THE UNIVERSITY OF BRITISH COLUMBIA

SwarmFuzz: Discovering GPS Spoofing Attacks in Drone Swarms

Yingao (Elaine) Yao, Pritam Dash, Karthik Pattabiraman

ECE, The University of British Columbia, Vancouver, Canada

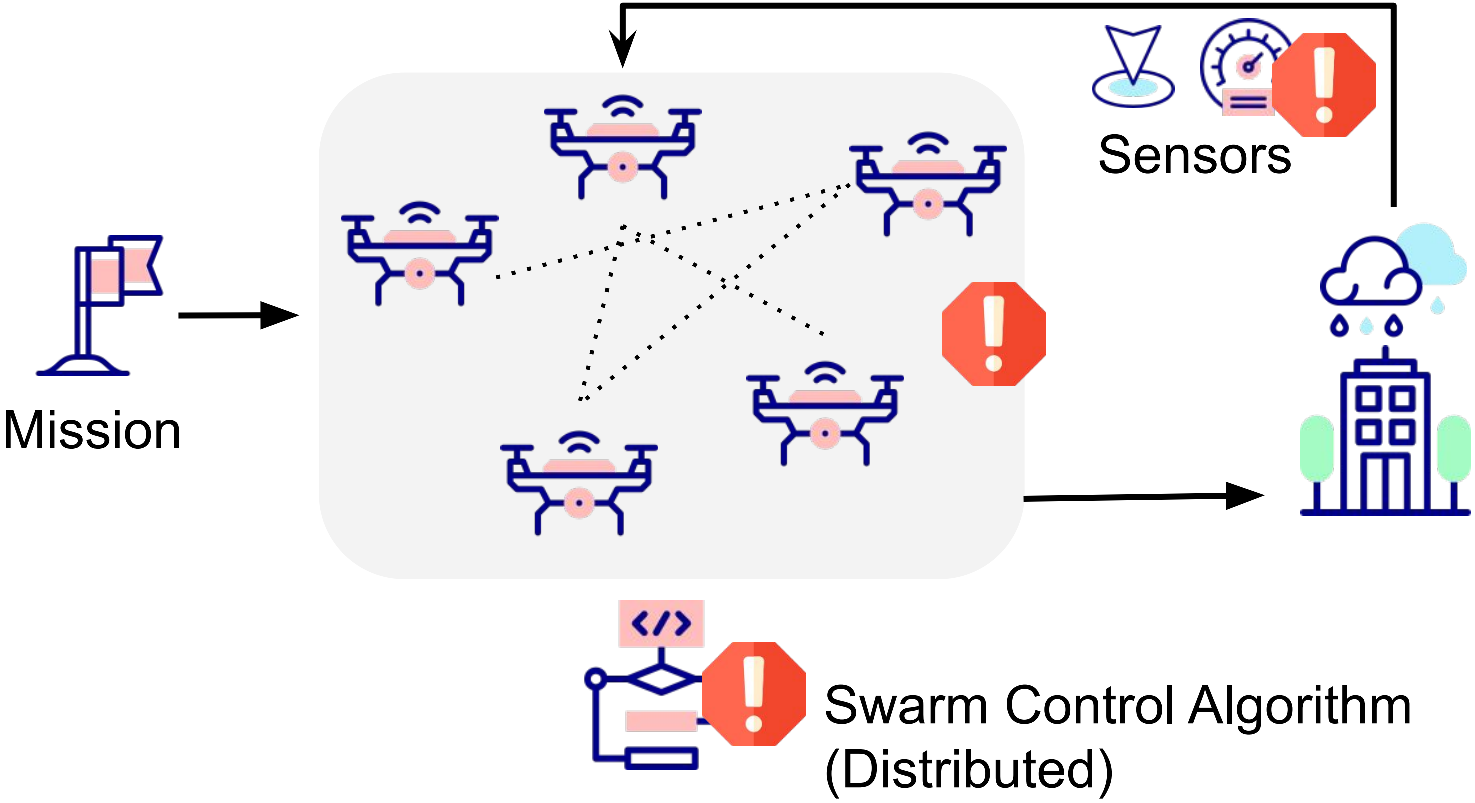
DSN 2023

Drone Swarms in Large-scale Missions

Agriculture

Search & Rescue

Drone Swarm System

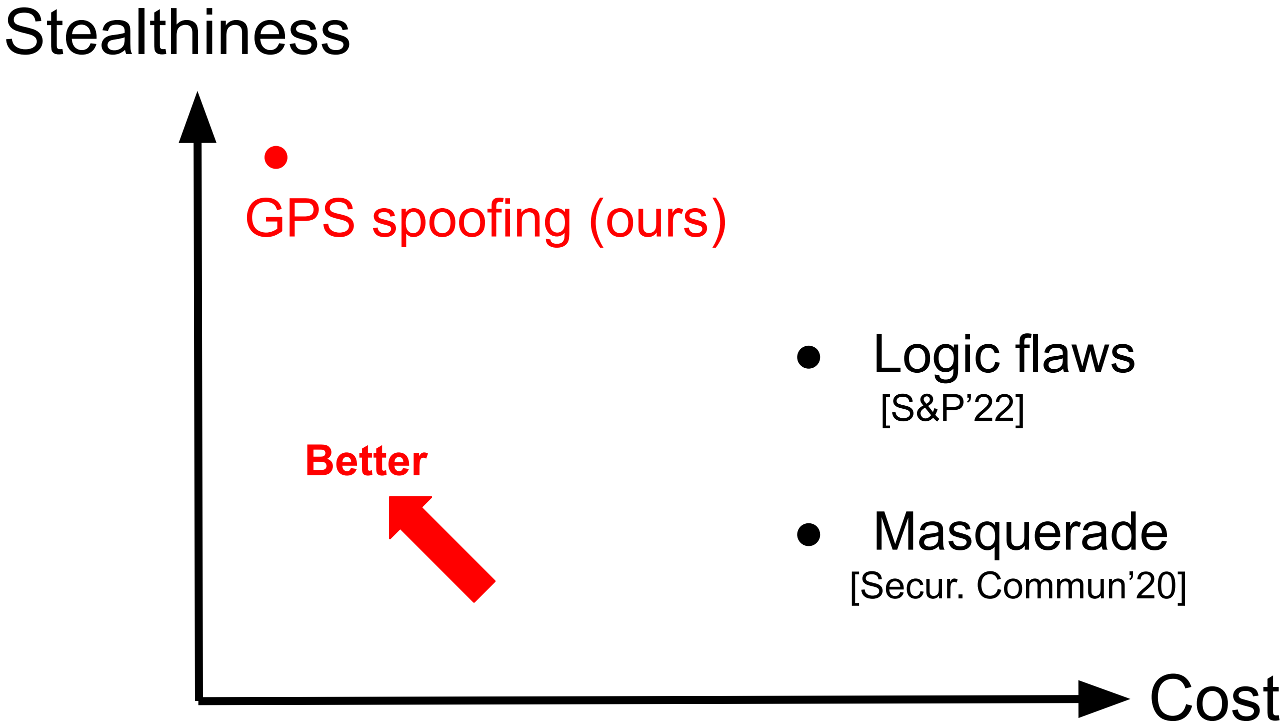


Mission

Sensors

Swarm Control Algorithm (Distributed)

