# The Process Matters: Ensuring Data Veracity in Cyber-Physical Systems

Group 8

K. Olsson S. Finnsson

DAT300, 6. Oct 2016

# Outline

1. Introduction
   - What is an Industrial Control System?
   - Purpose and Design
2. Attacking an ICS
   - Network level attacks
   - Process level attacks
   - Sensor level attacks
3. Tennessee Eastman (TE) process
   - Tennessee Eastman - General facts
4. Detection
   - Approach: Information theory
   - Data: Need for discretization
   - Entropy: Sensor-specific, plant-wide and cluster-based
   - Results

# Outline

- Industrial control systems (ICS) is an encompassing term for several control systems and instrumentation used in industrial production.[1]

[1] *Industrial control systems*, [Online]. Available: https://en.wikipedia.org/wiki/Industrial_control_system.

- Industrial control systems (ICS) is an encompassing term for several control systems and instrumentation used in industrial production.[1]
- They are used to control cyber-physical systems, such as sensors, actuators, motors and more.

[1] *Industrial control systems*, [Online]. Available: https://en.wikipedia.org/wiki/Industrial_control_system.

- Industrial control systems (ICS) is an encompassing term for several control systems and instrumentation used in industrial production.[1]
- They are used to control cyber-physical systems, such as sensors, actuators, motors and more.
- ICS have taken over the responsibilities of older analog systems.

[1] *Industrial control systems*, [Online]. Available: https://en.wikipedia.org/wiki/Industrial_control_system.

# Outline

# Purpose and requirements

- The main purpose of an ICS is to keep an industrial plant up and running as autonomously as possible.

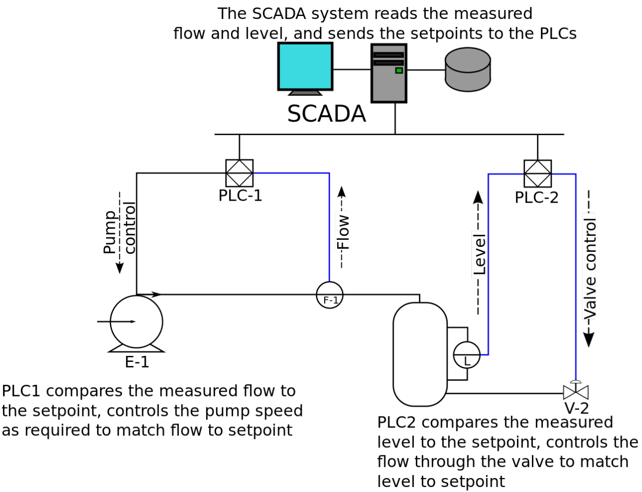
# Purpose and requirements

- The main purpose of an ICS is to keep an industrial plant up and running as autonomously as possible.
- In Industrial Processes one of the main considerations is availability and reliability of the systems, such that *uptime* is maximized.

# Design

- Simple hardware and simple protocols designed for high uptime, but no security.
- Multiple networks in a single ICS.

# ICS - an overview[2]




[2] *Scada schematic overview*, [Online]. Available: 'https://upload.wikimedia.org/wikipedia/commons/0/0c/SCADA_schematic_overview-s.svg'.

# Summary of ICS

- Large number of communicating devices
- Low inherent security

# Outline

# Network level attacks

- It is possible to attack the IT infrastructure used by the ICS resulting in loss of availability or malicious interference with the process.

# Network level attacks

- It is possible to attack the IT infrastructure used by the ICS resulting in loss of availability or malicious interference with the process.
- Well known mitigation techniques exist. Firewalls, intrusion detection systems and so on.

# Network level attacks

- It is possible to attack the IT infrastructure used by the ICS resulting in loss of availability or malicious interference with the process.
- Well known mitigation techniques exist. Firewalls, intrusion detection systems and so on.
- Have we then solved the problem of securing industrial control systems?

# Outline

# Process level attacks

- Attacking the physical process that the ICS controls.

# Process level attacks

- Attacking the physical process that the ICS controls.
- The process often has to interpret unmeasured quantities. E.g. change in pressure might be the result of temperature, flow or reaction speed.

# Process level attacks

- Attacking the physical process that the ICS controls.
- The process often has to interpret unmeasured quantities. E.g. change in pressure might be the result of temperature, flow or reaction speed.
- Non monitored equipment and processes can be used to influence other process.

# Outline

# Sensor level attacks

- In process automation sensors are considered fully trusted devices and the data they produce is trusted without further validation.

