Inter-device Sensor-Fusion for Action Authorization on Industrial Mobile Robots

Sarah Haas[1(✉)], Andrea Höller[1], Thomas Ulz[2], and Christian Steger[2]

[1] Development Center Graz, Infineon Technologies Austria AG, Graz, Austria
{sarah.haas,andrea.hoeller}@infineon.com
[2] Institute for Technical Informatics, Graz University of Technology, Graz, Austria
{thomas.ulz,steger}@tugraz.at

Abstract. Usage of mobile robots in industry increased significantly in recent years. However, mobile robots introduce additional safety issues for human workforce and pose a higher risk of failures in production due to possible abnormal robot behavior. Such abnormal behavior could, among other things, be caused by security weaknesses that entail attacks. These problems lead to a need for action authorization mechanisms to protect humans and mitigate possible costly failures. In this paper, we propose an authorization mechanism for critical actuator actions on industrial mobile robots. The mechanism relies on security principles that prevent adversaries from unauthorized action execution. To the best knowledge of the authors, no similar concept for secured action authorization for industrial mobile robots is currently known in research. Our evaluation shows more than 80% of additional safety hazard causes introduced by the lack of security can be mitigated with the proposed authorization mechanism.

1 Introduction

The increased automation in production facilities entails a rapid rise of mobile robots in industrial applications. The number of mobile robots in industrial automation will be even higher in future *Smart Factories* [27] and Industry4.0 environments. Mobile robots in future production facilities will typically interact with machines, humans, or other robots to fulfill any given task such as fetching material or delivering material. A research testbed for smart factory related technologies is the RoboCup Logistics League [21]. A mobile robot in this testbed basically consists of actuators, sensors, a central computing unit, and a communication unit. The actuators are used to interact with the physical world. The sensors are used to gain information about the environment. The central computing unit processes incoming and outgoing data, coordinates the components, and instructs them. The communication unit acts as router that connects all components on a mobile robot, and provides technologies to communicate to other devices via, e.g. Industrial WiFi.

The advantage of using a router in a robotic system also entails the advantage of modularity and easy connection of components. However, a disadvantage is that it would be easy for an adversary to access sensors and actuators by simply sending commands via the router, or even hijack a robot. It is generally easier to attack devices that communicate wirelessly. Especially Industry4.0 environments will deal with larger attack surface due to the increased connectivity, usage of wireless communication, increased data exchange, and many more [11]. Besides that, the safety of human workforce cannot be ensured anymore if an adversary is able to hack a robot. To support the safety of human workforce, security needs to be introduced [4,8,15]. Since the interaction between mobile robots and machines is a typical use case, and will occur in close proximity to human workforce, secured authorization of actuator actions needs to be performed. Actuator actions might include manipulations of production material with a robotic arm, or movement of production material between machine and mobile robot. If such actions can be initiated or manipulated by an adversary, human workforce will be exposed to serious safety hazards.

To prevent such malicious manipulations, we propose a secured authorization mechanism for critical actions on industrial mobile robots when interacting with other entities. The authorization approach uses available sensors to gain information about the current environmental states such as the distance between a mobile robot and a machine. This information is combined with security mechanisms to prevents adversaries from injecting or manipulating false action commands, and also prevent the mobile robot from executing harmful actions caused by software bugs or other errors.

To the best knowledge of the authors, no concept similar to the proposed one is currently known in research. The related work shows several applications for multi-sensor fusion and authorization concepts. However, none have combined these concepts before. Furthermore, none of the concepts take security into account even though the safety could easily be compromised by security weaknesses in many of these scenarios.

To summarize, the contributions of this paper are:

- The first action authorization approach for industrial mobile robots that relies on inter-device sensor-fusion and cryptographic principles to support the safety of human workforce.
- A combined safety and security analysis showing that more than 80% of the additional safety hazard causes can be mitigated by introducing the proposed authorization approach.

2 Related Work and Background

Secure Element

A secure element (SE) stores confidential data such as key material and is able to perform cryptographic operations such as signature computation or hash computation. General purpose microcontrollers or CPUs are typically prone to side channel attacks that spy on calculation times depending on the input or try to physically manipulate the CPU to compromise calculations that might reveal confidential data. SEs, in contrast, are tamper-resistant which means that they are built to withstand such attacks and are, therefore, used for security critical applications such as bank cards or trusted computing in, e.g. Trusted Platform Modules [1].