Security Threats



Stealthiness

GPS spoofing (ours)

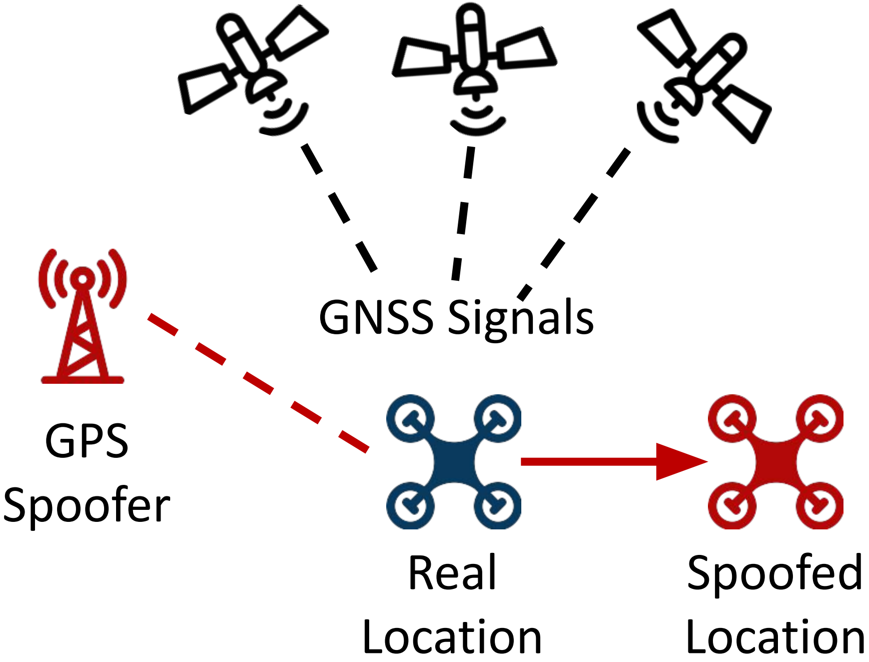
Logic flaws [S&P'22]

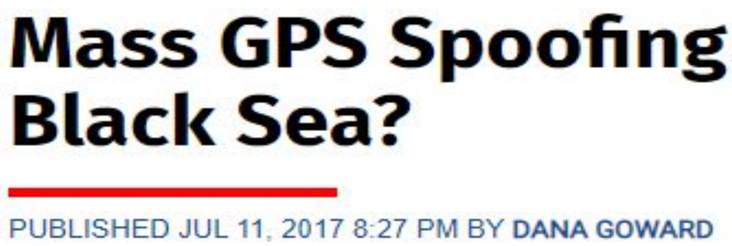
Masquerade [Secur. Commun'20]

Better

Cost

GPS Spoofing Attack

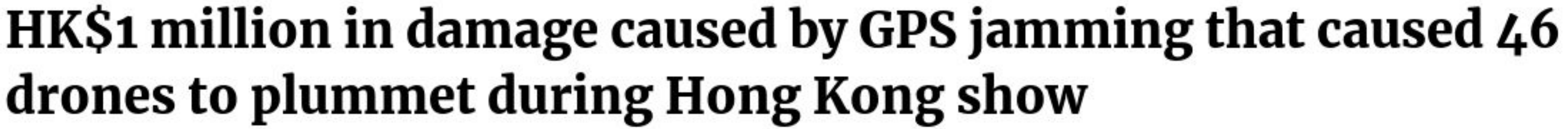




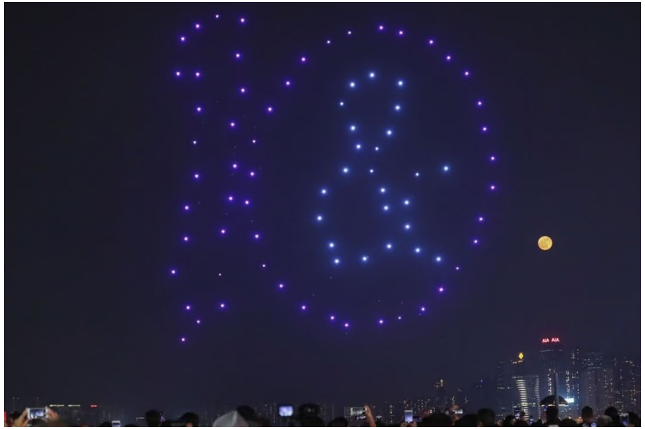
Mass GPS Spoofing Attack in Black Sea?

