# Privacy and Security in Video Surveillance

Thomas Winkler and Bernhard Rinner

**Abstract** Video surveillance systems are usually installed to increase the safety and security of people or property in the monitored areas. Typical threat scenarios are robbery, vandalism, shoplifting or terrorism. Other application scenarios are more intimate and private such as home monitoring or assisted living. For a long time it was accepted that the potential benefits of video surveillance go hand in hand with a loss of personal privacy. However, with the on-board processing capabilities of modern embedded systems it becomes possible to compensate this privacy loss by making security and privacy protection inherent features of video surveillance cameras. In the first part of this chapter we motivate the need for the integration of security and privacy features, we discuss fundamental requirements and provide a comprehensive review of the state of the art. The second part presents the TrustCAM prototype system where a dedicated hardware security module is integrated into a camera system to achieve a high level of security. The chapter is concluded by a summary of open research issues and an outlook to future trends.

## 1 The Need for Security and Privacy Protection

Reasons for deploying video surveillance systems are manifold. Frequently mentioned arguments are ensuring public safety, preventing vandalism and crime as well as investigating criminal offenses [40]. As part of that, cameras are often installed in public environments such as underground or train stations, in buses [39] or taxis [20], along roads and highways [8, 23], in sports stadiums or in shopping

Thomas Winkler
Alpen-Adria Universität Klagenfurt, Institute of Networked and Embedded Systems, Lakeside Park B02b, 9020 Klagenfurt, e-mail: thomas.winkler@aau.at

Bernhard Rinner
Alpen-Adria Universität Klagenfurt, Institute of Networked and Embedded Systems, Lakeside Park B02b, 9020 Klagenfurt, e-mail: bernhard.rinner@aau.at

malls [30, 31]. But video surveillance is no longer deployed only in public but also in private and more intimate environments. For example, in assisted living applications [10, 25, 62] cameras are used to monitor the interior of people's homes to detect unusual behavior of residents.

A major driving factor for this widespread deployment of cameras is that video surveillance equipment has become increasingly cheap and simple to use. As part of this development, today's video surveillance systems are no longer the closed, single-purpose systems they used to be. Modern systems are highly flexible which is primarily achieved via software. Camera devices usually come with powerful operating systems such as Linux as well as a variety of software libraries and applications running on top of it. Furthermore, these systems frequently make use of wireless network interfaces and are part of larger, often public, networks such as the Internet. The increasing size of the software stack and the relative openness of the network infrastructure turn many of today's video surveillance systems into attractive targets for both casual as well as professional attackers.

With the performance of modern embedded camera systems it is possible to make privacy protection an inherent feature of a surveillance camera. Sensitive data can be protected by various approaches including blanking, obfuscation or encryption. On-camera privacy protection is a clear advantage over server-side protection since it eliminates many potential attack scenarios during data transmission. When considering the software stack of an embedded camera system, privacy-protection is typically implemented at the application level. As a consequence, it is important to detect and avoid manipulations of the underlying software components such as the operating system or system libraries. Otherwise, an attacker might be able to manipulate the system and get access to sensitive data before privacy protection is applied. Depending on the application context, security guarantees such as integrity and authenticity are not only relevant for the system's software stack but also for delivered data. This is especially true for enforcement applications where captured images might serve as evidence at court.

## *1.1 Security and Privacy Requirements*

This section discusses the main security requirements for video surveillance applications. Making a camera system more secure not only offers benefits for camera operators. It is of equal importance for monitored persons. While this is obvious for aspects such as confidentiality, this also holds for, e.g., integrity of video data. If integrity is not protected, an attacker could modify video data in a way that intentionally damages the reputation of persons. The integration of the following basic security functionality is also a fundamental requirement for the design of high-level privacy protection techniques.

Integrity. Image data coming from a camera can be intentionally modified by an attacker during transmission or when stored in a database. Using checksums, digital signatures and watermarks, data integrity can be ensured. An often overlooked issue is that integrity protection is not only important for single frames but also for sequences. Simple re-ordering of images can substantially change the meaning of a video.

Authenticity. In many applications such as traffic monitoring and law enforcement, the origin of information is important. In visual surveillance, this is equivalent to knowing the identity of the camera that captured a video stream. This can be achieved by explicitly authenticating the cameras of a network and embedding this information into the video streams.

Freshness and Timestamping. To prevent replay attacks where recorded videos are injected into the network to replace the live video stream, freshness of image data must be guaranteed. Even more importantly, in many areas such as enforcement applications, evidence is required when a video sequence was recorded. Explicit timestamping of images not only answers the question when an image was taken, but at the same time also satisfies the requirement for image freshness guarantees.

Confidentiality. It must be ensured that no third party can eavesdrop on sensitive information that is exchanged between cameras or sent from the cameras to a monitoring station. Confidentiality must not only be provided for image and video data transmitted over the network but also for videos that, e.g., are stored on a camera to be transmitted at a later point in time. A common approach to ensure confidentiality is data encryption.

Privacy. In video surveillance, privacy can be defined as a subset of confidentiality. While confidentiality denotes the protection of all data against access by third parties, privacy means the protection of data against legitimate users of the system. For example, a security guard needs access to video data as part of her/his job. However, the identities of monitored persons are not required to identify unusual behavior. Privacy protection therefore can be interpreted as providing limited information to insiders while withholding sensitive, identity-revealing data.

Access Authorization. Access to confidential image data must be limited to persons with adequate security clearance. For access to highly sensitive data, involvement of more than one operator should be required to prevent misuse. If a video stream contains different levels of information (e.g., full video, annotations, ...), access should be managed separately for each level. Finally, all attempts to access confidential data should be logged.

Availability. A camera network should provide certain guarantees about availability of system services under various conditions. Specifically, reasonable resistance against denial of service attacks should be provided.

Clearly, these security properties are partially interdependent. It is, for example, meaningless to provide data confidentiality without implementing appropriate authorization mechanisms for accessing confidential data.

## 2 State of the Art

This section first presents an overview of the state of the art on security in video surveillance (Section 2.1). It is followed by a discussion of approaches towards privacy protection (Section 2.2). Section 2.3 summarizes our observations and outlines open issues for future research.

### *2.1 Video Surveillance Security*

Serpanos and Papalambrou [52] provide an extensive introduction to security issues in the domain of smart cameras. They discuss the need for confidentiality, integrity, freshness and authenticity for data exchanged between cameras. The authors acknowledge that embedded systems might not have sufficient computing power to protect all data using cryptography. In such a situation, they propose concentrating on protecting the most important data. This work also recognizes the partial overlap of confidentiality and privacy protection and emphasizes the importance of data protection not only against external attackers but also against legitimate system operators.

Senior et al. [51] discuss critical aspects of a secure surveillance system including what data is 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 for how long it is stored. Data confidentiality is ensured via encrypted communication channels. Privacy protection is addressed by re-rendering sensitive image regions. The resulting, multiple video streams contain different levels of data abstraction and are separately encrypted.

Schaffer and Schartner [49] present a distributed approach to ensure confidentiality in a video surveillance system. They propose that the video stream is encrypted using a hybrid cryptosystem. Encryption is performed for full video frames without differentiating between sensitive and non-sensitive image regions. A single system operator is not able to decrypt a video but multiple operators have to cooperate. This property is achieved by the fact that every operator is in possession of only a part of the decryption key.

Integrity protection of image and video data is an important security aspect. It can be addressed by means of, e.g., hash functions together with digital signatures or by embedding watermarks into the video content. An important design decision is whether the integrity protection technique is tolerant towards certain, acceptable image modifications or not. The work of Friedman [27] aims at “restoring credibility of photographic images” and therefore does not accept any image modifications. Specifically, authenticity and integrity of images taken with a digital still image camera should be ensured. This is achieved by extending the camera’s embedded microprocessor with a unique, private signature key. This key is used to sign images before they are stored on mass storage. The public key required for verification is assumed to be made available by the camera manufacturer. Friedman suggests that the software required for signature verification should be made publicly available. This work can be seen as one of the earliest approaches towards a trustworthy, digital camera system.

Qusquater [43] et al. propose an approach for integrity protection and authentication for digital video stored on tape in the DV format. They use SHA-1 to compute the hash of the image. To be less sensitive to transmission or tape errors, the authors suggest that the images are divided into blocks that are hashed separately. Authenticity is ensured by signing the hash values. The hash of the previous image is also included in the signature to maintain correct ordering of video frames.

Atrey et al. [2, 3] present a concept to verify the integrity of video data. In their work, they differentiate between actual tampering and benign image modifications. In this context, operations that do not change the video semantically such as image enhancements or compression are defined as acceptable. Tampering of video data is divided into spatial and temporal modifications. Spatial tampering includes content cropping as well as removal or addition of information. Temporal tampering refers to dropping or reordering of frames which might result from, e.g., network congestion. The authors argue that temporal tampering is acceptable as long as the semantic meaning of the video is not substantially affected. The proposed algorithm is based on a configurable, hierarchical secret sharing approach. It is shown to be tolerant to benign image modifications while tampering is detected.