One-Time Passwords
One-Time Passwords (OTPs) are password schemes where a password becomes invalid after its first use and were introduced by Lamport [14]. Lamport intended to overcome issues with plain text passwords such as interception of plain text passwords by adversaries and later, as a countermeasure against replay attacks. Such attacks capture a user's login credentials and use them to access a system [10]. OTPs use non-invertible cryptographic hash functions such as SHA-256 to create passwords. It is necessary that the same non-invertible cryptographic hash function is available on the client and the host for OTP generation and verification.

In 2005, M'Raihi et al. [19] proposed OTPs based *Hashed Message Authentication Codes* (HMAC) called *HOTP*. The authors introduced a counter that is combined with the secret key and forwarded to the hash function. The counter is synchronized with a trusted entity such as a server to enable the verification of the OTP. The counter is incremented by a specific amount known by the client and server every time an OTP is generated or validated. The counter enables individual passwords for unchanged data, and the secret key enables the authenticity of the client and server.

The tickets used for the authorization approach are based on HOTPs that provide integrity and authenticity. HOTPs use a secret key and moving factors such as a counter value to prevent the possibility of replay attacks. The secret key is used to protect the hashes from brute-force attacks on the counter value, and enables authenticity. The counter is necessary since the data exchanged between robot and machine might be identical. The same data would always result in the same valid hash value. If an adversary would capture a valid ticket sent by the robot, he could send it to the machine over and over again, and the machine would authorize the actions. To generate different tickets for identical data, the counter values are used to generate passwords that are only valid once. One might also use signatures instead of HOTPs, however, the problem of the exact same signature for equal input data would remain the same, and the operation would also require some kind of moving value for individual signature values. Furthermore, asymmetric cryptographic calculations such as signatures are much slower than symmetric cryptographic calculations such as HMACs.

Multi-sensor Fusion
Multi-sensor fusion is used to combine the data provided by several different sensors or other data sources to improve accuracies and robustness [9]. The concept of multi-sensor fusion has been used for years in a wide range of areas including artificial intelligence, medical diagnostics, environmental monitoring, robotics, and much more. Especially mobile robots strongly rely on multi-sensor

fusion since they deal with localization problems, odometry inaccuracies, and other many other problems [13,22].

Kam et al. [12] reviewed existing sensor fusion techniques for robot navigation back in 1997, especially addressing self-localization in maps constructed by the robot. Recent research by Lynen et al. [16] also addressed multi-sensor fusion for navigation and self-localization purposes in a framework that is able to process any absolute, relative, or delayed data from an almost unlimited amount of sensors. Even though, sensor fusion is widely used in robotic applications, it was, as far as we know, never used for authorization mechanisms before.

Authorization for Mobile Robots
Authorization was defined as granting privileges to processes or users by Fraser in 1997 [6] and is used widely in any operating system, company network or production system. Current research focuses on topics such as authorization and access rights in cloud environments [25] or Internet of Things (IoT) systems [5]. In the mobile robotics domain, authorization is not a key topic. A very simple authorization mechanism was shown by Gonçalves et al. [7]. The authors proposed a realistic sensor and actuator model for wheeled mobile robot simulations that included a boolean register whose value was checked before executing an action on a robotic arm. As far as we know, the only approach that includes authorization related to robotics and other mobile devices was proposed by Popovici et al. in 2003 [23]. The authors proposed a middleware platform for mobile devices that uses an authorization mechanism to prevent unauthorized entities from executing actions on, e.g., a robotic arm. This middleware checks an entities' rights to execute actions in a physical system but does not include any current environmental information.

To the best knowledge of the authors, none of the existing approaches use sensor-fusion or security measures in their authorization approaches, or would even combine these topics.

3 Proposed Authorization Mechanism

The interaction between robots and machines is a typical scenario that will occur in smart factories. If interactions between these entities is unauthorized, serious safety hazards for humans can occur, and production material could be damaged. Therefore, the approach proposed in this paper assures that only authorized actions are executed by actuators to support safety. The proposed mechanism uses the sensor data of both, robot and machine, to make sure that the robot is authorized to, for example, drop off production material on a machine. Using the combined sensor data instead of just the sensor data from the robot can prevent critical actions from being performed in case of errors on the robot or malicious manipulations, to protect human workforce and production material. The sensor data of the robot and a machine are combined and checked. If for example, the laser scanner values of the robot lie within a certain range while approaching a machine, the machine is notified. The machine would then check the light barrier on its input. If the light barrier is interrupted, the machine notifies the robot. The robot could then generate an authorization ticket and send it to the actuator. The actuator executes the command if the authorization ticket is valid. Each authorization ticket expires after it was used to overcome issues with replay attacks. The sensors in this paper are assumed to be trustworthy since this topic would exceed the scope of this paper, and other researchers already focus on sensor trustworthiness [18,24,26]. The required components to successfully execute an action are the central computing units (CCU) of both robot and machine, the robot's SE, the actuator's microcontroller, the actuator's SE, and the machine's SE. To perform action authorization, the following preconditions need to be fulfilled. (1) Robot, actuators and machine, are equipped with a SEs to store key material and securely compute and verify HOTPs. (2) Robot's and actuator's SEs share a secret key K_A and a counter cnt_A. (3) Machine's and robot's SE share a secret key K_M and a counter cnt_M. (4) Robot's SE and the sensors share secret keys K_S and counters cnt_S. (5) All secret keys and counter values are already stored in the corresponding SEs or sensors.

