Francis Mendoza
(fmendoz7@asu.edu)

Arizona State University

AEGIS

A Special-Purpose Computer Network For Strategic Cyber Defense

MOTIVATION: Critical Infrastructure Is Insecure

RESEARCH

51

GhostStripe attack haunts self-driving cars by making them ignore road signs

Cameras tested are specced for Baidu's Apollo

Laura Dobberstein

Fri 10 May 2024 // 14:04 UTC

PUBLIC SAFETY

Report: Chinese hackers targeted Texas power grid, Hawaii water utility, other critical infrastructure

BY CRAIG HUBER | NATIONWIDE
UPDATED 8:30 AM CT DEC. 12, 2023

ICS/OT

Kansas Water Facility Switches to Manual Operations Following Cyberattack

Ransomware possibly involved in a cybersecurity incident at Arkansas City's water treatment facility.

By Ionut Arghire
September 24, 2024

Russia-linked hacking group claims to have targeted Indiana water plant

By Sean Lyngaas, CNN
2 minute read · Published 4:08 PM EDT, Mon April 22, 2024

The Tipton, Indiana, wastewater treatment plant. From Tipton Municipal Utilities

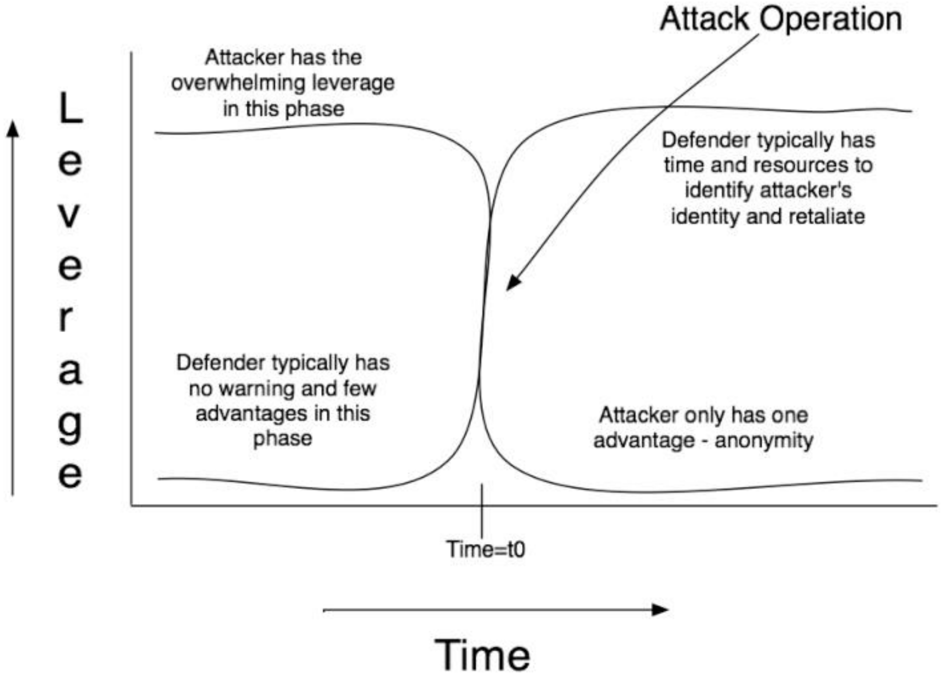
Half of workers in critical industry hit by cyber attacks – IoT is to blame, says Verizon

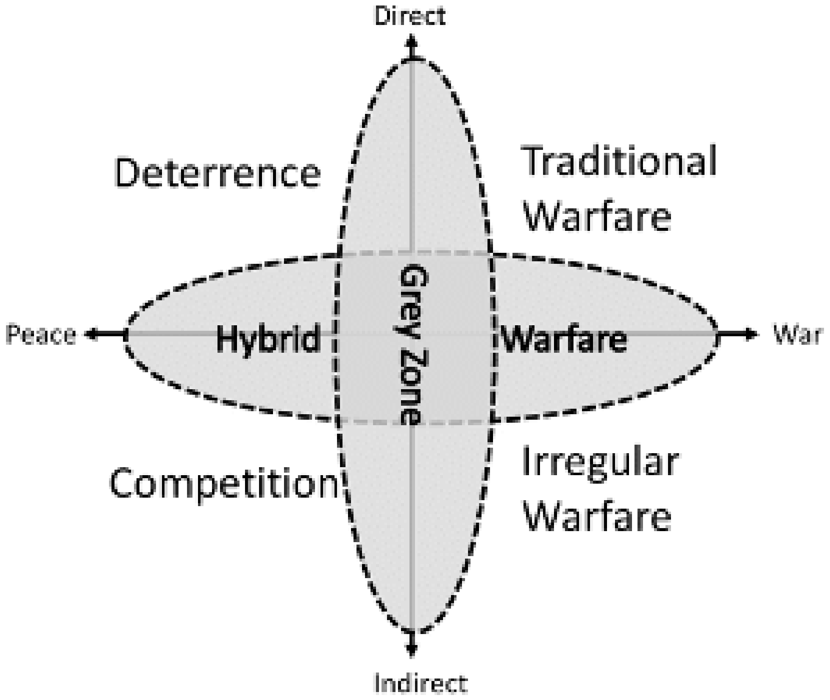
By James Blackman August 6, 2024

IoT Security

MOTIVATION: Why Cyber Defense Is Hard

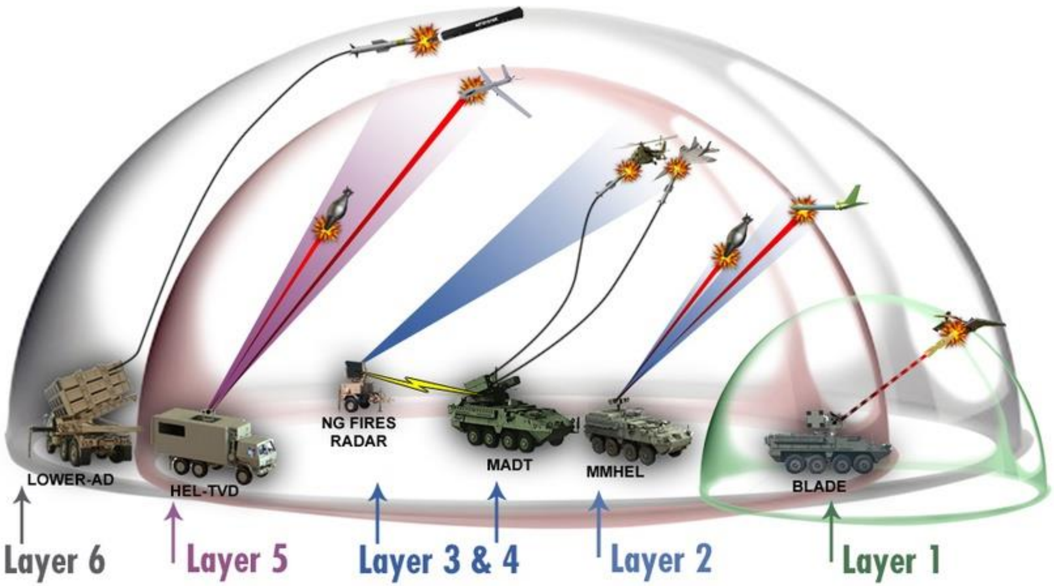
- **Conventional Military/"Kinetic" Solutions:** Impractical, high risk of escalation.
- **Legal Warfare/"Lawfare":** Ineffective, only usable in countries with mutual extradition treaties.
- **Counterhacking**: Difficult due to attribution, only an option for governments.
- The nature of cyberwarfare fundamentally operates in a **grey area** + asymmetrically favoring attacker.





MOTIVATION: Strategic Cyber Defense

- Currently, the security posture between different systems is often poor, greatly varies + largely independent.
- **Question:** Is a universal cyberspace equivalent of strategic missile defense *possible*?
- **Addendum**: Is an effective cyberspace defense doctrine *equivalent* of M.A.D possible?