He et al. [29] also discuss the design of a video data integrity and authenticity protection system. In contrast to other approaches, they do not operate on frames but on objects. Objects are separated from the video background using segmentation techniques. An advantage of this approach is that network bandwidth can be saved by transmitting primarily object data while background data is updated less frequently. Similar to Atrey et al. [2, 3], the authors require their integrity protection system to tolerate certain modifications such as scaling, translation or rotation. Considering these requirements, appropriate features are extracted from the detected objects as well as the background. A hash of these features together with error correction codes is embedded into the video stream as a digital watermark.

Digital watermarks are a popular technique to secure digital media content. A watermark is a signal that is embedded into digital data that can later be detected, extracted and analyzed by a verifier. According to Memon and Wong [36], a watermark can serve different purposes. This can be proof of ownership where a private key is used to generate the watermark. Other applications are authentication and integrity protection, usage control and content protection. Depending on the application domain, watermarks can be visible or invisible. When used for integrity protection, watermarks have the advantage that they can be designed such that they are robust against certain image modifications such as scaling or compression [1, 5]. An example where watermarking is used as part of a digital rights management system for a secure, embedded camera is presented by Mohanty [37]. He describes a secure digital camera system that is able to provide integrity, authenticity and ownership guarantees for digital video content. This is achieved using a combination of watermarking and encryption techniques. Due to the high computational effort, a custom hardware prototype based on an FPGA is used to meet the realtime requirements.

## 2.2 Privacy Protection in Video Surveillance

Cameras allow the field of view of observers to be extended into areas where they are not physically present. This "virtual presence" of an observer is not necessarily noticed by monitored persons. In the resulting, but misleading feeling of privacy, persons might act differently than they would in the obvious presence of other people. This example makes it apparent, that privacy in video surveillance is an issue that needs special consideration. But when trying to identify what forms of privacy protection are appropriate, the picture becomes less clear. One reason is that there is no common definition of privacy. As discussed in [38, 51], the notion of privacy is highly subjective and what is acceptable depends on the individual person as well as cultural attitudes.

As pointed out by Cavallaro [12] or Fidaleo et al. [24], it is usually more important to be able to observe the behavior of a person than knowing the actual identity. This is achieved by identification and obfuscation of personally identifiable information such as people's faces [15, 35]. Only in situations where, e.g., a law was violated, is this personal information is interesting and should be made available to authorized parties. The main challenge of such an approach is to determine which image regions are actually sensitive. As Saini et al. [45] argue, video data not only includes direct identifiers such as human faces but also quasi identifiers. These quasi identifiers are often based on contextual information and allow to infer the identity of persons with a certain probability. Such basic contextual information about an event includes, e.g., what happened, where did it happen and when did it happen. Vagts et al. [60, 59] present an approach that addresses privacy protection not at the sensor level but at a higher abstraction level. As part of their task-oriented privacy enforcement system, data is only collected if it is required for a surveillance task. For that purpose, each task must be fully specified before data collection is started.

In the following paragraphs we outline key aspects of privacy protection systems. They include basic protection techniques, multilevel approaches that support the recovery of unprotected data under controlled conditions and the need for involving monitored people by asking for their consent and giving them control over their personal data.

Privacy Protection Techniques. A common approach for privacy protection is the identification of sensitive image regions such as human faces of 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. Once the sensitive regions have been identified, different techniques can be applied to achieve de-identification. A very basic approach is blanking where sensitive regions are completely removed. An observer only can monitor the presence and the location of a person. Cheung et al. [17] apply video inpainting techniques to fill the blank areas with background. This way, an observer can no longer notice that information was removed from the video.

An alternative to simple blanking are obfuscation and scrambling where the level of detail in sensitive image regions is reduced such that persons can no longer be identified while their behavior remains perceptible. Researchers apply different techniques including mosaicing, pixelation, blurring [18, 61] or high, lossy compression. Work by Gross et al. [28] indicates the overall protection capabilities of such naive mechanisms are relatively low. A study by Boyle et al. [7] on the effects of filtered video on awareness and privacy indicates that pixelation provides better privacy protection than blurring. Another technique to protect sensitive image regions is scrambling. In its basic form, JPEG compressed images are obscured by pseudo-randomly modifying the DCT coefficients [21] of sensitive regions.

Abstraction techniques replace sensitive image regions with, e.g., bounding boxes or, in case of persons, with avatars, stick-figures and silhouettes [51]. 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 [54]. Depending on the type of abstraction, either behavior, identity or both can be preserved. Identities should be protected using encryption.

Data encryption is used by many systems to protect sensitive regions. When encrypted, regions of interest 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 or split keys as described in [49], a system can be designed that requires multiple operators to cooperate to decrypt the original data. Such a design provides a certain degree of protection against operator misuse.

Multilevel Privacy Protection. Support for multiple privacy levels denotes that one single video stream contains different levels of information. These could range from the unmodified, sensitive image regions over obfuscated versions with blurred faces to abstracted versions. Depending on their sensitivity, these levels can be separately encrypted with one or more individual encryption keys. A multilevel approach allows a privacy protection system to be designed 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 could get access to the original data that contains the identity of monitored persons.

Consent and Control. Ideally, monitored people should first be asked for consent before they are captured by a video surveillance system. Today, installed cameras 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. As these signs are easily overlooked, consent should be sought more actively. Users could be automatically notified about presence and properties of cameras, e.g., via their smartphone. Moreover, monitored people should remain in control of personal data captured by the system. If data is disclosed to a third party, explicit user permission should be required.

Some of these requirements have been addressed in research prototypes. By handing out dedicated devices or RFID tags to known and trusted users, a stronger form of awareness about video surveillance is realized [9, 61]. Users equipped with such devices are not only made aware of the installed cameras but even get a certain degree of control over their privacy. Cameras recognize them as trustworthy and remove or protect the corresponding image regions. The approach of Cheung et al. [16] goes even further. Using public key cryptography to protect personal information, users get full control over their privacy-sensitive data since they have to actively participate in the decryption of this data.

Cavallaro [11, 12] emphasizes 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. Similar ideas are discussed in the already mentioned work of Senior et al. [51]. They suggest that privacy is protected by extracting sensitive information and re-rendering the video into multiple streams individually protected by encryption.

Fleck [25, 26] employs smart cameras from Matrix Vision 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 a termination of its services. Protection against software attacks such as integrity checks or data encryption is not part of the current system.

Boult [6] 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 the actions of a person to be monitored without revealing the person's identity. The faces are only decrypted if, e.g., a crime was committed by the person. Encryption is performed as part of image compression and uses a combination of symmetric and asymmetric cryptography. Additionally, it is suggested that checksums of frames or sub-sequences are computed to ensure data integrity. In related work, Chattopadhyay and Boult present PrivacyCam [14], 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 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. [44] 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.

Moncrieff et al. [38] argue that most of the proposed systems rely on predefined security policies and are either too intrusive or too limited. Therefore, they suggest that dynamic data hiding techniques are applied. 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.

Dufaux and Ebrahimi [21] suggest scrambling of sensitive image regions. After detection of relevant areas, images are transformed using DCT. The signs of the coefficients 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 study by Dufaux and Ebrahimi [22] indicates that scrambling is superior to simple approaches such as pixelation and blurring.

A similar approach is discussed by Baaziz et al. [4] 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 detection of manipulation of image data. Limited reconstruction of manipulated image regions is possible due to redundancy introduced by the watermark. Yabuta et al. [68] also propose a system where DCT encoded image data is 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. Li et al. [32] present an approach towards recoverable privacy protection based on discrete wavelet transform. Information about sensitive image regions together with their wavelet coefficients are protected with a secret key. Data hiding techniques are used to embed this information into the resulting image.

Qureshi [42] proposes a framework for privacy protection in video surveillance based on decomposition of raw video into object-video streams. Based on a segmentation approach, pedestrians are identified. Tracking is performed using color features. The privacy of detected persons is protected by selectively rendering the corresponding objects. Advanced protection mechanisms such as encryption are left as future work. Also the system presented by Tansuriyavong and Hanaki [54] is based on detection of sensitive entities. In an office scenario, the silhouettes of detected persons are blanked. Additionally, the system integrates face recognition to identify previously registered persons. Configuration options allow the choice of what information should be disclosed – full images, silhouettes, names of known persons or any combination thereof.

Troncoso-Pastoriza et al. [56] 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. [24], 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 running on the server. Data received from the clients is fed into this privacy buffer. The buffer can be extended and configured by means of privacy filters and a privacy grammar. If incoming data is qualified as private by one of the privacy filters, the data does not leave the privacy buffer. Non-private data is forwarded to a routing component that manages distribution of data to interested clients.

