Cybersecurity Challenges in Large Industrial IoT Systems

Björn Leander[‡†], Aida Čaušević[†] and Hans Hansson[†]
[‡] ABB Industrial Automation, Process Control Platform,
[†] Mälardalen University,
Västerås, Sweden
bjorn.leander@se.abb.com, {aida.causevic, hans.hansson}@mdh.se

***Abstract*—To achieve efficient and flexible production at affordable prices, industrial automation is pushed towards a digital transformation. Such a transformation assumes an enhancement of current Industrial Automated Control Systems with a large amount of IoT-devices, forming an Industrial Internet of Things (IIoT). The aim is to enable a shift from automatic towards autonomous control in such systems. This paper discusses some of the main challenges IIoT systems are facing with respect to cybersecurity. We discuss our findings in an example of a flow-control loop, where we apply a simple threat model based on the STRIDE method to deduce cybersecurity requirements in an IIoT context. Moreover, the identified requirements are assessed in the light of current state of the art solutions, and a number of challenges are discussed with respect to a large-scale IIoT system, together with some suggestions for future work.**

I. Introduction

The manufacturing industry is going through a rapid evolution driven by the Internet technology applied in the industrial context. The paradigm shift is known as *Industry 4.0* in Europe and *Industrial Internet* in the USA. A common belief is that an emerging Industrial Internet of Things (IIoT) will provide optimization, cost-savings, and new business opportunities in several domains. According to the Industrial Internet Consortium (IIC) [1], an IIoT system will enable significant advances in optimizing decision-making, operations and collaborations among a large number of increasingly autonomous control systems. Big-data analysis using data from smart production equipment and smart products might for example provide intelligence for decision making. According to the IEC [2], a fundamental purpose of Industry 4.0 is to enable cooperation and collaboration between devices.

As described by Madsen [3], the trustworthiness of an information system is the degree of confidence that it performs as expected with respect to key characteristics during unexpected scenarios, such as: disruptions from the environment, human errors, system faults, and attacks from adversaries. An IIoT system will as well be judged based on its trustworthiness. The correct implementation of cybersecurity in an IIoT system will be one of the driving factors for its success, increasing its trustworthiness in several aspects such as: quality and integrity of information, asset availability, etc. However, many of the devices in an IIoT system will be resource constrained with regards to computational power, network bandwidth, etc., while there at the same time may be real-time requirements on signal handling. This combination of constraints and requirements yields unique challenges related to cybersecurity, as the traditional cryptographic methods add significant load both on network and CPU utilization.

Hermann et al. [4] describe the central design principles for Industry 4.0 as being: *1)* interconnection, *2)* technical assistance, *3)* decentralized decisions and *4)* information transparency. In this paper we mainly focus on interconnection, since reliable communication between devices in a heterogeneous environment is a fundamental requirement for enabling the remaining design principles.

The main contributions of our work are twofold: *1)* to uncover a number of cybersecurity related challenges in large-scale IIoT systems for which the current state of the art solutions need further improvements to be applicable, and *2)* presenting possible directions for future solutions for some of the more important of these challenges. We do this by applying the industry approved Microsoft STRIDE [5] threat modelling method on a number of typical scenarios in a simple example. From the resulting threat model, information regarding cybersecurity threats related to an IIoT system are discussed.

The paper is organised as follows. Section II introduces necessary background and concepts used in this paper. In Section III a working example is introduced Section IV expands the view to an IIoT system, including a threat model for the example based on STRIDE model, along with state of the art solutions for common countermeasures. In Section V we discuss challenges for a large-scale IIoT system from a cybersecurity perspective, while related works are described in Section VI. We present concluding remarks and outline directions for future work in Section VII.

II. Background

An IIoT system connects and integrates industrial control systems with enterprise systems, business processes and analytics. Boyes et al. [6] provide a more exhaustive definition of an IIoT system, based on a survey of existing definitions. This definition emphasize IIoT as a means for optimising overall production value. There exist several reference architectures related to IIoT, the most notable ones are: *Reference Architecture Module for Industry 4.0* (RAMI4.0) [2] suggested by IEC/PAS, and *Industrial Internet of Things Infrastructure* [7] suggested by the IIC.

For large scale IIoT applications, the complexity of the information infrastructure depends on:

1) ***System Size*** - In a factory or process industry there will be potentially many thousands or even millions of IIoT devices.
2) ***Composite devices*** - Complex devices will be composed of a number simpler devices, e.g., a smart mine hoist will consist of smart motors, transmission systems, brakes, sensors, etc.
3) ***Thing-to-Cloud Continuum*** - Different services related to specific devices or specific functions will exist anywhere from the device through edge nodes concentrating data to cloud nodes that collect and analyse data. For each device there could be any number of edge-, and cloud-nodes hosting related services, which will require communication and trust across organization boundaries in many applications.
4) ***Heterogeneous technologies*** - Many different manufacturers and industries using different technologies will implement and use these devices. At the same time, the devices are expected to be able to communicate with other devices and services along the thing-to-cloud continuum when needed. Interoperability between devices and services will be a paramount.
5) ***Multiple stakeholders*** - Different stakeholders will have interest in the devices, including device owner, device manufacturer, maintenance responsible, etc.
Therefore a large-scale IIoT system has advanced requirements on the information infrastructure. It will become an important task to address different levels of integration required in an IIoT infrastructure, as described in [8]:

1) Cross-technology integration of smart devices from different suppliers;
2) Cross-organization integration of information and services from different enterprises;
3) Cross-domain integration of business ecosystems from different industries.

