# Towards Secure Digital Twins

Tomas Kulik[1](✉), Cláudio Gomes[2], Hugo Daniel Macedo[2], Stefan Hallerstede[2], and Peter Gorm Larsen[2]

[1] Sweet Geeks, Innovations Allé 3, 7100 Vejle, Denmark
tomaskulik@icloud.com

[2] DIGIT, Department of Electrical and Computer Engineering, Aarhus University, Finlandsgade 22, 8200 Aarhus, Denmark

**Abstract.** Advanced digital technology is finding its way into industrial production and control systems. This led to development of further concepts such as digital shadow and digital twin. In the former an accurate model of the cyber-physical system (CPS) is used to monitor it virtually, while the latter provides a possibility to adapt the CPS's behavior. These developments are often welcome from the operators perspective, however they also pose new challenges in terms of cyber security: an operator could be led to believe the system is operating correctly due to the represented digital image while the CPS is under a cyber attack. In this paper we investigate several cyber security challenges of the digital twin technology and discuss potential mitigations for these challenges based on well established practices within the area of industrial control systems. We further describe the potential cyber attacks and mitigations using a semi-formal notation based on problem frames, we suggest in order to simplify the communication about cyber security challenges of digital twins between different stakeholders. This is shown within a context of a small case study. Finally we outline areas of research for the development of secure digital twin technology.

**Keywords:** Digital twins · Cyber security · Security model

## 1 Introduction

Digital twin technology is finding its way into different aspects of the modern society, especially within the industrial domain following the concepts of Industry 4.0 [27]. The access to digital representation of physical objects brings many benefits. For example, in manufacturing different configuration changes could be applied to the digital twin before being passed on to the physical system [11]. It is furthermore possible to provide a comprehensive overview of a system to a system operator, since the digital twin responds to the data provided from the physical device a similar way as the physical device itself. This could provide simplified troubleshooting employing a simple visual representation of the state of the physical devices.

This approach also poses several challenges. One of them is that the digital model needs to represent the physical object at a high degree of accuracy;

T. Margaria and B. Steffen (Eds.): ISoLA 2022, LNCS 13704, pp. 159–176, 2022.
https://doi.org/10.1007/978-3-031-19762-8_11

another challenge is that in some cases the data exchanged between the physical object and the digital twin must be close to real-time. Last but not least, is the need for cyber security assurances within the systems employing the digital twin technology, which is the focus of this paper. This aspect of digital twins is especially important for industrial use of digital twins because compromised security could lead to potentially harmful situations including unstable operation of the CPS with resulting physical and economic damage or even accidents leading to injury or death. To this end we perceive the digital twin as a prime target for potential attackers, similar to SCADA systems [20,22].

The practical difference between a digital twin and a SCADA system is the integration of the close to reality digital model of the CPS within the digital twin. In comparison the SCADA system mostly provides industrial connectivity modules to exchange data with the controlled plant via a user interface that enables the operator to interact with the system. It is important to note that in some cases digital twins have been proposed to be a potential solution to security challenges of large connected CPS [1]. However, even in this case the digital twin must be secure by itself in order not to provide a false sense of security for the operators and designers of the CPS.

Recently, an increase in attack surfaces within the industrial control systems has been observed [15]. This is mainly due to the additional connectivity being added to these systems. Utilizing a digital twin further increases the attack surface because the model underlying the digital twin might become an attack vector. Several ways of preventing attacks at industrial and specifically cyber physical systems have been proposed, such as among others, use of formal methods to create secure architectures [19], integration of different security controls [18] or the use of state estimators [16], where attack resilient state estimators have been proposed [21]. Securing digital twins consequently requires not only considering access control, network security and transmitted data integrity but also integrity of the model itself. One might also consider the question of how the digital twin is being used. For example, it could be shared by several entities and as such could be deployed within the cloud environment, creating security constraints for this environment, such as isolation of different users.

**Contribution.** While several cyber security challenges of digital twins have been pondered before [12] and a notion of digital twin trustworthiness has been presented [24], in this paper we explore potential security challenges of digital twins based on concrete types of attacks. We also describe potential mitigations for these attacks in the context of digital twins. Within the presented attacks we introduce multiple attack vectors against a digital twin. We further provide a supporting notation that could be used to describe security concerns of digital twins. We consider this is currently lacking and could provide benefits to the wider digital twin research and development community. We also present a case study based on a digital twin of our own design for an incubator system by means of which we discuss the listed security challenges. Finally, we present several open problems in regard to secure digital twins that pose interesting topics to be addressed by future research.

**Structure.** The rest of the paper is organized as follows: Section 2 presents work concerning security of digital twins with a focus on their industrial use and how they compare to our work. Section 3 introduces several cyber attacks against digital twins (possible attack vectors) as well as introduce a notation for describing these cyber attacks in a digital twin setting. Section 4 provides an overview of different mitigations that could be applied towards the attacks introduced before and discusses how these mitigations differ from their use within standard industrial control systems when applied to a digital twin setting. Section 5 describes the incubator system, considering the CPS and the digital twin including a potential cyber security challenge and mitigation within this setting. Section 6 presents several open problems that might be addressed by future research. Finally, Sect. 7 closes with concluding remarks.

## 2 Related Work

The majority of research works cover the security aspects of digital twin technology as yet another aspect to take into account while developing such systems, yet there are other works that focus on the security aspect itself or the usage of the digital twin as another security tool.

