

User-centric privacy awareness in video surveillance

Thomas Winkler · Bernhard Rinner

Published online: 1 July 2011
© Springer-Verlag 2011

Abstract Especially in urban environments, video cameras have become omnipresent. Supporters of video surveillance argue that it is an excellent tool for many applications including crime prevention and law enforcement. While this is certainly true, it must be questioned if sufficient efforts are made to protect the privacy of monitored people. Privacy concerns are often set aside when compared to public safety and security. One reaction to this situation is emerging: community-based efforts where citizens register and map surveillance cameras in their environment. Our study is inspired by this idea and proposes a user-specific and location-aware privacy awareness system. Using conventional smartphones, users not only can contribute to the camera maps, but also use community-collected data to be alerted of potential privacy violations. In our model, we define different levels of privacy awareness. For the highest level, we present a mechanism that allows users to directly interact with specially designed, trustworthy cameras. These cameras provide direct feedback about the tasks that are executed by the camera and how privacy-sensitive data is handled. A hardware security chip that is integrated into the camera is used to ensure authenticity, integrity and freshness of the provided camera status information.

Keywords Smart cameras · Video surveillance · Privacy · User feedback · Trusted computing

1 Introduction and motivation

Video surveillance has been recognized as a valuable tool for many applications including crime prevention and law enforcement. As a consequence, surveillance cameras are widely deployed especially in urban environments. This trend is fostered by many factors such as technological advances with the move from analog toward digital systems as well as considerable price drops of camera systems. This way, surveillance cameras have become truly ubiquitous sensors not only deployed by governments, but also by companies and even individuals. In London, for example, an average citizen is captured by surveillance cameras 300 times a day [7].

It is commonly agreed that video surveillance not only can help to increase public safety and security, but also bears some risks. One of them is an increasing loss of personal privacy. This problem is addressed from several sides with mixed success. Governmental regulations intend to protect the privacy of citizens. However, these regulations are difficult to enforce and usually lag behind the rapid developments of the surveillance industry. Several efforts from research and academia try to address the problem from a technological direction. Typically, they are based on the identification and protection of privacy-sensitive image regions such as human faces or vehicle license plates. Obviously, adding such technology to a camera system not only increases complexity, but also cost. Therefore, manufacturers and operators will be reluctant to adopt these techniques. Also, even if they are applied, it remains difficult for monitored people to differentiate

T. Winkler (✉) · B. Rinner
Alpen-Adria-Universität Klagenfurt,
Institute of Networked and Embedded Systems,
Lakeside Park B02b, 9020 Klagenfurt, Austria
e-mail: thomas.winkler@aau.at

B. Rinner
e-mail: bernhard.rinner@aau.at

between trustworthy and conventional cameras. One promising way to escape this dilemma is by community-based efforts. By registering and mapping surveillance cameras, these projects aim at raising public awareness about privacy problems in video surveillance.

In this study, we contribute to the state of the art in the following areas. First, we extend the idea of community-based registration of cameras with the goal of providing location-based notifications to users. This allows them to stay informed and aware about cameras in their environment. The overall goal is to increase public awareness of video surveillance and related privacy issues and, at the same time, put pressure on camera operators to integrate privacy protection into camera systems. Second, we describe different levels of privacy awareness that can be realized with such a system. The actually reachable level depends on the quality of available information as well as the deployed technical infrastructure. Our third and most important contribution is targeted toward reaching the highest awareness level. This level assumes the availability of specially designed camera systems. Users can directly interact with cameras to obtain feedback about running applications and integrated privacy protection mechanisms. While this study on direct user feedback is based on our previous contributions [51, 52], it goes beyond them in several ways. A revised version of the user-based attestation protocol initially proposed in [52] is presented. As part of this redesign, the protocol was not only elaborated in more detail, but also a significant speedup was achieved by introducing parallelism. This allows to also reduce the amount of time users have to point their smartphone's screen toward the camera. By shortening this period of time, the overall procedure becomes a lot more comfortable and convenient for users. This article also provides comprehensive performance measurements from our TrustCam prototype system. In our previous study, no TPM-equipped camera was available and only performance estimations based on a TPM emulator were made. Furthermore, an Android smartphone application was developed to be able to evaluate the user-centric aspects of the system under realistic conditions.

The remainder of this article is organized as follows. Section 2 explains the goals and underlying assumptions that motivate this study. Thereafter, Sect. 3 provides an overview of related study on privacy protection in video surveillance together with a comparison of existing approaches. Thereafter, in Sect. 4, we present our concept of a user-centric privacy awareness system based on community-collected information about cameras. Subsequently, in Sect. 5, we outline the design of a direct user-feedback system based on our TrustCAM prototype. Practical aspects including a prototype implementation and a detailed performance analysis are discussed in Sect. 6

Finally, Sect. 7 concludes the paper and presents an outlook to future studies.

2 Assumptions and goals

This section outlines the goals and assumptions that are fundamental for this study. Even though video surveillance raises many privacy and security questions, it has become an established and widely used tool for many safety and enforcement applications. Therefore, it would be naive to assume that video surveillance systems will vanish from public places, streets and cities. Instead, in this study we propose a concept that seeks to establish a balance between the needs of cameras operators and those of monitored people without (a) crippling the usefulness of camera systems and (b) relying solely on the goodwill of operators.

Increasing privacy awareness Today, video surveillance is already omnipresent, but many people are still unaware of this fact or do not pay attention. One reason for this attitude might be the perception that there is very little that an individual can do to change this situation. However, community-based efforts such as Wikipedia have illustrated that decentralized efforts can gain sufficient momentum and produce widely recognized content. This idea has been picked up by community-based projects that register and map video surveillance cameras in publicly accessible databases. Our study is inspired by this idea and suggests extending existing systems to pursue the following goals:

- Contributing to public databases of video surveillance databases should be simplified. We suggest the use of smartphones as a tool for the registration and mapping of new surveillance cameras. Additionally, the smartphones are used to define a personal privacy policy and to receive custom notifications when entering areas where this privacy policy is violated. We assume that the wide availability of information about existing camera infrastructure will help to increase public awareness of potential privacy problems.
- Raised public awareness and concerns about the loss of privacy will increase the pressure on operators to integrate privacy protection techniques into camera systems. Various implementation forms such as detection and hiding of sensitive image regions or encryption of image data are possible. Public demand eventually will also lead to new governmental regulations that stipulate what minimal protection must be provided by camera systems.

Balance the needs of operators and users This study does not suggest giving users control over what cameras do. In such a scenario, a video surveillance system would most likely no longer be usable for its intended purpose.

Therefore, in the presented approach operators remain in full control of their equipment and infrastructure. However, we assume that increasing public awareness and demand together with media coverage will hopefully convince camera manufacturers and operators to integrate privacy protection mechanisms into their products and systems.

Verifiable privacy protection When operators incur extra cost and integrate privacy protection into their cameras, they clearly also want to benefit from this effort. This can be achieved by advertising the implemented protection mechanisms to improve the operator's image and to increase public acceptance of video surveillance. However, advertising by its own is not enough. Monitored people need a reliable way to verify the claims of operators. A primary goal and core contribution of this study is the design of such a direct user feedback system built on Trusted Computing technology.

3 State of the art

Privacy protection in video surveillance has been recognized as a very important issue. As a consequence, various researchers have proposed different approaches to protected privacy of monitored persons [4, 7, 12, 28, 39]. We summarize related study on privacy protection in video surveillance and differentiate between approaches with and without selective protection and involvement of monitored people. Even if users are involved, control still remains in the hands of operators. Therefore, a logical next step is to empower users to more actively participate in privacy protection. This idea is fundamental for community-based approaches that register and map surveillance cameras installed in public places. After describing related projects, we conclude this section with a classification of privacy protection approaches and discuss observations and possible implications.

3.1 Privacy protection in video surveillance

Senior et al. [39] discuss critical aspects of a secure surveillance system including what data are available and in what form (e.g., raw images vs. metadata), who has access to data and in what form (e.g., plain vs. encrypted) and how long these are stored. User privacy is a major concern that is addressed in the proposed concept. Incoming videos are analyzed and sensitive information is extracted. The extracted data are re-rendered and multiple streams with different levels of data abstraction are created. By encryption of streams, multi-level access authorization is realized. The authors suggest that video analysis, processing and encryption could either be done by a dedicated privacy console or directly by the cameras.

Cavallaro [6, 7] argues that digitalization of video surveillance introduces new privacy threats. Therefore, personal and behavioral data should be separated directly on the camera. While system operators only get access to behavioral data, a separate stream containing personal data is made available to law enforcement authorities. A benefit of this strict separation is prevention of operator misuse. Possible implementation approaches are not discussed in this study.

Fleck [15, 16] employs smart cameras in an assisted living scenario. The cameras are used to monitor the behavior of persons and detect unusual behavior such as a fall. For that purpose, the cameras create a background model, which is the basis for detecting motion regions. Detected objects are tracked and their behavior is analyzed using support vector machines. Privacy protection is achieved by either transmitting only event information or replacing detected objects with abstracted versions. It is assumed that the camera's housing is sealed such that manipulation can be detected by the camera and leads to the termination of its services. Protection against software attacks such as integrity checks or data encryption is not part of the current system.

Moncrieff et al. [28] argue that most of the proposed systems rely on predefined security policies and are either too intrusive or too limited. Therefore, they suggest applying dynamic data-hiding techniques. Via context-based adaptation, the system could remove or abstract privacy-sensitive information during normal operation, while in case of an emergency, the full, unmodified video stream is automatically made available. This way, the system remains usable for the intended purpose, but protects privacy during normal operation.

Boult [4] argues that many existing approaches are targeted at removing privacy-sensitive image data without providing mechanisms to reconstruct the original image. Based on this observation, he proposes a system called PICO that relies on cryptography to protect selected image regions such as faces. It allows to monitor actions of a person without revealing the person's identity. The faces are only decrypted if, e.g., a crime was committed by the person. Encryption is supposed to be done as part of image compression and uses a combination of symmetric and asymmetric cryptography. Additionally, it is suggested to compute checksums of frames or sub-sequences to ensure data integrity. In a related study, Chattopadhyay and Boult present PrivacyCam [8], a camera system based on a Blackfin DSP clocked at 400 MHz, 32 MB of SDRAM and an Omnivision OV7660 color CMOS sensor. uClinux is used as the operating system. Regions of interest are identified based on a background subtraction model and resulting regions are encrypted using an AES session key. Rahman et al. [34] also propose that regions of interest are

encrypted. In their approach, they do not rely on established crypto-systems but propose that chaos cryptography is used.

Dufaux and Ebrahimi [12] suggest scrambling sensitive image regions. After detection of relevant areas, images are transformed using DCT. The signs of the coefficient of sensitive regions are then flipped pseudo-randomly. The seed for the pseudo-random number generator is encrypted. Decryption is only possible for persons who are in possession of the corresponding decryption key. According to the authors, the main benefits are minimal performance impact and that video streams with scrambled regions can still be viewed with standard players. A similar approach is discussed by Baaziz et al. [1] where in a first step motion detection is performed followed by content scrambling. To ensure data integrity, an additional watermark is embedded into the image, which allows detecting manipulation of image data. Limited reconstruction of manipulated image regions is possible due to redundancy introduced by the watermark. Yabuta et al. [54] also propose a system where DCT-encoded image data are modified. They however do not scramble regions of interest, but extract them before DCT encoding and encrypt them. These encrypted regions are then embedded into the DCT-encoded background by modifying the DCT coefficients.

Tansuriyavong et al. [41] present a system used in an office scenario that blanks the silhouettes of persons. Additionally, the system integrates face recognition to identify previously registered persons. Configuration options allow choosing what information should be disclosed—full images, silhouettes, names of known persons or any combination thereof.

Troncoso-Pastoriza et al. [45] propose a generic video analysis system that is coupled with a digital rights management (DRM) system. By exploiting the hierarchical structure of MPEG-4, the authors propose selective visualization of video objects either in clear or in obfuscated forms. Access to sensitive video objects is conditionally granted depending on the rights of the observer and the individual policies of monitored users. Sensitive content is protected by encryption. Intellectual Property Management Protection (IPMP) descriptors, as standardized in MPEG-4, are used to describe these encrypted streams. Access rights to protected video objects are formulated using the MPEG-21 Rights Expression Language (REL).