# Sensor level attacks

- In process automation sensors are considered fully trusted devices and the data they produce is trusted without further validation.
- Sensors are often closest to the physical process and sometimes the only way to monitor the process.

# Sensor level attacks

- In process automation sensors are considered fully trusted devices and the data they produce is trusted without further validation.
- Sensors are often closest to the physical process and sometimes the only way to monitor the process.
- **Veracity:** The property that an assertion truthfully reflects the aspect it makes a statement about.

# Sensor level attacks

- In process automation sensors are considered fully trusted devices and the data they produce is trusted without further validation.
- Sensors are often closest to the physical process and sometimes the only way to monitor the process.
- **Veracity:** The property that an assertion truthfully reflects the aspect it makes a statement about.
- A traditional network security approach is ineffective against these kind of attacks.

# Sensor Signal Spoofing

- Sensor signal spoofing is not straightforward.

# Sensor Signal Spoofing

- Sensor signal spoofing is not straightforward.
- It is necessary to mimic the behaviour of the real sensor.

# Sensor Signal Spoofing

- Sensor signal spoofing is not straightforward.
- It is necessary to mimic the behaviour of the real sensor.
- "Record-and-Playback".

# Sensor Signal Spoofing

- Sensor signal spoofing is not straightforward.
- It is necessary to mimic the behaviour of the real sensor.
- "Record-and-Playback".
- Runs Analysis, designing noise that is believable to the human operator.

# Sensor Signal Spoofing

- Sensor signal spoofing is not straightforward.
- It is necessary to mimic the behaviour of the real sensor.
- "Record-and-Playback".
- Runs Analysis, designing noise that is believable to the human operator.
- Triangle Approximation, Creating believable dynamic process behaviour.

- In a sequence of consecutive samples from a sensor, count the number of increasing or decreasing values ("runs up", "runs down")
- Count the distance travelled for each of those runs, up or down. Each run can be characterized by number of consecutive increasing/decreasing values and the distance travelled.
- Example:

$$[33.47 \quad 34.73 \quad 37.77] \rightarrow (+3, 4.3)$$

- The average distance travelled by each length of run can then be represented by a single distribution.
- Can be optimized, requires about 400 bytes of memory for combined code and data.

# Triangle Approximation I

## The algorithm

1. Declare a vertex at the first value.
2. Choose an arbitrary starting window of size $n$. Signal smoothing factor $s = \log n$.
3. Note minimum and maximum values of the window.
4. Draw a vertical line at sample n. Then draw two lines from the vertex, one through the minimum value and one through the maximum value, ending at the vertical line.
5. Declare a vertex at the midpoint of the vertical line at sample n.
6. Start drawing a triangle from the vertex on the vertical line.
7. Count the number of samples above (y) and below (z) the triangle.
8. When the number of samples above or below the triangle is above the threshold, $y$ or $z > s$, draw a vertical line through the current sample and declare a vertex at the midpoint.

# Triangle Approximation II

## The algorithm

9. If $y < z$, increase the slope of the top line and decrease the slope of the bottom line. If $y > z$ do the opposite.
10. If the number of samples between the current sample and the last vertex is $< 4n$, increase $n$.
11. If no new vertex is created within $4n$ samples, declare a vertex at the midpoint of the vertical line through the sample and decrease $n$.
12. Go to step 6.

- Network level attacks are well understood and can be combated with classic network security techniques.

- Network level attacks are well understood and can be combated with classic network security techniques.
- Process level attacks are hard to detect and can be devastating due to their nature.

- Network level attacks are well understood and can be combated with classic network security techniques.
- Process level attacks are hard to detect and can be devastating due to their nature.
- Sensor level attacks are feasible and hard to detect with traditional network security techniques as the sensor traffic looks normal.

- Network level attacks are well understood and can be combated with classic network security techniques.
- Process level attacks are hard to detect and can be devastating due to their nature.
- Sensor level attacks are feasible and hard to detect with traditional network security techniques as the sensor traffic looks normal.
- Run analysis and Triangle approximation can be used to spoof realistic dynamic sensor values making it hard for humans to detect.

# Outline

# Tennessee Eastman - General facts

- Model of an industrial chemical process[3]

[3] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993. DOI: http://www.sciencedirect.com/science/article/pii/009813549380018I.

# Tennessee Eastman - General facts

- Model of an industrial chemical process[3]
- Complex model

[3] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993. DOI: http://www.sciencedirect.com/science/article/pii/009813549380018I.