In [25], the authors models attacks on digital twins, and present a study on the abuse cases of digital twins. Compared to our technical descriptions of attacks and mitigations, the authors focus on the different attackers' strategies and the outcomes of attacks at specific phases of the lifecycle.

As examples of a digital twin as a security tool we have [4,7] both proposing using the digital twins assets in the design of the security aspects and attach modelling and mitigation. In [7], the authors propose a framework to generate digital twins from specifications for SCADA systems. The specifications may include specific security properties that shall hold within the system. As a proof of concept the authors propose a mitigation for man in the middle attacks such as the paradigmatic Stuxnet attack. In addition to being a security tool, the work of [4] proposes to use the digital twin in training and simulation, testing exercises for the security engineers.

Another aspect that has not been massively covered in publications with a technical aspect, but of importance and covered in philosophically oriented works like the one in [6] is the privacy impacts of a technology creating massive amounts of data and digital models of physical assets in the real world, which are used and tied to its users, who have the right to be protected from potential surveillance and discrimination.

## 3 Security Challenges

Digital twins face different kinds of security challenges. In this paper we introduce four security challenges with various levels of complexity and impact. In the first three cases we consider that the digital twin needs to be connected with the CPS via a network. But we also show that even an isolated digital twin can face potential security challenges. The four attack types we consider are *bandwidth sniffing*, *data injection*, *data delay* and *model corruption*. We describe the attacks in natural language followed by graphical a notation describing these attacks in a more succinct manner. We further introduce a notion of *direct* and *indirect* attack. In the case of a direct attack, the attacker interacts actively with components of the system, while in the case of an indirect attack the attacker utilizes methods such as side channel attacks where the attacker does not need to interact actively with the deployed system.

**Attack Description Syntax.** Textual descriptions of the different attacks are difficult to comprehend and explain, in particular, to the many non-expert –in security– stakeholders in a digital twin setting. We use context diagrams from the problem frames approach [14] to complement the textual descriptions semi-formally introducing the main concepts and how they are related. This is sufficient to understand how the different attacks and mitigations work. Context diagrams are composed of domains that describe the key aspects and participants. They do not describe an architecture. In fact, they only contain one machine domain representing "the software" or "the computer". In our diagrams it is always called *Digital Twin*. Some domains can be controlled like the *CPS* (whose behavior is predictable and therefore called *causal* and marked with a C). It is the objective of the Digital Twin to re-configure or augment the CPS, of course. The behavior of the *Attacker*, whose domain is called *biddable* and marked with a B, is not predictable in causal terms. Another relevant kind of domain is called *lexical*: they represent some form of data. Domains that are *designed* (by us) are marked with one vertical bar on the left and machine domains with two vertical bars. Domains that interact in some way are connected by edges that are annotated with the phenomena they share. Phenomena are abstractions of concepts from the real world that can be observed or measured (for a more thorough discussion see [13,14]). In order to emphasize the role of the network, we draw all edges connecting to it by "double bars". Attacks are described in oval dashed ellipses that are linked to the interactions that they manipulate.

### 3.1 Bandwidth Sniffing

This attack utilizes information gained about communication between the connected digital twin and the CPS as basis for further targeted attacks. The attacker does not acquire any confidential data that is being transmitted within the communication network connecting the digital twin and the CPS. In addition, the attacker does not try to inject malicious payload to the communication. However the attacker can learn some specific information by simply listening to the different communication channels without decoding the underlying traffic. The attacker can potentially discover which component of the CPS is currently active or even determine the activity within the CPS [28] based on the bandwidth used between the CPS and the digital twin. See Fig. 1 for a graphical description of the latter kind of attack. For the former kind of attack the domain "Activity" would need to be replaced by a domain "Component".

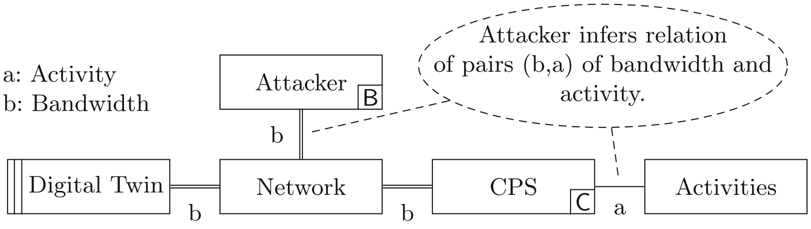

**Fig. 1.** Context diagram for bandwidth sniffing.

This information could in turn be used to execute cyber or physical attacks against the CPS in question. In many situations such information will be considered confidential and the system operator needs to be able to prevent the attacker from obtaining the information. While this attack is applicable to any connected system, the connected digital twin setup is uniquely well positioned for such attacks because a significant amount of data needs to be transmitted for processing by the digital twin model in order to keep the digital twin and the typically complex CPS in synchrony. This attack is an indirect side channel attack, where indirect information is used to gain knowledge.

### 3.2 Data Injection

Similarly to bandwidth sniffing, this kind of attack is mainly aimed at the system with connected digital twin. The attacker utilizes a network breach or a compromised entity within the network to inject malicious payload to the network.

