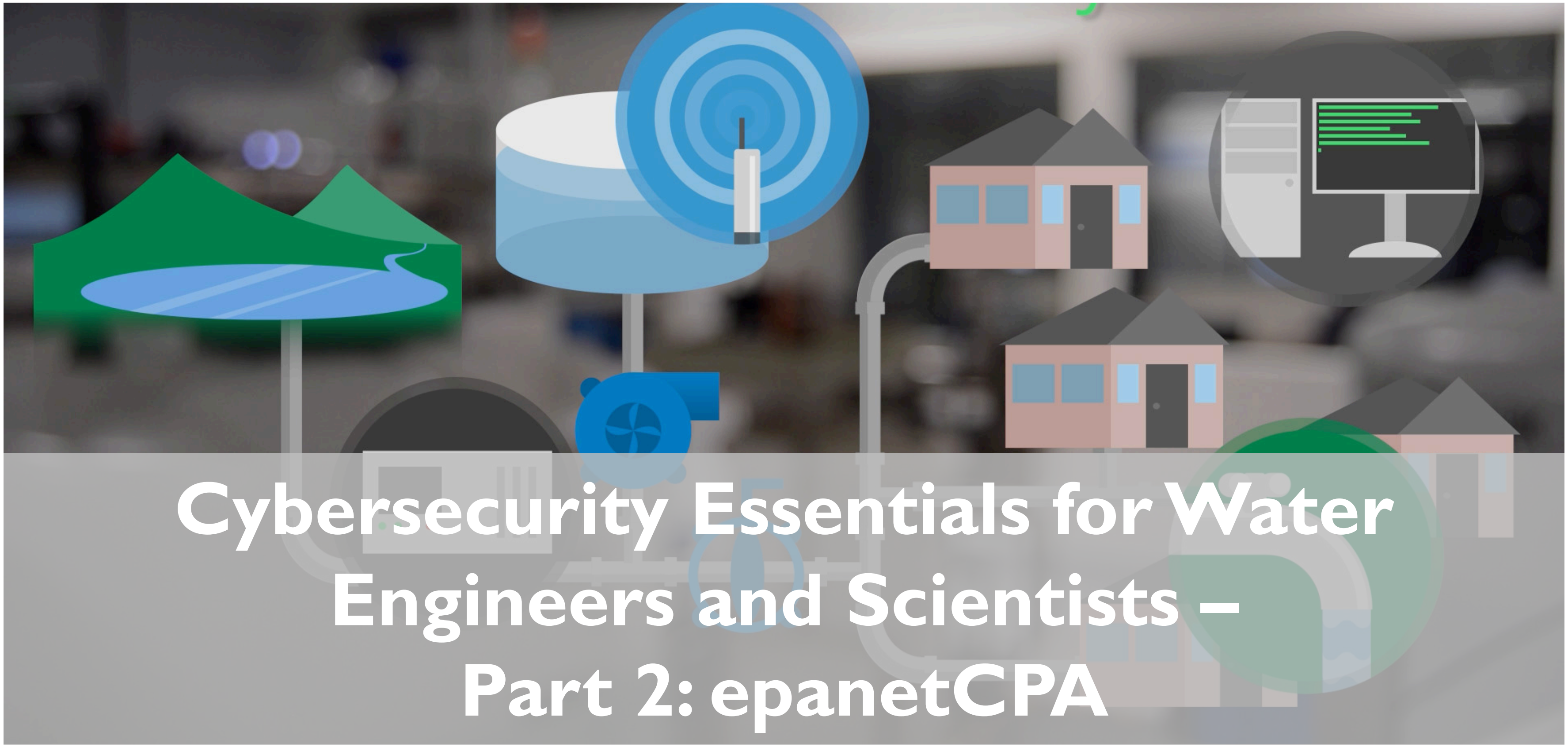
SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

@ EWRI 2018
June 3, 2018



# Cybersecurity Essentials for Water Engineers and Scientists – Part 2: epanetCPA

R. Taormina[1], S. Galelli[2]

[1] iTrust Centre for Research in Cyber Security, SUTD
[2] Pillar of Engineering Systems and Design, SUTD

iTrust
Centre for Research
in Cyber Security

# Outline

1. Intro to epanetCPA
2. Overview of the file system
3. Application examples
4. Exercises and free discussion

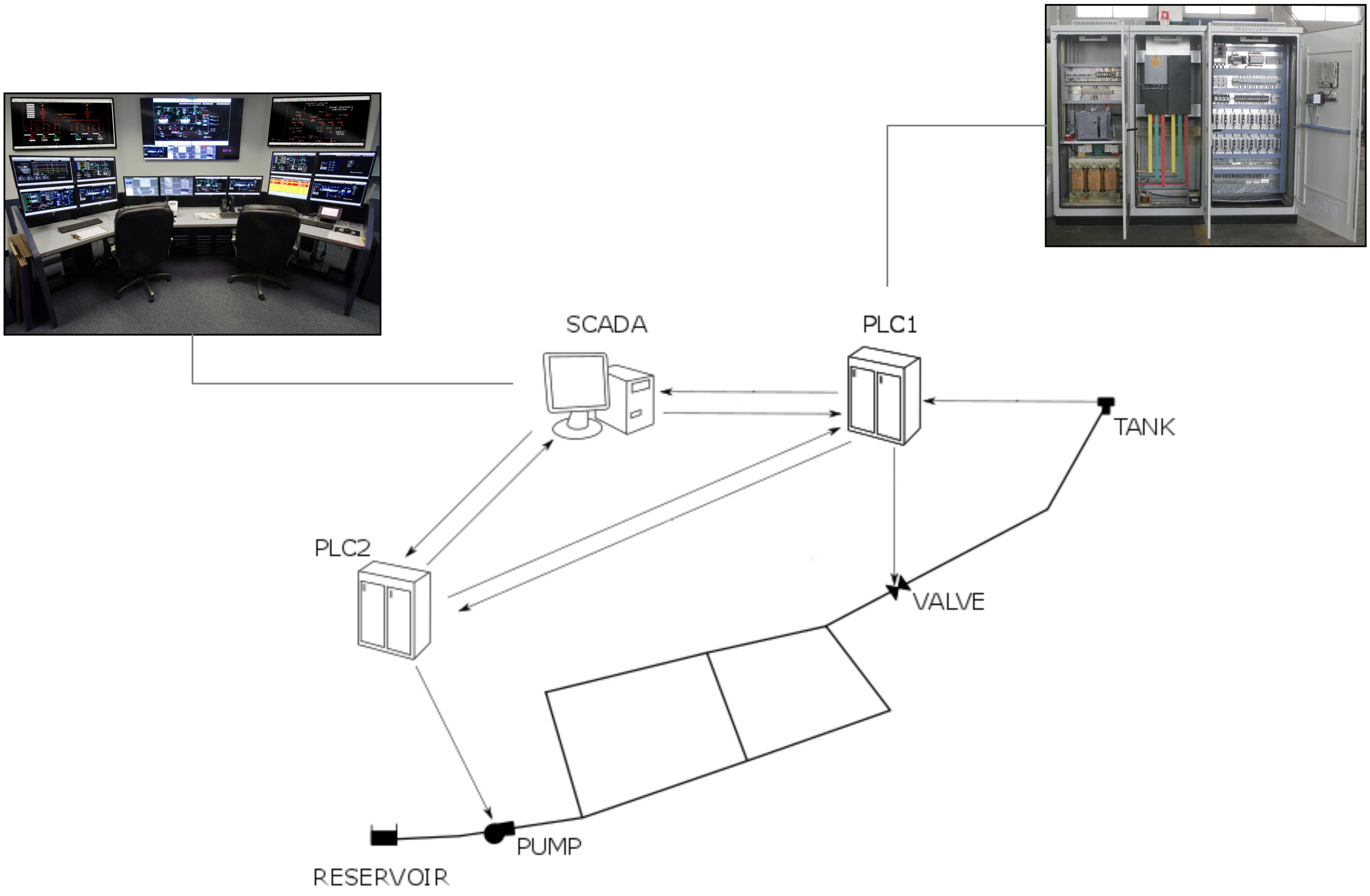
# 1. Intro to epanetCPA

**Key features**

- Allows to design cyber-physical attacks (type, duration, starting and initial condition etc.) and simulate their impacts via simulation with EPANET
- Suitable for both demand-driven and pressure-driven analysis
- Implementation in Matlab
- Open source (MIT License)
- https://github.com/rtaormina/epanetCPA