# Tennessee Eastman - General facts

- Model of an industrial chemical process[3]
- Complex model
  - 4 reactants $\longrightarrow$ 2 products

[3] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993. DOI: http://www.sciencedirect.com/science/article/pii/009813549380018I.

# Tennessee Eastman - General facts

- Model of an industrial chemical process[3]
- Complex model
  - 4 reactants $\longrightarrow$ 2 products
  - 41 measurements

[3] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993. DOI: http://www.sciencedirect.com/science/article/pii/009813549380018I.

# Tennessee Eastman - General facts

- Model of an industrial chemical process[3]
- Complex model
  - 4 reactants $\longrightarrow$ 2 products
  - 41 measurements
  - 12 adjustable variables

[3] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993. DOI: http://www.sciencedirect.com/science/article/pii/009813549380018I.

# An illustration[4]

246

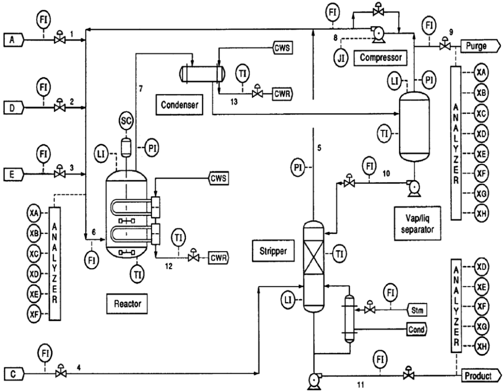



Fig. 1. Tennessee Eastman test problem.

J. J. Downs and E. F. Vogel

[4] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993. DOI:

# Outline

# Approach: Information theory

- Entropy: Randomness of information measured

# Approach: Information theory

- Entropy: Randomness of information measured
- For a discrete random variable $X$, with possible outcomes (values) $\{x_1, \ldots, x_n\}$, the entropy $H(X)$ is given by:

$$H(X) = \sum_{i=1}^{n} P(x_i) \cdot \log_a \left( \frac{1}{P(x_i)} \right)$$

, where $P(x_i)$ is the probability of symbol $x_i$ occuring.

# Outline

# Data: Need for discretization

- TE process model:

# Data: Need for discretization

- TE process model:
  - Double-precision floating point format

# Data: Need for discretization

- TE process model:
  - Double-precision floating point format
  - For each exponent: approximately $10^{16}$ variations

# Data: Need for discretization

- TE process model:
  - Double-precision floating point format
  - For each exponent: approximately $10^{16}$ variations
  - Remember the definition of entropy:

$$H(X) = \sum_{i=1}^{n} P(x_i) \cdot \log_2 \left( \frac{1}{P(x_i)} \right)$$

# Data: Need for discretization

- TE process model:
  - Double-precision floating point format
  - For each exponent: approximately $10^{16}$ variations
  - Remember the definition of entropy:

$$H(X) = \sum_{i=1}^{n} P(x_i) \cdot \log_2 \left( \frac{1}{P(x_i)} \right)$$

  - $n \ll 10^{16} \Rightarrow P(X) \sim \mathit{Uniform}$

# Data: Need for discretization

- TE process model:
  - Double-precision floating point format
  - For each exponent: approximately $10^{16}$ variations
  - Remember the definition of entropy:

$$H(X) = \sum_{i=1}^{n} P(x_i) \cdot \log_2 \left( \frac{1}{P(x_i)} \right)$$

  - $n \ll 10^{16} \Rightarrow P(X) \sim Uniform$
  - Each simulation would result in identical entropy

# Data: Need for discretization

- Method: Binning, but . . .

# Data: Need for discretization

- Method: Binning, but ...
- ...limited amount of bins would result in excessive number of mapping collisions

# Data: Need for discretization

- Method: Binning, but ...
- ...limited amount of bins would result in excessive number of mapping collisions
- Solution: quantize without limiting number of bins.
  For all $z \in \mathbb{R} : \lfloor |z| \rfloor = x \in \mathbb{N}_0$, and $\lfloor 10 \cdot (|z| - x) \rfloor = y \in \mathbb{N}_0$, then $f(z) \in \mathbb{Z}$ is defined as

