single HMAC calculation per fragment. This is important in IoT networks composed of Class 1 devices with limited flash and RAM capacities.

Combined with the behavior-based scoring provided by the PTE, the FSV acts as a fail-safe: even if a node has not yet been marked as untrustworthy, it cannot inject malformed fragments without detection. This aligns with research advocating for layered security in IoT, where lightweight cryptographic checks work in tandem with anomaly-based detection [39].

The superiority of this approach lies in its fine-grained, stateless validation capability. Unlike full-message authentication schemes that rely on MACs or digital signatures and require reassembly of the entire payload before validation, our method allows for incremental and forward-compatible verification during the reassembly process. This ensures that malicious fragments are caught as early as possible, reducing wasted buffer space and processing cycles. Compared to schemes like SecuPAN [15], which require a MAC for each fragment and involve shared key material and replay counters, our solution is lighter, faster, and easier to implement on constrained devices with less than 10 kB of RAM.

Moreover, this technique adds virtually no observable overhead in practical scenarios. As demonstrated in our experimental results, the energy cost of computing chained hashes per fragment was negligible—amounting to under 0.01 mW per node on average—even during high-volume attack scenarios. At the same time, its security gains were substantial: fragment drop rates decreased by over 90% and packet delivery reliability improved to over 98% even in adversarial conditions.

Unlike static firewalls or basic trust filters, this mechanism provides a cryptographic backstop for each fragment, ensuring that even a temporarily trusted node cannot slip through malformed or malicious fragments. This dual defense—combining behavioral reputation with per-fragment integrity checks—resonates with recent literature advocating multi-layered defenses for 6LoWPAN [13, 31].

The fragment integrity validator in Predictive-CSM represents a critical layer of defense that bridges the gap between behavioral security and data authenticity. It is efficient, scalable, and importantly, tailored to the operational realities of resource-constrained IoT devices. This makes it not only a complementary tool to trust scoring but a necessary one for achieving end-to-end packet integrity in hostile wireless environments. By validating fragments incrementally and independently, it allows the Predictive-CSM framework to maintain strong data integrity guarantees without sacrificing responsiveness or exhausting system resources.

3. Buffer Availability Estimation

In constrained 6LoWPAN networks, memory exhaustion is a serious threat due to the limited buffer capacity of IoT nodes. Fragment flooding attacks, malformed packets, or simply high background traffic can cause the reassembly buffer to overflow, leading to packet loss, service degradation, or denial of service. The Predictive-CSM framework mitigates this threat through a proactive trust-based filtering mechanism that helps ensure buffer availability.

We model the expected buffer utilization to evaluate how the system behaves under both normal and adversarial conditions. Let:

- B be the number of available reassembly buffer slots,
- λ be the arrival rate of valid (legitimate) fragments,
- A be the arrival rate of malicious or invalid fragments,
- τ be the timeout duration for fragment reassembly.

The buffer occupancy ratio ρ is given by:

$$\rho = \min\left(1, \frac{\lambda + A}{B \cdot \tau}\right) \tag{6}$$

From this, the probability that a buffer is available at any time is:

$$P_{\text{buffer}} = 1 - \rho \tag{7}$$

Without any form of pre-filtering, the malicious traffic A can quickly dominate the total load, especially in denial-of-service scenarios. In such cases, $\rho \to 1$ and $P_{\text{buffer}} \to 0$, meaning legitimate packets are increasingly dropped due to lack of available memory. This dynamic is a well-documented vulnerability in IoT routing and adaptation layers. The Predictive-CSM framework reduces this impact by dynamically lowering the trust scores of nodes that send malformed or suspicious fragments. Once a node's trust score falls below the threshold θ, its fragments are dropped early—before entering the reassembly buffer. As a result, the effective arrival rate of adversarial fragments A' decreases over time, pushing ρ downward and keeping P_{buffer} high.

Unlike reactive approaches that flush buffers after misuse is detected, Predictive-CSM acts proactively, preserving resources by preventing untrusted data from occupying memory. Prior studies have shown that early filtering based on trust or behavior patterns can extend device uptime and improve end-to-end delivery rates in similar constrained environments [2, 18].

By modeling this effect analytically, we confirm that Predictive-CSM not only improves security but also contributes to system stability and resource conservation—two key challenges in the deployment of real-world IoT networks.