PUBLISHED JUL 11, 2017 8:27 PM BY **DANA GOWARD**

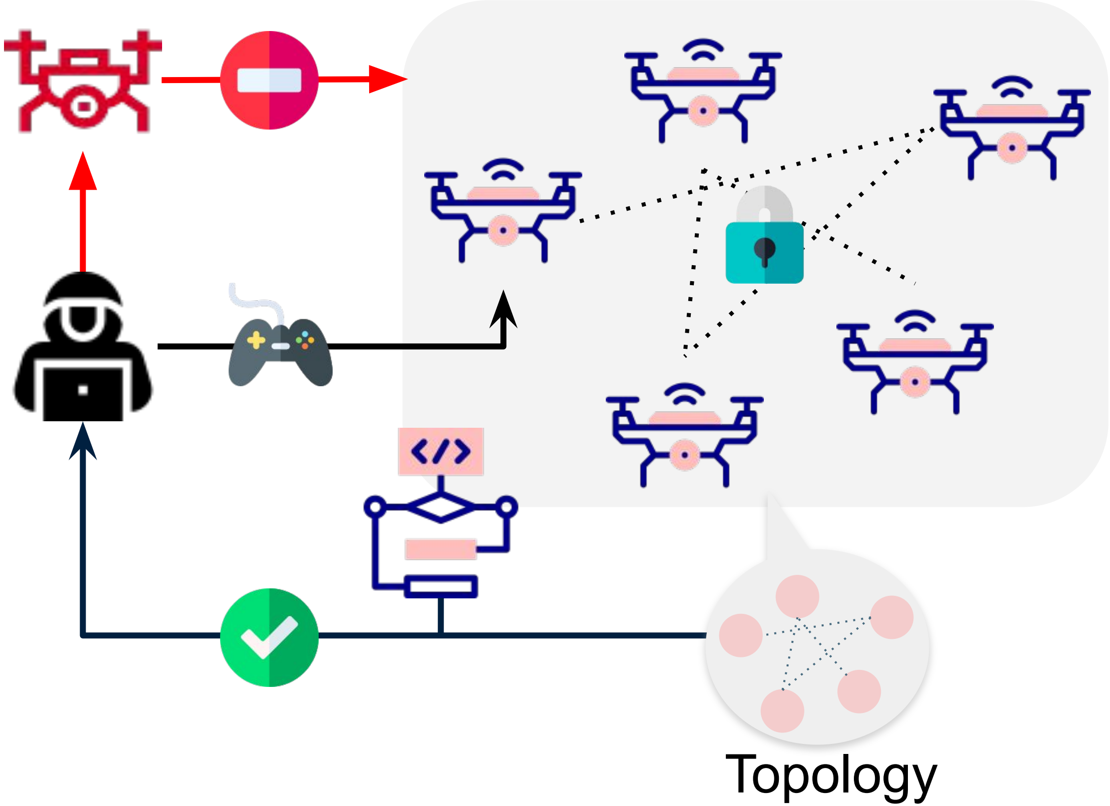
South China Morning Post



HK$1 million in damage caused by GPS jamming that caused 46 drones to plummet during Hong Kong show

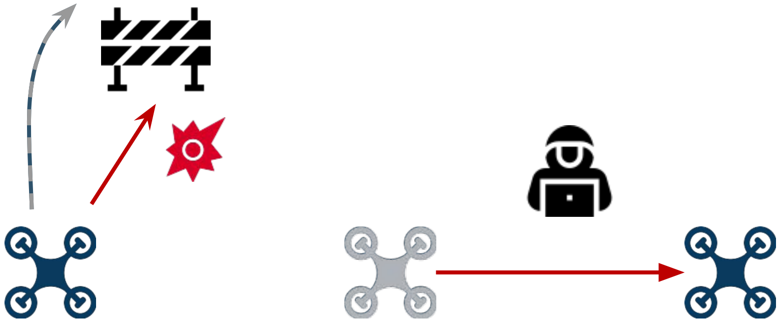


Threat Model



Swarm Propagation Vulnerabilities (SPVs)

- The vulnerabilities exploited by GPS spoofing attacks in drone swarms.



Victim Drone

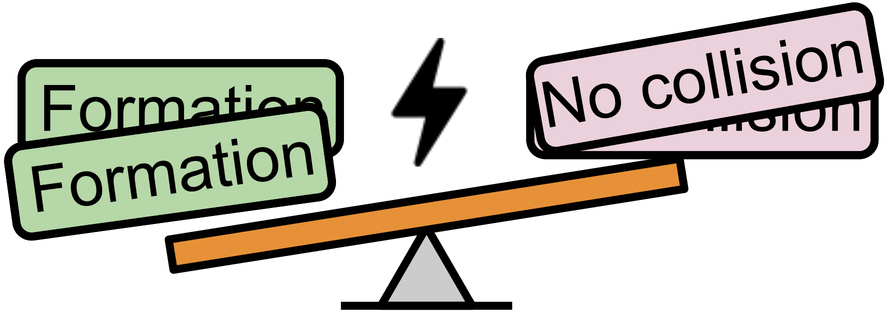
Target Drone
(Being spoofed)

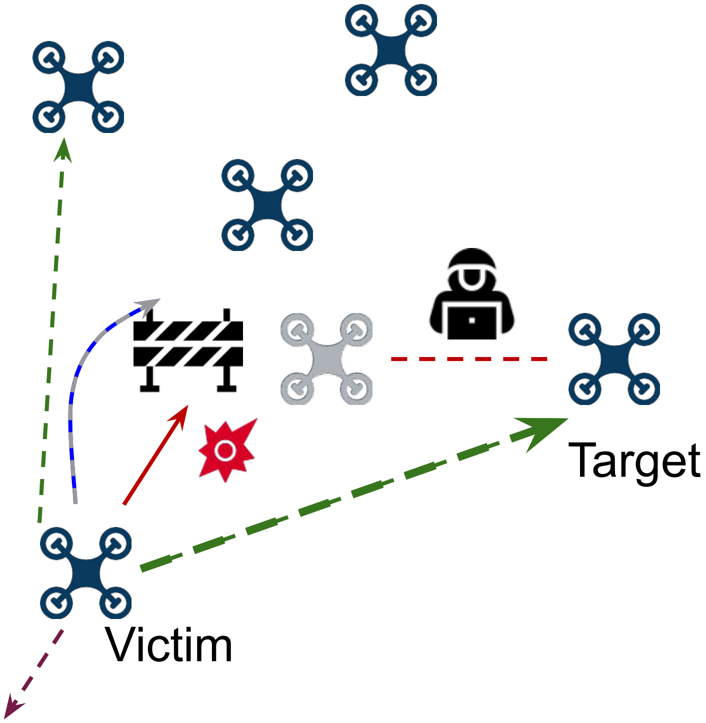
Low cost

Stealthy

What causes SPVs?

Answer: Design choices in swarm control algorithms.





