A Novel Side-Channel in Real-Time Schedulers

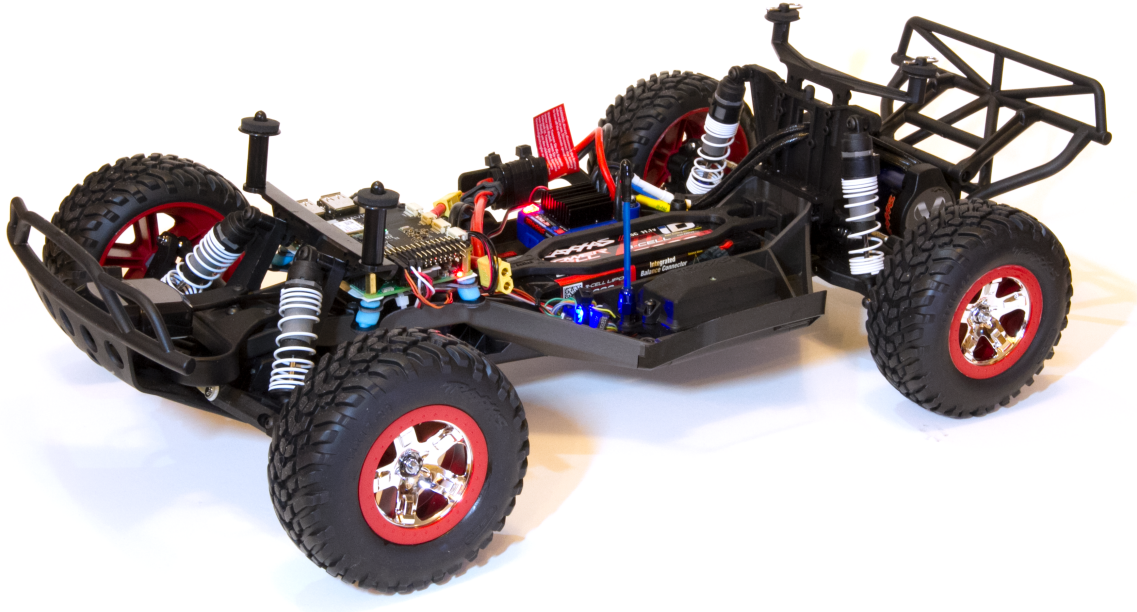
Chien-Ying (CY) Chen, Sibin Mohan, Rodolfo Pellizzoni,

Rakesh B. Bobba and Negar Kiyavash

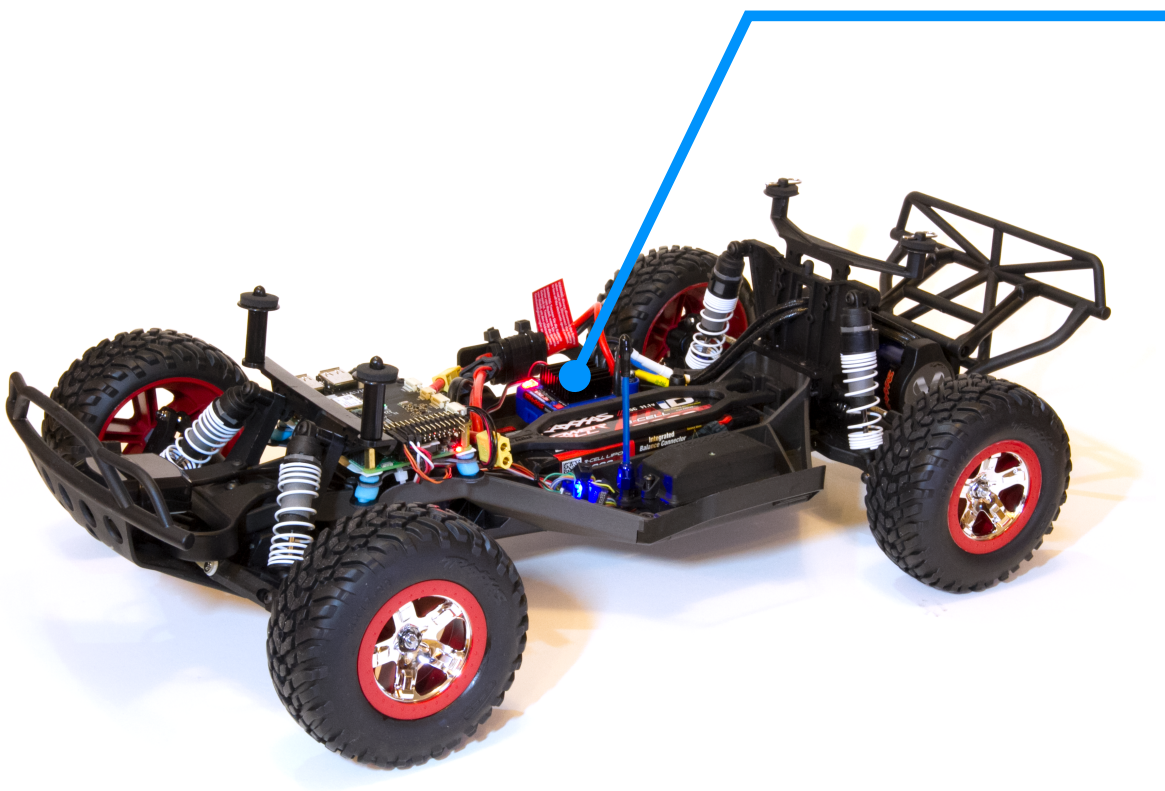
25th IEEE Real-time And Embedded Technology And Applications Symposium (RTAS'19)
April 17, 2019

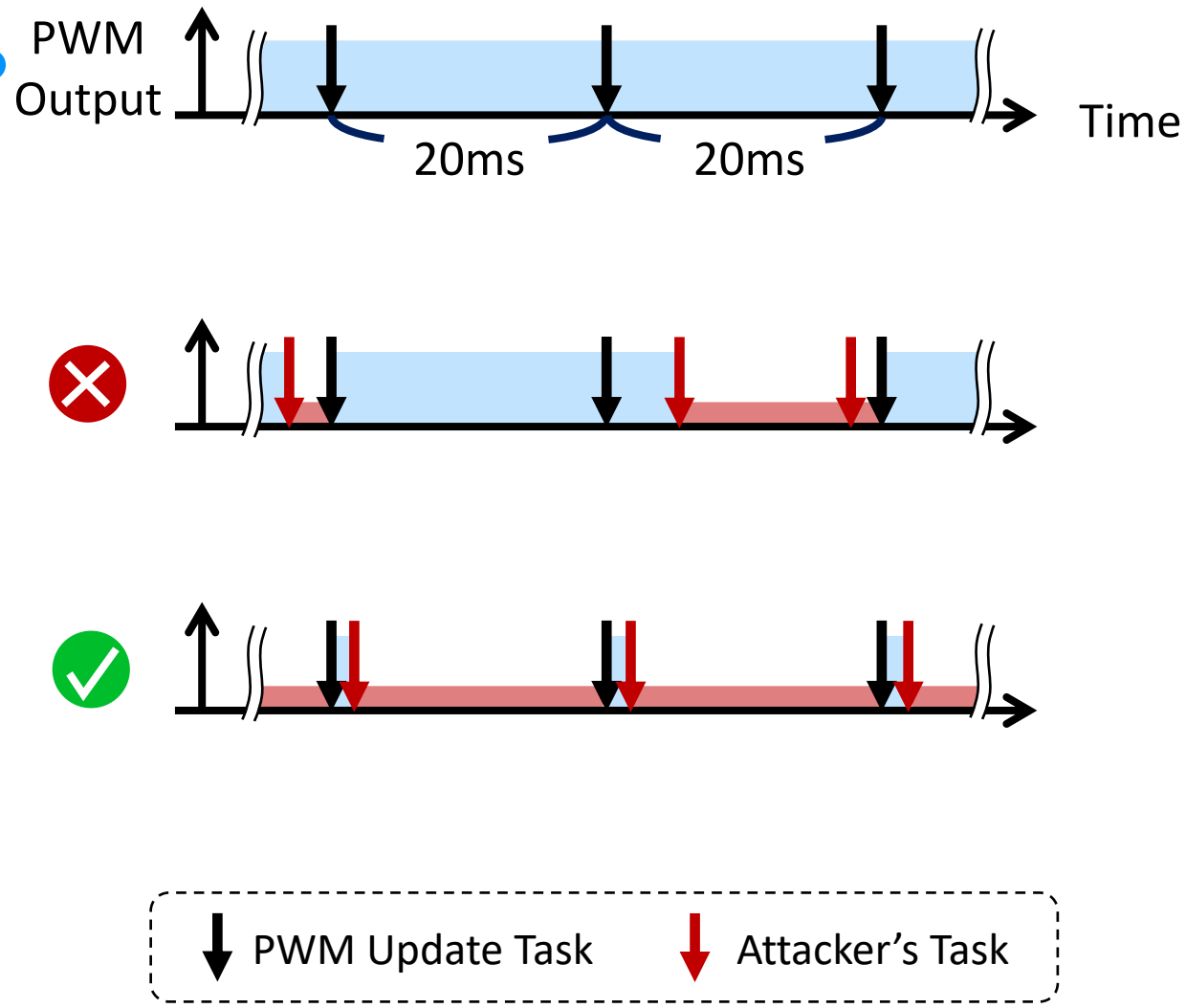
This work is supported by the National Science Foundation (NSF) under grant SaTC-1718952.

