Information Sciences 690 (2025) 121534

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A data anonymization methodology for security operations centers: Balancing data protection and security in industrial systems

Giacomo Longo [a], Francesco Lupia [c], Alessio Merlo [b], Francesco Pagano [a], Enrico Russo [a,*]

[a] DIBRIS, University of Genoa, Genoa, Italy
[b] CASD, School for Advanced Defense Studies, Rome, Italy
[c] DIMES, University of Calabria, Arcavacata di Rende (CS), Italy

A B S T R A C T

In an era where industrial Security Operations Centers (SOCs) are paramount to enabling cybersecurity, they can unintentionally become enablers of intellectual property theft through the data they analyze and retain. The above issue requires finding solutions to strike a balance between data protection and security. This paper proposes a real-time data anonymization framework designed to operate directly within network devices. Using an extensive case study, our approach demonstrates how valuable intellectual property associated with industrial processes can be protected without compromising the effectiveness of behavioral anomaly detection systems. The methodology is designed to be nonintrusive, reversible, and seamlessly portable on existing security solutions. We evaluated these properties through comprehensive experimental testing, which showed both the method's effectiveness in securing intellectual property and its suitability for continuous real-time operation.

1. Introduction

Digitalization and interconnection of industrial processes have led to significant advances in efficiency and productivity. However, these benefits come with increased cybersecurity risks that require particular attention in the current threat landscape. The process industry, essential for global supply chains and public health, has become a prime target for various attackers, including cybercriminals and industrial espionage agents. Given the potential impacts, such as contamination, supply disruption, and brand damage, robust countermeasures are imperative.

Several industry standards and frameworks, such as ISO/IEC 27001 [1], NIST Cybersecurity Framework [2], or IEC 62443 [3], recommend *continuous monitoring* processes or functions such as "Detect" and "Respond" as a critical part of comprehensive cybersecurity strategies. This need for continuous monitoring and rapid response to cybersecurity incidents has prompted the adoption of Security Operations Centers (SOCs) [4], which are instrumental in supporting activity with specialized personnel and tools [5]. They collect process data and logs from industrial networks and correlate events to identify malicious activities. Data retention over specified periods enables detailed forensic analysis, facilitating investigation and response to suspicious activities or incidents. Although transmitting industrial traffic to SOCs is essential for their operation, it also poses a significant challenge in safeguarding potentially sensitive data within the industrial control system. Data leakage from SOC systems exposes the plant under protection to the disclosure of intellectual property, trade secrets, and other proprietary knowledge [6], highlighting the need to balance security with data protection and address the risk of *data exposure*.

* Corresponding author.
E-mail address: enrico.russo@unige.it (E. Russo).

https://doi.org/10.1016/j.ins.2024.121534
Received 1 May 2024; Received in revised form 28 September 2024; Accepted 6 October 2024

Available online 10 October 2024

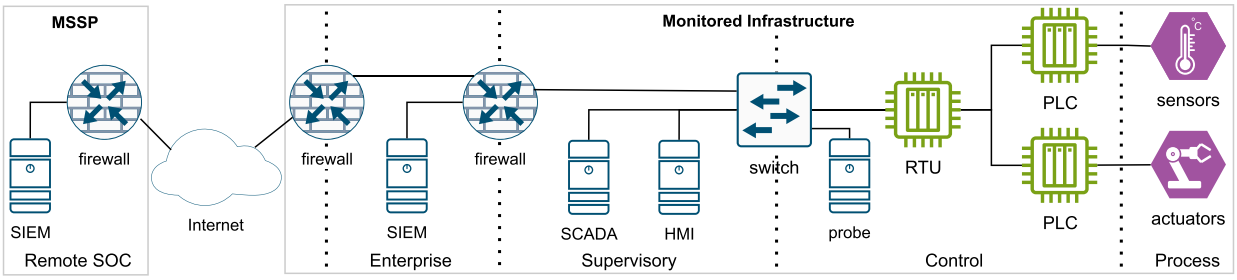


Fig. 1. The digital infrastructure.

This paper explores the challenges of transmitting industrial data to SOCs and the potential risks of their exposure, all while considering the delicate interplay between protecting sensitive data and operational needs for cybersecurity countermeasures. It proposes a novel methodology based on real-time traffic anonymization designed to mitigate risks of data exposure while protecting industrial assets.

The main contributions of the paper can be summarized as follows.

- We propose a methodology based on standard Linux capabilities that allows real-time network traffic capture and a custom solution to anonymize protocol data. Given the growing ubiquity of Linux-based network operating systems, our proposal can be executed across a wide range of modern networking equipment and seamlessly integrated with the existing infrastructure.
- We present a case study replicating a realistic industrial process, the potential attacks it may face, and how malicious activity can be detected by SOCs using state-of-the-art solutions.
- Through our case study, we conduct comprehensive testing to validate the real-time capabilities and effectiveness of our anonymization approach. We also illustrate that anomaly detection remains robust post-transformation and that our solution is designed to require no modifications to existing monitoring implementations.

Finally, it is essential to note that our methodology is designed to be applicable across a wide range of industrial contexts and protocols beyond those demonstrated in the case study. This adaptability also extends to nonindustrial domains, such as healthcare and financial services, indicating the potential for cross-domain applicability and innovation in anonymization techniques.

Structure of the paper. The remainder of the paper is organized as follows. We detail the specifics of a case study in Section 2, which serves as a working example throughout the remainder of the paper. Section 3 introduces the problem of data exposure in security solutions for anomaly detection and outlines our methodology. Section 4 presents the implementation of the case study, the experimental setup, and the experiments carried out to evaluate the methodology. Finally, we explore related work in Section 5 and conclude in Section 6.

2. Case study

This section presents our case study on digital infrastructure and its associated industrial process. Finally, assuming adversaries are attacking this infrastructure, we describe the countermeasures implemented to identify them.

2.1. Digital infrastructure

Fig. 1 depicts the digital infrastructure we consider. It follows a segmentation pattern similar to the Purdue Enterprise Reference Architecture [7].

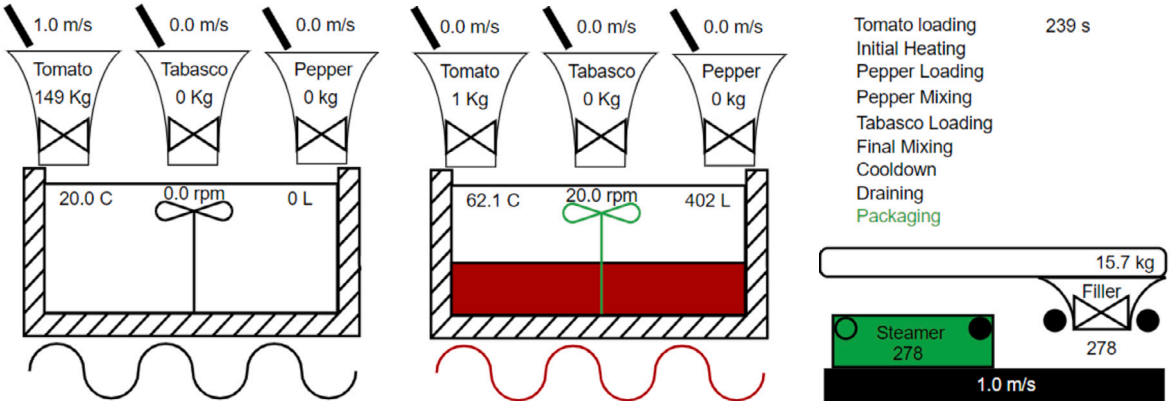
The far right of the diagram represents the Operational Technology (OT) layer of the infrastructure under monitoring. Then, it moves toward the Information Technology (IT) layer and finally to the remote Managed Security Service Provider (MSSP).

The *process* component represents the actual physical processes being controlled and monitored. This component includes *sensors* collecting data from the physical environment, such as temperature or pressure, and *actuators* that physically act based on commands from control systems, such as opening a valve or starting a motor.

In the *control* component, Programmable Logic Controllers (PCLs) gather sensor data and send commands to actuators. One or more Remote Terminal Units (RTUs) connect PLCs, use the data collected to make decisions, and order commands to control industrial processes.

The *supervisory* component comprises systems that offer high-level control of the process through a Supervisory Control and Data Acquisition (SCADA) system overseen by a Human Machine Interface (HMI) solution.

The systems and equipment within the supervisory and control components are interconnected via a switched network. A dedicated port on a switch, namely Switch Port ANalyzer *SPAN* port, captures a mirrored copy of network traffic acquired by a probe. The probe runs software that analyzes traffic and extracts information to be sent to a remote collector, Security Information and Event Management (SIEM). SIEM can monitor, detect, and respond to security threats. Our case study examines a probe that captures OT traffic, specifically the data sourced from the Modbus protocol.