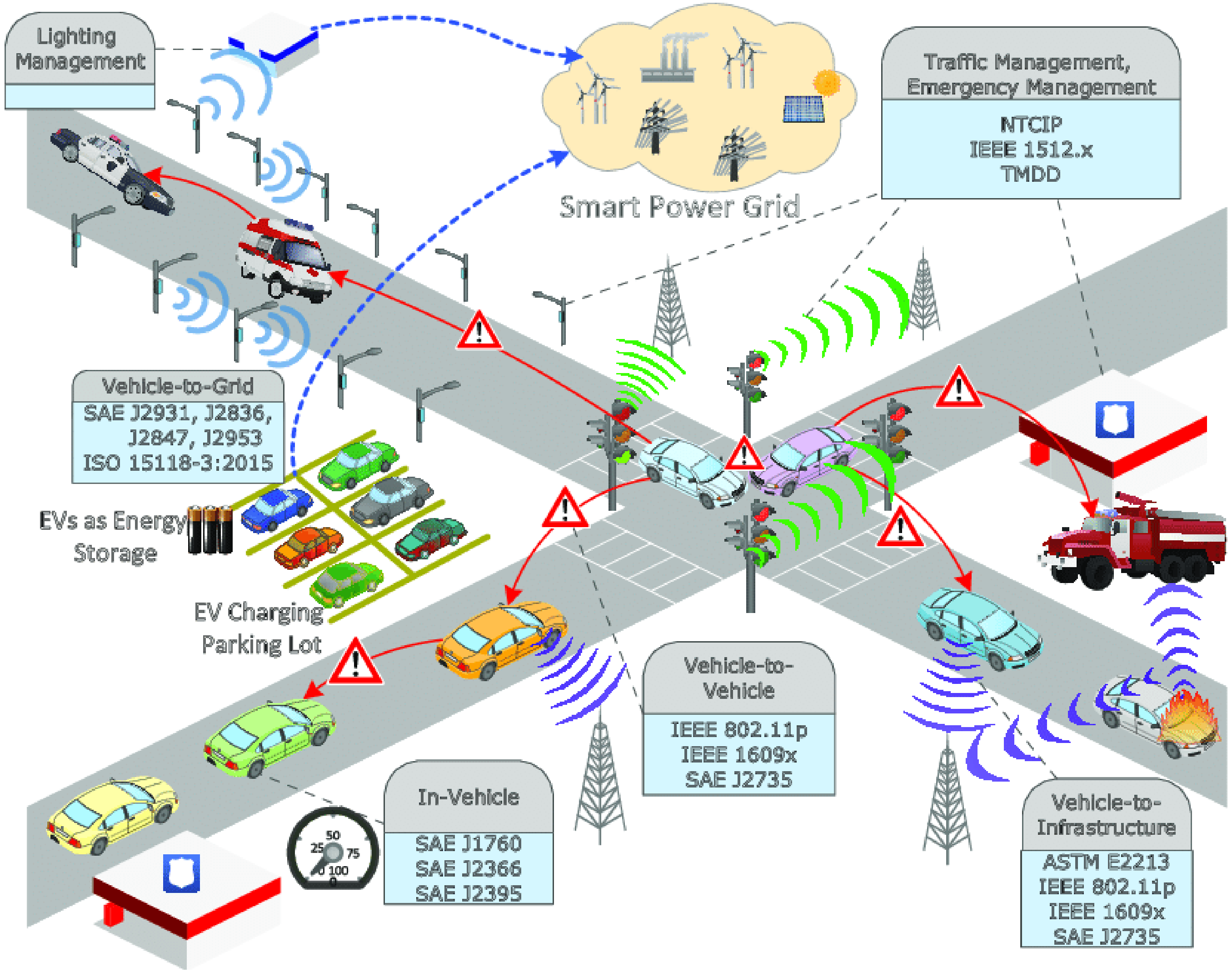
Our Approach

Cyber Defense + Cyber Deterrence Through Universally Hard Computational Cost

AEGIS addresses this problem through:

1. Extremely High Randomness (ERIS)
2. Rapid Real-Time Detection & Adaptation (ATHENA)
3. Universally Hard Computational Cost as a punitive deterrent (M-PoW)

BACKGROUND: Cyber-Physical Systems (CPS)



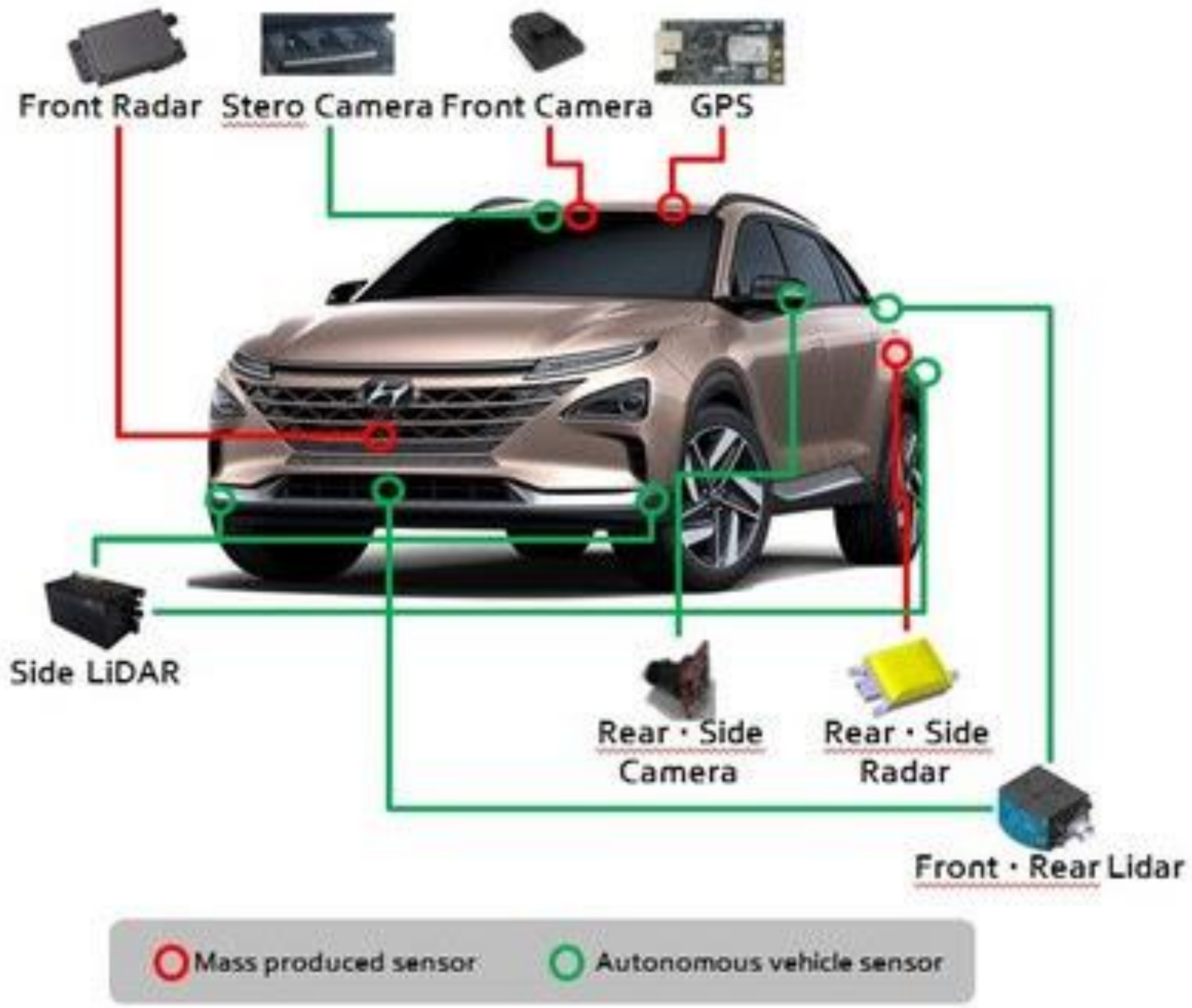
20 Billion (Targets)