Imagine you (an attacker) have control of a real-time task in an autonomous system



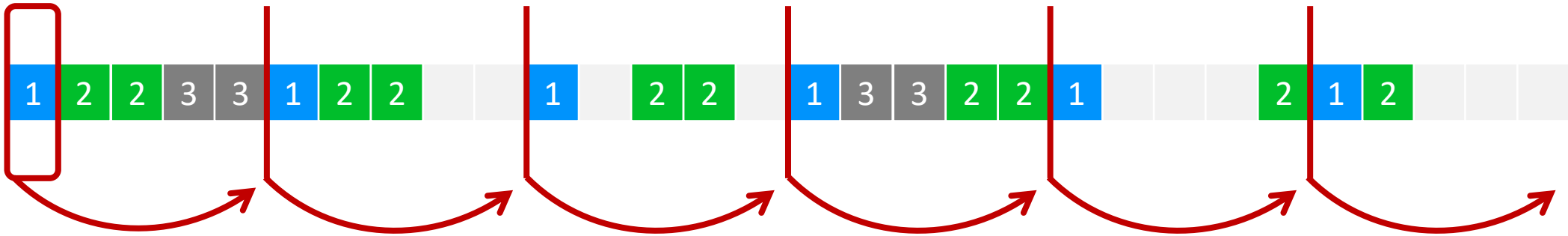
You want to take over control of the steering and throttle




PWM
Output
Time
20ms
20ms
PWM Update Task
Attacker's Task

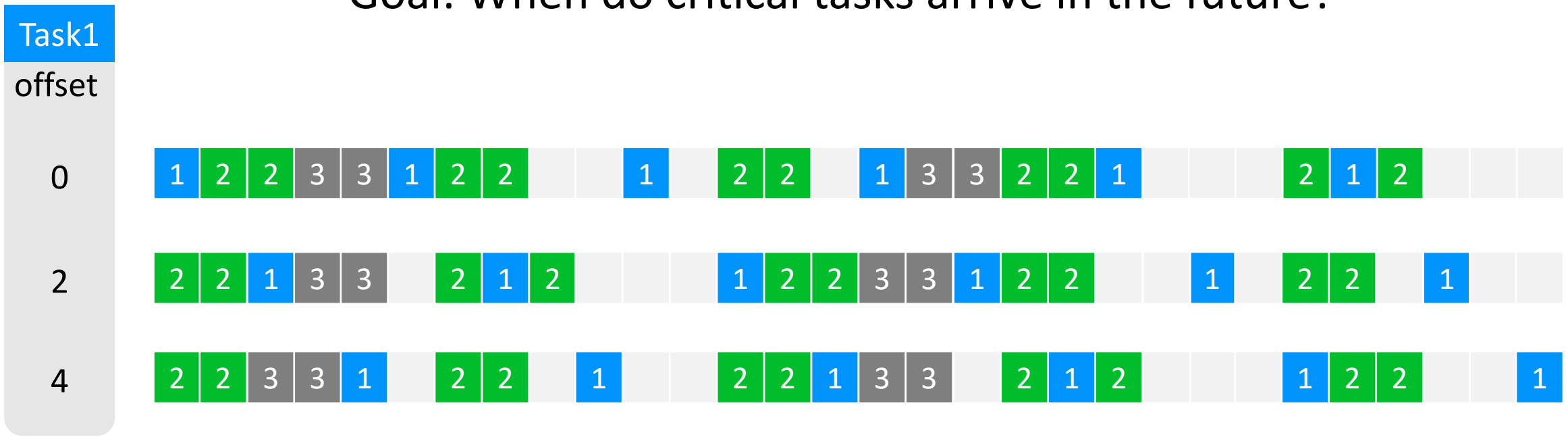
Problem Statement

Goal: When do critical tasks arrive in the future?



Problem Statement

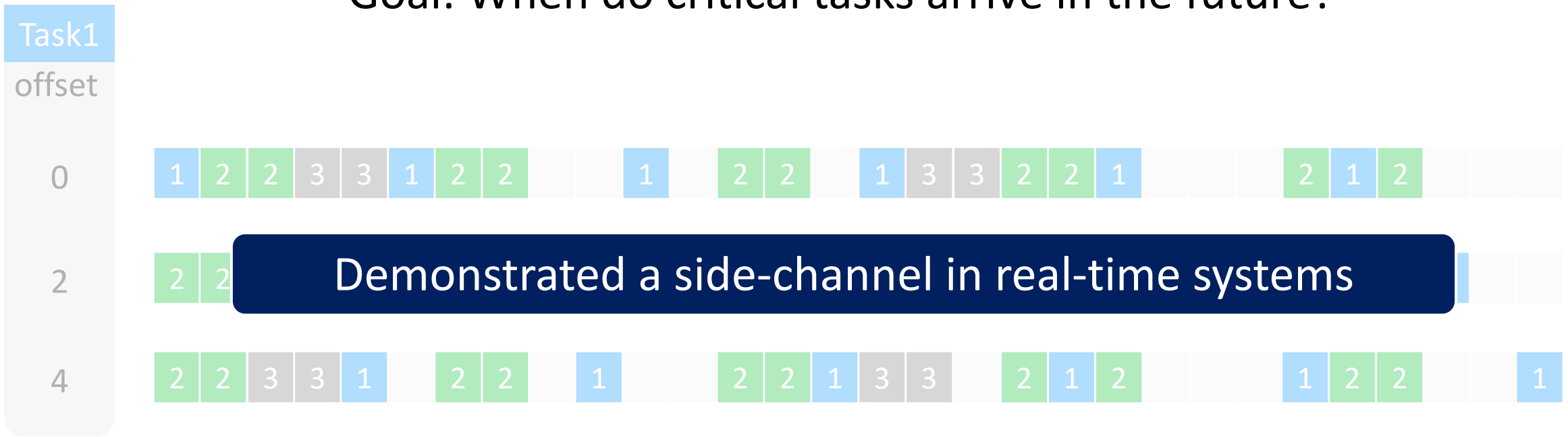
Goal: When do critical tasks arrive in the future?



Different task initial offsets yield completely different schedules.

Problem Statement

Goal: When do critical tasks arrive in the future?

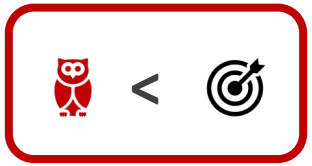


Different task initial offsets yield completely different schedules.

System and Adversary Model

- Uniprocessor, Fixed-Priority Hard Real-Time Systems
  - Attacker's task (observer task) *periodic* or *sporadic*
  - Victim task *periodic*
  - Other tasks *periodic* or *sporadic*

- Requirements
  - The attacker knows the **victim task's period**
  - The observer task has **lower priority** than the victim task.



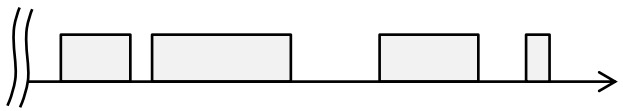
- Attack Goal
  - Infer the victim task's **initial offset** and predict its **future arrival time points**.

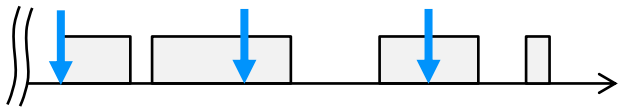
Attack Scenario Overview

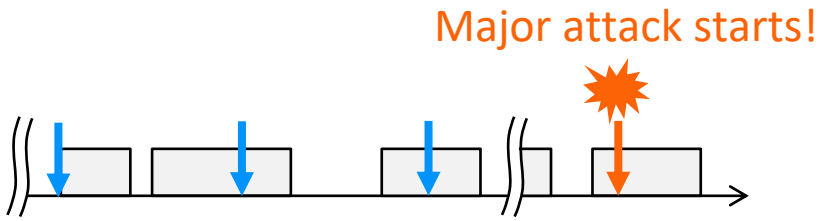
The attacker **observes** the schedule on the victim system.

