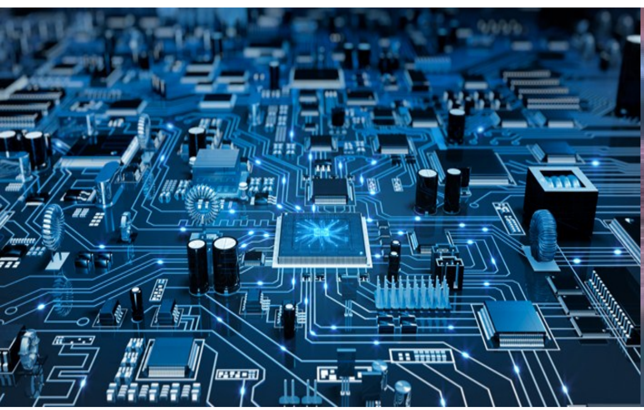
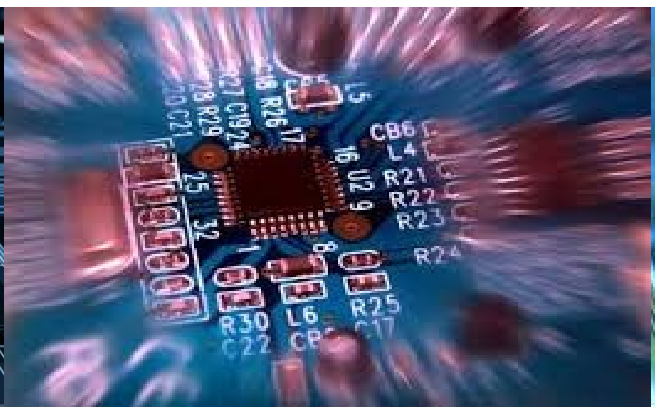
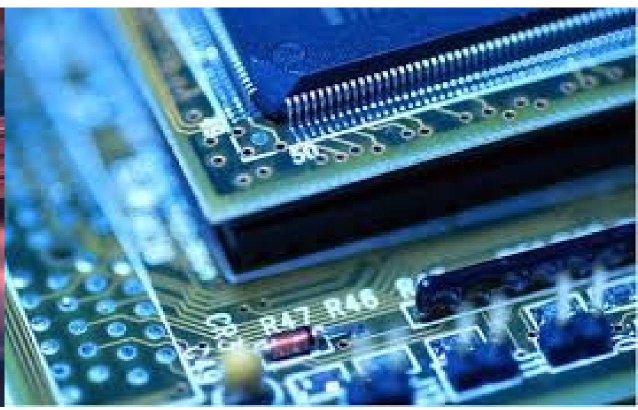
A Mutual Auditing Framework to Protect IoT against Hardware Trojans

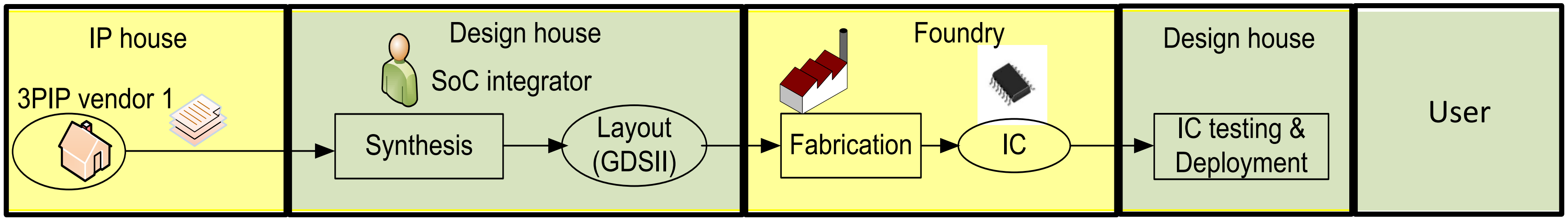
Chen Liu, Patrick Cronin, and **Chengmo Yang**

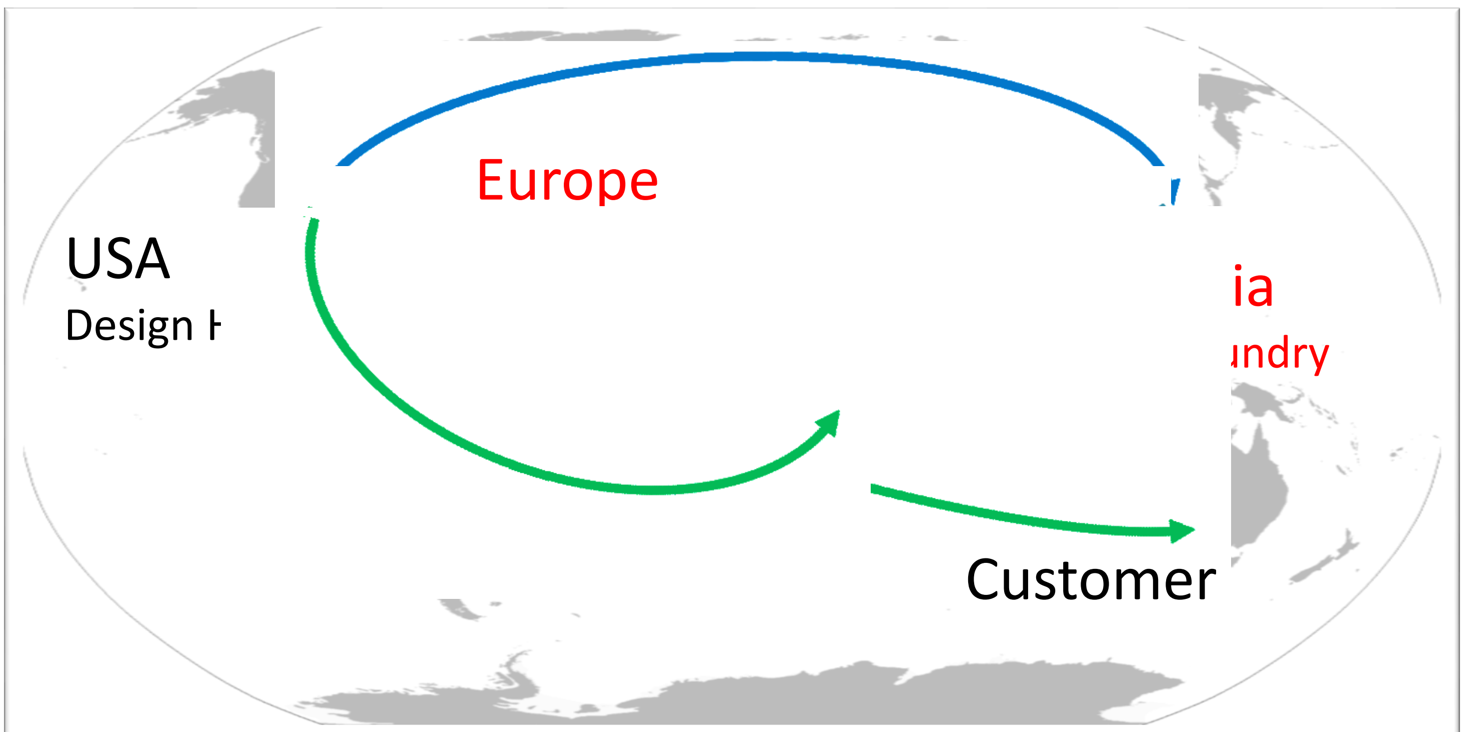
10/03/2016

Outline

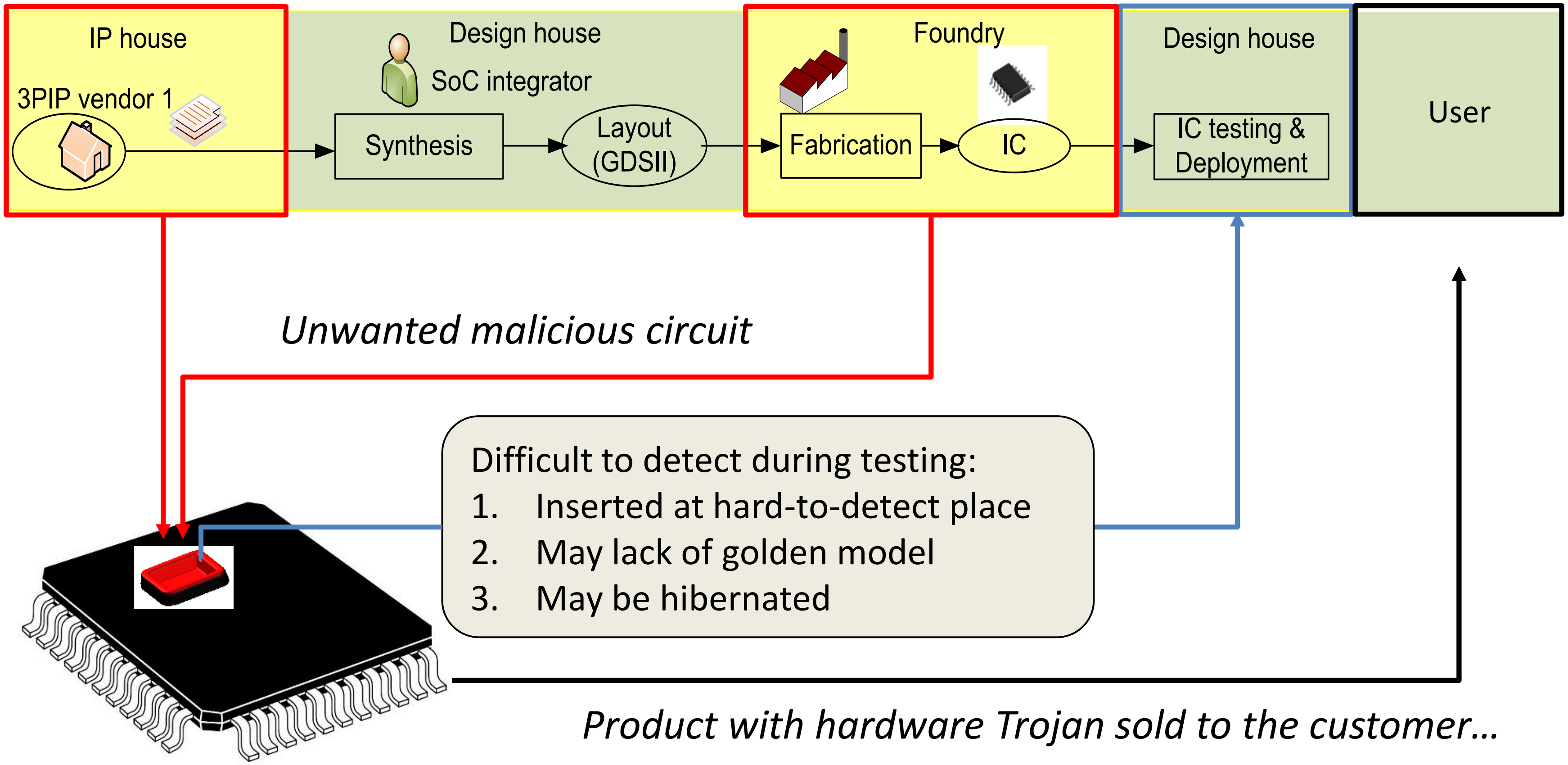
- Hardware Trojan in IoT
- Proposed Trojan detection scheme
- Simulation results
- Summary