- Motivations for IoT hacks: **access greater network** or **farm bandwidth**
- At least **90%** of all IoT devices talk over unencrypted channels.
- Overwhelming amount of IoT devices have **very poor security!**
- Bandwidth farming via IoT botnet great for launching DDoS attacks.
- **Every endpoint is a potential attack vector. The threat compounds.**

Use Case: Smart City Infra

- Sub-CPS in smart grid, traffic infrastructure, V2V mesh network, etc. share data.
- Nodes can be compromised **directly or indirectly.**

BACKGROUND: Individual Device (CAVs)



<u>Autonomous Vehicle Attacks</u>

- Driving policy calculations are **isolated & done locally**, but there are ways to compromise operation.
- Several attack vectors exist, but we will focus on layers 3, 4, 7 of OSI model
- **Malicious OTA Firmware Injection**: Inject firmware that may spoof sensor readings or cause incorrect operation of key components.
- **Sybil Attacks**: Spoof number & location of other vehicles.
- **DDoS Attacks**: Overwhelm tertiary vehicle systems to increase latency, shut down subsystems.
- ***And many, many more!***

APPROACH: AEGIS Network Topology

- A "Forest-Of-Trees" Topology, constituting a network with layers of devices composed of varying capabilities
- Holistic, Defense-In-Depth, Moving-Target. Network is closed, hardened, and microsegmented.

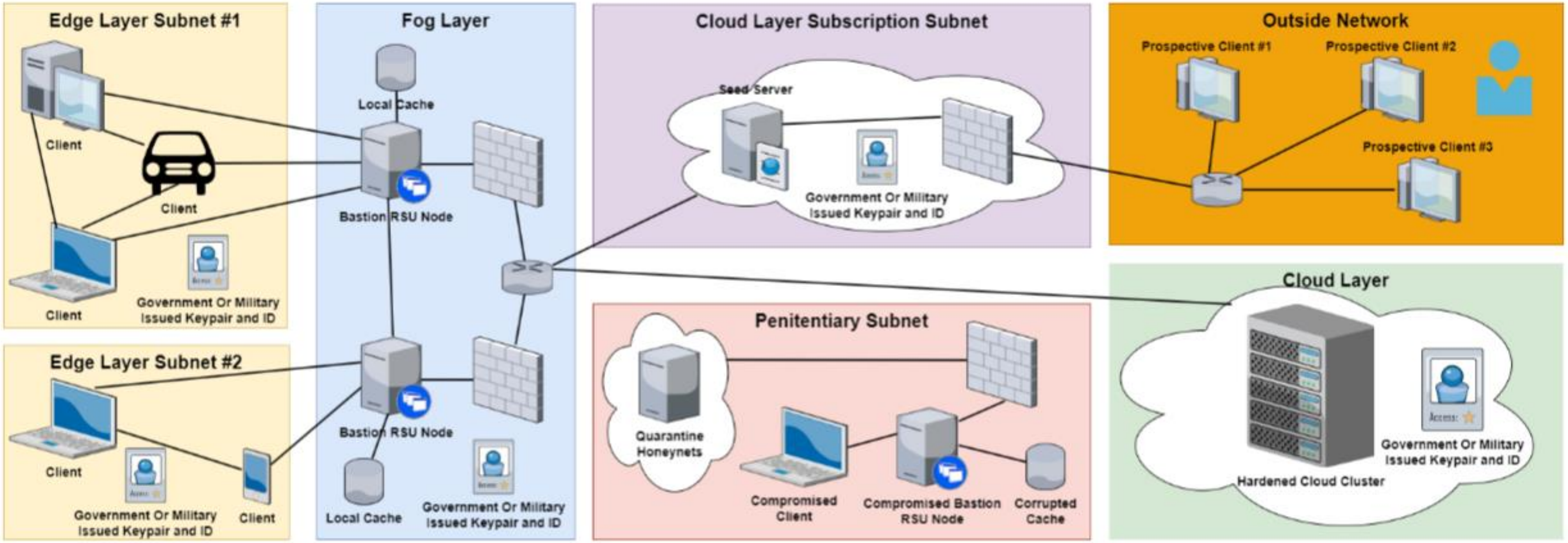
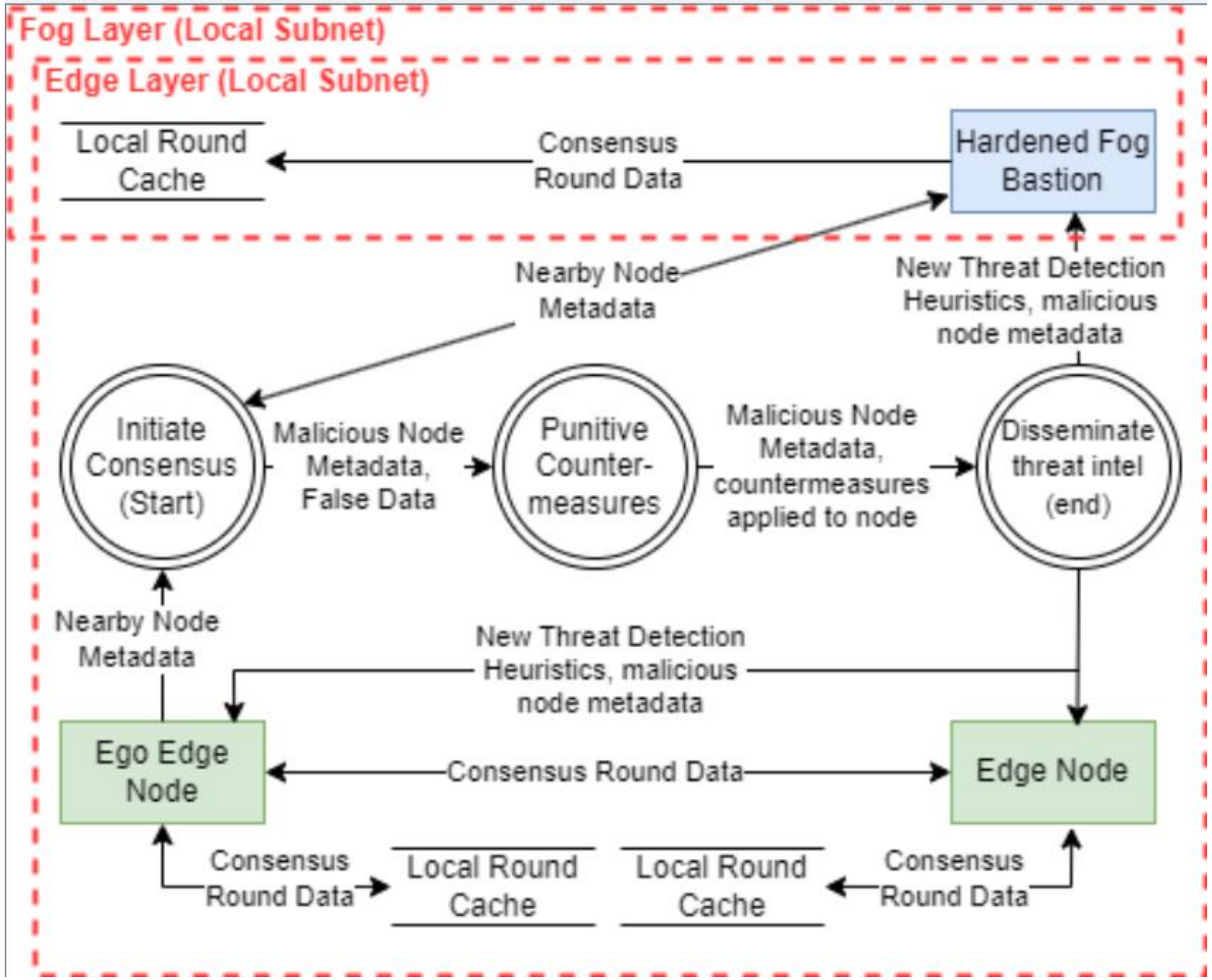
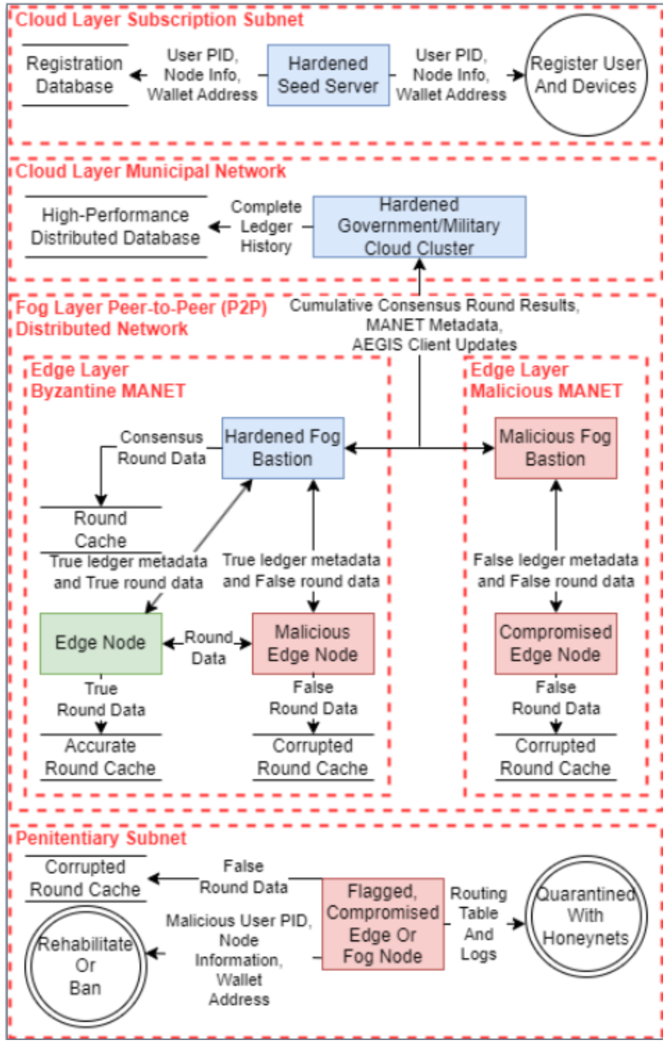