The attacker **analyzes** and **infers** precise timing information of the victim task.

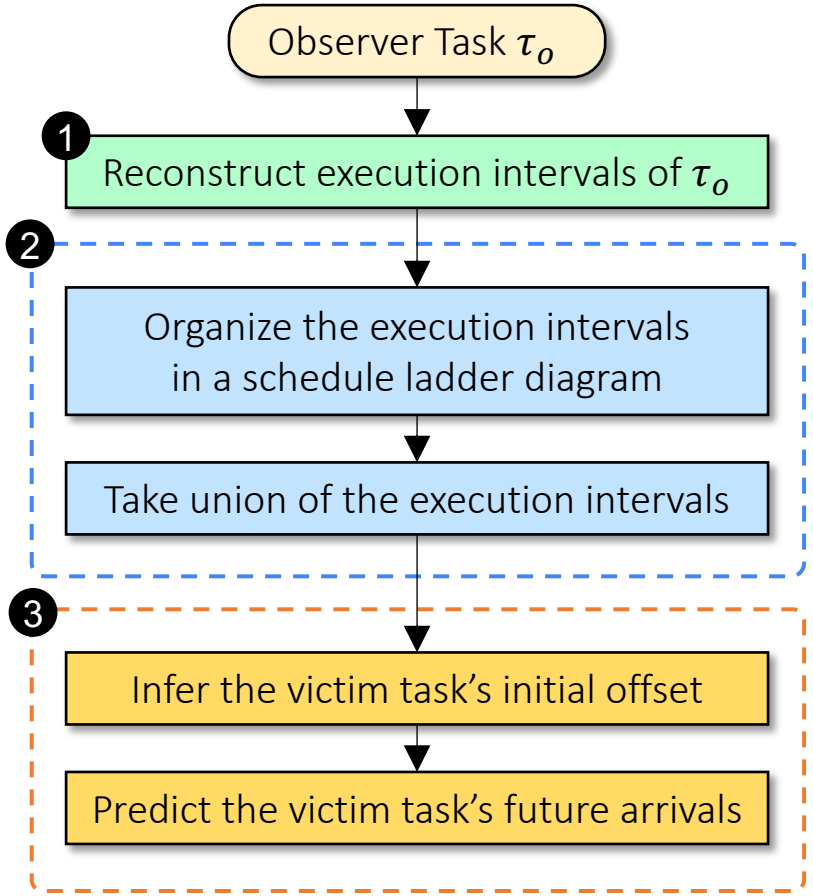
The attacker **launches a major attack** at a future instant that can cause the most amount of damage.







ScheduLeak Algorithms



Observe and reconstruct

Analyze and extract

Infer and predict

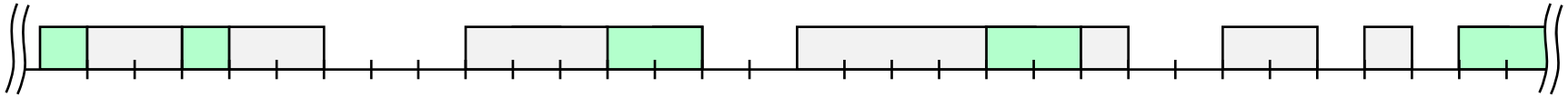
The ScheduLeak Algorithms

1. Reconstruct execution intervals of τ_o

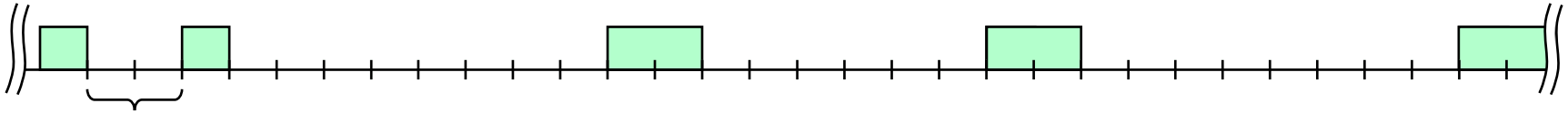
The observer task has **lower priority** than the victim task

Task ID	Period	Execution Time
Task 4	15	1
Observer Task	10	2
Victim Task	8	2
Task 1	6	1

System Schedule Ground Truth:



What the attacker observes:



Some tasks preempted the observer task.

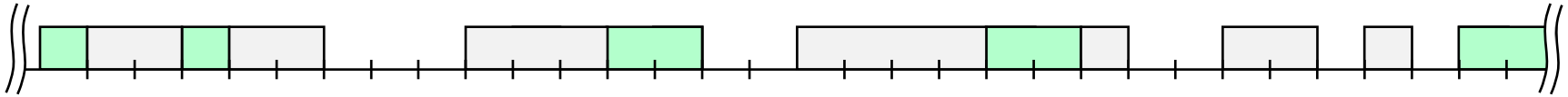
Observer Task τ_o Other Tasks

The ScheduLeak Algorithms

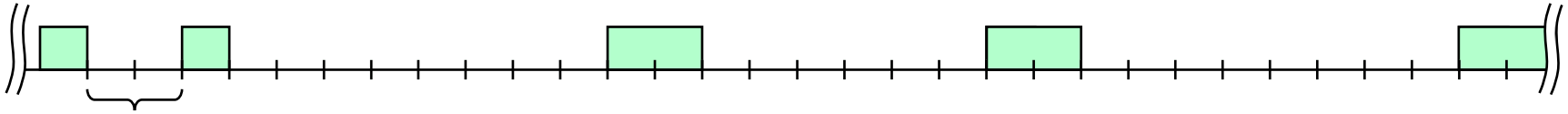
Task ID	Period	Execution Time
Task 4	15	1
Observer Task	10	2
Victim Task	8	2
Task 1	6	1

2 Organize the execution intervals in a schedule ladder diagram

System Schedule Ground Truth:



What the attacker observes:



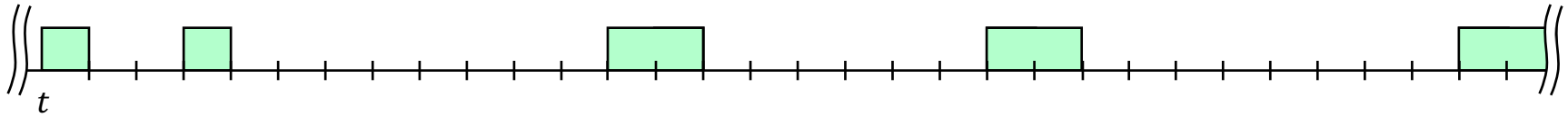
Some tasks preempted the observer task.

Observer Task τ_o Other Tasks

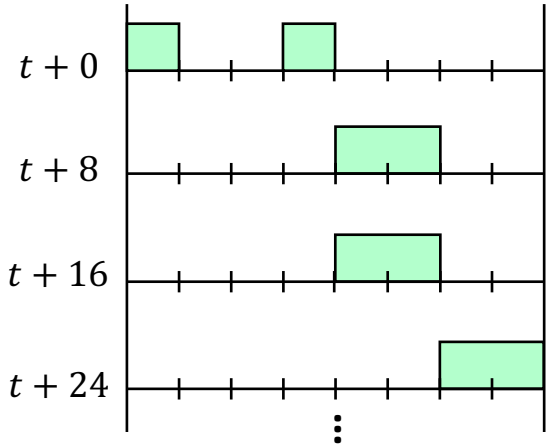
The ScheduLeak Algorithms

2 Organize the execution intervals in a schedule ladder diagram

Task ID	Period	Execution Time
Task 4	15	1
Observer Task	10	2
Victim Task	8	2
Task 1	6	1



Divide the timeline into sections of length = 8 (the victim task's period) and stack:

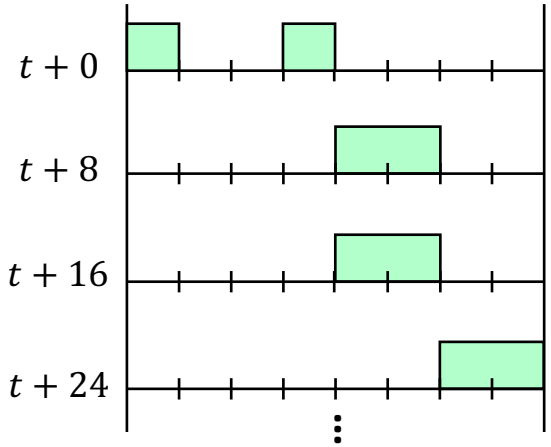


The ScheduLeak Algorithms

2. Organize the execution intervals in a schedule ladder diagram

→ Take union of the execution intervals

Task ID	Period	Execution Time
Task 4	15	1
Observer Task	10	2
Victim Task	8	2
Task 1	6	1