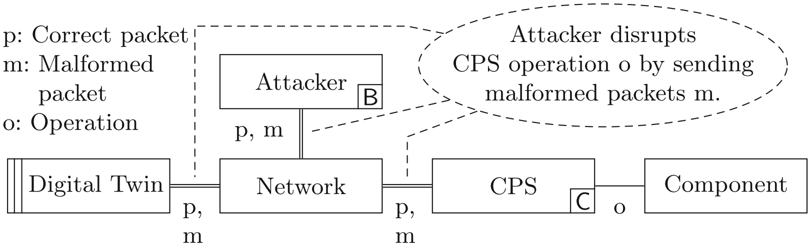

**Fig. 2.** Context diagram for data injection.

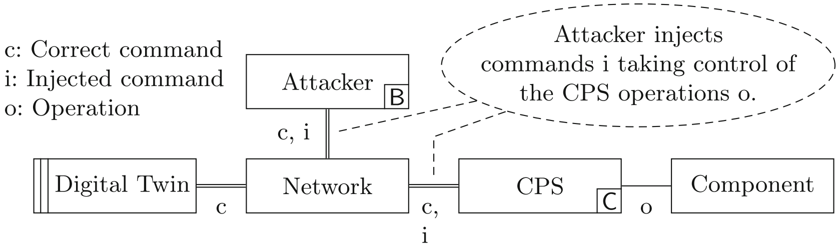

**Fig. 3.** Context diagram for data injection (variant 1).

For example, the attacker could simply inject malformed packets that could result in logic errors within the CPS (see Fig. 2). Alternatively, the attacker could also inject commands that seemingly originated from the digital twin in order to take over the control of the CPS as illustrated by Fig. 3. Finally, the attacker could inject falsified data that seemingly originated from the CPS causing the digital twin to provide a significantly deviating picture from the real state of the CPS (see Fig. 4).

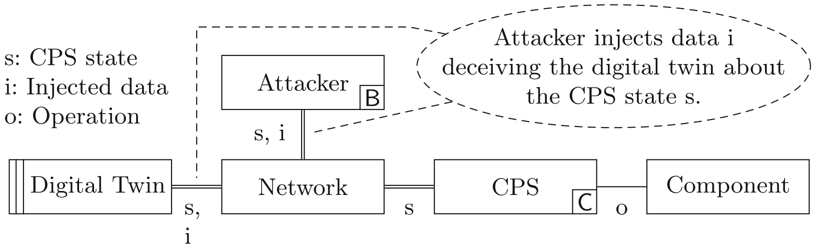

**Fig. 4.** Context diagram for data injection (variant 2).

Similar attacks have been also executed against industrial control systems with limited network connectivity [26], impacting the control network and subsequently causing catastrophic failure of the CPS.

We consider this one of the prime attacks that could be applied towards a digital twin given its data dependent nature. This class of attacks requires the attacker to be able to compromise the network and determine the correct formatting of either the data payload towards the digital twin or towards the CPS. Once the formatting is known, the attacker issued payload is simply mixed with the legitimate payload generated within the system. This attack can be considered as a direct attack against the digital twin enabled system, given that the attacker interacts with the network in a direct way.

### 3.3 Data Delay

This kind of attack is aimed at a system with a connected digital twin with real-time characteristics, a typical case as discussed in the introduction. The attacker attempts to slow down the communication from the CPS towards the digital twin [5]. This could be seen as a limited denial of service attack, where the attacker floods the network trying to prevent the system to reply to legitimate requests. In this case the attack does not attempt to prevent the communication between the digital twin and the CPS completely but it floods the network with enough packages to ensure that either the reaction of the digital twin will be significantly late and hence potentially cause a system malfunction (in case that the digital twin reaction is used a feedback for the CPS); or slowly force the digital twin to lose synchrony with the CPS (see Fig. 5).

In case that the digital twin needs to exchange data close to real-time, this type of attack could cause the digital twin to miss tight, crucial deadlines. In order to carry out such an attack the attacker needs to be present within the network and have an understanding of the specific network. Since the goal is to cause time delays in processing between the digital twin and the CPS, it is important for the attacker to send only as many packets as the network can handle, enabling the (cautious) attacker to avoid immediate detection. This attack can also be considered a direct attack where the attacker needs to have sufficient knowledge of the data being processed by the digital twin.

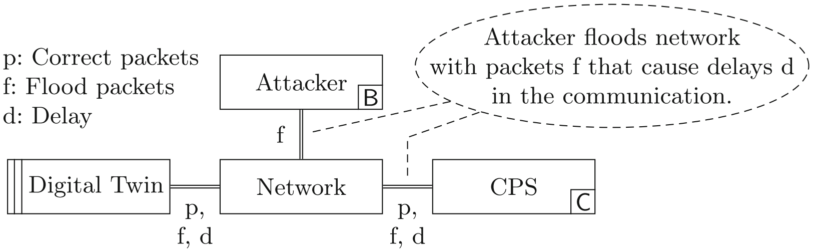

**Fig. 5.** Context diagram for data delay.

### 3.4 Model Corruption

This kind of attack aims at corrupting the model that the digital twin uses to represent the physical system. The models are often developed by multiple parties involving multiple developers or even multiple organizations; they are often stored in shared repositories that are used for version control. In this case the attacker would aim to attack the model at rest within the shared repository. The attack is based on injecting malicious code directly to the model causing the digital twin to either not represent the physical device truthfully, or provide malicious data payload to the physical device (see Fig. 6 for a description of the latter).