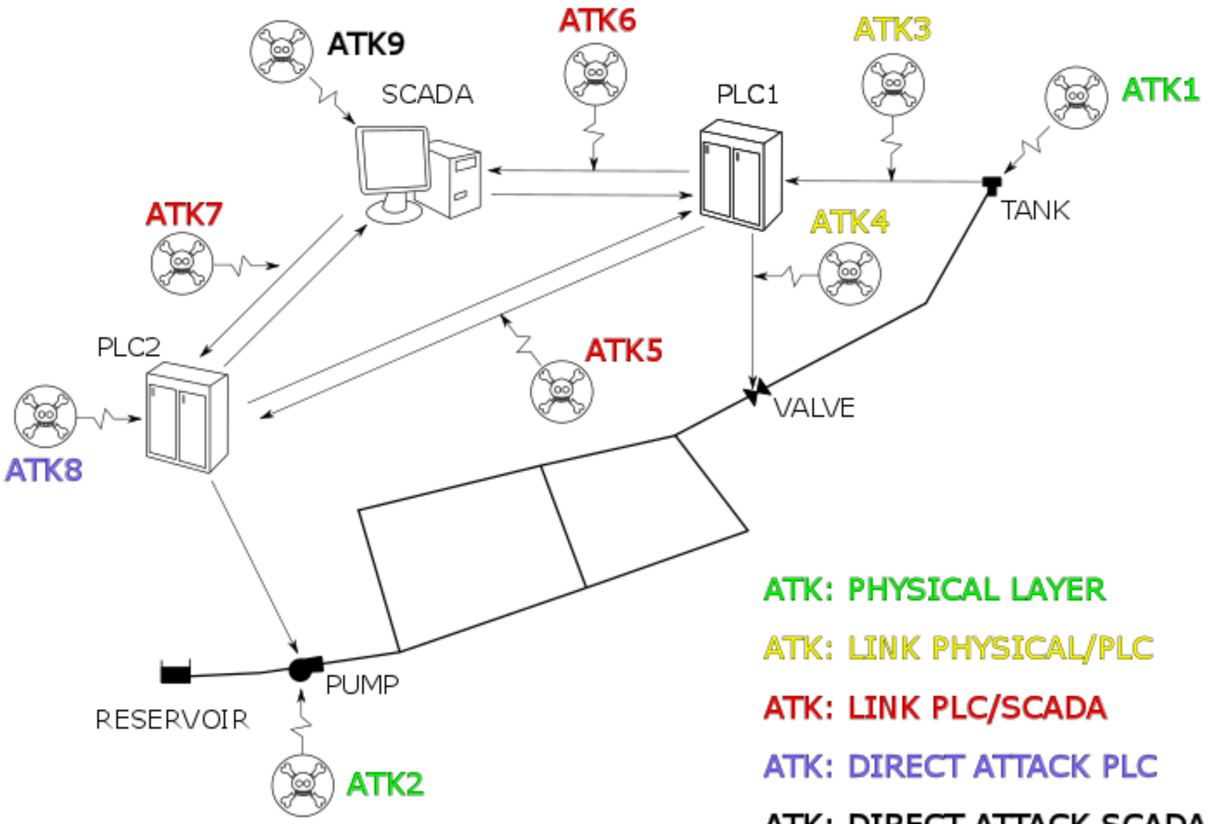
# 1. Intro to epanetCPA

## Interaction between cyber and physical layers



# 1. Intro to epanetCPA

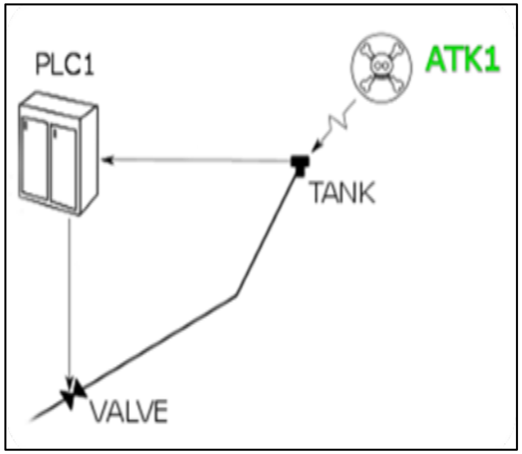
## Attack model



Taormina, R., Galelli, S., Tippenhauer, N.O., Salomons, E., Ostfeld, A. "Characterizing cyber-physical attacks on water distribution systems." *Journal of Water Resources Planning and Management* 143, no. 5 (2017): 04017009.

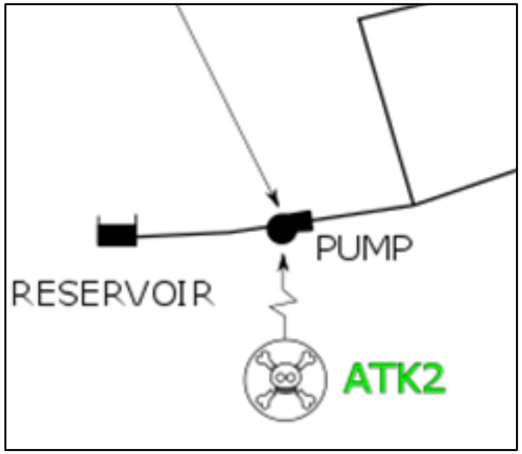
# 1. Intro to epanetCPA

### Attack model



**ATK1. Physical attack to a sensor**

- The attacker needs physical access to the sensor
- Sensor can be damaged, manipulated or replaced

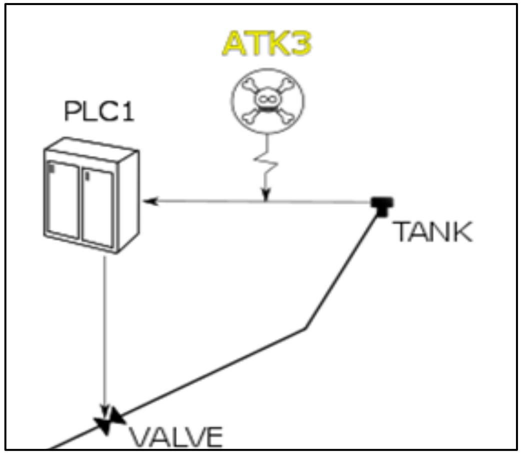


**ATK2. Physical attack to an actuator**

- The attacker needs physical access to the actuator
- The attacker can damage, deactivate/activate the actuator, or change its settings

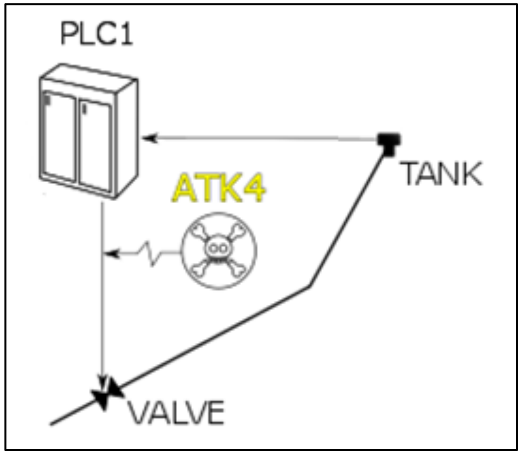
# 1. Intro to epanetCPA

**Attack model**



### ATK3. Attack to sensor-to-PLC link

- Link can be wireless connection or a hard-wire. The type of link determines whether the attacker needs physical access
- Actions: link interruption (DoS), manipulation of the data sent by sensor (deception), eavesdropping

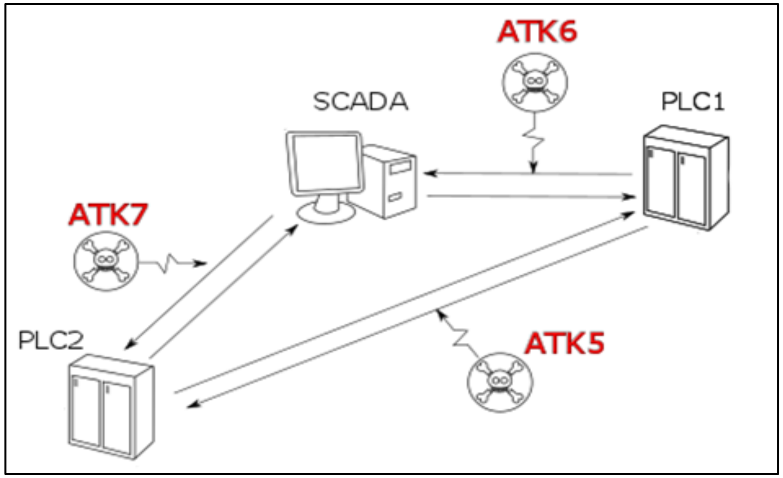


### ATK4. Attack to PLC-to-actuator link

- The considerations made for ATK3 regarding the nature of the connection link still hold
- Actions: link interruption (DoS), manipulation of control signals (deception), eavesdropping

# 1. Intro to epanetCPA

### Attack model



**ATK5. Attack to PLC-to-PLC link**

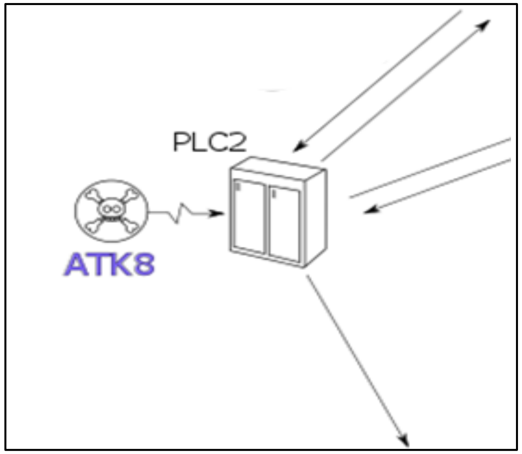
**ATK6. Attack to PLC-to-SCADA link**

**ATK7. Attack to SCADA-to-PLC link**

- Elements connected through a private network or internet.
- Actions: attacker can intercept the connection to eavesdrop or manipulate its content (deception), flood the communication channel with traffic (DoS), …

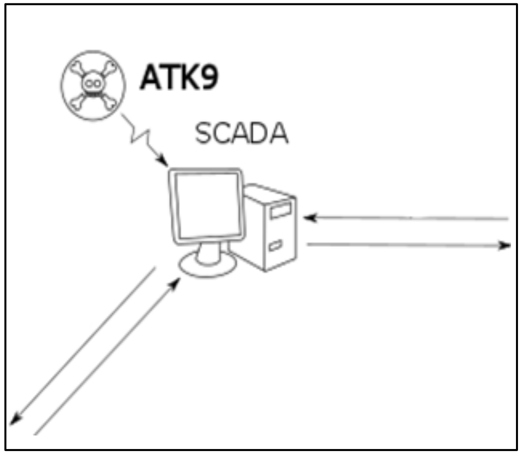
# 1. Intro to epanetCPA

## Attack model



### ATK8. Attack to PLC

- The adversary gains control of a PLC in the network
- Can perform DoS, deception and eavesdropping