Cybersecurity is the protection of a computer system from unauthorized actors' possibility to: steal or alter information in the system, disrupt or alter behaviour of a function or perform an unauthorized action [9]. Cybersecurity is seen as a cross-cutting concern of an IIoT system [1], as a system is not more secure than its weakest link, and any potential attack surface must be considered.

The STRIDE (**S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of Service and **E**levation of privilege) threat model is a method for classifying threats in an information system introduced by Microsoft [5]. It includes defining security zones in a data-flow diagram for the system, checking any security-zone interactions and then enumerating any threat per class for that interaction. For each threat, countermeasures are suggested and assessed. The use of STRIDE for threat modeling in IIoT has already been discussed in the literature [1], referring to an extension of STRIDE for the Azure IoT reference architecture [10], described by Shahan et. al [11]. Other possible methods for threat modeling could be considered, such as CVSS [12], PASTA [13], etc. As STRIDE is commonly used in industry it was selected for this work.

In this paper the focus is on a class of assets that in RAMI4.0 is defined as an *entity*, being an uniquely identifiable asset that has a digital world representation. *Device* is in this context used interchangeable with entity. Focus is on information, however the devices may be used for sensing or actuating in the physical world. Such devices in combination with their software are realized as Cyber Physical Systems (CPS), which share a number of characteristics differentiating them from traditional IT-Systems. The main difference being that actions for a CPS in the information-world can have direct real-world implications [14].

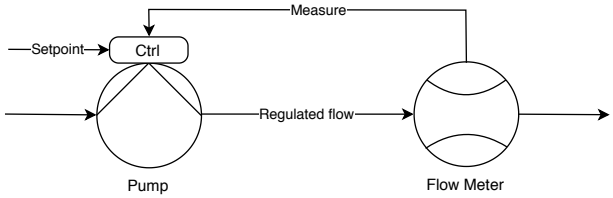

Fig. 1: Flow-control loop

III. A WORKING EXAMPLE

In this section we introduce a flow-control loop process as an example used to illustrate and derive challenges in the following work. It is chosen as being one of the simplest realistic control loops, common in industrial applications. The loop consists of a pump with built-in control logic that is regulating the flow through a pipe. The feedback is provided from a flow meter mounted in the pipe, see Fig. 1. This view of the process is only one example of a high-order integrated view of the control logic for the pump. Several other aspects exist for the pump in a practical industrial application, e.g., CAD drawing, location, current I/O values, status, graphics, maintenance log, I/O-value history, etc. In an Industrial Automated Control System (IACS), these aspects are usually accessible in some way, but not always in the same view or related to the same identity.

The presented example can be seen as a CPS, and in section IV, we put the example into the context of a large IIoT system. In a CPS, cybersecurity attacks on the system might have physical-world implications, e.g., loss of control of the pump could harm the process and potentially pose threats to humans working in vicinity to the process, or to the environment, depending on the function of the system and additional safety measures supplied in the IACS.

Here we focus on three scenarios related to the flow-control loop from the perspective of a traditional Industrial Automation and Control System (IACS) (i.e., a homogeneous environment where different actors communicate using the same protocols, have a common identification nomenclature, and live within the same network). The scenarios are chosen as being typical events in an industrial application.

A. Scenario 1 - Displaying a trend curve

An engineer wants to access current I/O-values from the pump and flow-meter in order to draw a trend-curve diagram to be displayed in a control room. The actions needed are: *1)* identify the pump and flow meter, *2)* check that there exist a service able to deliver relevant data for the respective device, *3)* use the service(s) to read the data, and *4)* display the trend-curve to the operator.

B. Scenario 2 - Replacing the pump device

In this scenario the pump in the flow control loop needs to be replaced due to some malfunction. To execute the scenario the required life-cycle actions of the old and new devices must be satisfied, including: *1)* a new pump must be acquired, *2)* the logical replacement in the IT-system, *3)* the physical replacement is executed by a technician on site, *4)* configuration of the new device must be performed so that it delivers the same functionality as the old pump. If there is substantial difference in functionality between the old and new devices, some services may need to be added or modified, e.g., the control logic could be implemented in a PLC if missing in the new pump-device.

