applied sciences

MDPI

Article

Secure and Privacy-Respecting Documentation for Interactive Manufacturing and Quality Assurance

Paul Georg Wagner [1,*], Christian Lengenfelder [2], Gerrit Holzbach [2], Maximilian Becker [1], Pascal Birnstill [2], Michael Voit [2], Ali Bejhad [1], Tim Samorei [1] and Jürgen Beyerer [2]

[1] Vision and Fusion Laboratory, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany; maximilian.becker@iosb.fraunhofer.de (M.B.); ali.bejhad@iosb.fraunhofer.de (A.B.); tim.samorei@iosb.fraunhofer.de (T.S.)

[2] Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, 76131 Karlsruhe, Germany; christian.lengenfelder@iosb.fraunhofer.de (C.L.); gerrit.holzbach@iosb.fraunhofer.de (G.H.); pascal.birnstill@iosb.fraunhofer.de (P.B.); michael.voit@iosb.fraunhofer.de (M.V.); juergen.beyerer@iosb.fraunhofer.de (J.B.)

* Correspondence: paul.wagner@kit.edu

Abstract: The automated documentation of work steps is a requirement of many modern manufacturing processes. Especially when it comes to important procedures such as safety critical screw connections or weld seams, the correct and complete execution of certain manufacturing steps needs to be properly supervised, e.g., by capturing video snippets of the worker to be checked in hindsight. Without proper technical and organizational safeguards, such documentation data carries the potential for covert performance monitoring to the disadvantage of employees. Naïve documentation architectures interfere with data protection requirements, and thus cannot expect acceptance of employees. In this paper we outline use cases for automated documentation and describe an exemplary system architecture of a workflow recognition and documentation system. We derive privacy protection goals that we address with a suitable security architecture based on hybrid encryption, secret-sharing among multiple parties and remote attestation of the system to prevent manipulation. We finally contribute an outlook towards problems and possible solutions with regards to information that can leak through accessible metadata and with regard to more modular system architectures, where more sophisticated remote attestation approaches are needed to ensure the integrity of distributed components.

Keywords: automated video documentation; workflow recognition; privacy-respecting manufacturing technologies; manufacturing security; human-computer interaction

Citation: Wagner, P.G.; Lengenfelder, C.; Holzbach, G.; Becker, M.; Birnstill, P.; Voit, M.; Bejhad, A.; Samorei, T.; Beyerer, J. Secure and Privacy-Respecting Documentation for Interactive Manufacturing and Quality Assurance. *Appl. Sci.* **2021**, *11*, 7339. https://doi.org/10.3390/app11167339

Academic Editor: José Machado

Received: 2 July 2021
Accepted: 7 August 2021
Published: 10 August 2021

1. Introduction

In recent years, integrating smart and intelligent manufacturing processes into production systems has become an increasingly important challenge. A multitude of research has been conducted regarding interdisciplinary and human-centric approaches to achieve this goal [1–5]. One realization emerging in the wake of this effort is that many different technologies and methodologies, e.g., human-machine interaction, networking, artificial intelligence (AI), smart sensing and data acquisition, have to be considered. Smart production facilities of the future will consist of networked, AI-assisted manufacturing environments in which humans will act and cooperate with the available assistance systems [6]. Increasing AI capabilities will make human work more flexible and more proactively supported by intelligent machines. A multitude of different sensors will monitor production processes for quality assurance reasons, among others, and provide a comprehensive digital twin of all production processes. In order to take human activities into account both for human-machine interaction and for proactive assistance and process mapping, production environments of the future need to be able to perceive humans and understand their actions.

Possible applications for assistance systems of this kind can be found wherever humans are the central execution agency of the production process. Despite increasing automation, assembly activities and quality assurance are still carried out by humans in many companies and in many areas of production even today. In particular, it can be observed that the human work process is adapted to the conditions of the surrounding automation. Instead of attaching importance to an intuitive procedure that is familiar to the human being, the worker is forced, for example, to check components for defects in short cycle times and then to document them on a PC or sometimes on a portable tablet [7]. As a result, the transfer of detected defects is forgotten or happens inaccurately. Error slippage leads to poor quality or high rework costs. Simple gesture recognition systems already make it possible to acquire documentation directly on the component by indicating and classifying detected defects by pointing with the finger [8]. As a result, additional documentation efforts are eliminated, error slips are minimized by direct annotation on the component, and defect positions can be precisely referenced on the digital twin of the component [3]. Besides error documentation, smart assistance functions have been examined in the context of numerous other applications as well. Relevant examples include advanced inspection and quality assurance systems [9–11], effective induction and training methods [12–14], as well as assistance systems that aim to establish inclusive [15], non-discriminatory [16] and immersive [17] work environments. In all of these cases, the detection of the worker and associated innovative interaction modalities offer the opportunity to develop assistance functions that optimally support human performance due to intuitive operability. Gesture recognition in combination with speech offers a form of communication that people use subconsciously and intuitively in everyday life. These techniques can be used advantageously in many applications of human-machine communication, especially in production.

Beyond innovations in human-machine interaction, video-based capturing of workers offers previously unconsidered possibilities for making production more flexible, safer and more reliable. It allows assistance systems to perceive and interpret the activities of the worker along with their context of action. Assembly activities no longer have to be explicitly confirmed, e.g., by pressing a button, if the respective work steps can be perceived and understood by the smart assembly environment [2]. In addition, individual reactions to people and functions that guide the worker step by step through the assembly process can be largely minimized if it is recognized that the assembly result is correct regardless of the individual procedure [18]. Without disturbing the worker in the execution of his activities, reached milestones can be proactively documented in the background with picture evidence, e.g., the bolting of safety-critical points, the quality-assuring inspection of welding spots or, in the simplest case, the correct insertion of the first-aid kit in the production of vehicles [19]. Considering the perceptual capabilities that a production environment gains through video-based acquisition modalities, a myriad of applications can be developed into truly smart production processes.

However, video-based assistance and documentation systems also have clear downsides that have not been addressed sufficiently in the past. While video-based solutions offer a wide variety of different capture options with only one sensor, they are often criticized for capturing personal information in the sensor data, which leads to privacy and data security issues. Particularly with regards to widespread video-based surveillance systems, such as dashcams operated by ride-hailing companies, data privacy concerns regarding captured video footage have been a topic of public discourse for a while [20]. Recently, the Chinese ride-hailing enterprise Didi Global has been suspended on grounds of data security and privacy issues, further emphasizing the need for privacy-friendly data acquisition solutions [21]. While privacy-preserving dashcams have already been researched [22], to our knowledge there is no proposal for mitigating data privacy risks of video-based assistance systems in production environments.

A recent survey, conducted among leaders in industry and academia, shows that both privacy protection and information security are considered to be prime challenges in the

realization of smart manufacturing environments [5]. Hence, when advanced assistance and documentation systems acquire large amounts of data, video material in particular, it is crucial to embed the respective systems in the company processes in a manner that complies with data protection requirements and involves employees or employee representatives. It is particularly important to ensure that the data is only processed for legitimate purposes, such as simplifying work processes, while at the same time ensuring that technical and organizational measures are in place to prevent any misuse. The surreptitious monitoring of employees' work performance and access to video data without the consent of the recorded data subjects has to be prevented.

In this paper, we describe a technical system architecture for a smart video documentation system (Section 2), for which we systematically derive data protection and data security requirements (Section 3). As our main contribution, in Section 4, we introduce an extended system architecture that includes advanced security mechanisms addressing these requirements and offering a high level of privacy protection for workers. We analyze the security of our approach and discuss its limitations in Section 5 before describing our proof-of-concept scenario and comparing our system with previous work in Section 6. Finally, we conclude with an outlook on possible solutions for more flexible and secure architectures in Section 7.

2. Basic System Architecture

Since the challenge of adequate data privacy protection in smart manufacturing systems is still largely unsolved, our goal in this work is to develop mechanisms for protecting the privacy and informational self-determination of workers subjected to these types of assistance systems. More concretely, we base our contribution on an existing collaborative manual assembly station, which is fitted with a multitude of sensors to monitor, supervise and individually assist the worker [2]. We extend this system with suitable privacy-protection components and integrate our solution into the physical assembly station as a proof-of-concept implementation. Nevertheless, our findings are generally applicable for any assistance systems that capture and store video footage of workers. In this section, we briefly describe the existing assembly station and extract a basic system design for it. Afterwards we describe how the assembly station can be extended with components to automatically detect and document critical work steps as video footage. Based on this abstract system design, we later derive data protection and data security requirements (Section 3) and develop advanced security mechanisms (Section 4).