Finally, the Networked Sensor Tapestry (NeST) software architecture by Fidaleo et al. [14] represents a more generic privacy protection approach. Its design is not limited to videos and images, but can handle arbitrary sensor data. The system uses a centralized architecture. An important component is the privacy buffer that is run on the server. Data received from the clients are fed into this privacy buffer. The buffer can be extended and configured

by means of privacy filters and a privacy grammar. If incoming data are qualified as private by one of the privacy filters, the data do not leave the privacy buffer. Non-private data are forwarded to a routing component that manages distribution of data to interested clients.

3.2 Selective privacy protection and user involvement

To protect the privacy of selected users, systems have been presented that allow to remove known, trusted users from captured video. Some approaches go even further and give monitored persons control over who is able to access personal video data. Due to limited reliability of computer vision in detecting personal image data, many researchers rely on portable devices carried by users for identification and localization.

Brassil [5] proposes a privacy enabling device (PED) that gives users control over their personal data. When activated, the PED records the location of the person together with timestamps. These data are uploaded to a clearinghouse. Before a camera operator discloses videos to a third party, the clearinghouse has to be contacted to check if an active PED was in the vicinity of the camera at the time in question. If so, video data have to be anonymized. Due to the absence of feedback, users have to trust camera operators to follow the advertised procedures. Wickramasuriya et al. [49] perform real-time monitoring of the environment to increase user privacy. In particular, they suggest the use of motion sensors to monitor rooms or areas. If motion is detected, an RFID reader is triggered that tries to read the RFID tag carried by the person who entered the area. If no RFID tag can be found or the security level of the tag does not grant access to the area, a camera that oversees the region is activated. Image regions containing persons with valid RFID tags are blanked such that only potential intruders remain visible.

Chinomi et al. [11] also use RFID technology to detect known users. RFID readers, deployed together with cameras, are used to localize RFID tags carried by users based on signal strength. This location information is then mapped to motion regions detected by the cameras. As the RFID tag identifies the person, the individual privacy policy can be retrieved from a database. This policy defines the relationship between the monitored person and potential observers. Based on this, different forms of abstracted data are delivered by the system. Abstractions include simple dots showing only the location of a person, silhouettes as well as blurred motion regions. Also Cheung et al. [9] use RFID for user localization. Corresponding motion regions are extracted from the video and encrypted with the user's public encryption key. This key is retrieved from a database via the user ID from the RFID tag. The blanked regions in the remaining image are filled with

background image data using video inpainting [10]. The encrypted regions are embedded into the compressed background image using data-hiding techniques similar to steganography. Since decryption of privacy-sensitive image regions requires the user's private key, active user cooperation is necessary to reconstruct the original image. A dedicated mediator establishes contact between users and observers who are interested in the video data. In a study from the same research group, Ye et al. [55] and Luo et al. [23] do not use RFID tags for identification, but biometric information. As part of their anonymous biometric access control system, iris scanners are installed at the entrances of areas under video surveillance. Based on this, authorized individuals are then obfuscated in the captured video. The anonymity of authorized persons is maintained by using homomorphic encryption.

An approach that does not need special devices carried by users is presented by Schiff et al. [36]. Their "respectful cameras" use visual markers such as yellow hard hats worn by people to identify privacy-sensitive regions. Specifically, they remove the person's faces from images. Spindler et al. [40] apply similar ideas in the context of building automation and monitoring applications. Personal data are obfuscated based on individual privacy settings. For identification and localization, the authors suggest relying on computer vision. For the prototype, this was not implemented but replaced by manual selection of privacy-sensitive regions.

3.3 Community-driven registration of surveillance cameras

Community-based projects such as Wikipedia have demonstrated the feasibility of collaborative efforts to produce high quality content. Similar concepts have been proposed to register and map video surveillance systems deployed in public areas. One such project is based on OpenStreetMap [29] and makes camera positions available as a map overlay. Another project with similar goals is MapCams.org [24]¹. In February 2010, the city of Paris announced a plan [33] to establish a police-controlled network of about 1,300 surveillance cameras. The locations of the cameras already installed as well as the planned cameras have been mapped on Google maps. Figure 1 shows screenshots of all three services. At present, these projects are in their infancy. New cameras are added via simple Web-interfaces. Dedicated applications for mobile phones allowing users to directly register camera locations using GPS are not yet available.

¹ Mapcams.org went offline due to technical difficulties in fall 2010. At the time of writing this article, it has not yet resumed its services.

We believe that such community efforts will attract considerable interest in the future. Public attention and awareness about video surveillance and related privacy problems are rising. At the same time, government regulations are not keeping up with the high pace at which new surveillance and monitoring technologies are developed and deployed. Community-based efforts can help to compensate this and empower citizens to actively participate in protecting their own privacy. Eventually, camera operators might recognize such community efforts as a way toward more transparency. They could use community platforms to publish information about their systems and thereby increase public acceptance of video surveillance. At the end, this might lead to a win-win situation for both sides.

3.4 Classification, observations and implications

Table 1 presents a comparison of the previously discussed approaches to protect user privacy in video surveillance. The table is split into two main categories, privacy protection and user involvement, with several sub-categories. Subsequently, we present the meaning of the individual categories and discuss how related work fits into these categories.

Detection of sensitive regions This denotes the capability of a system to detect privacy-sensitive image regions. These are, e.g., human faces or vehicle license plates. If this system component does not work reliably, privacy is at risk. A single frame of a video sequence where sensitive regions are not properly detected can break privacy protection for the entire sequence. As illustrated in Table 1, detection of sensitive image regions is a core component of all reviewed approaches.

Blanking One way to deal with sensitive image regions is to completely remove them from the image leaving behind blanked areas. While providing perfect user privacy, the usefulness of the system is reduced since not even basic user behavior can be observed and identities of persons are lost. Nevertheless, this basic approach is part of the majority of existing privacy protection systems. Some researchers such as Cheung et al. [10] aim to erase known, trustworthy persons from captured video. Instead of leaving behind blanked areas, they apply video inpainting techniques to fill the blank areas with background.

Obfuscation and scrambling The purpose of obfuscation is to reduce the level of detail in sensitive image regions such that persons can no longer be identified while their behavior remains perceptible. Researchers apply different techniques including mosaicing, pixelation, blurring [11, 49] or high, lossy compression. Image scrambling is a technique where sensitive regions in, e.g., JPEG compressed images are obscured by pseudo-randomly modifying the region's DCT coefficients [12]. As can be seen in

Fig. 1 Examples of community-driven registration and mapping of video surveillance cameras.

a Screenshot of surveillance camera locations in OpenStreetMap.org [29] for central Vienna. **b** Surveillance camera locations in central London registered on mapcams.org [24]. **c** Locations of 1,300 existing and planned police-controlled CCTV cameras in Paris [30, 33]

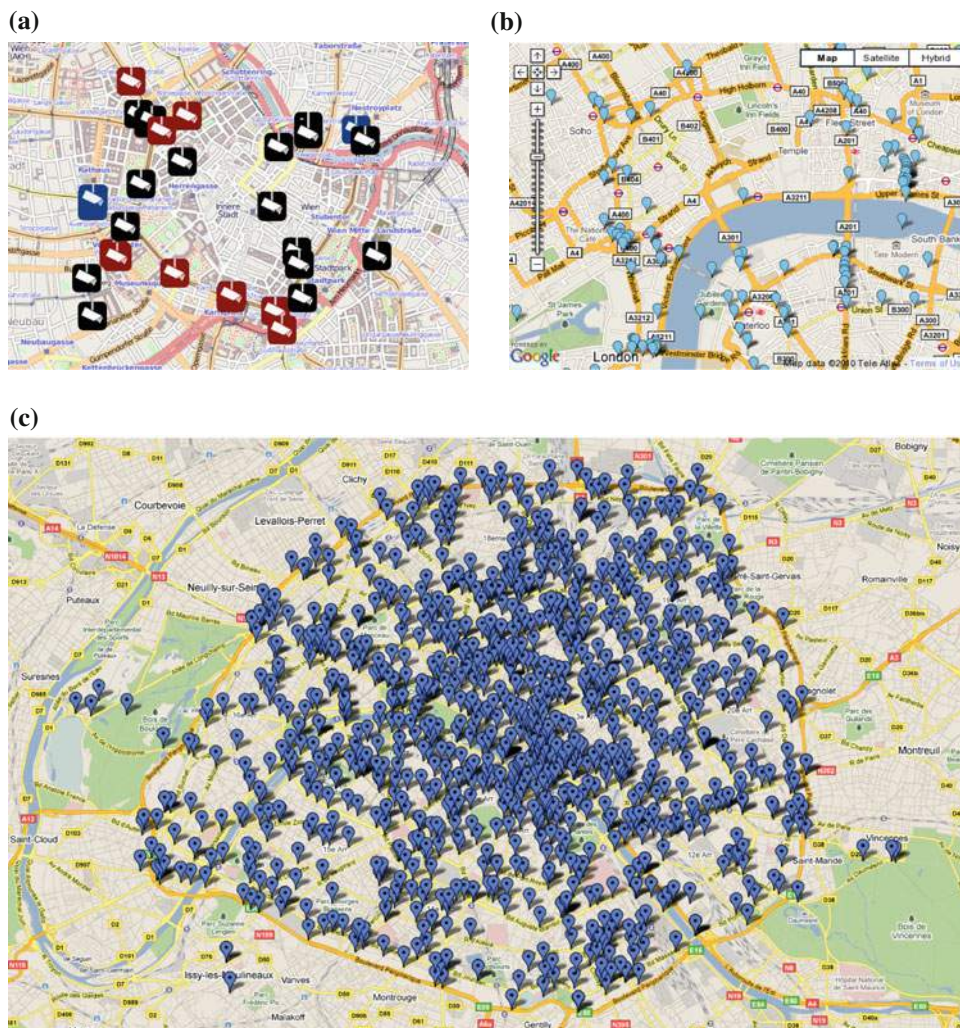


Table 1, about half of the approaches make use of obfuscation or scrambling techniques. In a recent study, Dufaux and Ebrahimi [13] present a framework for the evaluation of privacy protection mechanisms. Their results indicate that simple approaches such as pixelation and blurring offer only limited protection. Blurred or pixelated human faces can often still be recognized with standard face recognition algorithms. Contrary to that, scrambling mechanisms perform much better with recognition rates of nearly 0%.

Abstraction This popular technique replaces sensitive image regions with, e.g., bounding boxes or, in case of persons, with silhouettes and stick figures [39]. Another form of abstraction is meta-information attached to a video. This can be object properties such as position and dimensions, but also names of identified persons [41]. Depending on the type of abstraction, either behavior, identity or both can be preserved. Note that if identity is preserved, additional protection (e.g., by encryption) should be

considered. Abstraction is applied by about 50% of the reviewed works.

Encryption Data encryption is used by half of the systems presented in Table 1 to keep sensitive regions confidential. When encrypted, sensitive image regions can no longer be viewed by persons who do not have the appropriate decryption keys. Simple encryption not only protects the identity of monitored persons, but also their behavior. Upon decryption, both identity and behavior are revealed. By using multiple encryption keys, a system can be designed that requires multiple operators to cooperate in decrypting the original data. Such a design provides a certain degree of protection against operator misuse. Table 1 shows that encryption is a technique that is frequently used for privacy protection.

Multiple privacy levels Support for multiple privacy levels is usually based on encryption and denotes that one single video stream contains different levels of information. These could be the unmodified sensitive image

Table 1 Classification of related work on privacy protection in visual surveillance

	Privacy Protection						User Involvement	
	Sensitive Regions	Blanking	Obfuscation and Scrambling	Abstraction	Encryption	Multi-Level	Consent and Control	Feedback
Baaziz [1]	●	○	●	○	○	○	○	○
Boult [4] and Chattopadhyay [8]	●	○	○	○	●	○	○	○
Brassil [5]	●	●	●	○	○	○	●	○
Cavallaro [6,7]	●	●	○	●	○	●	○	○
Cheung et al. [9,10,23]	●	●	○	○	●	○	●	○
Chinomi et al. [11]	●	●	●	●	○	●	●	○
Dufaux and Ebrahimi [12,13]	●	○	●	○	●	○	○	○
Fidaleo et al. [14]	●	●	○	○	○	○	○	○
Fleck and Strasser [16,15]	●	●	●	○	○	○	○	○
Moncrieff et al. [28]	●	●	●	○	○	○	○	○
Schiff et al. [36]	●	●	○	○	○	○	●	○
Senior et al. [39]	●	○	○	●	●	●	○	○
Spindler et al. [40]	●	●	○	○	●	○	○	○
Tansuriyavong and Hanaki [41]	●	●	○	●	○	●	○	○
Wickramasuriya et al. [49]	●	●	●	●	○	●	○	○
Troncoso-Pastoriza et al. [45]	●	○	○	○	●	●	○	○
Winkler and Rinner [51]	●	●	○	●	●	●	○	○
Yabuta et al. [54]	●	●	○	○	●	○	○	○

The properties are described in Sect. 3.4. White bullets represent unsupported properties, gray bullets denote partially realized properties and black bullets stand for fully covered properties

regions, obfuscated versions with blurred faces as well as abstracted versions. Depending on their sensitivity, these levels can be encrypted with one or multiple encryption keys. A multi-level approach allows designing a privacy protection system that presents different types of information to observers depending on their security clearance. While low-privileged operators can only access the version of the stream where behavioral data is visible, supervisors or government agencies can get access to the original data that contains the identity of monitored persons. Table 1 illustrates that many researches combined different approaches into multi-level privacy protection systems.