3.1 Authorization Approach

The authorization approach is divided into 15 steps that can be seen in Table 1. The following section describes the authorization process in detail. In the text, the numbers in brackets refer to the line numbers in Table 1. Each instruction colored in red means that the calculation is done in an SE. The function $HOTP_G$ refers to the generation of an HOTP on a SE and also increments the counter values cnt_M, cnt_R or cnt_S. The $HOTP_G$ function requires a secret key, a counter and some data as inputs. The function $HOTP_V$ refers to a validation of an HOTP in an SE and also increments the counter values cnt_M, cnt_R or cnt_S. The validation of an HOTP can be done if the secret key, counter and data, as well as the HOTP to validate against, are provided as input. The secret key, counter and data must match the values used to generate the initial HOTP. Otherwise, the HOTP will not be valid. The *inRange* function checks whether the sensor data fulfills pre-defined conditions on the robot. The $fuse$ function fuses the sensor readings of the robot and machine, and checks if the pre-defined conditions for the robot's and machine's sensor readings are fulfilled. The function *ReqSenData* requests the sensor data from one or more sensors.

The authorization is initiated by the robot's CCU. The robot's CCU requests data from any sensor, e.g. the laser scanner to compute the distance from the next obstacle or the distance to equipment on the production floor. The sensor generates an HOTP over the data using a counter value and secret key, and provides the HOTP and data sd_R to the robot's CCU (1). The sensor's HOTP is validated by the robot's SE (2). The robot's CCU checks if the sensor data satisfies certain pre-defined conditions (3). A possible condition would be the distance measured by a laser scanner. If the distance is within a specific range, the condition is satisfied. If the sensor data's HOTP was valid, and the sensor data was in a certain range, the command *cmd* and sensor data sd_R are sent to the robot's SE. The received data is combined with the secret key K_M and counter cnt_M, and the request ticket *hotp* is generated (4). The command *cmd*, sensor data sd_R and hash *hotp* are sent to the machine (5). The machine's CCU

Table 1. Sequence diagram of a complete authorization process.

	Machine	**Robot**	**Actuator**
1 :		$sd_R \leftarrow ReqSenData()$	
2 :		$v \leftarrow HOTP_V(sd_R)$	
		$R \leftarrow K_M, cnt_M, cmd, sd_R$	
		$in \leftarrow inRange(sd_R)$	
3 :		**if** v AND in	
4 :		$\quad hotp \leftarrow HOTP_G(R)$	
5 :		$\xleftarrow{Send\ hotp, cmd, sd_R}$	
6 :	$sd_M \leftarrow ReqSenData()$		
7 :	$v_1 \leftarrow HOTP_V(hotp)$		
8 :	$v_2 \leftarrow HOTP_V(sd_M)$		
	$M \leftarrow K_M, cnt_M, hotp$		
	$in \leftarrow fuse(sd_M, sd_R)$		
9 :	**if** v_1 AND v_2 AND in		
10 :	$\quad auth \leftarrow HOTP_G(M)$		
11 :	$\xrightarrow{Send\ auth}$		
12 :		**if** $HOTP_V(auth)$	
		$\quad R \leftarrow K_A, cnt_A, cmd$	
13 :		$\quad act \leftarrow HOTP_G(R)$	
14 :		$\xrightarrow{Send\ act, cmd}$	
15 :			**if** $HOTP_V(act)$
			$\quad execute(cmd)$

requests the sensor data sd_M (6), and instructs the SE to validate both, the request ticket *hotp* and the sensor data's hash (7, 8). The sensor data of both robot and machine are passed to the $fuse$ function to perform the sensor fusion, and necessary checks on the sensor readings. The $fuse$ function simply returns $true$ if the data is valid or $false$ if the was invalid (8). If both hashes *hotp* and sd_M are valid, and the requested sensor data fulfilled the preconditions (9), the response ticket *auth* is generated by the SE (10). The response ticket is sent to the robot (11) and the robot's SE validates the received response ticket (12). If the ticket was valid, the robot's SE generates an authorization ticket *act* for the actuator (13). The authorization ticket *act* and command *cmd* are sent to the actuator (14) and if the actuator's SE confirms the validity of the received authorization ticket *act*, the command *cmd* is executed (15).