Hardware Trojan: malicious elements inserted in circuit



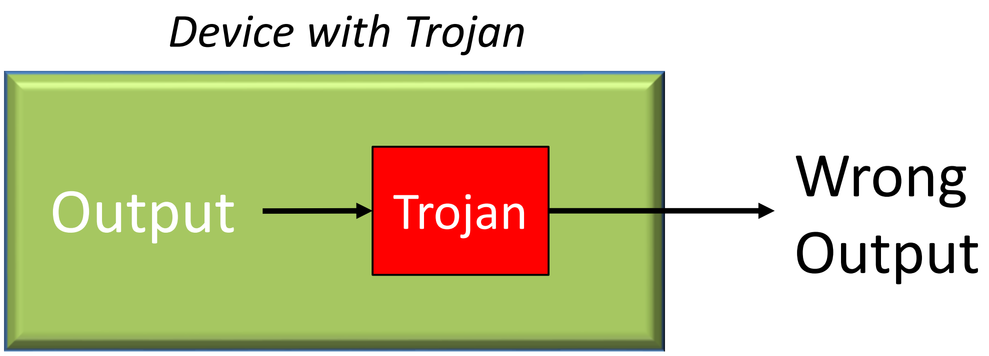


Hardware Trojan: malicious elements inserted in circuit

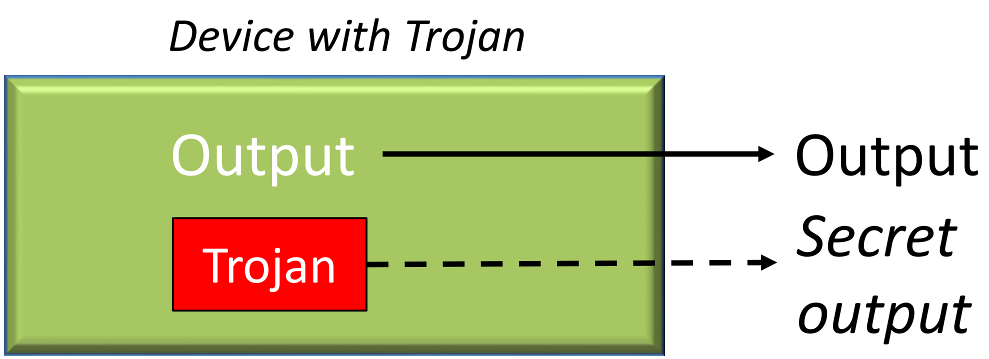


A Hardware Trojan may…

tamper output



send secret message

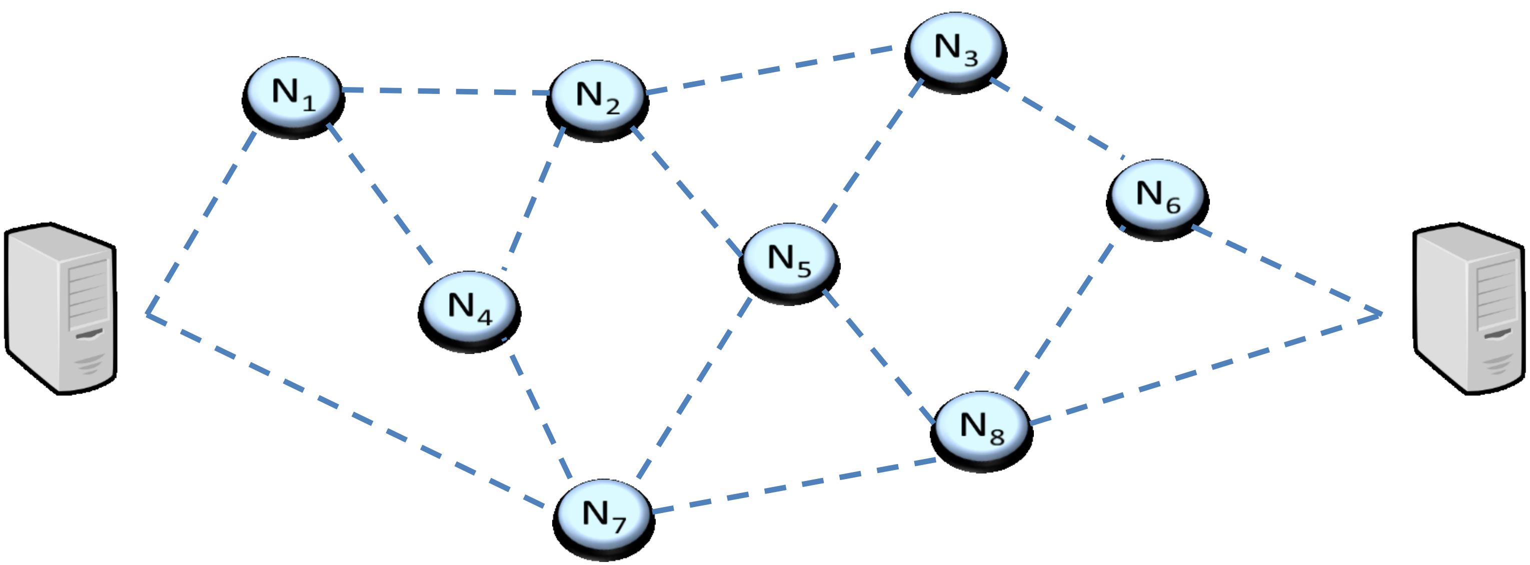


Hardware Trojans in a network may…

Internet of Things (IoT)

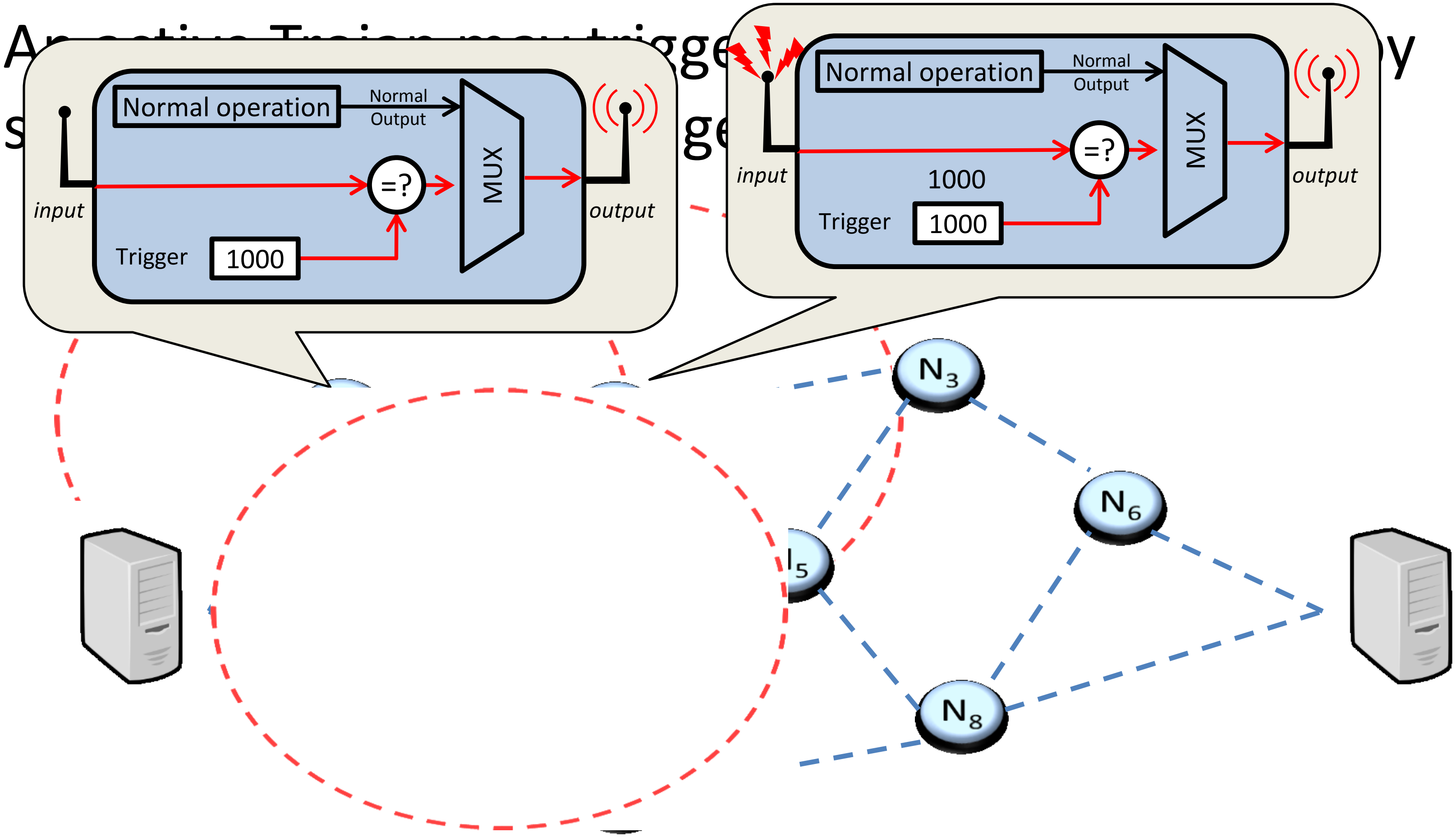
Trojans?