# Data: Need for discretization

- Method: Binning, but ...
- ... limited amount of bins would result in excessive number of mapping collisions
- Solution: quantize without limiting number of bins.
  For all $z \in \mathbb{R} : \lfloor |z| \rfloor = x \in \mathbb{N}_0$, and $\lfloor 10 \cdot (|z| - x) \rfloor = y \in \mathbb{N}_0$, then $f(z) \in \mathbb{Z}$ is defined as

$$f(z) = \begin{cases} -(10 \cdot x + y), & \text{for } z < 0 \\ 0, & \text{for } z = 0 \\ (10 \cdot x + y), & \text{for } z > 0 \end{cases}$$

# Outline

# Entropy: Sensor-specific, plant-wide and cluster-based

- Sensor-specific and plant-wide entropy

# Entropy: Sensor-specific, plant-wide and cluster-based

- Sensor-specific and plant-wide entropy
  - Sensor-specific entropy (se) is calculated for one particular sensor over a period of time ($n$ number of samples). Plant-wise entropy (pwe) on the other hand is calculated at a given sample time for all sensors ($n$ number of sensors) simultaneously. The combined entropy matrix looks like this:

# Entropy: Sensor-specific, plant-wide and cluster-based

- Sensor-specific and plant-wide entropy
  - Sensor-specific entropy (se) is calculated for one particular sensor over a period of time ($n$ number of samples). Plant-wise entropy (pwe) on the other hand is calculated at a given sample time for all sensors ($n$ number of sensors) simultaneously. The combined entropy matrix looks like this:

| | s1 | s2 | s3 | s4 | ... | sn | |
|---|---|---|---|---|---|---|---|
| t1 | • | • | s3 | • | • | • | pwe1 |
| t2 | • | • | s3 | • | • | • | pwe2 |
| t3 | • | • | s3 | • | • | • | pwe3 |
| ⋮ | • | • | s3 | • | • | • | ⋮ |
| tn | • | • | s3 | • | • | • | pwen |
| | se0 | se1 | se2 | se3 | ... | sen | |

# Entropy: Sensor-specific, plant-wide and cluster-based (continued)

- Sensor-specific and plant-wide entropy
  - Plant-wide entropy can effectively detect anomalies that affect multiple sensor measurements simultaneously. However, it cannot specify from which sensor(s) the disturbance originates.

# Entropy: Sensor-specific, plant-wide and cluster-based (continued)

- Sensor-specific and plant-wide entropy
  - Plant-wide entropy can effectively detect anomalies that affect multiple sensor measurements simultaneously. However, it cannot specify from which sensor(s) the disturbance originates.
  - Entropy for a specific sensor is calculated so that the affected sensor can be located. However, if the attacker is able to spoof the signal, sensor-specific entropy is rendered useless.

# Entropy: Cluster-based

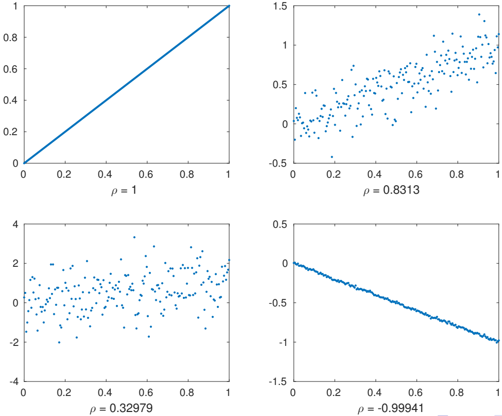
- Notion: calculate entropy in clusters based on sensor correlation $\rho$

$$\rho = \frac{\mathit{cov}\,(X, Y)}{\sigma_X \cdot \sigma_Y}$$

# Entropy: Cluster-based

- Notion: calculate entropy in clusters based on sensor correlation $\rho$

$$\rho = \frac{cov\left(X, Y\right)}{\sigma_X \cdot \sigma_Y}$$



# Entropy: Cluster-based (continued)

- Consider $\rho$ between each of the 41 measurements from the TE model.

[5] *Metis - family of graph and hypergraph partitioning software*, [Online]. Available: https://http://glaros.dtc.umn.edu/gkhome/views/metis.

# Entropy: Cluster-based (continued)

- Consider $\rho$ between each of the 41 measurements from the TE model. The resulting matrix can be considered an adjacency graph with weighted edges.

[5] *Metis - family of graph and hypergraph partitioning software*, [Online]. Available: https://http://glaros.dtc.umn.edu/gkhome/views/metis.