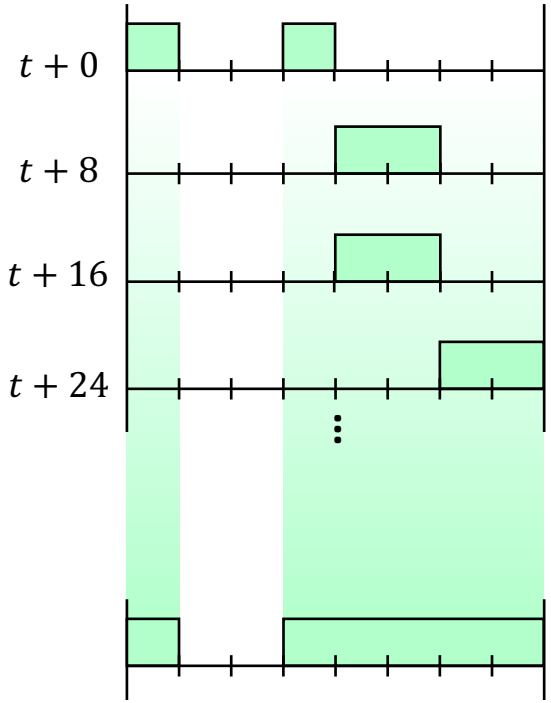


The ScheduLeak Algorithms

2. Organize the execution intervals in a schedule ladder diagram

→ Take union of the execution intervals

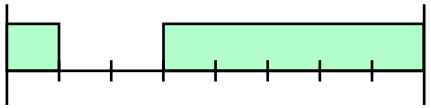
Task ID	Period	Execution Time
Task 4	15	1
Observer Task	10	2
Victim Task	8	2
Task 1	6	1



The ScheduLeak Algorithms

3 Infer the victim task's initial offset

Task ID	Period	Execution Time
Task 4	15	1
Observer Task	10	2
Victim Task	8	2
Task 1	6	1

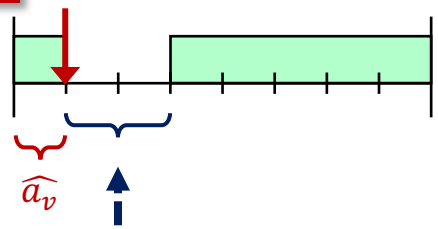


The ScheduLeak Algorithms

Task ID	Period	Execution Time
Task 4	15	1
Observer Task	10	2
Victim Task	8	2
Task 1	6	1

3 Infer the victim task's initial offset

We take the **earliest time point** of the empty column as the inference of the victim task's initial offset.



Occupied by the tasks with higher priorities (e.g. the victim task)

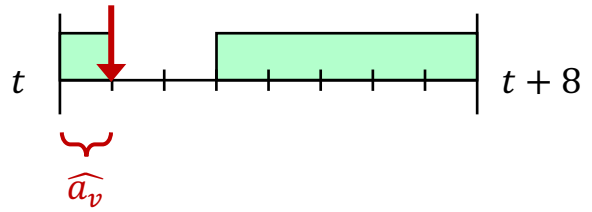
The ScheduLeak Algorithms

3 Infer the victim task's initial offset

↓

Predict the victim task's future arrivals

Task ID	Period	Execution Time
Task 4	15	1
Observer Task	10	2
Victim Task	8	2
Task 1	6	1



The **victim task's future arrival times** can be computed by

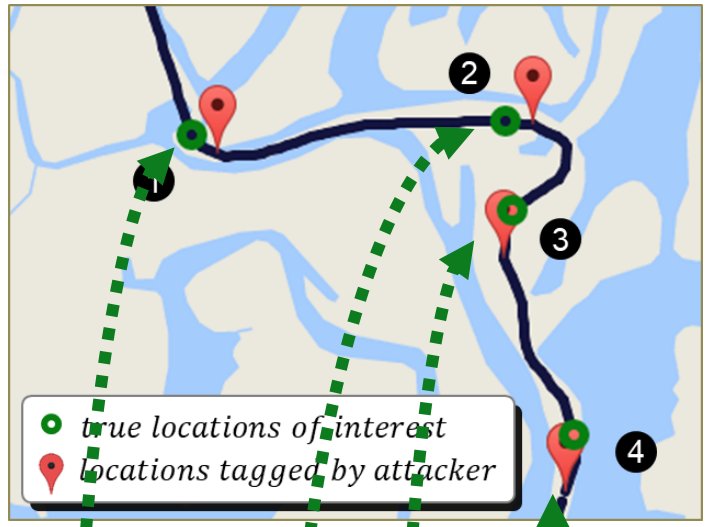
$$t + \widehat{a_v} + p_v \times T$$

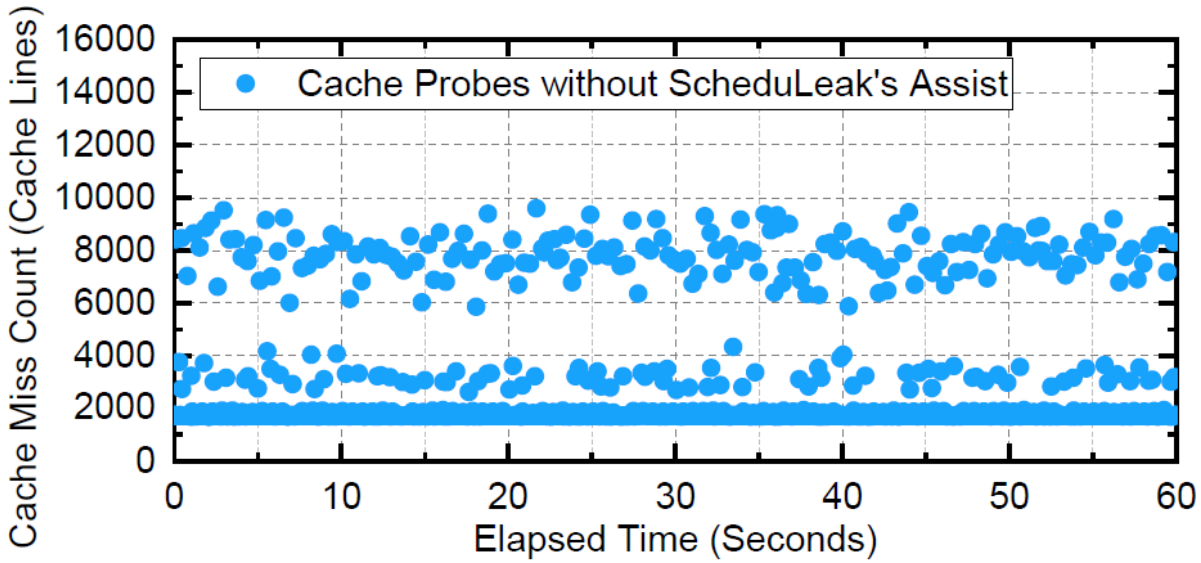
- t: ladder diagram starting point
- $\widehat{a_v}$: Inferred victim task's Initial offset
- p_v: victim task's period
- T: desired arrival number