4 Implementation Remarks

This section discusses a proof-of-concept implementation of our proposed authorization approach, and includes explanations regarding the sensors and SEs.

4.1 Proof-of-Concept Implementation

As a proof-of-concept, our proposed authorization approach was implemented using several Raspberry Pi 3 and SEs by Infineon Technologies. Figure 1 shows the setup of the implementation. The sensors are simple simulations but calculate HOTPs for their measurements. The Raspberry representing the machine and the Raspberry representing the CCU are both equipped with SEs, and communicate via a router with WiFi. The Raspberry representing the actuator is also equipped with an SE, and is connected via a router with Ethernet to the Raspberry representing the CCU.

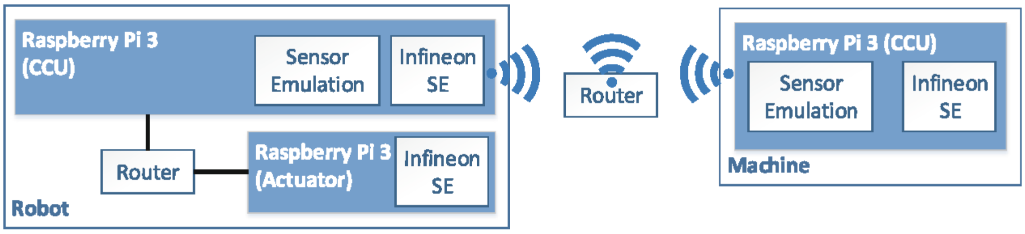

Fig. 1. Setup for the proof-of-concept with several Raspberry Pi 3 and SEs.

The proof-of-concept shows that the introduction of SEs in this scenario causes a significant increase of the overall runtime of one authorization. The mechanisms that provide tamper-resistance to such devices cause the overhead. However, in industrial use cases, the protection of key material and other confidential information, and the introduction of security is of utmost importance to support safety of human workforce and prevent damage on production material. Since mobile robots might perform real-time operations, the increased latency caused by the introduced security measures need to be taken into account when defining any real-time condition. However, since our proposed authorization mechanism is deterministic, the additional overhead can be calculated in advance. The overhead highly depends on the used hard- and software which makes it very hard to give a generally accepted assumption about the overhead. The overhead analysis would exceed the scope of this paper since the focus of this paper is to support the safety of human workforce by introducing security measures. The overhead analysis as well as possible optimizations to reduce the overhead are postponed to future work. One possibility for latency reduction would be to already start the authorization process while the robot approaches the machine, meaning that the steps taken on the robot before communicating to the machine can be done in parallel to the actual task of the robot. Another possibility would be the direct forwarding of the authorization ticket from the machine to the actuator. Furthermore, the machine could receive sensor readings periodically, meaning it does not need to actively request them. These optimizations would already reduce the latency significantly.

4.2 Application of SEs in the Authorization Approach

The SEs in this scenario are, in principle, optional since each HOTP computation could also be done by a general purpose CPUs in plain software. However, if an attacker is able to have physical access to a general purpose CPU, he could perform side-channel attacks that reveal the secret key used for HOTP generation. The SEs are used to increase security by protecting the key material, counter and also the password generation process from such side-channel attacks. Physical attacks tend to reveal confidential information by analyzing the devices interface using, e.g. power or time analysis attacks, or try to physically attack the device by bombarding them with a laser to generate an error. Remote attacks tend to reveal data stored in software such as files or folders, or try to hijack a device. SEs cannot protect a device from remote attacks but can protect the secret keys used to perform crpytographic operations since the secret key cannot be read from the device. The question whether an SE is necessary highly depends on the actual use case and desired security level. Furthermore, SEs introduce a trade-off between latency and security level since SEs tend to be much slower than general purpose CPUs due to the implemented protection mechanisms against side-channel attacks.