Figure 1 shows the assembly station [2] we use as a development platform to implement our privacy-protection system. The assembly station is fitted with a set of stereo cameras for tool tracking purposes and a depth-camera for detecting the worker. Furthermore, it features a projector that can present information and instructions about the current work step directly on the work piece. When the worker begins his task, the assembly station captures a live video stream from the depth camera, which provides both a video and a 3D point cloud. The assistance system uses the image data stream to automatically detect the currently executed work step and compares it to a domain-specific workflow model. Then the worker can be assisted by projecting information for this work step directly on the work piece. This manual assembly scenario is convenient for our purpose due to its relative simplicity and self-containment. Since all components shown in Figure 1 operate locally on a single computer, no network connections must be considered. From a data protection perspective, the aim is to implement as much system functionality as possible without having to store personal data. At the same time, there is a need to collect data to a certain extent for quality assurance and optimization purposes. Our goal is to extend this assistance system with the secure storage and retrieval of critical work data, without impeding the informational self-determination of assisted workers.

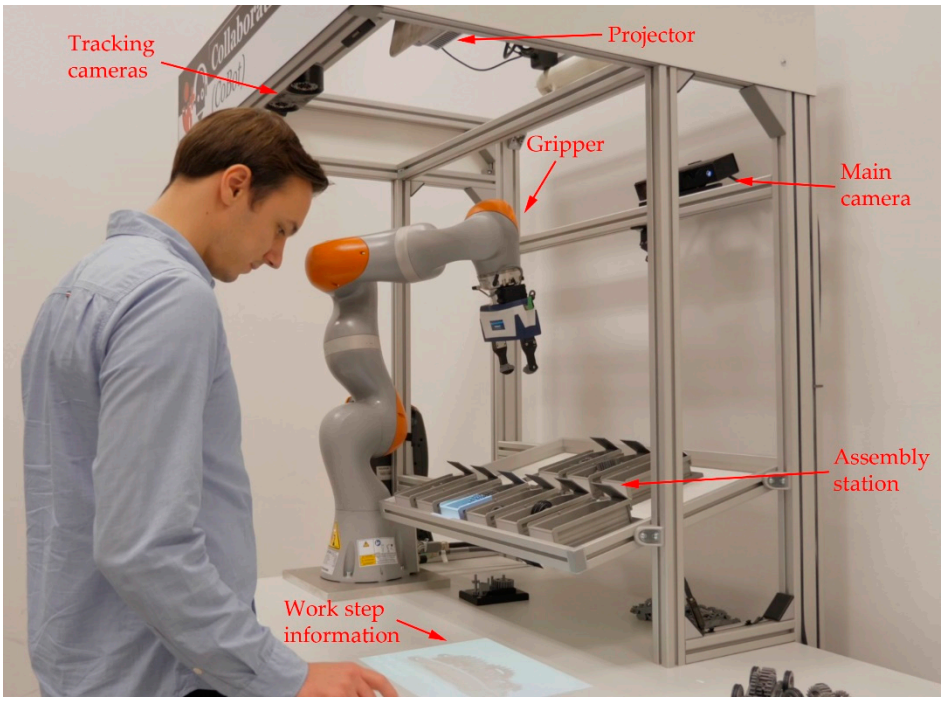

Figure 1. A collaborative manual assembly station.

As a first step, we extend the assembly station shown in Figure 1 with the capability of retaining videos of critical work procedures for documentation purposes. This can be done by integrating a video storage component with the existing workflow detection. If a critical work step is detected, a personalized workflow log and a personalized video snippet are generated and persisted on the hard drive. Due to possible delays caused by the detection component, a ring buffer is introduced to ensure that the first few seconds of such work steps are not missed. Furthermore, the captured documentation should be linked to the user's identity for later review. For this, we mount a smartcard reader near the assembly station, which the worker uses to authenticate and log in at the system before beginning his task. To be able to view the recorded data in case of an inspection, e.g., due to a problem with one of the workpieces, the system must also offer the possibility to retrieve and display the recorded work steps and workflow logs. For simplicity purposes, this is done by placing a monitor next to the assembly station that shows the recorded videos to supervisors.

An overview of the resulting system architecture including the described modifications is given in Figure 2. The documentation system consists of three components. The *workflow recognition* extracts the 3D body pose of the current worker from the video feed and the 3D point cloud of the depth camera. The worker's current activity is estimated using the 3D body pose. With constant observation and a graph-based model of the assembly process, this component is able to tell at any time which step of the process the worker is currently doing and what the next steps need to be. It recognizes if the worker diverges from the optimal path, and it can recognize points of interaction with the workpiece. The component is able to differentiate critical steps that need to be documented from normal steps using the assembly model. When recognizing a critical step, the video and critical data of that assembly step is sent to the storage. The *documentation & video storage* consists of a file storage for the raw video data of the assembly steps and a database for the recordings and corresponding metadata. The database associates the assembled part with the video data and the worker. The *authentication* component tells the documentation storage which user is logged into the system so that it can associate the current recordings to the currently active worker as well as the unique workpiece ID. Workers can authenticate by holding their personal smartcard near a reader mounted next to the assembly station. The authentication

is performed using certificates retrieved from an external user directory service accessed over the industry standard Lightweight Directory Access Protocol (LDAP). Besides classical smartcards, a smartphone can also serve as an authentication token, if it supports near-field communication (NFC). Using a smartphone for this purpose gives workers access to more advanced security features offered by the documentation system, which we will cover in the following sections.

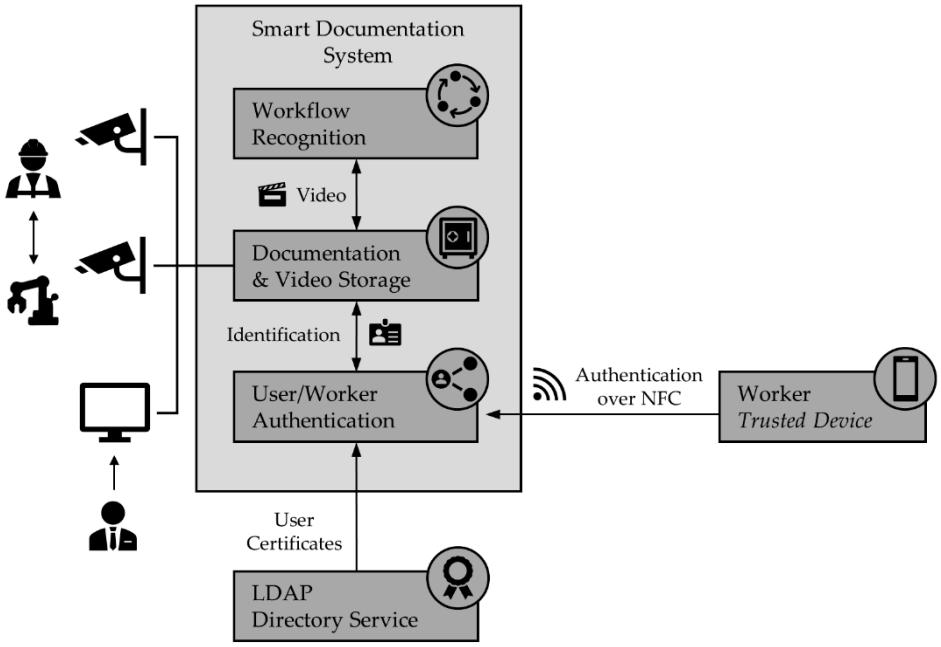

Figure 2. Basic system design.

3. Security and Privacy Model

While the system architecture described above offers automated documentation and workflow recognition capabilities for smart manufacturing processes, it does not yet deal with privacy aspects of the recorded video documentation. Workers that are subjected to automated video documentation have the right of privacy and must not be disproportionately disadvantaged, e.g., by using the recorded videos for covert performance evaluations. In this section we introduce the necessary protection goals, the attacker model and the trust model, based on which we will subsequently build a secure and privacy-respecting system architecture. Our methodology is as follows. We formulate protection goals relative to assets, e.g., data categories or system components. Based on these we subsequently categorize attackers by specifying which protection goals they intend to break. Finally, in the trust model we identify relevant actors regarding the system, their capabilities and describe as which attackers they can potentially appear. This modeling helps us, on the one hand to make decisions for security mechanisms and, on the other hand, to analyze them afterwards.