(a) Loading ingredients. (b) Mixing. (c) Packaging.

Fig. 2. The HMI for the industrial process.

A local SOC can conduct security monitoring of the OT infrastructure. In this scenario, the SIEM system is hosted within the *enterprise* component, which serves as the core of the IT network. A firewall regulates data traffic between the enterprise network and the monitored infrastructure. Alternatively, the enterprise can utilize the expertise and resources of an MSSP without needing a comprehensive in-house security infrastructure [8]. In such a setup, a remote SIEM hosted by the MSSP is reachable through a secure Internet connection established by two perimeter firewalls.

2.2. Industrial process

Our case study explores an industrial food processing plant preparing and bottling hot sauce. Fig. 2 represents an image excerpt of the HMI, visually representing the ongoing process. It includes real-time sensor readings and the status of actuators associated with the production line.

The production line is divided into three main zones: ingredient weighting (see Fig. 2a), food processing (see Fig. 2b), and packaging (see Fig. 2c).

The first zone is where ingredients are initially prepared and weighed. Portioning takes place via three weighting funnels, respectively, for tomato sauce, Tabasco sauce, and pepper, each fed by speed-controllable screw pumps (for sauces) or by a belt (for the peppers). Each funnel includes a load cell to sense the current amount of ingredient found within it.

The central zone comprises a mixing tank that cooks and pasteurizes the sauce. This mixer is equipped with sensors monitoring its level and temperature and controls for adjusting the mixing speed and turning on its heating element. Additionally, another setpoint commands the drainage of the tank contents towards the packaging zone.

The packaging zone features a controllable speed belt that is loaded with sauce containers at regular intervals. Along the belt line, two stations allow sanitization of the containers with hot steam and filling them with sauce drained from the tank. This area has sensors for both the weight sensed by the funnel feeding the filling area and two photocells at the extremities of each area to sense when a container has entered or exited the area.

All these five elements have PLCs controlling local functions, such as driving the ingredient feeding screws at a set speed. The measurements and setpoints are exposed via Modbus to a central coordinator responsible for orchestrating the industrial process according to a programmed recipe. Table 1 summarizes PLCs and their corresponding registers, detailing the type, address, kind, and a short explanation for each entry.

The coordinator cycles through the following nine steps, representing an end-to-end sauce production process:

1. **Tomato sauce loading**. The screw pump linked to the tomato sauce funnel activates and runs until the funnel load cell detects a minimum of $400kg$ of ingredient. Upon reaching this threshold, the screw pump stops, and the funnel opens, adding its contents to the mixer.
2. **Initial heating**. The mixer begins to stir the mixture at $20rpm$ while simultaneously heating it to $62\,°C$. This temperature is maintained for 8 minutes.
3. **Pepper loading**. The mixer temperature is lowered to $55\,°C$, and its speed is reduced to $5rpm$. Simultaneously, the pepper belt feeder is engaged until its weight sensor indicates it has reached $280kg$. Then, the peppers are added to the mixture.
4. **Pepper mixing**. The mixer temperature is raised to $58\,°C$, and its speed is further reduced to $2.5rpm$. The mixture is stirred for 4 minutes.
5. **Tabasco loading**. The Tabasco pump is activated until the funnel measures $20kg$. Then, the additional sauce is dumped into the mixer.
6. **Final mixing**. The mixture is cooled to $50\,°C$ and slowly stirred at $1rpm$ for 5 minutes.
7. **Cooldown**. The mixer heating element is turned off until the mixture temperature has settled at $25\,°C$ or below.

Table 1
PLCs and their registers.

PLC	Addr.	Kind	Description
Tomato Scale	IR1	RO16	Roller speed
	IR2	RO16	Scale weight
	HR1	RW16	Roller target speed
	CO1	RW1	Open funnel
Tabasco Scale	IR1	RO16	Roller speed
	IR2	RO16	Scale weight
	HR1	RW16	Roller target speed
	CO1	RW1	Open funnel
Pepper Scale	IR1	RO16	Roller speed
	IR2	RO16	Scale weight
	HR1	RW16	Roller target speed
	CO1	RW1	Open funnel
Mixer	IR1	RO16	Fluid Level
	IR2	RO16	Stirrer speed
	IR3	RO16	Temperature
	HR1	RW16	Stirrer target speed
	CO1	RW1	Open drain
	CO2	RW1	Energize heater
Packager	IR1	RO16	Scale weight
	IR2	RO16	Steamer activation count
	IR3	RO16	Filler activation count
	IR4	RO16	Belt speed
	HR1	RW16	Belt target speed
	DI1	RO1	Steamer entry photocell
	DI2	RO1	Steamer exit photocell
	DI3	RO1	Filler entry photocell
	DI4	RO1	Filler exit photocell
	CO1	RW1	Fill
	CO2	RW1	Open steam valve
Coordinator	IR1	RO16	Current state
	IR2	RO16	Time in state
	IR3	RO16	Cycle count
	CO1	RW1	Stop process

8. **Draining to packaging zone**. The mixing tank drain valve is opened, and its contents are transferred into the filling machine funnel.
9. **Packaging**. The packaging zone belt is started and runs at a constant speed. As soon as a container reaches the start of the sanitation area photocell, the coordinator activates the sanitizing steam jets and deactivates them when the container triggers the photocell's end. A similar process occurs for the filling tube, with its entry and exit photocells initiating and concluding the procedure. This process repeats until the funnel load cell indicates sufficient container material.

Certain aspects of the sauce production process, such as the timing of the initial heating, the ingredient ratios during portioning, and the temperature adjustments during mixing, must be considered valuable industrial secrets. Protecting these elements is crucial to safeguarding intellectual property and maintaining a competitive advantage.

2.3. Adversary model and defense strategies

We consider attackers with access to the control or supervisory components of the digital infrastructure. This situation enables them to engage with RTUs and PLCs through the switched network and the Modbus protocol. As a result, they can leverage the weaknesses of the communication protocol and exploit several known attacks [9–11]. Briefly, a taxonomy of potential threats comprises attacks on the industrial plant's confidentiality, availability, and integrity [12]. Confidentiality attacks include reading Modbus messages to gain access to sensitive messages or device configuration data. Disruptions to availability can cause PLCs to lose essential functions, such as generating or receiving Modbus communications, or fail. Compromising integrity entails injecting incorrect information or alterations in the settings of these units.

In our scenario, the attackers aim to sabotage the product by changing its recipe, potentially causing recalls and harming the company's reputation. To achieve this, they engage in harmful tactics designed to compromise the integrity of the industrial process. These tactics exploit well-known adversarial techniques against Industrial Control Systems (ICS), identified as *Manipulation of Control* [13] and *Modify Program* [14]. Of particular relevance to our context, they target industrial process data and are detected through industrial process data analysis.

We outline three representative attacks below, assuming that attackers have partial knowledge of the process operations and associated PLC controls.

A_1 **Burning the sauce**. The attackers manipulate the PLC settings to energize the heater (see *energize heater* of Mixer PLC in Table 1) and overheat the mixture, causing the sauce to burn and become unusable.

A_2 **Boosting the Tabasco quantity**. By altering the PLC-controlled Tabasco Scale (see *roller target speed* in Table 1), attackers increase the amount of Tabasco added to the sauce, drastically altering its intended flavor profile.

A_3 **Disabling the sanitizing steam jet**. The attackers upload rogue logic onto the PLC that controls the packaging process (see *Packager* PLC in Table 1). This logic turns off the commands for the sanitizing steam jets, compromising the sterilization process and potentially contaminating the product.

In response to the outlined threats, a series of countermeasures to enhance the security of the production process are implemented. They are consistent with the National Institute of Standards and Technology (NIST) recommendations on "Securing Manufacturing Industrial Control Systems" as reported in the NISTIR 8219 [15] document. A critical aspect of this defense strategy is using Behavioral

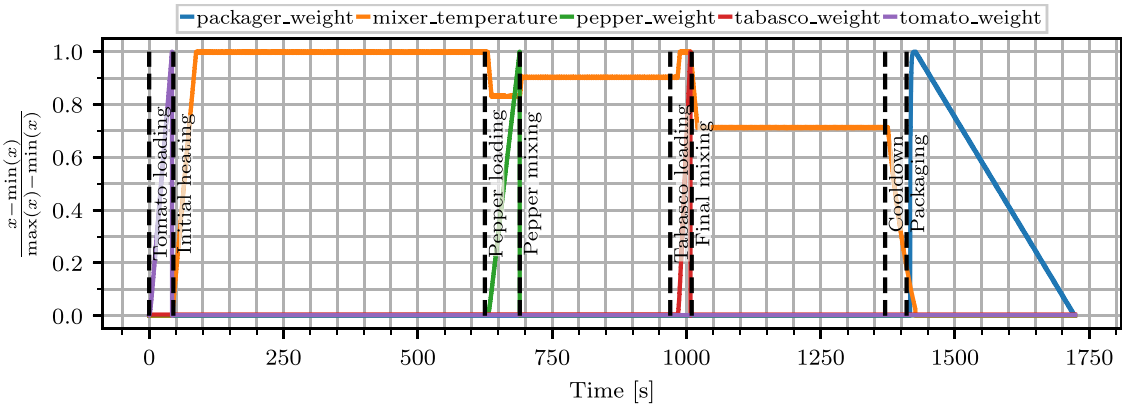


Fig. 3. Data gathered from SIEM during a production cycle.