To protect the privacy of only selected users, systems have been presented that allow to remove known, trusted users from captured video. Due to the limited reliability of computer vision to detect personal image data, many researchers rely on portable devices carried by users for identification and localization. One such approach is presented by Brassil [9]. He 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. This data is 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 has to be anonymized. Due to the absence of feedback, users have to trust camera operators to follow the advertised procedures.

Wickramasuriya et al. [61] perform realtime monitoring of the environment to increase user privacy. In particular, they suggest that motion sensors are used 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 that 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. [18] 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 that, 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. [16] 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 [17]. 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 work from the same research group, Ye et al. [69] and Luo et al. [33] 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 that, authorized individuals are then obfuscated in the captured video. Anonymity of authorized persons is maintained by using homomorphic encryption.

An approach that does not need electronic devices that are carried by users is presented by Schiff et al. [50]. Their “respectful cameras” use visual markers such as yellow hard hats worn by people to identify privacy-sensitive regions. Specifically, they remove person’s faces from images. For marker detection and tracking, probabilistic AdaBoost and particle filtering are used. Spindler et al. [53] apply similar ideas in the context of building automation and monitoring applications. Personal data is 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.

### *2.3 Observations and Open Issues*

Most research on privacy and security in video surveillance is on selected and isolated topics. Figure 1 gives an overview of the three major areas. The majority of work addresses data-centric security and privacy issues which include authenticity and integrity of data, data freshness, timestamping as well as confidentiality. Ideally, data-centric security guarantees should be provided for the entire lifetime of data, i.e., from the moment an image is captured by the camera’s sensor until the image and all derived data are deleted. As a consequence, data-centric security involves all components of a visual sensor network including monitoring stations as well as video archives. Adequate access authorization techniques must be integrated such that sensitive data can be accessed only by legitimate users.

When considering the architecture of a VSN node it is apparent that data-centric protection features are implemented typically as part of the camera’s applications. To be able to provide meaningful security guarantees for captured and processed data the VSN device itself must be secured. This aspect, which is referred to as node-centric security in Figure 1, is rarely addressed in related work. In a holistic approach, the security of both the VSN’s hardware as well as its software stack must be taken into account. Otherwise, the protection achieved by application level security mechanisms must be questioned.

The third major group of security issues shown in Figure 1 is network-centric security where a primary goal is a secure channel between two communication partners. This could be two cameras or one camera and a monitoring or archiving facility. A secure communication channel must provide basic non-repudiation and confidentiality properties. To a certain extent, there might be a redundancy between network channel security and data-centric security. The actual protection requirements depend on the specific application. An additional and equally important aspect is secure collaboration of multiple cameras. To facilitate secure collaboration, a range of topics must be considered such as secure data sharing and aggregation, localization and topology control, camera discovery and lookup mechanisms as well as inter-camera time synchronization.

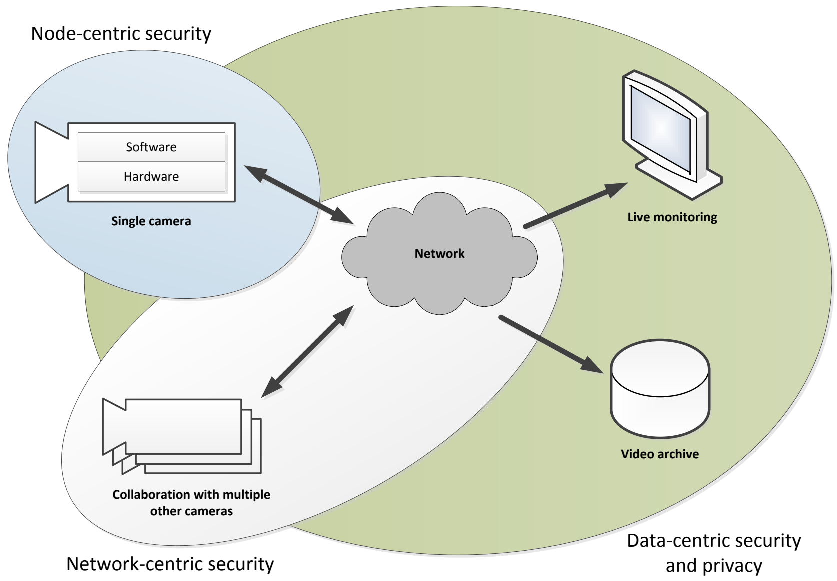

**Fig. 1** The security requirements discussed in this chapter can be classified into three groups. First, node-centric security refers to security of the camera's hardware as well as its software stack. Second, network-centric security covers security of the communication channel and security aspects for inter-camera collaboration which include secure data sharing and aggregation techniques, camera discovery, topology control or time synchronization. The third group is data-centric security which denotes security (e.g., integrity, authenticity, etc.) and privacy protection for data from its creation to its deletion.

In our review of related work we identified some of the most important open issues.

Comprehensive Privacy Protection. The meaning of privacy in video surveillance is still a vague term. As discussed previously there is consensus that privacy protection denotes the protection of persons' identities while their behavior remains visible. However, it is not clear if the proposed protection techniques such as pixelation, blurring or scrambling are actually effective. Research by Dufaux and Ebrahimi [22] and Gross et al. [28] indicates that basic obfuscation techniques might provide less protection than previously thought. Additionally, object-based privacy protection mechanisms assume the availability of reliable detection algorithms for the identification of sensitive image regions. A mis-detection in a single frame of a video sequence can be sufficient to breach privacy for the entire sequence. Based on this observations, Saini et al. [47] suggest to rely on global protection techniques instead of object-based approaches. Global approaches apply uniform protection operations (e.g., downsampling, coarse quantization or edge detection) to the entire raw image and are therefore not prone to errors in the detection of sensitive regions.

But identity leakage does not result only from primary identifiers such as human faces. Contextual information [48] such as the location, the time and the observed action can also be sufficient to derive the identity of persons. The usefulness of this contextual information depends directly on the knowledge of the observer. One approach to reduce the likelihood of identity leakage via contextual information is to ensure that monitoring of video data is performed by randomly chosen persons without knowledge about the observed area and context [46]. The practical feasibility of such approaches is yet to be determined.

Regardless of the chosen approach – privacy protection reduces usually the amount of information that is available in a video and therefore privacy protection has a negative impact on system utility. An important aspect will be to explore the privacy vs. system utility design space and to determine a suitable and most probably application specific tradeoff.

Holistic Security Concept. There is still a lack of work that considers security and privacy in VSNs in a holistic way. It is apparent that most security solutions are situated at the application level and that node-centric security is not taken into account. Substantial work has been targeted at data- and network-centric security. But without addressing security of VSN nodes themselves, high-level protection mechanisms are literally built on sand. VSN designers will have to collaborate with engineers from other embedded system domains such as mobile handsets to promote the development of standardized node-centric security solutions.

Sensor-level Security. Securing the VSN device is an important yet complicated task. On modern embedded camera systems a large amount of software is executed. This includes the operating system with all its subsystems such as the network stack as well as system libraries and middeleware components. Due to the substantial size of these software components it is impractical to fully verify them. As a consequence these components have to be implicitly trusted. One potential approach to address this issue would be to bring security and privacy protection closer to the sensor or even making them part of the sensor. If security and privacy are guaranteed at the sensor level, then the camera and its relatively large software stack would no longer have to be considered as trusted entities. This approach implies two major challenges: First, it is unclear what type of privacy protection is suitable and feasible at the sensor level. Second, sensor-level privacy protection means that image processing and analysis applications on the camera must be adapted to deal with pre-processed and pre-filtered data. A critical question is the identification of an appropriate tradeoff between sensor-level security and privacy protection and the remaining utility of the camera host system.

## 3 TrustCAM: A Camera with Hardware Security Support

This section describes an approach that specifically addresses two major issues outlined previously in Section 2.3: node-centric security and providing data-centric security guarantees for all data that is delivered by the camera. The presented TrustCAM prototype [65, 66, 63, 64] puts a strong focus on node security to ensure that high-level data protection algorithms can be built on a solid basis. A fundamental question in computer security is whether a software solution can provide adequate levels of security or if an immutable hardware component is required that acts as a trust anchor. The later is assumed by an industry initiative called Trusted Computing Group (TCG). The main output of the group is a set of open specifications for a hardware chip – the Trusted Platform Module (TPM) [57] – and software infrastructure such as the the TCG Software Stack (TSS) [58]. The TPM chip implements a small and well defined set of core security functions which can not be altered by the TPM's host system. This approach of a hardware-based security solution has been adopted by the TrustCAM project for embedded smart cameras. The TrustCAM prototype as shown in Figure 2 incorporates an Atmel AT97SC3203S TPM chip which is used to various security aspects including recording the boot process and software state of the camera device, securely storing cryptographic keys or digitally signing and encrypting outgoing data.