C. Scenario 3 - Replace software in pump device

The pump manufacturer has discovered a fault or weakness in the current software version running on the pump device, requiring a patch being applied to resolve the issue. The scenario is executed in the following steps: *1)* the manufacturer of the pump notifies the plant organisation about a new patched software version for the pump-device, *2)* the patch is distributed to a maintenance technician, *3)* within a time-slot for planned maintenance a technician updates the pump device software.

IV. A Threat Model from an IIoT perspective

Let us assume that the described example is a part of an IIoT system. Any aspect of the pump described in Section III could be represented by a separate service in this context. For example, a CAD-drawing related to the pump-device could be stored as a pdf-file directly in the device, accessible from a Product Life-cycle Management (PLM) system in the process owner's IT-network, or available at the pump manufacturer web-site. In this way each device may be related to any number of services with endpoints distributed through the device-to-cloud continuum.

Scenarios 1 and 2, described above, will be performed in very much the same way in the IIoT perspective, only the environment will differ. For example, in Scenario 1, different services for I/O-data and for the trend curve might have end-points in different security zones, communicate with different protocols, etc.

Scenario 3 on the other hand might differ substantially when performed in an IIoT system. The manufacturer might have direct access to some services related to the pump, e.g., a service containing information on current software version. The pump-device could have direct access to a service publishing new software revisions. Assuming previously described, the scenario will be executed as follows: *1)* the manufacturer publishes a new revision of the software containing the patch, *2)* then triggers the pump to perform an update, alternatively the pump-device could cyclically check for availability of updates, and finally *3)* the pump will download and perform the update automatically at a convenient time-slot.

Assuming an IIoT system setup, every device and service should be treated as being placed in separate security zones. In threat modeling every interaction crossing a security-zone boundary must be analyzed. A simple threat model for scenarios 1-3 using the STRIDE classification method is presented in Table I. From this model, a number of additional requirements for devices and services that are part of an IIoT system can be deduced.

Note that the threat model is abstracted, a number of additional technical details should be accounted for when analyzing these scenarios in a system with specific protocols, operating systems, etc. Some aspects have been intentionally left out, e.g., physical security, threats from social engineering, etc., as they are not of interest for this work.

The additional security requirements, as listed in the threat model, can be sorted out based on the level of responsibility needed for their implementation:
1) ***Service:*** integrity and encryption of Data at Rest (DAR), hardening, resource limitation of unauthorized inbound connections, parameter bound checks, auditing.
2) ***Device:*** integrity and encryption of DAR, secure boot, function for purge of sensitive data, tamper free storage, anti-malware software, service sandboxing.
3) ***Organization:*** policies on actions to take when provisioning and decommissioning devices.
4) ***Infrastructure:*** identification, authentication and authorization of devices, services, users, integrity and encryption of Data in Motion (DIM) including forward/backward security, malware detection, audit log monitoring, intrusion detection systems (IDS).

When considering a large scale IIoT system as described in Section II, requirements related to the infrastructure are likely to be the most challenging ones. Therefore, in the following we assess different countermeasures deduced from the threat model and related to infrastructure, and enumerate their related state of the art or best-practice solutions.

A. Identification

In Scenario 1 applied to an IIoT system, the trend-service must be able to identify the pump and flow-meter to find service end-points for receiving I/O-data for the devices. It is reasonable that actors communicating in any way must be able to deduce the identity of each other. In Scenario 2, the physical pump device is replaced, so there is a need to propagate changed identity to dependent actors, or update mapping between identities in different name-spaces so that the system as a whole retain its functionality.

Radio Frequency Identification (RFID) [15] is used as means for contact-less transfer of identity in several IIoT applications, e.g., in logistics. In network technology, MAC addresses may be used to uniquely identify an Ethernet card. In software technologies, Global Unique Identifiers (GUID) are often used for identifying different entities. Serial numbers and physical addresses could also be used for identification. An entity may hold several unique identities that are relevant for different actors. To interact, each actor must have knowledge of at least one of the other entity identities, and there must be a well-known and trusted method for translation between different name-spaces.

B. Authentication

None of the identification schemes described in the previous section provide proof of identity per se. RFID technology