Demonstration 2
Cache-Timing Side-Channel Attacks

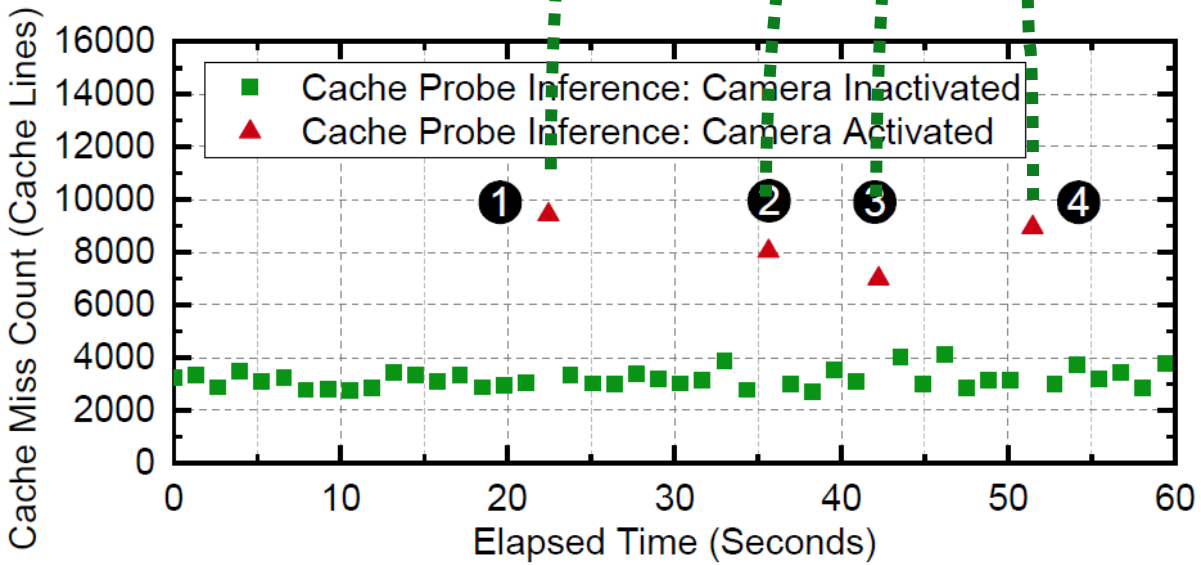
Attack Goals:

- Probe (coarse-grained) memory usage of victim task
- Recover locations of interest points where memory usage (of victim task) is high





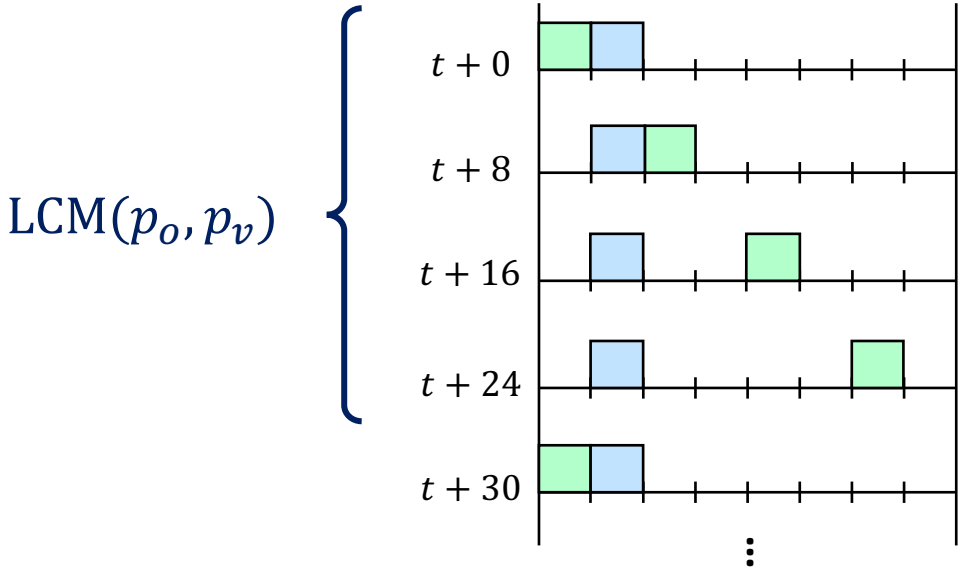
Without ScheduLeak-based Information



With ScheduLeak-based Information

Correlation between Observer Task & Victim Task

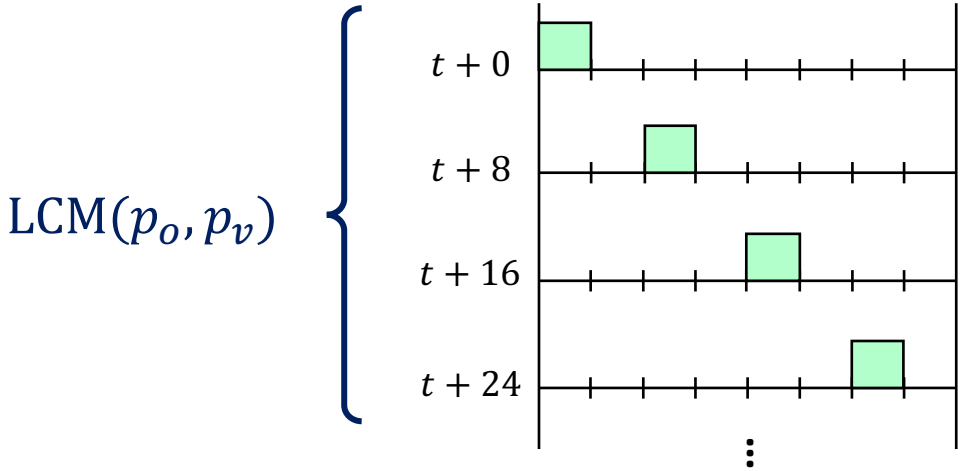
Task	Period
Observer Task	10
Victim Task	8



A ladder diagram with width = 8 (the victim task's period)

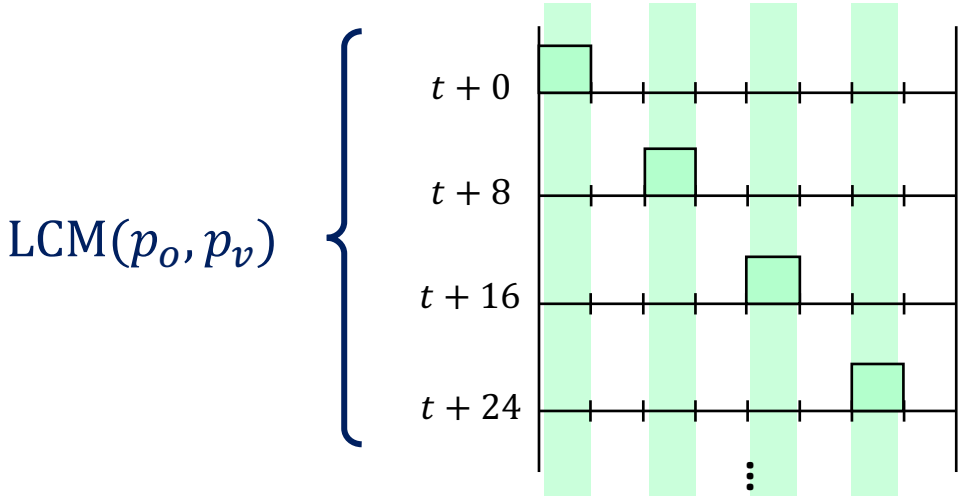
Correlation between Observer Task & Victim Task

