SecureIoD:A Secure Data Collection and Storage Mechanism for Internet of Drones

Cong Pu and Andrew Wall
Dept. of Computer Sciences and Electrical Engineering
Marshall University
Huntington, WV, United States

Imtiaz Ahmed
Dept. of Electrical Engineering and Computer Science
Howard University
Washington, DC, United States

Kim-Kwang Raymond Choo
Dept. of Information Systems and Cyber Security
University of Texas at San Antonio
San Antonio, TX, United States

The 23rd IEEE International Conference on Mobile Data Management (IEEE MDM), June 6 – 9, 2022

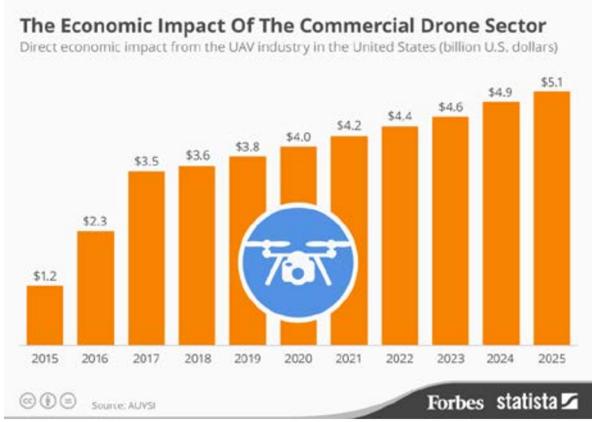
Outline

- Introduction & Research Motivation
- Preliminary Background
  - Physical Unclonable Function and Henon Map
- Secure Data Collection and Storage Mechanism (*SecureIoD*)
  - System and Adversary Models
  - Mutual Authentication and Key Establishment
  - Miner ZSP Selection and Block Generation
- Security Verification and Analysis & Performance Evaluation
- Concluding Remarks

Introduction

- Drones have attracted considerable attention for various applications
  - disaster/emergency response
  - infrastructure inspection
  - smart cities

- “Economic Impact of Drones” (Statista)
  - the commercial drone market is to be valued at USD 5 billion by 2025
- Future opportunities in the emerging technology field of drones are limitless



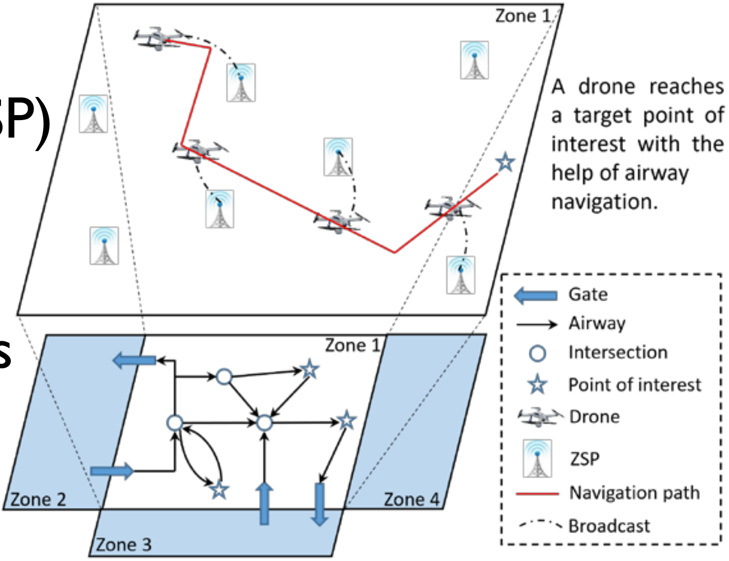
The Economic Impact Of The Commercial Drone Sector

Direct economic impact from the UAV industry in the United States (billion U.S. dollars)

Source: AUVSI

Introduction

- To fully exploit drones, Internet of Drones (IoD) is proposed
  - mobile drones
  - stationary Zone Service Provider (ZSP)
    - acts as access point
  - airspace is partitioned into zones
  - adjacent zones are reachable via gates
  - zone is administrated by ZSP(s)



A drone reaches a target point of interest with the help of airway navigation.