3.1. Protection Goals

As indicated earlier, the primary protection goal we are concerned with is the *confidentiality of documentation data* in the interest of the data subject, be it video snippets or workflow protocols. Technical means are supposed to protect the data subject against covert performance monitoring based on documentation data. This requires that access to documentation data must be granted by the data subject. We specify this protection goal even further by requiring that at least one other trusted party besides the data subject must consent to the access to the documentation data so that the data subject cannot be pressurized to grant access.

The data subject as well as the operating organization have an interest in the *integrity and authenticity of documentation data*. Both parties are interested in ensuring that documentation data cannot be manipulated unnoticed. More specifically, this means that no documentation data is changed, deleted or added unnoticed outside the system behavior given by the system specification.

To meet these protection goals, we must go even further and demand, as derived protection goals, that the *integrity and authenticity of the smart documentation system* is ensured and that the *authenticity of parties accessing documentation* data is verified. The former protection goal ensures that the smart documentation system cannot be manipulated or even replaced unnoticed.

We do not cover protection goals on metadata, which are used to retrieve documentation data and to request authentication of the parties before access, since the approach presented in this paper is not capable of restricting access to metadata and preventing potential leaks of sensitive information that could be derived from metadata. However, we discuss this question as well as envisioned countermeasures in Section 5.

3.2. Attacker Model

We now characterize abstract attackers by what protection goals they try to break. We consider a privacy attacker, a modifying attacker and a destructive attacker as a subtype of the modifying attacker.

The *privacy attacker* aims to break the confidentiality of documentation data. This attacker can be represented by the operating organization or employer and attempts to access documentation data in violation of agreements made in order to gain insight into the performance of its employees. As described in Section 3.1, this attacker can also achieve his goal by manipulating or replacing the smart documentation system with a seemingly identical but insecure system.

The *modifying attacker* tries to break the protection goal of the integrity and authenticity of the documentation data so to fabricate evidence. He can be represented by the data subject trying to disguise errors that occurred during the execution of his work and that have been captured by the smart documentation system. Alternatively, he could try to remove the link between an error and himself. The modifying attacker can also be represented by the operating organization or employer acting with the motivation of foisting an error on an employee. A special case of the modifying attacker is a *destructive attacker*, who wants to delete a data record in order to destroy evidence. Both the operator and the worker could pursue this interest.

We thus consider only internal attackers within an organization. We exclude external attackers by assuming that the smart documentation system is operated within an access-restricted building and by assuming that the attackers considered subsume the attack vectors and possibilities of external attackers.

3.3. Trust Model

The trust model describes the actors of interest in terms of operation, use and the protection goals introduced in Section 3.1. We discuss with respect to what protection goals an actor is trusted or untrusted, i.e., may act as an attacker in the latter case.

The first actor to be considered is the *worker* as the data subject. Data associated to this actor is processed by the smart documentation system. The worker is interested in the protection goals of confidentiality as well as integrity and authenticity of documentation data. Nevertheless, the worker is not fully trusted as he may have an interest in acting as a modifying attacker, e.g., to disguise evidence of errors by modification or deletion of documentation data.

The *system supplier* creates the smart documentation system and puts it into operation. We accept him as fully trustworthy. This means that he has no interest in breaking protection goals, nor does he support any other actor in breaking protection goals. We cannot avoid this assumption because the system supplier is also responsible for the integration and

configuration of security features. The system supplier has administrative rights on the system, which he uses after commissioning for maintenance purposes only. However, we neglect the case of system maintenance in this work as we consider the system supplier as fully trusted.

The operating organization is the main antagonist of the primary protection goal of confidentiality of documentation data. We represent the operating organization through its *system administrator* who has administrative rights on the documentation system. We concede that he can become a privacy attacker at the direction of the organization's management, attempting to break the confidentiality of the documentation data by means of his privileges and capabilities. The system administrator is also not fully trusted with respect to the integrity and authenticity of the documentation data, which he may attempt to compromise at the direction of management, i.e., he may also become a modifying attacker. Both as a privacy attacker and as a modifying attacker, the system administrator may also attempt to target the derived protection goals of integrity and authenticity of the smart documentation system to achieve his attacker interests.

Finally, we introduce a trusted party within the operating organization, i.e., an actor whom the worker can trust sufficiently to protect his privacy interests. This party is authorized to agree with the operating organization under which technical and organizational conditions the smart documentation system can be implemented in the organization. We introduce this party as a *trustworthy subject* representing, for example, the works council of the operating organization. The trustworthy subject takes an examining stance towards the smart documentation system and the corresponding organizational processes, which we need for the system design introduced in the following section.

4. Privacy-Preserving System Design

In order to achieve the identified protection goals under the previously presented attack and trust model, we develop and implement *4Crypt*—A privacy-preserving video documentation system for smart manufacturing applications. Our system allows the continuous monitoring and logging of critical manufacturing steps using off-the-shelf video equipment, while still protecting the privacy interests of recorded workers. For this, 4Crypt automatically encrypts all captured video footage and securely distributes the encryption keys to a set of trusted individuals using a secret-sharing protocol. Decrypting the videos is only possible with the consent of both the recorded worker and at least one of the employee's trusted representatives. This application of the 4-eyes principle prevents the employer from pressuring workers into yielding keys required to decrypt videos without legitimate cause. As a result, the 4Crypt system protects the recorded worker from illegitimate surveillance and covert performance evaluation, while still allowing the employer to reap the benefits of an automated video documentation and workflow recognition. Furthermore, 4Crypt provides strong cryptographic security guarantees against malicious interfering and tampering with the privacy-protection modules. Workers can verify the integrity of the 4Crypt system by using their personal smartphone on a dedicated NFC-reader, which prevents the system operator from disabling the encryption layer or extracting any encryption keys. This is achieved by adding a trusted platform module (TPM) to each system and by implementing a remote attestation protocol over NFC.

With 4Crypt we extend the basic version of our smart video documentation system presented in Section 2 by a number of privacy-protection components, most importantly an encryption/decryption module as well as a TPM-based attestation service. The resulting system design is presented in Figure 3 with new components shown in yellow.

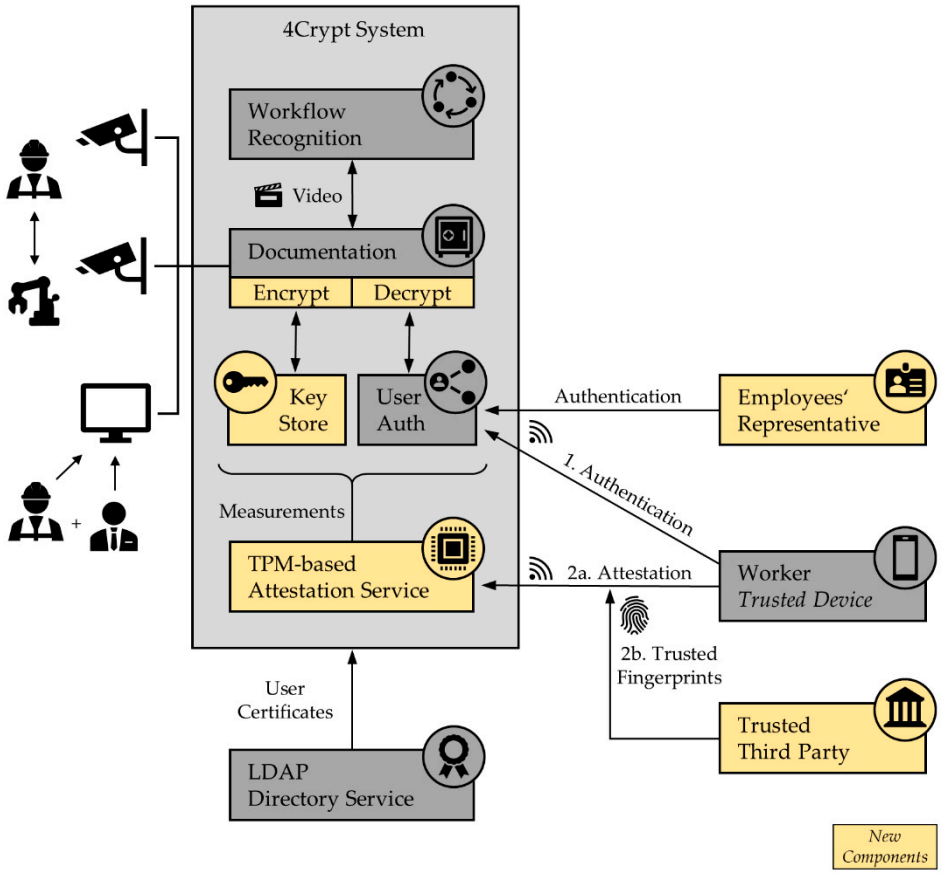

Figure 3. 4Crypt system design.