User consent and control Today, camera installations are often marked with signs or stickers that advertise their presence. User consent to video surveillance is given implicitly by acknowledging these signs when entering the

area. By handing out PEDs or RFID tags to known and trusted users, some of the approaches from Table 1 realize a stronger form of awareness about video surveillance. Users equipped with PEDs or RFID tags are not only made aware of the installed cameras, but also get a certain degree of control over their privacy. By carrying the devices with them, they can actively choose not to be captured by the system. The approach of Cheung et al. [9] goes even further. By using public key cryptography to protect personal information, users get full control over privacy-sensitive data since they have to actively participate in the decryption of this data.

User feedback In current systems, users have to trust operators to protect their privacy. To establish this trust, Senior et al. [39] suggest that surveillance equipment should be certified and the results should be made visible,

e.g., by stickers attached to cameras. For users, however, it is difficult to evaluate if this certification is still valid. The software of a smart camera might have been changed by the operator without re-certification of the system. In Sect. 5 of this study, we present an alternative approach to provide user feedback. Using a handheld device, users can directly query the current status of a camera.

The review of related work leads to the following observations:

- Many systems rely on computer vision to detect privacy relevant information such as human faces. Unfortunately, computer vision is not yet advanced enough to work reliably under all conditions. This, however, is vital since a single mis-detected frame subverts all privacy protection efforts for an entire video sequence. But, it is not only the reliability of computer vision that is problematic. Work by Saini et al. [35] demonstrates that even if primary identifiers such as human faces are removed, persons often can still be identified based on secondary identifiers. Secondary identifiers are, e.g., spatial and temporal information as well as activities that are characteristic for a person.
- In many approaches, dedicated devices for localization and identification of persons are used. While this works well when applied for closed communities such as the staff of a hospital [49], we believe that this is not scalable enough for larger deployments such as an entire city. Furthermore, it must be noted that RFID security is typically not considered. In an actual system, countermeasures against, e.g., cloning of tags is a critical requirement.
- Currently, privacy protection is largely the responsibility of system operators. Users have to trust that operators implement adequate protection mechanisms. However, it is questionable if operators voluntarily take the extra cost and complexity of integrating privacy protection techniques without being forced by, e.g., government regulations. By putting tools for privacy awareness and protection into the hand of users, this situation might slowly change in the future.

4 User-centric privacy awareness

Even though user feedback is an important aspect to increase public acceptance of video surveillance, the classification of related work in Sect. 3.4 illustrates that it is rarely addressed. In principle, user feedback techniques should provide monitored people with tools that allow them to get information about a camera system. Basic feedback could include information such as the owner of a camera, its purpose, who has access to recorded video data

as well as how long this data is stored. In more advanced forms, information about the system status in terms of applied privacy protection techniques and executed applications could be made available. A concept for providing user feedback via direct interaction between a camera and a user will be presented in Sect. 5.3 It relies on Trusted Computing remote attestation techniques, which allow provision of reliable system status information.

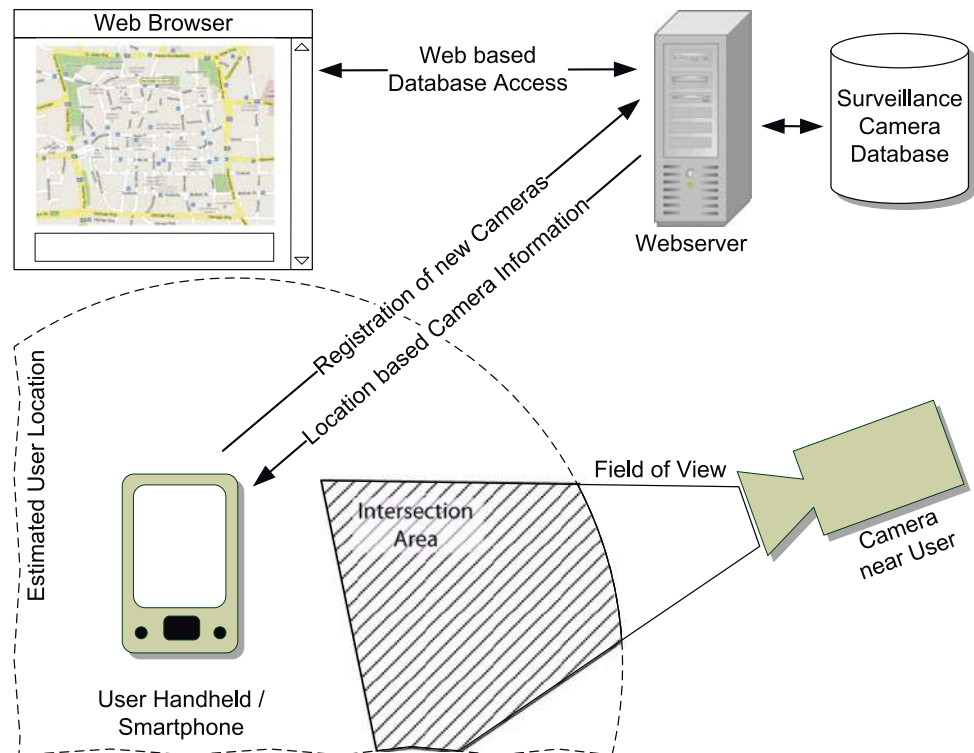
For the successful deployment of such a user-based attestation scheme, operator cooperation is required. However, we cannot automatically assume this cooperation as given. To still be able to increase privacy awareness despite the current lack of operator cooperation, we propose a more general concept. Within this concept, user-based attestation serves as the strongest and most reliable way of providing feedback. Eventually, public demand and governmental regulations might require surveillance system operators to disclose more information about their systems. Once this happens, our concept can unfold its full potential.

The main goal of the proposed system is to collect information about areas under video surveillance and to make this data freely available. The system is not designed to differentiate between different types of users by selectively protecting the privacy of a few, trusted individuals. Our approach is based on the work of existing community projects that register and map video surveillance cameras (compare Sect. 3.3). It extends these projects in several ways and makes use of the collected data to provide personalized privacy alerts as illustrated in Fig. 2. Participants interact with the system via their smartphones. The two primary use cases for the smartphone are the following.

Camera registration Via a dedicated application and the smartphone's GPS receiver, users can register yet unknown cameras and upload this information to the community database. Aside from basic location information, the application allows additional data to be entered. Camera orientation can be sketched on a live map preview, and properties such as the type of the camera (e.g., fixed, PTZ, . . .) can be entered. As an alternative to the smartphone application, cameras can also be registered via a conventional Web interface. The system is also open to camera operators who seek to provide transparency by disclosing information about their surveillance cameras. Due to its openness, the system will probably face the same problems regarding correctness of information as other open, community-controlled platforms such as Wikipedia. These issues can be addressed by, e.g., integrating rating and reputation mechanisms.

The advanced form of the system, as discussed in Sect. 5.3, allows users to directly interact with cameras using visual communication. A camera reports its status and the applied privacy protection mechanisms.

Fig. 2 The user-centric privacy awareness system allows monitored people to actively contribute to a public database that contains the locations and properties of the installed cameras. This information is used in a customizable alerting system



User-specific alerts The second use case for the smartphone is a user-specific alert system. Users can define individual privacy policies based on a list of camera properties stored in the community database. Example properties are: “camera streams unencrypted video”, “camera is operated by XYZ” or “camera can zoom”. Data from the community database is used for personalized alerts in case the user approaches an area with cameras that violate this privacy policy. To avoid information overload, the privacy policy also allows defining filters such that certain information is excluded. A user can, e.g., decide that cameras that are operated by a trusted entity (e.g., the government or the police) should not be reported. Furthermore, users can specify that they only want to be notified about recently added cameras. Alternatively, various other forms of presenting the collected information can be imagined. For example, the privacy badge by Gisch et al. [18, 19] visualizes privacy loss using the metaphor of a dosimeter that shows the accumulated loss of privacy.

The system bears similarities to the privacy awareness system for UbiComp environments presented by Langheinrich [22]. In his approach, users carry a privacy assistant in the form of a PDA. The concept assumes that UbiComp environments are equipped with privacy beacons that announce which data-collecting services are present in the environment. The user’s privacy assistant collects and forwards this information to the user’s personal privacy proxy located on the Internet. The privacy proxy compares

the user’s privacy policies with the settings of the data-collecting devices as announced by the privacy beacon. If mismatches are detected, services in the user’s environment are selectively disabled such that compliance with the user’s privacy policy is achieved. A noteworthy difference between Langheinrich’s system and our concept is that Langheinrich allows users to modify the behavior of the installed data-collecting devices. In contrast to that, the presented privacy awareness system is limited to providing location-based notifications. The primary reason for this decision is that in areas under video surveillance, usually many people with varying privacy requirements are present. The only appropriate way to handle such a situation is to apply the superset of these policies. It can be assumed that this would make video surveillance system useless in most situations since the majority of the cameras would be deactivated via the users’ policies.

Therefore, we explore a different approach and focus on increasing awareness about the loss of privacy without giving users control over the cameras. The underlying assumption is that privacy protection should be integrated into the camera’s computer vision applications. If thereby privacy is inherently protected, there is no need for users to exercise direct control over what cameras are doing. What, however, is absolutely necessary is a mechanism that allows users to verify that cameras operate in a privacy-preserving way. The fundamental idea of this approach is to establish a balance between the needs of operators and

those of monitored people such that the system remains usable for its intended purpose, while privacy is protected in a way that is verifiable for users.

As already mentioned, operator cooperation cannot be expected from the beginning. Therefore, the concept supports different privacy awareness levels which are subsequently described. While basic awareness levels can already be achieved today and without operator cooperation, advanced levels also depend on the deployment of new infrastructure in the form of trustworthy cameras. The privacy awareness levels provided by the system are the following.

Level 0—no awareness This lowest level of awareness is typical for today's video surveillance systems. Most people are unaware of cameras in their environment. If at all, they are informed about the presence of cameras via printed notes or attached stickers. People who choose not to participate in one of the higher levels will remain at this lowest awareness level. By not participating, these users do not lose anything compared to the status quo.

Level 1—basic awareness With the information available in today's evolving surveillance camera databases and maps, users can be notified about the presence of cameras in their environment. Camera location information is used by the user's smartphone to issue an alarm if the user approaches an area under video surveillance.

Level 2—extended awareness A prerequisite for extended privacy awareness is the availability of additional information about installed camera systems. Besides basic information such as the location, this could be, e.g., the orientation and field of view of cameras, who owns and operates them as well as their purpose. While externally visible properties such as orientation and estimated field of view can be added to the database by individuals, other information cannot. Attributes such as ownership and purpose can, most likely, only be made available by system operators.

Level 3—full awareness with direct user feedback At level 2, users have to trust the correctness of information about system properties made available in community databases. Level 3 goes beyond that by providing direct feedback and proof about the tasks executed on a camera. Based on cryptographic techniques, users can query the status of a camera and receive evidence as to which applications are executed. Status information obtained directly from a camera can also be uploaded into the community database to make it accessible to other users. Clearly, this status information is only a snapshot in time, but it nevertheless can provide valuable privacy information for other users.

5 Design of a direct user feedback system

In this section, we describe a camera system that supports direct user feedback. This means that the camera provides

functionality that allows an interested user to directly query the camera's status and get information that goes beyond basic data typically available in community databases. To facilitate that, we assume that the camera is a modern, smart camera system. Smart cameras essentially are embedded computer vision systems that come with substantial computing power, memory as well as network connectivity. This class of camera systems is no longer used only in research, but has also begun to be deployed in commercial applications. The cameras can not only do on-board image processing and analysis, but are also powerful enough to facilitate the integration of security and privacy protection mechanisms.