Anomaly Detection (BAD), which focuses on identifying deviations from normal operating patterns within system data often indicative of potential cybersecurity incidents.

In detail, the defense solution involves deploying a monitoring system that continuously analyzes Modbus traffic. The system utilizes a probe (see Section 2.1) to gather data transmissions between the PLCs and the central coordinator, capturing operational commands and setpoints. The collected data is then forwarded to a specialized SIEM on-premise or remotely.

The core BAD functionalities are executed within the SIEM system, utilizing a well-established, state-of-the-art detection solution based on statistical and machine learning paradigms to identify anomalies within operational data [16,17]. This approach detects unusual patterns in operational data, providing early warning of potential security incidents. By tracking deviations from the norm, the system effectively protects against actions that compromise the integrity of the process, effectively countering threats similar to the attacks mentioned above.

3. Anonymization methodology

This section introduces the problem of data protection in SIEM systems implementing BAD and then presents our solution to mitigate the issue. Specifically, it outlines our anonymization methodology applied to data obtained from network traffic and transmitted to SIEM systems.

3.1. BAD solutions and data protection

The defense mechanism based on BAD relies on collecting extensive historical operational data, which is critical for training and continuously updating detection capabilities. Additionally, keeping such a history enables SOC operators to reconstruct the events following a security incident accurately. However, this data collection, while critical for safeguarding the production process against cyberattacks, also introduces a potential risk of extracting secrets and sensitive information related to the industrial recipe (refer to Section 2.2).

In Fig. 3, we present an annotated plot of data obtained from the SIEM, representing a single production cycle. Additionally, we label each stage in the production process. By analyzing this information, it is possible to deduce the timings associated with an entire production cycle. As a result, almost all intellectual property related to the production can be extracted by querying this dataset.

To illustrate, here are some example queries from our case study and their corresponding potential impact on intellectual property.

Q1 Ratio between max (pepper_weight), max (tabasco_weight), and max (tomato_weight). This query returns the ratio of ingredients that belong to the mixture, allowing for reversing-engineering the portioning associated with the recipe.

Q2 avg(mixer_temperature). This query enables adversaries to determine the mixer temperature during each phase, gaining insights into the characteristic temperatures of the recipe.

Q3 Cardinality of bottler_weight for $t > 1420s$. This query determines how many product cartons have been produced and will be sold by the company. Constant access to this figure would allow a malicious actor to sell this insider knowledge to traders interested in beating the market by estimating the company's revenue before publishing accounting sheets.

Q4 $\frac{\max(\text{bottler\_weight})}{\mathbf{Q3}}$. This query provides a means to determine the size of containers used for selling products. Even though the subject of our case study sells these items on the open market, information like this might be less readily available in other contexts. For example, in ammunition packaging, the quantity of explosives per product is an extremely sensitive figure.

It should be noted that other systems within SCADA implementations collect data similar in depth and sensitivity to SIEM. For example, historians are tasked with storing and managing time series data from industrial processes and similarly collect a considerable amount of operational data. The difference is that SCADA components like historians are typically situated within the well-guarded

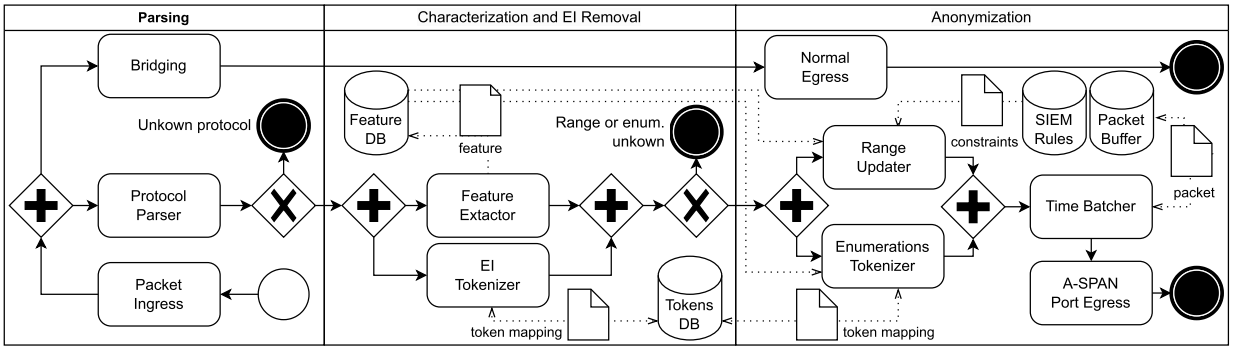


Fig. 4. Workflow of our anonymization methodology.

confines of the OT network. In contrast, SIEM systems extend beyond these secure boundaries into the Enterprise network or are even hosted externally by MSSP, introducing additional vectors for potential data leakage.

Given the critical need to safeguard sensitive operational data within SIEM systems, employing data anonymization techniques is crucial. However, the adopted methodology must meet the requirements outlined in the following section.

3.2. Methodology overview

As previously mentioned, data anonymization represents a viable methodology for protecting sensitive information stored in SIEM by transforming identifiable data into a format that does not expose real identities or proprietary information. Nevertheless, for anonymized data to retain its utility and ensure the continued effectiveness of BAD solutions, the employed anonymization technique must adhere to a set of requirements, as detailed below.

R1 *Real-time*. The technique must process and analyze data in real-time as it is received from the network.
R2 *Non-intrusive*. The technique must not diminish the ability to detect security anomalies in the industrial plant.
R3 *Reversible*. The technique must allow legitimate data owners to recover original information, even while inhibiting the probe from identifying them.
R4 *Compatible*. The anonymization technique must transparently integrate with the existing technological stack of the industrial environment and the security solution in place. For example, it does not necessitate modification in operation protocols, SIEM software, or BAD solution.

Fig. 4 depicts the workflow of our methodology, which aims to facilitate data anonymization within SIEM systems using BAD while complying with the requirements outlined above.

We assume it runs on network switches that can be placed in substitution for equipment installed on the monitored infrastructure (see Section 2.1). In particular, it provides a special port, namely *A-SPAN* (*Anonymizing*-SPAN), that functions similarly to a traditional SPAN port. However, it applies the anonymization methodology before returning a mirror of the traffic.

Our solution consists of three phases. The *Parsing* phase initiates by decoding the content of a packet for supported protocols. The *Characterization and Explicit Identifier removal* phase categorizes fields of a decoded packet, e.g., the cardinality or range, and replaces explicit identifiers. Finally, the last phase culminates with the *Anonymization* process.

We detail each phase in the sections below.

3.3. Parsing

Parsing starts once a packet is received by the *Packet Ingress*. There, the packet is copied before being sent through the normal *Bridging* path within the switch. The copied packet is instead processed by a *Protocol Parser*, which is responsible for decoding the entirety of the packet and keeping track of stream-based protocols like TCP. The duplicated packet is promptly discarded (see the *Unknown protocol* event) if the parser is unsuccessful, such as with malformed packets or unsupported protocols. This precaution is taken to prevent the release of potentially sensitive information to the security probe, as the anonymization system may not manage it. For example, an unparsed ICMP [18] packet might contain parts of a packet that failed to be routed, revealing the network identity of the sender endpoint.

3.4. Characterization and Explicit Identifier removal

Once the packet enters the *Characterization and Explicit Identifier (EI) removal* phase, it undergoes concurrent analysis and modification processes. The *Feature Extractor* analyzes the packet to extract information such as the cardinality of each variable field within the packet, the range of values for numeric fields, and identifying enumeration values, which occur when the cardinality C is below a user-provided threshold k_C. This data, calculated for each unique combination of EIs, is then stored within a *Feature Database*.