The main idea of 4Crypt is to add a transparent encryption layer into the video processing pipeline, which implements a hybrid encryption scheme that restricts the employer's access to captured video files. In the following sections we describe the three main tasks of 4Crypt and how they were solved in greater detail. More concretely, 4Crypt needs to:

1. Record, encrypt and securely store videos in a way that even the system operator cannot access them.
2. Retrieve a stored video with consent from both the recorded worker and at least one of the employee's trusted representatives.
3. Offer a way of verifying the integrity of the protection components before a worker begins with a monitored work step.

4.1. Video Encryption

Whenever a worker executes a critical work step in the manufacturing process, a new video should be recorded for documentation purposes. This video needs to be encrypted in a way that it is accessible only for the recorded worker in cooperation with at least one of the trusted representatives. In order to achieve this requirement, we developed a multi-level encryption scheme that includes a suitable secret-sharing mechanism for the encryption keys (c.f. Figure 4).

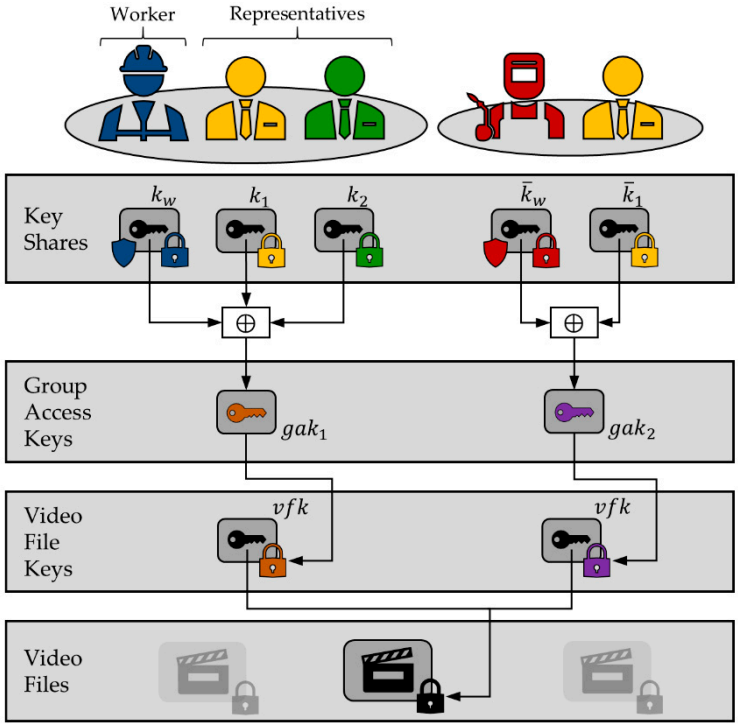

Figure 4. Encryption and key sharing architecture of 4Crypt.

Secret-sharing schemes as described in [23] are well-researched methods to distribute a specific secret to a set of participants without disclosing the secret to any one party. Participants only receive a so-called *share* of the secret, which in itself does not contain any information. Only when a certain number of participants disclose their respective share, the underlying secret can be recovered.

The 4Crypt system implements an encryption scheme that uses a secret-sharing algorithm to distribute encryption keys to a number of trusted individuals. The details of this key-sharing mechanism are shown in Figure 4, with key usages displayed as arrows. 4Crypt automatically encrypts recorded videos with a randomly drawn, symmetric video file key (VFK). In Figure 4, this is displayed in the bottom two levels for an exemplary video file. The video file keys are not known to either the recorded worker, the trusted representatives, or the employer. Instead, they remain stored on the documentation system and are themselves encrypted with one or more group access keys (GAK). A video-specific group access key is created by randomly drawing one symmetric key share k_i for each group member (either a worker or a trusted representative) and then performing a bitwise XOR-operation on the shares. In Figure 4, this is shown in the top two levels, with the XOR-operation denoted by the symbol $\oplus$. As a result, the group access keys can only be recovered by collecting the key shares of all participants in the respective group. Finally, the key shares are asymmetrically encrypted using the public keys of the respective group members and stored on the documentation system. This encryption scheme allows 4Crypt to securely store videos in a way that they can only be recovered if all members of at least one group consent by disclosing their video-specific key shares.

The resulting video acquisition and encryption process based on this key sharing architecture is shown in Figure 5. At first, the worker w authenticates and logs in by holding his personal smartphone on the NFC-reader near the assembly station. Once the critical work step is about to be performed, 4Crypt initiates the recording process. The video documentation system then creates a new symmetric encryption key vfk for the video file. Furthermore, a new group access key gak has to be created for each configured group consisting of the authenticated worker w and a set of n trusted representatives. For this, the documentation system generates a set of random key shares k_j for each member of the group and calculates the group access key gak by XORing them together. The key share

k_w of the worker is sent back to the worker's smartphone over NFC to asymmetrically sign and encrypt it with the worker's private/public keys. The key shares of the other group members are encrypted with the respective public key retrieved from the external LDAP service. All encrypted/signed key shares are then stored in the local key store. Finally, the video file key is encrypted with each group access key and deposited in the key store as well. Once the recording has been initialized in that way, each video image captured by the cameras mounted at the assembly station is automatically encrypted with the video file key.

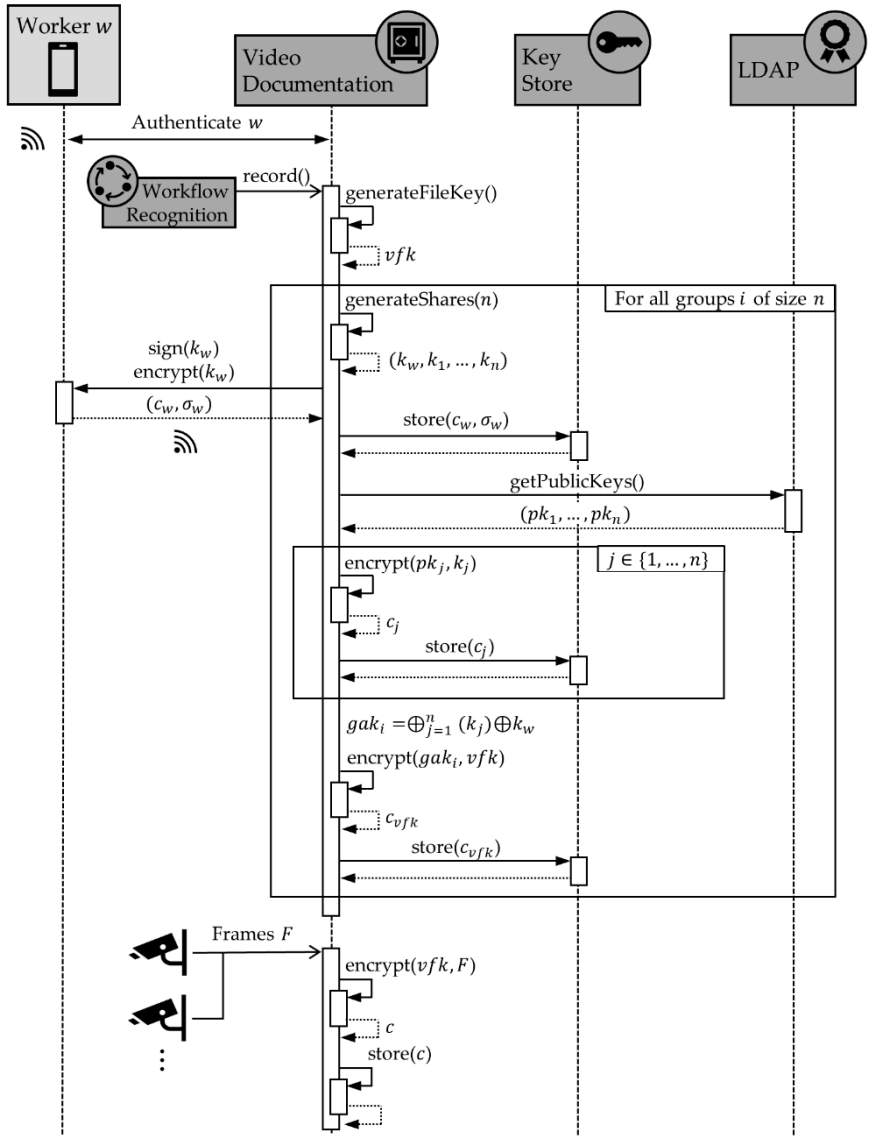

Figure 5. 4Crypt video encryption process.