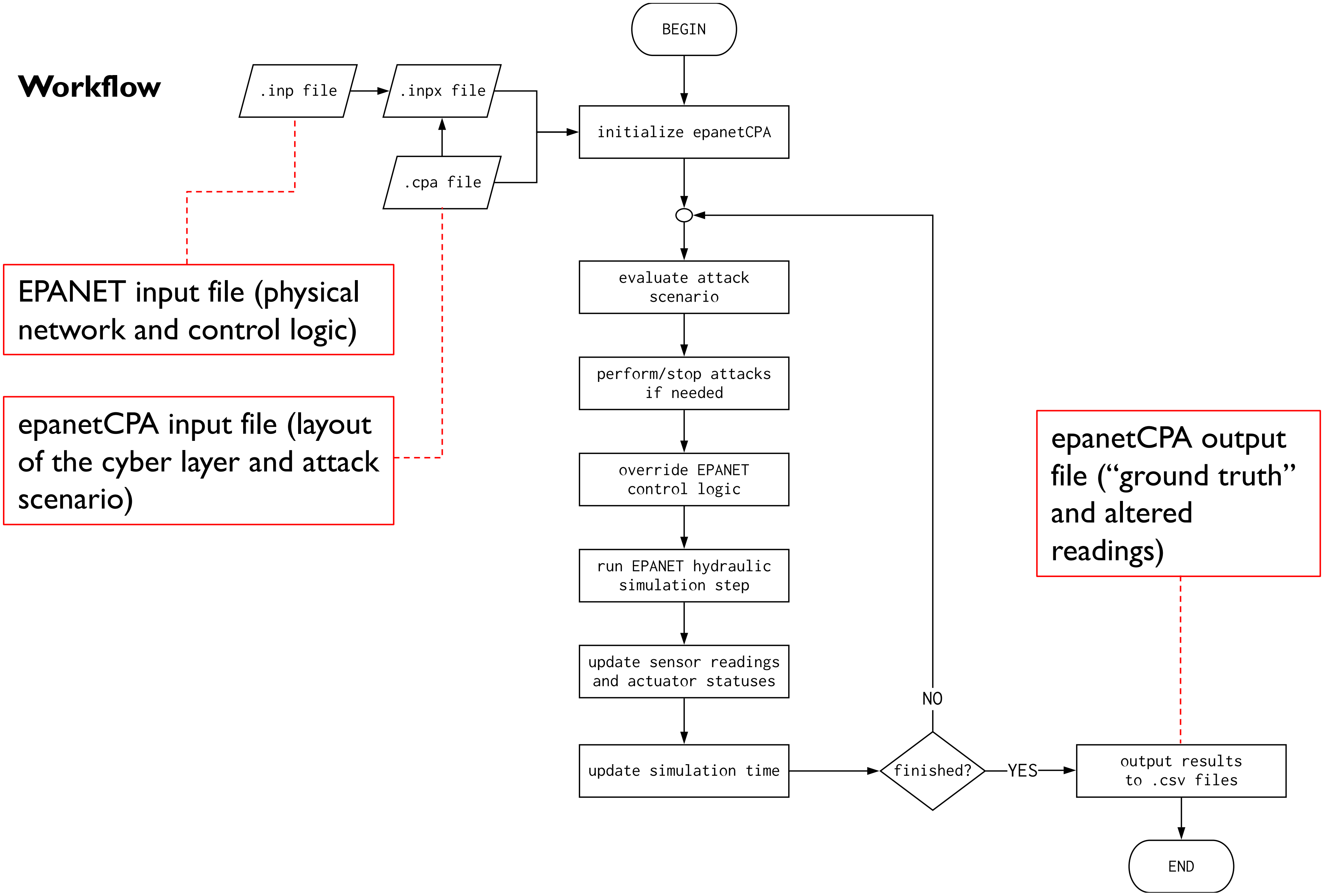


### ATK9. Attack to SCADA

- The adversary gains control of SCADA either via local or remote attack

# 1. Intro to epanetCPA

## Workflow


BEGIN
.inp file
.inpx file
.cpa file
initialize epanetCPA
evaluate attack scenario
perform/stop attacks if needed
override EPANET control logic
run EPANET hydraulic simulation step
update sensor readings and actuator statuses
update simulation time
finished?
NO
YES
output results to .csv files
END

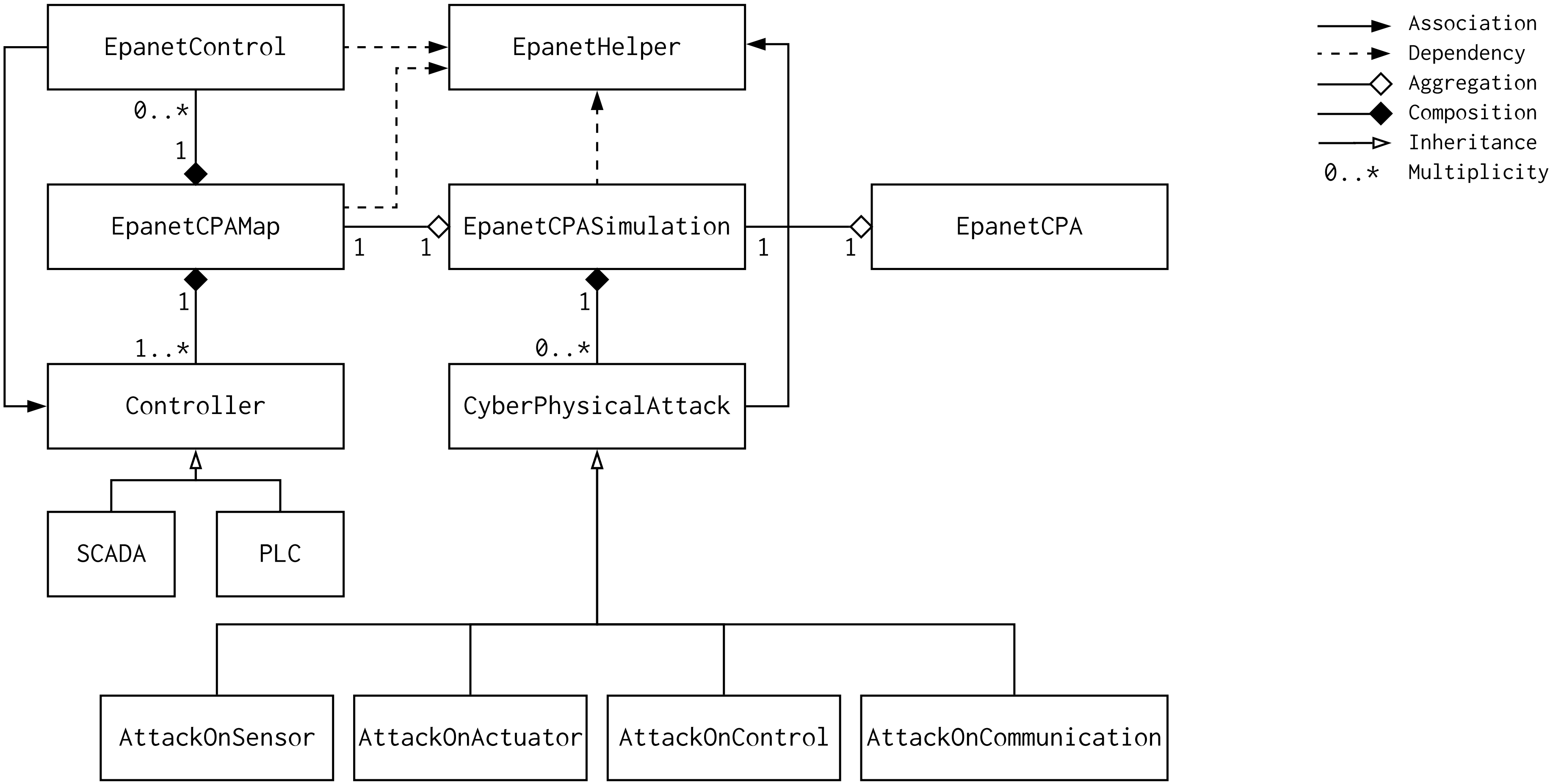
EPANET input file (physical network and control logic)

epanetCPA input file (layout of the cyber layer and attack scenario)

epanetCPA output file ("ground truth" and altered readings)