4. Comparison with Other Solutions

To contextualize the effectiveness of Predictive-CSM, it is essential to compare it against prominent existing solutions designed to secure 6LoWPAN from fragmentation-related attacks. These include vanilla 6LoWPAN (with no fragment-level security), CSM-6LoWPAN (relying solely on routing-layer trust), and more heavyweight protocols like SecuPAN, which apply cryptographic protections to every fragment.

Vanilla 6LoWPAN offers no security for fragment origin, structure, or completeness. Fragments are accepted purely on structural criteria, making the system extremely vulnerable to buffer-reservation and replay attacks [17]. As our simulations demonstrated, this baseline system suffers over 40% packet loss under moderate attack pressure and performs poorly in adversary

detection ($P_{\text{bypass}} \approx 1$). Its main advantage is low overhead, but at the cost of being effectively defenseless.

CSM-6LoWPAN, while an improvement, limits its protections to the routing layer using hop-by-hop trust chains. It can filter fragments from previously untrusted nodes but cannot validate individual fragments in a packet chain. This limitation makes it susceptible to impersonation and replay attacks, especially when adversaries spoof FRAG1 headers from recently trusted nodes. As shown in our results, CSM mitigates basic DoS attempts but cannot achieve delivery reliability above 88% in more advanced attack scenarios. Its reaction time is also slower, often requiring multiple failed interactions to trigger blacklist behavior.

SecuPAN represents a cryptographically strong approach that signs each fragment with a MAC and uses shared keys and nonces to prevent forgery [15]. While effective in theory, it introduces significant complexity: fragment processing must include cryptographic verification; nonce synchronization becomes fragile in high-loss networks; and memory usage increases due to the per-fragment state. For resource-constrained devices with limited RAM and processing power, these drawbacks are non-trivial. Previous evaluations show a 25–30% increase in energy usage under typical IoT conditions [13].

In contrast, **Predictive-CSM offers a hybrid solution** that combines adaptive trust modeling with lightweight fragment integrity verification—achieving strong security guarantees with minimal overhead. It detects adversarial behavior within 4–7 seconds, maintains delivery ratios above 98%, and consumes less energy under attack than both CSM and SecuPAN, as evidenced in Tables 3 and 4. It does not require key management beyond what is already used in RPL, nor does it impose per-fragment encryption or reassembly constraints. Its fragment chaining technique ensures that even temporarily trusted nodes cannot insert malicious fragments without breaking the hash sequence.

Most importantly, Predictive-CSM is inherently adaptive. It allows nodes to recover from transient failures and penalizes only consistent misbehavior. This flexibility not only reduces false positives but aligns the security mechanism with the dynamic nature of real-world IoT environments, where packet loss and timing irregularities are common and not always malicious.

Taken together, these comparisons make it clear: Predictive-CSM fills a critical gap left by prior methods. It introduces per-fragment security without heavy cryptographic load, detects advanced attacks like header replay that evade CSM, and preserves both energy and memory—making it highly deployable in today's constrained wireless sensor networks.

5. Overall Model Synthesis

Bringing the model together, the Predictive-CSM framework increases delivery success, reduces energy waste from malformed packets, and minimizes false positives. It does so using adaptive, self-healing trust mechanisms and stateless cryptographic checks that are computationally inexpensive.

In constrained IoT settings where memory, energy, and processing power are limited, Predictive-CSM achieves a superior trade-off between defense, performance, and sustainability compared to alternatives like key-exchange based authentication (which are too heavy) or signature-free systems (which are too permissive). It achieves real-time rejection of evolving threats while preserving the light footprint demanded by low-power embedded devices.

8 Conclusion

In this work, we introduced Predictive-CSM, a robust and lightweight security framework that enhances 6LoWPAN networks by integrating dynamic trust modeling with per-fragment integrity

validation. By extending the Chained Secure Mode with two complementary layers—an adaptive trust engine and a cryptographic hash-chaining mechanism—our approach addresses the core vulnerabilities associated with 6LoWPAN fragmentation, including buffer-reservation, fragment spoofing, and header-replay attacks.

Through detailed simulations using Contiki-NG and Cooja, we demonstrated that Predictive-CSM significantly improves delivery performance and security under both benign and adversarial conditions. Compared to existing solutions such as vanilla 6LoWPAN, CSM-integrated stacks, and cryptographically intensive methods like SecuPAN, our framework achieved higher packet delivery ratios, faster attacker detection, and lower power consumption—all without requiring substantial memory or computational overhead.