In previous studies [50, 51, 53], we have presented an approach toward a trustworthy embedded smart camera. It is an example how a camera operator could integrate privacy protection mechanisms in a secure way. The approach is based on a combination of computer vision and cryptography to achieve multi-level protection. Figure 3 shows an overview of the system. Based on a periodic heartbeat, system operators can reliably check the status of a camera. This includes the currently executed applications as well as the detection of system reboots. Persons who are present in the field of view of the camera are protected by encrypting the motion regions. Furthermore, multiple levels of data abstraction are supported, which is demonstrated by replacing persons with their silhouettes. Such abstracted data representations allow operators at the control station to monitor the behavior of persons while their identity is protected. For the decryption of the original motion regions, the system requires the cooperation of multiple operators to prevent misuse by individuals. We would like to emphasize that the system presented in Fig. 3 only serves as an example of how security and privacy protection can be integrated into a camera system. We neither recommend nor require a specific approach, but operators are free to choose the most appropriate solution.

A primary focus of this work is on presenting a concept how the integrated privacy protection mechanisms can be reliably reported to monitored people. For a user, privacy protection mechanisms are not obvious. From the outside, a trustworthy camera system does not look different from a conventional system. Users only can trust that a system behaves as advertised by the operator. With our user feedback concept, we manage to overcome this limitation.

Before discussing the technical aspects, we would like to outline the user experience we want to achieve. A primary goal is an intuitive mechanism that enables users to (1) select the camera they are interested in, (2) securely query the status of this camera and (3) get a comprehensible description of the system and privacy protection properties of the camera. Based on these requirements, the following major challenges can be identified:

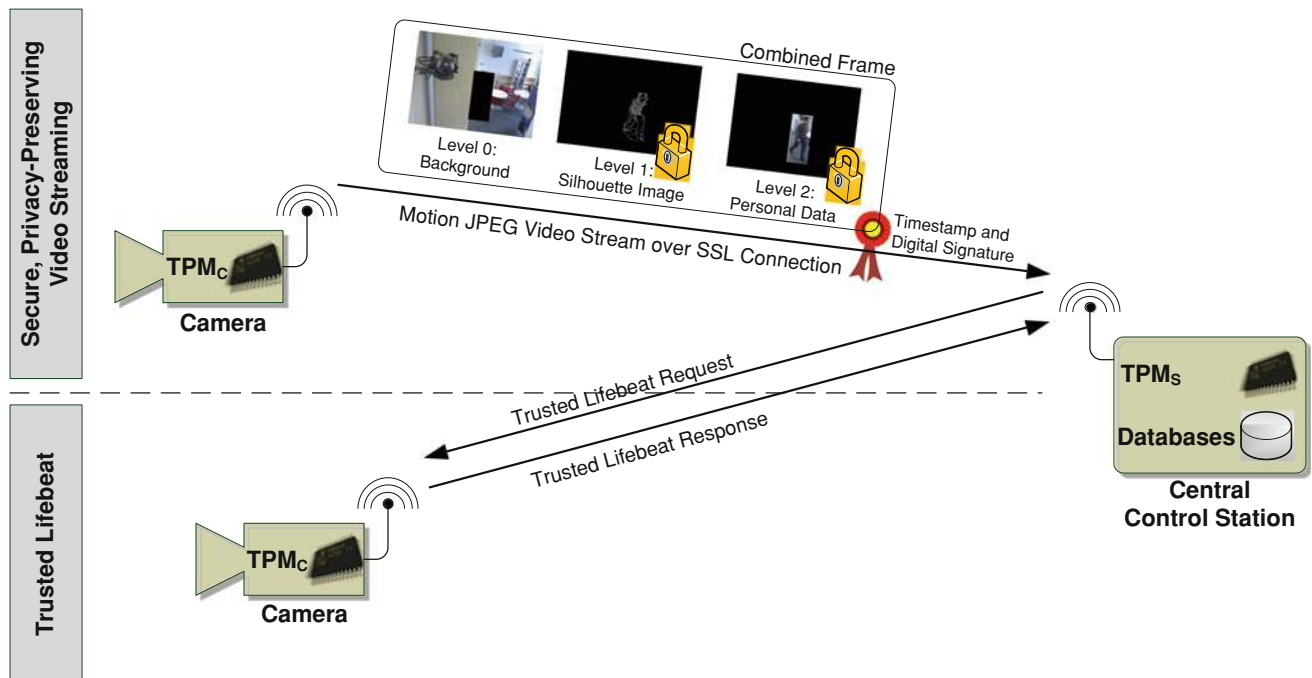


Fig. 3 From the central control station, the status of a camera can be reliably checked via a trusted lifebeat. Secure video streaming ensures confidentiality, integrity, authenticity and freshness of image data.

Secure status reporting Whether privacy is properly protected primarily depends on the software that is running on the embedded camera system. As a consequence, users require a mechanism to find out what applications are executed. To achieve this goal, we rely on trusted computing (TC) [25, 46] which is a hardware-based security solution. In TC, a microchip is used to record and report the status of a system by taking hashes of each system component. These hash values are securely stored and can be reported to an external challenger. In TC, this secure system status reporting is called attestation. For readers who are not familiar with TC, we provide an overview of the most relevant TC concepts in Sect. 5.1

Authentic communication channel A central aspect is the establishment of a communication channel between the user and the camera that is guaranteed to be authentic. Wireless communication is no ideal choice because it is difficult to ensure that the response actually comes from the intended camera. Therefore, we rely on an approach similar to secure device pairing techniques proposed for mobile phones [26]. An in-depth discussion of our system, including the involved protocols, is provided in Sect. 5.3

Comprehensibility for end-user The outcome of the attestation process is a set of hashes that represent the software running on the camera. For an average user, these hashes have little meaning. Therefore, an entity is required that translates these hashes into human comprehensible

descriptions and system properties. In our concept, we rely on an external, trusted third party for translation of hash values into properties. The underlying concepts are outlined in Sect. 5.3

5.1 Trusted computing fundamentals

Trusted Computing [25] is an industry initiative headed by the Trusted Computing Group (TCG) [46]. The main output of the group is a set of specifications for a hardware chip—the Trusted Platform Module (TPM) [48]—and software infrastructure like the TCG Software Stack (TSS) [47]. The TPM typically is implemented as a microcontroller (execution engine) with accelerators for RSA and SHA-1. Additionally, the TPM provides a random number generator and limited amount of volatile and non-volatile memory.

RSA keys can be generated for different purposes like encryption or signing. Upon creation, keys can be declared to be migratable or not. While migratable keys can be transferred to a different TPM, non-migratable keys cannot. Regardless of key type and migratability, a private TPM key can never be extracted from the chip as plaintext but only in encrypted form. By definition, every key must have a parent key that is used to encrypt the key when it has to be swapped out of the TPM due to limited internal memory. At the top of this key hierarchy is the Storage

Root Key (SRK), which never leaves the TPM. TC defines three roots of trust.

Root of Trust for Measurement (RTM) In TC, measuring is the process of computing the SHA-1 hash of an application binary before it is executed. Typically starting from an immutable part of the BIOS, a chain of trust is established where each component in the chain is measured before control is passed to it. The measurements are stored inside the TPM in memory regions called platform configuration registers (PCRs). As available memory in the TPM is limited, a special operation called TPM_Extend is used to write to PCRs:

$$PCR[i] = \text{SHA-1}(PCR[i] || \text{measurement}).$$

With the extend operation, the current PCR value is not overwritten, but the new measurement is accumulated with the current PCR value. TPM_Extend computes the hash of the current PCR value concatenated with the new measurement. This accumulated value is written back into the PCR.

Root of trust for reporting (RTR) Reporting of the platform status is called attestation and is done with the TPM_Quote command. As part of that, PCR values get signed inside the TPM using a key unique to that TPM. In theory, this key could be the endorsement key (EK), which is inserted into the TPM upon manufacturing. For privacy reasons, however, not directly the EK but alias keys are used. They are called Attestation Identity Keys (AIKs) and are generated with the help of an external trusted third party.

Root of trust for storage (RTS) The RTS allows the use of the TPM to securely store data. Binding of data refers to encrypting data with a TPM key and hence guaranteeing that these data only are accessible by this specific TPM instance. Sealing of data allows to specify a set of PCR values the data are associated with. As with binding, the unsealing can only be done by the specific TPM instance that holds the private sealing key. Additionally, the plaintext is only released if the current PCR values match those specified upon sealing.

5.2 Related work on user-based attestation

In this section, we briefly sketch related work on the design and implementation of user-based attestation systems. The primary goal of user-based attestation is to provide a mechanism where users can directly verify the state of a platform in an ad hoc manner. A major problem highlighted by Parno [31] is the absence of a reliable way to establish the identity of a TPM inside a computer. As a consequence, a malicious machine could forward TPM-related requests of a user to another TPM-enabled, unmodified machine, which then would provide valid response messages. This type of attack is called a *cuckoo attack*. The author argues that the establishment of the

TPM identity hence is a fundamental precondition for reliably attesting the software state of a platform. In conclusion, the work suggests adding a special-purpose hardware interface that allows an external device to directly communicate with a TPM.

For the purpose of trustworthy kiosk computing, Tögl and Hutter [43, 44] extend this idea and propose the integration of an Near Field Communication (NFC) interface into the TPM. Via the NFC interface, a user with a trusted, NFC-enabled handheld device can write a nonce into a dedicated register of the TPM. This nonce is then included in the subsequent TPM_Quote operation. The establishment of the nonce requires the user to bring the handheld into close proximity (a few centimeters) of the TPM. This ensures that the attestation response actually comes from the intended machine. As the NFC-based establishment of the nonce bypasses the software stack of the host machine, malicious software on the host cannot manipulate the attestation process.

With Seeing-Is-Believing (SIB) [26], McCune et al. take a different approach using visual communication to establish an authentic communication channel between mobile phones. Visual communication has the advantages that it is intuitive for users and that attacks on the communication are easily spotted. In this procedure, called demonstrative identification, a 2D barcode containing a key is displayed by one smartphone, which then is captured using the camera of the second phone. Performing this procedure also in the opposite direction allows to establish a mutually authenticated communication channel. In cases where one of the devices does not have a display, the authors propose to attach a sticker with the printed barcode to the system. This approach is also proposed by Garriss et al. [17] in their study targeted toward the realization of trustworthy and personalized computing environments on public kiosks. However, as discussed in [31, 43], this approach is problematic because stickers are easily modified or replaced and hence cannot help to reliably prevent cuckoo attacks. Bangerter et al. [2] also use the visual channel together with a dedicated, proprietary security token to attest the state of a system. Using this device, a logical and secure channel between the token and an attestation server is established. Messages from the server are sent to the token by flickering the screen of the attested system. The message encoded in this flickering is captured by the token's camera.

Other researches pursue similar ideas but use different communication techniques to establish a local, authentic channel. With Loud and Clear, Goodrich et al. [20] propose a system that uses audio communication for device pairing. In this system, authentication data is encoded in English phrases. It is the task of the user to compare these phrases played by the devices. The authors argue that one

advantage of the system is that it can operate over larger distances than, e.g., visual solutions. This however also makes the system more vulnerable to cuckoo attacks as identification of the talking device might not be as intuitive as with visual approaches.

5.3 Visual user-based attestation

As already mentioned, a main challenge of user-based attestation is the proper selection of the intended camera and the establishment of an authentic communication channel. To be feasible for average users, this process needs to be intuitive and largely automated. At the same time, it must be ensured that cuckoo attacks are properly prevented. Typically, cameras are not mounted in places easily reachable by users. Consequently, a dedicated hardware interface to the TPM is not an option. Similar considerations hold true for NFC communication. A more natural choice for a camera system is the visual channel. Existing approaches like SIB would allow the camera system to authenticate the user's handheld device. For authentication in the opposite direction, the camera would either need a display or a barcode sticker attached to it. Usually, there is little use for an extra display on a surveillance camera and barcode stickers are easily manipulated [17, 43]. For that reason, we present a different approach that still uses the visual channel for authentication of the camera but does not require a display or stickers on the camera. To achieve that, we make use of Trusted Computing technology to ensure certain system properties as part of the attestation process. Before describing the details of our approach in Sect. 5.3.3, we briefly discuss the system architecture, setup procedures as well as design assumptions we have made.