Wireless communication



Nodes

Servers



Entire network down in a short while

Fault tolerance does not work

Catastrophe

Why difficult?

Problem to solve: hardware Trojan collusion in IoT

Previous hardware Trojan countermeasures [1,2,3] may detect single Trojan but not **Trojan collusion**

Previous IoT security solutions target attacks from outside of the network [4,5,6] but not **attacks from the inside**

Our goal: prevent hardware Trojan in IoT from mutually triggering

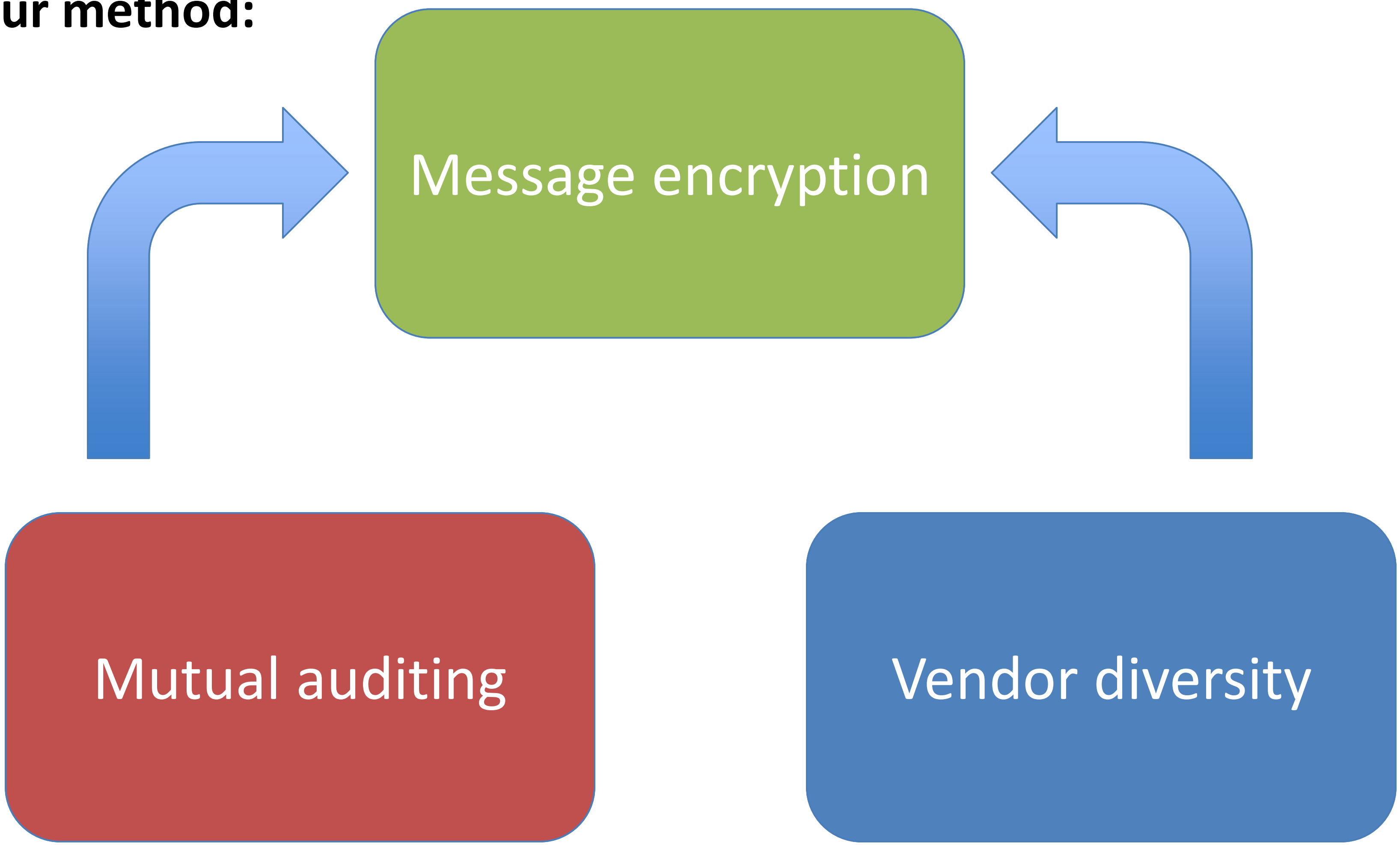
[1] M. Banga and M. S. Hsiao, "A Novel Sustained Vector Technique for the Detection of Hardware Trojans"
[2] S. Bhunia, M.S. Hsiao, M. Banga, and S. Narasimhan. "Hardware Trojan Attacks: Threat Analysis and countermeasures"
[3] K. Xiao, X. Zhang, and M. Tehranipoor. "A Clock Sweeping Technique for Detecting Hardware Trojans Impacting Circuits Delay"
[4] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey"
[5] A. Wood and J. A. Stankovic, "Denial of service in sensor networks"
[6] C. Jaikaeo, C. Srisathapornphat, and C.-C. Shen, "Diagnosis of sensor networks"

Outline

- Hardware Trojan in IoT
- Proposed Trojan detection scheme
- Simulation results
- Summary

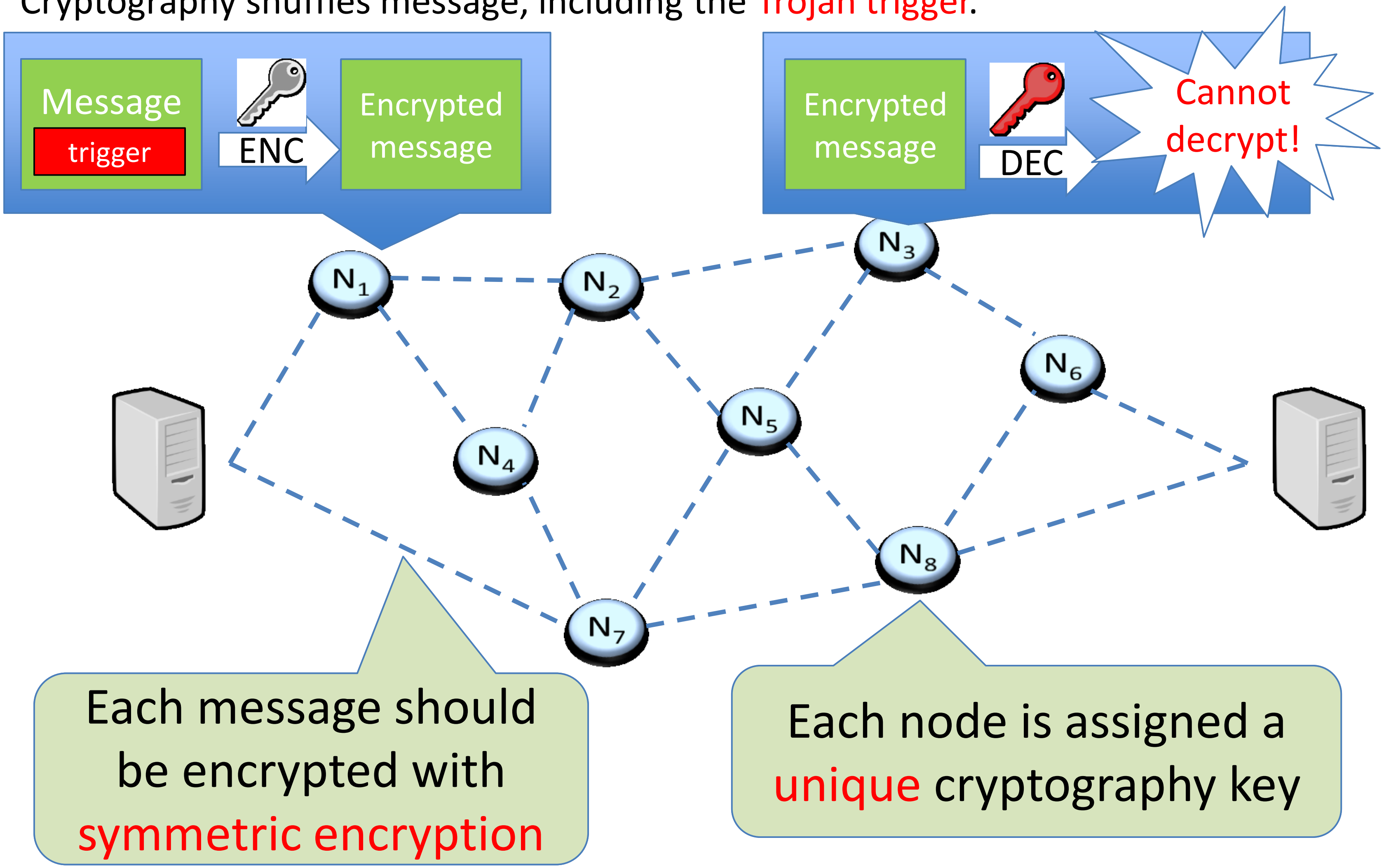
Our goal: prevent hardware Trojan in IoT from mutually triggering