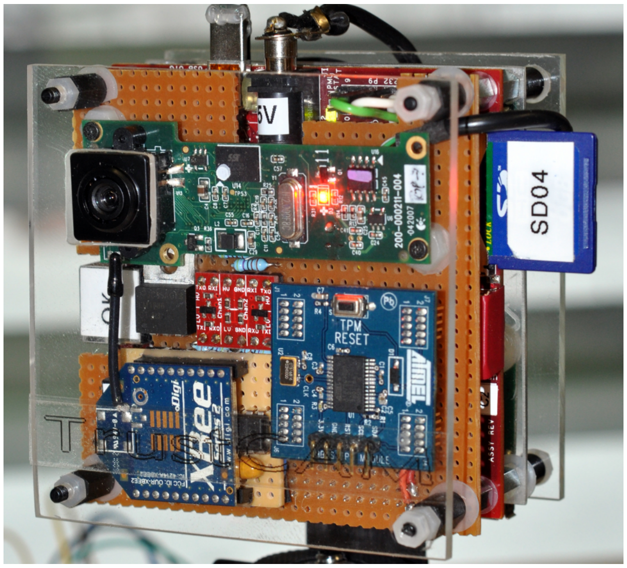

**Fig. 2** The TrustCAM prototype. The image sensor, the XBee radio and the Atmel TPM can be seen on the front circuit board. Behind this board are the processing board and WiFi radio.

The system largely consists of commercial, off-the-shelf components. It is based on the BeagleBoard [55] (rev. C2) embedded processing platform. The board is equipped with an OMAP 3530 SoC from Texas Instruments. The OMAP SoC features a dual-core design and contains an ARM Cortext A8 processor which is clocked at up to 600 MHz and an additional TMS320C64x+ DSP that can run at speeds of up to 480 MHz. For stability reasons, the clock frequency of the TrustCAM's ARM core is set to 480 MHz. The DSP is not used in the current version of the prototype. The prototype is equipped with 256 MB of LPDDR RAM and 256 MB NAND flash memory. A CMOS image sensor (Logitech QuickCam Pro 9000) is connected via USB. Wireless connectivity is provided by an RA-Link RA-2571 802.11b/g WiFi adapter. An additional, low-performance wireless communication channel is implemented via an 802.15.4 based XBee radio connected to one of the platform's UARTs.

## *3.1 Trusted Computing Preliminaries*

This section provides a brief overview of the most important Trusted Computing (TC) and TPM concepts. More detailed information can be found in the specifications of the TCG [57] and auxiliary sources [34, 13]. The TPM is typically implemented as a secure 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. With an Opt-In process, users can choose if they want to make use of the TPM.

RSA keys can be generated for different purposes such as encryption or signing. Upon creation, keys can be declared migratable or not. While migratable keys can be transferred to a different TPM, non-migratable keys can not. 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:

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

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 state 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 to use the TPM to securely store data. Binding of data refers to encrypting data with a TPM key and hence guaranteeing that this data only is accessible by this specific TPM instance. Sealing of data allows to specify a set of PCR values the data is associated with. Like unbinding, 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.

## *3.2 System Architecture and Setup*

The primary goals of the TrustCAM system design are to provide authenticity, integrity, freshness and timestamping as well as confidentiality and multilevel privacy protection for streamed image and video data. As illustrated in Figure 3, each TrustCAM of a visual sensor network (VSN) is assumed to be equipped with a TPM chip subsequently called $TPM_C$. Throughout the VSN, network connectivity is provided by wireless communication in single or multi-hop mode. For this work, cameras are assumed to be controlled and operated from a central facility subsequently called the Control Station (CS). A fundamental assumption is that the CS is a secure and trustworthy facility.

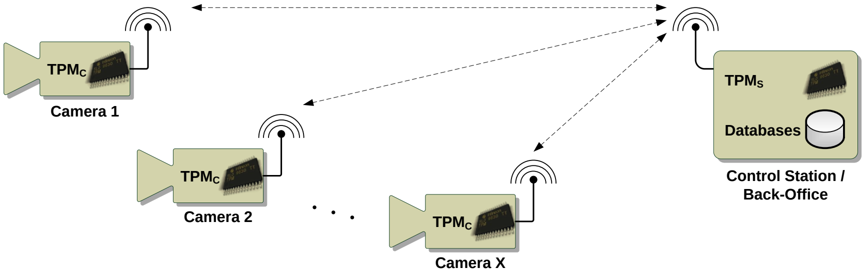

**Fig. 3** A network of $X$ TPM-equipped TrustCAMs which are managed by a central control station.

Figure 3 shows a network consisting of $X$ TrustCAM nodes and one central control station. Not only the cameras, but also the control station is equipped with a TPM subsequently referred to as $TPM_S$. In addition to $TPM_S$, the CS also hosts several databases to store cryptographic keys generated during camera setup as well as data received from the cameras.

It is assumed that camera setup is done when cameras are 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 done individually for each of the $X$ cameras of the network.

TPM Ownership. Initially, the camera's TPM has to be activated. Calling the TakeOwnership operation of $TPM_C$ sets an owner password 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 CS database.

Identity Key Creation. An Attestation Identity Key ($K_{AIK}$) serves as an alias for the TPM's Endorsement Key ($K_{EK}$) and is used during platform attestation. In contrast to a conventional PC, there are not multiple human users on a TrustCAM. The system software running on the camera takes the role of a single system user. Moreover, all cameras in the network are uniquely identified and well known by the operators. Consequently, there is no need for the anonymity gained by using multiple AIKs in conjunction with a PrivacyCA [41]. Therefore, only a single Attestation Identity Key $K_{AIK}$ is generated during setup that serves for platform attestation. The public part $K_{AIK_{pub}}$ is stored in the CS database.

Signature Key Creation. For signing data such as events or 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.

Binding Key Creation. To ensure confidentiality and privacy protection, sensitive image data sent from the camera to the CS has to be encrypted. Encryption should be done at different levels including the full images as well as special regions of interest where, e.g., motion or faces have been detected.

To ensure confidentiality, at least one non-migratable binding key $K_{BIND\_1}$ is created by the control station's $TPM_S$. The public portion of this key, $K_{BIND\_1_{pub}}$, is exported from $TPM_S$ and stored on the camera. Note that the private part of $K_{BIND\_1}$ cannot be exported from $TPM_S$ and therefore, data encrypted with $K_{BIND\_1_{pub}}$ can only be decrypted at the CS and not by an intermediate attacker who interferes with the transmission. To decrypt data bound with $K_{BIND\_1_{pub}}$, the usage password of the key has to be supplied by the system operator. To avoid that a single operator who knows this usage password and has access to the control station can decrypt data, additional binding keys $K_{BIND\_2}$ to $K_{BIND\_N}$ are generated. Privacy sensitive data can be encrypted sequentially with multiple binding keys. Assuming that no single operator knows all the usage secrets for these binding keys, two or more operators have to cooperate to decrypt the data. The $N$ binding keys can be used also to realize different security levels. Data at different abstraction levels (e.g., full images vs. images where people's faces have been removed vs. textual event descriptions) can be encrypted with different binding keys. Depending on security clearance, only certain abstraction levels can be accessed by an operator.

Table 1 summarizes the cryptographic keys that are generated as part of the setup procedure of a single camera.

| | **Control Station** | **TrustCAM** |
|---|---|---|
| Endorsement Key | $K_{EK_{pub}}$ | $K_{EK}$ |
| Storage Root Key | - | $K_{SRK}$ |
| Attestation Identity Key | $K_{AIK_{pub}}$ | $K_{AIK}$ |
| Signature Key | $K_{SIG_{pub}}$ | $K_{SIG}$ |
| Binding Keys | $K_{BIND\_1}$ | $K_{BIND\_1_{pub}}$ |
| | $K_{BIND\_2}$ | $K_{BIND\_2_{pub}}$ |
| | ... | ... |
| | $K_{BIND\_N}$ | $K_{BIND\_N_{pub}}$ |

**Table 1** The cryptographic keys generated during setup of a single camera. The “Control Station” and “TrustCAM” columns denote the storage location of the keys. Binding keys are generated by $TPM_S$ while all other keys are generated by $TPM_C$. All keys are non-migratable, 2048 bit RSA keys. The *pub* subscript denotes the public RSA key.

Once the setup procedure is complete, the camera can be deployed. The boot process of the camera as well as its entire software state including all executed applications is recorded in the PCRs of its $TPM_C$. To monitor both the availability and the executed applications we have previously proposed a trusted lifebeat. The involved tursted lifebeat protocols, the mapping of camera timestamps to world time as well as the trusted boot procedure of TrustCAM are fully detailed in [65].

## *3.3 Video Confidentiality, Authenticity and Integrity*

The TrustCAM system is designed to ensure (1) confidentiality of all image data as a protection against external attackers and (2) selective privacy protection to provide system operators with sufficient information to fulfill their duties without automatically revealing the identity of monitored persons. Furthermore, the proposed design provides (3) authenticity, (4) integrity and (5) timestamping guarantees for delivered data.