5.3.1 System architecture and setup

The system architecture extends the one shown in Fig. 3. In addition to the Trusted Computing-enabled cameras and the operator's control station (CS), two additional entities are introduced. The first one is a handheld device that is used by monitored persons to perform the attestation of a camera. This handheld could be any modern smartphone. The second new entity is a TrustCenter that generates a trust report based on the PCR values obtained from camera attestation.

To be able to generate such a report, the TrustCenter has to know the measurements of applications potentially running on a camera. To achieve that, we assume the cooperation of camera operators by disclosing the camera firmware, including the source code, to the TrustCenter. The TrustCenter can then review the applications and store the corresponding measurements together with a description of application properties in its database. We believe

that both, camera operators and users, can benefit from such a model. On the one hand, operators can demonstrate their commitment to openness and privacy protection while their intellectual property rights are protected, since source code is only disclosed to the TrustCenter and not the general public. On the other hand, users benefit from the system since they can learn what the cameras in their environment are doing and how they handle personal data.

As far as user-based attestation is concerned, a Trusted Computing-enabled camera has to undergo a number of setup steps before it is deployed. It is assumed that this setup is done when the camera is under full control of the operating personnel. The main part of the setup involves the generation of TPM keys on the camera and at the control station. All keys are generated as 2048 bit RSA keys. The following setup steps and the key generation are performed individually for each camera.

TPM ownership Calling the `TPM_TakeOwnership` operation of the camera's TPM_C sets an owner secret and generates the storage root key K_{SRK} . The owner secret is not required during normal operation of the camera and is set to a random value unique to every camera. For maintenance operations, the camera's owner secret is stored in the database of the control station.

Identity key creation An attestation identity key serves as an alias for the endorsement key (K_{EK}) and is used during platform attestation. Contrary to a PC system, one single AIK is sufficient since there are no humans who actively use the camera and whose privacy needs to be protected. The single attestation identity key K_{AIK} serves for platform attestation and certification of other TPM keys. In our concept, the TrustCenter also takes the additional role of a PrivacyCA [32]. As part of AIK creation, it issues an AIK certificate for $K_{AIK_{pub}}$. A copy of this AIK certificate $Cert_{AIK}$ is stored on the camera. From there, it is sent to the user as part of the attestation response. The TrustCenter is expected to maintain revocation lists for the issued AIK certificates and to provide a service that allows users to check the status of a certificate.

Signature key creation For signing data such as images delivered by the camera, a non-migratable signing key K_{SIG} is created with K_{SRK} as its parent. Being non-migratable ensures that the private key cannot leave the camera's TPM_C . This provides assurance that data signed with this particular key really originates from this specific camera.

Table 2 summarizes the cryptographic keys generated as part of the camera setup.

5.3.2 Assumptions

As a hardware-based security solution, Trusted Computing is intended to provide higher levels of security that a pure

Table 2 The cryptographic keys generated during setup of a single camera

Key Name	Abbreviation
Storage root key	K_{SRK}
Attestation identity key	K_{AIK}
Signature key	K_{SIG}

All keys are non-migratable, 2,048 bit RSA keys

software solution can achieve. It, however, was not designed to withstand sophisticated and expensive hardware attacks as performed by Tarnovsky [42]. Since our main concern in this work are software attacks, we do not address attacks on camera hardware. We however assume that many attacks can be made a lot more complicated when using specifically designed camera enclosures and circuit boards. Moreover, many hardware attacks involve a reboot of the system, which can be detected by operators via our trusted lifebeat [53].

The other major hardware component of our system is a handheld device that allows users to interact with the system and to perform the user-based attestation. In this study, we use the handheld as a tool but do not address security questions related to this device. We assume that the software on the handheld is trustworthy and that no malicious software components are installed. Related study that investigates the design of a trustworthy handheld device is presented by, e.g., Selhorst et al. [38]. Furthermore, we assume that the handheld is preloaded with the public key certificate of the TrustCenter.

5.3.3 Attestation protocol

Our user-based attestation protocol consists of two separate phases. The first phase, shown in Fig. 4, serves two purposes. First, the user selects the camera of interest via visual communication using a handheld device. Second, the camera status is attested and the attestation results are evaluated with the help of the external TrustCenter. Once the first phase of the attestation protocol is complete, the user knows whether or not there is a trustworthy camera. However, due to potential cuckoo attacks, it is not yet guaranteed that this trustworthy camera is the one selected by the user. This issue is addressed by the second phase of the attestation protocol which is depicted in Fig. 5. The design of the second phase is based on the knowledge that the trustworthy camera from phase one has some specific properties. In our case, a required property is the support of a special operation that can only be triggered via the visual channel. The user now requests this operation from the camera. Based on the outcome of the second attestation phase, the user learns if the trustworthy camera from phase

one is identical to the actually intended camera. Note that in a practical implementation, the two phases of the attestation protocol are executed directly one after another. Users will perceive the two attestation phases as one single process.

Subsequently, we go into the technical details of our user-based attestation protocol. The numbers of the steps in the following description correspond to the indices in Figs. 4 and 5.

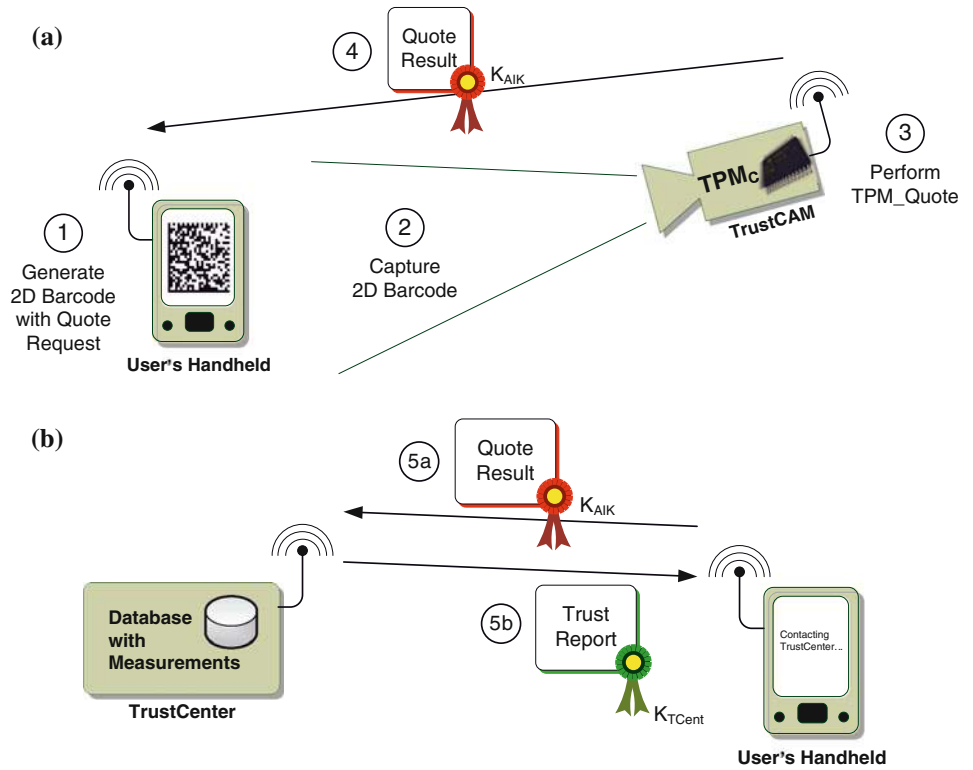
Attestation phase one The first phase depicted in Fig. 4, starts with the generation of a 2D barcode on the user's handheld device in step 1. This barcode contains a TPM_Quote request together with a randomly generated nonce N_1 to ensure freshness, the list of the PCRs to be quoted and the IP address of the handheld's wireless interface. Next, the user presents the barcode to the camera to be attested by pointing the handheld with the displayed barcode toward the camera. In step 2, the camera captures an image and extracts N_1 , the list of PCRs and the IP address from the barcode. In step 3, it then performs the TPM_Quote command using K_{AIK} as shown in Eq. 1.

$$Quote_{Res} = TPM_Quote_{K_{AIK}}(N_1, PCR_{List}). \quad (1)$$

Next, a wireless connection is established to the IP of the user's handheld and, in step 4, the signed quote result $Quote_{Res}$, the PCR measurement log PCR_{Log} and the AIK certificate $Cert_{AIK}$ are sent back to the handheld.

Using these data, the handheld has to perform the following two steps: (a) With the help of the TrustCenter, it has to be verified that $Cert_{AIK}$ has been issued for an AIK protected by a TPM that is part of a camera that belongs to the network. This also includes certificate revocation checks. (b) The signature of the quote result $Quote_{Res}$ has to be verified and the content of the quote blob has to be examined. This includes checking nonce N_1 as well as evaluating the provided PCR values together with the PCR measurement log PCR_{Log} . To offload work from the handheld, we submit the quote blob and the PCR log to the TrustCenter, which evaluates the blob in conjunction with the log. The individual PCR values are compared to the hashes of the firmware and the applications that have been submitted for review by the camera manufacturer or operator. The TrustCenter reconstructs the PCR log step by step and compares the reconstructed PCR values to those signed by the TPM. Once the validation of all attestation data is completed, the TrustCenter generates a report that includes descriptions and properties of the camera's firmware and the individual applications executed on the camera. This trust report is digitally signed and sent back to the user's handheld. On the handheld, integrity and authenticity of the trust report is verified using the preloaded TrustCenter public key certificate.

Fig. 4 The first phase of the user-based attestation protocol consists of vision-based camera selection, camera attestation and evaluation of the attestation results with the help of the external TrustCenter. **a** Using a handheld device and visual communication, a user selects a camera to be attested. Once attestation is completed, the camera sends the results back to the handheld via wireless communication. **b** To reduce complexity of the handheld application, an external TrustCenter is contacted for validation of the attestation results. The TrustCenter returns a signed trust report that includes descriptions and properties of the camera’s firmware as well as the applications executed on the camera



If all checks were successful, the user now has assurance that (a) the quote came from a camera that belongs to the network and (b) the camera is in a trustworthy state. The user however does not yet have the assurance that the quote actually came from the camera to which the 2D barcode was presented. This specific camera might have been subverted by an attacker. Instead of performing a local quote that would reveal this fact, the malicious software on the camera could grab an image, extract N_1 , PCR_{List} and IP and forward this data to an unmodified camera. This camera then responds with a valid quote result. This would lead the user to believe that the camera in front of her is in the reported, trustworthy state while it is actually running malicious software. This attack pattern is possible since we assume that Trusted Computing-enabled cameras allow remote attestation to be performed not only when triggered via the visual channel but also via the wireless channel. Wireless attestation is an important asset of operators to check camera status remotely, as we have demonstrated with our trusted lifebeat [53].

Attestation phase two As mentioned previously, we require a camera to have certain properties to be admitted to the second attestation phase. Specifically, we require the camera to support a *GrabAndSignImg* command, which reads an image from the camera sensor and digitally signs it using the TPM. Furthermore, we require that the *GrabAndSignImg* command can only be triggered via visual communication and the camera supports no wirelessly

accessible functionality that signs remotely provided input data. If these required properties are not confirmed by the trust report from phase one, the attestation process is canceled and the user is informed about the failure.

The primary purpose of the second attestation phase, as shown in Fig. 5, is to ensure that trustworthy camera from phase one actually is the one intended by the user. This phase starts with step 6 where a new 2D barcode is generated by the user’s handheld. The barcode encodes a *GrabAndSignImg* request, a new nonce N_2 and the IP address of the verifier’s handheld. This barcode is presented to the same camera as the first barcode was presented to. As part of the *GrabAndSignImg* function, in step 7 the camera reads an image from the sensor. The image, which contains the barcode with nonce N_2 , is signed in step 8 with the non-migratable TPM signing key K_{SIG} as shown in Eq. 2.