- In a variety of IoD applications
  - massive volume of highly critical data are collected and transmitted over open network
    → data security and privacy challenges
  - drones are resource-constrained and considered to be defenseless to security attacks
    → suffering from security attacks

} **secure data collection and storage framework**

Research Motivation

- Drones and ZSPs communicate over *insecure wireless channel*
  - mutually authenticate each other before sharing critical info.
    - traditional cryptographic mechanisms? comput. and comm. overhead
    - lightweight security and cryptographic protocol
- An adversary might *capture a drone and extract credentials*
  - drones should have tamper-resistant module to safeguard info.
    - defend against both software based and physical memory disclosure attacks
- The centralized server / approach has significant *weaknesses*
  - *decentralized data storage mechanism*
    - guarantee quality-of-service (QoS) requirements
    - reduce administrative costs
    - eliminate single point of failure

Our Contribution

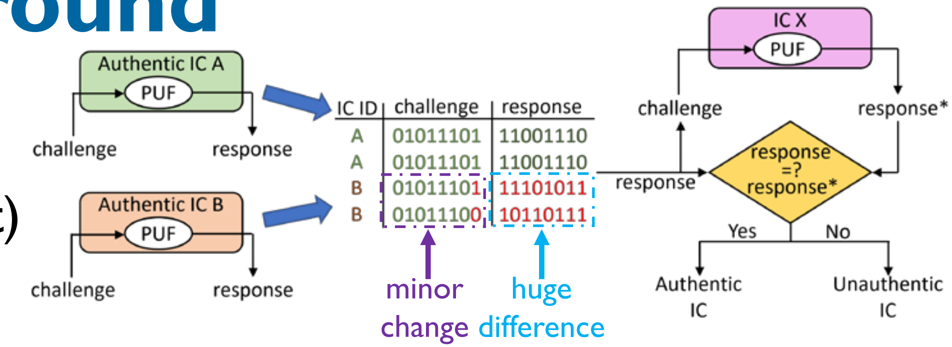
- This paper
  - proposes a secure data collection and storage mechanism (***SecureIoD***) for the IoD environment.
    - Drones and ZSPs first authenticate each other and establish a secure session key based on physical unclonable function and Henon map.
    - ZSPs pack the collected data into blocks and compete to add their blocks into the blockchain.
      - a joint Proof-of-Work (PoW) and Proof-of-Stake (PoS) consensus mechanism is proposed to select the miner ZSP.
        - the more transactions are in the block, the easier a ZSP can solve the cryptographic puzzle.
  - conducts security verification using *AVISPA* and *Scyther*
  - develops a real-world testbed for performance evaluation

Conclusion:

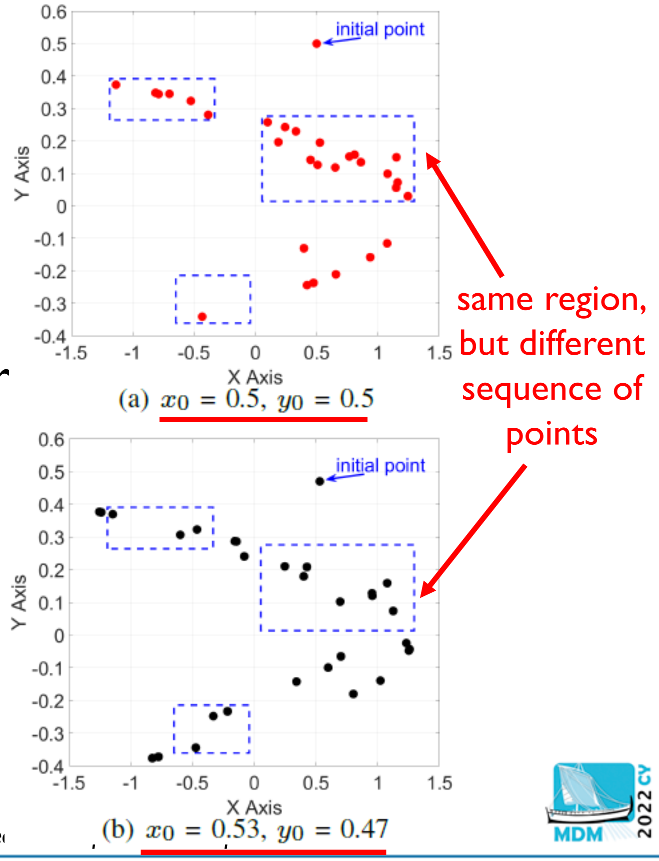
SecureIoD: better performance; viable and competitive approach for ensuring secure data collection and storage in the IoD environment.