Simultaneously, the *EI Tokenizer* replaces each explicit identifier with a synthetic counterpart or token [19].
Formally, for a packet containing n_{EI} explicit identifiers, the EI Tokenizer:

1. Assembles a metaidentifier $I = \langle EI_1, \dots, EI_{n_{EI}} \rangle$.
2. If no mapping for I is found in the tokens database
   (a) Generates a synthetic identifier $\hat{I} = \langle \hat{EI}_1, \dots, \hat{EI}_{n_{EI}} \rangle$. Each $\hat{EI}$ is generated according to the field format found in the packet, e.g., an IPv4 address gets replaced by another one.
   (b) Persists the mappings $M = I \to \hat{I}$ and $M^{-1} = \hat{I} \to I$ to the tokens database.
3. Gets the mapping M associated with I from the database.
4. It applies M to the packet, replacing I with $\hat{I}$.

However, the EI fields are not removed from the ongoing packets in our solution. This decision is based on the need to potentially access this information for anomaly analysis, which is crucial for maintaining the integrity of the industrial process and ensuring the effectiveness of the SIEM.

To this end, we replace each EI value, maintaining the private mapping between the original and anonymized values. Because the EI mapping is only known internally, the transformation performed on the EIs is not invertible from external agents. Following these processes, the packet is discarded if insufficient information exists for its original EI combination I (see *Range or enum. unknown* event). More specifically, a packet is discarded if, for a particular I, all fields received in the past show uniform values, i.e., $C = 1$ $\forall i = 1, \dots, n_F$ with n_F being the number of fields found in the packet. This ensures that only packets with sufficient and diverse information are sent to the anonymization phase.

3.5. Anonymization

During the anonymization phase, either the *Range Updater* or the *Enumerations Tokenizer* modifies the parsed packet fields based on the classification received from the *Feature Extractor*.

Numeric fields undergo an invertible transformation by the *Range Updater* to alter the packet values without affecting the anomaly detection capabilities of the probe. The transformation T used consists of a linear shift and rescale transformation. It and its inverse T^{-1} are implemented as follows.

$$T(x,k,o) = k(x+o) \qquad T^{-1}(x,k,o) = \frac{x}{k} - o \tag{1}$$

To best determine the parameters for these transformations and maximize the differences between the anonymized and original versions, the transformation parameters are determined via an optimization problem. Such parameters k and o are the solution to this maximization problem, where x_{min}, x_{max} are the minimum, maximum data ranges as measured by the feature extractor and D_{min}, and D_{max} are the field domain bounds. It is worth noticing that the applied transformation can only be inverted with knowledge of the k and o parameter values, which must remain confidential to the owner.

$$\underset{k,o}{\operatorname{argmax}}\ k_r \cdot (k-1)^2 + k_o \cdot \frac{o^2}{D_{\max}^2} \tag{2}$$

subject to

$$k(x_{max} + o) \le D_{\max} \tag{3a}$$

$$k(x_{min} + o) \ge D_{\min} \tag{3b}$$

$$0.8 \le k \le 10 \tag{3c}$$

Equation (2) is the objective function to be maximized. This function consists of two terms. The first term, influenced by the user-defined coefficient k_r, relates to the rescaling factor, favoring ks far from the value 1. Similarly, the second term, weighted by the coefficient k_o, reflects the extent of the offset applied to the data mean. In minimizing the objective function, k_r determines the extent to which the algorithm prioritizes increasing data variability, whereas k_o adjusts the degree to which the algorithm shifts the values.

Additionally, constraints detailed in Equations (3a), (3b), and (3c) constrain the values of k and o. These constraints prevent the selected parameters from exceeding the maximum representable value (3a), falling below the minimum representable value (3b), and choosing a too lossy transformation (3c). In this context, we chose a maximum dynamic range compression of 20%, and a precautionary maximum dynamic range expansion of 10 dB.

As the shift rescale transformation also alters the standard deviation of the observed values when $k \neq 1$, the system allows the replacement of the constraint given in Equation (3c) with $k = 1$ if this alteration conflicts with the SIEM anomaly detection methodology.

Similarly, suppose the optimization codomain ($\mathbb{R}$) is incompatible with the field on-wire format. In that case, the final transformation is augmented with a further domain projection step, e.g., a conversion to integer for ModBus register values whose domain is in $\mathbb{N}_0$.

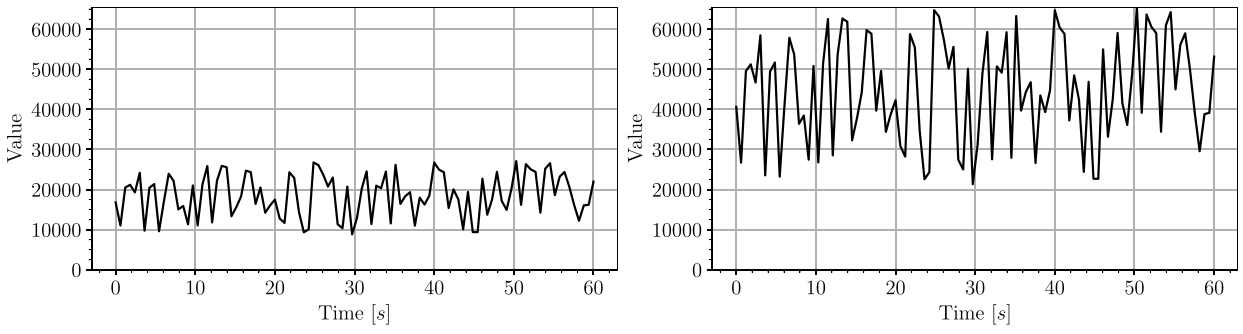


Fig. 5. Effects of the shift and rescale transformation on an example time series.

Table 2
ModBus value domain.

Description	Unit	$V_{\min}$	$V_{\max}$
Time	$[s]$	0	1800
Speed	$[\frac{m}{s}]$	0	1
Mass	$[kg]$	0	1000
Volume	$[l]$	0	1000
Angular velocity	$[rpm]$	0	60
Temperature	$[C]$	0	150

Finally, the *Range Updater* saves k and o in the features database.

Fig. 5 shows the effects of the transformation on an example time series, using $k_d = k_o = 1$ as user parameters. There, the transformed time series shows an expanded dynamic range compared to the original one while preserving the trends shown in the data.

For enumeration fields, the operation performed by the *Enumerations Tokenizer* is similar to that of the *EI Tokenizer*, but it targets the contents of the fields themselves. This process replaces the enumeration values with tokens and stores this mapping in the tokens database.

After transforming each field, the anonymized packet is forwarded to the *Time Batcher*. This component collects packets in its storage packet buffer and, after a certain period, B releases them in order. The initial batching period, $B(t_0)$, is estimated by measuring the time required to fill the buffer with n packets, where n is a user-selectable parameter. B is then adjusted with a proportional controller, based on the formula $B(t) = B(t-1) + k_p \cdot [n - \hat{n}(t-1)]$, where k_p is the gain of the proportional controller, and $\hat{n}$ is the number of packets found in the batch.

Following processing by the batcher, packets are sent out of the *A-SPAN* port to be picked up by the probe. Simultaneously, the packets initially directed to the normal bridge datapath are forwarded to the network.

4. Experimental evaluation

In this section, we describe our implementation of the methodology presented in Section 3. Then, we assess the performance and effectiveness of our approach through the case study and discuss the results.

4.1. Implementation

The experiments for this article were carried out using a virtual scenario, executing the process described in Section 2. The implementation leverages a containerized framework that can run virtual IT/OT systems and seamlessly connects with physical simulators, as detailed in [20,21].

In this environment, the PLCs, which are written in Python, run their logic at 10 Hz, following the typical scan inputs-process-write outputs cycle. They communicate using the ModBus/TCP [22,23] protocol.

$$\text{MBFloat}(x) : \mathbb{R} \to \mathbb{N}_0 \in [0, 2^{16}) = \min\left\{(2^{16}-1), \max\left[0, (2^{16}-1)\cdot\frac{x - V_{min}}{V_{max} - V_{min}}\right]\right\} \tag{4}$$

All floating point values were transformed to 16-bit fixed precision values as per Equation (4), with V_{min} and V_{max} set according to the expected values within the process (Table 2 shows this mapping). Furthermore, all field-level PLCs (non-coordinators) were connected to a physics simulator, generating the process responses to the controls as serially connected ModBus/RTU slaves.

Every network endpoint is attached to a Linux bridge, implementing a MAC learning Ethernet switch. In this switch, a Rust program that leverages AF_{PACKET} sockets [24] takes care of implementing all non-bridging tasks from the workflow of Fig. 4. It should be noted that currently a significant number of manufacturers produce white-box switches equipped with Linux-based or

Table 3
Transformation parameters.