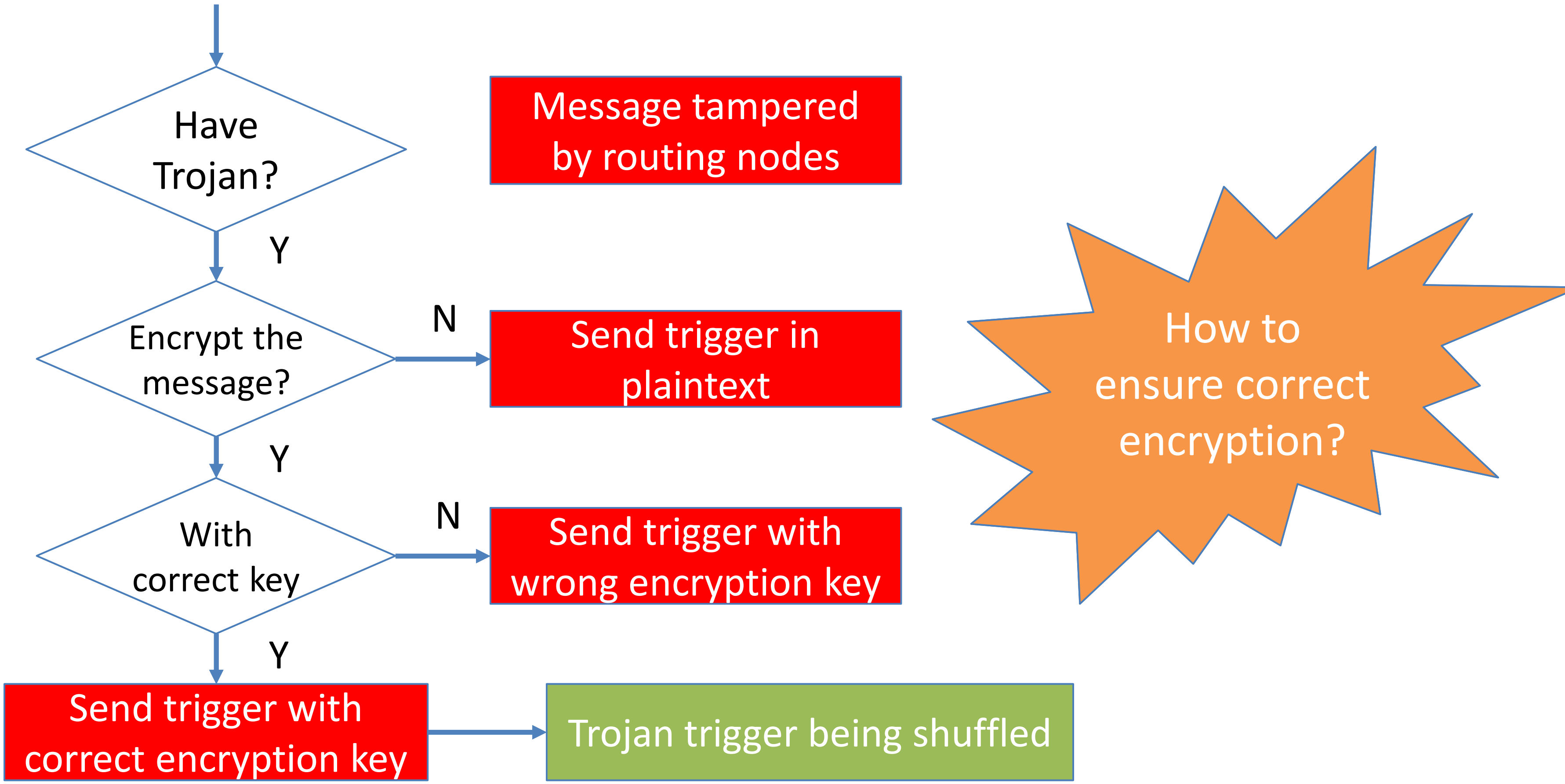
Our method:



Encryption to shuffle Trojan trigger

Cryptography shuffles message, including the Trojan trigger.





HOWEVER, encryption by itself cannot fully solve the problem!

Let’s introduce
Mutual auditing

Node mutual auditing

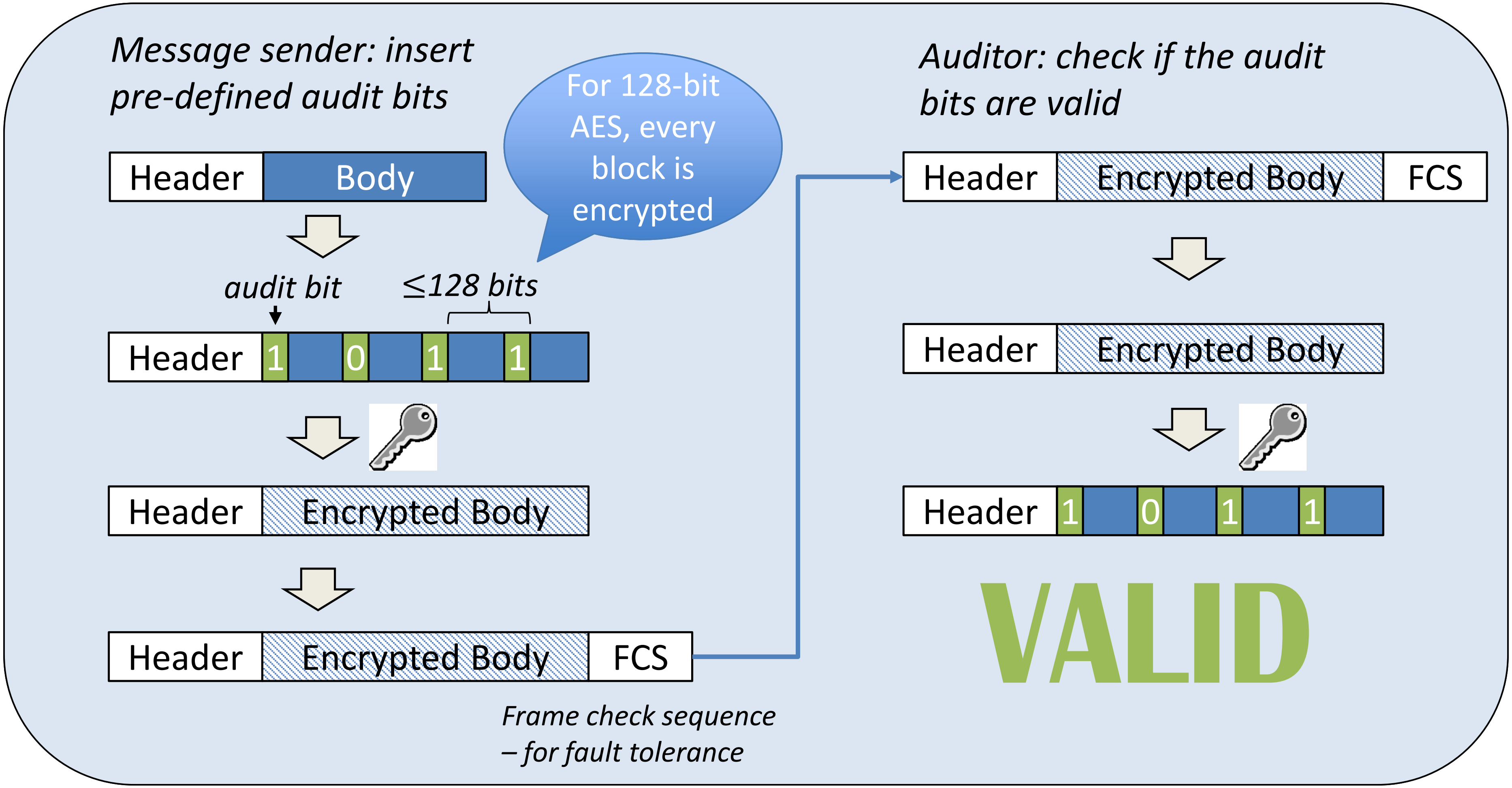
First-hop auditing: each node is audited by its neighbor nodes

How to perform audit?