4.2. Video Retrieval

In order to retrieve and display a previously recorded video, it has to be decrypted on the documentation system. For this, all members of at least one group must consent to the decryption of the video by disclosing their respective key share. Figure 6 shows the process of a video decryption in 4Crypt.

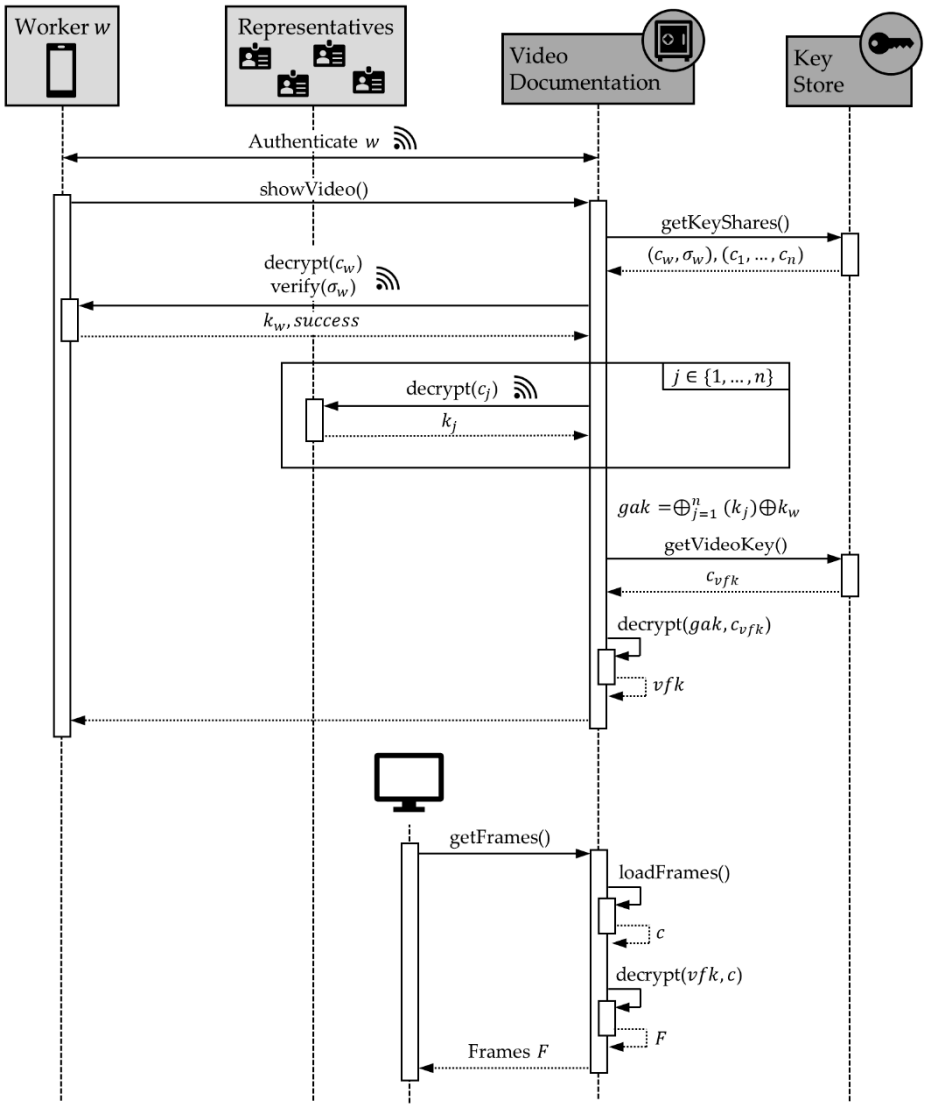

Figure 6. 4Crypt video decryption process.

At first, the worker that has been recorded authenticates by holding his smartphone on the NFC-reader and selecting the video that should be displayed. This retrieves the video-specific, encrypted key shares of all necessary group members from the key store. To prevent a modifying attacker from foisting a faked video on the worker (c.f. Section 5), his key share k_w is decrypted and its signature σ_w is verified on the smartphone. Similarly, each member of the selected group (usually a set of trusted representatives) must decrypt their respective key share k_j by holding their smartcards on the NFC-reader as well. Once all group members including the recorded worker have consented to the video disclosure in that way, the group access key gak can be restored by XORing all decrypted key shares. Then the video file key vfk is retrieved from the key store and decrypted using the group access key. Finally, the video frames are decrypted using the video file key and shown on the connected display.

4.3. Integrity Verification

In the previous sections we have shown how 4Crypt encrypts video files and implements a secret-sharing algorithm on the cryptographic keys to enforce consent of both the worker and a set of trusted representatives before videos are disclosed. This solution prevents the employer from decrypting any videos by himself in order to use them for illegitimate purposes such as the covert performance evaluation of workers. However,

since the employer operates 4Crypt alongside the video surveillance system, he has comprehensive physical access to the 4Crypt hardware and software. Hence, the employer could simply turn off the privacy-protection modules and bypass the encryption altogether by proactively tampering with the 4Crypt system components directly on the machine. Since the goal of 4Crypt is to protect the privacy interests of recorded workers against a malicious employer abusing the video documentation system, why should the workers trust the employer with setting up 4Crypt correctly in the first place? Due to this consideration, it is necessary to give the workers the opportunity to verify the integrity of the 4Crypt privacy-protection modules before being subjected to the video surveillance. That way the workers would be able to detect malicious tampering with the 4Crypt system or confirm that their privacy is in fact being protected before starting a supervised work step.

In 4Crypt, we enable workers to confirm the integrity of the documentation system by adding *trusted platform modules* (TPMs) to the system design. A TPM is a dedicated hardware chip that extends a computer with basic security related features. TPMs operate as trusted hardware security modules and are always designed according to a specification developed by the Trusted Computing Group (TCG) [24]. Each TPM holds several cryptographic keys that can be used to encrypt data, identify the computer system and attest to its current software stack. For this, the TPM contains volatile *platform configuration registers* (PCRs) that can be used to measure the current hardware and software configuration as an unforgeable fingerprint. During a trusted boot sequence each boot stage hashes the software stack of the next stage and writes this measurement into the PCRs. By extending these measurements all the way up to the user applications and their configuration, the fingerprints accumulated in the PCRs uniquely represent the current state of the software stack that is running on the measured system.

Building on this measurement process, an external third party can remotely verify that the measured system is running a certain software stack in a certain state. This process is called *remote attestation*. During a remote attestation, the external verifier requests the log of the conducted measurements along with a so-called *quote*, which contains the fingerprints collected during the measured boot sequence and serves as proof of the current system state. In order to prevent the challenged system from lying about the fingerprints stored in its PCR registers, the remote verifier checks that the provided quote is signed by the TPM with a valid *attestation identity key* (AIK). The AIK is an asymmetric cryptographic key pair that has been created by the TPM during a prior enrollment phase. While the public part of the AIK is known to all involved participants (usually it is certified by a trusted party), the private key never leaves the TPM. Hence if the quote signature is correct, the verifier is convinced that the system under test did not lie about its fingerprints in the measurement log. Then the verifier evaluates the measurement log and confirms that the measured fingerprints of the applications running on the remote system match the expected fingerprints of a correctly configured and unmodified software stack.

By utilizing this TPM-based remote attestation technique in 4Crypt, we give workers the opportunity to verify if the documentation system is trustworthy before stepping in front of the video cameras. For this we add a TPM to the 4Crypt machine and conduct a measured boot sequence that includes measurements of the 4Crypt software and its configuration. The measurements are performed on a Linux machine with TrustedGRUB [25] and the Integrity Measurement Architecture (IMA) developed by Sailer et al. [26], which since has been integrated in the Linux kernel. As a result, each 4Crypt system can prove that it has not been maliciously modified by presenting its measurement log and a corresponding quote. These measurements can then be compared to valid fingerprints of an unmodified 4Crypt software stack that have been deposited at a *trusted third party* (TTP) during the deployment of the documentation system. To facilitate the verification process of 4Crypt systems, we designed and implemented a remote attestation protocol that can be executed over NFC. This allows workers to install an Android application on their personal smartphones and use it to attest the integrity of any 4Crypt system by simply holding their phone near the provided NFC-reader. We execute our attestation protocol

in conjunction with the smartphone-based authentication of the worker that takes place before each recorded work step. That way the remote attestation seamlessly integrates into the normal operation of the video documentation system. Figure 7 shows how our NFC-capable remote attestation protocol works on a high level. We implemented this protocol in accordance with the Trusted Attestation Protocol (TAP) that has recently been standardized by the Trusted Computing Group [27]. For the attestation service on the documentation system, we use JavaCard and Google's gRPC framework [28]. For the Android application we implemented a custom-built NFC service handler.