Variable	k	o	x_{min}	x_{max}	$T(x_{min})$	$T(x_{max})$
Packager						
Scale weight	1.25	2214	0	48000	2768	62768
Belt target speed	1.0	0	0	65535	0	65535
Mixer						
Fluid level	1.25	2714	0	47000	3392	62142
Stirrer speed	2.81	661	0	22000	1857	63677
Stirrer target speed	2.81	661	0	22000	1857	63677
Temperature	1.7	1297	8738	27218	17060	48476
Pepper Scale						
Roller speed	1.0	0	0	65535	0	65535
Scale weight	3.21	208	0	20000	668	64868
Roller target speed	1.0	0	0	65535	0	65535
Tabasco Scale						
Roller speed	1.0	0	0	65535	0	65535
Scale weight	9.99	780	0	5000	7792	57742
Roller target speed	1.0	0	0	65535	0	65535
Tomato Scale						
Roller speed	1.0	0	0	65535	0	65535
Scale weight	2.27	435	0	28000	987	64547
Roller target speed	1.0	0	0	65535	0	65535

(a) Unconstrained (Eq. (3c)).

o	x_{min}	x_{max}	$T(x_{min})$	$T(x_{max})$
Packager				
8767	0	48000	8767	56767
0.0	0	65535	0	65535
Mixer				
9267	0	47000	9267	56267
21767	0	22000	21767	43767
21767	0	22000	21767	43767
14789	8738	27218	23527	42007
Pepper Scale				
0.0	0	65535	0	65535
22767	0	20000	22767	42767
0.0	0	65535	0	65535
Tabasco Scale				
0.0	0	65535	0	65535
30267	0	5000	30267	35267
0.0	0	65535	0	65535
Tomato Scale				
0.0	0	65535	0	65535
18767	0	28000	18767	46767
0.0	0	65535	0	65535

(b) $k = 1$ constraint.

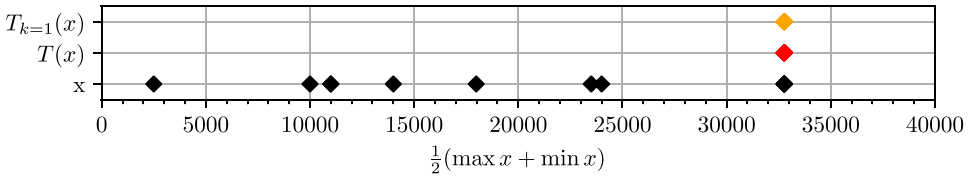


Fig. 6. Homogeneity in domain midpoints after the transformation.

Linux-compatible Network Operating Systems [25]. These systems are designed to natively support a wide range of custom software, including our methodology, through direct execution on the switch.

The anonymized packets are then sent to a Zeek probe, whose output is forwarded to an OpenSearch cluster.

OpenSearch acts as the SIEM of this scenario, with its built-in anomaly detection system [26] implementing a state-of-the-art BAD system.

4.2. Results

In the following, we present the results related to the transformation parameters identified through the optimization process, the efficacy of anomaly detection in anonymized data flows, and the effectiveness of our approach in protecting sensitive data. Finally, we will also report the performance figures of our implementation.

4.2.1. Optimization parameters

Tables 3a and 3b present the anonymization parameters determined by optimization in the unconstrained and offset only cases, with k = 1. These values were obtained using a grid search over the parameter space, exploring all combinations of k values ranging from 0.8 to 5 in 0.01 increments and o values spanning from 0 to 10000.

From these tables, it becomes apparent that when x_{min} and x_{max} are very close to D_{min} and D_{max} (respectively, 0 and 65535 in our case), the offset-only transformation (Table 3b) cannot alter the data range due to the original data domain being already almost fully utilized. In contrast, its unconstrained counterpart (Table 3a) succeeds in altering the data domain. Moreover, it is worth noting that the midpoint of the transformed domain is located closer to the domain center w.r.t. its non altered counterpart. We report this observation graphically in Fig. 6, showing the position of each domain midpoint.

4.2.2. Anomaly detection performance

Table 4 presents the success rate of the anomaly detection solution as anonymization settings are varied. To test this, 100 runs were executed with multiple probes receiving either the unmodified network traffic (first row) or one of several possible anonymized data flows (second row onwards).

Table 4
Anomaly detection success with varying anonymization settings.

EI Tokens	Range Update	Enum. Tokens	Time Batch	A_1	A_2	A_3
✗	✗	✗	✗	✓	✓	✓
✗	✗	✓	✗	✓	✓	✓
✓	✗	✗	✗	✓	✓	✓
✓	✗	✓	✗	✓	✓	✓
✓	Table 3a	✓	1 s	✓	✓	✓
✓	Table 3a	✓	2 s	✓	✓	✓
✓	Table 3a	✓	5 s	✓	89%	✓
✓	Table 3a	✓	10 s	✓	82%	✓
✓	Table 3a	✓	30 s	✓	46%	✓
✓	Table 3b	✓	1 s	✓	✓	✓
✓	Table 3b	✓	2 s	✓	✓	✓
✓	Table 3b	✓	5 s	✓	91%	✓
✓	Table 3b	✓	10 s	✓	81%	✓
✓	Table 3b	✓	30 s	✓	45%	✓

Table 5
Reconstructing the portioning among ingredients.

	Tomatoes	Tabasco	Peppers
Ratio	57.14	2.86	39.97
Δ[%]	0.02	0.0	0.03

(a) Non-anonymized data.

	Tomatoes	Tabasco	Peppers
Ratio	16.55	69.78	13.68
Δ [%]	40.59	66.92	26.32

(b) Anonymized data.

	Tomatoes	Tabasco	Peppers
Ratio	26.44	41.77	31.79
Δ [%]	30.71	38.92	8.21

(c) Anonymized data ($k = 1$).

Consequently, all 14 probes operated on the same packets, with the same content, order, and timing, except for differences in the applied anonymizations.

In the table, the left side illustrates which anonymization measures are enabled/disabled (represented by a ✓or a ✗), the chosen parameters for the range updater, and the time batcher window size. On the right side, we correctly count how many trials in which the anomaly detection system was able to detect attacks A_1, A_2, and A_3. When detection succeeded across all trials, this count is replaced by a check mark (✓).

From the data, it is evident that detection of Attack A_2, whose datapoint holds a non-zero value for ≈ 50 seconds (as shown in Fig. 3), can experience diminished or even halved detection performance when the time batcher window size is set to an excessively coarse value. This performance degradation occurs because once the window size surpasses the timings associated with the observed dynamics, most measurements get grouped in a few time clusters, making it highly challenging for the anomaly detection solution to model the expected evolution of a given variable over time.

4.2.3. Intellectual property protection

Regarding the concerns raised in Section 3.1, we recall the four example queries to validate the capability of our methodology in protecting intellectual property. In particular, we conducted queries before and after anonymization, comparing them with the values in the recipe. Our goal was to determine the precision of reverse engineering by an actor accessing the SIEM system. Below, we present results for each query following the application of MBFloat^{-1} (the inverse of Equation (4)) with correct values for V_{min} and V_{max}. Assuming that attackers know the correct values of V_{min} and V_{max} represents a worst-case scenario for the ModBus protocol, as it presumes attackers know the mapping used for floating-point values (see Section 4.1). However, this assumption is a generalization applying to other industrial protocols where no such transformation occurs.

Q1 This query involves reconstructing the portioning among ingredients. In the recipe, the portioning is[1] 57.17% for tomatoes, 2.86% for Tabasco sauce, and 40% for peppers. Table 5 summarizes the measured portioning (*Ratio*) and percentage deviation (Δ) w.r.t. the actual recipe for the non-anonymized (Table 5a), anonymized (Table 5b), and anonymized with $k = 1$ (Table 5c) data flows. When looking at the SIEM for the non-anonymized data flow, an attacker could measure the portioning as 400.7 kg for tomatoes, 20.05 kg for Tabasco, and 280.19 kg for peppers. When these quantities are converted into ratios, they correspond to a recipe of 57.17% for tomatoes, 2.86% for Tabasco sauce, and 39.97% for peppers. An error of, respectively, 0.02%, 0.0%, and 0.03%, corresponds to the plant's precision in following the recipe and the measurement errors from the load cells. Instead, when anonymization is introduced, attackers measure a ratio of 16.55% tomatoes, 69.78% Tabasco, and 13.68% peppers. This portioning differs from the recipe by 40.59%, 66.92%, and 26.32%. Finally, when the anonymization system is constrained with the $k = 1$ condition, the measured ratio is 26.44%, 41.77%, and 31.79%, with the error decreasing to 30.71%, 38.92%, and 8.21%.