5 Combined Safety and Security Analysis

To highlight the safety enhancing security features of our proposed approach, a combined safety and security analysis is conducted. The analysis uses the basic idea of a combined safety and security as suggested by Macher et al. [17] for the automotive domain. However, the functional safety analysis is adopted to match the ANSI/RIA R15.06 norm [2] for industrial robot safety. For the security analysis, a threat analysis [20] including countermeasures enabled by the proposed authorization approach is executed. The threats are then categorized similar to the risk level determination in the ANSI/RIA R15.06. The combination of the safety and security analysis show that additional causes for the existing safety threats arise from a lack of security. However, the analysis also shows that the proposed authorization mechanism reduces the additional safety hazard causes significantly. Before the analysis is performed, assumptions on the attacker and attack possibilities are made. (1) Taking over a robot is assumed to be possible since the equipped wireless communication technology opens a wider attack surface, and the attacker can directly attack the robot without the need to infiltrate the company network. (2) Taking over a machine remotely is assumed to be not attractive for an attacker since machines are connected by wire to the factory network, meaning that an attacker would have to gain access to the network, and then would need to make it to the machine. It is assumed to be too much of an effort since an attacker would attack a more attractive target such as a central server rather than one specific machine. (3) An attacker trying to locally extract the secret keys from machine or robot is assumed to be possible since an attacker would then be able to read and send commands to devices and into the network using valid secret keys. Extracting the keys would not require to take over a machine or robot since it can be done with different side-channel attacks.

5.1 Methodology

This section shows how identified threats are categorized similar to the ANSI/RIA R15.06 risk assessment [2] to show the severity of the identified threats.

Table 2. Classification of required resources *RR* to execute a threat.

Level	Resource	Example
RR0	No tool	Manipulation of sensors with Mirror, Laser Pointer, Tape, etc.
RR1	Simple tool	Laptop, Smart Phone for network access, etc.
RR2	Standard tool	Network Sniffer, Oszilloscope for message capturing, power or time analysis attacks, etc.
RR3	Advanced tool	MITM tools, tools for targeting attacks for manipulation of messages, advanced physical attacks, etc.

Table 3. Classification of required know-how *RK* to execute a threat.

Level	Know-How	Example
RK0	Basic	Functionality of sensors, accessing of networks, use of physical interfaces, etc.
RK1	Advanced	Basic physical/remote Attack know-how, network know-how, e.g. protocols, power analysis etc.
RK2	Expert or Insider	Advanced remote/physical attack Know-how, e.g. targeting attacks, MITM, etc.

Table 4. Classification of required accessability *RA* to execute a threat.

Level	Access	Example
RA0	Remote	Other country, outside facility, etc.
RA1	Local	Physical presence at attacked target

The threat level determination based on the risk assessment determination described in the ANSI/RIA 15.06 [2] utilizes the categories required resource *RR*, required know-how *RK* and required accessibility *RA*. The *RR* gives examples of the required tools to successfully deploy the security threat (see Table 2).

The RK defines the necessary know-how an attacker has to have to successfully execute the attack (see Table 3). The RA defines if an attack can be launched remotely, or if an attacker needs to be physically present to execute an attack (see Table 4). These categories are used to identify the severity of a threat and determine its threat level as shown in Table 5. The higher the threat level, the more severe a threat is when exploited in a system.

Table 5. Threat level determination matrix depending on the required resource, know-how and access based on the ANSI/RIA R15.06 risk level determination matrix.

Required	Required	Required	Threat Level				
Resource	Know-How	Accessibility	neglible=0	low=1	medium=2	high=3	very high=4
RR0	RK0	RA2	0				
RR0	RK1	RA2		1			
RR0	RK2	RA2			2		
RR1	RK0	RA1/RA2		1			
RR1	RK1	RA1/RA2			2		
RR1	RK2	RA1/RA2			2		
RR2	RK0	RA1/RA2		1			
RR2	RK1	RA1			2		
RR2	RK1	RA2				3	
RR2	RK2	RA1				3	
RR2	RK2	RA2					4
RR3	RK0	RA1/RA2			2		
RR3	RK1	RA1				3	
RR3	RK1	RA2					4
RR3	RK2	RA1/RA2					4

5.2 Threat Analysis and Threat Level Determination

To apply the defined methodology on our proposed authorization approach, we perform a threat analysis according to Myagmar et al. [20], and determine the threat level of each threat. Table 7 lists the identified threats T, countermeasures C and remaining residual risks R. Table 6 shows the determination of the threat level for each identified threat.