Classification	Scenario	Threat	Counter measure
Spoofing	1,3	Service-endpoint spoofed - impersonation attack.	Identification and authentication of service endpoints.
	2	New Device is counterfeit.	Integrity of type-identification, policy on verification of authenticity of purchased products.
	1,3	Replay attack intended to trick a service to e.g., leak information.	Using e.g., session tokens to invalidate old messages.
Tampering	1,3	DIM tampered.	Integrity of DIM.
	1	DAR tampered	Integrity of DAR, tamper-free storage.
	2	Software of new device tampered.	Malware detection.
	3	Patch tampered during transfer.	Integrity check of patch before being applied, malware detection.
Repudiation	1,3	Device/Service claiming not received data/patch.	Audit log for accessing data.
	1,3	Device/Service claiming not sending data/patch.	Audit log for sending data.
	1,3	Actor claiming not attempting to access restricted information.	Audit log for failed access attempts.
Information Disclosure	1	Information intercepted and relayed to unintended receiver.	Encryption of DIM.
	2	Decommissioned device contains retrievable sensitive information.	Encryption of DAR, purge of disk/non-volatile memory, tamper free storage.
	2	Key material on decommissioned device could be used to decipher recorded network traffic.	Purge of cryptographic data, tamper free storage.
	2	Decommissioned device could be reconnected as a "rogue device" to intercept information.	Policy on revocation of decommissioned privileges.
	2	New device is not patched to latest version and can therefore contain vulnerabilities on provisioning.	Policy on up-to-date software on provisioning.
	1,3	Malware leaking data.	Intrusion detection systems (IDS), malware detection, encryption of DAR.
Denial of Service	1,3	Connectivity or bandwidth attack - overload of requests in any direction.	Hardening, limiting allowed requests from one endpoint, limiting amount of resources needed for handling a not-authorized connection, firewalls, etc.
	1,2,3	Malware alters the system behaviour	Malware detection methods.
	1,3	Replay attack intended to disrupt or alter functionality.	Using e.g., session token to invalidate old messages.
	1,3	Routing attack disrupting data flows.	IDS and malware detection also on routing nodes.
Elevation of Privileges	1,3	A legitimate actor gains access of a resource without proper privileges.	Authorization using least privilege principle.
	1,3	A process running on the same device as another service gains access to e.g., memory or disk outside its intended scope.	Proper sand-boxing, parameter-bound checking, etc.

TABLE I: A simplified threat model derived using STRIDE model

is for example vulnerable to both impersonation and Denial of Service (DoS) attacks [16]. Authentication is the method where an actor presents proof for a given identity, usually referred to as credentials. Therefore all actors (e.g., an I/O-data service for the pump, a software patch publisher, etc.) must be able to authenticate themselves. Furthermore, for actors that have to interact with each other, a common method for authentication is needed.

A number of techniques exist for authentication, e.g., using a shared secret (password), digital certificates (x.509) and signatures such as RSA-PSS, DSA, BLS, bio-metrical measures (e.g., fingerprints), etc. Using a trusted third party for providing authentication is a way to enable actors to establish trust without prior knowledge of each other. Kerberos [17] is one such protocol for a secure authentication over a non-secure network using a trusted third part, where several implementations exists, using different combinations of cryptographic algorithms. OpenID is an open standard and protocol commonly used for enabling websites to authenticate users on the website with e.g., Google or Facebook as identity providers. Signatures from certificates with a common trusted root certificate is another way to provide authentication.

C. Authorization

Authorization is a method of connecting an identity with a set of privileges. In the case of Scenario 2, the new pump must be authorized to perform any action the old defective pump was able to e.g., reading I/O data from flow sensor, at the same time as the defective pump must have all its privileges revoked.

Granting and validating privileges of an actor can be done in several ways such as: *1)* Identity based authority, meaning that the owner of the resource the actors wants to access, keeps a record of identities paired with privileges e.g., Access Control Lists (ACL); *2)* Attribute Based Access Control (ABAC) where attributes of an actor are used in deciding authority; *3)* Role-based Access Control (RBAC), the

owner of the resource allocates certain privileges to specific roles, and there is a way to deduce a role from an identity; *4)* Information flow focused methods based on sensitivity levels of information and clearance levels of actors.

Available technical solutions include OAuth [18], an open standard for delegating authorization for HTTP-based applications. Extensible Access Control Markup Language (XACML) is a standard and framework that can be used for describing access control policies using both ABAC and RBAC. PERMIS [19] is a framework focusing on RBAC, but using a certificates based infrastructure to define roles and privileges.

D. Integrity

Integrity of data is the characteristic proving that the data have not been maliciously or accidentally changed or destroyed [20]. Data needs to be checked for integrity, both when reading from storage to protect against tampering of Data at Rest (i.e. DAR) and when receiving data over a network (i.e., DIM). Using check-sums and Message Authentication Code (MAC) are standard methods for ensuring integrity of data.

In Scenario 3, the integrity of the data flow from device to manufacturer is crucial, as tampering of data could prevent urgent patches being applied. Integrity of the patch itself is also very important, as a tampered software update would possibly inject malware into the device. Software ID (SWID) tagging as described in ISO/IEC 19770-2 [21] is one technique to assure software update integrity.

In Scenario 2, the authenticity of the new pump-type can be questioned. It would be desirable to detect a counterfeit, as it could contain malware, under-perform, etc. To prove that the device software is authentic it might provide a digital signature from the vendor based on a certificate originating from a well known certification authority.

E. Encryption

Encryption is a method of rendering data unreadable to anyone not holding a deciphering key. Symmetric encryption methods, such as AES, use the same key for encryption and decryption. In asymmetric encryption methods, such as RSA, the key for encryption and decryption differs and this is the basis for any public-key technique where the key used for encryption is made public and the key for decryption is secret. Asymmetric encryption enables secure communication without previously shared secrets. The strength of any encryption algorithm is related to the length of the deciphering key.