Q2 This query aims to recover the mixture's cooking temperature during the various stages of the recipe. For example, during the *Initial Heating* phase, the recipe specifies a cooking temperature of 62 °C. When queried, the SIEM containing the original data stream

[1] We round all percentages in this section to two digits after the decimal point.

measures 62.09 °C, enabling attackers to recover the mixture temperature from that data accurately. Running the same query on the anonymized data flow would yield a nonsensical temperature of $2310.45\,°C$. Assuming that capable attackers could quickly discover this deception, they might attempt to reverse engineer the temperature quantization by first determining the measure precision, which involves dividing the ambient temperature (20 °C) by x_{min}. This calculation reveals that x can be multiplied by $\frac{20}{x_{min}} = \frac{20}{8738} \approx 0.0023\,°C$ to recover the real temperature value. Then, in order to reconstruct the cooking temperature they can feed x_{max} to this function. In the case of the original non-anonymized data flow, this procedure calculates a cooking temperature of approximately $\frac{20}{8738} \cdot 27218 \approx 62.3\,°C$. However, even this more sophisticated method fails once anonymization is applied, yielding the erroneous values $56.8\,°C$ and $35.7\,°C$ (with the $k = 1$ constraint).

Q3 This query estimates the number of packages sent out for sale based on the number of distinct levels in the final packaging weight funnel. Since the transformations performed within the anonymization system do not alter the value dynamics, the original and anonymized data flows allow the reconstruction of the number of packages produced (1000).

Q4 In this query, which merges the previous one with the weight identified in the funnel, the actual process size of 0.7 kg matches the resulting weight of 0.702 kg without anonymization. However, in the anonymized data flow, the estimated carton size becomes either 3.68 kg or 9.5 kg (when $k = 1$). This discrepancy occurs because, despite the correct number of cartons being found with **Q3**, the transformation applied to the weight value renders the result inaccurate.

Finally, Fig. 7a displays data collected over time from the SIEM, similar to Fig. 3 but with a differing Y-axis scale. Figs. 7b and 7c illustrate the data within the same time frame following the application of both unconstrained and constrained transformations without the use of time batching. Both techniques preserve the original data stream overall shape, although with altered value ranges. Additionally, the unconstrained anonymization plot shows that the k multiplication factor has altered the amplitude of each variable oscillation.

4.2.4. Performance

The experiments run using the anonymization logic on a 14-core Intel i9-13900H processor. Anonymization of the nominal network traffic of the entire plant took between 5% (minimum) and 6.7% (maximum) of a single CPU core, with an average utilization of 5.9%. Memory usage remained between 40 and 50 MB, averaging 45.18MB.

As the plant network traffic did not fully utilize the system maximum throughput, we ran a dedicated performance testing run consisting of 1000 loops of a pre-recorded 22531284 packet (6340MiB) data stream. These tests yielded minimum, average, and maximum throughput figures of 306.01, 308.33, and 310.117 Megabits per second per core, respectively. These per core figures correspond to an overall average throughput over 4.3 Gigabits per second when all CPU cores are active. Regarding packets per second, the throughput remained at a minimum of 0.537, an average of 16.433, and a maximum of 32.32 Megapackets per second per core.[2]

4.3. Discussion

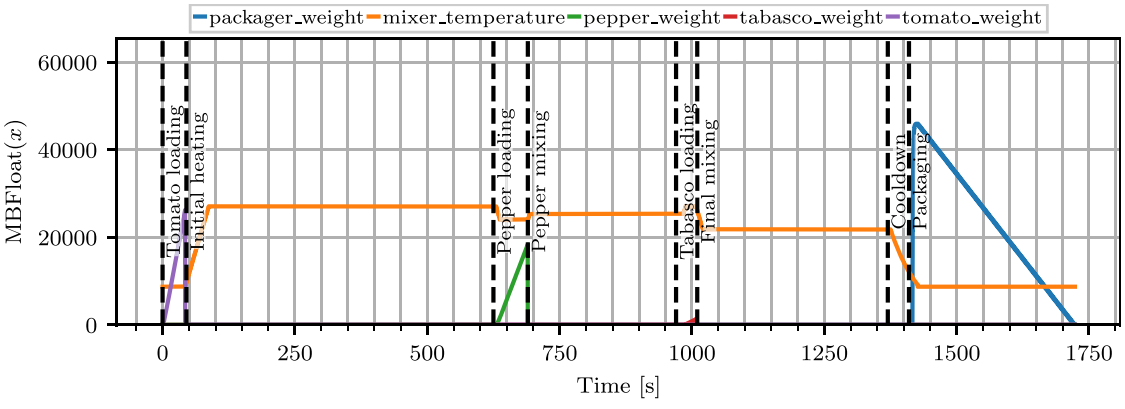
We experimentally evaluate the correspondence of our proposed solution with the requirements outlined in Section 3.2 using the results presented in the previous section. With regard to **R1**, the performance figures of the system indicate that it can support live anonymization for a network with multi-gigabit throughput, thanks to its low memory usage and inherent parallelism achievable when processing network data flows. As a result, this solution should run well on most Linux-based network operating systems available for installation on switching hardware. As far as **R2** is concerned, the analysis above demonstrates that when the time batcher period is set according to the time constants associated with the dynamics of the monitored system, the performance of the attached anomaly detectors remains unaffected by the presence of the system, thus fulfilling this requirement. **R3** is fulfilled by design due to tokenization mapping, which always allows reconstructing the original identities found in anomalies and the possibility of using T^{-1} to recover the original values found in the data. Lastly, **R4** is entirely satisfied by our system as it does not require any modifications to the plant or probes; its operation is functionally equivalent to a normal bridge SPAN port, except for the added anonymization features.

4.3.1. Limitations

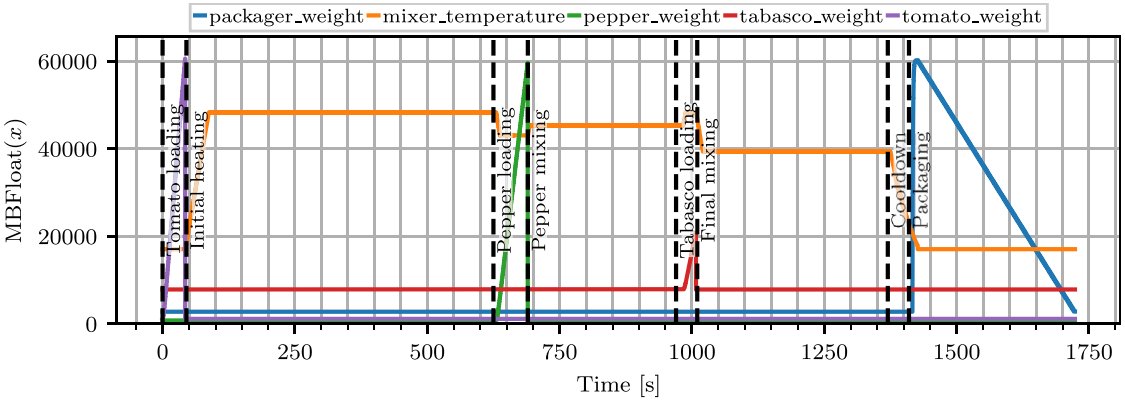
Although the proposed system supports all the desirable attributes mentioned, its implementation may face challenges depending on application context. For instance, the system discussed in this paper works with an unencrypted protocol that is conveniently auditable and modifiable from the network device. Modifying the system to function with encrypted traffic might require the deployment of middleboxes that break encryption with architectures similar to the one described in [27]. Furthermore, implementing this solution for complex protocols may require significant effort. This effort may require handling protocols and ongoing tasks such as inspecting, anonymizing, and rewriting their content.

Finally, since this solution overrides the observed dynamics detected by the instrumentation probes, its deployment should be initiated from scratch or involve resetting the already observed data. This is because the modifications introduced by the anonymization system render the two data sets incompatible. The transition could also be smoothed out by resetting the anomaly detector status, retraining it on an anonymized copy of its historical data, and reattaching the system. Suppose that the underlying industrial process

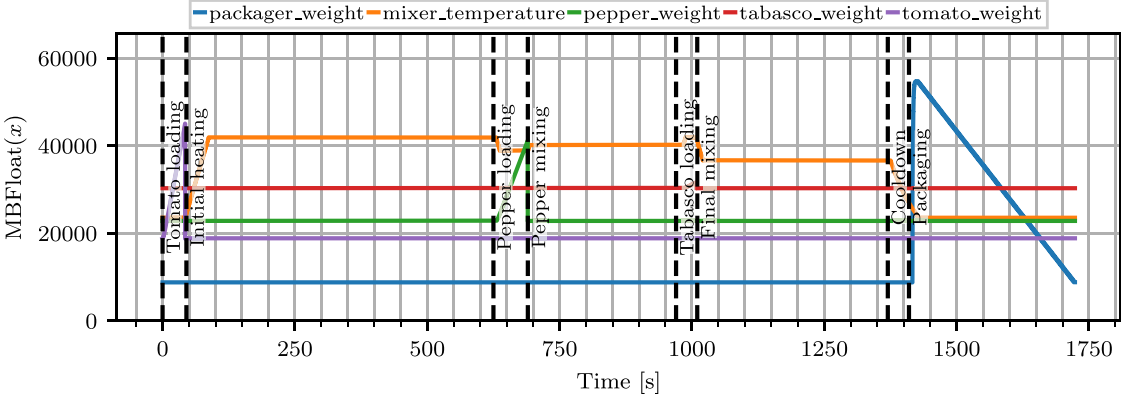
[2] 10^6 packets processed for each available CPU core.