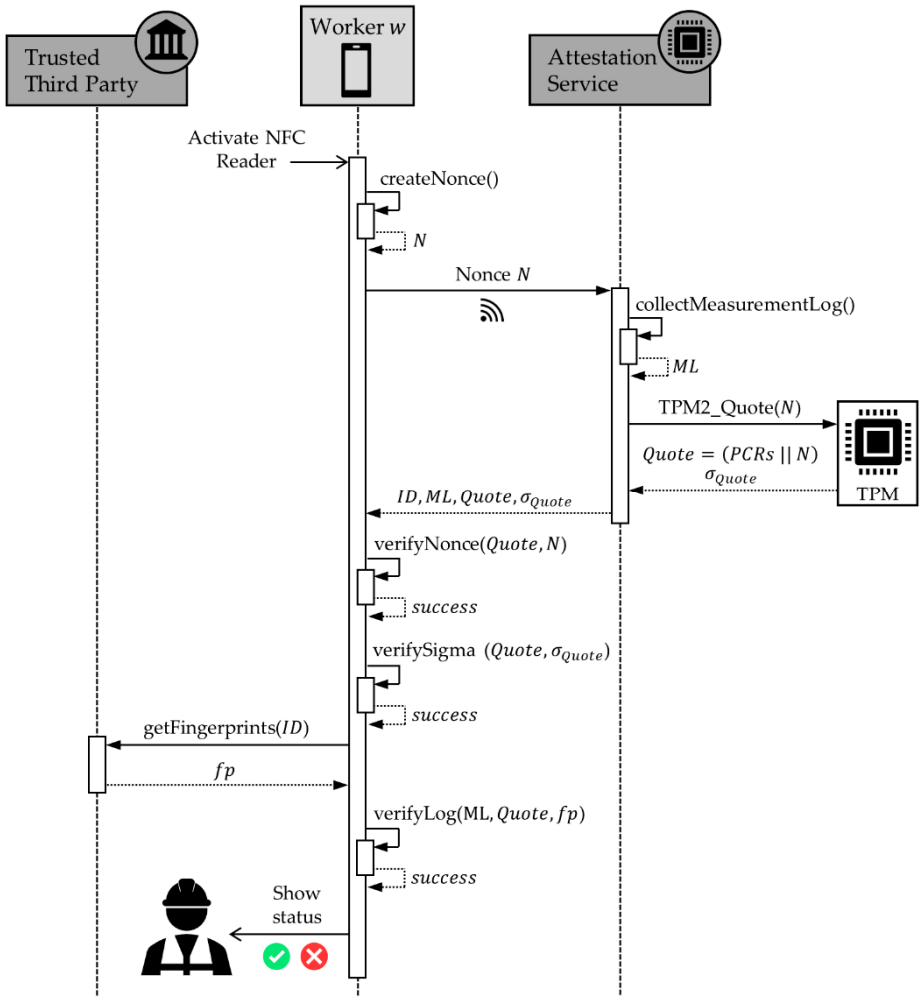

Figure 7. NFC-capable remote attestation protocol integrated into 4Crypt.

Whenever a worker holds his smartphone near the NFC-reader at the assembly station, a fresh nonce is randomly drawn and transmitted over NFC to the attestation service running on the 4Crypt system. The attestation service then uses the TPM to create a signed quote that includes the current fingerprints stored in the PCRs as well as the received nonce. The quote and its signature are then transmitted back to the smartphone and verified. This verification process consists of three steps. First, the verifier checks if the previously selected nonce is in fact contained in the quote. This prevents the prover from reusing old quotes in a replay attack. Then the verifier checks the validity of the quote signature under the known AIK public key. If this is successful as well, the verifier finally must evaluate whether the attested fingerprints match the unmodified 4Crypt system. For this the verifier retrieves a set of valid fingerprints from the trusted third party and compares them to the fingerprints contained in the quote. If all verification steps are successful, the Android application shows that the worker can trust the video documentation system. If instead the

attestation process reveals that the documentation system has been modified, the Android application stores the signed quote as proof of the illegitimate system modification. Then the worker can safely refuse to operate under video surveillance without having to fear negative repercussions by the employer.

5. Security Analysis

In this section, we analyze to what extent the system architecture presented in the previous section meets the protection goals introduced in Section 3, given the considered attackers and the assumed trust relationships.

5.1. Assumptions

In order to focus the security analysis on the extended system architecture, we first need to introduce some additional assumptions. We assume that all hardware and software components, especially including the TPM 2.0 and the operating system (ROS2), are trusted. For the software components, this means that they are implemented correctly, are free of bugs and are configured correctly once they are put into operation by the system supplier. Video streams between the sensors/cameras and the smart documentation system are transmitted over an encrypted channel, which we assume to be secure. We also assume that current state-of-the-art cryptographic methods from established libraries are used and that they can be considered cryptographically secure. Regarding the deployment of the smart documentation system, we assume that it is integrated into a collaborative manual assembly station as described in Section 2. We also assume that valid system states accepted by the integrity verification mechanism presented in Section 4.3 do not contain any kind of remote access, such as SSH. The last two assumptions mean that system administrators must also log in to the documentation system locally, so that they would be seen by a worker who is currently using the system. Regarding attestation, we assume that the trusted system supplier maintains valid system states of the 4Crypt system at the TTP, which makes them available for the attestation service. The mobile applications that the worker uses for attestation and authentication on his private device are also provided by the system supplier and assumed to be trusted. Wherever password authentication of users/administrators is possible, we assume that strong passwords are used. We exclude social engineering from the analysis, which means that passwords and PINs from smartphones and smartcards must be considered unattainable to attackers.

5.2. Attack Vectors

We first introduce the attack vectors of the attackers. The privacy attacker wants to get hold of documentation data in plain text. Since the encryption is assumed to be secure, the points in time when the data is available in unencrypted form and how it could be accessed in these situations have to be considered. The first occurrence of documentation data on the system in plain text is when it is temporarily stored on disk prior to encryption. The privacy attacker would need to gain access to the system with the appropriate permissions and read the data before the plaintext is deleted immediately after encryption. Given the throughput of AES on current CPU, the duration of the time span to at least read out parts of the video prior to encryption is dominated by the duration of the video recording. The second time is when a data item is decrypted, i.e., when accessed by authorized parties. Again, the privacy attacker must gain access to the system while this access is taking place. The privacy attacker may replace users' certificates to impersonate authorized parties during a regular request to access documentation data. Finally, the privacy attacker may also attempt to prevent documentation data from being encrypted at all by replacing the entire system with an apparently identical system.

The modifying attacker aims to create inconsistencies by modifying documentation data. We distinguish between two goals here: to authentically modify an encrypted video and to add an apparently authentic video. To authentically modify an encrypted video, e.g., to shorten it so that an error that occurred is no longer visible, he would have to become a

privacy attacker, i.e., he would have to overcome the privacy attacker's attack vector and the same hurdles. In order to create a seemingly authentic video, for example, by re-using an old video documenting an error to a worker, the modifying attacker must be able to reproduce the encryption and signature steps for the fake video in such a way that the process goes unnoticed. The destructive attacker wants to delete an encrypted video. To do this, he must gain access to the system with the appropriate permissions.