Figure 3.3: AEGIS topology diagram with different responsibilities and capabilities.

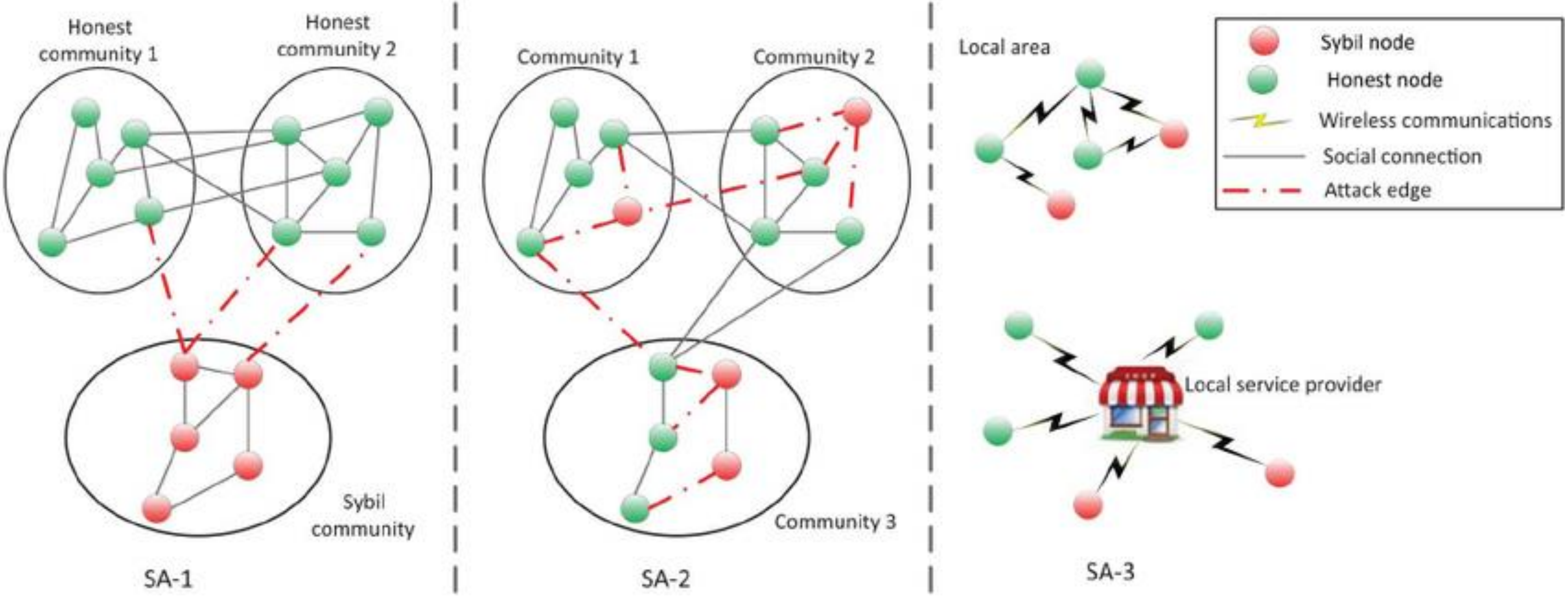
APPROACH: Threat Model (Local Subnet)



APPROACH: Threat Model (Whole Network)



APPROACH: ERIS for Moving-Target Defense



APPROACH: ERIS for Moving-Target Defense

Algorithm 3: Dynamic Subnet Allocation for ERIS

1: Initialize network node list N and subnet list S
2: Define maximum subnet size $maxSize$
3: **for** each node $n \in N$ **do**
4: Calculate potential subnets based on proximity and current entropy
5: Assign node n to subnet $s \in S$ that maximizes entropy
6: **if** size of subnet s exceeds $maxSize$ **then**
7: Trigger reconfiguration for subnet s
8: **end if**
9: **end for**
10: **return** Updated subnet list S

$$H(S) = -\sum_{i=1}^{k} p_i \log p_i \tag{4.1}$$

where $H(S)$ is the entropy of subnet configuration S, p_i represents the proportion of nodes in the i-th subnet, and k is the total number of subnets.