If the trend service described in Scenario 1 is accessing sensitive data, it must be protected from unintended viewers both at rest (i.e., DAR) and in motion (i.e., DIM). The transfer of a software patch between a publisher and device, as described in the previous section related to Scenario 3, should also be protected from eavesdropping. An attacker could, e.g., use the software to perform a binary analysis of it as a part of preparing a future attack.

To protect DIM it must be decided in which layer the protection should be implemented (e.g., link-layer, network, transport, or application). Securing communication only at the lowest levels might work well for some applications, but may not provide granular enough security controls for all applications. For example if using a message broker to handle communication, sensitive data only intended for the receiver of the message must be encrypted before reaching the broker. A combination of network/transport and application layer protection could be applicable in such cases. Common state-of-the-art protocols for protection of DIM are: IPSec, Transport Layer Security (TLS), Datagram Transport Layer Security (DTLS), Wireless Transport Layer Security (WTLS). For situations where the intended receiver is not know, or there are several receivers, Attribute Based Encryption (ABE) [22] can be used. Using this technique a publisher is decoupled from a subscriber using a trusted key-host.

F. Audit log

An audit log is a record or set of records containing timestamped actions of predefined types, typically security related events such as failed login attempts or access to sensitive data. What should be logged depends on the security policy of the organisation. Audit logging is needed to perform forensic analysis and prove repudiation. Monitoring audit logs is also a way to detect attack attempts, as a wide range of attack vectors may be utilized before an attack is successful. The audit logs are themselves therefore possible attack-targets and must therefore be protected at storage and transferred using secure channels. For example, in Scenario 3, the patch publisher might claim that the software patch is sent to the pump device, while on the other hand the pump device might claim not to have received any patch. Audit logging is used to prove or disprove these competing claims.

G. Malware detection

Malware stands for for malicious software, meaning software performing actions not desired in the system, e.g., leaking data, altering device behaviour, using up local or remote resources, etc. Several of the deduced threats and countermeasures are related to a device or service malfunctioning. The patch being applied in Scenario 3 might contain malware, the pump installed in Scenario 2 might contain malware, any of the I/O-data being displayed in the trend in Scenario 1 might deviate from the real values due to a malfunction, etc. There are several possible root causes for a defective behavior of a device (i.e., mechanical or electrical error, network disturbances, etc.) and the method for detecting the fault differs based on the root cause.

Methods for detecting malware include trusted boot, software attestation, IDS, application white-listing and anti-virus software. Trusted boot is used to assert that any software loaded during the boot is the expected one (e.g., an operating system). Attestation is a method were the executing software of a device is validated often using a challenge-response method. Both self-attestation and remote attestation is possible. Trusted Platform Module (TPM) [23] is a hardware module that among other things can support trusted boot and self-attestation. SMART [24] is an example of an architecture for providing attestation for resource constrained devices. IDS [25] is a mechanism for monitoring activities in a system, compare with the past behavior and known attack patterns, and report upon finding anomalies.

V. Challenges and future directions

In Section IV previously introduced scenarios (see Section III) are generalized to a number of basic security requirements with respect to the infrastructure of a large-scale IIoT system. In this section we discuss some of the challenges for such an infrastructure in an IIoT context.

A. Interoperability

For the previously discussed security related requirements, there are several applicable state of the art solutions. Different protocols for communication, authorization and authentication exist, as well as many algorithms for encryption and information integrity. Considering a large IIoT, there will be devices and services implemented using competing technologies that must be able to communicate within the same system. Frustaci et al. [26] provide a classification of commonly used IoT protocols at physical, network and application layer, including each protocols' security issues and related solutions. To combine several exiting protocols and standards currently in use in industrial applications (e.g., OPC UA, PROFINET, MODBUS, etc.) it becomes a challenging task to enable basic interoperability with regards to communication. It does not seem feasible to limit the communication capabilities in an IIoT system to a few interoperable protocol implementations, as it will put unreasonable constraints on the devices. Instead a unifying methodology that allow cross-technology communication is required.

Let us consider Scenario 1 described in Section III. I/O-data from the sensor could e.g. be accessible from an OPC UA server, I/O-data from the pump-device could be accessible from a MQTT message broker running on an edge device. For every such type of data-source that must be handled by the trend-service, a significant amount of implementation work is required. Furthermore, it will be virtually impossible to know at design-time which types of data-sources the trend service must be able to support.

Looking at the available architectures it is not clear how to achieve interoperability between competing technologies, or technologies never intended to be interoperable when designed. As a solution, RAMI4.0 requires that all entities must have an administrative shell and component manager that exposes services and data in a very uniform way to be part of an Industry 4.0 system.