The trust dynamics model allowed nodes to continuously adapt to neighbor behavior, penalizing inconsistencies while preserving resilience in the face of transient disruptions. Meanwhile, the fragment-level hash chain provided a stateless, efficient method of validating data authenticity, ensuring that even fragments from once-trusted sources could not be used to compromise the system. Together, these two mechanisms created a security posture that was both responsive and scalable—key attributes for real-world IoT deployments where unpredictability and constraint are the norm.

Perhaps most importantly, Predictive-CSM offers a pragmatic security solution that does not trade off usability for robustness. Its design is compatible with current 6LoWPAN standards and can be integrated into existing protocol stacks with minimal changes. This makes it not just a theoretical improvement, but a viable candidate for securing next-generation wireless sensor networks in smart homes, industrial monitoring, and mission-critical sensing applications.

Future work will explore integrating physical-layer signal analysis for trust scoring, applying lightweight machine learning to detect stealthy attacks, and extending the framework for mobile IoT networks with dynamic topologies. Nonetheless, the results presented here make a compelling case that secure, efficient, and adaptive fragment handling is not only possible—but essential—for the evolving IoT landscape.

References

1. Aaqib M, Ali A, Chen L, Nibouche O (2023) Iot trust and reputation: a survey and taxonomy. Journal of Cloud Computing 12(1):42
2. Abbasi M, Al-Anbagi I (2019) Mitigation of fragmentation-based dos attacks in 6lowpan networks using early drop mechanisms. Ad Hoc Networks
3. Ahmad R, Wazirali R, Abu-Ain T, Almohamad TA (2022) Adaptive trust-based framework for securing and reducing cost in low-cost 6lowpan wireless sensor networks. Applied Sciences 12(17):8605
4. Alyami S, Alharbi R, Azzedin F (2022) Fragmentation attacks and countermeasures on 6lowpan internet of things networks: Survey and simulation. Sensors 22(24), URL `https://www.mdpi.com/1424-8220/22/24/9825`
5. Alyami S, Alharbi R, Azzedin F (2022) Fragmentation attacks and countermeasures on 6lowpan internet of things networks: Survey and simulation. Sensors 22(24), URL `https://www.mdpi.com/1424-8220/22/24/9825`
6. Bao F, Chen IR (2012) Dynamic trust management for internet of things applications. In: Proceedings of the 2012 International Workshop on Self-Aware Internet of Things, Association for Computing Machinery, New York, NY, USA, Self-IoT '12, p 1–6, DOI 10.1145/2378023.2378025, URL `https://doi.org/10.1145/2378023.2378025`