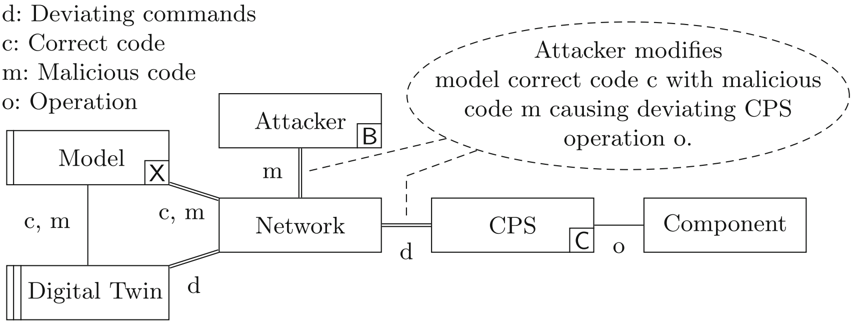

**Fig. 6.** Context diagram for model corruption (network).

Similar attacks have been proposed as a possible attack vectors to PLCs [30], impacting the control loop of a physical system and potentially leading to unsafe situations. In order to carry out this attack, potential attackers would need to gain access to the shared repository where they could inject the code. This attack is also well suited for insider attackers where a legitimate entity turns malicious. Another option is injection of malicious code to third party libraries used by the model (see Fig. 7). Similarly to the above, in this case the attacker would need to gain access to the storage of the library code and ensure that the malicious code will be included in the library when it gets deployed. However, it cannot be directly detected on the observed network unless all used libraries are included in this. This attack can be considered an indirect attack.

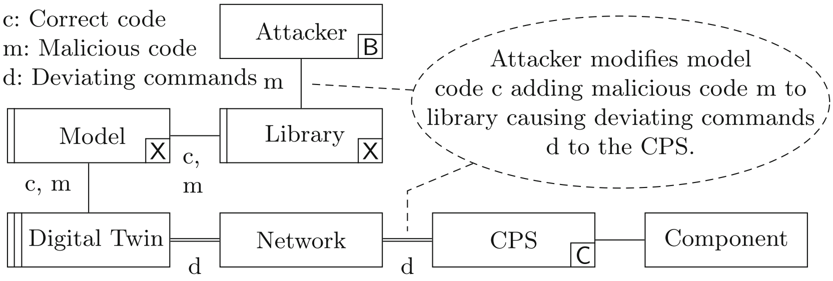

**Fig. 7.** Context diagram for model corruption (library).

# 4 Mitigations

In this section we propose specific mitigations that could be applied against the cyber attacks presented in the preceding section. The mitigations are based on approaches used in security of industrial control systems, and have been applied against similar attacks. We present similarities and differences between security needs of industrial control systems operating with and without digital twins. The approaches we present are: *Fragmentation and data padding*; *Signatures and tokens*; *Threshold monitoring and network-aware digital twin models*; and *Model integrity checks.* As before, we first describe the mitigations using natural language followed by a graphical description to explain the mitigations in a compact format.

## 4.1 Fragmentation and Data Padding

These approaches can be used as a mitigation for the *Bandwidth sniffing* (Sect. 3.1) attack. The mitigation either utilizes data fragmentation, i.e., splitting the data into smaller chunks and sending these over the network, changing the nature of the bandwidth utilization. Similarly the data padding approach changes the nature of the bandwidth utilization by adding more data to the original payload in order to keep the bandwidth utilization steady [29]. Once these techniques are employed it becomes difficult for the attacker to gather information about different targets simply by observing the bandwidth utilization (see Fig. 8).

It is important to note that this mitigation might directly conflict with some optimization strategies for the system, especially if network traffic shall be optimized to minimize the bandwidth utilization. It is however an effective mitigation that has been proposed for use within industrial control systems. One aspect of this mitigation that needs to be considered when incorporating a digital twin within a system is the need for the model and communication interfaces to be able to handle either the fragmented or padded data. This needs to be considered bidirectionally. As such, the digital twin needs to remove padding or fragmentation from incoming data payloads, as well as add these to the outgoing data payloads to ensure that the bandwidth stays protected from bandwidth sniffing attacks.

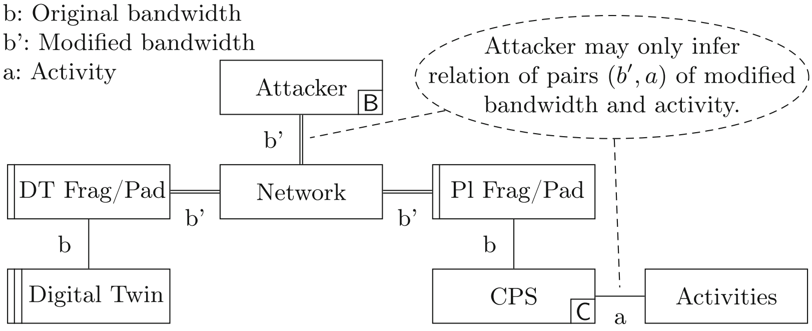

**Fig. 8.** Context diagram for bandwidth sniffing mitigation.

## 4.2 Signatures and Tokens

A mitigation scheme that can be applied towards the *Data injection* attack (Sect. 3.2) is the addition of digital signatures to the data transferred between legitimate entities [23]. This scheme has been utilized within different kinds of systems in order to ensure data integrity. One of the benefits of this mitigation is that it does not require use of more complicated schemes such as state estimators. It may be added to the majority of communication protocols because the signature or cryptographic token becomes a part of the regular data payload. See Fig. 9 for a graphical description of this mitigation. Two lexical domains DT Sign and PL Sign have been added to sign payload from the digital twin and the CPS, respectively. The protection mechanism relies on the data sources ability to cryptographically sign the generated data. The signature is subsequently validated at the data sink and any invalid signature is rejected and may be considered a potential intrusion.