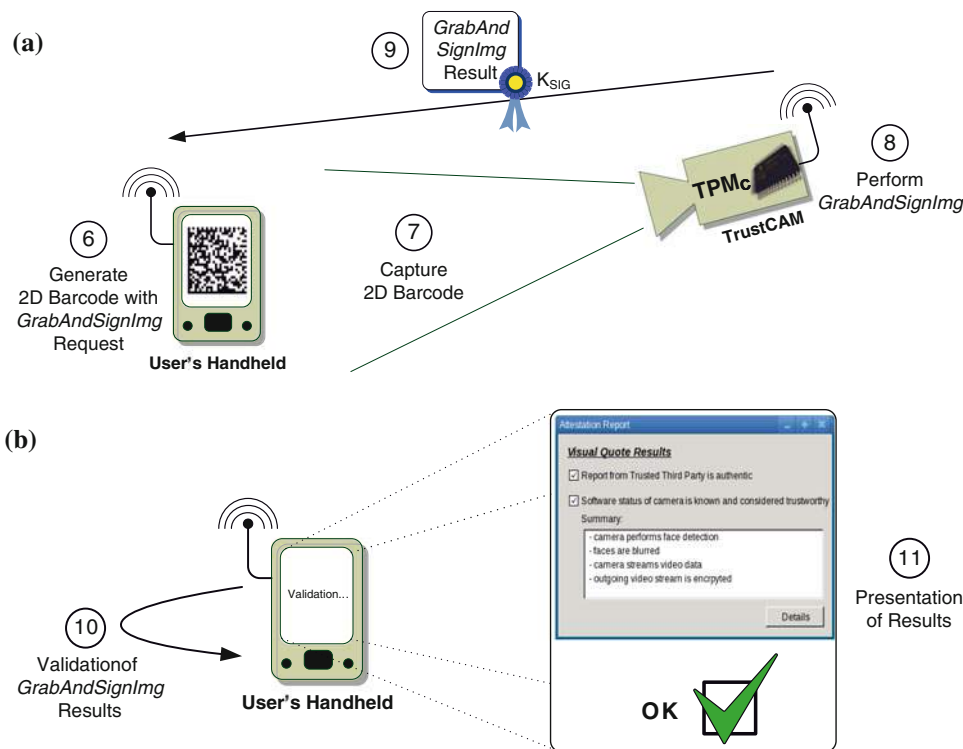
$$Sig_{img} = TPM_Sign_{K_{SIG}}(img). \tag{2}$$

Next, in Eq. 3, K_{SIG} is certified using K_{AIK} .

$$Cert_{SIG} = TPM_CertifyKey_{K_{AIK}}(K_{SIG_{pub}}, keyInfo). \tag{3}$$

The certificate $Cert_{SIG}$ consists of the signed hash of the public signing key $K_{SIG_{pub}}$ and the TPM_CERTIFY_INFO2 structure. It contains information about the key, including that it is, e.g., non-migratable. In step 9, the captured image img , the image signature Sig_{img} , the public signature key $K_{SIG_{pub}}$ and the certificate $Cert_{SIG}$ are sent back to the

Fig. 5 The second phase of the user-based attestation protocol only is entered if the first phase was successfully completed. It is designed to ensure that the trustworthy camera from phase one actually is the one intended by the user. **a** On the user’s handheld, a second 2D barcode is generated that triggers the *GrabAndSignImg* operation of the camera. **b** On the user’s handheld, the result of the *GrabAndSignImg* request is validated to ensure its freshness and that it comes from the trustworthy camera attested in phase one



handheld. In step 10, the application on the handheld has to perform the following three verification steps: (a) The image signature Sig_{img} has to be verified. (b) The certificate $Cert_{SIG}$ of K_{SIG} must be verified using the public AIK from $Cert_{AIK}$, which has also been used for quote validation in step 5. Assuming that the information in $TPM_CERTIFY_INFO2$ proves that K_{SIG} is non-migratable, it is ensured that the quote and the signed image come from the same camera. (c) From the barcode of the signed image, nonce N_2' has to be extracted and compared with N_2 . This ensures that the signed image was freshly captured by the camera intended by the user.

If these three steps were successful, the user knows that the quote in step 3 and the image signature in step 8 were performed by the same TPM and hence come from the same camera. Since the *GrabAndSignImg* request can only be triggered visually, it is assured that the trustworthy camera actually is the one in front of the user. Attacks on the visual communication channel between the user and the camera would be easy to spot. Cuckoo attacks where the grabbed image is forwarded wirelessly to the trustworthy camera from phase one are eliminated by the fact that the trustworthy camera does not support such functionality. This property has been assured by the trust report from the TrustCenter.

Finally, in step 11, the trust report including the properties and descriptions of the applications executed on the camera is presented to the user. The descriptions and properties should be formulated in a way that allows

average users to understand what the camera is doing and how privacy-sensitive data are managed. Ideally, the report should contain several levels of detail ranging from an abstracted trust decision to providing full details about the software running on the camera. This way, different levels of user knowledge and expectation can be satisfied.

6 Prototype implementation and evaluation

In this section, we discuss the prototypical implementation of selected system components and present corresponding evaluation results. Specifically, we concentrate on the realization of the direct-user feedback with our TrustCAM camera prototype [51].

The prototype shown in Fig. 6 is largely built from commercially available components. TrustCAM is based on the BeagleBoard [3], which has a dual-core processor with an ARM Cortex A8 CPU clocked at 480 MHz and a TMS320C64x+ digital signal processor running at 360 MHz. The system is equipped with 256 MB RAM and 256 MB NAND flash. Via USB, we connect a color SVGA CMOS sensor (Logitech QuickCam Pro 9000) and an RA-Link RA-2571 802.11b/g WiFi adapter. An XBee radio provides a second, low-performance communication channel. Finally, an Atmel AT97SC3203S—the only commercial TPM designed for embedded devices—is connected to the mainboard via the I2C bus.

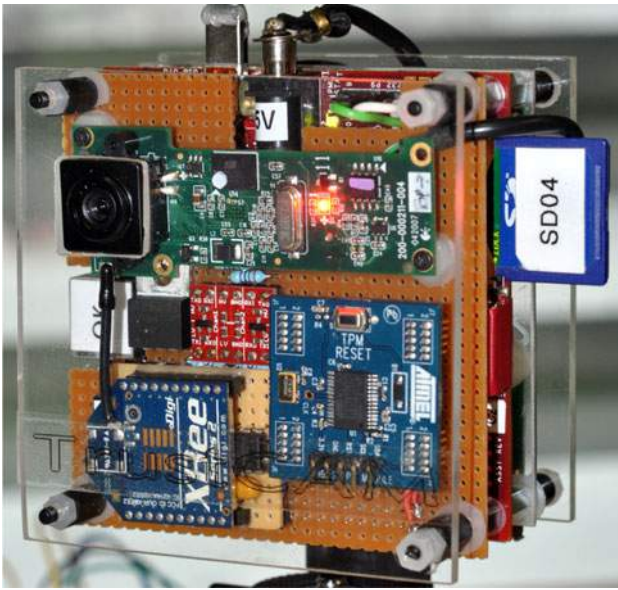


Fig. 6 The TrustCAM prototype system equipped with an Atmel I2C TPM chip

As operating system, we use an ARM Linux system together with a customized, OMAP specific kernel. To simplify the development of computer vision applications, we rely on a custom middleware system presented in [37]. This system allows composing applications from re-usable, independent components. To implement the chain of trust as shown in Fig. 7, we modified the camera’s bootloader [53]. Measurements of the bootloader, the Linux kernel and the root file system are stored in the PCRs of the TPM. Additionally, we measure the components of the executed computer vision applications. This approach allows keeping the number of measurements small, while at the same time full information about the executed vision applications is provided. For application level TPM access, we rely on the TrouSerS [21] TCG software stack with a modified trusted device driver library (TDDL). Our custom TDDL implementation manages the interaction with the Atmel I2C TPM.

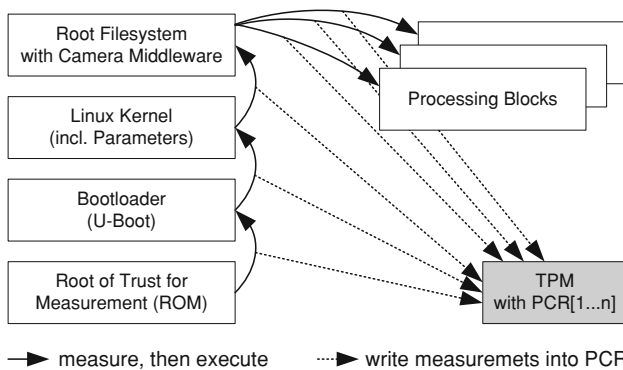


Fig. 7 The chain of trust as implemented on the TrustCAM prototype

We use a Samsung Galaxy S i9000 smartphone running Android 2.1 as handheld device for the attestation prototype. It is equipped with a 1-GHz ARM CPU, 512 MB RAM and a 4-inch screen with a resolution of 480x800 pixels. For 2D barcode generation, we use the open-source ZXing [56] library. The primary target of the library is the Java programming language with versions available for Java SE, ME as well as Android. Furthermore, a C++ port of the library exists, which makes it suitable for embedded systems without Java environment as well as Apple iOS-based devices.

The 2D barcodes generated on the handheld contains the request id (1 byte; *Quote* or *GrabAndSignImage*), the nonce N_x (20 bytes) and the IP address of the handheld (4 bytes). For the prototype, we do not send a list of PCRs but include all PCRs in the quote. The 25 bytes of the request are encoded in a QR tag with a size of 21×21 modules. Figure 8 shows a barcode example of the first attestation phase.

6.1 Evaluation and performance considerations

Our primary evaluation goal is to provide runtime measurements for the attestation process. Overall runtime has a high impact on user experience and should therefore be kept as short as possible. Additionally, we want to determine the practical feasibility of barcode detection. As part of that, we also investigate achievable distances between handheld and camera with devices with different screen sizes.

Attestation performance directly depends on the runtime for the individual protocol steps from Sect. 5.3.3. Therefore, we implemented the required functionality on our TrustCAM prototype. Table 3 summarizes the protocol components together with the measured runtimes. Barcode

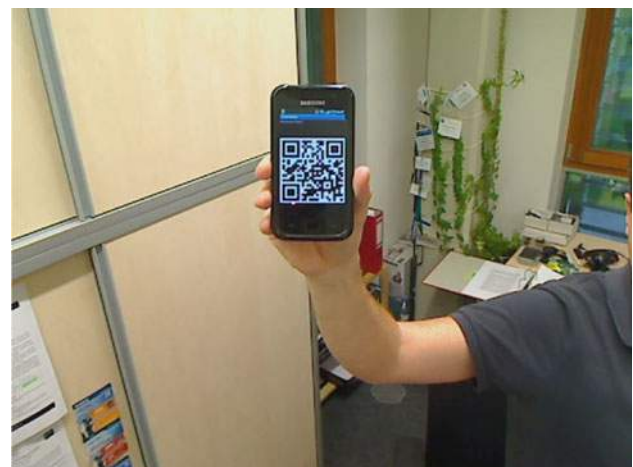


Fig. 8 A 2D barcode containing a visual attestation request as seen by the camera

Table 3 Runtime analysis for visual user-based attestation broken down to individual processing steps

Step	Device	Action	Runtime
1	Smartphone	2D barcode generation (<i>Quote</i> request)	41 ms
2	TrustCAM	Detection and decoding of 2D barcode	~ 135 ms
3	TrustCAM	Performing the <i>Quote</i> operation	
		TPM_OIAP	47 ms
		TPM_Quote	837 ms
		TSS Overheads	25 ms
4	TrustCAM	Transmission of <i>Quote</i> results	4 ms
5	Smartphone	<i>Quote</i> validation and interaction with TrustCenter	t.b.d
6	Smartphone	2D barcode generation (<i>GrabAndSignImage</i> request)	41 ms
7	TrustCAM	Detection and decoding of 2D barcode	~ 135 ms
8	TrustCAM	Performing <i>GrabAndSignImage</i>	
		SHA-1	2 ms
		TPM_OIAP (2x)	94 ms
		TPM_Sign	804 ms
		TPM_CertifyKey	849 ms
		TSS Overheads (2x)	50 ms
9	TrustCAM	Transmission of <i>GrabAndSignImage</i> results	28 ms
10	Smartphone	Evaluation of <i>GrabAndSignImage</i> results	82 ms
11	Smartphone	Visualization of trust report	12 ms
Total runtime			~ 3,186 ms

The step numbers correspond to those of Sect. 5.3 The runtimes are rounded averages over ten runs

generation on the smartphone (steps 1 and 6) on average takes 41 ms. Decoding of a barcode by the camera (steps 2 and 7) requires about 135 ms. We would like to emphasize that this runtime should not be considered as an upper limit. Barcode detection is influenced by many factors such as lighting conditions, viewing angle, distance between the handheld and camera, screen size of the handheld as well as the cameras optics and resolution. With our prototype setup we achieve reliable barcode detection for distances of up to 65 cm. The effective barcode size on the 4-inch screen of the Samsung smartphone is 5×5 cm. To evaluate achievable distances with handhelds with larger screens we used an Apple iPad. On the 9.8-inch screen of the tablet, the displayed 2D barcode has a size of 14.5×14.5 cm. This allows us to achieve reliable barcode detection for distances of up to 150 cm.