$$\text{Trigger Reconfiguration if } H(S) < H_{\text{threshold}}$$

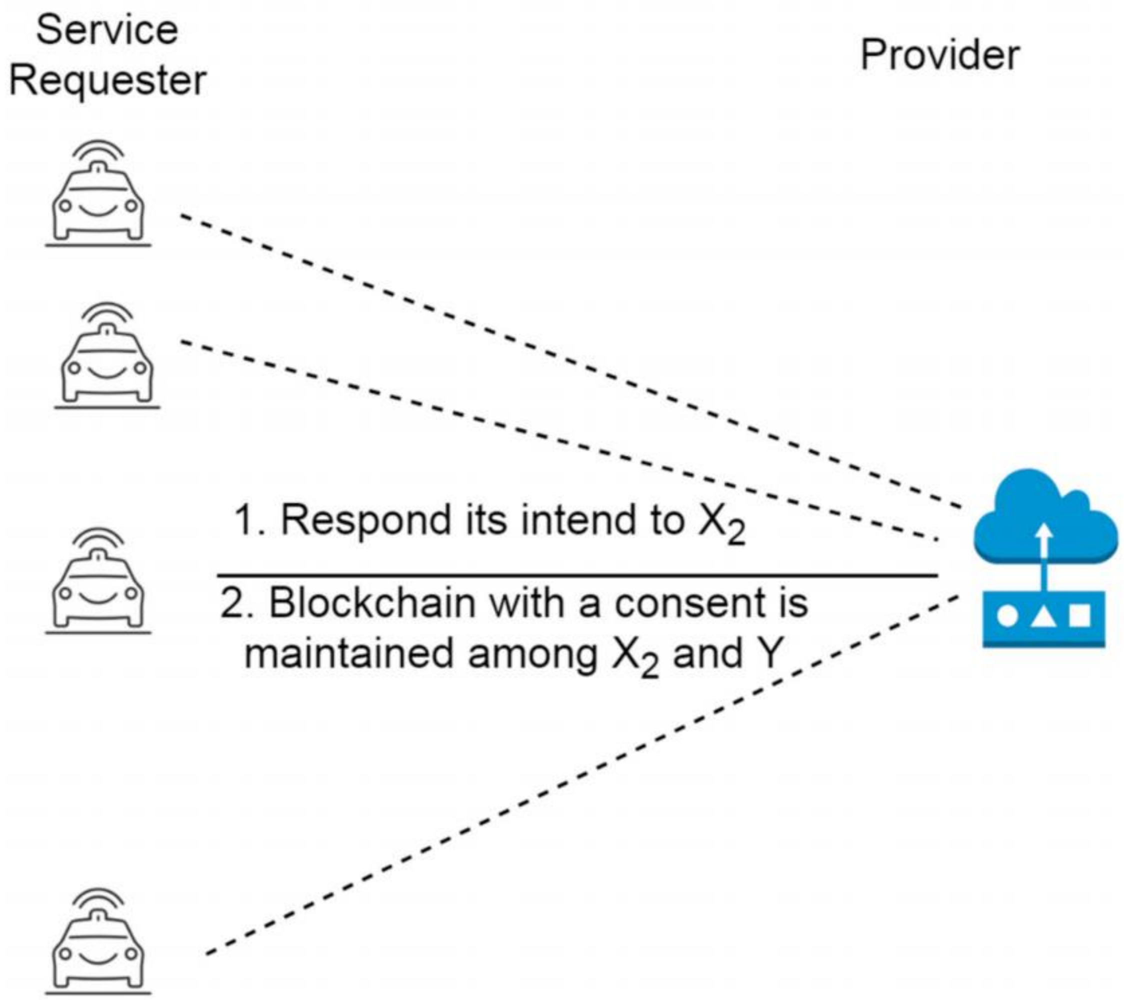
APPROACH: ERIS for Moving-Target Defense

- $P_M(t)$: Probability of controlling the MANET within time limit (t).
- $P_{V_i}(t)$: Probability of controlling the i^{th} security group within time limit (t).
- $P_{C_{ph}}(t)$: Probability of retaining control in the MANET despite churn to make a successful attack within time (t).
- $P_{C_{v_i}}(t)$: Probability of retaining control in the i^{th} security group despite churn to make a successful attack within time (t).

The comprehensive success probability is expressed below in equation 4.3:

$$P_s(t) = P_M(t) \cdot P_{C_{ph}}(t) \cdot \prod_{i=1}^{l} \left(P_{V_i}(t) \cdot P_{C_{v_i}}(t)\right) \tag{4.3}$$

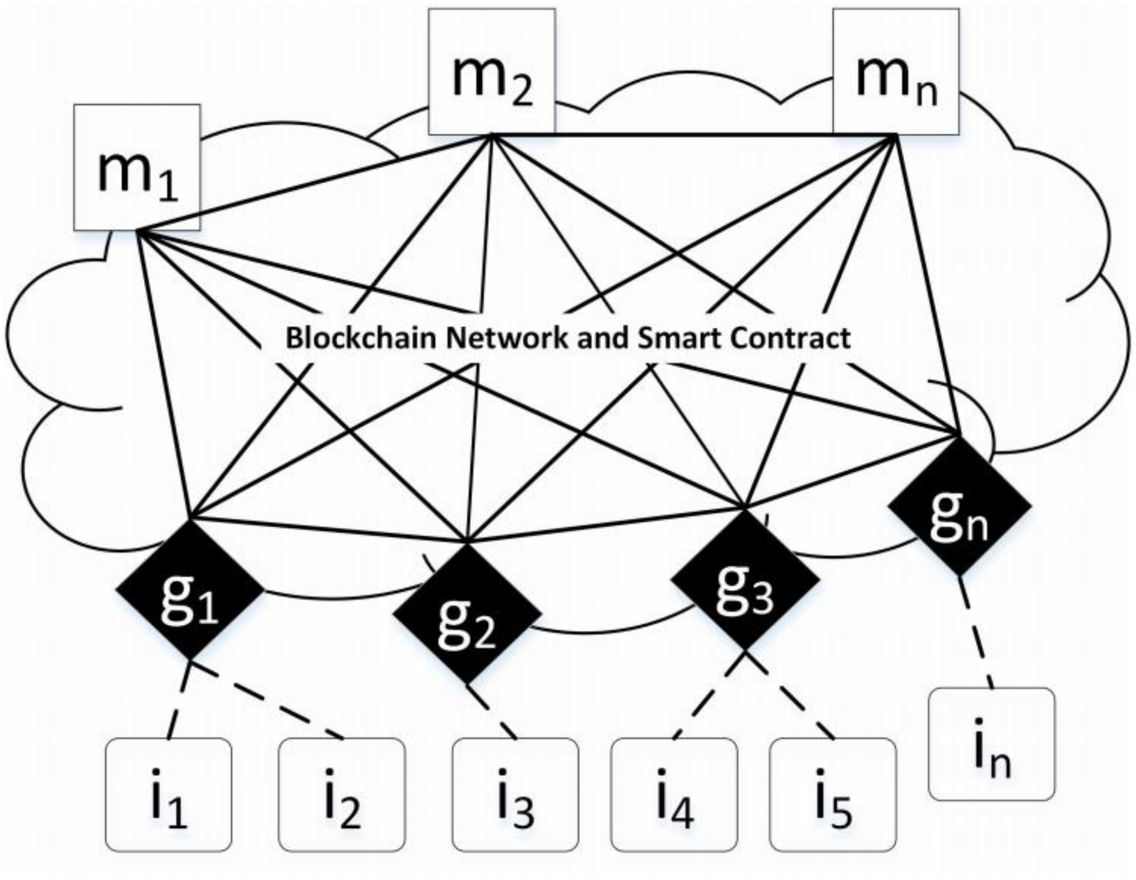
APPROACH: ATHENA For Threat Detection



Algorithm 4: Handshake Algorithm in AEGIS

```
Input: ego_node, nearby_nodes[], rsu
ego_hash ← hash(targeting_service_code(ego_node))
rsu_hash ← hash(targeting_service_code(rsu))
if ego_hash ≠ rsu_hash then
  Abort handshake. ego_node cannot form or join a MANET under this rsu.
else
  candidate_nodes[] ← ∅
  for all node ∈ nearby_nodes[] do
    node_hash ← hash(targeting_service_code(node))
    if node_hash = rsu_hash then
      candidate_nodes[].add(node)
    end if
  end for
  if size(candidate_nodes[]) > predefined_threshold then
    Form MANET with ego_node and candidate_nodes[]
  else
    Output failure to form MANET
  end if
end if
Output: Formed MANET or failure indication.
```