Preliminary Background

- Physical Unclonable Function (PUF)
  - similar to biometrics (i.e., fingerprint)
  - designed based on unique physical characteristics
  - taking an input (‘challenge’), and producing an unique output (‘response’)
    - challenge-response pair (CRP)
- Chaotic System
  - deterministic system exhibiting nonlinear behavior
  - Henon map $\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n \end{cases}$
    - two-dimensional dynamical system
    - displaying chaos with certain parameters and initial conditions
    - without the same initial conditions, the same chaos cannot be reproduced



IC ID	challenge	response
A	01011101	11001110
A	01011101	11001110
B	01011101	11101011
B	01011100	10110111

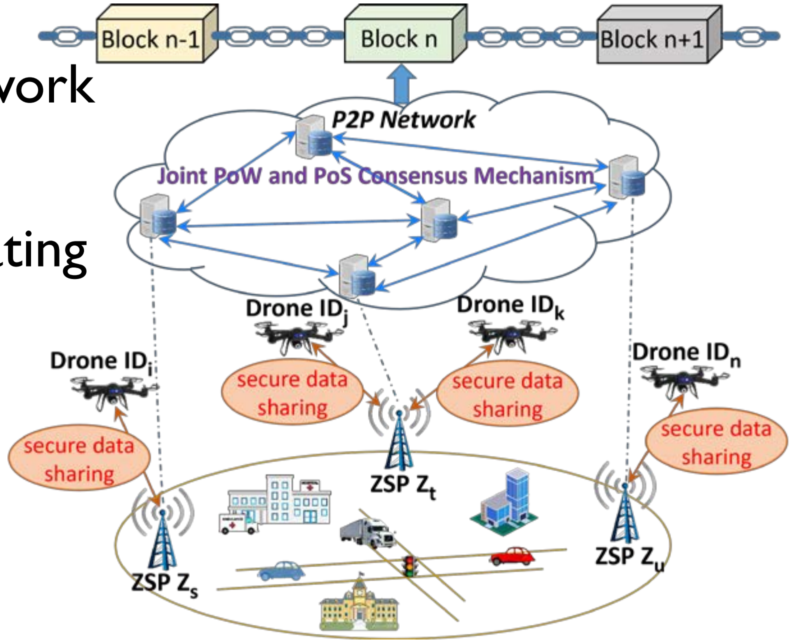


(a) $x_0 = 0.5,\ y_0 = 0.5$

(b) $x_0 = 0.53,\ y_0 = 0.47$

System and Adversary Models

- System model
  - two comm. entities: drones and ZSPs
  - drones are equipped with sensors, communication devices, and PUF enabled integrated circuit
  - ZSPs form a peer-to-peer (P2P) network



- Adversary model
  - any two entities who are communicating over an insecure wireless channel are untrustworthy
- Two tasks (P2P netw.):
  i. generate a block: collect, validate, and pack data into a block.
  ii. add the block in the blockchain: compete to add block in blockchain using the consensus mechanism.

SecureIoD:
Secure Data Collection and Storage Mechanism

- The basic idea of *SecureIoD*:
  1. Drones and ZSPs first mutually authenticate each other and establish a secure session key based on physical unclonable function and Henon map before sharing any sensitive data via an insecure wireless channel.
  2. ZSPs pack the collected data into blocks and compete to add their blocks into the blockchain based on the proposed joint Proof-of-Work (PoW) and Proof-of-Stake (PoS) consensus mechanism.

- *SecureIoD* is composed of two parts:
  - i. Mutual Authentication and Key Establishment