Our goal

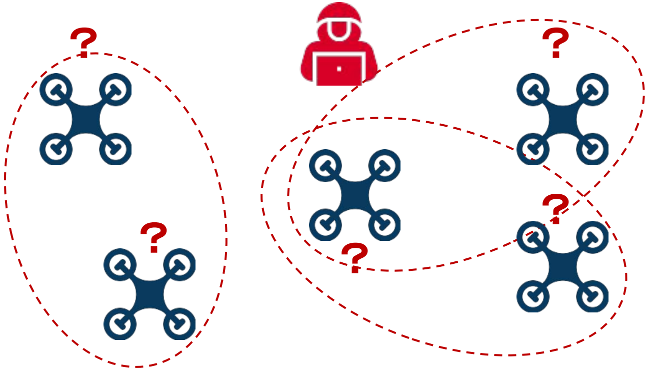
I. To automatically find SPVs before swarm deployment

II. To assess the swarm missions against SPVs

Automatically finding SPVs: Challenges

Challenge 1 (**C1**)

- Selection of target-victim drone pairs
  - A large number of combinations



Observation 1

- Target Drone
  - Most influential
- Victim Drone
  - Under the most influence
  - Closest to the obstacle

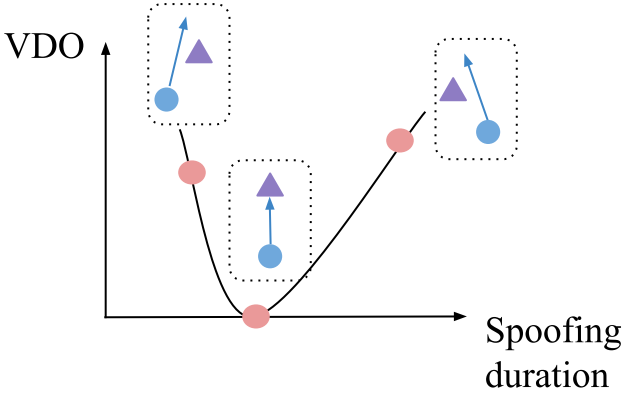
Automatically finding SPVs: Challenges

*Challenge 2 (**C2**)*