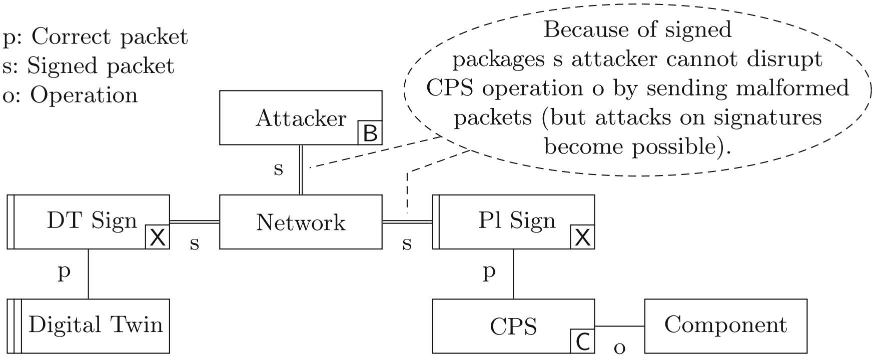
p: Correct packet
s: Signed packet
o: Operation
Because of signed packages s attacker cannot disrupt CPS operation o by sending malformed packets (but attacks on signatures become possible).
Attacker
B
s
DT Sign
X
Network
Pl Sign
X
s
s
p
p
Digital Twin
CPS
C
o
Component

**Fig. 9.** Context diagram for data injection mitigation.

One of the challenges of this scheme is the need for the data source to protect secrets that are used within the signature generation as well as being computationally sufficiently powerful to sign the data payload before sending without causing unacceptable delays. In case that the attacker gains access to a secret used for signature generation it becomes possible to inject data into the system with valid signatures that the sink cannot distinguish from legitimate data. While this simple scheme has potential security issues because the two lexical domains can be attacked in turn, this remains an established and effective way achieve security of critical digital twin enabled industrial control systems.

### 4.3 Threshold Monitoring and Network-Aware Digital Twin Models

This layered mitigation is aimed at detecting and limiting the impact of the *Data delay* attack (Sect. 3.3). The mitigation is based on monitoring of network activity and determining whether different threshold parameters have been reached, e.g., a certain amount of data packets or network latency. Data delay attacks are especially difficult to detect in low rate attack scenarios. To this end several threshold based analysis mechanisms have been considered for different types of systems [3]. Within a digital twin enabled system the threshold analysis could be integrated directly into the model. As such, the digital twin is aware of the network performance under normal circumstances as well as what it considers a data delay attack (see Fig. 10). This requires the digital twin designers to have domain knowledge about the system at hand (not only from the functionality perspective but also concerning the network setup) and expected data load. If a data delay attack is detected, the digital twin could utilize its understanding of the network to limit the effect of the attack. Possible counter measures include locking the source addresses from the network communication or limiting its own communication with the system to provide more bandwidth for legitimate data packets. More advanced schemes could be utilized if the system employs additional DDoS protection mechanisms such as resource scaling. However, this often requires external components, i.e., cloud resource orchestration and is not practical in resource constrained environments.

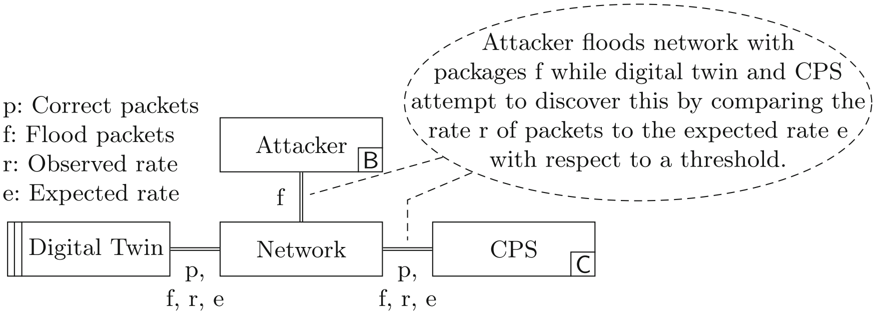
p: Correct packets
f: Flood packets
r: Observed rate
e: Expected rate
Attacker
B
f
Digital Twin
Network
CPS
C
p,
f, r, e
p,
f, r, e
Attacker floods network with packages f while digital twin and CPS attempt to discover this by comparing the rate r of packets to the expected rate e with respect to a threshold.

**Fig. 10.** Context diagram for data delay mitigation.

### 4.4 Model Integrity Checks

This mitigation ensures that the model utilized within the digital twin does not integrate malicious code and applies to the *Model corruption* attack (Sect. 3.4). This requires stringent access controls towards the repository that stores the model. Furthermore, the model itself must be validated before it is loaded onto the digital twin. To do this, only a model that is digitally signed by the authors must be allowed to be loaded. See Fig. 11 for a description of this mitigation. The additional lexical domain “Auth” is required to keep the authentication data for the signing keys. Furthermore, all of the potential libraries should be checked against provided hashes to ensure that the library has not been modified. In case that the hashes are not provided by the authors of the libraries, these must be created upon induction of the libraries to the code base of the model, where the induction process shall involve a thorough review of these libraries. This approach is a well known scheme [2] that is nowadays proposed to be used with additional schemes such as watermarking (embedding specific cryptographic elements in the code), dynamic whitelisting (dynamically determining which libraries are allowed to be loaded based on their signatures) or even formal analysis (analyzing the model against specific properties on the implementation level). While digital signing by itself would not protect the model from insider attacks and requires secure access to the signing keys, it provides a first level of guarantee that a genuine model is present within the digital twin. It should be mentioned that more advanced methods for integrity checks could be utilized. This approach has a very low impact on performance and other design constraints of digital twins, hence model signing is widely applicable even for models of digital twins that are not necessarily considered security critical.