    ↳ secure data communication
  - ii. Miner ZSP Selection and Block Generation

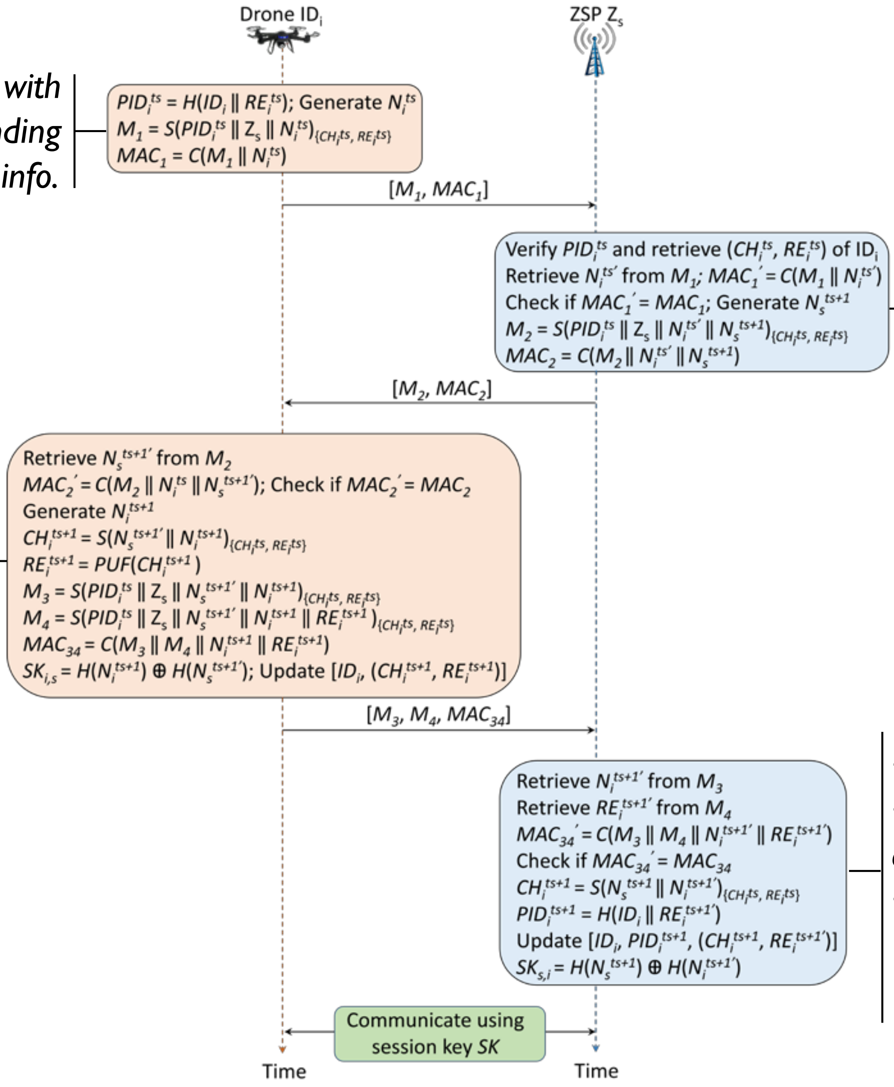
    ↳ secure data storage

}

secure data collection and storage framework

SecureIoD:
Secure Data Collection and Storage Mechanism

- *drone ID_i initiate comm. with ground station through sending encrypted identity (pseudonym) info.*



$PID_i^{ts} = H(ID_i \| RE_i^{ts})$; Generate N_i^{ts}
$M_1 = S(PID_i^{ts} \| Z_s \| N_i^{ts})_{\{CH_i ts,\, RE_i ts\}}$
$MAC_1 = C(M_1 \| N_i^{ts})$

$[M_1, MAC_1]$

Verify PID_i^{ts} and retrieve (CH_i^{ts}, RE_i^{ts}) of ID_i
Retrieve $N_i^{ts'}$ from M_1; $MAC_1' = C(M_1 \| N_i^{ts'})$
Check if $MAC_1' = MAC_1$; Generate N_s^{ts+1}
$M_2 = S(PID_i^{ts} \| Z_s \| N_i^{ts'} \| N_s^{ts+1})_{\{CH_i ts,\, RE_i ts\}}$
$MAC_2 = C(M_2 \| N_i^{ts'} \| N_s^{ts+1})$

- *verify identity info. of drone ID_i*
- *check message integrity*
- *send a random number to drone ID_i*
  - *used for session key*