APPROACH: ATHENA For Threat Detection



Algorithm 5: Heartbeat Protocol in AEGIS

```
Input: final_round_data, local_RSU_cache, kademlia_DHT
malicious_nodes, honest_nodes, nonresponsive_nodes ←
    ExtractNodes(final_round_data)
for node ∈ malicious_nodes do
    UpdateCacheAndDHT(node, local_RSU_cache, kademlia_DHT, 'malicious')
    behavior_type ← IdentifyMaliciousBehavior(node)
    if Not PreviouslyRecorded(behavior_type) then
        DisseminateThreatIntelligence(behavior_type)
        UpdateTargetingServiceWithHeuristic(behavior_type)
    end if
end for
for node ∈ honest_nodes do
    UpdateCacheAndDHT(node, local_RSU_cache, kademlia_DHT, 'honest')
end for
for node ∈ nonresponsive_nodes do
    UpdateCacheAndDHT(node, local_RSU_cache, kademlia_DHT, 'nonresponsive')
end for
Output: Updated kademlia_DHT and local_RSU_cache, Dissemination of new threat
    intelligence (if applicable)
```

APPROACH: M-PoW Cost Deterrent

Algorithm 6: Dual Consensus Protocol in AEGIS

```
Input: subnet_data, network_state_components
violating_nodes ← BOSCO(subnet_data, network_state_components)
for node ∈ violating_nodes do
    if TargetingService(node) == "malicious" then
        ExponentialSlidingCost(node)
        MedusaStunlock(node)
    end if
end for
validated_txns ← ProofOfWork(subnet_data)
for txn ∈ validated_txns do
    Add txn to subnet transaction pool
end for
Output: Updated node statuses in the subnet, Validated transactions for the subnet,
    Nodes flagged as malicious by TargetingService
```

Dynamic Difficulty Adjustment

A modified version of the classical dynamic difficulty adjustment formula is designed to adapt the mining difficulty based on the rate of transactions and the current network load to ensure computational feasibility for IoT devices:

$$D(t) = D_0 \cdot \left(1 + \alpha \left(\frac{\overline{\lambda}(t)}{\lambda_{\text{ref}}}\right)\right) \tag{4.5}$$

where:

- $D(t)$: Difficulty at time t.
- D_0: Base difficulty.
- α: Adjustment factor, which scales the difficulty based on network conditions.
- $\overline{\lambda}(t)$: Average transaction rate at time t.
- λ_{ref}: Reference transaction rate for normal operation.

APPROACH: M-PoW Cost Deterrent

Churn Factor for Dynamic Difficulty

A new formula where the churn factor adjusts the difficulty in response to the rate of node churn in the network, reducing the difficulty to accommodate sudden drops in network participation, as long as the network size stays within a sufficient range to be sufficiently resilient against byzantine faults:

$$\text{ChurnFactor}(t) = \exp\left(-\beta \cdot \left|\frac{d|\mathcal{N}_i(t)|}{dt}\right|\right) \tag{4.6}$$

where:

- β: Sensitivity parameter that modulates the effect of churn.
- $\mathcal{N}_i(t)$: Number of active nodes in the network at time t.

APPROACH: M-PoW Cost Deterrent

Stair-Stepping Difficulty Levels

A modified version of the classic stair-stepping algorithm provides more gradual changes in difficulty to prevent large fluctuations and maintain stability:

$$D(t + \Delta t) = D(t) \cdot \left(1 + \gamma \cdot \text{Sign}\left(\frac{\Delta\bar{\lambda}}{\Delta t}\right) \cdot \text{ChurnFactor}(t)\right) \tag{4.7}$$

where:

- Δt: Time increment for difficulty adjustment.
- γ: Step size for difficulty adjustment.
- $\Delta\bar{\lambda}$: Change in the average transaction rate.

APPROACH: M-PoW Cost Deterrent

Probabilistic and Bounded Cost Functions

This modified function accounting for churn ensures that the computational cost remains within a feasible range while still being probabilistic:

$$P(t) = \frac{1}{1 + \exp\left(-\xi\left(\overline{\lambda}(t) - \lambda_{\text{target}}\right)\right)} \tag{4.8}$$

$$\text{Cost}(t) = \text{BaseCost} \cdot \left(1 - \frac{\text{ChurnFactor}(t)}{\theta}\right) \tag{4.9}$$

where:

- $P(t)$: Probabilistic cost function at time t.
- ξ: Factor controlling the sensitivity to deviations from the target rate λ_{target}.
- θ: Normalization factor to ensure the cost stays within bounds.

EXPERIMENTS: Physical Network Setup

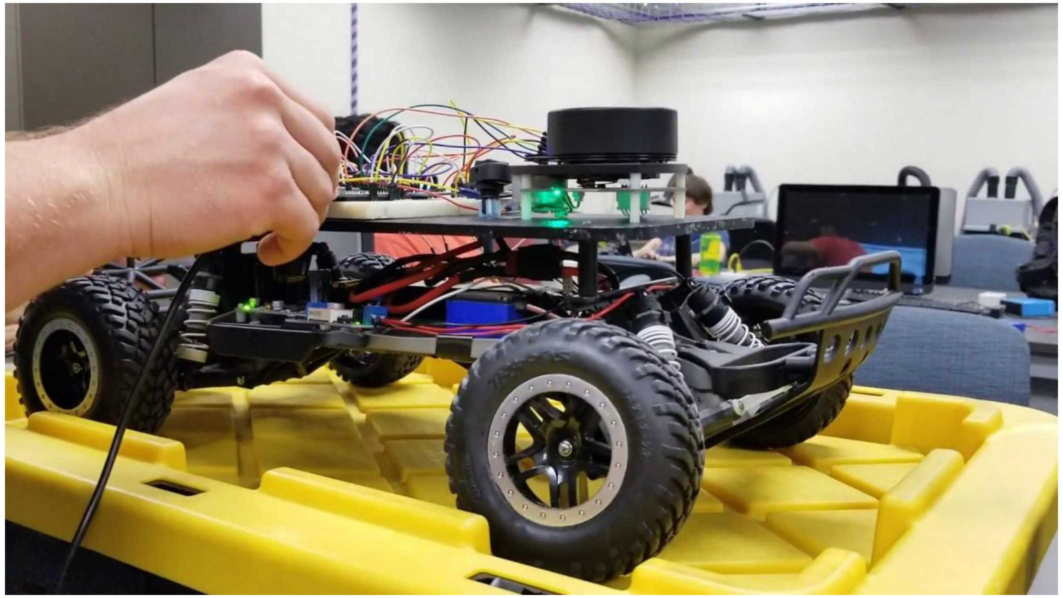
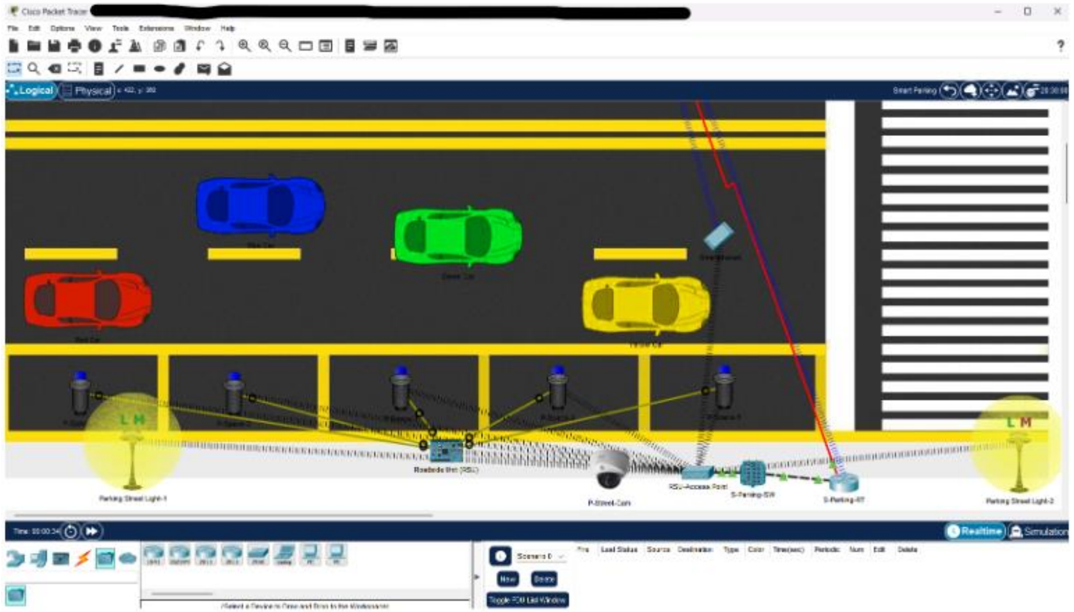
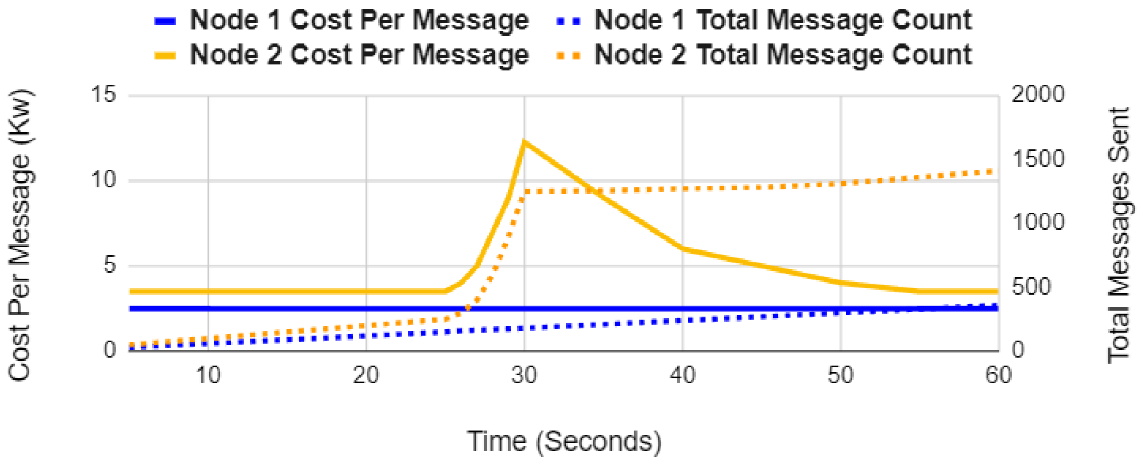