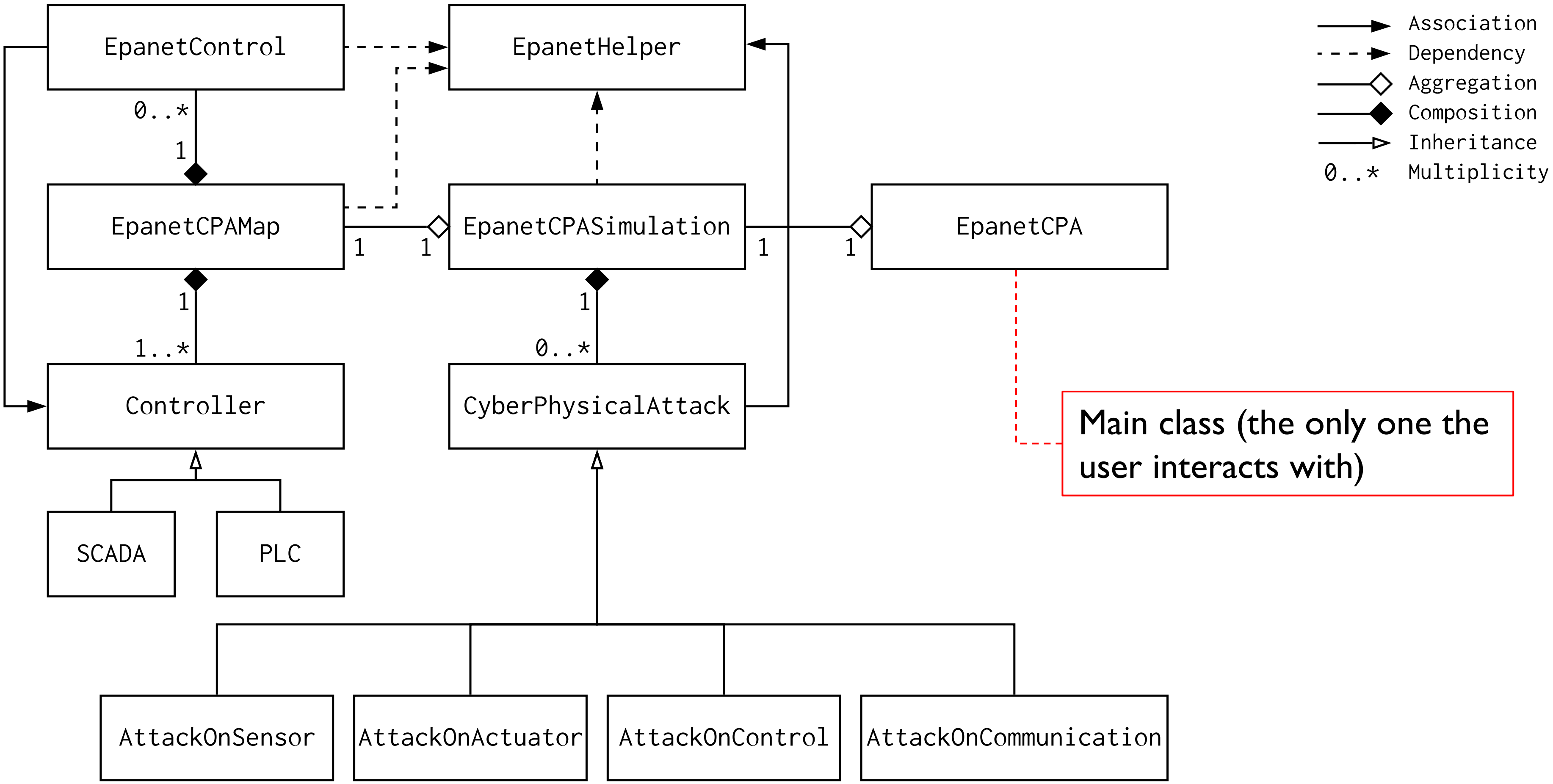
# 1. Intro to epanetCPA

## Structure



# 1. Intro to epanetCPA

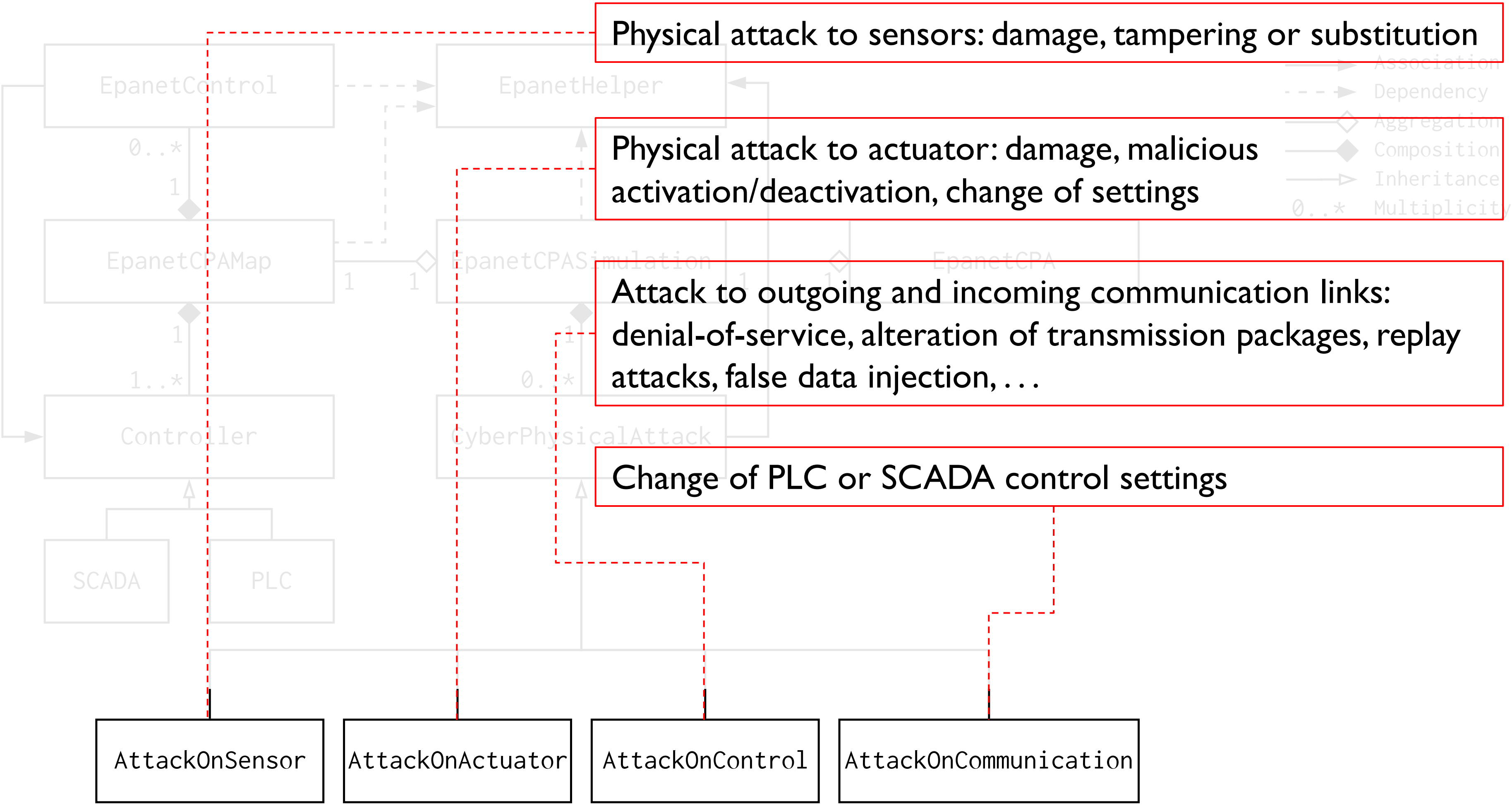
## Structure



Main class (the only one the user interacts with)

# 1. Intro to epanetCPA

## Structure



Physical attack to sensors: damage, tampering or substitution

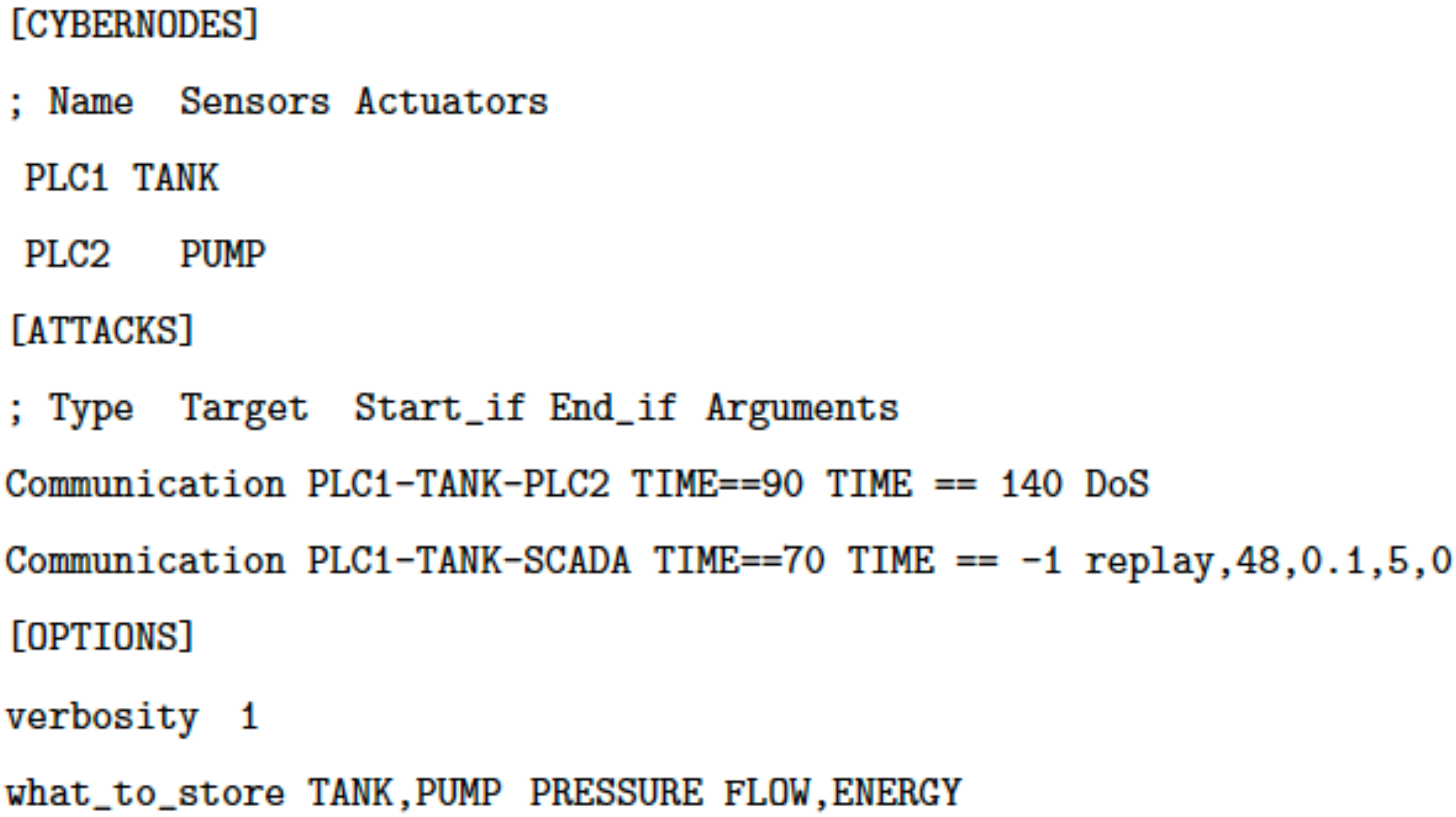
Physical attack to actuator: damage, malicious activation/deactivation, change of settings

Attack to outgoing and incoming communication links: denial-of-service, alteration of transmission packages, replay attacks, false data injection, . . .