(a) No anonymization.



(b) Anonymized according to Table 3a.



(c) Anonymized according to Table 3b.

Fig. 7. Effect of the anonymization on the SIEM measurements.

remains unchanged. In that case, the data collected from the security system before such a change will still contain the intellectual property that our proposal intends to protect.

5. Related work

Our work specifically focuses on protecting sensitive data in industrial contexts. Related work includes both techniques explicitly tailored for industrial environments and broader data privacy methods. Although the latter approaches are mainly designed to hide personal sensitive information, many underlying strategies are also suitable for protecting industrial information.

Several approaches have already been proposed in the literature to design protection mechanisms for industrial processes while maintaining the confidentiality of sensitive data [28,29]. These approaches include methods such as data obfuscation and perturbation [30,31], data anonymization [32,33], limiting information sharing [34] and the use of cryptographic solutions [35,36]. Mostly,

these approaches are geared towards protecting structured data, such as those found in traditional SQL databases [37]. However, the digital transformation of industrial activities has increased the amount and complexity of the data that are processed. As a result, existing privacy protection methods and schemes are often inadequate, particularly in scenarios where a more dynamic and immediate response to privacy and security threats is required [38].

Some research works address the problem of processing sensitive data through SIEM systems. For instance, Menges et al. [39] propose a GDPR-compliant SIEM architecture that employs pseudonymization and encryption techniques to protect data throughout the entire processing lifecycle, from collection to incident reporting. However, the authors highlight that threat identification performance can be impaired due to the loss of data granularity from pseudonymization. Moreover, the reliance of the architecture on specific cryptographic methods and its focus on GDPR compliance may limit its adaptability across diverse industrial contexts and existing SIEM infrastructures, as these methods destroy the original semantics of the data. In contrast, our approach uses real-time anonymization, which prevents data re-identification while guaranteeing a suitable level of utility and providing greater protection against exposure.

Vazão et al. [40] propose a GDPR-compliant SIEM system based on the Elastic Stack. They experimentally demonstrate that their solution achieves good performance and scalability in a real-world environment. It should be noted that, analogously to the approach by Menges et al., there are no guarantees that data cannot be re-identified in this case. Moreover, the dependency on specific technologies like the Elastic Stack and related plugins and the need for particular configurations and policies across various industrial contexts present limitations. Our solution, conversely, has no impact on SOC operations and infrastructures, providing a more adaptable and less intrusive integration.

Coppolino et al. [41] focus on enabling privacy-preserving managed security services by using hardware-assisted Trusted Execution Environments (TEE) and Homomorphic Encryption (HE). The authors argue that while this approach offers high security for specific use cases with stringent data privacy requirements, such as healthcare or financial services, it may not scale quickly across different size scenarios due to hardware requirements and the computational cost of HE. This is hardly surprising given the computational limitations of the HE algebra. Our solution, in contrast, balances data protection and operational needs without relying on highly computational techniques such as HE that preserve original data statistics but affect their original semantics, deeply affecting their utility. This makes it suitable for various industrial applications.

Larrucea et al. [42] define guidelines to support privacy in the software development process by modifying the ISO/IEC 29110 [43] profile. In addition, they propose an anonymization strategy based on differential privacy. The differential privacy implemented adds noise sampled from a Laplace distribution to the data, making them different from the original by a predefined constant. However, adding white noise to the data has proved weak because it can be easily filtered [44]. Varanda et al. [45] propose a solution to pseudonymize log data by substituting personal data. The proposed solution aggregates different data on a central server and substitutes only sensitive tokens with randomly generated ones, like personal identifiers or query searches. This solution impacts the utility of the single anonymized data but does not affect its distribution, meaning the privacy level of the overall distribution remains the same.

Fahad et al. [46] propose a framework to protect SCADA system data using a clustering methodology to alter sensitive information based on network traffic types. Moreover, a privacy-preserving approach that emphasizes data aggregation to mitigate risks of unauthorized access to sensitive information has been proposed in [47]. Finally, a framework for cross-plant process monitoring has been introduced in [48], leveraging federated learning to facilitate the exchange of model parameters instead of raw data, thus enhancing data protection across manufacturing systems. Our approach stands out from the aforementioned studies as it focuses on process data anonymization and directly addresses the problem of real-time data anonymization while preserving the process's non-intrusiveness and reversibility. Furthermore, our solution is compatible with current industrial environments and security solutions. It does not require modifications to operational protocols or existing security infrastructure, thus providing a transparent integration. This completely differs from previous work proposals that require significant alterations to existing systems or processes. Finally, by presenting a case study and the results of a thorough experimental activity, we stress that our paper not only proposes a theoretical solution but also demonstrates its applicability and effectiveness in a real-world scenario.

6. Conclusions

This paper presented a methodology for real-time traffic anonymization in industrial networks, addressing the balance between data protection and security in SOC monitoring. Through a detailed and realistic case study, we first revealed how SIEM systems and the data they store for anomaly detection can be an issue in protecting sensitive data or industrial secrets from data leakage. We then demonstrated that data anonymization can be used as a countermeasure. Our solution works in real-time within the existing IT/OT infrastructure, seamlessly integrating with the technological stack and security solutions without compromising anomaly detection capabilities. Experimental results exhibited its effectiveness and high throughput. In future work, we plan to further enhance and validate our methodology by applying it to additional industrial protocols such as OPC-UA. Additionally, we intend to enhance the capabilities of the range updater with more transformations, increasing its adaptability and effectiveness across various data types and scenarios.

CRediT authorship contribution statement

Giacomo Longo: Writing – original draft, Methodology, Formal analysis, Conceptualization. **Francesco Lupia:** Writing – original draft, Data curation, Conceptualization. **Alessio Merlo:** Writing – original draft, Investigation. **Francesco Pagano:** Validation, Conceptualization. **Enrico Russo:** Writing – original draft, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially funded by the NextGenerationEU project “Security and Rights in CyberSpace” SERICS (PE00000014). It was carried out while Giacomo Longo was enrolled in the Italian National Doctorate on Artificial Intelligence run by the Sapienza University of Rome in collaboration with the University of Genoa.

Data availability

Data will be made available on request.