Task	Period
Observer Task	10
Victim Task	8



Correlation between Observer Task & Victim Task

$C(\tau_o, \tau_v) = 0.5$



Task	Period
Observer Task	10
Victim Task	8

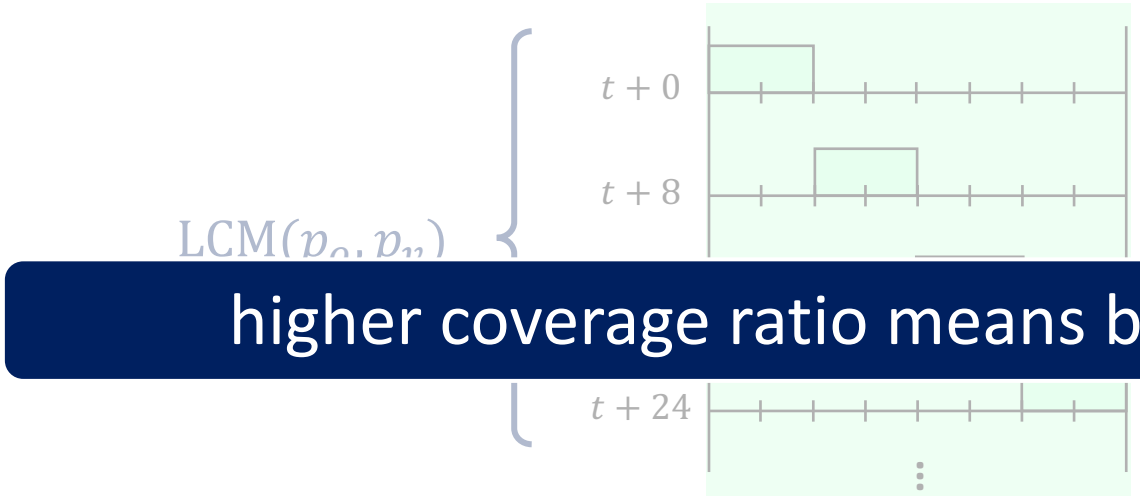
Coverage Ratio

$$C(\tau_o, \tau_v) = \frac{e_o}{GCD(p_o, p_v)}$$

Correlation between Observer Task & Victim Task

$C(\tau_o, \tau_v) = 1.0$

Task	Period
Observer Task	10
Victim Task	8



higher coverage ratio means better inference

Coverage Ratio

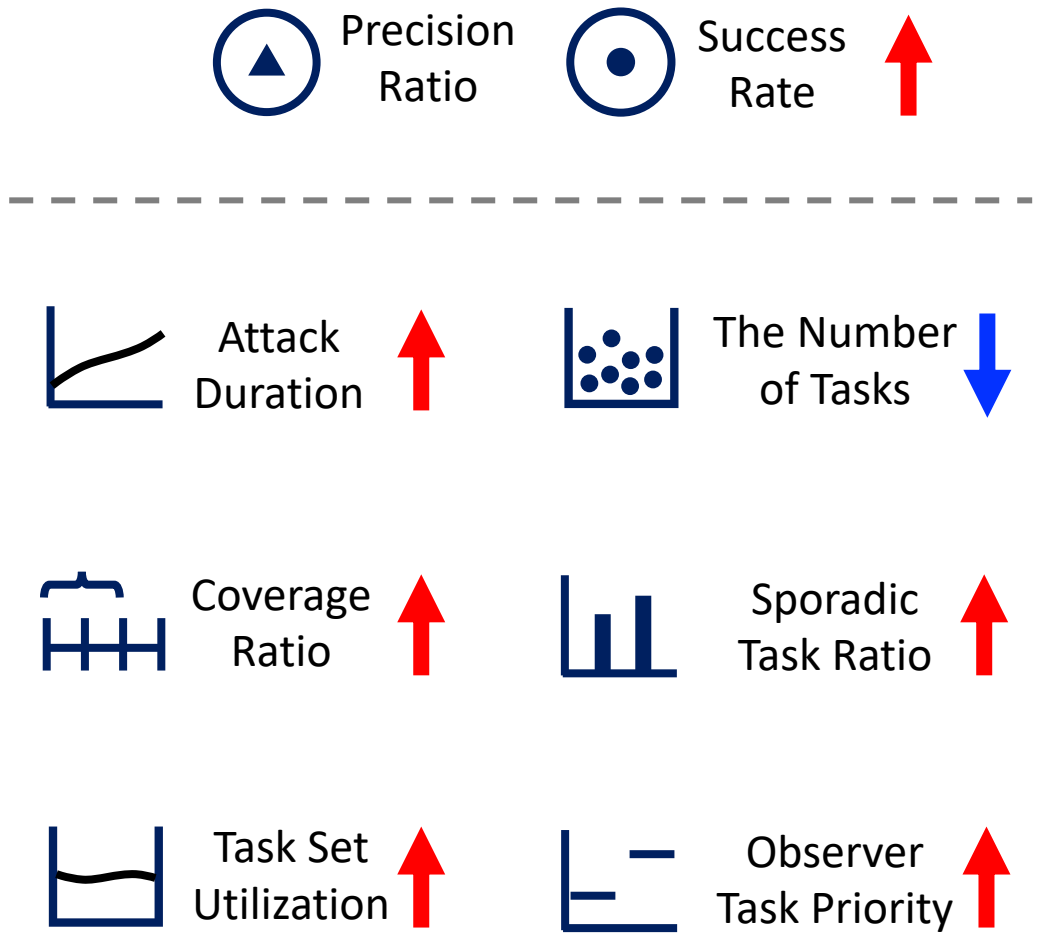
$$C(\tau_o, \tau_v) = \frac{e_o}{GCD(p_o, p_v)}$$

Simulation-based Performance Evaluation

- Metrics
  - **Inference Precision Ratio**
    the ratio of how close the inference to the true initial offset
  - **Inference Success Rate**
    an inference is successful if attacker is able to exactly infer the victim task's initial offset
- Variables

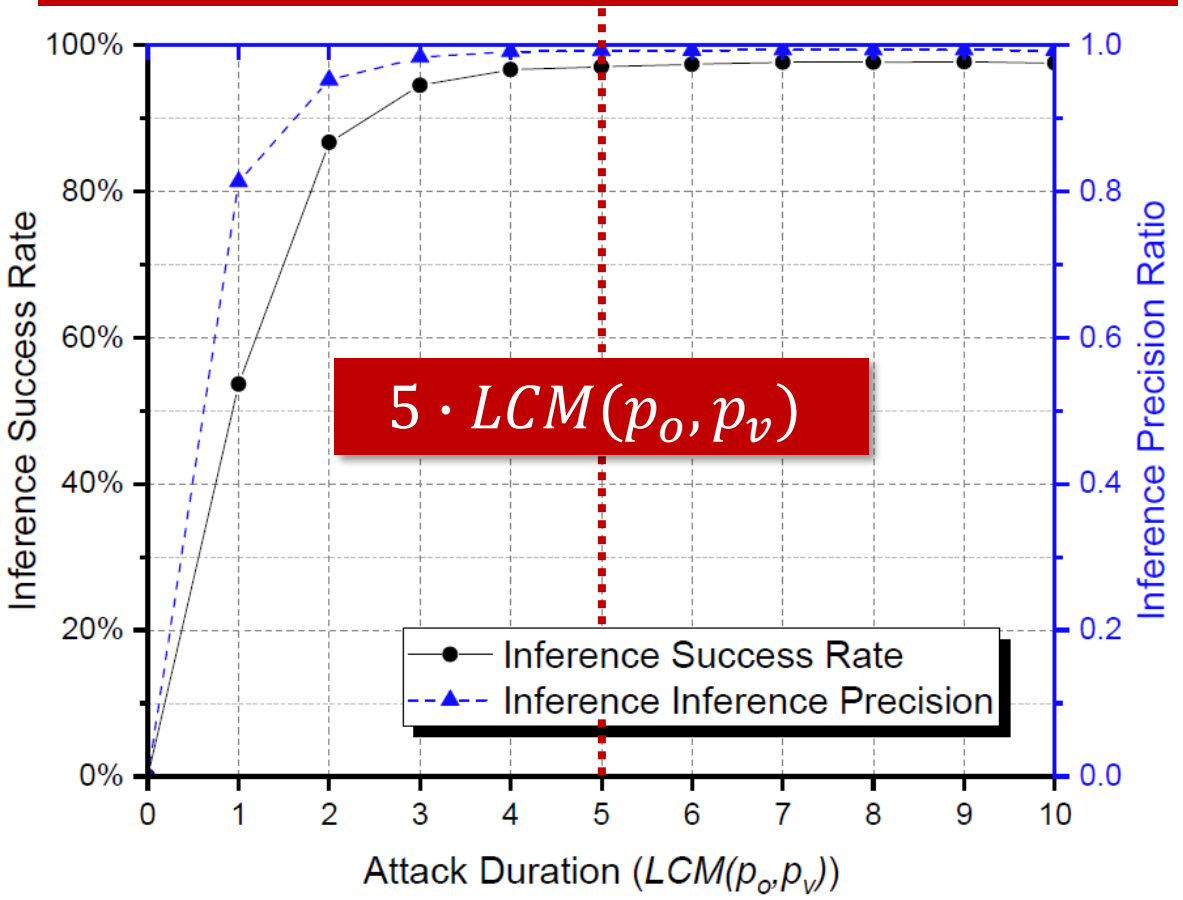
Attack Duration | Task Set Utilization | The Number of Tasks | Sporadic Task Ratio | Observer Task Priority | Coverage Ratio

Experiment Result Highlights



Experiment Result Highlights

Precision Ratio = 0.99, Success Rate = 97%



Precision Ratio

Success Rate

Attack Duration

The Number of Tasks

Coverage Ratio

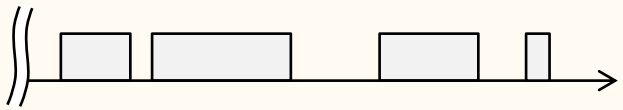
Sporadic Task Ratio

Task Set Utilization

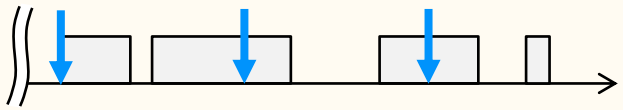
Observer Task Priority

Conclusion

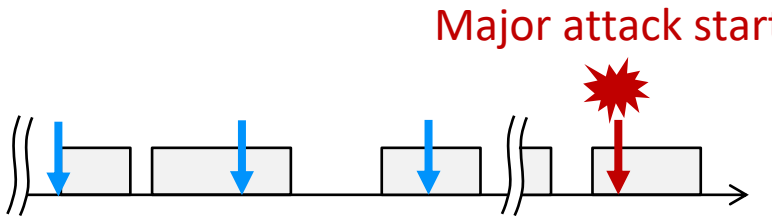
The attacker **observes** and **analyzes** the schedule on the victim system.