# Entropy: Cluster-based (continued)

- Consider $\rho$ between each of the 41 measurements from the TE model. The resulting matrix can be considered an adjacency graph with weighted edges.
- Cluster graph with respect to edge weights (correlation between measurements)

[5] *Metis - family of graph and hypergraph partitioning software*, [Online]. Available: https://http://glaros.dtc.umn.edu/gkhome/views/metis.

# Entropy: Cluster-based (continued)

- Consider $\rho$ between each of the 41 measurements from the TE model. The resulting matrix can be considered an adjacency graph with weighted edges.
- Cluster graph with respect to edge weights (correlation between measurements)
- Graph partitioning software METIS[5] used for initial clustering. Deciding number of clusters challenging - for TE process $\approx 10 - 13$

[5] *Metis - family of graph and hypergraph partitioning software*, [Online]. Available: https://http://glaros.dtc.umn.edu/gkhome/views/metis.

# Entropy: Cluster-based (continued)

- Consider $\rho$ between each of the 41 measurements from the TE model. The resulting matrix can be considered an adjacency graph with weighted edges.
- Cluster graph with respect to edge weights (correlation between measurements)
- Graph partitioning software METIS[5] used for initial clustering. Deciding number of clusters challenging - for TE process $\approx 10 - 13$
- Time-window (period over which entropy was calculated) set to 45 minutes (75 samples) and smoothing of the sensor signals applied. At first non-overlapping time-windows used - resulting in poor detecting capabilities for weakly correlated sensors. Also large variation in entropy outside attack-window.

[5] *Metis - family of graph and hypergraph partitioning software*, [Online]. Available: https://http://glaros.dtc.umn.edu/gkhome/views/metis.

# Entropy: Cluster-based (continued)

- Consider $\rho$ between each of the 41 measurements from the TE model. The resulting matrix can be considered an adjacency graph with weighted edges.
- Cluster graph with respect to edge weights (correlation between measurements)
- Graph partitioning software METIS[5] used for initial clustering. Deciding number of clusters challenging - for TE process $\approx 10 - 13$
- Time-window (period over which entropy was calculated) set to 45 minutes (75 samples) and smoothing of the sensor signals applied. At first non-overlapping time-windows used - resulting in poor detecting capabilities for weakly correlated sensors. Also large variation in entropy outside attack-window.
- To solve these weaknesses a sliding time-window was used to calculate the entropy. Downside is a delay for the uncorrelated samples to dominate entropy.

[5] *Metis - family of graph and hypergraph partitioning software*, [Online]. Available: https://http://glaros.dtc.umn.edu/gkhome/views/metis.

# Outline

# Simulated results

- Ability to detect attacks
  - Can effectively detect questionable *veracity* in changes to the data

# Simulated results

- Ability to detect attacks
  - Can effectively detect questionable *veracity* in changes to the data
  - Still able to detect anomalies when the attacker has knowledge of the sensor clustering
- Complexities not fully resolved

# Simulated results

- Ability to detect attacks
  - Can effectively detect questionable *veracity* in changes to the data
  - Still able to detect anomalies when the attacker has knowledge of the sensor clustering
- Complexities not fully resolved
  - Wrongful inclusion of sensor in cluster produces false positive alarms.

# Simulated results

- Ability to detect attacks
    - Can effectively detect questionable *veracity* in changes to the data
    - Still able to detect anomalies when the attacker has knowledge of the sensor clustering
- Complexities not fully resolved
    - Wrongful inclusion of sensor in cluster produces false positive alarms.
    - If cluster consists of similar signals (type and scale), spoofing all of them using just one signal will result in cluster that is both *plausible* and *correlated*. Important to form clusters from signals of different types and scales.

# Summary

- ICS systems are often legacy system that require high uptime. This gives rise to outdated and vulnerable hardware, protocols and system design.

# Summary

- ICS systems are often legacy system that require high uptime. This gives rise to outdated and vulnerable hardware, protocols and system design.
- Sensor noise and Dynamic process behaviour can be believably spoofed

# Summary

- ICS systems are often legacy system that require high uptime. This gives rise to outdated and vulnerable hardware, protocols and system design.
- Sensor noise and Dynamic process behaviour can be believably spoofed
- Tennessee Eastman process is useful as a testbed

# Summary

- ICS systems are often legacy system that require high uptime. This gives rise to outdated and vulnerable hardware, protocols and system design.
- Sensor noise and Dynamic process behaviour can be believably spoofed
- Tennessee Eastman process is useful as a testbed
- Entropy- and cluster-based detection is a viable approach.