Using the layered databus architecture pattern is one way of handling communication interoperability. Such architecture only requires for each logical layer the existence of a common data model allowing the entities within that layer to communicate. Between each layer there is a databus gateway, enabling flow between layers. For interoperability between layers there is a need for adapters in gateways to translate between the data-models [7]. The idea of the layered databus is however not to allow free communication between arbitrary endpoints. To allow secure communication in this context seems to be quite difficult, as data will be transformed at every gateway.

Yet another way of looking at solutions for interoperability between actors without prior knowledge of each others technological stack is to use an App-centric view. Assume that a service S_1 running on a device D_1 wants to access a service S_2, but they are implemented using competing technologies. Now, if D_1 can execute concurrent services (e.g., using docker containers) and there is a well-defined secure method for local communication between services on D_1 (e.g., an internal message broker), then it would be enough that a service is created such that it can execute on D_1 which reads the data from S_2 and then post the data on the internal message broker. It should be noted that any of the suggested solutions will require some predefined functionality for fetching meta-data about a device or a service.

B. Management of privileges, identity mappings and data classifications

In any system there is a need to administer different characteristics for included actors.

- Manage Identities (e.g., add and remove users, map device IDs to location and functions, etc.).
- Manage Privileges (Which user/service/device is allowed to read specific information or execute an action.).
- Classification of data (Whether the data should be encrypted at rest, in motion, is it sensitive or not, does it contain information that requires it to be stored in accordance with e.g., GDPR, etc.).
- Maintenance Scheduling (e.g., when to replace the pump in Scenario 2, when to apply the patch in Scenario 3, an so on).

Even for a quite small amount of actors this can be a tedious task. For a large scale IIoT system, the number of actors is huge and their relationship may not be predefined. Such administration might be complex and time-consuming. Therefore, for a technology only requiring high-level configuration and promising autonomy at the lower levels, the management at the lower level must somehow be automated. For example, in scenario 2, there could be logic based on proximity detecting the new pump device and assigning it roles and privileges accordingly.

One has to keep in mind that the best-practice for authorization is the principle of least privilege. This principle states that an actor will only be granted privileges needed to perform its intended function. In an IIoT system it will be difficult to beforehand deduce what the least privileges are, potentially forcing higher privileges being granted than actually required. This could lead to a conflict with the least-privilege principle. Solving the issue of automatic management of identities, privileges, etc., remains an open question.

For some situations there might not even exist an omnipotent actor able to decide on privileges. In the case of a smart city with autonomous vehicles and smart traffic control it is reasonable that, to ensure traffic safety, a vehicle from another city or country should be allowed to communicate with other vehicles and infrastructure without prior knowledge or registration in this specific system. Smart contracts utilizing block-chains could be a way forward for preserving reliability in such scenarios [27], as well as zero-knowledge proof [28].

C. Fault and anomaly Detection

There is always an amount of uncertainty when evaluating the state of the real world. Any sensing device has a tolerance-level indicating how exact the sensor is, and for any actuating device the effect of the actuation will be based on a model of reality, which never is perfect. In scenario 1, an undetected anomaly being presented to the operator could lead to erroneous decisions being made. Malware introduced in the example system, e.g., by a malicious software update introduced in scenario 3, could lead to loss of control, as well as information leakage.

Common ways to decrease the level of uncertainty is to use e.g., secondary data-sources, or to compare model data with sensor data. These techniques could possibly be an extended form of detecting malfunctioning devices, as well as a methods for intrusion detection, which currently are a growing area of research [29].

Attestation as a method to detect malware at high-end devices is a very promising technique, but, as described by Sadeghi el al. [30], current solutions do not scale well, especially not for low-end devices. To find applicable solutions for IIoT including attestation of large amounts of devices in parallel, so called *swarm-attestation*, is still an open field for research.

The standard IDS is focused on probing network traffic to monitor and detect anomalies or predefined attack patterns and report findings to a security function (i.e., a human or a machine) so that the anomaly can be classified. These systems are well suited for detecting suspicious patterns in communication, but will have increasing difficulties in finding anomalies in data content, as the data itself often will be encrypted in an IIoT system.

Another issue of increasing importance for the IDS is knowing which traffic to monitor. In a heterogeneous IIoT system there will be devices communicating using any number of diverse wired and wireless technologies. The fifth generation telecommunication standard (i.e., 5G) is believed to be an enabling technology for wireless cross-component communication in IIoT systems [31]. In a scenario where communication is done partially using wireless communication capabilities such as 5G, the traditional IDS with trusted nodes inspecting passing traffic will not work, as much of the traffic will not pass the trusted node. For such scenarios, IDS in an ad-hoc mobile network could possibly be used [25]. Such an IDS is based on collaborating agents being deployed on many nodes, using joint status and voting to decide on anomaly detection.

Monitoring audit logs is a way for early detection of attempted intrusions, as the logs will contain information on failed access attempts. It would be possible to use these as means for intrusion detection in an IIoT system. To be useful, the detection system must be able to remotely access and monitor audit logs for a wide range of devices and services and automatically detect unexpected patterns. Security Information and Event Management (SIEM) [32] is a technology that focus on storage and analysis of audit logs. How well existing SIEM solutions perform in and scale to a heterogeneous IIoT system must be further investigated.