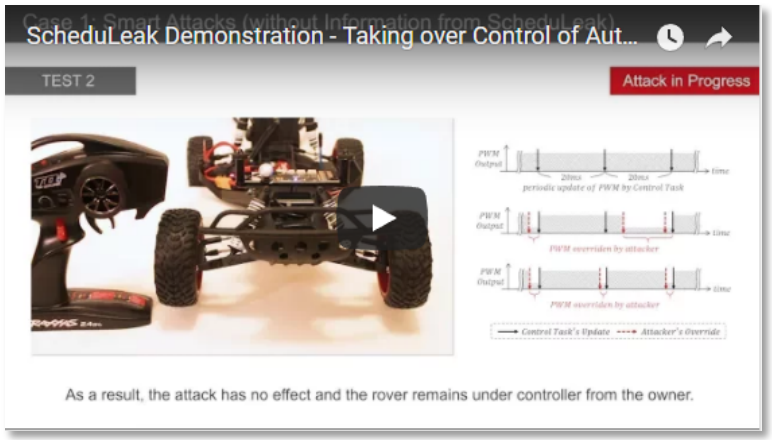
The attacker **infers** precise timing information of the victim task.

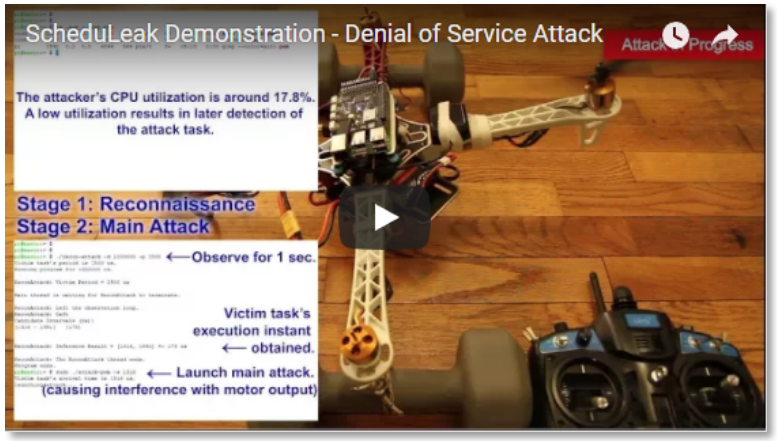


The attacker **launches a major attack** at a future instant that can cause the most amount of damage.



Thank you.





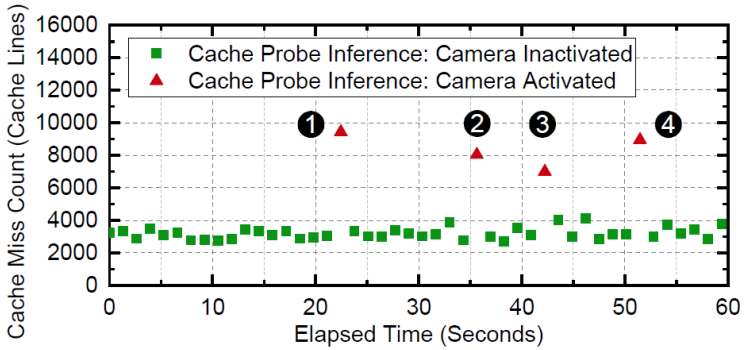
Demo videos are available at https://scheduleak.github.io/

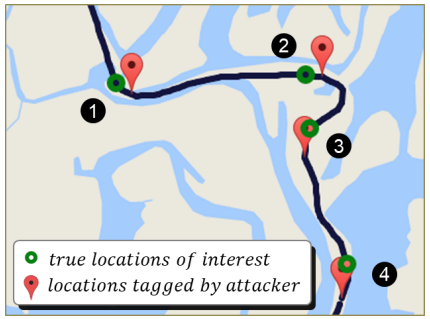
This work is supported by the National Science Foundation (NSF) under grant SaTC-1718952 and ONR N00014-13-1-0707.

Backup Materials

Implementation and Attack Case Studies

- Cache-timing side-channel attacks
  - FreeRTOS
  - Zedboard Xilinx Zynq®-7000
  - Hardware-in-the-loop UAV





- Interference with control (actuation signals) of CPS
  - Real-time Linux
  - Raspberry Pi 3 Model B
  - Ground rover/quadcopter

Coverage Ratio

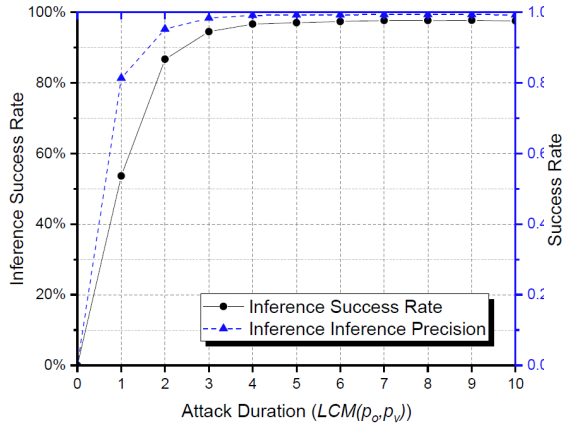
Coverage ratio is defined as

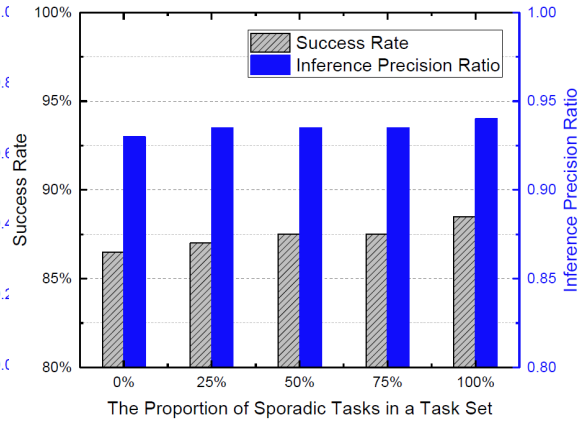
$$C(\tau_o, \tau_v) = \frac{e_o}{GCD(p_o, p_v)}$$

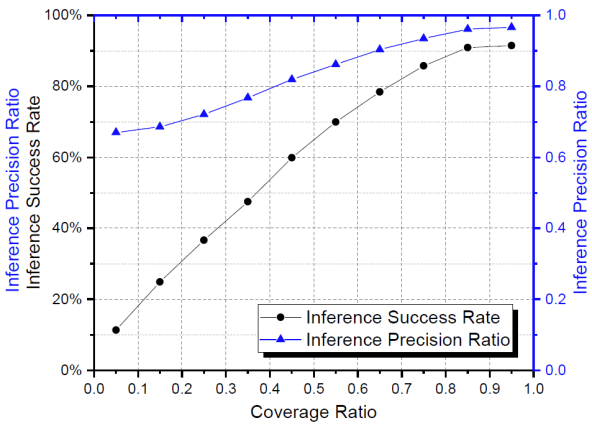
The coverage ratio can be loosely interpreted as the proportion of the time columns covered by the observer task in the schedule ladder diagram.

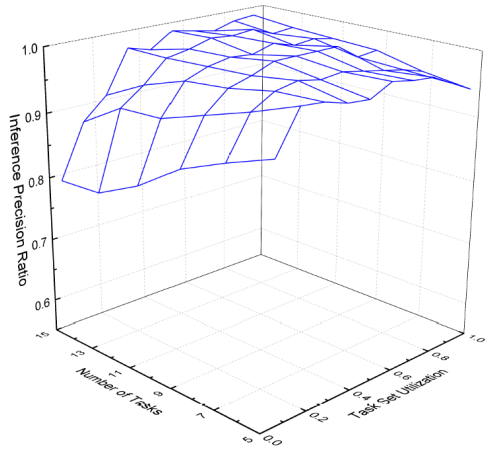
If $C(\tau_o, \tau_v) \geq 1$, then the attacker can observe the victim task. Otherwise it is not guaranteed that the victim task is observable by the attacker.