Figure 7: Cluster of Raspberry Pi 4's used to model the fog layer (left) and GPU rig used to model the cloud layer (right).

EXPERIMENTS: Realistic Virtual Network Emulation

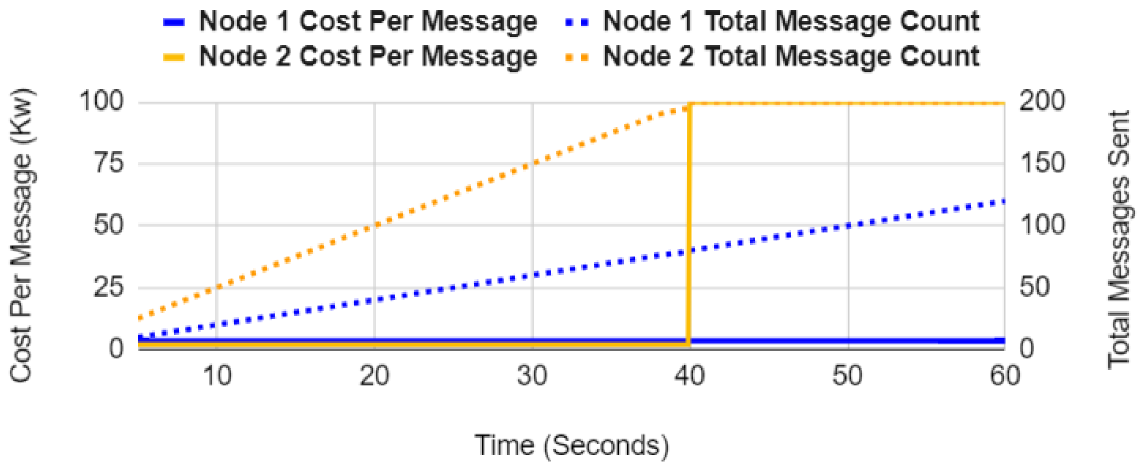


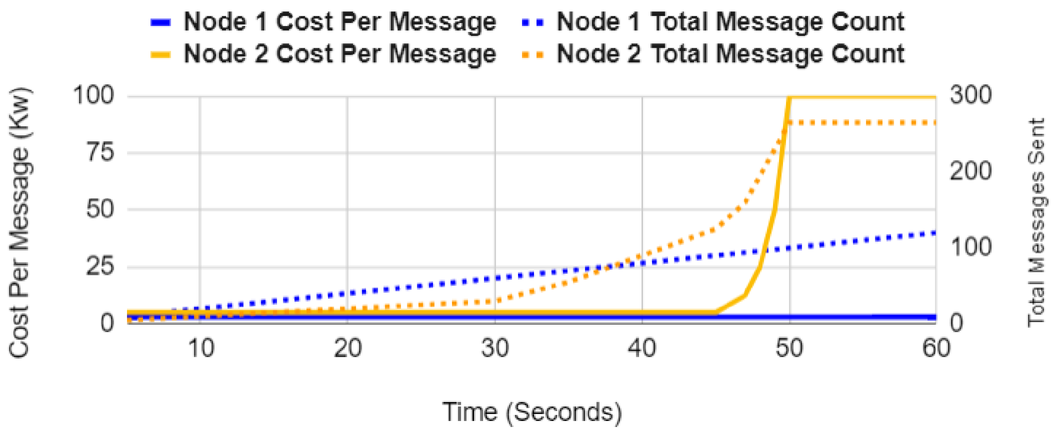
RESULTS: Punitive Cost Deterrent For Network Attacks



Results

- **Graph #1**: Node begins a DDoS but stops and ceases attack, burning down fee over time to normal
- **Graph #2**: Node begins a DDoS until it runs out of compute, is quarantined, and subsequently banned
- **Graph #3**: Node attempts non-volumetric attack (ie: zip-bomb), gets detected by AEGIS, and has its holdings slashed all at once