Change of PLC or SCADA control settings

# 1. Intro to epanetCPA

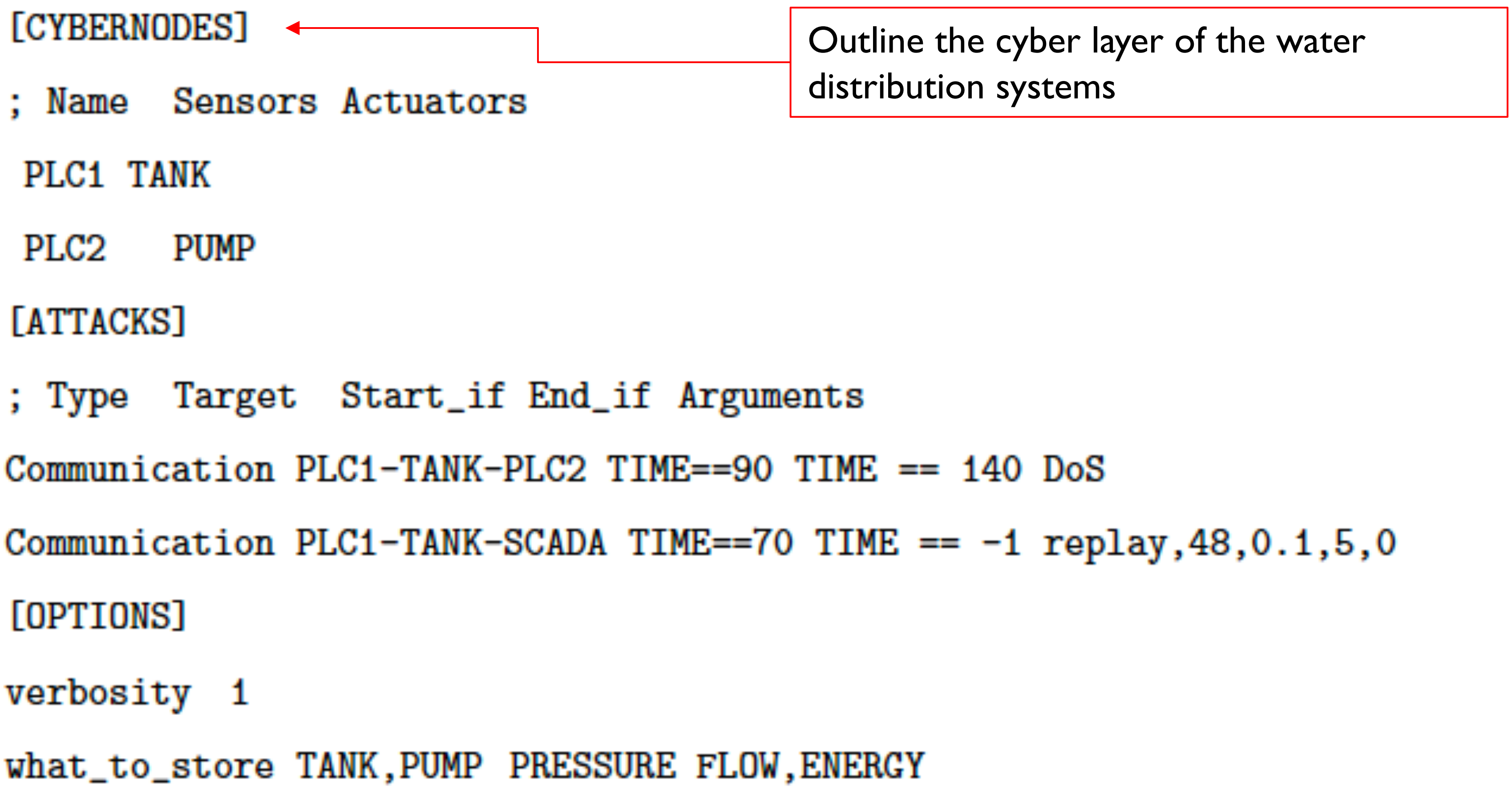
**The .cpa input file**



```
[CYBERNODES]
; Name   Sensors Actuators
 PLC1 TANK
 PLC2    PUMP
[ATTACKS]
; Type   Target  Start_if End_if Arguments
Communication PLC1-TANK-PLC2 TIME==90 TIME == 140 DoS
Communication PLC1-TANK-SCADA TIME==70 TIME == -1 replay,48,0.1,5,0
[OPTIONS]
verbosity  1
what_to_store TANK,PUMP  PRESSURE FLOW,ENERGY
```

# 1. Intro to epanetCPA

**The .cpa input file**

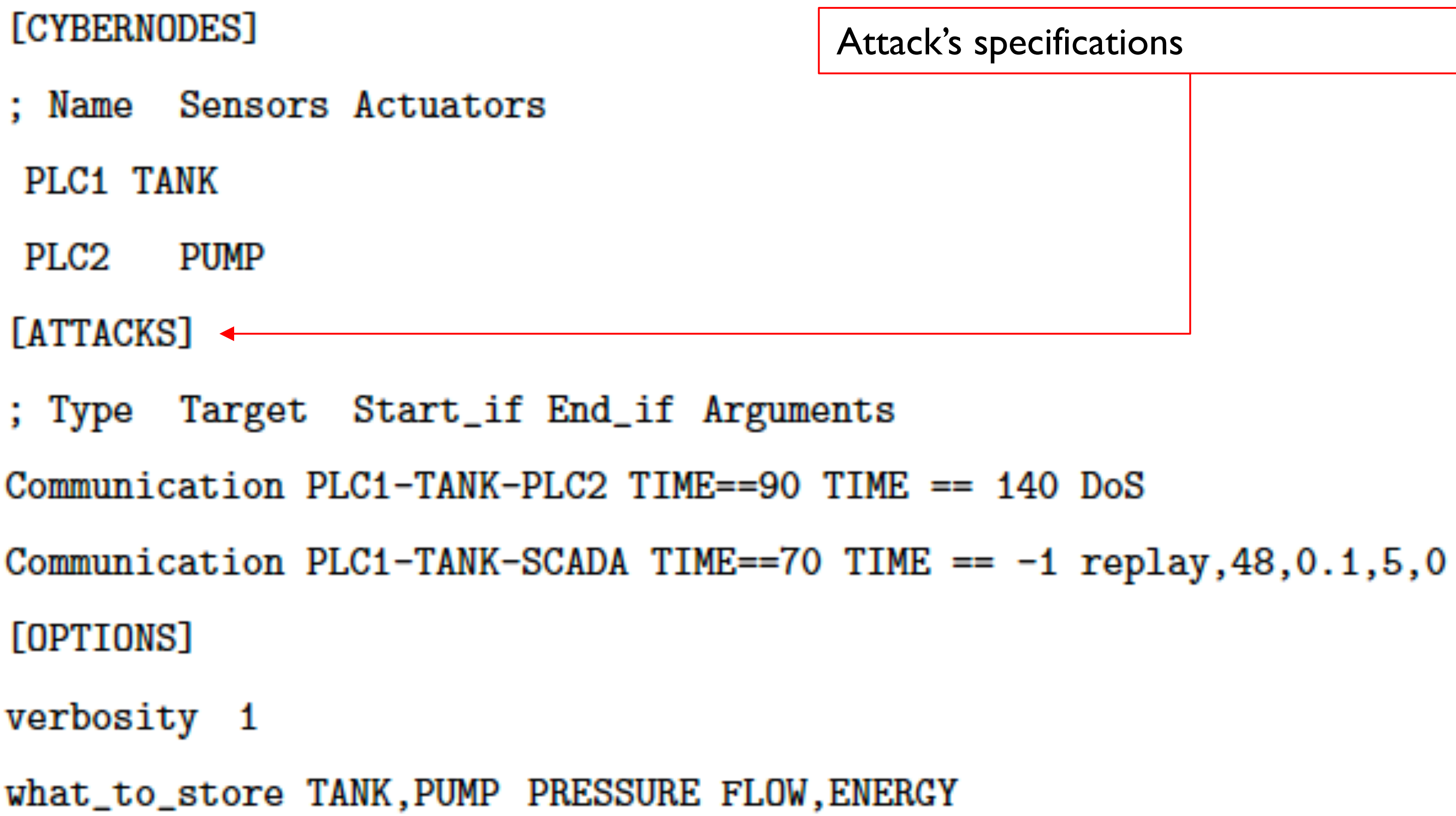


```
[CYBERNODES]
; Name  Sensors Actuators
 PLC1 TANK
 PLC2   PUMP
[ATTACKS]
; Type  Target  Start_if End_if Arguments
Communication PLC1-TANK-PLC2 TIME==90 TIME == 140 DoS
Communication PLC1-TANK-SCADA TIME==70 TIME == -1 replay,48,0.1,5,0
[OPTIONS]
verbosity  1
what_to_store TANK,PUMP  PRESSURE FLOW,ENERGY
```