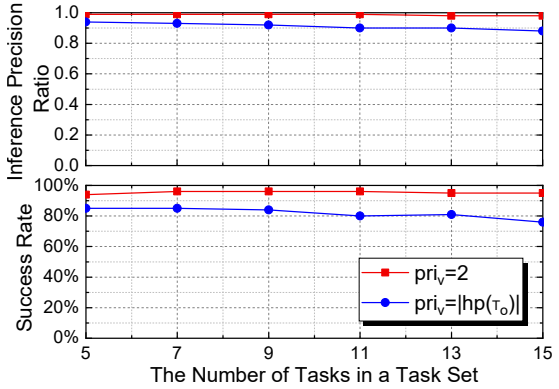
Experiment Results

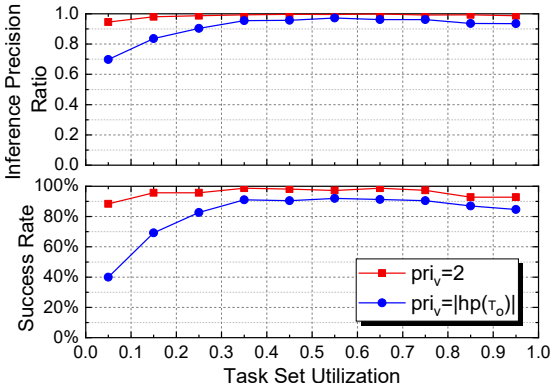






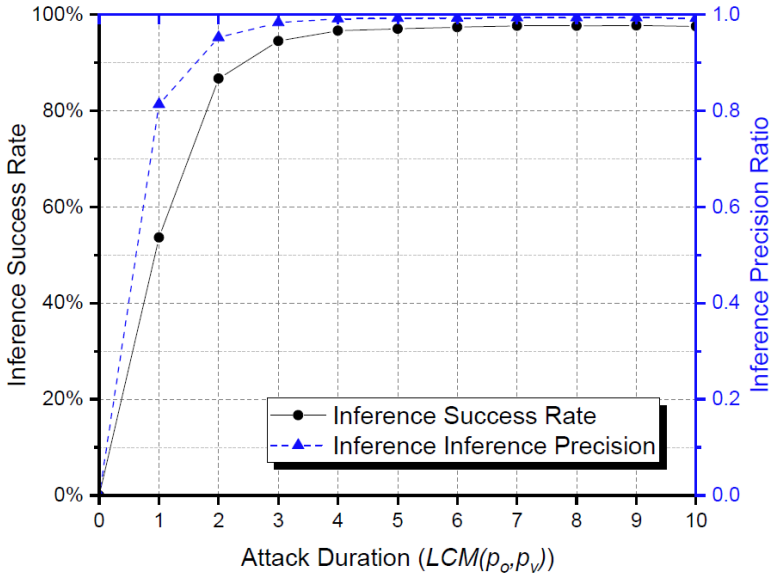






Experiment Results

Observation Duration



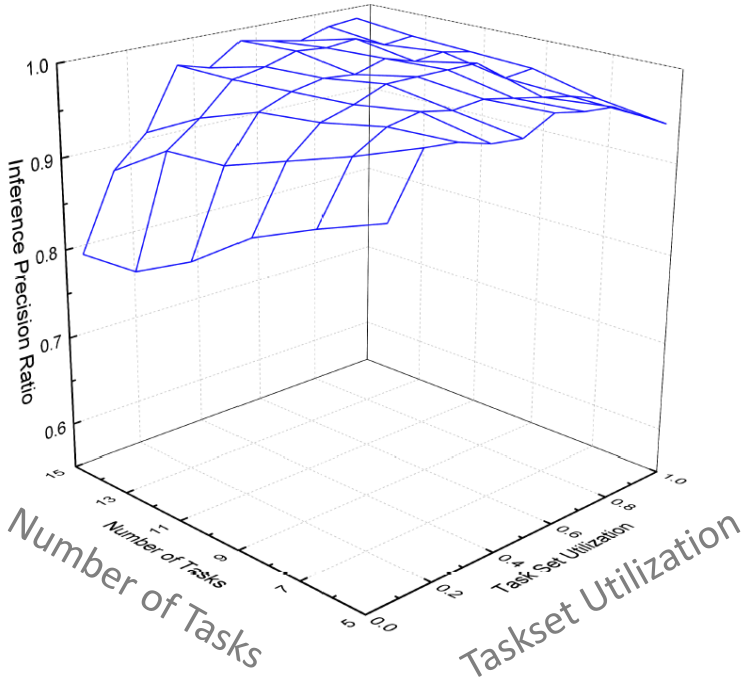
Success rate and precision ratio are stabilized after $5 \cdot LCM(p_o, p_v)$

- Success rate: 97%
- Precision ratio: 0.99

Note: each data point represents the mean of 12000 tasksets for the given observation duration.

Experiment Results

Number of Tasks & Taskset Utilization



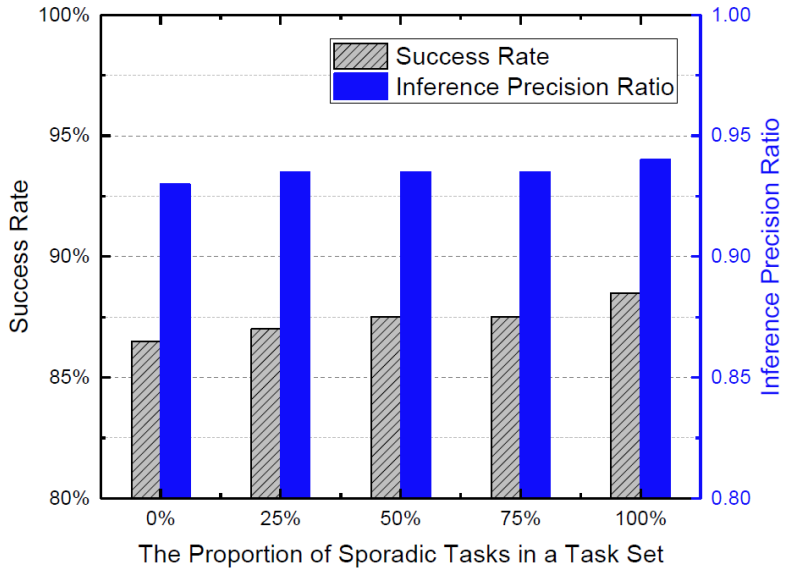
the inference precision ratio **decreases** as the number of tasks in a task set **increases**.

The inference precision ratio **increases** as the task set utilization **increases**.

Note: The drop is mainly caused by a high priority victim task with low task set utilization.
Observation duration is $10 \cdot LCM(p_o, p_v)$.

Experiment Results

Proportion of Sporadic Tasks



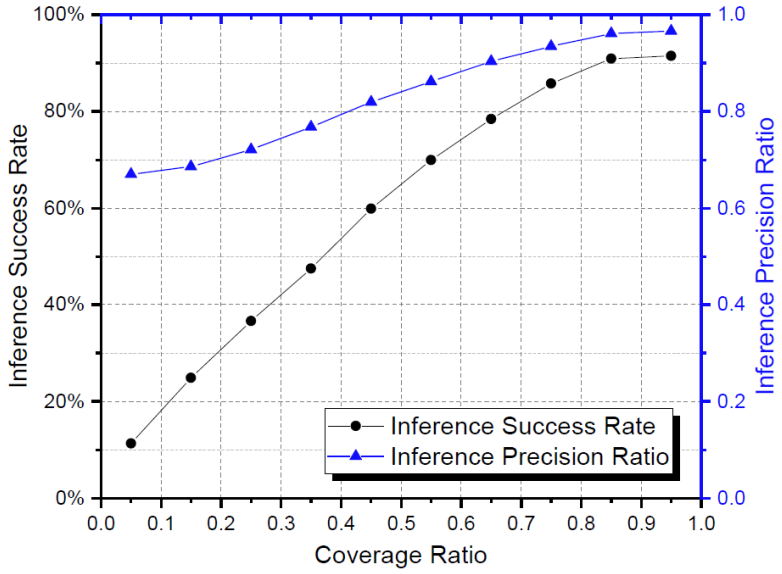
The algorithms perform better with sporadic tasks, with a **ascending trend** as the proportion of sporadic tasks **increases**.

The change in the performance is **less than 1%**, which is subtle.

Note: 0% means a taskset contains no sporadic task (all periodic tasks).
Observation duration is $10 \cdot LCM(p_o, p_v)$.

Experiment Results

Coverage Ratio



Higher coverage ratio yields better success rate and inference precision.

The success rate is about 59.9% (precision ratio is 0.819) when the coverage ratio is around 0.5.

Note: Each data point represents the mean of 12000 tasksets.
Observation duration is $10 \cdot LCM(p_o, p_v)$.