The basic concept is shown in Figure 4. Image data grabbed from the camera’s sensor is first analyzed and regions of interest (ROI) are detected. The definition of regions of interest depends on the application and can range from motion areas over vehicle license plates to people’s faces. The ROI are then extracted from the image. The remaining background image $Img_{BACK}$ as well as the extracted, original ROI $Img_{ROI}$ are compressed. Additionally, one or more abstracted versions $Img_{ABST\_[1...A]}$ of the ROI are created. Abstracted versions can be images where,

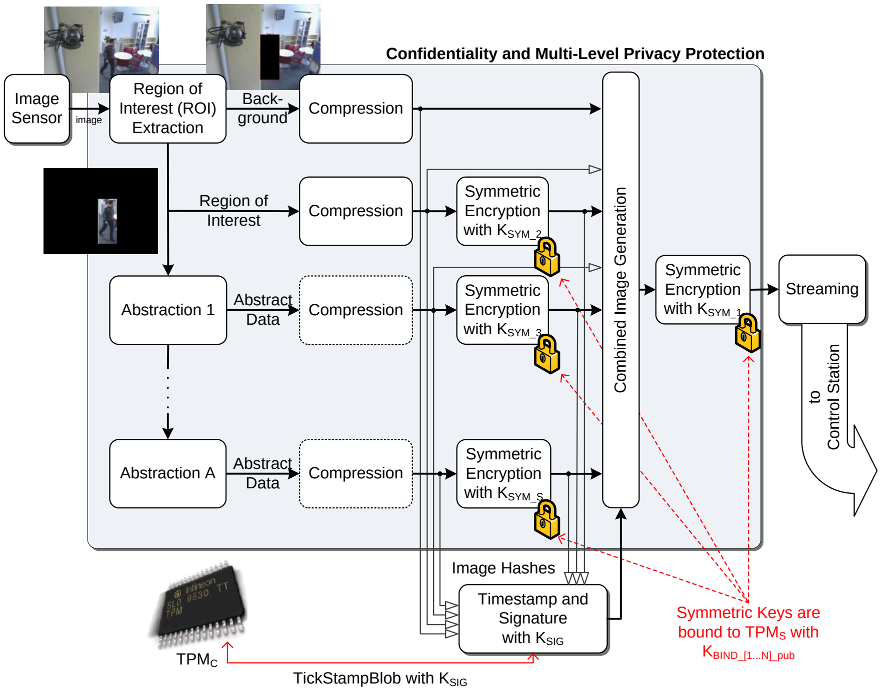
Confidentiality and Multi-Level Privacy Protection
Image Sensor
image
Region of Interest (ROI) Extraction
Back-ground
Compression
Region of Interest
Compression
Symmetric Encryption with K_SYM_2
Abstraction 1
Abstract Data
Compression
Symmetric Encryption with K_SYM_3
Abstraction A
Abstract Data
Compression
Symmetric Encryption with K_SYM_S
Combined Image Generation
Symmetric Encryption with K_SYM_1
Streaming
to Control Station
Image Hashes
Timestamp and Signature with K_SIG
TPM_C
TickStampBlob with K_SIG
Symmetric Keys are bound to TPM_S with K_BIND_[1...N]_pub

**Fig. 4** Captured images are analyzed and regions of interest (ROI) are extracted. Abstracted versions of the ROI, the unmodified ROI as well as the remaining background are separately compressed. The ROI parts of the video stream are encrypted with symmetric session keys that are bound to $TPM_S$. Hash values of the compressed images and the encrypted ROI images are signed and timestamped by $TPM_C$. The background image, the encrypted ROI images, the ROI hashes and the signature are combined into a common container which is then encrypted. Subsequently, the video data is streamed to the control station.

e.g., faces are blurred or persons are replaced with stick figures or generic avatars. Alternatively, the output of the abstraction process can be also non-image data such as a textual description. While compression of abstracted data is optional and depends on the actual data type, encryption is mandatory:

$$\begin{aligned} Img_{ROI_{enc}} &= \mathtt{Encrypt}_{K_{SYM\_2}}(Img_{ROI}).\\ Img_{ABST\_1_{enc}} &= \mathtt{Encrypt}_{K_{SYM\_3}}(Img_{ABST\_1}).\\ &\qquad\qquad\qquad\qquad\ldots\\ Img_{ABST\_A_{enc}} &= \mathtt{Encrypt}_{K_{SYM\_S}}(Img_{ABST\_A}). \end{aligned}$$

Upon startup of the streaming session, the symmetric session keys $K_{SYM\_[2...S]}$ are bound to the control station's $TPM_S$ using the non-migratable binding keys $K_{BIND\_2_{pub}}$ to $K_{BIND\_N_{pub}}$:

$$\begin{aligned}K_{SYM\_2_{bound}} &= \mathtt{Bind}_{K_{BIND\_3_{pub}}}(\mathtt{Bind}_{K_{BIND\_2_{pub}}}(K_{SYM\_2})).\\ K_{SYM\_3_{bound}} &= \mathtt{Bind}_{K_{BIND\_4_{pub}}}(K_{SYM\_3}).\\ &\dots\\ K_{SYM\_S_{bound}} &= \mathtt{Bind}_{K_{BIND\_N_{pub}}}(K_{SYM\_S}).\end{aligned}$$

Binding $K_{SYM\_2}$ successively with two independent binding keys enforces the four-eyes principle for the original ROI at the control station where two operators have to cooperate to decrypt the data. Decryption at the control station requires knowledge of the usage passwords of the respective binding keys. Depending on individual security clearance, an operator might be able to, e.g., decrypt the background image and an abstracted version of the regions of interest that reveals the behavior of monitored persons. ROI versions that contain a person's identity are reserved for, e.g., supervisors with higher clearance. To prevent operator misuse, especially sensitive data can be protected by double-encryption of the symmetric session key such that two operators have to cooperate to decrypt the data. This is illustrated for $K_{SYM\_2}$ which is used to encrypt the original ROI. It is protected twice using $K_{Bind\_2}$ and $K_{Bind\_3}$.

To couple data integrity and authenticity guarantees with data confidentiality, the encrypt/sign/encrypt approach discussed by Davis [19] is applied. As shown in figure 4, the hashes of the plain image regions $Img_{BACK}$, $Img_{ROI}$ and $Img_{ABST\_[1...A]}$ as well as those of their encrypted equivalents are computed. Including both in the signature demonstrates that the plaintext as well as the ciphertext come from the same origin and provides protection against plaintext substitution attacks. Furthermore, by signing the plaintext, non-repudiation guarantees are given. Additionally, the system operator can correlate the inner encryption with the outer encryption by checking that the used binding keys all belong to the same camera. This protects against potential "surreptitious" forwarding attacks [19].

$$\begin{aligned}H_{BACK} &= \mathtt{SHA-1}(Img_{BACK}).\\ H_{ROI} &= \mathtt{SHA-1}(Img_{ROI}).\\ H_{ABST\_[1...A]} &= \mathtt{SHA-1}(Img_{ABST\_[1...A]}).\\ H_{ROI_{enc}} &= \mathtt{SHA-1}(Img_{ROI_{enc}}).\\ H_{ABST\_[1...A]_{enc}} &= \mathtt{SHA-1}(Img_{ABST\_[1...A]_{enc}}).\end{aligned}$$

The individual hash sums are concatenated and a common hash sum $H_{Img}$ is computed:

$$H_{Img} = \mathtt{SHA-1}(H_{BACK}||H_{ROI}||H_{ABST\_[1...A]}||H_{ROI_{enc}}||H_{ABST\_[1...A]_{enc}}).$$

Due to performance limitations of current TPM implementations it is impossible to sign and timestamp every image hash $H_{Img}$ individually. Instead, an accumulated hash sum for a group of $F$ frames is computed:

$$AccSum_{Img[1\ldots F]} = \texttt{SHA-1}(AccSum_{Img[1\ldots(F-1)]}||H_{Img})).$$

This accumulated hash sum, the current tick values as well as the accumulated hash sum of the previous image group are then singed and timestamped by the camera's $TPM_C$:

$$TickStamp_{Res} = \texttt{TPM\_TickStampBlob}_{K_{SIG}}(TSN_{Img_F}||TCV_{Img_F}||TRATE_{Img_F}|| \\ AccSum_{PrevGroup}||AccSum_{Img[1\ldots F]}).$$

In the next step, the various components are combined into a common image container $Img_{COMB}$:

$$Img_{COMB} = [ImageParts, ImageHashes, K_{SYM\_[2\ldots S]_{bound}}, Timestamp].$$

with:

$$\begin{aligned} ImageParts &= [Img_{BACK}, Img_{ROI_{enc}}, Img_{ABST\_[1\ldots A]_{enc}}]. \\ ImageHashes &= [H_{ROI}, H_{ABST\_[1\ldots A]}]. \\ Timestamp &= [TickStamp_{Res}, TSN_{Img_F}, TSV_{Img_F}, TRATE_{Img_F}, start_{idx}, end_{idx}]. \end{aligned}$$

This combined image includes the background image, the encrypted original ROI as well as the encrypted abstracted ROI images. Additionally, it contains the hashes of the original and abstracted ROI images, the bound session keys and, in the case of the end of a frame group, the group's timestamp and signature together with start and end indices. Finally, the combined image $Img_{COMB}$ is encrypted using $K_{SYM\_1}$ which, in turn, is bound to $TPM_S$:

$$\begin{aligned} Img_{COMB_{enc}} &= \texttt{Encrypt}_{K_{SYM\_1}}(Img_{COMB}). \\ K_{SYM\_1_{bound}} &= \texttt{Bind}_{K_{BIND\_1_{pub}}}(K_{SYM\_1}). \end{aligned}$$

Since all image data including the background and the regions of interest as well as the derived abstracted versions are encrypted, confidentiality of all personal information is ensured. This also includes personal information that was accidentally missed by the ROI detection algorithm. Furthermore, using non-migratable signing keys for data singing guarantees data authenticity and integrity. Validation of associated timestamps and the mapping of local camera timestamps to world time is discusses in detail in [65]. In the last step, the encrypted, combined image data and the bound session key are streamed to the control station.