$[M_2, MAC_2]$

Retrieve $N_s^{ts+1'}$ from M_2
$MAC_2' = C(M_2 \| N_i^{ts} \| N_s^{ts+1'})$; Check if $MAC_2' = MAC_2$
Generate N_i^{ts+1}
$CH_i^{ts+1} = S(N_s^{ts+1'} \| N_i^{ts+1})_{\{CH_i ts,\, RE_i ts\}}$
$RE_i^{ts+1} = PUF(CH_i^{ts+1})$
$M_3 = S(PID_i^{ts} \| Z_s \| N_s^{ts+1'} \| N_i^{ts+1})_{\{CH_i ts,\, RE_i ts\}}$
$M_4 = S(PID_i^{ts} \| Z_s \| N_s^{ts+1'} \| N_i^{ts+1} \| RE_i^{ts+1})_{\{CH_i ts,\, RE_i ts\}}$
$MAC_{34} = C(M_3 \| M_4 \| N_i^{ts+1} \| RE_i^{ts+1})$
$SK_{i,s} = H(N_i^{ts+1}) \oplus H(N_s^{ts+1'})$; Update $[ID_i, (CH_i^{ts+1}, RE_i^{ts+1})]$

- *check message integrity*
- *generate a random number*
- *calculate a new CRP and pseudonym*
- *send a random number and new CRP to ground station*
- *calculate session key*
  - *using random numbers from itself and ground station*

$[M_3, M_4, MAC_{34}]$

Retrieve $N_i^{ts+1'}$ from M_3
Retrieve $RE_i^{ts+1'}$ from M_4
$MAC_{34}' = C(M_3 \| M_4 \| N_i^{ts+1'} \| RE_i^{ts+1'})$
Check if $MAC_{34}' = MAC_{34}$
$CH_i^{ts+1} = S(N_s^{ts+1} \| N_i^{ts+1'})_{\{CH_i ts,\, RE_i ts\}}$
$PID_i^{ts+1} = H(ID_i \| RE_i^{ts+1'})$
Update $[ID_i, PID_i^{ts+1}, (CH_i^{ts+1}, RE_i^{ts+1'})]$
$SK_{s,i} = H(N_s^{ts+1}) \oplus H(N_i^{ts+1'})$

- *check message integrity*
- *calculate and update pseudonym and CRP of drone ID_i*
- *calculate session key*
  - *using random numbers from itself and drone ID_i*

Communicate using session key *SK*

SecureIoD:
Miner ZSP Selection and Block Generation

- After collecting data from drones, ZSPs put data into blocks and try to add them into the blockchain.
  - find a hash value satisfying the following target criterion

$$H(ZSP_{ID}, ts, prevHash, nonce) \geq Hash_{ID}^{th},$$

- ZSP_{ID}: ZSP ID
- *ts*: current timestamp
- *prevHash*: previous block's hash value
- *nonce*: calculating the block's hash value
- $Hash_{ID}^{th}$: hash threshold of ZSP ZSP_{ID}
  - control the difficulty level of cryptographic puzzle / block generation speed

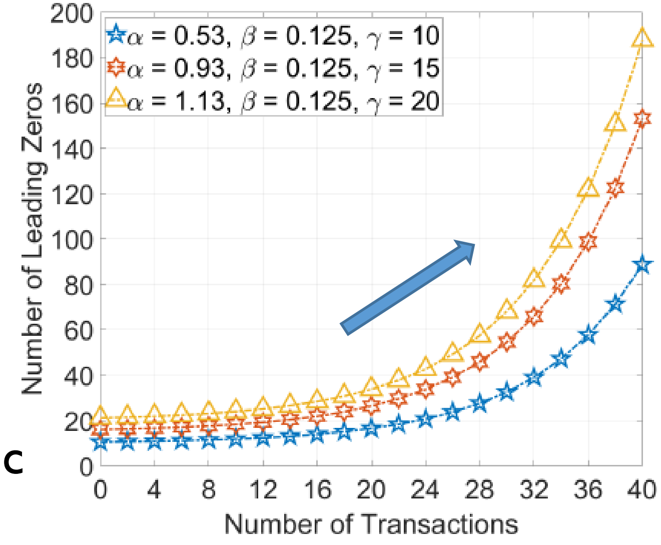
$$Hash_{ID}^{th} = concat(zeros(N_{stake}), Tgt^{th}),$$

the number of leading zeros in $Hash_{ID}^{th}$ $\longrightarrow$

$$\underline{N_{stake}} = [\gamma + \alpha \cdot e^{N_{trans} \cdot \beta}],$$

$$\underline{Tgt^{th}} = rand(2^{N_{hash} - N_{stake}} - 1),$$

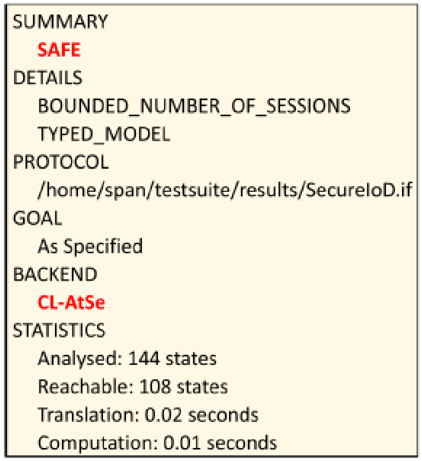
$\longleftarrow$ the random numbers following N_{stake}