D. Emerging threats and technologies

The secure operation of a device is limited to the capabilities available in the device, as implemented by the manufacturer. A device may be secure at provisioning, but its continuous state with regards to security is dependent on its possibility to adapt to emerging threats and technologies. Considering that the average lifetime for machine equipment is expressed in decades [33], it will be impossible to equip devices with hardware capabilities that will match requirements for state of the art in security lasting the whole expected lifetime. It is however essential that the software for the device is kept up to date with current threats and adapts to emerging technologies as long as possible. Secure patch management and methods for assessing the status of a device software with regards to security functions is therefore of great importance to handle the risks introduced in scenario 3 as well as keeping the device software up to date. When replacing hardware as described in scenario 2, it is important to make sure that the new device is able to conform not only the functional requirements of the system, but also with regards to current cybersecurity state of the art technology.

When adding IIoT features in a brownfield system, e.g., exposing information to the Cloud, this is usually done by putting a gateway device between the information producer and consumer. The gateway will provide the security functionality required for devices that it is servicing [1]. A similar approach can possibly be used for keeping out-dated IIoT devices secure. Such a solution would require that all communication from the IIoT-device can be relayed to the gateway/proxy. For devices with wireless networking capabilities, for example built-in mobile communication chips, this solution may not be straightforward, depending on the device capabilities. In general, handling emerging threats and technologies for resource-constrained devices is very much an open issue.

VI. Related Work

Frustaci et al. [26], provide a thorough analysis of current state of the art for securing IoT devices and data, as well as an evaluation of identified critical security issues related to IoT. The focus is on resource-constrained devices for consumer use, assuming that those devices will rely on “built-in security”. In some aspects this is clearly the case also for industrial applications, as devices both for industrial and consumer applications will be constrained with regards to computational and storage capacity. However, this does not hold for software. As we have indicated in this paper, there is a clear requirement on device software to be patched to counteract emerging threats and discovered vulnerabilities. There is also an emphasis on the physical layer bringing the highest risk to the IoT-system. In contrast, most of the physical risk to the devices are not considered in this paper. For industrial applications there is usually already a layer of physical security with fences, locked doors, access control, etc. This may not hold for all industrial applications, e.g., geographically distributed processes such as a gas or oil pipeline.

Chiang et al. [34] discuss several fundamental challenges, using traditional cloud technology within the emerging IoT, and provide arguments for using fog nodes to counteract some of these challenges, e.g., related to latency requirements,

bandwidth constraints, intermittent connectivity, etc. The focus is IoT in broad terms, including both consumer and industrial applications. A number of security related challenges are discussed, some of which are described more in depth in our work, e.g., keeping security credentials and software up to date and protecting resource-constrained devices. In this paper we have consistently suggested that services will be spread out through the thing-to-cloud continuum, thereby including fog nodes. This will also be true for security related services, such as IDS, remote attestation, etc. Chiang et al. also acknowledge that fog technology introduces new security challenges, as such nodes are as diverse and distributed as IoT devices, as opposed to cloud which in general operates in a protected environment.

Sadeghi et al. [30] provide an overview of security challenges for Cyber-Physical Production Systems (CPPS). Integrity of device software is discussed as one of the challenges, with attestation of integrity needed to be performed by a trusted entity. Secure IoT management is also discussed as one of the important areas for future research. In this context the notion of "pairing" of devices are used, as done with PIN-codes on Bluetooth devices (i.e., pairing headset with cell-phone), which could be an interesting way to handle inter-device identification and authorization with minimal human interaction.

VII. Conclusions

In this paper we focus on the emerging IIoT systems being a combination of Industrial Automated Control Systems and Internet technology. In such systems, smart CPS devices and services are applied throughout the device-to-cloud continuum with heterogeneous technologies and multiple stakeholder that put high requirements on the underlying infrastructure. We have discussed a number of challenges in such a setup from a cybersecurity perspective using an example of a flow-control loop process. For such an example we describe three scenarios and apply a STRIDE threat model to deduce and discuss cybersecurity challenges within an IIoT perspective.

As future work, we aim to analyze emerging IIoT systems in more detail, focusing on cybersecurity aspects identified in this paper in order to provide required solutions. The goal is to analyze solutions applicable to a truly modular infrastructure for cybersecurity that scale well with regards to large-scale IIoT systems.

Acknowledgements

This work is supported by ABB, the industrial postgraduate school Automation Region Research Academy (ARRAY), and SAFSEC-CPS, projects funded by The Knowledge Foundation. Additional support is provided by Serendipity, a project funded by The Swedish Foundation for Strategic Research. The authors would like to acknowledge Mikael Rudin and Tomas Lindström for valuable discussions and feedback when writing this paper, as well as anonymous reviewers for their comments.