Echo auditing: each auditor node is also audited by the node before

Node mutual auditing

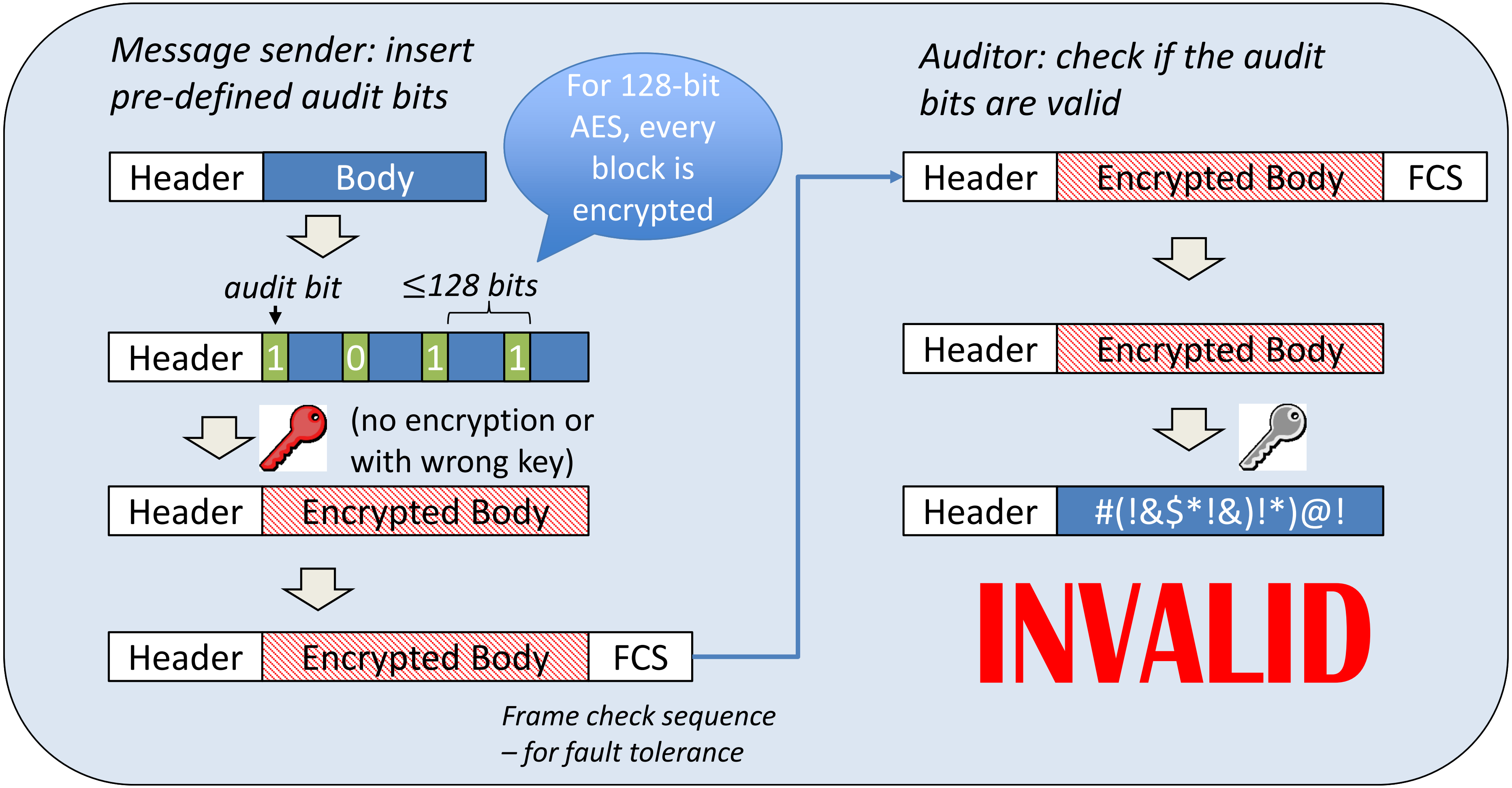
First-hop auditing: each node is audited by its neighbor nodes



Echo auditing: each auditor node is also audited by the message sender

Node mutual auditing

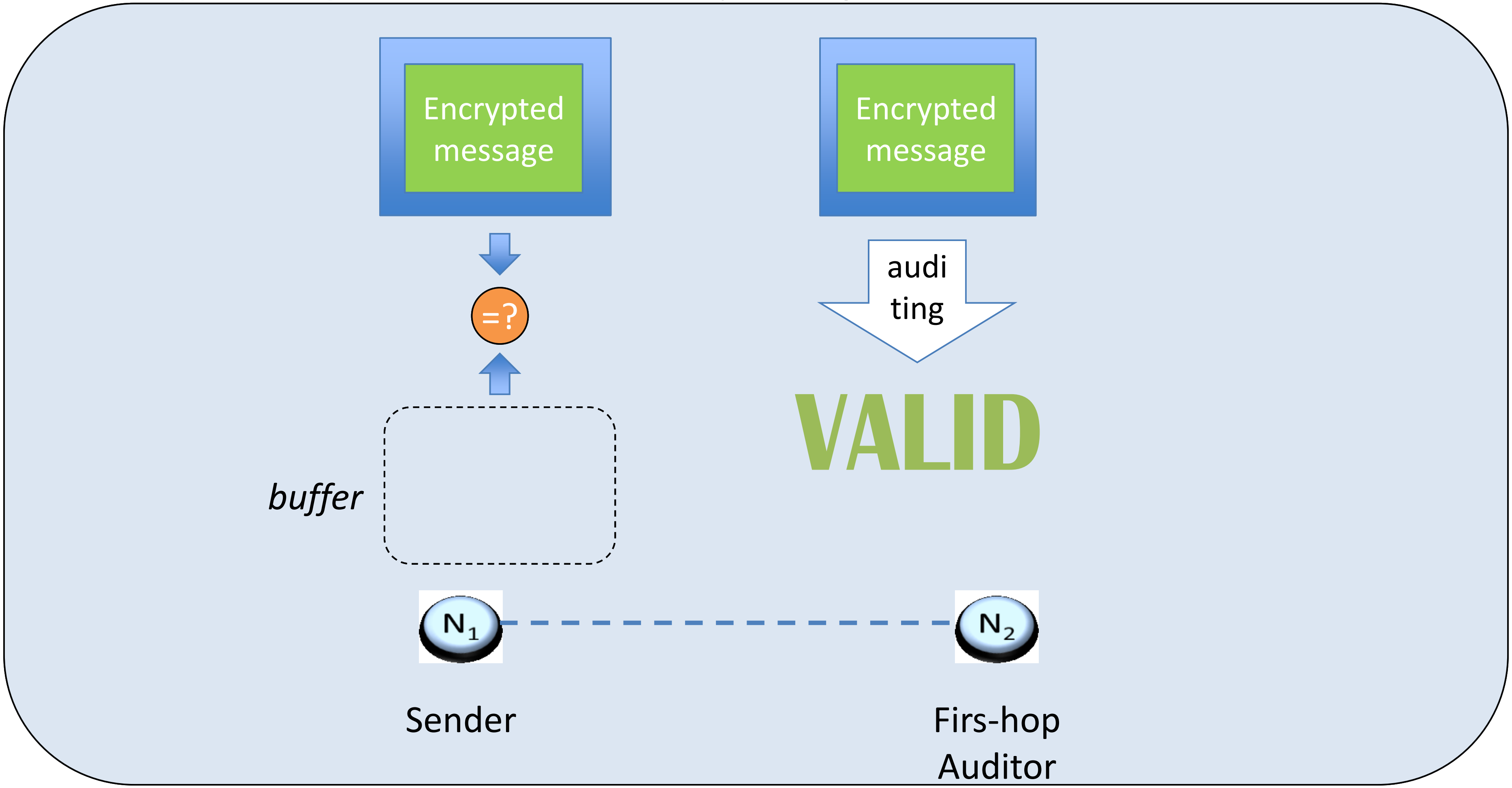
First-hop auditing: each node is audited by its neighbor nodes