At the control station, a system operator can decrypt the individual image parts depending on the knowledge of the usage passwords of the camera's binding keys.

Typically, an operator can only decrypt a subset of the included data. As a consequence, not all hash values of the ROI ($H_{ROI}$) and abstracted ROI ($H_{ABST[1...A]}$) images can be computed. To still be able to verify the signature of the frame group, the operator can substitute the missing hashes with those from the $ImageHashes$ field included in the combined image. This approach allows verification of the overall signature of the frame group as well as the integrity and authenticity of those image parts which are accessible by the operator. The strategy used is based on the star chaining concept for hash values proposed by Wong and Lam [67] and has two main advantages. First, an operator can validate the integrity and authenticity of those image parts he actually sees and has legitimate access to. No decryption of additional image components is required. Second, on the camera only one single hash value (the accumulated $H_{Img}$) has to be sent to $TPM_C$ for signing and timestamping despite the various individual parts the combined image might contain. This is an important advantage when considering the low performance of current TPM chips.

To illustrate the verification of the timestamp and signature, the following example is given. Operator 1 ($OP1$) at the control station knows the usage secrets for $K_{Bind\_1}$ and $K_{Bind\_4}$ which gives him access to the background image ($Img_{BACK}$) and the first abstracted ROI image ($Img_{ABST\_1}$). For signature verification, the control station software computes the hashes of these two images:

$$H_{OP1\_BACK} = \texttt{SHA-1}(Img_{BACK}).$$
$$H_{OP1\_ABST\_1} = \texttt{SHA-1}(Img_{ABST\_1}).$$

Likewise, the hashes of the included encrypted image regions are computed:

$$H_{OP1\_ROI_{enc}} = \texttt{SHA-1}(Img_{ROI_{enc}}).$$
$$H_{OP1\_ABST\_[1...A]_{enc}} = \texttt{SHA-1}(Img_{ABST\_[1...A]_{enc}}).$$

Due to access limitations, operator 1 cannot compute the hashes $H_{ROI}$ and $H_{ABST\_[2...A]}$ since the usage passwords for the binding keys required to decrypt the corresponding image parts are unknown. The missing hashes are substituted with $H_{ROI}$ and $H_{ABST\_[2...A]}$ from the $ImageHashes$ field of $Img_{COMB}$:

$$H_{OP1\_Img} = \texttt{SHA-1}(H_{OP1\_BACK}||H_{ROI}||H_{OP1\_ABST\_1}||H_{ABST\_[2...A]}|| \\ H_{OP1\_ROI_{enc}}||H_{OP1\_ABST\_[1...A]_{enc}}).$$

The hash sum $H_{OP1\_Img}$ now serves as input for the computation of the expected accumulated hash sum which, in turn, is used for group signature verification.

Finally, it must be noted that the number of abstraction levels, the video compression algorithms, the container format for the combined image as well as the streaming format can be freely chosen by the application developer. Note that the discussed approach focuses on the protection of outgoing, sensitive image data. It does not cover control and status messages exchanged between cameras or the control station. For that purpose, additional mechanisms such as Transport Layer Security (TLS) can be considered.

### 3.4 Implementation Aspects

For the prototype, all image areas where motion is detected are defined as sensitive. From the extracted ROI, an abstracted version is created using edge-detection. The background image $IMG_{BACK}$ allows the presence and position of persons to be observed, the edge-detected ROI $IMG_{EDGE}$ gives access to behavioral information and the original ROI $IMG_{ROI}$ reveals both behavior and identity of detected persons/moving objects. Next, the background and the two ROI images are compressed. JPEG compression is used for the background and the original ROI while the black and white edge-detected ROI is compressed using zlib. The compressed regions of interest $Img_{EDGE}$ and $Img_{ROI}$ are encrypted using AES 265 in CBC mode and the AES session keys are bound to CS's $TPM_S$ using the binding keys that have been generated for this camera during setup:

$$
\begin{aligned}
K_{AES\_1_{bound}} &= \mathtt{Bind}_{K_{BIND\_1_{pub}}}(K_{AES\_1}).\\
K_{AES\_2_{bound}} &= \mathtt{Bind}_{K_{BIND\_2_{pub}}}(K_{AES\_2}).\\
K_{AES\_3_{bound}} &= \mathtt{Bind}_{K_{BIND\_4_{pub}}}(\mathtt{Bind}_{K_{BIND\_3_{pub}}}(K_{AES\_3})).
\end{aligned}
$$

The video format that was chosen for the prototype is Motion JPEG (MJPEG). As shown in Figure 5, the encrypted image regions $Img_{ROI_{enc}}$ and $Img_{EDGE_{enc}}$ are embedded into the background image as custom EXIF data. Likewise, the bound AES keys $K_{AES\_2_{bound}}$ and $K_{AES\_3_{bound}}$ as well as the SHA-1 hashes of the unencrypted $Img_{ROI}$ and $Img_{EDGE}$ are included.

Subsequently, the SHA-1 hash of the concatenated hash sums of $Img_{BACK}$, $Img_{EDGE}$, $Img_{ROI}$, $Img_{EDGE_{enc}}$ and $Img_{ROI_{enc}}$ is computed and is fed into the previously described hash accumulation procedure of the frame group. The accumulated hash then is signed and timestamped by $TPM_C$ once the end of the frame group is reached. The resulting signature and timestamp data as well as the start and end indices of the frame group are included in the EXIF data of combined image shown in Figure 5.

At the control station, the streamed frames have to be decrypted before viewing. Note that access to the original ROI $IMG_{ROI}$ requires the cooperation of two security guards since the corresponding AES session key $K_{AES_3}$ is bound with the two binding keys $K_{BIND\_3}$ and $K_{BIND\_4}$. The right part of figure 6 shows the video stream at the control station where the background image is overlayed with the decrypted, edge-detected region of interest.

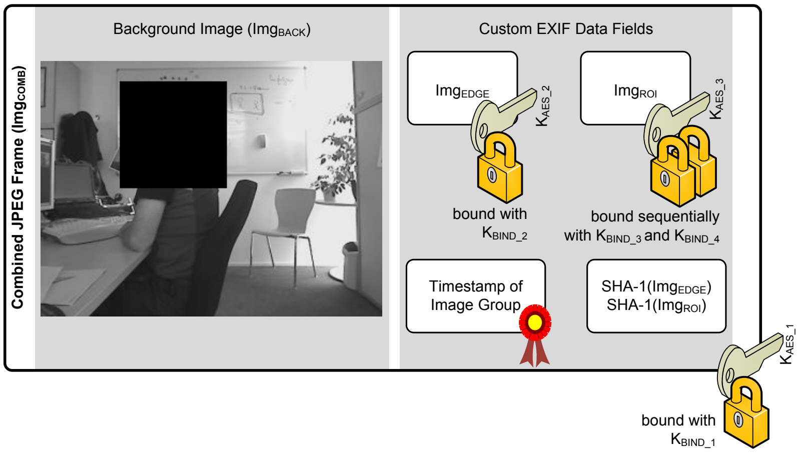

**Fig. 5** The encrypted ROI image ($Img_{ROI_{enc}}$) as well as the encrypted edge image ($Img_{EDGE_{enc}}$) are embedded into the JPEG background image as custom EXIF data. The same is done for the bound AES keys as well as the SHA-1 hashes of $Img_{ROI}$ and $Img_{EDGE}$. At the end of a frame group, the group's signature and timestamp are also included in the EXIF data. The combined image ($Img_{COMB}$) is then encrypted and streamed to the control station.