Outline the cyber layer of the water distribution systems

# 1. Intro to epanetCPA

**The .cpa input file**

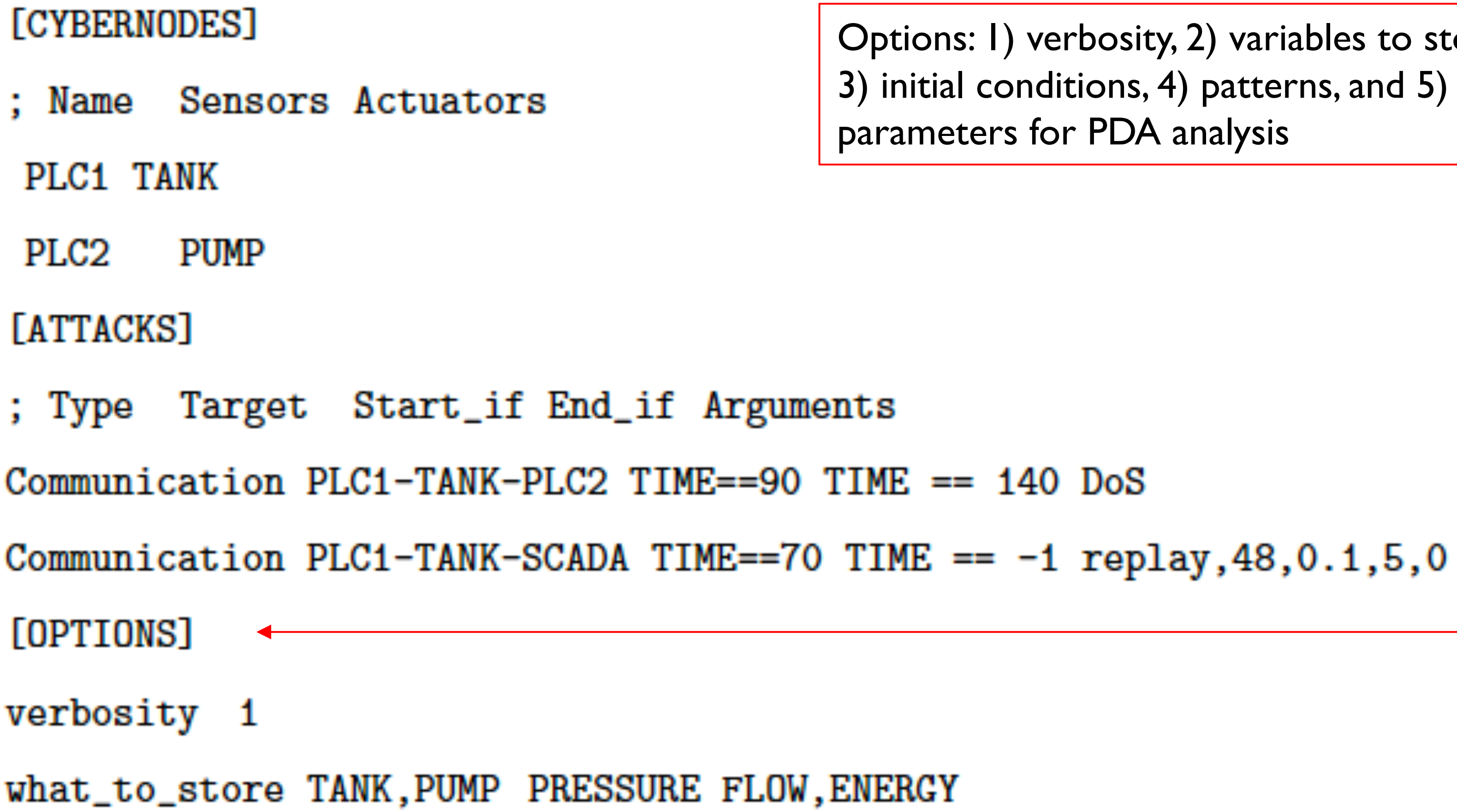


```
[CYBERNODES]
; Name   Sensors Actuators
 PLC1 TANK
 PLC2    PUMP
[ATTACKS]
; Type   Target   Start_if End_if Arguments
Communication PLC1-TANK-PLC2 TIME==90 TIME == 140 DoS
Communication PLC1-TANK-SCADA TIME==70 TIME == -1 replay,48,0.1,5,0
[OPTIONS]
verbosity  1
what_to_store TANK,PUMP  PRESSURE FLOW,ENERGY
```

Attack's specifications

# 1. Intro to epanetCPA

**The .cpa input file**



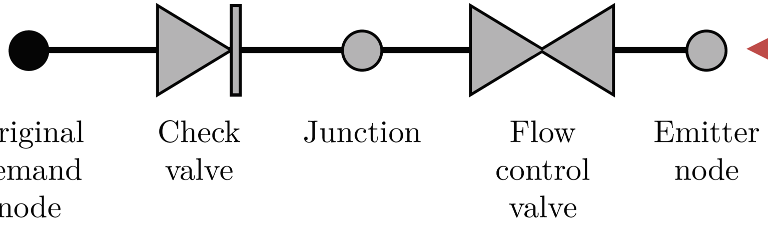
```
[CYBERNODES]
; Name   Sensors Actuators
 PLC1  TANK
 PLC2     PUMP
[ATTACKS]
; Type   Target   Start_if End_if  Arguments
Communication PLC1-TANK-PLC2 TIME==90 TIME == 140 DoS
Communication PLC1-TANK-SCADA TIME==70 TIME == -1 replay,48,0.1,5,0
[OPTIONS]
verbosity  1
what_to_store TANK,PUMP  PRESSURE FLOW,ENERGY
```

Options: 1) verbosity, 2) variables to store, 3) initial conditions, 4) patterns, and 5) parameters for PDA analysis

# 1. Intro to epanetCPA

### The .cpa input file

The PDA analysis is based on the approach of Abdy Sayyed et al., 2015 (“Noniterative application of EPANET for pressure dependent modelling of water distribution systems”, *Water Resources Management*, 29(9), 3227-3242)



Demand set to 0

$$q = C(p)^{\gamma}$$

FCV setting = actual demand at time *t*

Function of actual demand and chosen HFR

```
[CYBERNODES]
; [illegible] Sensors Actuators
 PLC1 TANK
 PLC2 [illegible]
[ATTACKS]
; Type  Target
Communication PLC1-TANK-PLC2 TIME==90 TIME == 140 DoS
Communication PLC1-TANK-SCADA TIME==70 TIME == -1 repl[illegible]
[OPTIONS]
verbosity  1
what_to_store TANK,PUMP  PRESSURE FLOW,ENERGY
pda_options 0.5  0  20 Wagner
```

# 1. Intro to epanetCPA

**The .inpx input file**

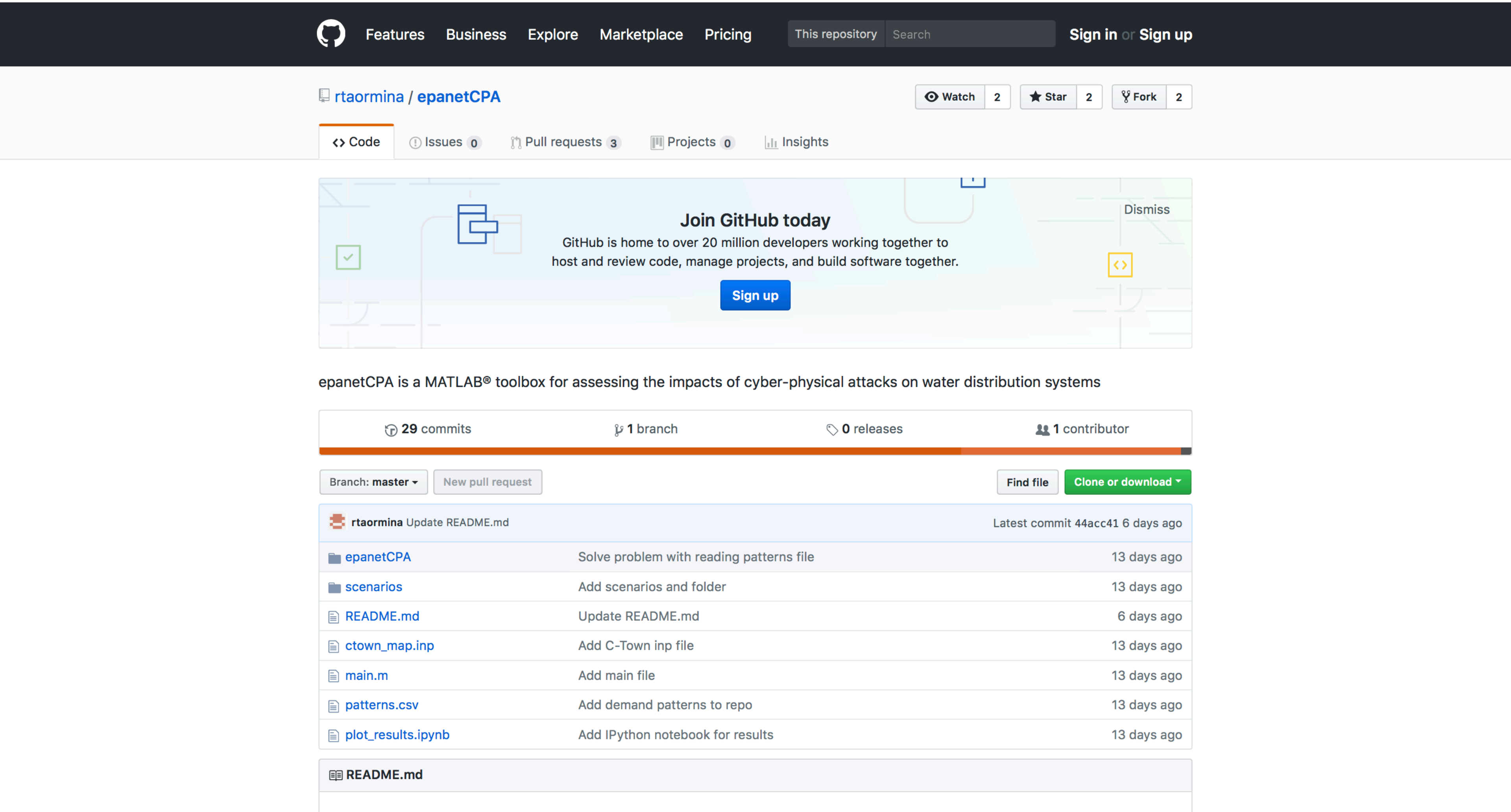
- epanetCPA modifies the .inp file and creates the (augmented) .inpx file
- The file contains some extra dummy controls to override the control logic when attacks are in place
- Modifications to the map to allow the simulation of tank overflows and PDA analysis
- It also features user-specified initial tank levels and demand patterns if the initial conditions and patterns options are specified to overwrite the original values contained in the .inp file

# 1. Intro to epanetCPA

**The output files**

- Comma-separated files
- One file contains the "ground truth", i.e., the actual values of the variables at the physical layer
- If the simulation features attacks that manipulate sensor readings, the toolbox outputs an additional file to track the false information sent to these cyber-nodes. Each entry in this tabular file has five attributes that contain:
  - The timestamp at which the alteration occurred
  - The layer affected by the alteration (a PLC identifier, SCADA, or the physical layer in case of sensor damage or replacement)
  - The sensor being altered
  - The value of the altered reading
  - The EPANET variable being altered