Echo auditing: each auditor node is also audited by the message sender

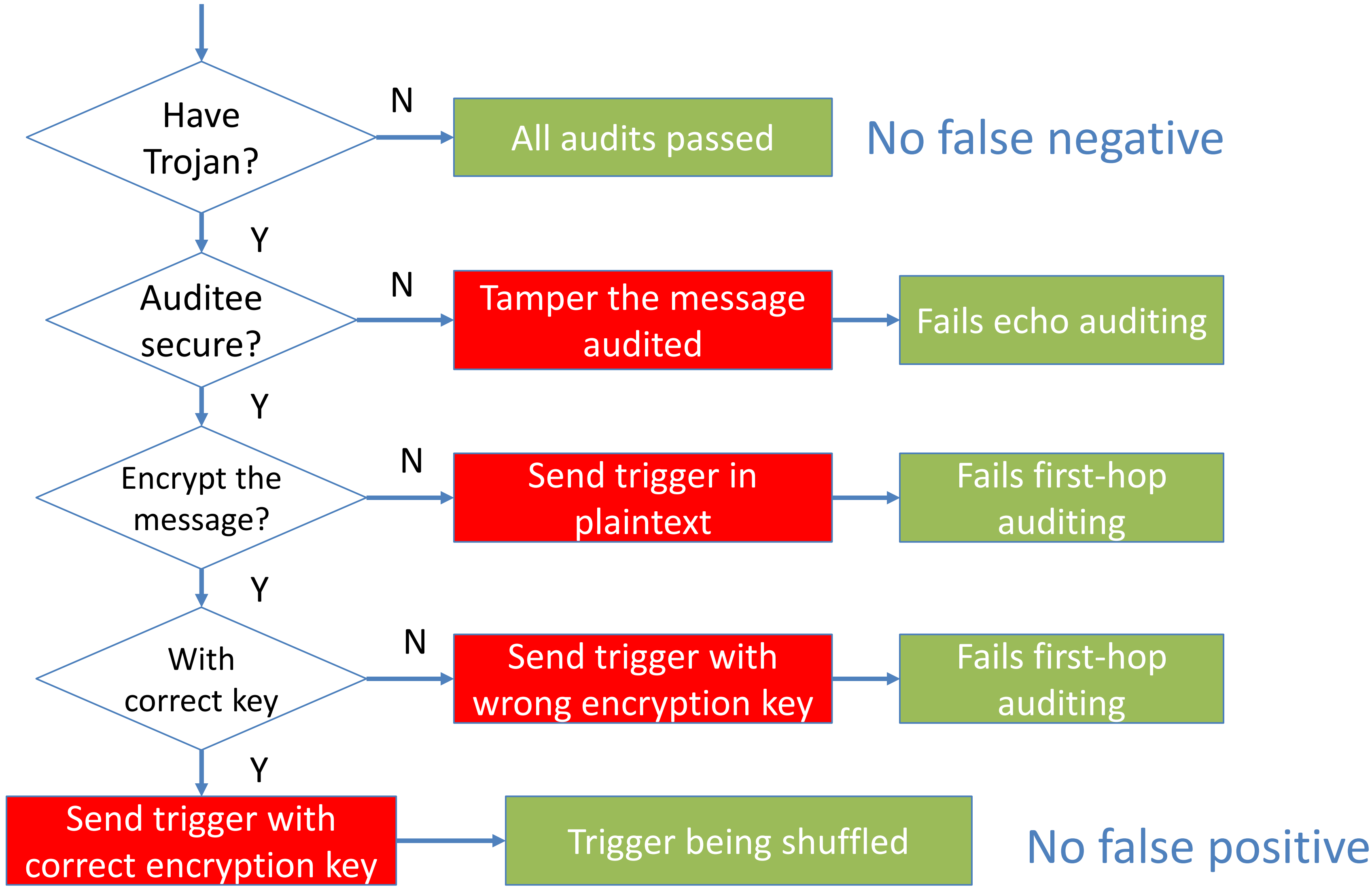
Node mutual auditing

First-hop auditing: each node is audited by its neighbor nodes



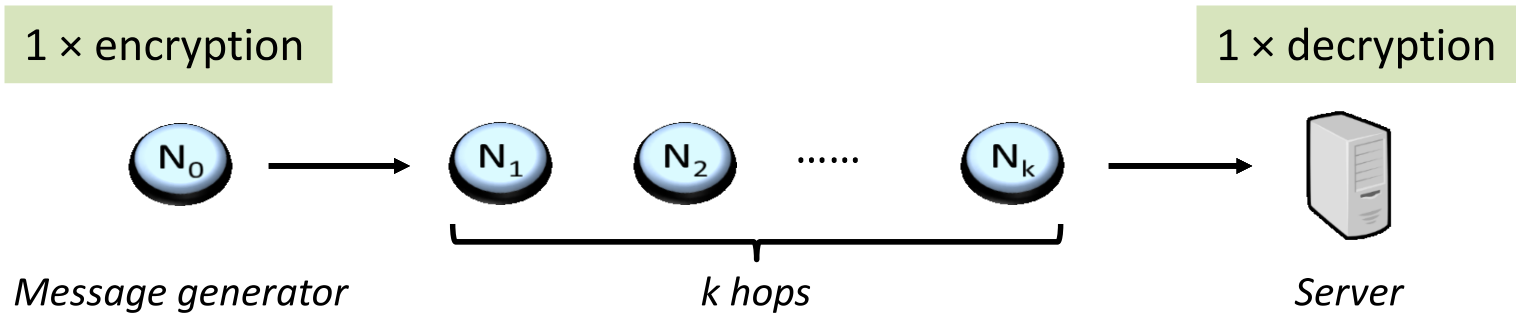
Echo auditing: each auditor node is also audited by the message sender

Security analysis for a node

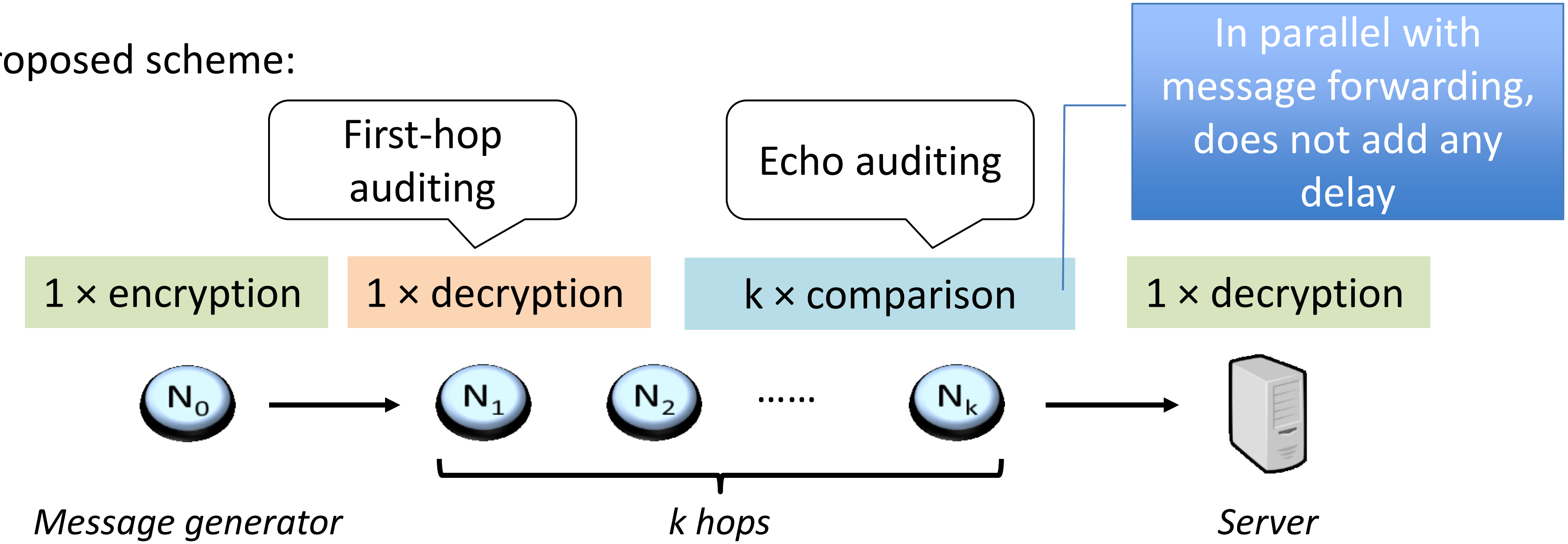


Overhead analysis

Regular IoT with message encryption:

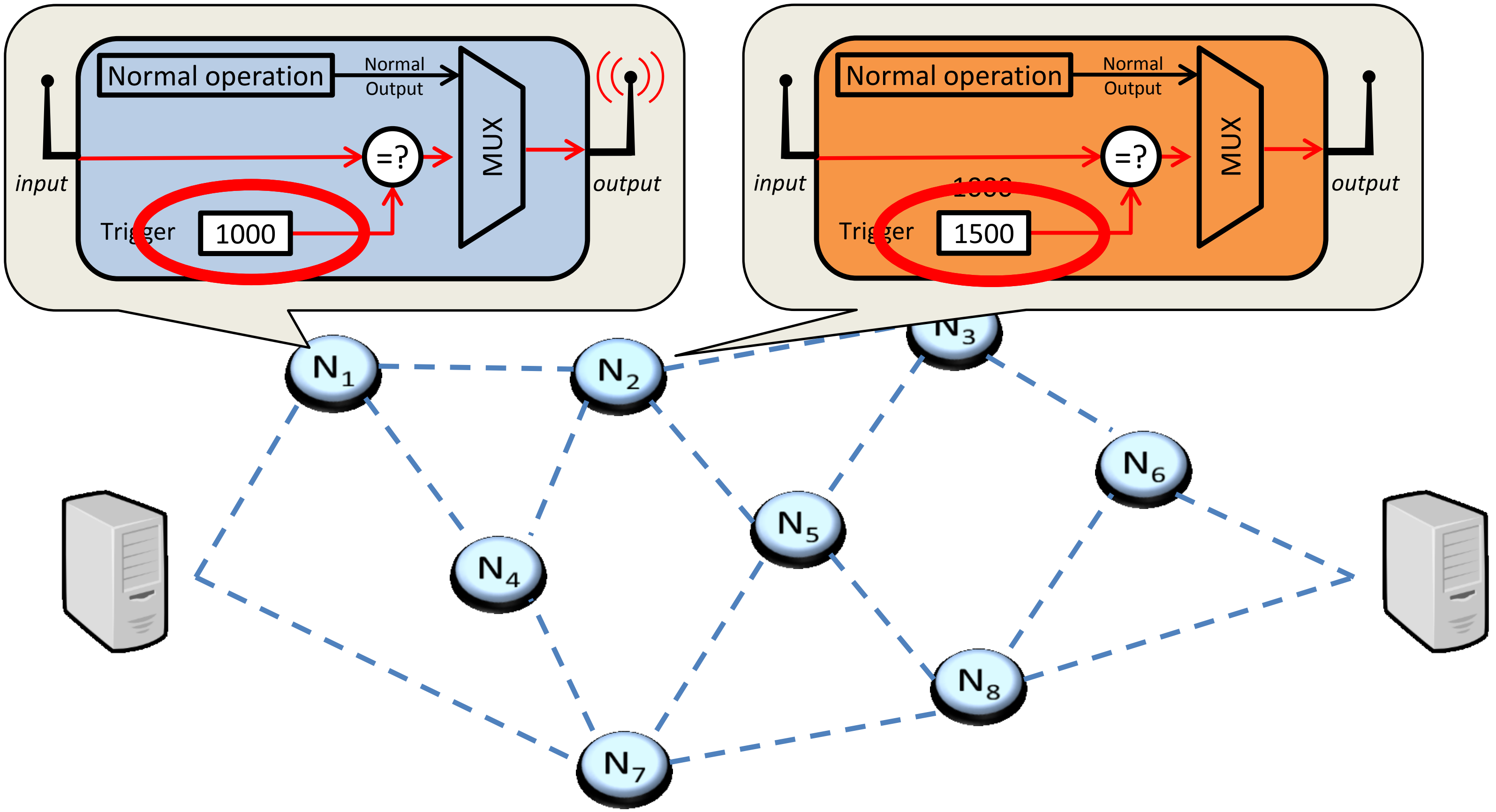


Proposed scheme:



How to prevent auditor and auditee from collusion?