- Selection of attack parameters
  - Spoofing start time
  - Spoofing duration

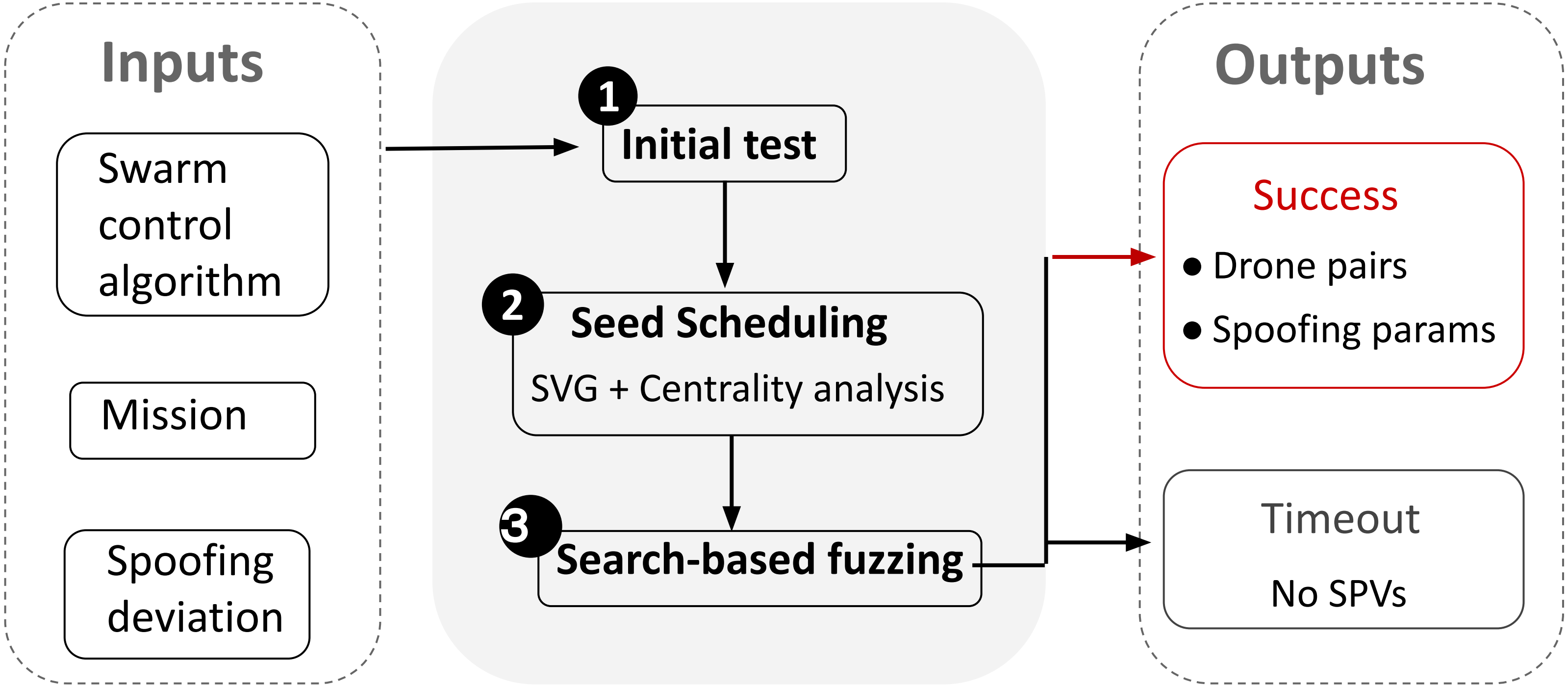
Observation 2

- Minimize VDO* distance
  - Convex optimization



*VDO: the Victim drone's closest Distance to the Obstacle

Our solution: SwarmFuzz



Inputs

- Swarm control algorithm
- Mission
- Spoofing deviation

1. Initial test
2. Seed Scheduling

   SVG + Centrality analysis
3. Search-based fuzzing

Outputs

Success

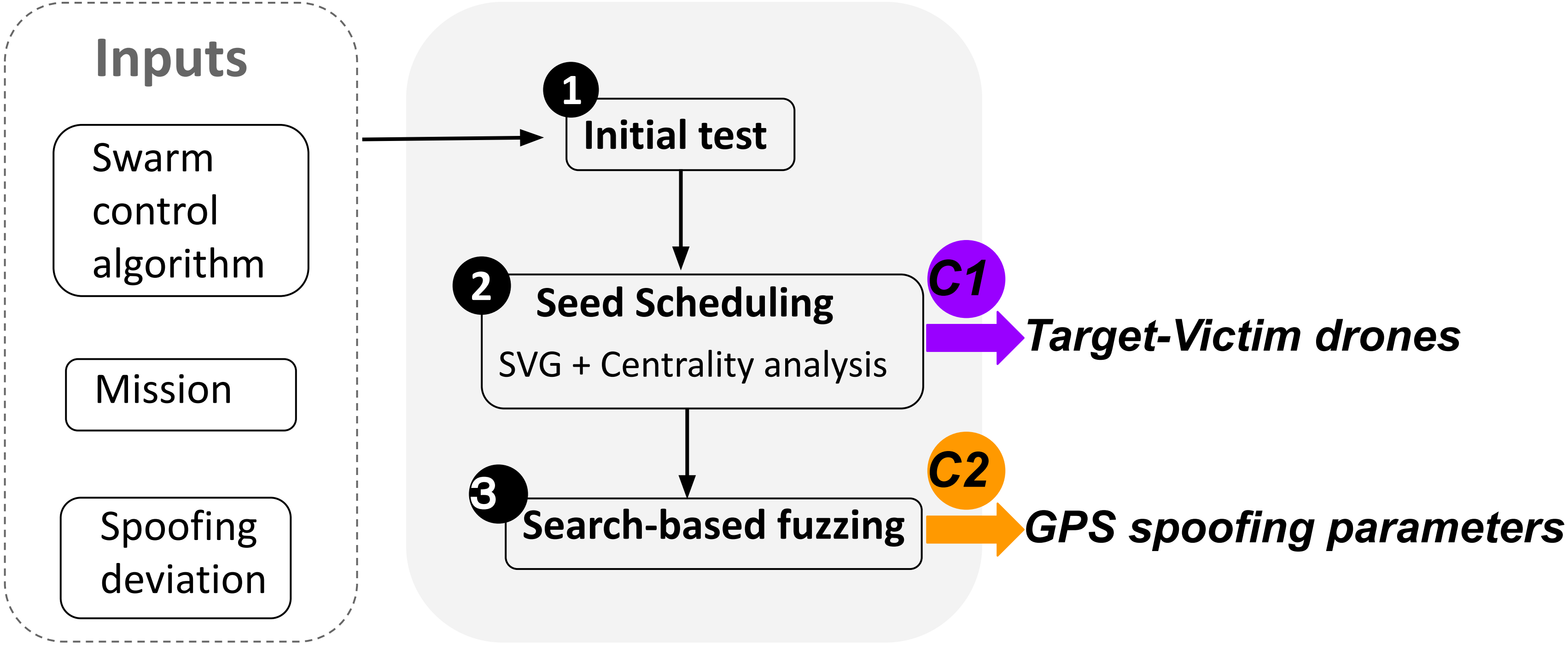
- Drone pairs
- Spoofing params

Timeout