5.3 Results

For a mobile robot with a robotic arm, the safety analysis identified a total of 27 hazards with 38 safety hazard causes. The safety hazards are inspired by Bartos [3] and the ANSI/RIA 15.06 norm [2]. The safety hazards for the machine are not listed separately since they are a subset of the safety hazards identified for the mobile robot. Since security threats can also cause safety hazards, each threat was applied to the safety hazard scenarios to check whether the security threat could cause a safety hazard.

Table 6. Threat level determination for the identified security threats.

Threat	Required Resource	Required Know-How	Required Accessibility	Threat Level
T1	RR2	RK1	RA1	2
T2	RR2	RK1	RA2	3
T3	RR1	RK1	RA1	2
T4	RR3	RK2	RA1	4
T5	RR1	RK1	RA1	2
T6	RR0	RK1	RA2	1
T7	RR1	RK0	RA1	1
T8	RR2	RK1	RA1	2
T9	RR3	RK2	RA1	4
T10	RR3	RK1	RA1	3
T11	RR1	RK0	RA1	1

For the proposed authorization approach, the combined analysis shows that 10 of the 27 identified hazards could also be caused by the identified security threats from Table 7. The left hand side of Table 8 shows the safety analysis according to the ANSI/RIA R15.06 risk level assessment for all safety hazards that can also be caused by security threats. The right hand side of Table 8 shows the corresponding security threats that can cause the safety hazard. To identify which safety hazard can be caused by malicious actions of an attacker, each security threat listed in Table 7 was applied to each safety hazard. As the analysis shows, 10 safety hazards can also be caused when security threats are exploited. As an example, safety hazard #13 where the laser scanner is blinded, can intentionally be caused by the two security threats $T7$ where an attacker would physically manipulate the sensor using e.g. tape, and $T11$ where the attacker would perform a DoS attack to prevent the laser scanner from sensor readings by flooding it with messages. The other listed security threats cannot cause this safety hazard.

The combined safety and security analysis shows that for 10 safety hazards a total of 43 additional causes due to the lack of security can be identified for an insecure authorization scenario. The bar chart in Fig. 2 shows the number of total additional safety causes and the number of mitigated causes when applying the proposed secured action authorization for each risk level. The bar on the left hand side in blue shows the number of total additional safety hazard causes for each risk level, and the bar on the right hand side in green shows the mitigated causes.

The threat analysis lists countermeasures enabled by our proposed security-enhanced action authorization approach. The introduced security measures reduce the number of additional safety hazard causes from 43 to 7 for the proposed authorization approach. 36 additional safety hazard causes can be mitigated by our proposed authorization mechanism according to the countermeasures identified in Table 7. The 7 remaining additional causes are all related to

Table 7. Threat analysis of the proposed authorization mechanism. The most important threats, possible countermeasures and remaining residual risks are listed.

Threat	Countermeasure or Residual Risk
(T1) Backdoors in SE, either intentional or unintentional, on robot or machine. Weak implemented or wrong cryptography in SE.	(C1) SEs are certified for a specific security level to prevent the existence of backdoors, and check that strong and correct cryptographic algorithms are used.
(T2) Physical attack (e.g., side-channel attacks) to reveal the secret key on a robot or machine.	(C2) SE provides tamper resistance; therefore, the shared secret is protected from being extracted by an attacker.
(T3) Injection of false sensor data on the machine or robot.	(C3) The computed sensor data hashes prevent manipulations during transfer.
(T4) Manipulation of request ticket, response or authorization ticket during transmission.	(C4) The integrity provided by the generated HOTPs would reveal the manipulations and prevent the action execution.
(T5) Injection of false commands to execute an action.	(C5) The approach prevents execution of commands with missing or invalid tickets since the ticket cannot be validated.
(T6) Physical manipulation of a robot's or a machine's sensor to generate false sensor values.	(C6) The sensors would generate a matching hash for the sensor values. However, the action would not be authorization since the non-manipulated sensor's values would not fulfill the given condition.
(T7) Denial-of-Service (DoS) attack on communication interfaces or tickets.	(R7) DoS attacks would prevent a system from executing tasks since the system is overloaded with request or data. These attacks cannot be mitigated by any security mechanism as they don't try to manipulate a service but shut it down.
(T8) Replay attack on request, response or authorization ticket.	(C8) The counter prevents old tickets from being valid since the HOTP expires after the first use.
(T9) Relay attack on wireless communication interface.	(C9) Relay attacks are related to man-in-the-middle attacks. The adversary tries to act as if he was a valid device. However, the attacker is not in possession of the secret key and cannot generate a valid ticket.
(T10) Manipulation of sensor data on the machine or robot during transmission.	(C10) The sensors use HMACs for their values to provide integrity and prevent manipulations during transmit.
(T11) DoS attack on sensors.	(R11) DoS attacks on sensors would prevent the robot or machine from acquiring sensor measurements. These attacks cannot be mitigated by any security mechanism.