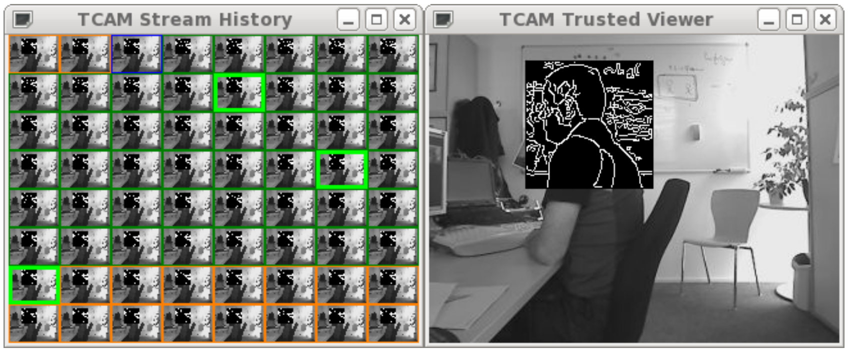

**Fig. 6** The live viewer at the control station. On the right the current frame with the decrypted, edge-detected ROI is displayed. The left window shows the content of a circular buffer with the last 64 frames. The current frame is marked with a blue border. Frames with a signature that has not yet been verified have an orange border while successfully verified frames have a dark green border. The last frame of a group has a light green border.

Accumulated image signatures and timestamps of frame groups are validated at the control station. Assuming that this validation is successful, the operator at the CS has assurance that neither the individual images of a frame group nor their order was modified and the images of the group come from the expected camera. Freshness checks and world time mapping can be done as described in [65].

The left side of the live stream viewer example of figure 6 shows a circular buffer that contains thumbnails of the last 64 received frames together with their verification status. For frames with orange borders, the frame group signature was not yet received. Already verified frames have a green border and the last frame of a group is marked with a light green border. If authenticity and integrity of an image group cannot be verified before the circular buffer wraps around, a warning message is displayed and streaming is interrupted.

### *3.5 Performance Considerations*

Table 2 presents the frame rates that are achieved on TrustCAM in different streaming modes. The sensor can deliver images either as uncompressed YUYV data or as a JPEG compressed version. Input images are delivered at a resolution of 320×240 or 640×480 pixels. For internal processing, input images are converted to either RGB or grayscale. The "Plain Streaming" column of table 2 shows the achieved streaming frame rates if no ROI is extracted and neither encryption nor digital signatures are performed. Therefore, this column reflects the baseline streaming performance of the system without any additional security or privacy protection.

The second column, "Image Timestamping", shows the delivered frame rates if groups of outgoing images are timestamped. Overheads for the TPM Tickstamp-Blob command are eliminated from the critical path by signing frame groups and executing the TPM operations asynchronously. As a consequence, the small performance impact that can be observed for some cases in the "Image Timestamping" column result from the additional computation of the accumulated SHA-1 hash for a frame group. Performance impacts on video streaming can be observed if YUYV input images are used. In this case, the images have to be JPEG compressed before being hashed and streamed. JPEG compression is computing intensive and puts a high load on the OMAP's ARM CPU. Therefore, even the small additional effort of the SHA-1 computation results in a reduction in the frame rate.

The "Image Encryption" column of table 2 presents the achieved frame rates if a randomly placed region of interest is extracted from the input image, the ROI images are encrypted and embedded into the remaining background and, finally, the combined image is encrypted. For data encryption, AES 256 in CBC mode is used. Encryption runtimes for typical input sizes range from 1.6 ms (8 kB) to 15.4 ms (80 kB). Across all input format combinations, a considerable impact on the achieved streaming framerate can be observed. Another slight performance reduction can be perceived in the last column of table 2 which presents the frame rates if both image timestamping and encryption (ROI size 200x200 pixels) are performed.

To investigate the cause for the substantial performance impact that is apparent in the "Image Encryption" column of table 2, the involved processing steps have been analyzed in detail (see [65] for details). This analysis reveals that the runtime overheads for AES 265 encryption and SHA-1 computation are acceptable. AES encryption for the compressed ROI takes around 1.5 ms while only 1 ms is required for the compressed edge image. For the combined image, where the encrypted ROI and edge image are embedded as EXIF data, AES encryption requires between 4 and 9 ms. Binding of the AES session keys using the public binding keys of $TPM_S$

| **Input Format** | | **Internal** | **Plain** | **Image** | **Image** | **Image Encryption** |
|---|---|---|---|---|---|---|
| Resolution | Type | **Format** | **Streaming** | **Timestamping** | **Encryption** | **and Timestamping** |
| **320x240** | **YUYV** | **Gray** | 25.0 fps | 25.0 fps | 20.5 fps | 19.7 fps |
| | **JPEG** | | n/a | n/a | 13.5 fps | 13.2 fps |
| | **YUYV** | **RGB24** | 25.0 fps | 24.4 fps | 12.4 fps | 12.0 fps |
| | **JPEG** | | 25.0 fps | 25.0 fps | 10.3 fps | 10.1 fps |
| **640x480** | **YUYV** | **Gray** | 13.1 fps | 12.8 fps | 9.6 fps | 9.2 fps |
| | **JPEG** | | n/a | n/a | 5.1 fps | 5.0 fps |
| | **YUYV** | **RGB24** | 6.5 fps | 6.4 fps | 5.1 fps | 5.0 fps |
| | **JPEG** | | 25.0 fps | 25.0 fps | 4.0 fps | 3.9 fps |

**Table 2** Frame rates (avg. over 1000 frames) for different types of video streaming between TrustCAM and CS via WiFi. In the “Plain Streaming” case, JPEG or YUYV frames are delivered by the sensor. JPEG frames are directly streamed as a MJPEG video stream. Note that JPEG images delivered by the sensor unit are in RGB. A conversion to grayscale would only add an extra overhead for decompression and recompression and is therefore omitted (cells marked with *n/a*). YUYV frames are converted to grayscale or RGB24 before they are JPEG compressed and streamed. The “Image Timestamping” column presents the achieved frame rates if groups of full, unmodified images are signed and timestamped. The “Image Encryption” column shows the frame rates that are achieved if a randomly placed region of interest of 200×200 pixels is extracted, an edge-detected version is created and the individual image parts ($Img_{ROI}$, $Img_{EDGE}$ and $Img_{COMB}$) are encrypted before streaming. Finally, the last column shows the achieved frame rates when doing both – image timestamping/signing and encryption – before streaming.

takes about 5 ms and has to be done only at startup of the streaming application or when new session keys are created. Finally, SHA-1 computation requires between 2 and 3.1 ms. Overall, the direct performance impact of the added security and privacy functions is acceptable. The biggest bottleneck – the slow TPM – could be removed from the critical processing path. Additionally, TPM commands are executed in parallel to the main CPU and therefore this does not have an influence on the image processing blocks.

## 4 Concluding Remarks and Outlook

Security and privacy protection are crucial properties of video surveillance systems, since they capture and process sensitive and private information. In this chapter we presented an overview of existing privacy protection and security solutions. A key observation is that there is still a lack of approaches that consider security and privacy in video surveillance in a holistic way. It is apparent that most security solutions are situated at the application level and that node-centric security is not taken into account. A lot of work has been targeted at data- and network-centric security. But without taking the security of camera devices themselves into account, high-level protection mechanisms are literally built on sand.

With bringing security and privacy protection onto camera devices, one can achieve reasonable protection against attacks on data that is delivered by surveillance cameras. However, only limited protection is applied for data while it is on the camera. It is an open research topic to identify suitable approaches for on-device data protection. One potential approach is to bring security and privacy protection even closer to the data source by integrating dedicated security functions into the image sensor. If security and privacy are guaranteed at the sensor level, then the camera and its relatively large software stack would no longer have to be considered as trusted entities. This approach contains two main challenges: First it is unclear what type of privacy protection is suitable and feasible at the sensor level. Second, sensor-level privacy protection means that image processing and analysis applications on the camera must be adapted to deal with pre-processed and prefiltered data. A related question is if and how privacy protection can be objectively measured. Since privacy depends on personal as well as cultural attitudes, technical approaches alone will be insufficient. A thorough exploration of the privacy protection design space will also have to involve extensive user surveys to determine which privacy protection techniques are appropriate.

**Acknowledgements** This work was performed as part of the project *TustEYE: Trustworthy Sensing and Cooperation in Visual Sensor Networks*[1]. The work was supported by funding from the European Regional Development Fund (ERDF) and the Carinthian Economic Promotion Fund (KWF) under grant KWF-3520/23312/35521.