No SPVs

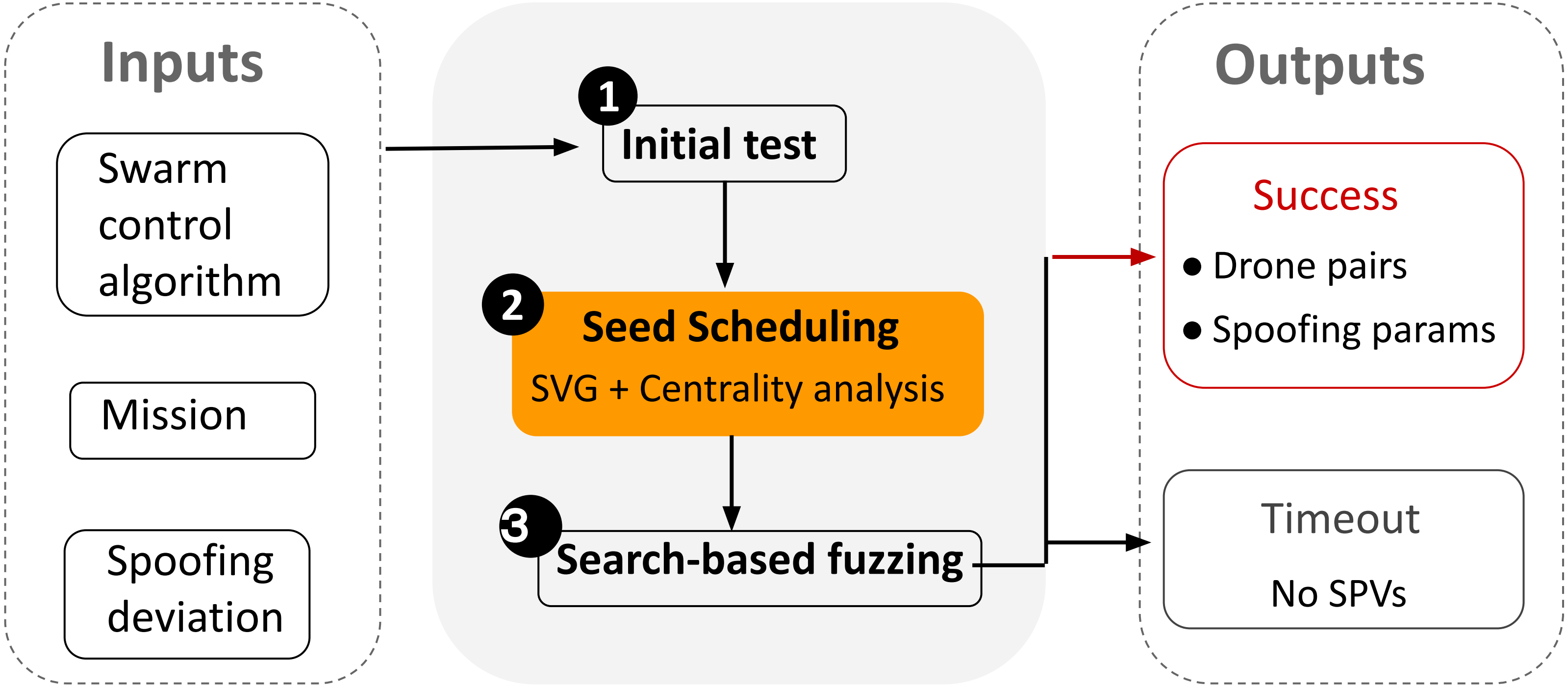
SVG: Swarm vulnerability graph

Our solution: SwarmFuzz



*SVG: Swarm vulnerability graph

Our solution: SwarmFuzz



Inputs

- Swarm control algorithm
- Mission
- Spoofing deviation

1 Initial test

2 Seed Scheduling

SVG + Centrality analysis

3 Search-based fuzzing

Outputs

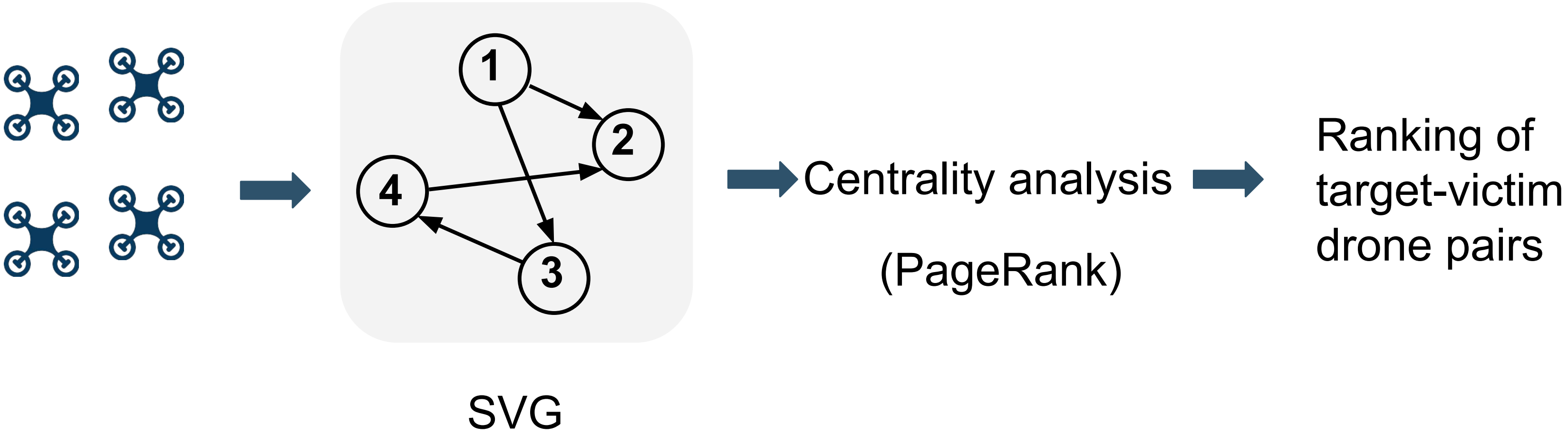
Success

- Drone pairs
- Spoofing params

Timeout

No SPVs

Seed Scheduling



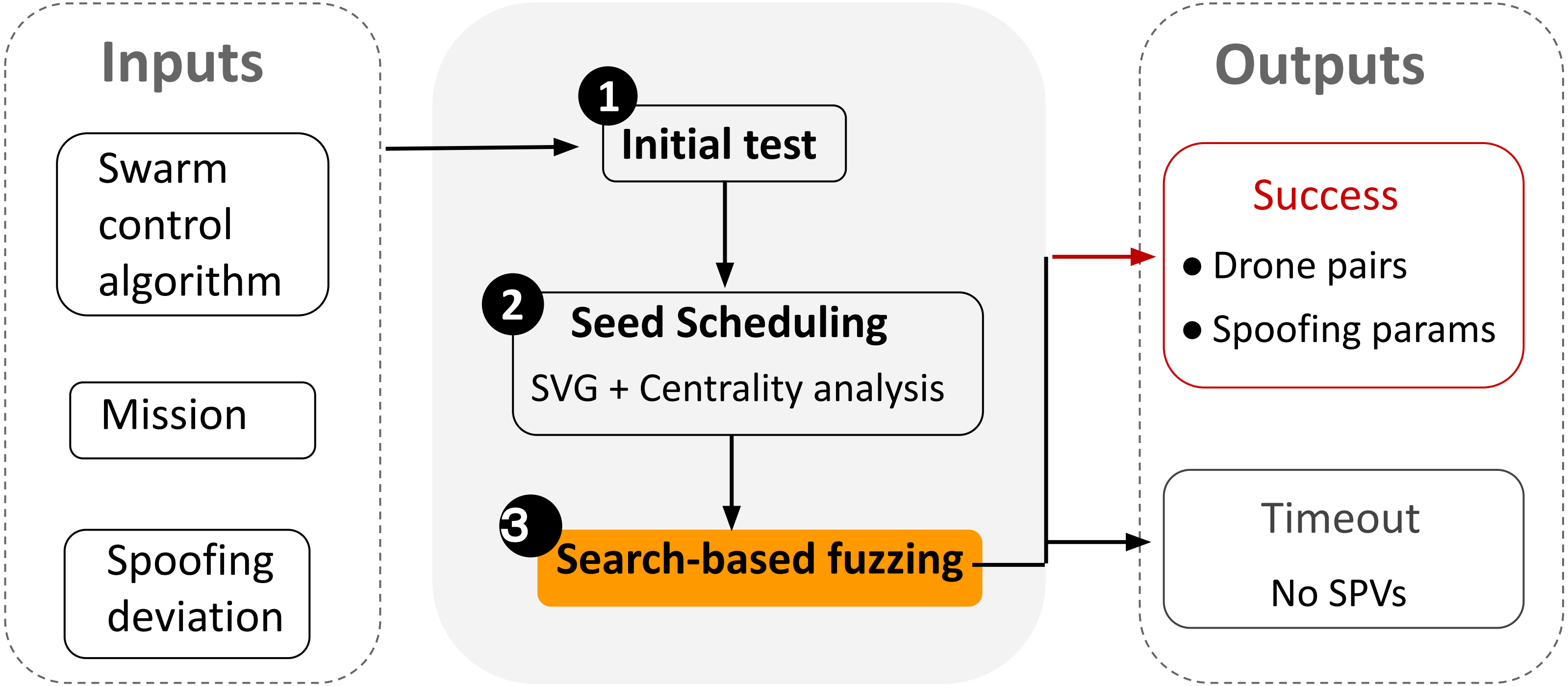
SVG

Centrality analysis

(PageRank)