# 2. Overview of the file system



rtaormina / epanetCPA

## Join GitHub today

GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.

epanetCPA is a MATLAB® toolbox for assessing the impacts of cyber-physical attacks on water distribution systems

29 commits | 1 branch | 0 releases | 1 contributor

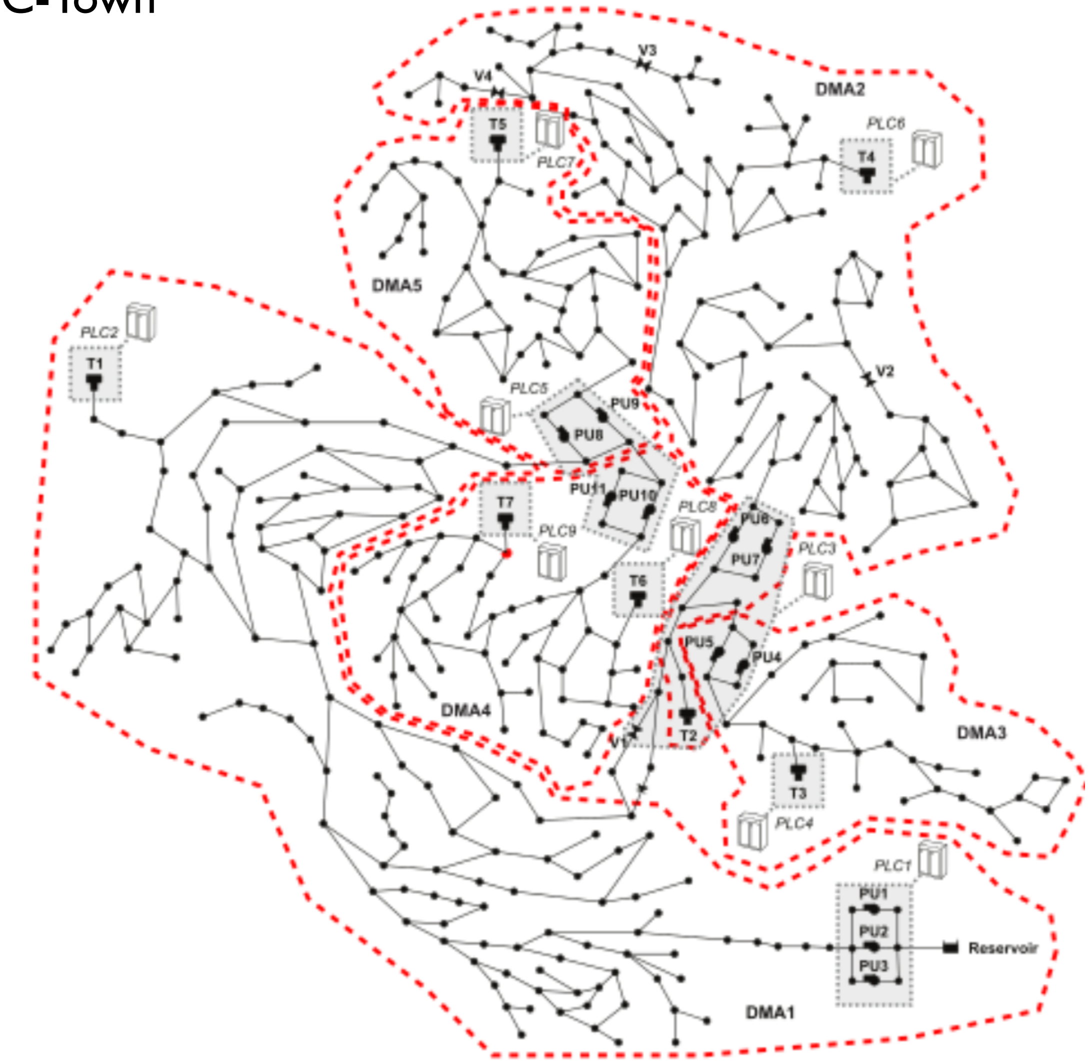
**rtaormina** Update README.md — Latest commit 44acc41 6 days ago

| | | |
|---|---|---|
| epanetCPA | Solve problem with reading patterns file | 13 days ago |
| scenarios | Add scenarios and folder | 13 days ago |
| README.md | Update README.md | 6 days ago |
| ctown_map.inp | Add C-Town inp file | 13 days ago |
| main.m | Add main file | 13 days ago |
| patterns.csv | Add demand patterns to repo | 13 days ago |
| plot_results.ipynb | Add IPython notebook for results | 13 days ago |

**README.md**

# 3. Application examples

C-Town



**C-Town network**

- Junctions, 388
- Pipes, 429
- Tanks, 7
- Pumps, 11
- Valves, 4

- PLCs, 9
- SCADA

# 3. Application examples

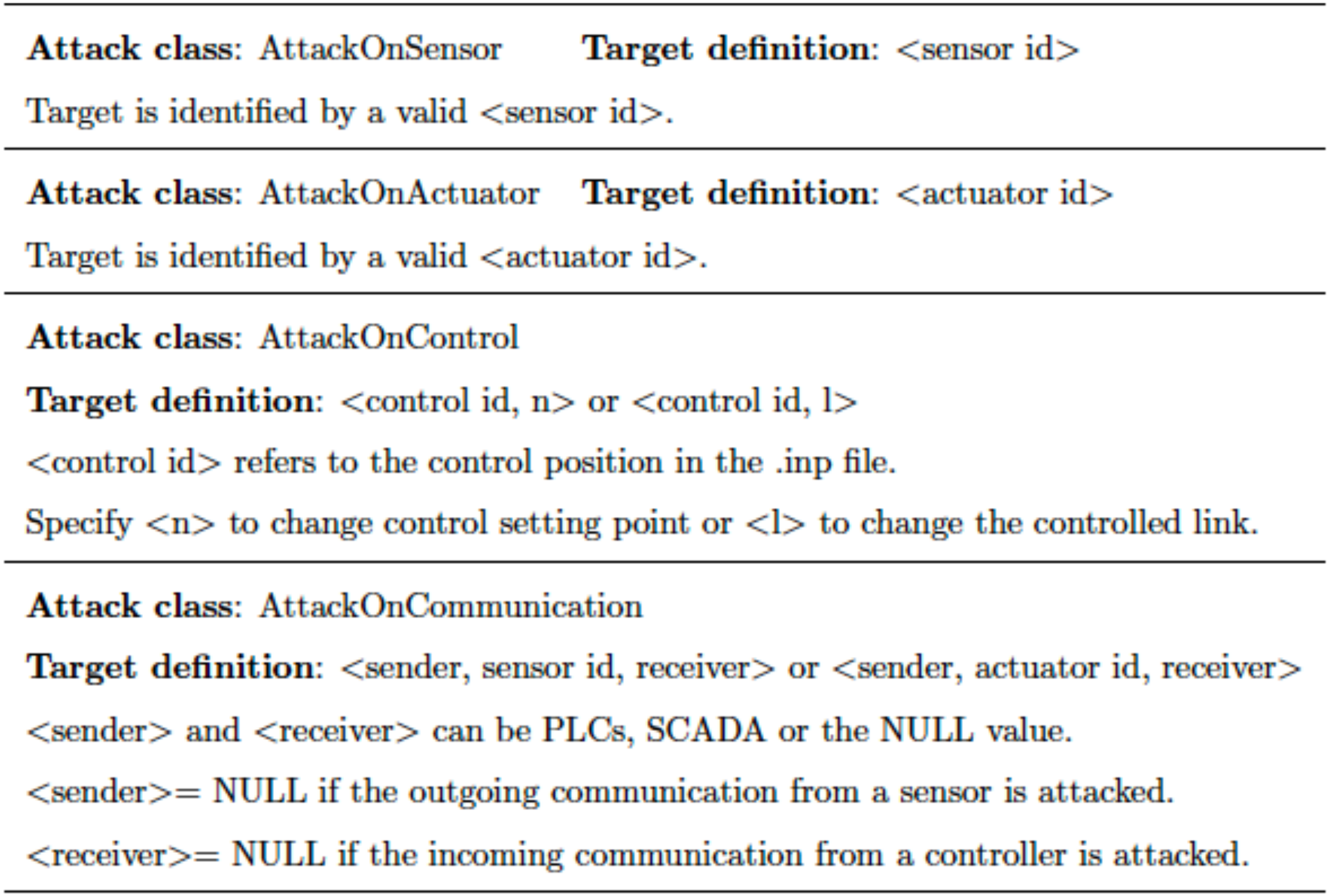
Attacks:

- *scenario01.cpa* Manipulation of sensor readings arriving to PLC3. The attacker shows that tank T2 is full. The PLC closes valve V2, thus preventing the flow to reach the tank and disconnecting part of the network.
- *scenario02.cpa* Same as scenario01 but run using the pressure driven engine to obtain more reliable results.
- *scenario03.cpa* The attacker modifies the control logic of PLC5 so that some of the controlled pumps (PU10, PU11) switch on/off intermittently.
- *scenario04.cpa* Denial-of-service of the connection link between PLC2 and PLC1. PLC1 fails to receive updated readings of T1 water level and keeps the pumps (PU1,PU2) ON. This causes a surge in the tank T1.
- *scenario05.cpa* Same as scenario04 but this time the attacker conceals the tanks surge from SCADA by altering the data sent by PLC2 to SCADA.

# Appendix

# Appendix 1

Target identification for epanetCPA attack classes



**Attack class**: AttackOnSensor **Target definition**: <sensor id>

Target is identified by a valid <sensor id>.

**Attack class**: AttackOnActuator **Target definition**: <actuator id>

Target is identified by a valid <actuator id>.

**Attack class**: AttackOnControl

**Target definition**: <control id, n> or <control id, l>

<control id> refers to the control position in the .inp file.