Execution of the quote operation on the TPM consists of command authorization (TPM_OIAP), the actual TPM_Quote command and overhead for command processing and serialization in the TSS. With the Atmel I2C TPM, this results in a total runtime for step 3 of 909 ms. It illustrates the fact that TPMs are primarily designed for low costs instead of high performance. As our evaluations have shown [53], TPMs of other manufacturers have slightly better performance, but are not available with interfaces for embedded systems. Subsequently, in step 4, the quote result is transmitted back to the smartphone where it is analyzed (step 5). In our concept, we rely on an

external TrustCenter for quote validation. For the prototype we use a rudimentary TrustCenter implementation and therefore cannot provide meaningful performance measurements. However, considering the details of this process as well as the performance of current 3G networks, we estimate it to take up to several seconds in a real implementation. We are going to revisit this issue at the end of this section where we discuss optimizations that can be applied to the attestation process.

After the second 2D barcode was generated and detected (steps 6 and 7), the camera performs the *GrabAndSignImage* operation (step 8). Runtime consists of computing the SHA1 hash of the image, execution of TPM_Sign and TPM_CertifyKey operations as well as command authorization (TPM_OIAP) and TSS overheads. These individual runtimes accumulate to 1,799 ms. The results of the *GrabAndSignImage* command are sent back to the smartphone (step 9, 28 ms) where they are analyzed (step 10, 82 ms) and finally visualized and displayed (step 11, 12 ms). Overall, this results in a runtime for visual attestation of ~3,186 ms. Note that this number considers all core components of the attestation process, but omits implementation specific overheads for, e.g., program loading or synchronization between processing blocks.

While we believe that the achieved attestation runtime already is acceptable, it can be optimized further. This is done by parallelizing certain steps of the protocol as shown in Fig. 9a. Specifically, the creation of the second barcode

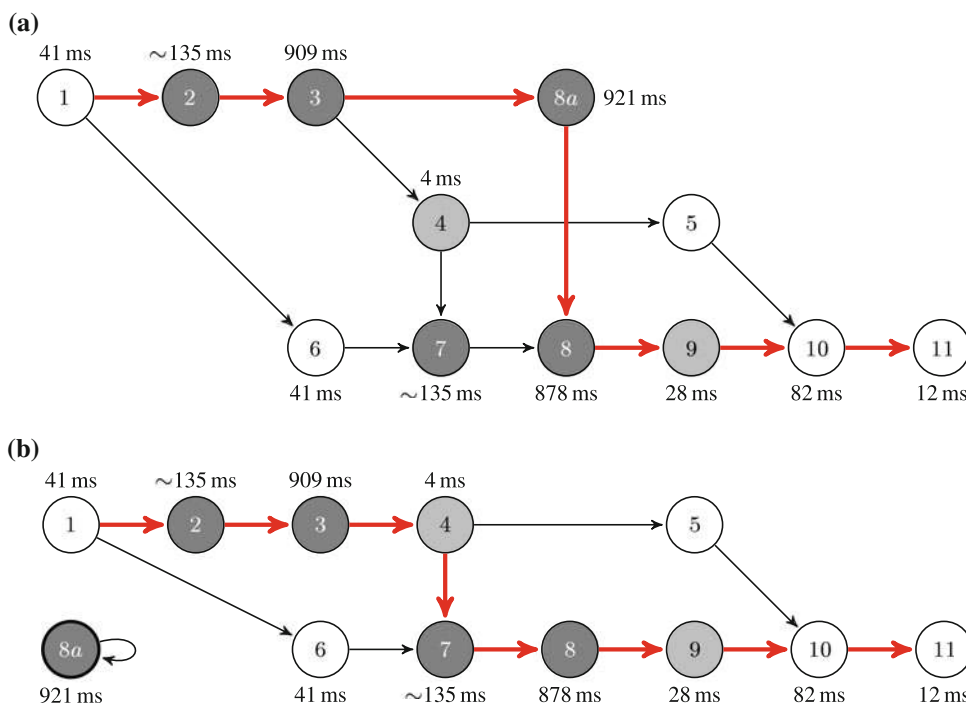


Fig. 9 Runtime optimizations of visual user-based attestation by parallelization of processing steps. *White circles* correspond to processing steps on the handheld, *dark gray circles* to steps on the camera and *light gray circles* to steps involving both parties (e.g., data transmission). The *numbers* in the *circles* correspond to the protocol description in Sect. 5.3.3 and the runtime analysis in Table 3. Runtimes are printed above or below the *circles*. No fixed runtime is assigned to step 5, which represents the data exchange with the external TrustCenter. Thick, *red arrows* denote the critical path for the achievable runtime. **a** The creation of the second barcode (step 6)

(step 6) can already be started after the first barcode was generated and displayed. Furthermore, we introduce a new processing step 8a, which represents the certification of the signing key K_{SIG} that is used for the subsequent *GrabAndSignImage* request. Step 8a consists of TPM_Certify-Key, TPM_OIAP and TSS overheads. It has an accumulated runtime of 921 ms. As shown in Fig. 9a, the certification step 8a can be started immediately after the quote command is completed.

Another opportunity for introducing parallelism is the interaction with the TrustCenter. Once the quote result was submitted to the TrustCenter in step 5, it was not strictly necessary to wait for the results. The smartphone can continue with the second phase of the attestation protocol. However, the TrustCenter report is required before the results of the *GrabAndSignImage* can be evaluated in step 10. No later than at this point, the trust report is needed to determine if the camera is trustworthy and implements a genuine version of the *GrabAndSignImage* command. If it turns out that the camera is not trustworthy, steps 6–9 have

can be immediately started once the first barcode creation is completed. Step 8a is the certification of the signing key K_{SIG} using K_{AIK} . It can be started directly after the *Quote* result was transmitted (step 4). Runtime of the critical path is shortened to 3,006 ms. **b** Step 8a is the certification of the signing key K_{SIG} using K_{AIK} . This certification does not have to be done as part of the attestation, but can be done previously during idle periods. The resulting certificate can be stored on the camera. Runtime of the critical path is shortened to 2,224 ms

been performed needlessly. If, however, the camera is reported to be trustworthy, and the overall attestation runtime and the time the handheld has to be pointed toward the camera are significantly reduced.

Considering the runtimes between steps 4 and 10, the TrustCenter has more than 1.8 s to generate and return the trust report before step 10 is reached. As illustrated in Fig. 9a, parallelization reduces the critical path of the attestation process to 3,006 ms. At first, this might seem like a marginal improvement compared to the 3,186 s from the fully sequential approach from Table 3. The picture changes, when recalling that the time required for interaction with the TrustCenter was not included in the sequential approach. The time it takes the TrustCenter to generate and return the report must be added to the 3,186 ms of the sequential version, while the time window of 1.8 s is already included in the 3,006 ms of the parallelized variant.

If taking into account that the certification of the signing key used in *GrabAndSignImage* is independent of the

actual signing process, it can be already performed in advance, e.g., during TPM idle periods. As shown in Fig. 9b, this allows a reduction of the attestation runtime to 2,224 ms. Note that this also means that the time the TrustCenter has to generate and return its report is reduced to little more than 1 s. While this might not be sufficient for the TrustCenter to generate the report, it still is an important improvement since it reduces the time the handheld has to be pointed toward the camera.

6.2 Presentation of trust reports

We expect that users of the system have very different backgrounds and knowledge about camera systems, privacy protection and security. Therefore, it is difficult to find the right tradeoff between comprehensibility and the amount and level of detail. To overcome this problem, we suggest providing different abstraction levels and allow users to choose the form that is most suitable for them. The different levels are based on the information contained in the report of the TrustCenter. For the prototype, we designed two different forms of how the results are presented. The first one is shown in Fig. 10a and contains information at a high level that requires very little technical background. This includes whether the camera's software is known by the TrustCenter and if it is considered trustworthy. Furthermore, information is provided about the camera's purpose and its owner. Finally, the report includes privacy-relevant properties of the software running on the camera. The second view we implemented is

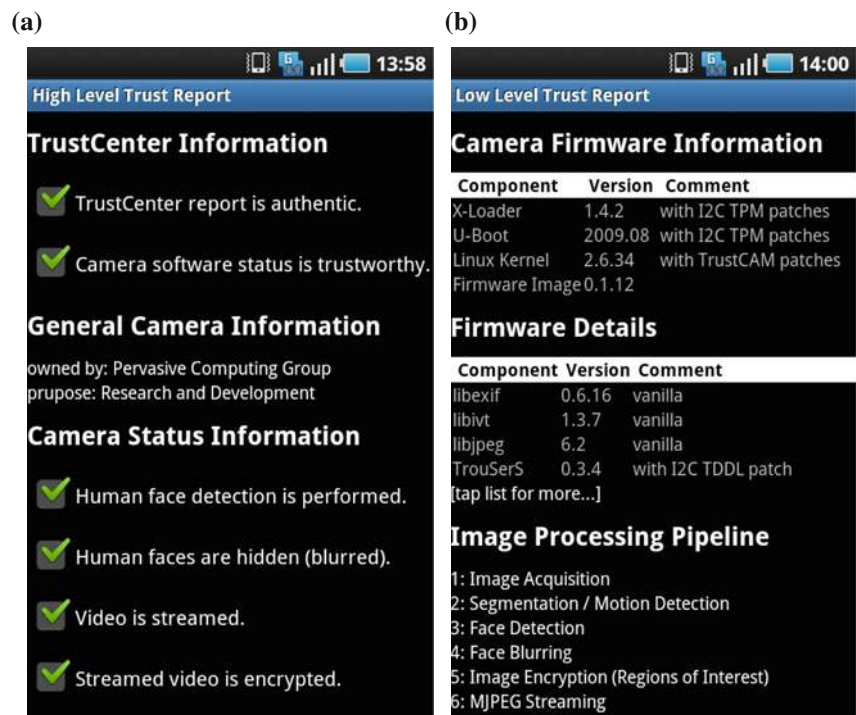
shown in Fig. 10b. Contrary to the first one, it provides full details of the camera's software configuration. To demonstrate the wide range of possible presentation styles, we intentionally chose those two very different views. Clearly, any abstraction level in between is possible.

6.3 Discussion

Evaluation results of our prototype are very encouraging because they illustrate the practical feasibility of our approach. The time users have to hold their phone steadily and directed toward the camera is acceptable. User experience could be increased further by giving intermediate status feedback. Since users cannot see the display of the phone, acoustic feedback could be used to indicate the attestation progress.

A crucial assumption of our concept is that the camera operator is honest and does not use intentionally modified cameras. It would be relatively easy to include two processing boards into the camera housing. For user-based attestation, only the trustworthy system is used while the other one servers other tasks which remain unnoticed by the user. One way to eliminate this kind of setup would be to require an additional certification of camera systems by some other party than the camera operator or manufacturer. This could, e.g., be some government agency. But ultimately, there has to be some entity that can be trusted. This is not much different from other, established security concepts such as SSL certificates used to protect https connections.

Fig. 10 Screenshots of different variants of how TrustCenter reports can be presented. **a** A high-level camera status report. **b** Full camera details including names and versions of relevant software components



The achieved distances for barcode detection are relatively short, which might be a problem with cameras that, e.g., are mounted very high. For the prototype, we directly perform barcode detection for unmodified, captured images. We assume that performance could be increased by enhancing and preprocessing the input images. Performance of visual tagging systems continues to evolve and novel systems such as Bokode [27] are reported to work for distances of up to several meters. On the other hand, the relatively low detection distance reduced the risk that barcodes are captured by another camera. A potential attacker might try to read the barcode from a distance using a camera with a high-quality zoom lens. As a protection against this type of attack, we suggest that the phone's screen is equipped with a privacy protection filter. These filters are frequently used on business laptops to restrict the viewing angle such that the displayed content is only visible when looking straight onto the screen.

Another type of attack that must be considered is denial of service (DoS) attacks. A simple and cheap form of attack would be to print a 2D barcode with an attestation request on a sheet of paper and attach it, e.g., to a wall in the field of view of the camera. The camera would then continuously detect and decode the barcode in every captured frame. To prevent successive executions of the TPM_Quote command, the camera could keep recently seen nonces N_1 in a cache and only execute the quote command if the nonce is fresh. Nevertheless, the pointless effort of detecting the same barcode over and over again remains. Considering the runtime for barcode detection, this can easily overload the camera and render it useless for its actual surveillance tasks. To limit this risk, we suggest not to check every captured frame for a barcode, but to perform barcode detection only at a predefined interval (e.g., every 5 s). If the same barcode was read consecutively, this interval could be increased.