7. Elgendy H, Aly SM, Nabil A (2020) Trust-based model for secure routing in wireless sensor networks. Wireless Personal Communications 113:1–17
8. Fatima M, Rehman O, Rahman IMH, Ajmal A, Park SJ (2024) Towards ensemble feature selection for lightweight intrusion detection in resource-constrained iot devices. Future Internet 16(10), DOI 10.3390/fi16100368, URL https://www.mdpi.com/1999-5903/16/10/368
9. Firoozi F, et al (2021) A trust-based method for secure communication in internet of things using edge and fog computing. Journal of Systems Architecture
10. Ghosh A, et al (2018) Trustlite: Lightweight trust management scheme for resource-constrained iot devices. Ad Hoc Networks
11. Ghubaish A, Al-Rubaye M, Tsourdos A (2021) A comprehensive survey of trust management in iot. IEEE Internet of Things Journal 8(6):4022–4037
12. Glissa G, Rachedi A (2019) Secure and efficient data transmission for constrained iot devices: A survey. Computer Networks 149:113–133
13. Glissa G, Rachedi A, Meddeb A (2019) Secure and efficient data transmission for constrained iot devices: A survey. Computer Networks 149:113–133
14. Hongliang Tian ML (2025) A lightweight iot data security sharing scheme based on attribute-based encryption and blockchain. Computers, Materials, Continua 83(3):5539–5559, URL http://www.techscience.com/cmc/v83n3/60981
15. Hossain M, Karim Y, Hasan R (2018) Secupan: A security scheme to mitigate fragmentation-based network attacks in 6lowpan. In: Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, Association for Computing Machinery, New York, NY, USA, CODASPY '18, p 307–318
16. Hossain M, et al (2018) Secupan: A security scheme to mitigate fragmentation-based network attacks in 6lowpan. Proc ACM CODASPY
17. Hummen R, Hiller J, Wirtz H, Henze M, Shafagh H, Wehrle K (2013) 6lowpan fragmentation attacks and mitigation mechanisms. Association for Computing Machinery, New York, NY, USA, WiSec '13, p 55–66, DOI 10.1145/2462096.2462107, URL https://doi.org/10.1145/2462096.2462107
18. Jangra A, et al (2021) An adaptive trust-aware buffer management strategy for resource-constrained iot devices. Journal of Systems Architecture
19. Khan M, Hayat A (2023) Real-time anomaly detection in iot networks. International Journal of Information Security 22:245–257
20. Kumar S, Kumar D, Dangi R, Choudhary G, Dragoni N, You I (2024) A review of lightweight security and privacy for resource-constrained iot devices. Computers, Materials and Continua 78(1):31–63
21. Lakshmi V, et al (2021) A lightweight data authentication model for iot networks using incremental hmac verification. Journal of Network and Computer Applications
22. Lakshmi V, et al (2021) Lightweight hmac verification for 6lowpan fragments. J Network and Computer Applications
23. Li J, Wang L (2022) A lightweight cryptographic protocol for 6lowpan. IEEE Transactions on Information Forensics and Security 17:456–469
24. Liu Y, Wang J, Yan Z, Wan Z, Jäntti R (2023) A survey on blockchain-based trust management for internet of things. IEEE Internet of Things Journal 10(7):5898–5922, DOI 10.1109/JIOT.2023.3237893
25. Mansoor A, Ali Q (2023) Integrating blockchain for data integrity in iot. Journal of Systems Architecture 131:102590
26. Mendoza A, Jara AJ, Skarmeta AF (2015) Adaptive trust management for 6lowpan routing. Computer Networks

27. Okporokpo O, Olajide F, Ajienka N, Ma X (2023) Trust-based approaches towards enhancing iot security: A systematic literature review. URL https://arxiv.org/abs/2311.11705, 2311.11705
28. Osterlind F, Dunkels A, Eriksson J, Finne N, Voigt T (2006) Cross-level sensor network simulation with cooja. In: Proceedings of 31st IEEE Conference on Local Computer Networks, IEEE, pp 641–648
29. Palattella MR, Accettura N, Vilajosana X, Watteyne T, Grieco LA, Boggia G, Dohler M (2013) Standardized protocol stack for the internet of (important) things. IEEE communications surveys , tutorials 15(3):1389–1406
30. Raoof A, Lung CH, Matrawy A (2020) Introducing network coding to rpl: The chained secure mode (csm). URL https://arxiv.org/abs/2006.00310, 2006.00310
31. Sasi T, Lashkari AH, Lu R, Xiong P, Iqbal S (2024) A comprehensive survey on iot attacks: Taxonomy, detection mechanisms and challenges. Journal of Information and Intelligence 2(6):455–513
32. Sharma A, Gupta R (2021) An adaptive trust management system for iot devices. Journal of Network and Computer Applications 178:102918
33. Su B, Du C, Huan J (2020) Trusted opportunistic routing based on node trust model. IEEE Access 8:163077–163090, DOI 10.1109/ACCESS.2020.3020129
34. Sultana S, Khan A (2022) Trust-aware rpl for 6lowpan security. Sensors 22(6)
35. Sultana S, Khan A, Aslam N (2022) An efficient trust-aware routing protocol for securing 6lowpan in iot networks. Sensors 22(6):2295
36. Team CN (2018) Contiki-ng: The os for next generation iot devices URL https://contiki-ng.org
37. Thubert P (2020) Rfc 8930 - on forwarding 6lowpan fragments over a multi-hop ipv6 network. https://rfc-editor.org/rfc/rfc8930.txt
38. Wang J, Liu H, Zhang F (2020) A blockchain-based lightweight data authentication protocol for 6lowpan. IEEE Internet of Things Journal
39. Xie L, Wang C, Zhou W (2020) A secure fragmentation mechanism for low-power networks. In: Proceedings of IEEE ICC
40. Yousuf A, Ismail ASKP (2017) Trust management in wireless sensor networks: an overview. Journal of Sensors
41. Zhang Y, Chen X (2022) A hybrid trust-based framework for iot security. Future Generation Computer Systems 128:103–113