5.3. Countermeasures

We begin our analysis with the privacy attacker's attack vectors. In the first case, tapping a video at the time before it was encrypted, immediately after recording, cannot be prevented completely because the operating system's access control mechanisms cannot protect against a system administrator acting as a privacy attacker. With his privileged permissions, he can copy an unencrypted video to another location before it is deleted and he can also execute programs to automatize attacks. However, this kind of attack is made considerably more difficult by two circumstances. First, it cannot be carried out remotely, because even an attacking administrator does not have remote access, so the attacker would be observed by the worker. Second, the integrity verification mechanism from Section 4.1 ensures that manipulation of the smart documentation system, for example, by starting a program for automatizing an attack, can only take place after the worker has already attested the smart documentation system and started his work. Furthermore, the system would have to be restarted in time to prevent a failed attestation by the next user/worker after an attack. We therefore argue that this kind of attack can neither be automated nor scaled and becomes unattractive for the privacy attacker, who is interested in obtaining multiple videos to gain surreptitious insights into the work performance of employees. The second case is similar for the system administrator attacking on the employer's instructions. He must be on site to manually copy the video just decrypted by the authorized parties to another location. With the limitation to a single video, to which the authorized parties already allow access to anyway, this does not seem to be a particularly attractive profit for the privacy attacker. For the third attack vector, the attacking system administrator would need to replace certificates of workers in the directory service for authentication and in the key store for encryption. The first step is possible, since we do not trust the directory service, but it does not achieve the goal on its own, since it still does not enable video decryption. The second step of this attack is complicated by the fact that the key store is part of the attestation process according to Section 4.3. As in the prior cases, the attacker would have to be on site, act after the attestation has been performed by the worker and reset the system before another attestation by a worker takes place. Otherwise at least one attestation would fail. For the fourth attack vector, the attacker completely replaces the smart documentation system. However, this attack must again fail because the attacking administrator cannot generate a TPM quote for the forged system that is indistinguishable from a valid state of the real system.

The modifying attacker's first attack vector, i.e., to authentically modify an encrypted video, requires a successful privacy attacker's attack which has been discussed previously, and which is even harder to conduct by a worker without a privileged user account. Hence only the second attack vector remains to be analyzed: to create a seemingly authentic video foisting an error on somebody else. We consider the case of the system administrator acting as a modifying attacker. We assume that the attacking administrator has a video sequence in plain text that is indistinguishable from an authentic video. We further assume that he can use functions of the components of the smart documentation system, so that he can easily generate partial keys and encrypt them with any public keys from the key store. However, the attacking administrator cannot perform a crucial step: he cannot forge the signature over a timestamp and the key share of the worker to whom the video is to be foisted because he cannot get hold of the worker's private key, which does not leave the trusted private smartphone. For a worker acting as a modifying attacker, this attack vector

does not exist, since he cannot use the components of the smart documentation system as he wishes and has no access to the personal smartphones of other workers.

Finally, the destructive attacker remains to be considered. The attack to delete an encrypted video is possible for an attacking administrator because the operating system's access control mechanism is not effective against him. However, the absence of the video would be noticed once a search for the workpiece in question did not return any hit. The mechanisms of the operating system are, however, effective against a worker acting as a destructive attacker, so that it is not possible for him to delete a video.

5.4. Limitations

Before summarizing our security analysis given the protection goals, we briefly discuss some limitations of our approach. First, there are the known limitations of TPM-based integrity attestation against internal attackers, particularly with administrative privileges [29]. Conceptionally this only allows us to verify/ensure integrity of a system at the time-of-check, which means that manipulations at the time-of-use cannot be excluded entirely. Possible solutions draw on more advanced trusted computing technologies, i.e., trusted execution environments such as Intel's Software Guard Extensions (SGX). The second aspect to discuss is the monolithic system deployment we assume in our approach. For distributed architectures, i.e., with our components running on different computing resources, the described integrity verification mechanism is not sufficient anymore. If we, for instance, separate the workflow recognition component from the documentation component, the former would have to ensure the trustworthiness of the latter so that, as a consequence, a transitive attestation mechanism according to the trust chain of components would be required. Finally, our approach does not protect against information leaks via metadata. If we can query the smart documentation system for documentation data as we wish and if the system then reveals whose authentication is required to access these records, an attacker might be able to derive some statistical estimates about particular workers' performances (e.g., in terms of the more records per time unit the better).

5.5. Summary/Fulfillment of the Protection Goals

Considering the assumptions given in Section 5.1 and the limitations described above, we consider the protection goals to be met. Revisiting our assumptions, the most problematic in practice are the monolithic deployment with all system components running on one computing resource and the exclusion of remote access for the administrator. The former issue is subject of ongoing work with more advanced trusted computing technologies, which can also alleviate the conceptional limitation of TPMs that leads to the latter issue.

6. Application and Results

The work presented in this paper was prototypically integrated into a manual assembly workstation and can be applied to a multitude of real-world use cases. In this section we describe the proof-of-concept application of a safety-critical screw connection and briefly discuss the benefits of 4Crypt compared to the basic system design presented in Section 2 as well as previous proposals.

Safety-critical screw connections occur in many production processes, for example in the manufacturing of vehicles such as cars and planes. These types of screw connections need to be executed by qualified workers using the right tools with the correct settings (e.g., the right amount of torque). As an automated documentation and workflow recognition system, the 4Crypt system presented in this paper can provide an invaluable instrument to supervise and document the correct execution of these safety-critical work steps. For this, both our basic system design (c.f. Section 2) and the extended system design (c.f. Section 4) include components for worker authentication, workflow recognition and video documentation. However, only the extended system design encrypts captured videos and provides an interface for the worker to verify the integrity of the included privacy-protection mechanisms. In any case, workers first authenticate and log in to 4Crypt by

holding their personal smartphone on the NFC-reader near the assembly station. By authenticating the worker prior to any work step, we can determine if the executing worker is indeed qualified for the task at hand. Simultaneously, a TPM-based remote attestation protocol is conducted over the same NFC connection. During this attestation process, unique fingerprints of the assistance system's current software configuration are transmitted to the worker's smartphone. By comparing the fingerprints to expected "good" values, the smartphone notifies the worker if the privacy-protection mechanisms of the assistance system can be trusted. Only if these checks are successful, are the cameras activated and the worker steps in front of the assembly station. During the conducted work step, the captured video feed is analyzed by the workflow recognition component and synchronized with a model of the expected workflow. That way the assistance system can warn the worker if an important work step is missed or executed incorrectly. In case of a safety-critical screw connection, this could happen if the worker leaves out some screws, forgets to properly tighten them after placement, uses an unsuitable wrench for the work step, or grabs the wrong kind of screws from the shelf. Furthermore, the system creates a textual documentation of the performed work steps for the supervisor and retains the encrypted video feed in case of a complaint. As described in detail in Section 4, decryption of the retained video file is only possible if the corresponding worker and at least one of the employee's trusted representatives consent to the disclosure by placing their respective smartphone and/or smartcard on the assembly station's NFC-reader.

In comparison with previous approaches, our system stands out primarily in the way it provides strong privacy guarantees for workers. While most existing video documentation systems do not include privacy-protection components at all, there are some proposals for more privacy-friendly smart manufacturing environments. These approaches often include data obfuscation and anonymization schemes [30], retain decentralized data access logs [31], or define privacy best practices (privacy-by-design) when designing assistance systems for smart production facilities [32]. However, in all cases it requires trust in the efforts of the system operator to design and deploy a privacy-respecting smart manufacturing system. As we motivated in Section 3, this assumption is often unsound since there are clear conflicts of interests between the employer as system operator and the workers. Our approach is novel in the way it gives the worker the possibility to cryptographically verify that nobody can read the captured video documentation without consent, and hence empowers true informational self-determination. All in all, we have shown in our proof-of-concept scenario that 4Crypt fulfills the requirements of smart manufacturing environments in terms of worker assistance, automated process documentation and also with regards to privacy protection. Analyzing the performance of 4Crypt in actual productive systems together with industry partners is one of our goals for future work.

7. Conclusions and Future Work

In this work we introduced automated documentation and workflow recognition systems and motivated their utility in smart manufacturing use cases. Based on this we presented our own design of a smart video documentation system capable of recording critical work steps and automatically extracting workflow logs during the manufacturing process. Furthermore, we pointed out the privacy threats and challenges of operating video documentation systems in situations where workers are being recorded. In order to minimize the risk of covert performance evaluation by the employer, we proposed and implemented 4Crypt as a privacy-respecting video documentation system. 4Crypt offers a multi-level encryption scheme implementing the 4-eyes principle on recorded videos to prevent the employer from pressuring workers into disclosing videos without proper cause. Additionally, we designed and implemented a TPM-based remote attestation protocol that can be executed over NFC. This allows workers to verify the trustworthiness of the documentation system using their own personal smartphone before stepping in front of the video cameras. As shown in our concluding security analysis, 4Crypt satisfies the confidentiality and integrity requirements of both the employer and the workers.