Change of the number of leading zeros (N_{stake}) against the number of transactions (N_{trans}) in the block.

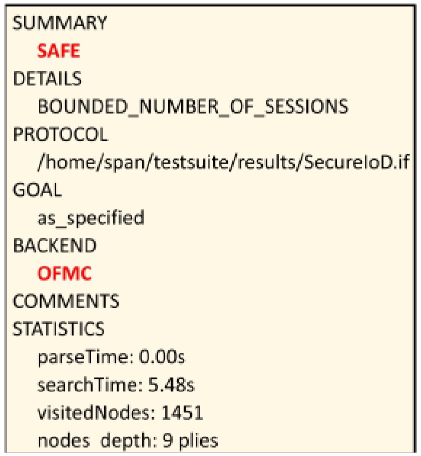
Security Verifications / Analysis

Security Verification Using *AVISPA*



```
SUMMARY
    SAFE
DETAILS
    BOUNDED_NUMBER_OF_SESSIONS
    TYPED_MODEL
PROTOCOL
    /home/span/testsuite/results/SecureIoD.if
GOAL
    As Specified
BACKEND
    CL-AtSe
STATISTICS
    Analysed: 144 states
    Reachable: 108 states
    Translation: 0.02 seconds
    Computation: 0.01 seconds
```

(a)



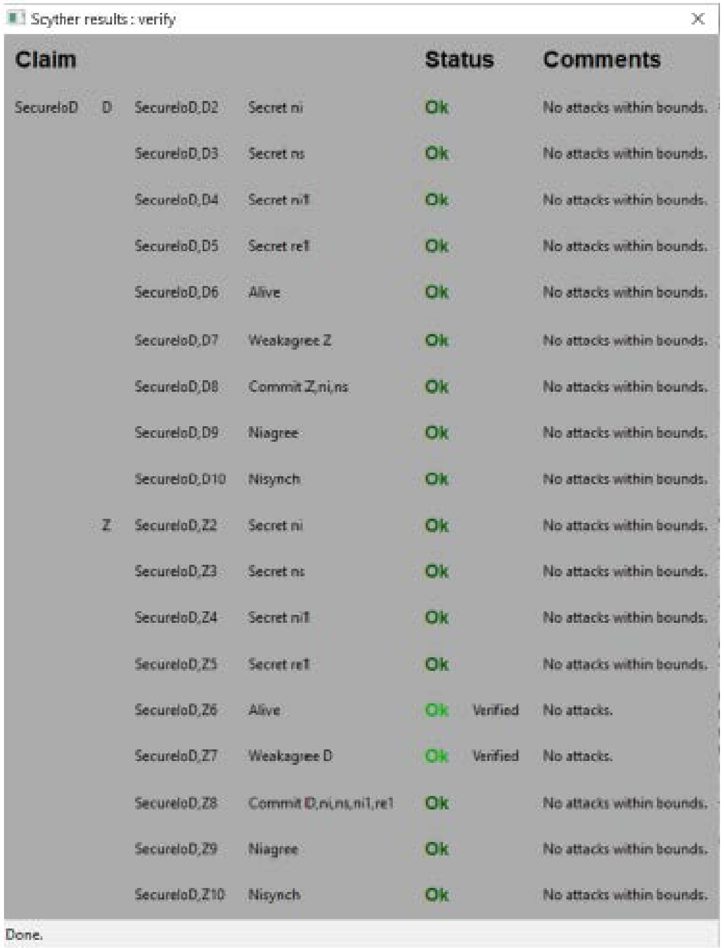
```
SUMMARY
    SAFE
DETAILS
    BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
    /home/span/testsuite/results/SecureIoD.if
GOAL
    as_specified
BACKEND
    OFMC
COMMENTS
STATISTICS
    parseTime: 0.00s
    searchTime: 5.48s
    visitedNodes: 1451
    nodes  depth: 9 plies
```