References

[1] International Organization for Standardization, ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection–Information security management systems–Requirements, International Organization for Standardization, 2022.
[2] C. Pascoe, S. Quinn, K. Scarfone, The NIST Cybersecurity Framework (CSF) 2.0, NIST Cybersecurity White Papers (CSWP), National Institute of Standards and Technology, Gaithersburg, MD, 2024, https://doi.org/10.6028/NIST.CSWP.29, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=957258.
[3] R. Piggin, Development of industrial cyber security standards: IEC 62443 for scada and industrial control system security, in: IET Conference on Control and Automation 2013: Uniting Problems and Solutions, Institution of Engineering and Technology, 2013.
[4] M. Vielberth, F. Bohm, I. Fichtinger, G. Pernul, Security operations center: a systematic study and open challenges, IEEE Access 8 (2020) 227756–227779, https://doi.org/10.1109/access.2020.3045514.
[5] S. Bhatt, P.K. Manadhata, L. Zomlot, The operational role of security information and event management systems, IEEE Secur. Priv. 12 (5) (2014) 35–41, https://doi.org/10.1109/msp.2014.103.
[6] G. Bloom, B. Alsulami, E. Nwafor, I.C. Bertolotti, Design patterns for the industrial Internet of things, in: 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), IEEE, 2018, pp. 1–10.
[7] T.J. Williams, The Purdue enterprise reference architecture, Comput. Ind. 24 (2–3) (1994) 141–158, https://doi.org/10.1016/0166-3615(94)90017-5.
[8] D. Deshpande, Managed security services: an emerging solution to security, in: Proceedings of the 2nd Annual Conference on Information Security Curriculum Development, InfoSecCD05, ACM, 2005, pp. 107–111.
[9] D.M. Thomas, N. Pandey, V.K. Shukla, A.V. Singh, Attack vectors and susceptibilities of the modbus in tcp/ip model, in: 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), IEEE, 2021, pp. 1–5.
[10] C. Parian, T. Guldimann, S. Bhatia, Fooling the master: exploiting weaknesses in the modbus protocol, Proc. Comput. Sci. 171 (2020) 2453–2458, https://doi.org/10.1016/j.procs.2020.04.265.
[11] B. Chen, N. Pattanaik, A. Goulart, K.L. Butler-purry, D. Kundur, Implementing attacks for modbus/tcp protocol in a real-time cyber physical system test bed, in: 2015 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR), IEEE, 2015, pp. 1–6.
[12] P. Huitsing, R. Chandia, M. Papa, S. Shenoi, Attack taxonomies for the modbus protocols, Int. J. Crit. Infrastruct. Prot. 1 (2008) 37–44, https://doi.org/10.1016/j.ijcip.2008.08.003.
[13] T0831-manipulation of control, Available from MITRE, https://attack.mitre.org/techniques/T0831/, 2024.
[14] T0889-modify program, Available from MITRE, https://attack.mitre.org/techniques/T0889/, 2024.
[15] J. McCarthy, M. Powell, K. Stouffer, C. Tang, T. Zimmerman, W. Barker, T. Ogunyale, D. Wynne, J. Wiltberger, Securing Manufacturing Industrial Control Systems: Behavioral Anomaly Detection, National Institute of Standards and Technology, 2020, https://doi.org/10.6028/nist.ir.8219.
[16] L. Rosa, T. Cruz, M.B.d. Freitas, P. Quitério, J. Henriques, F. Caldeira, E. Monteiro, P. Simões, Intrusion and anomaly detection for the next-generation of industrial automation and control systems, Future Gener. Comput. Syst. 119 (2021) 50–67, https://doi.org/10.1016/j.future.2021.01.033.
[17] A. Bécue, I. Praça, J. Gama, Artificial intelligence, cyber-threats and industry 4.0: challenges and opportunities, Artif. Intell. Rev. 54 (5) (2021) 3849–3886, https://doi.org/10.1007/s10462-020-09942-2.
[18] Internet Control Message Protocol, RFC 792 (Sep. 1981), https://doi.org/10.17487/RFC0792, https://www.rfc-editor.org/info/rfc792.
[19] R. Kumar, J. Novak, B. Pang, A. Tomkins, On anonymizing query logs via token-based hashing, in: Proceedings of the 16th International Conference on World Wide Web, 2007, pp. 629–638.
[20] G. Longo, A. Orlich, S. Musante, A. Merlo, E. Russo, Macyste: a virtual testbed for maritime cybersecurity, SoftwareX 23 (2023) 101426, https://doi.org/10.1016/j.softx.2023.101426.
[21] G. Longo, F. Lupia, A. Pugliese, E. Russo, Physics-aware targeted attacks against maritime industrial control systems, J. Inf. Secur. Appl. 82 (2024) 103724, https://doi.org/10.1016/j.jisa.2024.103724.
[22] G. Thomas, Introduction to the modbus protocol, Extension 9 (4) (2008) 1–4.
[23] A. Swales, et al., Open modbus/tcp specification, Schneider Electr. 29 (3) (1999) 19.
[24] Packet mmap, The Linux kernel documentation, https://docs.kernel.org/networking/packet_mmap.html, 2024.
[25] A. AlSabeh, E. Kfoury, J. Crichigno, E. Bou-Harb, Leveraging sonic functionalities in disaggregated network switches, in: 2020 43rd International Conference on Telecommunications and Signal Processing (TSP), IEEE, 2020, pp. 457–460.
[26] S. Guha, N. Mishra, G. Roy, O. Schrijvers, Robust random cut forest based anomaly detection on streams, in: International Conference on Machine Learning, PMLR, 2016, pp. 2712–2721.
[27] European Telecommunications Standards Institute, TS 103 523-3 - CYBER; Middlebox Security Protocol; Part 3: Enterprise Transport Security, version 1.3.1, 8 2019.
[28] N. Moustafa, B. Turnbull, K.-K.R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of things, IEEE Int. Things J. 6 (3) (2019) 4815–4830, https://doi.org/10.1109/JIOT.2018.2871719.
[29] B. Zhao, K. Fan, K. Yang, Z. Wang, H. Li, Y. Yang, Anonymous and privacy-preserving federated learning with industrial big data, IEEE Trans. Ind. Inform. 17 (9) (2021) 6314–6323, https://doi.org/10.1109/TII.2021.3052183.
[30] C. Dwork, Differential privacy: a survey of results, in: Theory and Applications of Models of Computation, 2008, https://api.semanticscholar.org/CorpusID:2887752.
[31] R. Shokri, V. Shmatikov, Privacy-preserving deep learning, in: CCS ’15, Association for Computing Machinery, 2015.

[32] C.C. Aggarwal, On k-anonymity and the curse of dimensionality, in: VLDB '05, VLDB Endowment, 2005, pp. 901–909.
[33] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, A. Zhu, Anonymizing tables, in: T. Eiter, L. Libkin (Eds.), Database Theory - ICDT 2005, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 246–258.
[34] R. Shokri, G. Theodorakopoulos, P. Papadimitratos, E. Kazemi, J.-P. Hubaux, Hiding in the mobile crowd: locationprivacy through collaboration, IEEE Trans. Dependable Secure Comput. 11 (3) (2014) 266–279, https://doi.org/10.1109/TDSC.2013.57.
[35] K. Gai, M. Qiu, H. Zhao, J. Xiong, Privacy-aware adaptive data encryption strategy of big data in cloud computing, in: 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), 2016, pp. 273–278.
[36] B. Pinkas, Cryptographic techniques for privacy-preserving data mining, ACM SIGKDD Explor. Newsl. 4 (2) (2002) 12–19, https://doi.org/10.1145/772862.772865.
[37] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, H. Qi, Beyond inferring class representatives: user-level privacy leakage from federated learning, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, IEEE Press, 2019, pp. 2512–2520.
[38] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, Z. Lin, When machine learning meets privacy: a survey and outlook 54 (2) (2021), https://doi.org/10.1145/3436755.
[39] F. Menges, T. Latzo, M. Vielberth, S. Sobola, H.C. Pöhls, B. Taubmann, J. Köstler, A. Puchta, F. Freiling, H.P. Reiser, G. Pernul, Towards gdpr-compliant data processing in modern siem systems, Comput. Secur. 103 (2021) 102165, https://doi.org/10.1016/j.cose.2020.102165, https://www.sciencedirect.com/science/article/pii/S0167404820304387.
[40] A.P. Vazão, L. Santos, R.L. de, C. Costa, C. Rabadão, Implementing and evaluating a gdpr-compliant open-source siem solution, J. Inf. Secur. Appl. 75 (2023) 103509, https://doi.org/10.1016/j.jisa.2023.103509, https://www.sciencedirect.com/science/article/pii/S2214212623000935.
[41] L. Coppolino, S. D'Antonio, G. Mazzeo, L. Romano, L. Sgaglione, Prisiem: enabling privacy-preserving managed security services, J. Netw. Comput. Appl. 203 (2022) 103397.
[42] X. Larrucea, I. Santamaria, Dealing with privacy for protecting information, in: Systems, Software and Services Process Improvement: 28th European Conference, EuroSPI 2021, Krems, Austria, September 1–3, 2021, Proceedings 28, Springer, 2021, pp. 518–530.
[43] Iso/iec 29110-4-1, Available from iso.org, https://www.iso.org/standard/67223.html, 2018.
[44] H. Kargupta, S. Datta, Q. Wang, K. Sivakumar, On the privacy preserving properties of random data perturbation techniques, in: Third IEEE International Conference on Data Mining, 2003, pp. 99–106.
[45] A. Varanda, L. Santos, R.L.d.C. Costa, A. Oliveira, C. Rabadão, The general data protection regulation and log pseudonymization, in: International Conference on Advanced Information Networking and Applications, Springer, 2021, pp. 479–490.
[46] A. Fahad, Z. Tari, A. Almalawi, A. Goscinski, I. Khalil, A. Mahmood, Ppfscada: privacy preserving framework for scada data publishing, Future Gener. Comput. Syst. 37 (2014) 496–511, https://doi.org/10.1016/j.future.2014.03.002.
[47] M.A. Ferrag, L. Maglaras, H. Janicke, J. Jiang, A survey on privacy-preserving schemes for smart grid communications, arXiv:1611.07722 [abs], 2016, https://api.semanticscholar.org/CorpusID:2663804.
[48] K. Wang, Z. Song, High-dimensional cross-plant process monitoring with data privacy: a federated hierarchical sparse pca approach, IEEE Trans. Ind. Inform. 20 (3) (2024) 4385–4396, https://doi.org/10.1109/TII.2023.3323685.