7 Conclusions and future work

In this work, we presented a concept for user-centric privacy awareness in video surveillance. The proposed system follows a community-based approach and empowers monitored persons to actively participate in registering cameras. The collected information is used to warn users of violations of their personal privacy policy. Users do not need specific devices, but all functionality can be implemented on conventional smartphones. Moreover, the system can be launched with support for basic privacy awareness levels and is easily extendable to more advanced levels once the required, trustworthy camera systems become available.

The primary focus of this study is on the realization of the highest privacy protection level. We have presented a system architecture and in-depth description for a direct user feedback mechanism. Based on established security technology, users gain insight into applications running on a camera. With the help of the TrustCenter, behavioral information and human comprehensible descriptions can be obtained. We presented an extensive performance analysis based on a prototype implementation. The results illustrate the practical feasibility of the approach using available technology.

In a future study, we intend to investigate several aspects including a more complete TrustCenter implementation, a security analysis of the proposed protocols and extended evaluations of barcode detection performance under various conditions (e.g., indoor vs. outdoor). Finally, a user study could help to determine how, in what form and at what detail TrustCenter reports should be presented to maximize user benefit.

References

1. Baaziz, N., Lolo, N., Padilla, O., Petngang, F.: Security and privacy protection for automated video surveillance. In: Proceedings of the International Symposium on Signal Processing and Information Technology, pp. 17–22 (2007)
2. Bangarter, E., Djakov, M., Sadeghi, A.R.: A demonstrative ad hoc attestation system. In: Proceedings of the International Conference on Information Security, pp. 17–30 (2008)
3. BeagleBoard: TI OMAP3530 based Embedded System. <http://beagleboard.org>. Accessed: April 2011
4. Boulton, T.E.: PICO: Privacy through invertible cryptographic obscuration. In: Proceedings of the Workshop on Computer Vision for Interactive and Intelligent Environments, p. 12 (2005)
5. Brassil, J.: Using mobile communications to assert privacy from video surveillance. In: Proceedings of the Parallel and Distributed Processing Symposium, p. 8 (2005)
6. Cavallaro, A.: Adding privacy constraints to video-based applications. In: Proceedings of the European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology, p. 8 (2004)
7. Cavallaro, A.: Privacy in video surveillance. *IEEE Signal Process Mag* 24(2), 168–169 (2007)
8. Chattopadhyay, A., Boulton, T.E.: PrivacyCam: a privacy preserving camera using uCLinux on the Blackfin DSP. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, p. 8 (2007)
9. Cheung, S.C.S., Paruchuri, J.K., Nguyen, T.P.: Managing privacy data in pervasive camera networks. In: Proceedings of the International Conference on Image Processing, p. 4 (2008)
10. Cheung, S.C.S., Zhao, J., Venkatesh, M.V.: Efficient object-based video inpainting. In: Proceedings of the International Conference on Image Processing, pp. 705–708 (2006)
11. Chinomi, K., Nitta, N., Ito, Y., Babaguchi, N.: PriSurv: Privacy protected video surveillance system using adaptive visual abstraction. In: Proceedings of the International Multimedia Modeling Conference, p. 144 (2008)

12. Dufaux, F., Ebrahimi, T.: Scrambling for video surveillance with privacy. In: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop, pp. 160–166 (2006)
13. Dufaux, F., Ebrahimi, T.: A framework for the validation of privacy protection solutions in video surveillance. In: Proceedings of the International Conference on Multimedia and Expo, pp. 66–71 (2010)
14. Fidaleo, D.A., Nguyen, H.A., Trivedi, M.: The networked sensor tapestry (NeST): a privacy enhanced software architecture for interactive analysis of data in video-sensor networks. In: Proceedings of the International Workshop on Video Surveillance and Sensor Networks, pp. 46–53 (2004)
15. Fleck, S., Straßer, W.: Towards secure and privacy sensitive surveillance. In: Proceedings of the International Conference on Distributed Smart Cameras, p. 7 (2010)
16. Fleck, S., Strasser, W.: Smart camera based monitoring system and its application to assisted living. *Proc IEEE* 96(10), 1698–1714 (2008)
17. Garriss, S., Cáceres, R., Berger, S., Sailer, R., van Doorn, L., Zhang, X.: Trustworthy and personalized computing on public kiosks. In: Proceedings of the International Conference on Mobile Systems, Applications, and Services, pp. 199–210 (2008)
18. Gehring, S., Gisch, M.: The privacy badge revisited—enhancement of a privacy-awareness user interface for small devices. In: Proceedings of the Workshop on Security and Privacy Issues in Mobile Phone Use, p. 8 (2008)
19. Gisch, M., Luca, A.D., Blanchebarbe, M.: The privacy badge: a privacy-awareness user interface for small devices. In: Proceedings of the International Conference on Mobile Technology, Applications, and Systems and the International Symposium on Computer Human Interaction in Mobile Technology, pp. 583–586 (2007)
20. Goodrich, M.T., Sirivianos, M., Solis, J., Tsudik, G., Uzun, E.: Loud and clear: human-verifiable authentication based on audio. In: Proceedings of the International Conference on Distributed Computing Systems, p. 10 (2006)
21. IBM: TrouSerS TCG Software Stack. <http://trousers.sourceforge.net>. Accessed: April 2011
22. Langheinrich, M.: A privacy awareness system for ubiquitous computing environments. In: Proceedings of the International Conference on Ubiquitous Computing, pp. 315–320 (2002)
23. Luo, Y., Ye, S., Samson Cheung, S.c.: Anonymous subject identification in privacy-aware video surveillance. In: Proceedings of the International Conference on Multimedia and Expo, pp. 83–88 (2010)
24. Mapcams.org: Mapcams.org Website. <http://www.mapcams.org/>. Accessed: September 2010
25. Martin, A.: The ten page introduction to trusted computing. Tech. Rep. RR-08-11, Oxford University Computing Laboratory (2008)
26. McCune, J.M., Perrig, A., Reiter, M.K.: Seeing-is-believing: using camera phones for human-verifiable authentication. *Int. J. Secur. Netw.* 4(1/2), 43–56 (2009)
27. Mohan, A., Woo, G., Hiura, S., Smithwick, Q., Raskar, R.: Bokode: imperceptible visual tags for camera based interaction from a distance. In: Proceedings of the International Conference on Computer Graphics and Interactive Techniques, pp. 1–8 (2009)
28. Moncrieff, S., Venkatesh, S., West, G.: Dynamic privacy in public surveillance. *IEEE Comput* 42(9), 22–28 (2009)
29. OpenStreetMap.org: OpenStreetMap.org Video Surveillance Camera Overlay. <http://osm.vdska.de>. Accessed: April 2011
30. OWNI: CCTV Camera Positions in Paris. <http://app.owni.fr/camera-paris/> (2011). Accessed: April 2011
31. Parno, B.: Bootstrapping trust in a “trusted” platform. In: Proceedings of the Usenix Workshop on Hot Topics in Security, p. 6 (2008)
32. Pirker, M., Tögl, R., Hein, D., Danner, P.: A PrivacyCA for anonymity and trust. In: Proceedings of the International Conference on Trusted Computing, pp. 101–119 (2009)
33. Prefecture de Police: Plan de Vidéoprotection pour Paris. <http://www.prefecturedepolice.interieur.gouv.fr/Prevention/Videoprotection/Plan-de-vidioprotection-pour-Paris> (2010). Accessed: April 2011
34. Rahman, S.M.M., Hossain, M.A., Mouftah, H., Saddik, A.E., Okamoto, E.: A real-time privacy-sensitive data hiding approach based on chaos cryptography. In: Proceedings of the International Conference on Multimedia and Expo, pp. 72–77 (2010)
35. Saini, M., Atrey, P.K., Mehrotra, S., Emmanuel, S., Kankanhalli, M.: Privacy modeling for video data publication. In: Proceedings of the International Conference on Multimedia and Expo, pp. 60–65 (2010)
36. Schiff, J., Meingast, M., Mulligan, D.K., Sastry, S., Goldberg, K.Y.: Respectful cameras: selecting visual markers in real-time to address privacy concerns. In: Proceedings of the International Conference on Intelligent Robots and Systems, pp. 971–978 (2007)
37. Schriebl, W., Winkler, T., Starzacher, A., Rinner, B.: A pervasive smart camera network architecture applied for multi-camera object classification. In: Proceedings of the International Conference on Distributed Smart Cameras, p. 8 (2009)
38. Selhorst, M., Stüble, C., Feldmann, F., Gnaida, U.: Towards a trusted mobile desktop. In: Proceedings of the International Conference on Trusted Computing, pp. 78–94 (2010)
39. Senior, A., Pankanti, S., Hampapur, A., Brown, L., Tian, Y.L., Ekin, A., Connell, J., Shu, C.F., Lu, M.: Enabling video privacy through computer vision. *IEEE Secur. Priv. Mag.* 3(3), 50–57 (2005)
40. Spindler, T., Wartmann, C., Hovestadt, L., Roth, D., van Gool, L., Steffen, A.: Privacy in Video Surveilled Areas. In: Proceedings of the International Conference on Privacy, Security and Trust, p. 10 (2006)
41. Tansuriyavong, S., Hanaki, S.: Privacy Protection by concealing Persons in circumstantial Video Image. In: Proceedings of the Workshop on Perceptive User Interfaces, pp. 1–4 (2001)
42. Tarnovsky, C.: Hacking the smartcard chip. In: Black Hat DC Presentations (2010)
43. Tögl, R.: Tagging the turtle: local attestation for kiosk computing. In: Proceedings of the International Conference on Advances in Information Security and Assurance, pp. 60–69 (2009)
44. Tögl, R., Hutter, M.: An approach to introducing locality in remote attestation using near field communications. *J Supercomput* 55, 1–21 (2010)
45. Troncoso-Pastoriza, J.R., Pérez-Freire, L., Pérez-gonzález, F.: Videosurveillance and privacy: covering the two sides of the mirror with DRM. In: Proceedings of the Workshop on Digital Rights Management, pp. 83–94 (2009)
46. Trusted Computing Group: TCG Website. <https://www.trustedcomputinggroup.org>. Accessed: November 2010
47. Trusted Computing Group: TCG Software Stack Specification (TSS) Version 1.2, Level 1, Errata A (2007)
48. Trusted Computing Group: TPM Main Specification Version 1.2, Level 2, Revision 103 (2007)
49. Wickramasuriya, J., Datt, M., Mehrotra, S., Venkatasubramanian, N.: Privacy protecting data collection in media spaces. In: Proceedings of the International Conference on Multimedia, pp. 48–55 (2004)
50. Winkler, T., Rinner, B.: A systematic approach towards user-centric privacy and security for smart camera networks. In: Proceedings of the International Conference on Distributed Smart Cameras, p. 8 (2010)
51. Winkler, T., Rinner, B.: TrustCAM: security and privacy-protection for an embedded smart camera based on trusted computing. In: Proceedings of the International Conference on

- Advanced Video and Signal-Based Surveillance, pp. 593–600 (2010)
52. Winkler, T., Rinner, B.: User-based attestation for trustworthy visual sensor networks. In: Proceedings of the Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, p. 8 (2010)
 53. Winkler, T., Rinner, B.: Securing embedded smart cameras with trusted computing. *EURASIP J. Wirel. Commun. Netw.* 2011, 20 (2011) (Special Issue on Security and Resilience for Smart Devices and Applications)
 54. Yabuta, K., Kitazawa, H., Tanaka, T.: A new concept of security camera monitoring with privacy protection by masking moving objects. In: Proceedings of the Pacific-Rim Conference on Multimedia, p. 12 (2005)
 55. Ye, S., Luo, Y., Zhao, J., Cheung, S.C.S.: Anonymous biometric access control. *EURASIP J. Inf. Secur.* 2009, 18 (2009). (Special Issue on Enhancing Privacy Protection in Multimedia Systems)
 56. ZXing Community: Zebra Crossing (ZXing) Multi-format 1D/2D Barcode Image Processing Library. web: <http://code.google.com/p/zxing>. Accessed: November 2010

Copyright of Multimedia Systems is the property of Springer Science & Business Media B.V. and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.