(b)

SecureIoD is secure against

- drone capture attack
- drone impersonation attack
- message modification attack
- ZSP spoofing attack

Security Verification Using *Scyther*



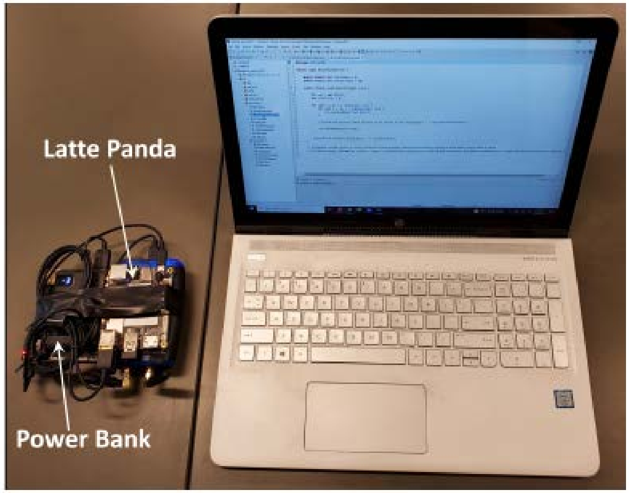
Scyther results : verify

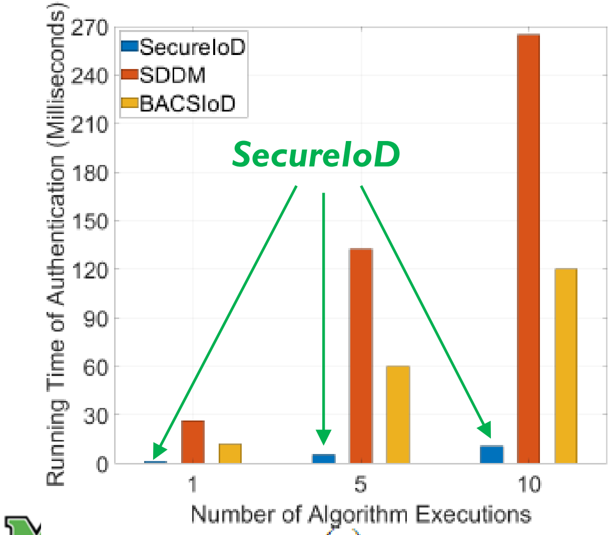
Claim				Status		Comments
SecureIoD	D	SecureIoD,D2	Secret ni	Ok		No attacks within bounds.
		SecureIoD,D3	Secret ns	Ok		No attacks within bounds.
		SecureIoD,D4	Secret ni1	Ok		No attacks within bounds.
		SecureIoD,D5	Secret re1	Ok		No attacks within bounds.
		SecureIoD,D6	Alive	Ok		No attacks within bounds.
		SecureIoD,D7	Weakagree Z	Ok		No attacks within bounds.
		SecureIoD,D8	Commit Z,ni,ns	Ok		No attacks within bounds.
		SecureIoD,D9	Niagree	Ok		No attacks within bounds.
		SecureIoD,D10	Nisynch	Ok		No attacks within bounds.
	Z	SecureIoD,Z2	Secret ni	Ok		No attacks within bounds.
		SecureIoD,Z3	Secret ns	Ok		No attacks within bounds.
		SecureIoD,Z4	Secret ni1	Ok		No attacks within bounds.
		SecureIoD,Z5	Secret re1	Ok		No attacks within bounds.
		SecureIoD,Z6	Alive	Ok	Verified	No attacks.
		SecureIoD,Z7	Weakagree D	Ok	Verified	No attacks.
		SecureIoD,Z8	Commit D,ni,ns,ni1,re1	Ok		No attacks within bounds.
		SecureIoD,Z9	Niagree	Ok		No attacks within bounds.
		SecureIoD,Z10	Nisynch	Ok		No attacks within bounds.

Done.