Ranking of target-victim drone pairs

Our solution: SwarmFuzz



Inputs

Swarm control algorithm

Mission

Spoofing deviation

1 Initial test

2 Seed Scheduling

SVG + Centrality analysis

3 Search-based fuzzing

Outputs

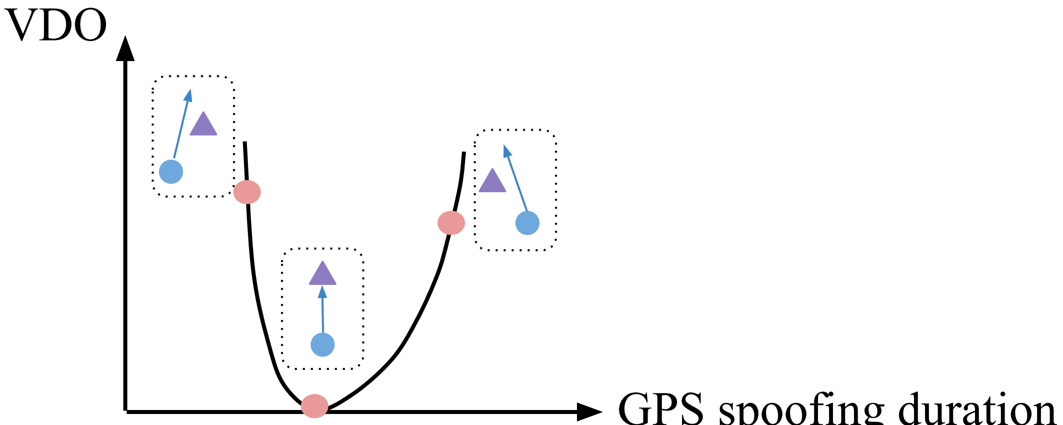
Success

- Drone pairs
- Spoofing params

Timeout

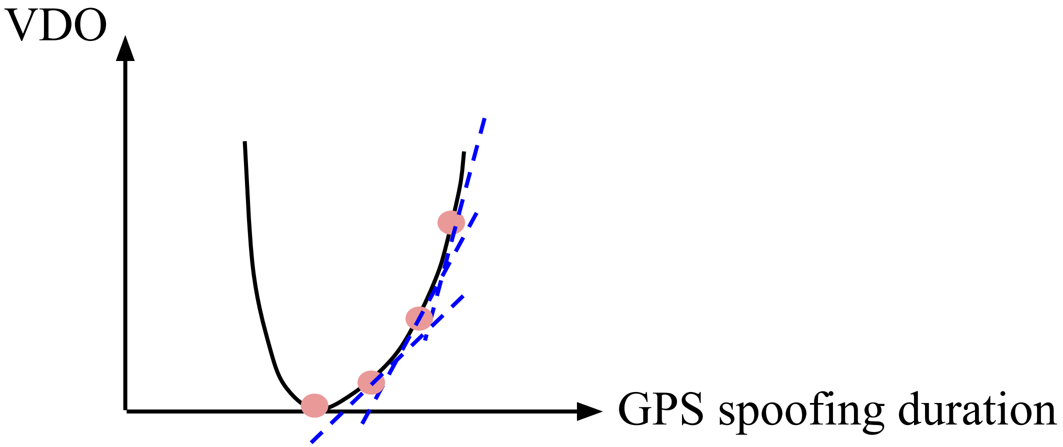
No SPVs

Search-based fuzzing



Convex optimization

Search-based fuzzing

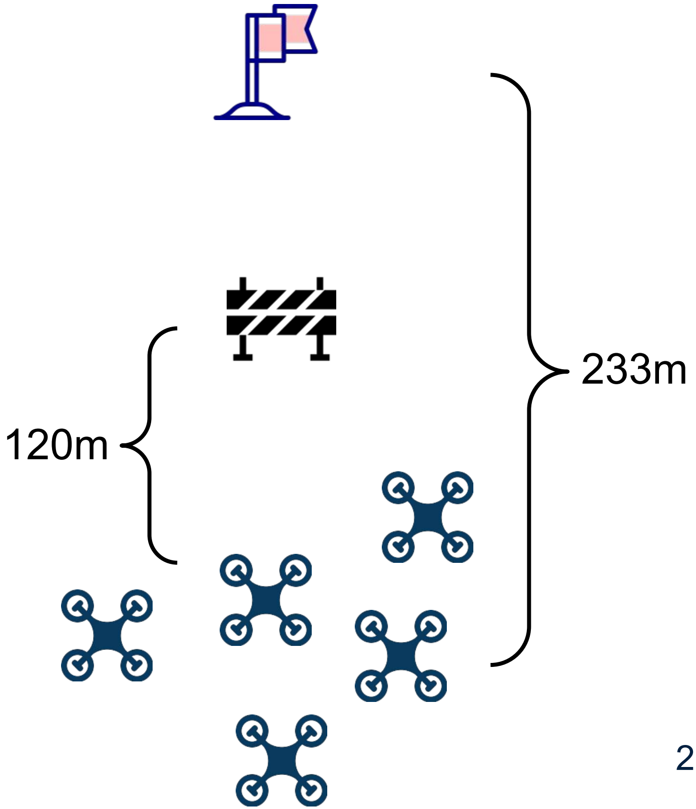


Convex optimization

Gradient-descent search

Evaluation

- Simulator: Swarmlab
- Swarm control algorithm: Viscek
- Swarm size: 5 /10 / 15 drones
- GPS spoofing deviation: 5 /10m (acceptable GPS fault)
- Success: victim drone crashes



Effectiveness of SwarmFuzz

Success rates of SwarmFuzz in finding SPVs

	5 drones	10 drones	15 drones
5m spoofing	*21%*	*36%*	*54%*
10m spoofing	*49%*	*59%*	*74%*

Avg. *48.8%*

Highly effective for different swarm configurations

Effectiveness of SwarmFuzz

Success rates of SwarmFuzz in finding SPVs

	5 drones	10 drones	15 drones
5m spoofing	*21%*	*36%*	*54%*
10m spoofing	*49%*	*59%*	*74%*

Larger swarm sizes → Higher success rate

Effectiveness of SwarmFuzz

Success rates of SwarmFuzz in finding SPVs

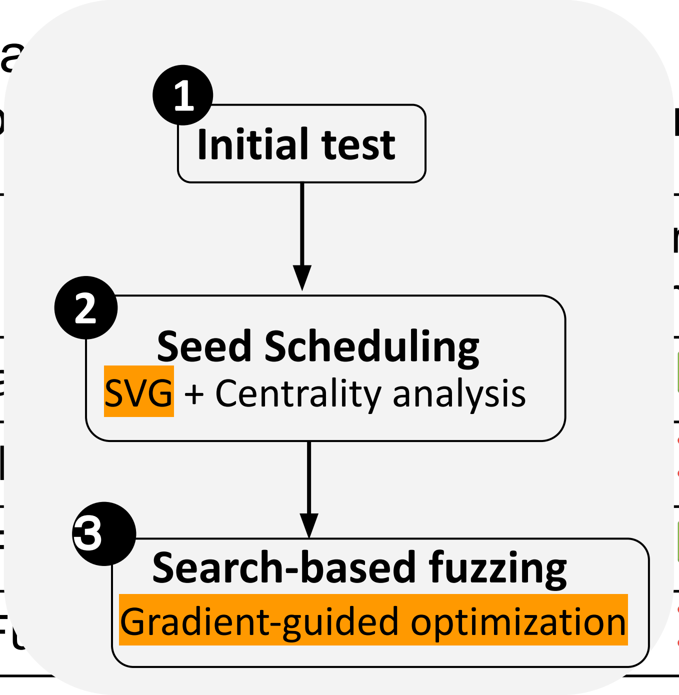
	5 drones	10 drones	15 drones
5m spoofing	*21%*	*36%*	*54%*
10m spoofing	*49%*	*59%*	*74%*