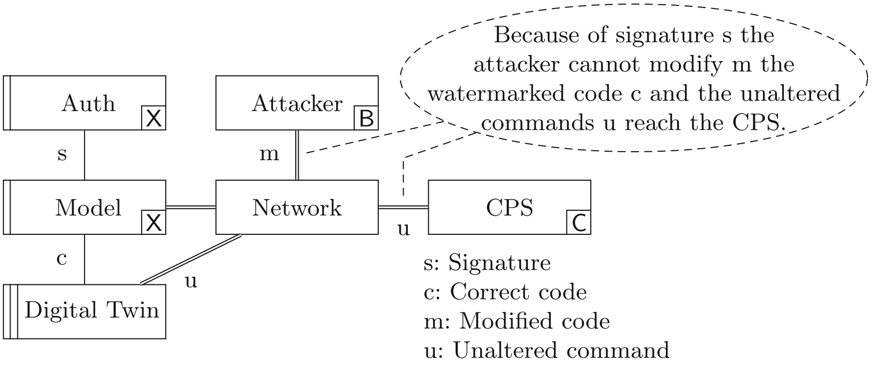
Because of signature s the attacker cannot modify m the watermarked code c and the unaltered commands u reach the CPS.
Auth
X
Attacker
B
s
m
Model
X
Network
CPS
C
u
c
u
Digital Twin
s: Signature
c: Correct code
m: Modified code
u: Unaltered command

**Fig. 11.** Context diagram for model corruption (network) mitigation.

## 5 Case Study

This section introduces an example of a digital twin that is open and simple enough to be easily understood my most researchers and practitioners. The physical twin details are described in [10] and the digital twin is detailed in [8,9]. The content for this section is adapted from [8], with a focus on the communication architecture.

### 5.1 Physical Twin

The incubator system is a traditional control system, comprised of a controller and a plant. The plant is composed of a styrofoam box, a fan, three temperature sensors, and a heating device called a heatbed. Due to the room temperature always being smaller than the desired temperature inside the incubator, whenever the heatbed is off, the temperature inside the box drops. Therefore the controller can regulate the temperature by turning the heatbed on or off. The fan is usually always on to ensure air circulation and therefore avoid exceedingly hot spots inside the box.

The controller communicates with the driver of the plant using a RabbitMQ server, and the driver of the plant communicates with the relays that activate the heatbed and fans using a library. This is summarized in Fig. 12.

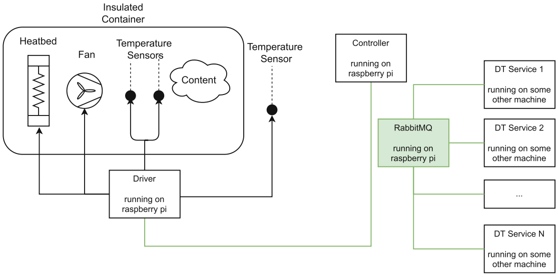

**Fig. 12.** Diagram of the communication among different digital twin services.

### 5.2 Digital Twin

The Digital Twin (DT), in the context of the incubator case study (see [8,9]), consists of a number of services that communicate via RabbitMQ messages, as illustrated in Fig. 12, each with one of the following goals:

**Data Storage.** We use InfluxDB to store the time series data and model parameters.

**Visualization.** We use InfluxDB's web application to create dashboards for querying relevant data streams;

**State Estimation.** We use a Kalman Filter (which uses the model parameters stored in the InfluxDB) to estimate the hidden state of the system (hidden state means variables that are part of the model but are not directly measurable from the plant)

**What-if Simulation.** We use a simulator that can be asked to run hypothetical simulations on past or future extrapolated data.

**Self-adaptation Manager.** The responsibility of this service is to implement a MAPE-K loop [17] that enables optimization of the control parameters whenever something in the environment of the incubator changes (more details in [8]). For example, when the lid is opened, the self adaptation manager will use the Kalman filter to detect an anomaly, and then carry out a number of simulations that attempt to find new parameters for the model. Finally, a new control policy is synthesized based on the newly found model parameters. A control policy refers to the optimal parameters of the controller, according to some cost function.

### 5.3 Example Security Challenges

In this subsection, we give concrete examples of the challenges introduced in Fig. 3 in the context of the incubator.

**Bandwidth Sniffing.** An attacker could use bandwidth sniffing to identify which service are involved in the implementation of the MAPE-K loop, because there is a burst of network activity when an anomaly is detected.

**Data Injection.** There are multiple examples of this attack: an attacker can inject fake sensor measurements into the Kalman filter service, which can lead to an anomaly being detected, which in turn can lead to the synthesis of a potentially unsafe control policy; or the attacker (disguised as the self-adaptation manager) might send a fake packet with a new control policy directly to the controller.

**Data Delay.** It is critical that anomalies are detected as soon as they occur. An attacker might delay the detection of an anomaly until it is too late. An example of this is if the lid of the incubator is open, the control loop is typical on an high power control policy, because of the excessive heat dissipation. If a person closes the lid, the self-adaptation manager typically reacts quickly to change the control policy to a low power mode, to avoid excessive warming in the incubator. Any extra seconds in this process might lead to unsafe temperatures in the incubator.