DoS attacks caused by the threats $T7$ and $T11$. These attacks would shut down the authorization process since the communication interface or CPU cannot execute tasks anymore due to a huge amount of incoming data. Both causing threats $T7$ and $T11$ were categorized with a low threat level, meaning that the damage

Table 8. Risk level determination of safety hazards with additional causes due to security threats. Threats marked red remain as additional hazards after application of the security measures.

#	Safety Hazard	Injury Severity	Exposure	Prob. of Avoidance	Risk Level	Security Threats causing the Hazard
1	Robot tips over	S3	E2	A1	high	T1-6, T8-T10
2	Person struck by robot or arm	S3	E1	A1	high	T5, **T11**
3	Robot strikes object	S3	E2	A2	high	T1-6, T8-T10
7	Failure of obstacle avoidance system	S2	E2	A1	medium	T3, T6, T10, **T11**
13	Laser scanner blinded	S2	E2	A1	medium	T6, **T11**
14	False Sensing	S2	E2	A1	medium	T3, T6, T10, **T11**
19	CPU overheat and shut down	S2	E2	A1	medium	**T7**
22	Tactile system failure	S3	E2	A3	medium	T3, T6, T10, **T11**
23	Bumper collision avoidance failure	S3	E1	A1	very high	T3, T6, T10, **T11**
26	Laser energy hazard to person's eyes	S1	E1	A2	low	T1, T2, T5, T8

to a system when the threat is executed is low since shutting down the authorization process means that no action execution is performed on the robot or machine.

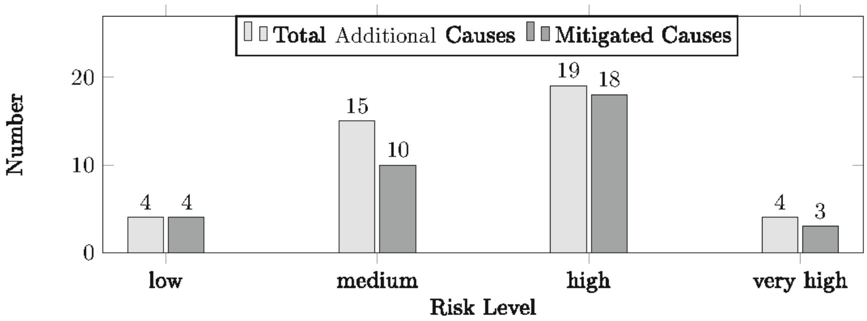

Fig. 2. Numbers of total additional safety hazard causes introduced by a lack of security, and number of additional safety hazard causes mitigated by the proposed authorization approach for each risk level.

6 Conclusion

In this paper, a sensor-fusion based authorization mechanism for industrial mobile robots and equipment on a production floor is shown. The mechanism fuses the sensor data of multiple devices to validate the physical presence of the robot at the corresponding machine, and that only authorized actions are performed to protect human workforce and in further consequence production material from damage or harm. The mechanism is supported by SEs to increase the security and protect the key material from being revealed by an adversary.

The combined safety and security analysis shows that a significant number of additional safety hazard causes is introduced by a lack of security. The analysis shows that around 83% of the additional safety hazard causes due to a lack of security can be mitigated with the proposed secured authorization mechanism.

Acknowledgment. This work has been performed in the project Power Semiconductor and Electronics Manufacturing 4.0 - (Semi40), under grant agreement No 962466. The project is cofunded by grants from Austria, Germany, Italy, France, Portugal and - Electronic Component Systems for European Leadership Joint Undertaking (ECSEL JU).