As future work, we plan to improve the security guarantees of 4Crypt even further by using advanced trusted execution environments such as Intel's Software Guard Extensions (SGX) instead of TPMs. This can reduce the remaining attack vectors identified in Section 5 and prevent information leakage via metadata. Furthermore, we want to refine the 4Crypt architecture with regards to a more modular system design, which would allow 4Crypt to support more flexible use cases in distributed environments.

Author Contributions: Conceptualization, M.V., P.B.; assembly station, C.L., G.H.; security model, P.B., P.G.W.; system design, P.G.W.; security analysis, P.B.; software, A.B., T.S.; editing, M.B., P.G.W.; supervision, J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Helmholtz Association (HGF) through the Competence Center for Applied Security Technology (KASTEL), subtopic 46.23.04 Engineering Security for Production Systems.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Baroroh, D.; Chu, C.H.; Wang, L. Systematic literature review on augmented reality in smart manufacturing: Collaboration between human and computational intelligence. *J. Manuf. Syst.* **2020**, in press. [CrossRef]
2. Lengenfelder, C.; Frese, C.; Zube, A.; Voit, M.; Beyerer, J. A cooperative HCI assembly station with dynamic projections. In Proceedings of the 52th International Symposium on Robotics, Online, 9–10 December 2020; pp. 1–8.
3. Lengenfelder, C.; Holzbach, G.; Voit, M.; Beyerer, J. Defect Annotation on Objects Using a Laser Remote Control. In *International Conference on Human-Computer Interaction*; Springer: Cham, Switzerland, 2020; pp. 535–542.
4. Wang, B.; Tao, F.; Fang, X.; Liu, C.; Liu, Y.; Freiheit, T. Smart manufacturing and intelligent manufacturing: A comparative review. *Engineering* **2020**, *7*, 738–757. [CrossRef]
5. Winkler, D.; Korobeinykov, A.; Novak, P.; Lüder, A.; Biffl, S. Big Data Needs and Challenges in Smart Manufacturing: An Industry-Academia Survey. Technical Report. 2021. Available online: https://qse.ifs.tuwien.ac.at/wp-content/uploads/CDL-SQI-2021-05.pdf (accessed on 6 August 2021).
6. Tsutsumi, D.; Gyulai, D.; Takács, E.; Bergmann, J.; Nonaka, Y.; Fujita, K. Personalized work instruction system for revitalizing human-machine interaction. *Procedia CIRP* **2020**, *93*, 1145–1150. [CrossRef]
7. Gewohn, M.T. *Ein Methodischer Beitrag zur Hybriden Regelung der Produktionsqualität in der Fahrzeugmontage*; KIT Scientific Publishing: Karlsruhe, Germany, 2019.
8. Braeuer-Burchardt, C.; Siegmund, F.; Hoehne, D.; Kuehmstedt, P.; Notni, G. Finger Pointer Based Human Machine Interaction for Selected Quality Checks of Industrial Work Pieces. In Proceedings of the 52th International Symposium on Robotics, Online, 9–10 December 2020.
9. Thamm, S.; Huebser, L.; Adam, T.; Hellebrandt, T.; Heine, I.; Barbalho, S.; Velho, S.K.; Becker, M.; Bagnato, V.S.; Schmitt, R.H. Concept for an augmented intelligence-based quality assurance of assembly tasks in global value networks. *Procedia CIRP* **2021**, *97*, 423–428. [CrossRef]
10. Yang, X.; Plewe, D.A. Assistance systems in manufacturing: A systematic review. In *Advances in Ergonomics of Manufacturing: Managing the Enterprise of the Future*; AISC: Chicago, IL, USA, 2016.
11. Zhou, J.; Lee, I.; Thomas, B.; Menassa, R.; Farrant, A.; Sansome, A. Applying spatial augmented reality to facilitate in-situ support for automotive spot welding inspection. In Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry, Hong Kong, China, 11–12 December 2011.
12. Fang, B.; Wei, X.; Sun, F.; Huang, H.; Yu, Y.; Liu, H. Skill learning for human-robot interaction using wearable device. *Tsinghua Sci. Technol.* **2019**, *24*, 654–662. [CrossRef]
13. Gorecky, D.; Worgan, S.F.; Meixner, G. COGNITO: A cognitive assistance and training system for manual tasks in industry. In Proceedings of the 29th Annual European Conference on Cognitive Ergonomics, Rostock, Germany, 24–26 August 2011.
14. Prinz, C.; Kreimeier, D.; Kuhlenkötter, B. Implementation of a learning environment for an Industrie 4.0 assistance system to improve the overall equipment effectiveness. *Procedia Manuf.* **2017**, *9*, 159–166. [CrossRef]
15. Mark, B.; Hofmayer, S.; Rauch, E.; Matt, D.T. Inclusion of workers with disabilities in production 4.0: Legal foundations in Europe and potentials through worker assistance systems. *Sustainability* **2019**, *11*, 5978. [CrossRef]

16. Peruzzini, M.; Pellicciari, M. A framework to design a human-centred adaptive manufacturing system for aging workers. *Adv. Eng. Inform.* **2017**, *33*, 330–349. [CrossRef]
17. Malik, A.A.; Masood, T.; Bilberg, A. Virtual reality in manufacturing: Immersive and collaborative artificial-reality in design of human-robot workspace. *Int. J. Comput. Integr. Manuf.* **2020**, *33*, 22–37. [CrossRef]
18. Fullen, M.; Maier, A.; Nazarenko, A.; Aksu, V.; Jenderny, S.; Röcker, C. Machine learning for assistance systems: Pattern-based approach to online step recognition. In Proceedings of the 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 22–25 July 2019; Volume 1.
19. Pattke, M.; Martin, M.; Voit, M. Towards a Mixed Reality Assistance System for the Inspection After Final Car Assembly. In *International Conference on Human-Computer Interaction*; Springer: Cham, Switzerland, 2019.
20. Kim, J.; Park, S.; Lee, U. Dashcam Witness: Video Sharing Motives and Privacy Concerns across Different Nations. *IEEE Access* **2020**, *8*, 110425–110437. [CrossRef]
21. Reuters: China Orders Didi App Downloads Suspended Over Data Violation. Available online: https://www.reuters.com/world/china/china-cyberspace-agency-says-didi-illegally-collects-user-data-2021-07-04/ (accessed on 20 July 2021).
22. Wagner, P.G.; Birnstill, P.; Krempel, E.; Bretthauer, S.; Beyerer, J. Privacy Dashcam—Towards Lawful Use of Dashcams through Enforcement of External Anonymization. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*; Springer: Cham, Switzerland, 2017.
23. Beimel, A. Secret-sharing schemes: A survey. In *International Conference on Coding and Cryptology*; Springer: Berlin/Heidelberg, Germany, 2011.
24. Trusted Computing Group: TCG Specification Architecture Overview. Specification Revision 1.4. 2007. Available online: https://trustedcomputinggroup.org/wp-content/uploads/TCG_1_4_Architecture_Overview.pdf (accessed on 9 August 2021).
25. TrustedGRUB2. Available online: https://github.com/Rohde-Schwarz/TrustedGRUB2 (accessed on 1 July 2021).
26. Sailer, R.; Zhang, X.; Jaeger, T.; Van Doorn, L. Design and implementation of a tcg-based integrity measurement architecture. *USENIX Secur. Symp.* **2004**, *13*, 223–238.
27. Trusted Computing Group: TCG Trusted Attestation Protocol (tap) Information Model. 2019. Available online: https://trustedcomputinggroup.org/wp-content/uploads/TNC_TAP_Information_Model_v1.00_r0.36-FINAL.pdf (accessed on 9 August 2021).
28. gRPC: A High Performance, Open Source Universal RPC Framework. Available online: https://grpc.io/ (accessed on 1 July 2021).
29. Wagner, P.G.; Birnstill, P.; Beyerer, J. Challenges of Using Trusted Computing for Collaborative Data Processing. In *International Workshop on Security and Trust Management*; Springer: Cham, Switzerland, 2019.
30. Wong, K.; Kim, M.H. Privacy protection for data-driven smart manufacturing systems. *Int. J. Web Serv. Res.* **2017**, *14*, 17–32. [CrossRef]
31. Wan, J.; Li, J.; Imran, M.; Li, D.; Amin, F.E. A blockchain-based solution for enhancing security and privacy in smart factory. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3652–3660. [CrossRef]
32. Mannhardt, F.; Petersen, S.A.; Oliveira, M.F. A trust and privacy framework for smart manufacturing environments. *J. Ambient. Intell. Smart Environ.* **2019**, *11*, 201–219. [CrossRef]