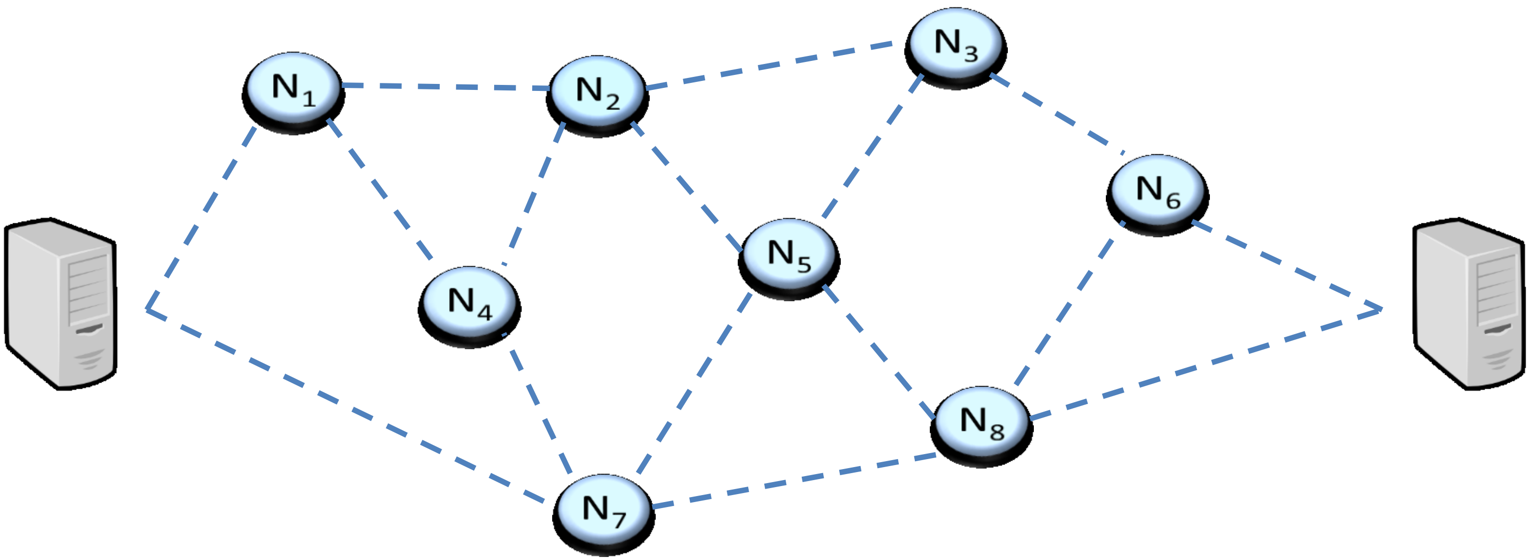
Node vendor diversity



Different vendors → Different triggers → Unable to mutually trigger

Node vendor diversity – how many vendors?

One vendor per node = 100% secure = huge overhead



Different vendors → Different triggers → Unable to mutually trigger

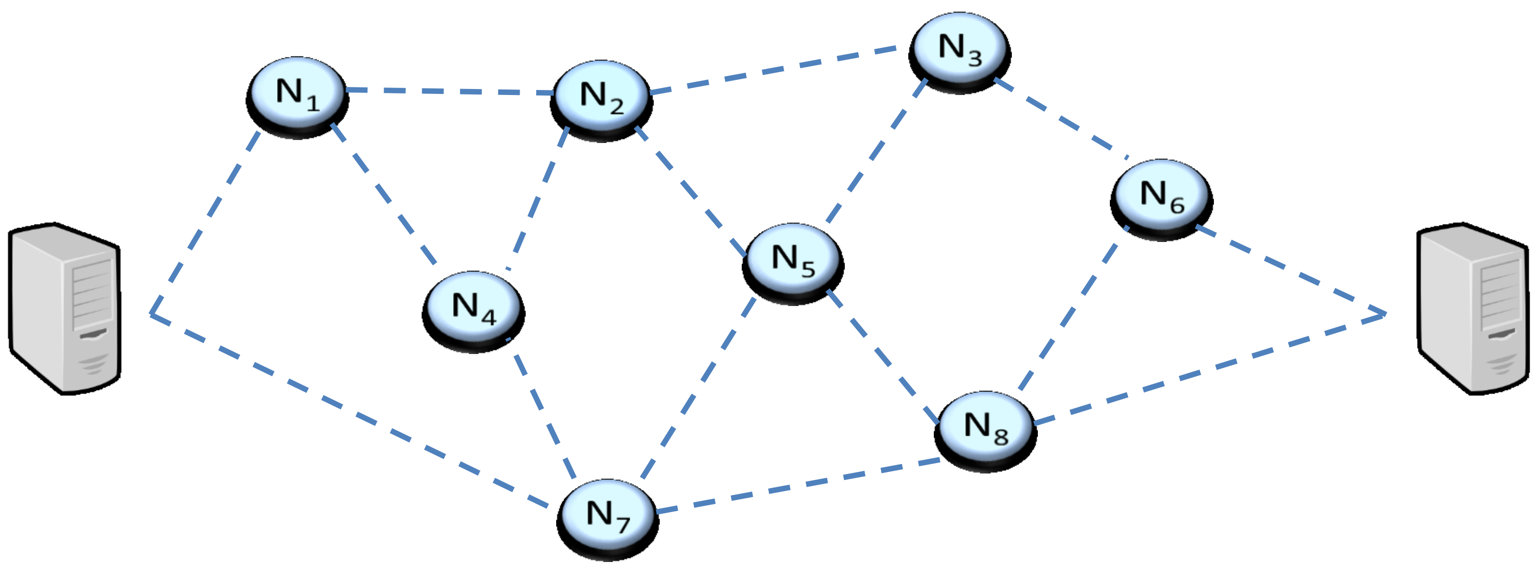
Node vendor diversity – how many vendors?

Color of **auditee** ≠ Color of **auditor** → *Secure*

Determin

8 vendors → 3 vendors

coloring



Node routing map → Graph coloring algorithm → Node vendor selection

Outline

- Hardware Trojan in IoT
- Proposed Trojan detection scheme
  - Message encryption
  - Mutual auditing
  - Vendor diversity
- Simulation results
- Summary

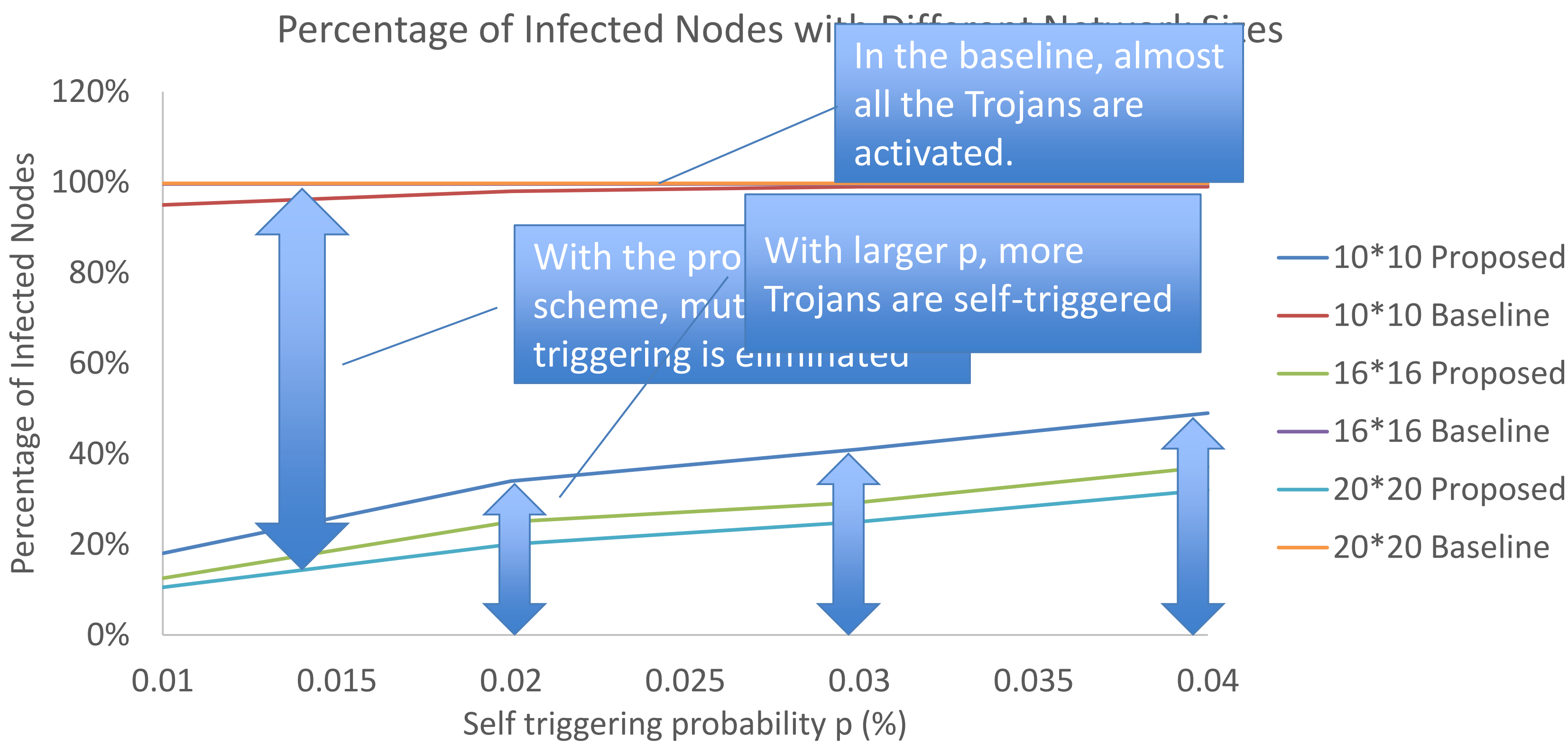
Methodology

Parameters		Values
Simulation tool		NS-2
Network scale	network size	10 × 10 to 20 × 20
	max bandwidth	100 MB/s
	expected traffic	40 to 100 packets/s
Network parameters	packet size	200 B body + 78 B metadata
	packet processing time	1 ms per hop
	cryptography overhead	1 ms per 128 bits

Security study by simulating Trojan activation

A hibernated Trojan can be either:

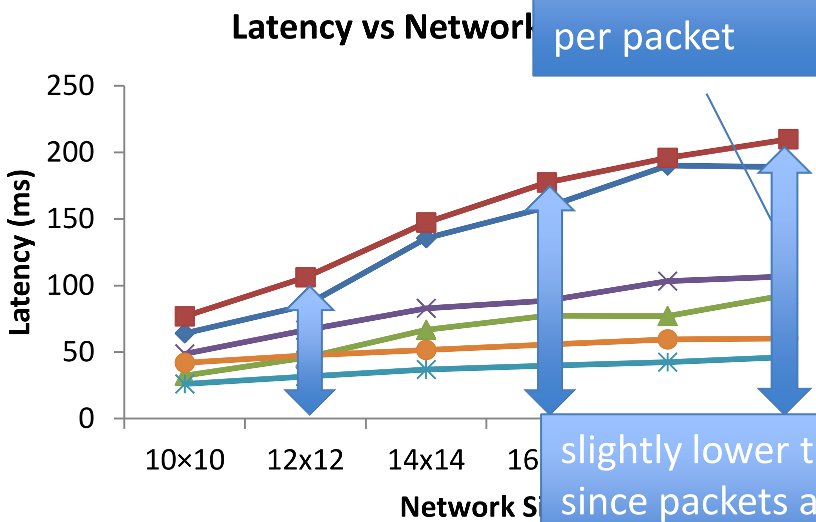
- **Self triggered** with a probability of *p* per packet
- **Mutually triggered** by successfully receiving and decoding triggering message sent by active Trojan from the same vendor