**Model Corruption.** Models are used almost in all DT services (state estimation, anomaly detection, what-if simulation, and self-adaptation), and the incubator digital twin uses controller models, plant models, and CPS (controller and plant combined) models. Any model manipulation leads to these DT services malfunctioning. For example, an incorrect model causes false anomalies to be detected, and in turn may cause incorrect control policies to be synthesized.

## 6 Open Problems

In this section we use the presented attacks, mitigations and the case study as a basis to discuss open research and engineering topics within digital twin security. The list presented within this section is not exhaustive as we merely aim at pointing at topics that could be acted upon with respect to the current state of the art of the digital twin area. We believe these are good starting points for further contributions to digital twin security.

The attacks presented in this paper are not only applicable to digital twins, but can be applied to wide variety of industrial control systems. In order to provide more targeted solutions for attack mitigation it is important that a clear taxonomy and definitions are created in order to be able to clearly categorize the system as a digital twin enabled or not. Clear informal and formal definitions of different digital twin enabled systems are required. We believe that the increased understanding and clarity will lead to an easier exchange of ideas with security researchers and engineers in the area of digital twins.

Another aspect is the design and development of security-aware protocols specifically for digital twins. We see digital twins as an area where models could be aware of the underlying security, including the data transfer protocol. As such it could provide continuous runtime analysis of the communication between the digital twin and the physical system. Specific challenges that need to be addressed within this topic are the minimization of the overhead such analysis would incur as well as simplicity of the design of such protocols. This approach would provide good security assurances for a data heavy digital twin system connected via an untrusted or a semi-trusted network.

We further see the need for development of a clear generalized notation for reasoning about security challenges of digital twins. In this aspect we have provided several examples in this paper, however we suggest that more work is done in this area and a possible catalogue of cyber attacks and respective mitigations is created. This could be in turn be utilized by the engineering teams developing digital twin enabled systems to semi-formally, yet clearly communicate the security aspects of the systems they create. We think, that such notation would provide a clear way of communication during the engineering of digital twin systems.

We would also suggest utilization of formal methods for analysis of different aspects of security of digital twin systems. Different attacks and mitigations could be expressed formally and applied to the formal model of the digital twin system. This could contribute to the development of a catalogue providing formal models in the area of digital twins where suitable. As security attacks are very broad and need to consider, e.g., aspects of physical materials used or social aspects about people involved, formal models will only cover some aspects of the overall security concerns.

Finally an investigation into the complementary nature of security methods based on anomaly detection and state estimation and traditional security protocols that could be utilized within the digital twin area is necessary. As we have discussed earlier, the large amount of communication required for operating digital twins means that the overhead must be kept low. Providing new options for combining these complementary methods will help to reduce the overhead.

## 7 Concluding Remarks

In this paper we have discussed security challenges and possible mitigations for digital twin enabled systems. We have described four specific kinds of challenges that such systems face and introduced mitigations for these challenges. To address these challenges in a way acceptable in practice, the defining characteristics of digital twin enabled systems need to be taken into consideration. Otherwise, implemented security measures might render a digital twin system inoperable. We have outlined several open problems, answers to which, will provide functioning security for digital twin systems. Besides gaining a better understanding about what comprises digital twin system, what different kinds of such systems must be considered, a catalogue of relevant security challenges and mitigations is needed. The four challenges that we have discussed can be a starting point for this, focusing on specific needs for digital twins. In the presentation of the challenges we have used semi-formal notation to state the challenges more clearly and help to communicate them with stakeholders of digital twin systems with diverse (engineering) backgrounds. Such a notation can serve to document the challenge in such way that they can easily be communicated widely.

**Acknowledgements.** We are grateful to the Poul Due Jensen Foundation, which has supported the establishment of a new Centre for Digital Twin Technology at Aarhus University.

## References

1. Atalay, M., Angin, P.: A digital twins approach to smart grid security testing and standardization. In: 2020 IEEE International Workshop on Metrology for Industry 4.0 IoT, pp. 435–440 (2020). https://doi.org/10.1109/MetroInd4.0IoT48571.2020.9138264
2. Badhwar, R.: The case for code signing and dynamic white-listing. In: The CISO's Next Frontier, pp. 259–264. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-75354-2_32
3. Baskar, M., Jayaraman, R., Karthikeyan, C., Anbarasu, V., Balaji, A., Arulananth, T.: Low rate DDoS mitigation using real-time multi threshold traffic monitoring system. J. Ambient Intell. Humanized Comput., 1–9 (2021). https://doi.org/10.1007/s12652-020-02744-y
4. Becue, A., et al.: Cyberfactory# 1-securing the industry 4.0 with cyber-ranges and digital twins. In: 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), pp. 1–4. IEEE (2018)
5. Bianchin, G., Pasqualetti, F.: Time-delay attacks in network systems. In: Koç, Ç.K. (ed.) Cyber-Physical Systems Security, pp. 157–174. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98935-8_8
6. Bruynseels, K., Santoni de Sio, F., van den Hoven, J.: Digital twins in health care: ethical implications of an emerging engineering paradigm. Front. Genet. **9** (2018). www.frontiersin.org/article/10.3389/fgene.2018.00031. https://doi.org/10.3389/fgene.2018.00031
7. Eckhart, M., Ekelhart, A.: Towards security-aware virtual environments for digital twins. In: Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, CPSS 2018, pp. 61–72. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3198458.3198464
8. Feng, H., Gomes, C., Gil, S., Mikkelsen, P.H., Tola, D., Larsen, P.G.: Integration of the MAPE-K loop in digital twins. In: 2022 Annual Modeling and Simulation Conference (ANNSIM). IEEE, San Diego, July 2022
9. Feng, H., Gomes, C., Thule, C., Lausdahl, K., Iosifidis, A., Larsen, P.G.: Introduction to digital twin engineering. In: 2021 Annual Modeling and Simulation Conference (ANNSIM), pp. 1–12. IEEE, Fairfax, July 2021. https://doi.org/10.23919/ANNSIM52504.2021.9552135
10. Feng, H., Gomes, C., Thule, C., Lausdahl, K., Sandberg, M., Larsen, P.G.: The incubator case study for digital twin engineering. arXiv:2102.10390 [cs, eess], February 2021
11. Golovina, T., Polyanin, A., Adamenko, A., Khegay, E., Schepinin, V.: Digital twins as a new paradigm of an industrial enterprise. Int. J. Technol. **11**(6), 1115 (2020). https://doi.org/10.14716/ijtech.v11i6.4427
12. Holmes, D., Papathanasaki, M., Maglaras, L., Ferrag, M.A., Nepal, S., Janicke, H.: Digital twins and cyber security - solution or challenge? In: 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), pp. 1–8 (2021). https://doi.org/10.1109/SEEDA-CECNSM53056.2021.9566277