Larger GPS spoofing deviation → Higher success rate

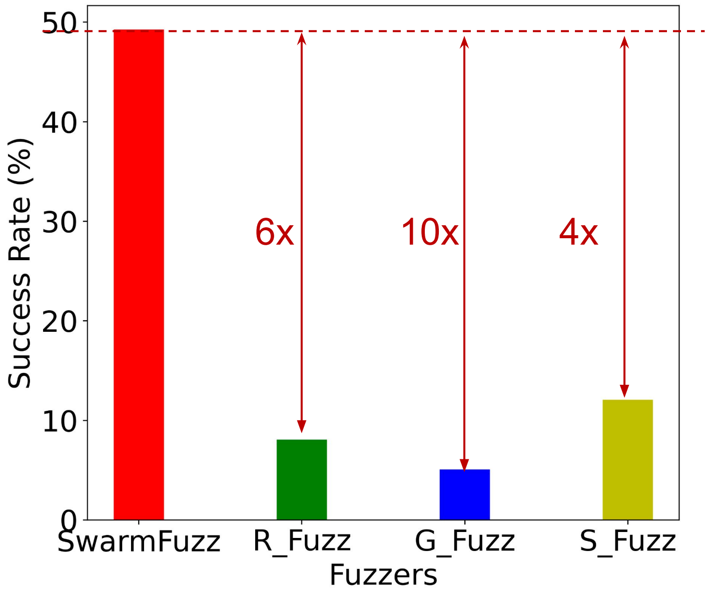
Ablation study

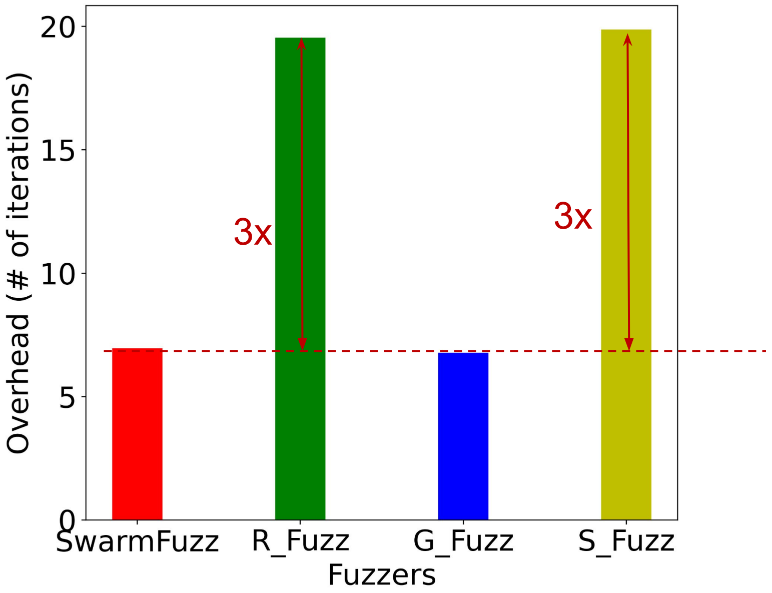
- Metrics
  - Success ra
  - Runtime o ns)

	nt-guided ization
Swa	✅
R_ I	❌
G_F	✅
S_Fu	❌



Ablation study





SVG boosts the success rate by up to 10x.

Gradient-guided optimization reduces the overhead by up to 3x.

Takeaways

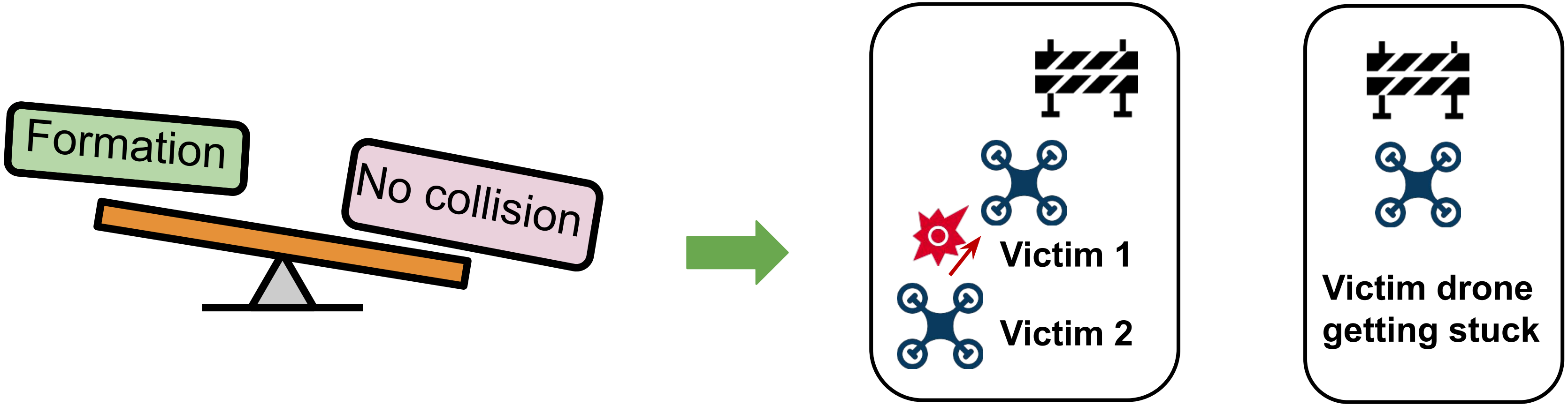
- Swarm missions with a **larger size** are more vulnerable
  - ➢ Secure large-size drone swarms

- If the swarm mission is found to be vulnerable to SPVs
  - ➢ **Tune the parameters** in the control algorithm

- Need **fault-tolerance** mechanisms

Future work

- Extend SwarmFuzz to other swarm control algorithms



Formation

No collision

Victim 1

Victim 2

Victim drone getting stuck

Summary

- **SPVs**: vulnerabilities in swarm control algorithms exploited by GPS spoofing attacks
- **SwarmFuzz: A fuzzing framework to discover SPVs, and help to evaluate the resilience of the swarm beforehand**
- Use SVG and gradient descent to find SPVs efficiently
- Code at: https://github.com/DependableSystemsLab/SwarmFuzz

Yingao (Elaine) Yao
elainey@ece.ubc.ca

Summary

-