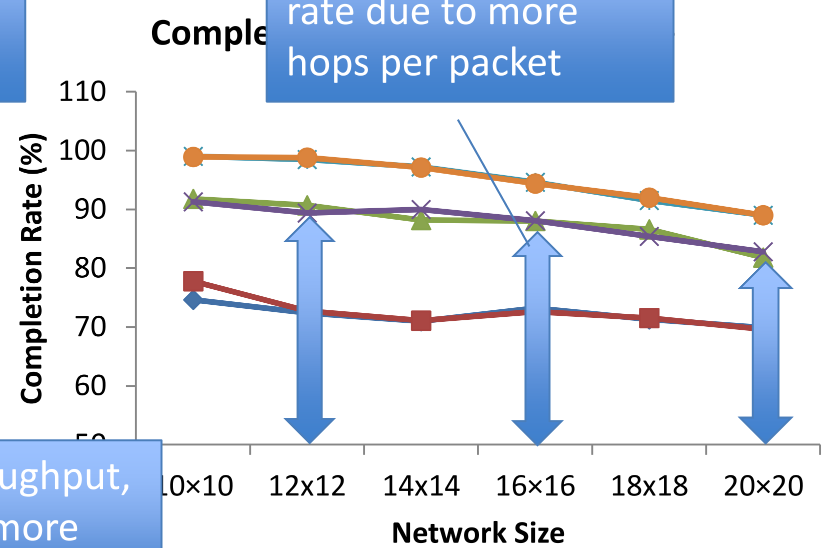


Performance ev

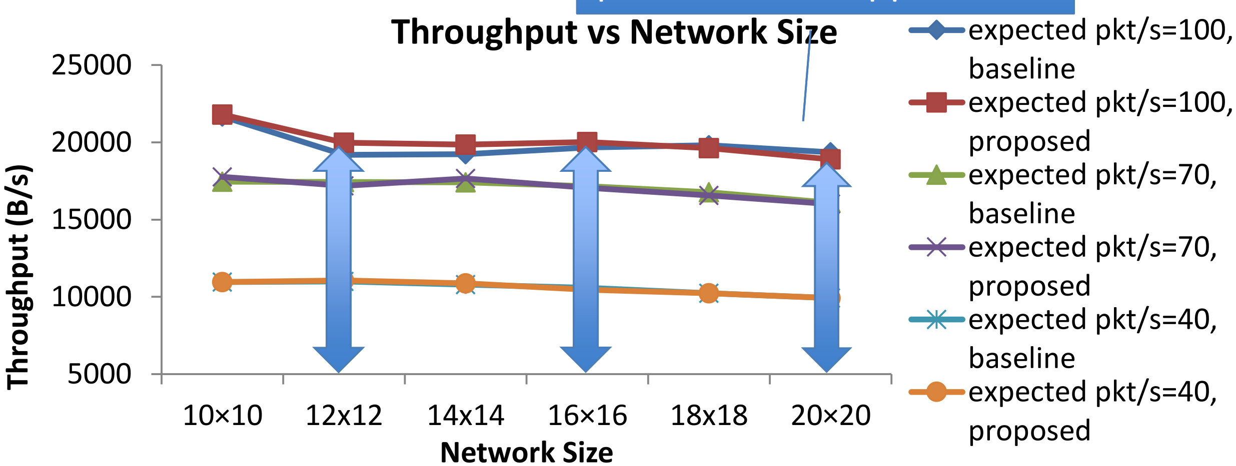
latency increases due to more hops per packet

lower completion rate due to more hops per packet





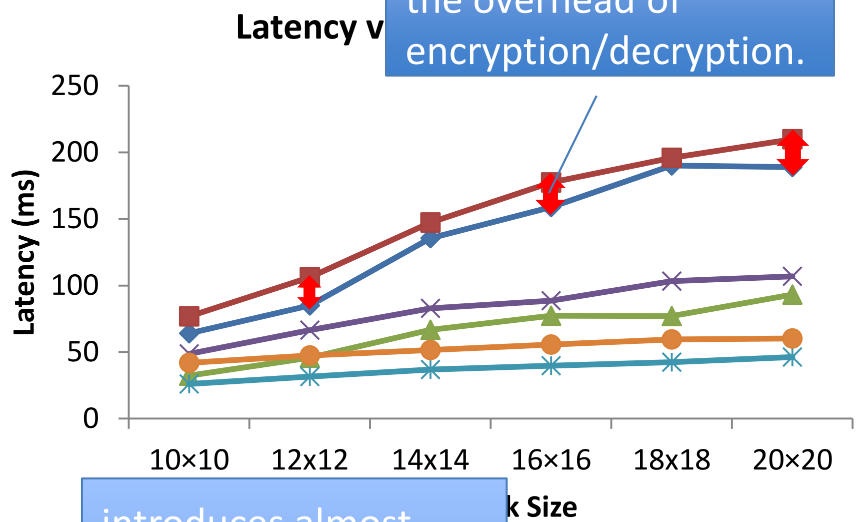
slightly lower throughput, since packets are more prone to be dropped



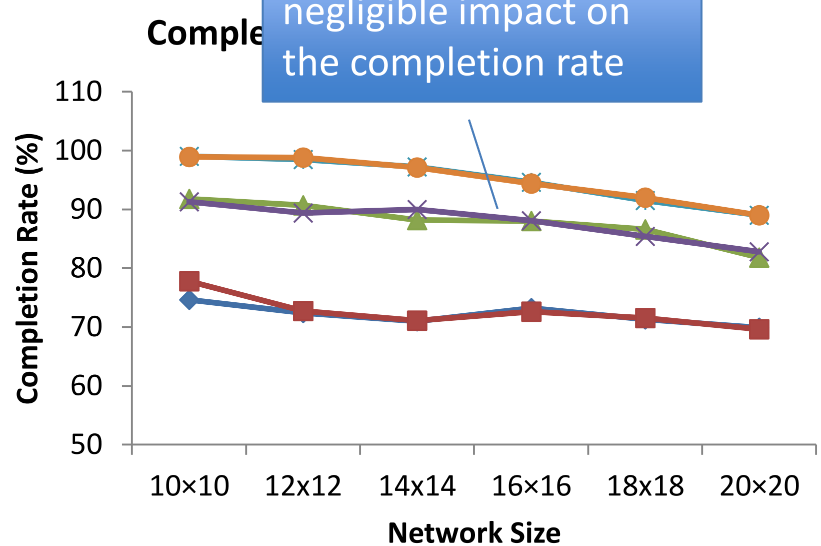
With larger network size..

Performan

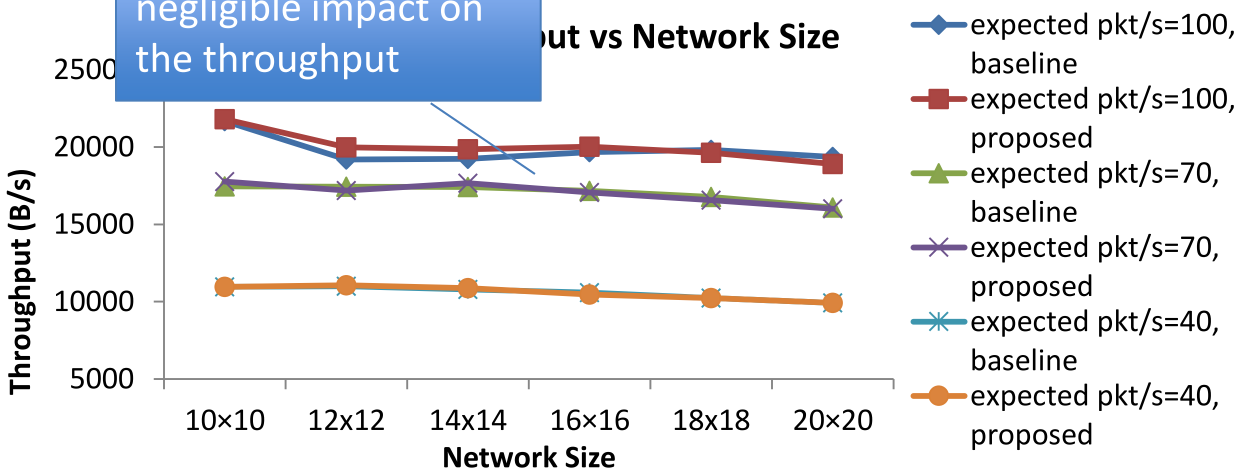
introduces constant latency (~25ms), due to the overhead of encryption/decryption.



introduces almost negligible impact on the completion rate



introduces almost negligible impact on the throughput



With the proposed scheme…

Outline

- Hardware Trojan in IoT
- Proposed Trojan detection scheme
- Simulation results
- Summary

Summary

- Problem:
  - Hardware Trojans are malicious and covert changes to the circuits which are difficult to detect during testing.
  - In IoT, hardware Trojans in different nodes may mutually trigger each other to cause catastrophe.
- Proposed framework:
  - Goal: prevent hardware Trojans in IoT from mutually triggering.
  - Method combines:
    - message encryption
    - node mutual auditing
    - node vendor diversity
- Simulation results show that the proposed scheme:
  - Prevents hardware Trojans from mutually triggering each other.
  - Introduces a constant (~25ms) latency to each packet regardless of the network size and traffic volume.