13. Jackson, M.: Software Requirements and Specification: A Lexicon of Practice, Principles and Prejudices. Addison-Wesley (1995)
14. Jackson, M.: Problem Frames. ACM Press (2001)
15. Kayan, H., Nunes, M., Rana, O., Burnap, P., Perera, C.: Cybersecurity of industrial cyber-physical systems: a review. ACM Comput. Surv. (2022). https://doi.org/10.1145/3510410
16. Kazemi, Z., Safavi, A.A., Naseri, F., Urbas, L., Setoodeh, P.: A secure hybrid dynamic-state estimation approach for power systems under false data injection attacks. IEEE Trans. Industr. Inf. **16**(12), 7275–7286 (2020). https://doi.org/10.1109/TII.2020.2972809
17. Kephart, J., Chess, D.: The vision of autonomic computing. Computer **36**(1), 41–50 (2003). https://doi.org/10.1109/MC.2003.1160055
18. Krutz, R.L.: Securing SCADA Systems. John Wiley & Sons (2005)
19. Kulik, T., Boudjadar, J., Tran-Jørgensen, P.: Security verification of industrial control systems using partial model checking. In: Proceedings of the 8th International Conference on Formal Methods in Software Engineering, FormaliSE 2020, pp. 98–108. Association for Computing Machinery, United States (2020). 8th International Conference on Formal Methods in Software Engineering; Conference date: 07 October 2020 Through 08 October 2020. https://doi.org/10.1145/3372020.3391558
20. Mo, Y., Chabukswar, R., Sinopoli, B.: Detecting integrity attacks on SCADA systems. IEEE Trans. Control Syst. Technol. **22**(4), 1396–1407 (2013)
21. Pajic, M., et al.: Robustness of attack-resilient state estimators. In: 2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), pp. 163–174. IEEE, Berlin, April 2014. https://doi.org/10.1109/ICCPS.2014.6843720
22. Paridari, K., O'Mahony, N., El-Din Mady, A., Chabukswar, R., Boubekeur, M., Sandberg, H.: A framework for attack-resilient industrial control systems: attack detection and controller reconfiguration. Proc. IEEE **106**(1), 113–128 (2018). https://doi.org/10.1109/JPROC.2017.2725482
23. Pöhls, H.C.: JSON sensor signatures (JSS): end-to-end integrity protection from constrained device to IoT application. In: 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 306–312 (2015). https://doi.org/10.1109/IMIS.2015.48
24. Suhail, S., Hussain, R., Jurdak, R., Hong, C.S.: Trustworthy digital twins in the industrial internet of things with blockchain. IEEE Internet Comput. (2021)
25. Suhail, S., Zeadally, S., Jurdak, R., Hussain, R., Matulevičius, R., Svetinovic, D.: Security attacks and solutions for digital twins. arXiv preprint arXiv:2202.12501 (2022)
26. Tian, J., Tan, R., Guan, X., Xu, Z., Liu, T.: Moving target defense approach to detecting Stuxnet-like attacks. IEEE Trans. Smart Grid **11**(1), 291–300 (2020). https://doi.org/10.1109/TSG.2019.2921245
27. Uhlemann, T.H.J., Lehmann, C., Steinhilper, R.: The digital twin: realizing the cyber-physical production system for industry 4.0. Procedia CIRP **61**, 335–340 (2017). https://doi.org/10.1016/j.procir.2016.11.152. www.sciencedirect.com/science/article/pii/S2212827116313129. The 24th CIRP Conference on Life Cycle Engineering
28. Xiong, S., Sarwate, A.D., Mandayam, N.B.: Defending against packet-size side-channel attacks in IoT networks. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2027–2031 (2018). https://doi.org/10.1109/ICASSP.2018.8461330

29. Yan, W., Hou, E., Ansari, N.: Defending against traffic analysis attacks with link padding for bursty traffics. In: Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, pp. 46–51 (2004). IEEE (2004)
30. Zonouz, S., Rrushi, J., McLaughlin, S.: Detecting industrial control malware using automated plc code analytics. IEEE Secur. Priv. **12**(6), 40–47 (2014). https://doi.org/10.1109/MSP.2014.113