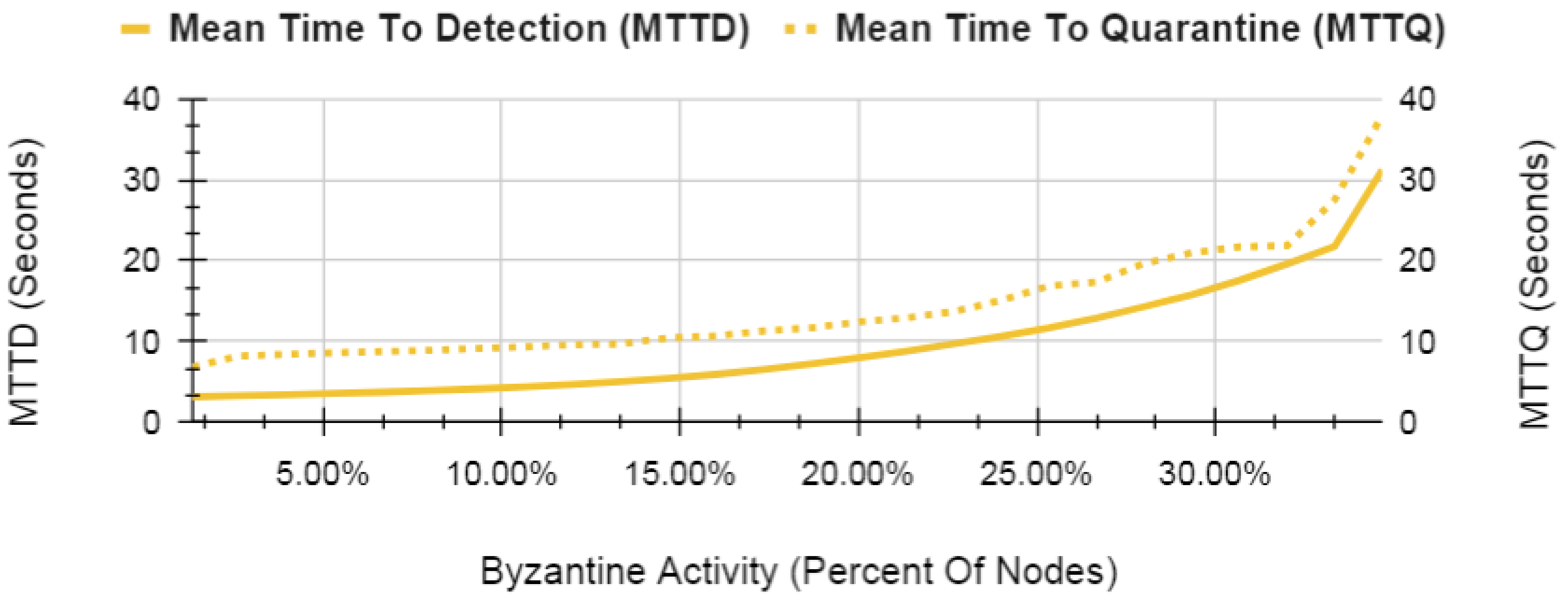




RESULTS: Holistic Defense Against A Variety Of Attacks

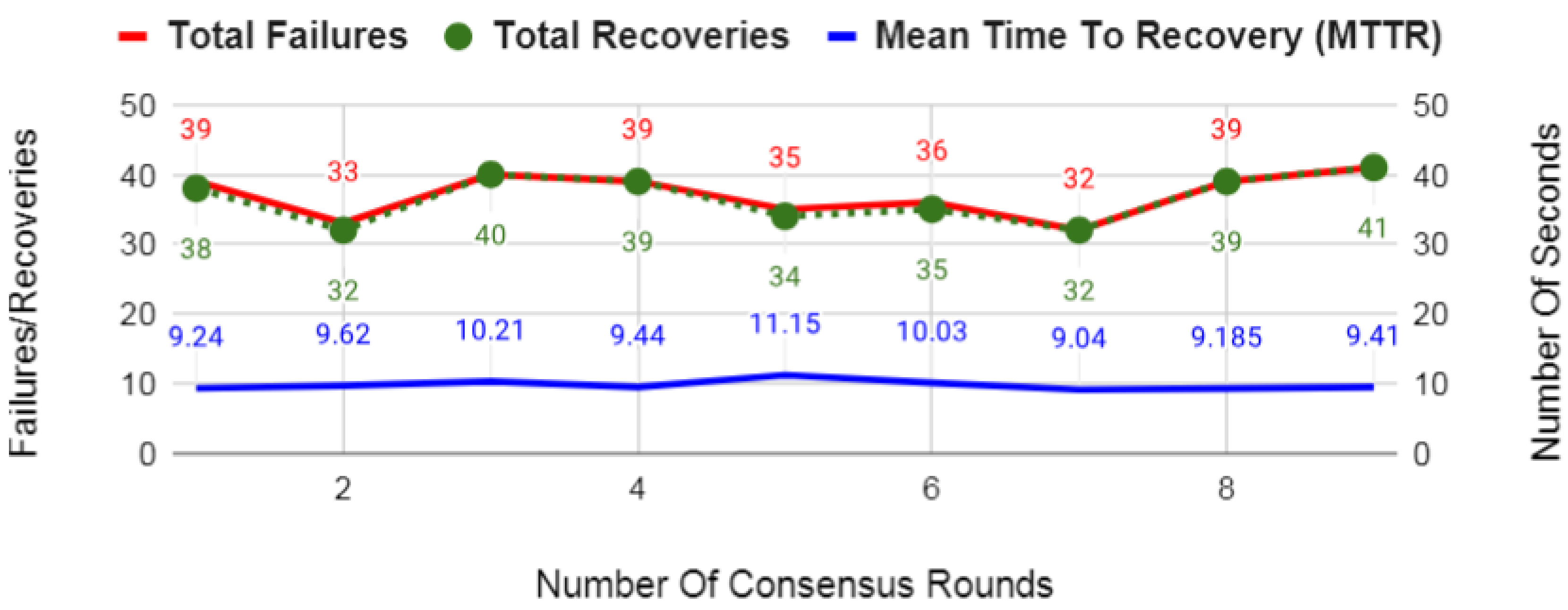
Type Of Attack (MITRE ATT&CK ID)	Countermeasure
Eavesdropping **(T1430)**	Uses AES-256 encryption at rest and TLS 1.3 in transit with mutual authentication to ensure data confidentiality.
Sybil Attacks **(T1098)**	Requires cryptographic staking tied to device identity; high resource costs deter fake identities.
Man-in-the-Middle Attacks **(T1557)**	Utilizes TLS 1.3 with mutual authentication; dual-consensus detects anomalies; ERIS reduces predictability.
Replay Attacks **(T1003)**	Implements time-stamped messages and nonces; dual-consensus validates freshness; ATHENA monitors patterns.
Message Tampering **(T1565)**	Uses digital signatures and integrity checks; consensus mechanisms detect alterations; ATHENA responds.
Wormhole Attacks **(T1430)**	ERIS's dynamic subnet formation hinders wormholes; ATHENA detects routing anomalies.
Blackhole Attacks **(T1499)**	Dual-consensus identifies malicious nodes; ATHENA quarantines them; reroutes communications.
Jamming Attacks **(T1495)**	Detects communication disruptions; devices switch frequencies or use alternatives when possible.
Spoofing Attacks **(T1556)**	Employs PKI with RSA 2048-bit encryption and device certificates to prevent impersonation.
DoS and DDoS Attacks **(T1498)**	Adaptive rate limiting and resource metering; high-attrition defense increases attackers' costs.
Routing Attacks **(T1592)**	ERIS prevents routing manipulation; dual-consensus validates routing; ATHENA detects anomalies.
Side-Channel Attacks **(T1407)**	Implements constant-time cryptography; isolates sensitive operations; hardware security modules used.

RESULTS: AEGIS Quickly Detects & Quarantines Threats



As the number of byzantine nodes is scaled in a 50-node network, AEGIS Mean Time To Detection (MTTD) and Mean Time To Quarantine (MTTQ) of the network increases, however the network remains effective at removing threats until the 33% byzantine fault tolerance **(3f + 1)** threshold.

RESULTS: AEGIS Has High Resiliency & Quick Recovery



Resiliency and Recovery metrics over various consensus rounds utilizing ERIS. With 366 recoveries/369 failures, AEGIS demonstrates a **99.2% recovery rate.** Mean-Time-To-Recovery averages at **9.73 seconds/subnet**.

RESULTS: AEGIS Is Performant & Power-Efficient

Table 2: Comparative Analysis of Time-to-Finality

Consensus Mechanism	Time-to-Finality (Seconds)
AEGIS	0.3 to 5 seconds
Honey Badger BFT	1 to 3 seconds
IOTA	10 seconds
Hashgraph	3 to 5 seconds

Table 3: Comparative Analysis of AEGIS vs. Hashcash.

Metric	AEGIS	Hashcash	% Diff.	Ratio
Attempts	15.35	1.48M	-99.999%	1.04×10^{-5}
Elapsed Time (s)	1.33×10^{-5}	0.964	-99.999%	1.38×10^{-5}
Hashpower Util.	594.38	1.54M	-99.961%	3.86×10^{-4}
Kilowatt-Hours (kWh)	1.84×10^{-11}	1.34×10^{-6}	-99.999%	1.38×10^{-5}
Cost (USD, $0.13/kWh)	2.40×10^{-12}	1.74×10^{-7}	-99.999%	1.38×10^{-5}
Cost (BTC)	5.99×10^{-17}	4.35×10^{-12}	-99.999%	1.38×10^{-5}

WARFARE IS ALWAYS CHANGING