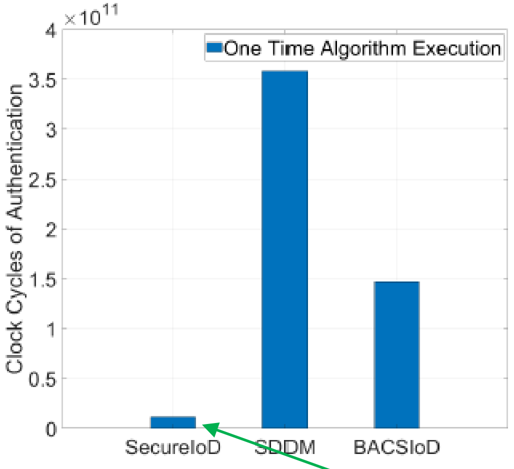
Experimental Evaluation

Real-world Testbed:

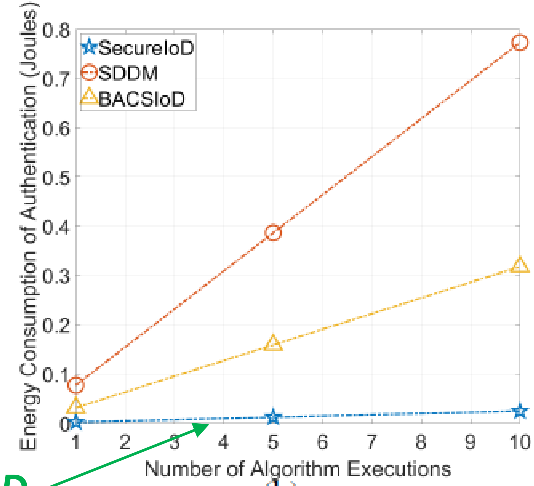




(a)

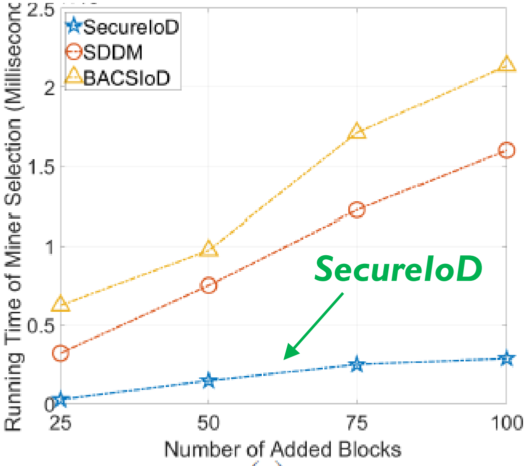


(a)

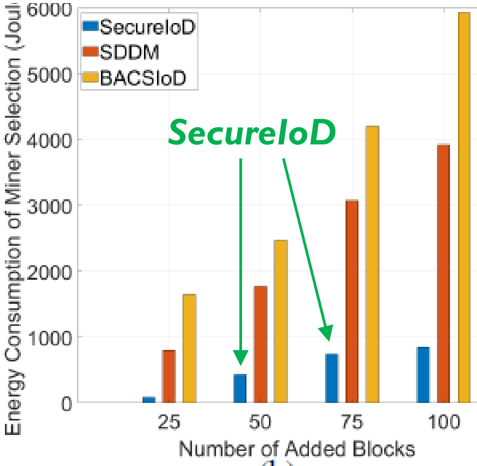


(b)

SecureIoD



(a)



(b)

COMPARISON OF COMMUNICATION COST

SecureIoD

Metrics	*SecureIoD*	SDDM	BACSIoD
Number of Messages	3	7	3
Energy Consumption (joule)	3.38×10^{-4}	7.88×10^{-4}	3.38×10^{-4}

Concluding Remarks

- Developed a secure data collection and storage mechanism (***SecureIoD***) in the IoD.
  - Drones and ZSPs first mutually authenticate each other and establish a secure session key before sharing any sensitive data via an insecure wireless channel.
  - ZSPs pack the collected data into blocks and compete to add their blocks into the blockchain based on the proposed joint Proof-of-Work (PoW) and Proof-of-Stake (PoS) consensus mechanism.
- We verified the security of *SecureIoD* through specific security protocol verification tools (i.e., *AVISPA* and *Scyther*) and security analysis.
  - *SecureIoD* is a secure protocol and immune to many cyber attacks
- We developed a real-world testbed and conducted experimental study.
  - *SecureIoD* provides better performance in terms of running time, CPU time, clock cycle, and energy consumption

Any Questions?

Email: cong.pu@ieee.org

SecureIoD source codes and its security verification programs are publicly available at the https://github.com/congpu/SecureIoD.