References

[1] Industrial Internet Consortium, "Industrial Internet of Things Volume G4 : Security Framework," 2016.

[2] International Electrotechnical Commission, "Smart Manufacturing - Reference Architecture Module Industry 4.0 (RAMI4.0)," 2016.

[3] W. Madsen, *Trust in Cyberspace*. National Academies Press, 1999.

[4] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *Proceedings of the Hawaii International Conference on System Sciences*, vol. 2016-March, pp. 3928–3937, IEEE, 2016.

[5] Microsoft, "The STRIDE Threat Model." https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20), 2005. [Online; accessed 5-march-2019].

[6] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Computers in Industry*, vol. 101, pp. 1–12, June 2018.

[7] IIC, "The Industrial Internet of Things Volume G1: Reference Architecture," Tech. Rep. November, Industrial Internet Consortium, 2017.

[8] S. Mumtaz *et al.*, "Massive Internet of Things for Industrial Applications: Addressing Wireless IIoT Connectivity Challenges and Ecosystem Fragmentation," *IEEE Ind. Elec. Magazine*, vol. 11, no. 1, 2017.

[9] R. Kissel, *Glossary of key information security terms, Rev. 2*. U.S. Dept. of Commerce, National Institute of Standards and Technology, 2013.

[10] Microsoft, "Microsoft Azure IoT Reference Architecture." https://aka.ms/iotrefarchitecture, 2018. [Online; accessed 29-may-2019].

[11] R. Shahan and B. Lamos, "Internet of Things (IoT) security architecture." https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-security-architecture, 2018. [Online; accessed 29-may-2019].

[12] FIRST, "Common Vulnerability Scoring System." https://www.first.org/cvss/, 2019. [Online; accessed 29-may-2019].

[13] T. UcedaVelez and M. M. Morana, *Risk Centric Threat Modeling: process for attack simulation and threat analysis*. Wiley & Sons publishing, 2015.

[14] R. Baheti and H. Gill, "Cyber-pysical Systems," *The impact of control technology*, vol. 12, no. 1, pp. 161–166, 2011.

[15] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, pp. 25–33, 01 2006.

[16] G. Avoine, *Encyclopedia of Cryptography and Security: RFID Security*, pp. 1044–1045. Boston, MA: Springer US, 2011.

[17] J. G. Steiner, C. Neuman, and J. I. Schiller, "Kerberos: An Authentication Service for Open Network Systems," *WTEC 1988: Proceedings of the USENIX Winter 1988 Technical Conference*, pp. 191–202, 1988.

[18] D. Hardt, "The OAuth 2.0 Authorization Framework." Internet Requests for Comments, October 2012.

[19] D. W. Chadwick and A. Otenko, "The PERMIS X.509 role based privilege management infrastructure," *Future Generation Computer Systems*, vol. 19, no. 2, pp. 277–289, 2003.

[20] M. H. Weik, *Computer Science and Communications Dictionary: Data integrity*, pp. 350–350. Boston, MA: Springer US, 2001.

[21] ISO IEC, "ISO/IEC 19770-2:2015 IT Asset Management Part 2: Software Identification tag," tech. rep., ISO/IEC, 2015.

[22] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data," in *13th ACM Conference on Computer and Communications Security*, ACM, 2006.

[23] D. Challener, *Encyclopedia of Cryptography and Security: TPM*, pp. 1308–1310. Boston, MA: Springer US, 2011.

[24] K. E. Defrawy, A. Francillon, D. Perito, and G. Tsudik, "SMART: Secure and Minimal Architecture for (Establishing a Dynamic) Root of Trust," *NDSS*, 2012.

[25] Q. Gu, *Encyclopedia of Cryptography and Security: Intrusion Detection in Ad Hoc Networks*, pp. 620–623. Boston, MA: Springer US, 2011.

[26] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the iot world: Present and future challenges," *IEEE Internet of Things Journal*, vol. 5, pp. 2483–2495, 2018.

[27] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[28] B. Schoenmakers, *Encyclopedia of Cryptography and Security: Zero-Knowledge*, pp. 1401–1403. Boston, MA: Springer US, 2011.

[29] J. Giraldo *et al.*, "A Survey of Physics-Based Attack Detection in Cyber-Physical Systems," *ACM Computing Surveys*, vol. 51, no. 4, 2018.

[30] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," *Proceedings of the 52nd Annual Design Automation Conference on - DAC '15*, pp. 1–6, 2015.

[31] S. Li, L. D. Xu, and S. Zhao, "5G Internet of Things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.

[32] S. Bhatt, P. K. Manadhata, and L. Zomlot, "The Operational Role of Security Information and Event Management Systems," *IEEE Security & Privacy*, no. October, 2014.

[33] A. A. Erumban, "Lifetimes of machinery and equipment: Evidence from dutch manufacturing," *Review of Income and Wealth*, vol. 54, jun 2008.

[34] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.