Specify <n> to change control setting point or <l> to change the controlled link.

**Attack class**: AttackOnCommunication

**Target definition**: <sender, sensor id, receiver> or <sender, actuator id, receiver>

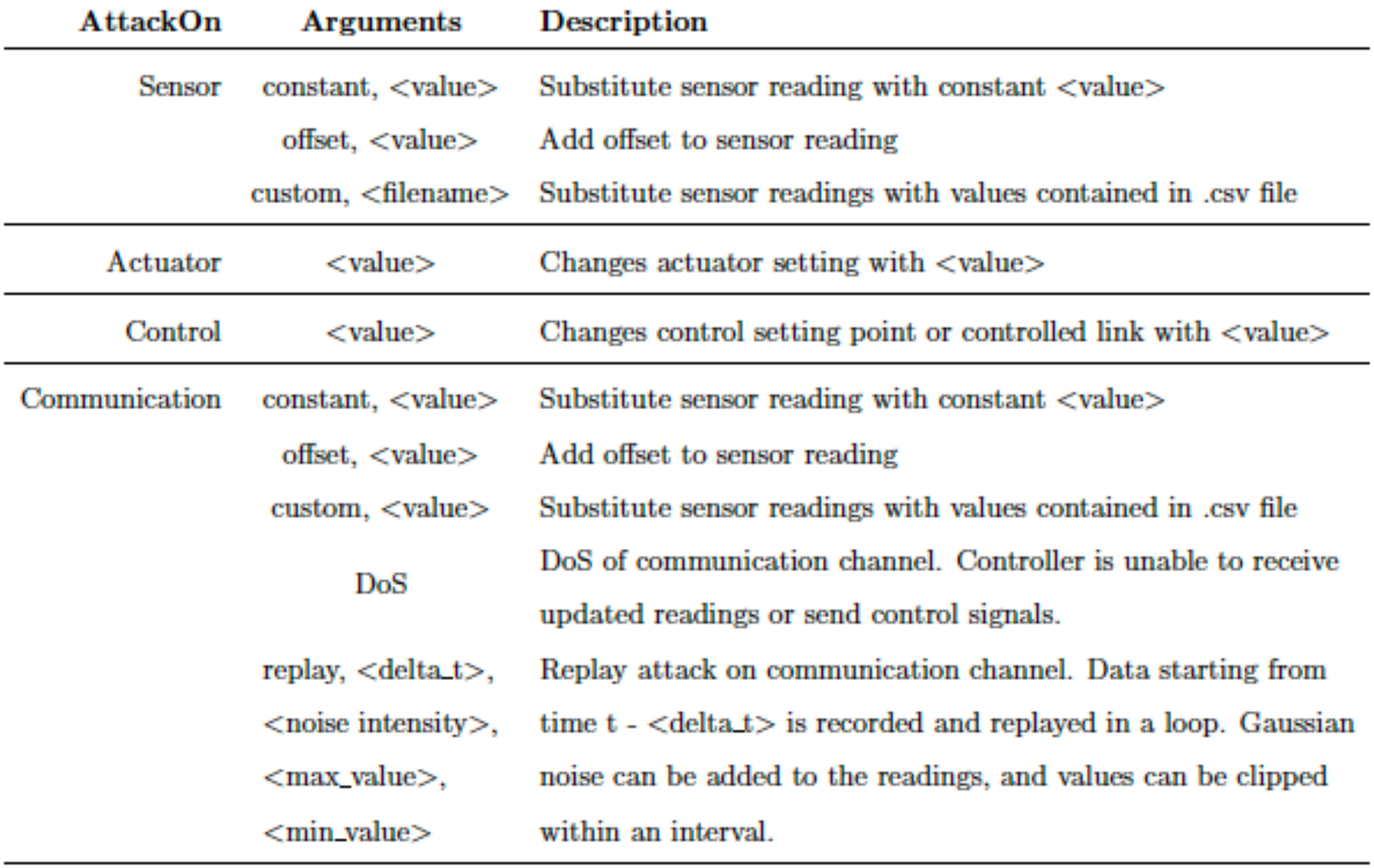
<sender> and <receiver> can be PLCs, SCADA or the NULL value.

<sender>= NULL if the outgoing communication from a sensor is attacked.

<receiver>= NULL if the incoming communication from a controller is attacked.

# Appendix 2

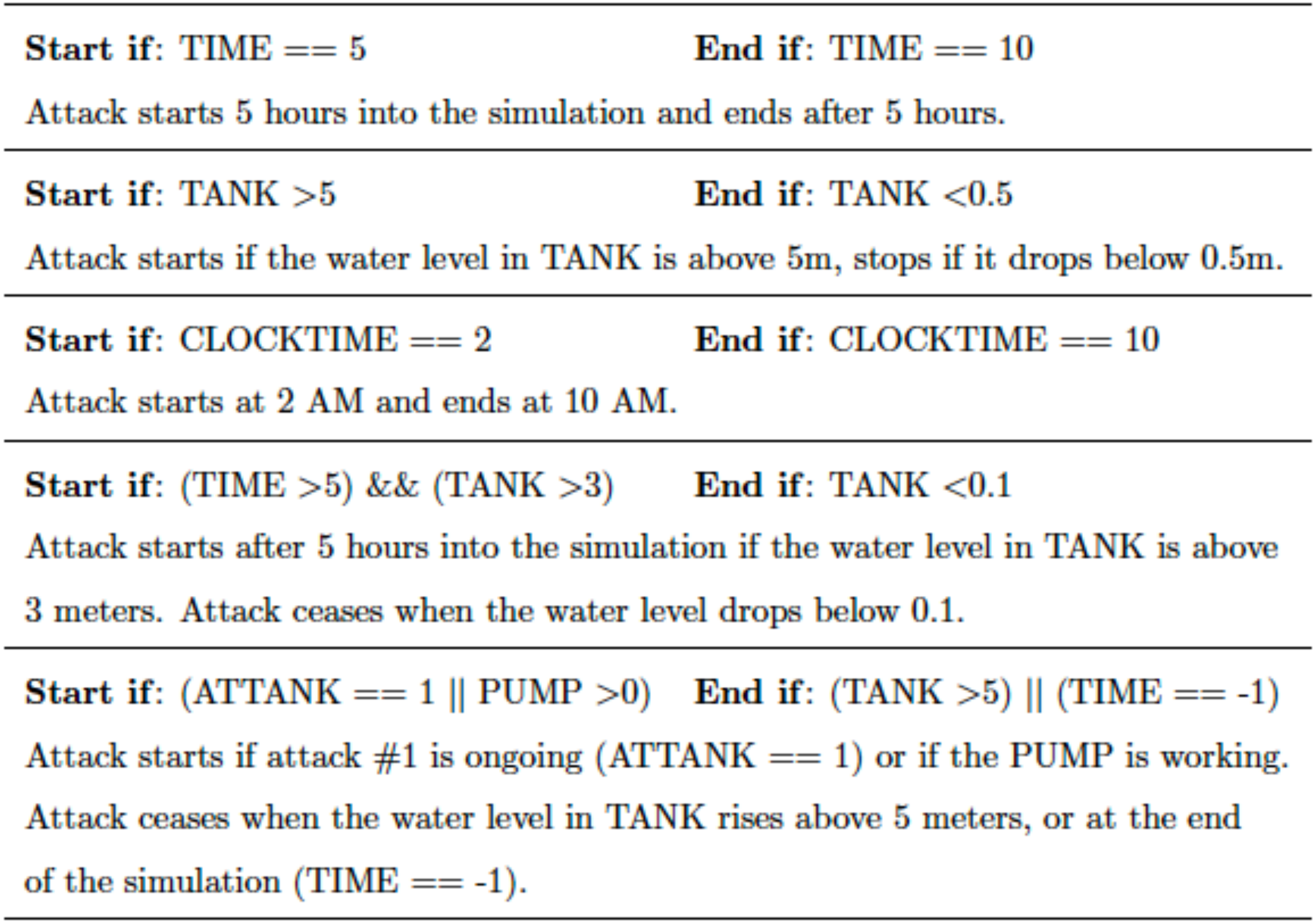
Arguments for epanetCPA attack classes



| AttackOn | Arguments | Description |
|---|---|---|
| Sensor | constant, <value> | Substitute sensor reading with constant <value> |
| | offset, <value> | Add offset to sensor reading |
| | custom, <filename> | Substitute sensor readings with values contained in .csv file |
| Actuator | <value> | Changes actuator setting with <value> |
| Control | <value> | Changes control setting point or controlled link with <value> |
| Communication | constant, <value> | Substitute sensor reading with constant <value> |
| | offset, <value> | Add offset to sensor reading |
| | custom, <value> | Substitute sensor readings with values contained in .csv file |
| | DoS | DoS of communication channel. Controller is unable to receive updated readings or send control signals. |
| | replay, <delta_t>, <noise intensity>, <max_value>, <min_value> | Replay attack on communication channel. Data starting from time t - <delta_t> is recorded and replayed in a loop. Gaussian noise can be added to the readings, and values can be clipped within an interval. |

# Appendix 3

Examples of attacks starting and ending conditions



**Start if**: TIME == 5 **End if**: TIME == 10

Attack starts 5 hours into the simulation and ends after 5 hours.

**Start if**: TANK >5 **End if**: TANK <0.5

Attack starts if the water level in TANK is above 5m, stops if it drops below 0.5m.

**Start if**: CLOCKTIME == 2 **End if**: CLOCKTIME == 10

Attack starts at 2 AM and ends at 10 AM.

**Start if**: (TIME >5) && (TANK >3) **End if**: TANK <0.1

Attack starts after 5 hours into the simulation if the water level in TANK is above 3 meters. Attack ceases when the water level drops below 0.1.

**Start if**: (ATTANK == 1 || PUMP >0) **End if**: (TANK >5) || (TIME == -1)

Attack starts if attack #1 is ongoing (ATTANK == 1) or if the PUMP is working. Attack ceases when the water level in TANK rises above 5 meters, or at the end of the simulation (TIME == -1).