References

1. ISO/IEC 11889-1 Trusted platform module library - Part 1: Architecture, August 2015
2. Robotic Industries Association: ANSI/RIA R15.06-2012 AmericanNational Standard for Industrial Robots and Robot Systems - Safety Requirements. Technical report (2013)
3. Bartos, R.J.: System safety analysis of an autonomous mobile robot. Technical report, Fernald Environmental Restoration Management Corp., Cincinnati, OH (United States). Fernald Environmental Management Project (1994)
4. Bloomfield, R., Netkachova, K., Stroud, R.: Security-informed safety: if it's not secure, it's not safe. In: Gorbenko, A., Romanovsky, A., Kharchenko, V. (eds.) SERENE 2013. LNCS, vol. 8166, pp. 17–32. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40894-6_2
5. Cirani, S., Picone, M., Gonizzi, P., Veltri, L., Ferrari, G.: IoT-OAS: an OAuth-based authorization service architecture for secure services in IoT scenarios. IEEE Sens. J. **15**(2), 1224–1234 (2015)
6. Fraser, B.Y.: Site Security Handbook (1997). RFC2196
7. Gonçalves, J., Lima, J., Oliveira, H., Costa, P.: Sensor and actuator modeling of a realistic wheeled mobile robot simulator. In: IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2008, pp. 980–985. IEEE (2008)
8. Grieco, L.A., et al.: IoT-aided robotics applications: technological implications, target domains and open issues. Comput. Commun. **54**, 32–47 (2014)
9. Hall, D.L., Llinas, J.: An introduction to multisensor data fusion. Proc. IEEE **85**(1), 6–23 (1997)
10. Haller, N.: The S/KEY One-Time Password System (1995). RFC 1760
11. He, H., et al.: The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing other computational intelligence. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 1015–1021, July 2016
12. Kam, M., Zhu, X., Kalata, P.: Sensor fusion for mobile robot navigation. Proc. IEEE **85**(1), 108–119 (1997)

13. Kim, J.H., Keller, B., Lattimer, B.Y.: Sensor fusion based seek-and-find fire algorithm for intelligent firefighting robot. In: 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1482–1486, July 2013
14. Lamport, L.: Password authentication with insecure communication. Commun. ACM **24**(11), 770–772 (1981)
15. Line, M.B., Nordland, O., Røstad, L., Tøndel, I.A.: Safety vs security? In: Proceedings of 8th International Conference on Probabilistic Safety Assessment and Management (PSAM 2006), New Orleans, USA (2006)
16. Lynen, S., Achtelik, M.W., Weiss, S., Chli, M., Siegwart, R.: A robust and modular multi-sensor fusion approach applied to MAV navigation. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3923–3929, November 2013
17. Macher, G., Höller, A., Sporer, H., Armengaud, E., Kreiner, C.: A combined safety-hazards and security-threat analysis method for automotive systems. In: Koornneef, F., van Gulijk, C. (eds.) SAFECOMP 2015. LNCS, vol. 9338, pp. 237–250. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24249-1_21
18. Mirzamohammadi, S., Chen, J.A., Sani, A.A., Mehrotra, S., Tsudik, G.: Ditio: trustworthy auditing of sensor activities in mobile & IoT devices. In: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, p. 14. ACM (2017)
19. M'Raihi, D., Bellare, M., Hoornaert, F., Naccache, D., Ranen, O.: HOTP: an HMAC-based one-time password algorithm. Technical report (2005)
20. Myagmar, S., Lee, A.J., Yurcik, W.: Threat modeling as a basis for security requirements. In: Symposium on Requirements Engineering for Information Security (SREIS), vol. 2005, pp. 1–8. Citeseer (2005)
21. Niemueller, T., Ewert, D., Reuter, S., Ferrein, A., Jeschke, S., Lakemeyer, G.: RoboCup logistics league sponsored by festo: a competitive factory automation testbed. In: Jeschke, S., Isenhardt, I., Hees, F., Henning, K. (eds.) Automation, Communication and Cybernetics in Science and Engineering 2015/2016, pp. 605–618. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42620-4_45
22. Pfitzner, C., et al.: 3D Multi-sensor data fusion for object localization in industrial applications. In: 41st International Symposium on Robotics, ISR/Robotik 2014, pp. 1–6, June 2014
23. Popovici, A., Frei, A., Alonso, G.: A proactive middleware platform for mobile computing. In: Endler, M., Schmidt, D. (eds.) Middleware 2003. LNCS, vol. 2672, pp. 455–473. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-44892-6_23
24. Rezvani, M., Ignjatovic, A., Bertino, E., Jha, S.: Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks. IEEE Trans. Dependable Secure Comput. **12**(1), 98–110 (2015)
25. Sun, W., Yu, S., Lou, W., Hou, Y.T., Li, H.: Protecting your right: attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. In: 2014 Proceedings IEEE, INFOCOM, pp. 226–234. IEEE (2014)
26. Yi, X., Bouguettaya, A., Georgakopoulos, D., Song, A., Willemson, J.: Privacy protection for wireless medical sensor data. IEEE Trans. Dependable Secure Comput. **13**(3), 369–380 (2016)
27. Zuehlke, D.: SmartFactory-towards a factory-of-things. Ann. Rev. Control **34**(1), 129–138 (2010)