## References

1. M.G. Albanesi, M. Ferretti, and F. Guerrini. A Taxonomy for Image Authentication Techniques and its Application to the Current State of the Art. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 535–540, 2001.
2. Pradeep K. Atrey, Wei-Qi Yan, Ee-Chien Chang, and Mohan S. Kankanhalli. A Hierarchical Signature Scheme for Robust Video Authentication using Secret Sharing. In *Proceedings of the International Conference on Multimedia Modelling*, pages 330–337, 2004.
3. Pradeep K. Atrey, Wei-Qi Yan, and Mohan S. Kankanhalli. A Scalable Signature Scheme for Video Authentication. *Multimedia Tools and Applications*, 34(1):107–135, 2006.
4. Nadia Baaziz, Nathalie Lolo, Oscar Padilla, and Felix Petngang. Security and Privacy Protection for Automated Video Surveillance. In *Proceedings of the International Symposium on Signal Processing and Information Technology*, pages 17–22, 2007.
5. Franco Bartolini, Anastasios Tefas, Mauro Barni, and Ioannis Pitas. Image Authentication Techniques for Surveillance Applications. *Proceedings of the IEEE*, 89(10):1403–1418, 2001.
6. Terrance Edward Boult. PICO: Privacy through Invertible Cryptographic Obscuration. In *Proceedings of the Workshop on Computer Vision for Interactive and Intelligent Environments*, pages 27–38, 2005.

[1] TrustEYE website: http://trusteye.aau.at

7. Michael Boyle, Christopher Edwards, and Saul Greenberg. The Effects of Filtered Video on Awareness and Privacy. In *Proceedings of the Conference on Computer Supported Cooperative Work*, pages 1–10, 2000.
8. Michael Bramberger, Josef Brunner, Bernhard Rinner, and Helmut Schwabach. Real-Time Video Analysis on an Embedded Smart Camera for Traffic Surveillance. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 174–181, 2004.
9. Jack Brassil. Using Mobile Communications to Assert Privacy from Video Surveillance. In *Proceedings of the Parallel and Distributed Processing Symposium*, page 8, 2005.
10. CARE Consortium. CARE - Ambient Assisted Living: Safe Private Homes for Elderly Persons. http://care-aal.eu/. last visited: April 2011.
11. Andrea Cavallaro. Adding Privacy Constraints to Video-Based Applications. In *Proceedings of the European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, page 8, 2004.
12. Andrea Cavallaro. Privacy in Video Surveillance. *IEEE Signal Processing Magazine*, 24(2):168–169, March 2007.
13. David Challener, Kent Yoder, Ryan Catherman, David Safford, and Leendert van Doorn. *A Practical Guide to Trusted Computing*. IBM Press, 2008.
14. Ankur Chattopadhyay and Terrance Edward Boult. PrivacyCam: A Privacy Preserving Camera Using uClinux on the Blackfin DSP. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
15. Datong Chen, Yi Chang, Rong Yan, and Jie Yang. Tools for Protecting the Privacy of Specific Individuals in Video. *EURASIP Journal on Applied Signal Processing*, 2007(1):107–116, 2007.
16. Sen-Ching Samson Cheung, Jithendra K. Paruchuri, and Thinh P. Nguyen. Managing Privacy Data in Pervasive Camera Networks. In *Proceedings of the International Conference on Image Processing*, pages 1676–1679, 2008.
17. Sen-Ching Samson Cheung, Jian Zhao, and M Vijay Venkatesh. Efficient Object-Based Video Inpainting. In *Proceedings of the International Conference on Image Processing*, pages 705–708, 2006.
18. Kenta Chinomi, Naoko Nitta, Yoshimichi Ito, and Noboru Babaguchi. PriSurv: Privacy Protected Video Surveillance System Using Adaptive Visual Abstraction. In *Proceedings of the International Multimedia Modeling Conference*, pages 144–154, 2008.
19. Don Davis. Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM, PGP, and XML. In *Proceedings of the USENIX Technical Conference*, pages 65–78, 2001.
20. Danielle Dawson, Patrik Derby, Aaron Doyle, Chiara Fonio, Laura Huey, Mat Johanson, Stéphane Leman-Langlois, Randy Lippert, David Lyon, Anne-Marie Pratte, Emily Smith, Kevin Walby, and Blair Wilkinson. A Report on Camera Surveillance in Canada (Part Two): Surveillance Camera Awareness Network. Technical report, The Surveillance Project, 2009.
21. Frédéric Dufaux and Touradj Ebrahimi. Scrambling for Video Surveillance with Privacy. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshop*, pages 160–166, 2006.
22. Frédéric Dufaux and Touradj Ebrahimi. A Framework for the Validation of Privacy Protection Solutions in Video Surveillance. In *Proceedings of the International Conference on Multimedia and Expo*, pages 66–71, 2010.
23. Dan Farmer and Charles C Mann. Surveillance Nation (Part I). *Technology Review*, 4:34–43, 2003.
24. Douglas A. Fidaleo, Hoang-Anh Nguyen, and Mohan Trivedi. 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*, pages 46–53, 2004.
25. Sven Fleck and Wolfgang Straßer. Smart Camera Based Monitoring System and its Application to Assisted Living. *Proceedings of the IEEE*, 96(10):1698–1714, 2008.
26. Sven Fleck and Wolfgang Straßer. Towards Secure and Privacy Sensitive Surveillance. In *Proceedings of the International Conference on Distributed Smart Cameras*, page 7, 2010.

27. Gary L Friedman. The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image. *IEEE Transactions on Consumer Electronics*, 39(4):905–910, 1993.
28. Ralph Gross, Latanya Sweeney, Fernando De Torre, and Simon Baker. Model-Based Face De-Identification. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshop*, page 8, 2006.
29. Dajun He, Qibin Sun, and Qi Tian. A Secure and Robust Object-Based Video Authentication System. *EURASIP Journal on Advances in Signal Processing*, 2004(14):2185–2200, 2004.
30. Frank Helten and Bernd Fischer. What do people think about CCTV? Findings from a Berlin Survey. Technical report, Berlin Institute for Social Research, 2004.
31. Stefan Krempl and Andreas Wilkens. Datenschützer beanstanden Videoüberwachung in ECE-Einkaufszentren. http://heise.de/-1187205, 2011. last visited: March 2012.
32. Guangzhen Li, Yoshimichi Ito, Xiaoyi Yu, Naoko Nitta, and Noboru Babaguchi. Recoverable Privacy Protection for Video Content Distribution. *EURASIP Journal on Information Security*, 2009:11, 2009.
33. Ying Luo, Shuiming Ye, and Sen-Ching Samson Cheung. Anonymous Subject Identification in Privacy-Aware Video Surveillance. In *Proceedings of the International Conference on Multimedia and Expo*, pages 83–88, 2010.
34. Andrew Martin. The Ten Page Introduction to Trusted Computing. Technical Report RR-08-11, Oxford University Computing Laboratory, December 2008.
35. Isabel Martínez-Ponte, Xavier Desurmont, Jerome Meessen, and Jean-François Delaigle. Robust Human Face Hiding Ensuring Privacy. In *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services*, page 4, 2005.
36. Nasir Memon and Ping Wah Wong. Protecting Digital Media Content. *Communications of the ACM*, 41(7):35–43, July 1998.
37. Saraju P. Mohanty. A Secure Digital Camera Architecture for Integrated Real-Time Digital Rights Management. *Journal of Systems Architecture*, 55(10-12):468–480, October 2009.
38. Simon Moncrieff, Svetha Venkatesh, and Geoff West. Dynamic Privacy in Public Surveillance. *IEEE Computer*, 42(9):22–28, September 2009.
39. Steven Ney and Kurt Pichler. Video Surveillance in Austria. Technical report, Interdisciplinary Centre for Comparative Research in the Social Sciences, Austria, 2002.
40. Clive Norris. A Review of the Increased Use of CCTV and Video-Surveillance for Crime Prevention Purposes in Europe. Technical report, Department of Sociological Studies, University of Sheffield, United Kingdom, 2009.
41. Martin Pirker, Ronald Tögl, Daniel Hein, and Peter Danner. A PrivacyCA for Anonymity and Trust. In *Proceedings of the International Conference on Trust and Trustworthy Computing*, pages 101–119, 2009.
42. Faisal Z. Qureshi. Object-Video Streams for Preserving Privacy in Video Surveillance. In *Proceedings of the International Conference on Advanced Video and Signal-Based Surveillance*, pages 442–447, 2009.
43. Jean-Jacques Qusquater, Benoit Macq, Marc Joye, N. Degand, and A. Bernard. Practical Solution to Authentication of Images with a Secure Camera. *Storage and Retrieval for Image and Video Databases*, 3022(1):290–297, 1997.
44. Sk. Md. Mizanur Rahman, M. Anwar Hossain, Hussein Mouftah, A. El Saddik, and Eiji Okamoto. A Real-Time Privacy-Sensitive Data Hiding Approach based on Chaos Cryptography. In *Proceedings of the International Conference on Multimedia and Expo*, pages 72–77, 2010.
45. Mukesh Saini, Pradeep K. Atrey, Sharad Mehrotra, Sabu Emmanuel, and Mohan S. Kankanhalli. Privacy Modeling for Video Data Publication. In *Proceedings of the International Conference on Multimedia and Expo*, pages 60–65, 2010.
46. Mukesh Saini, Pradeep K. Atrey, Sharad Mehrotra, and Mohan S. Kankanhalli. Anonymous Surveillance. In *Proceedings of the International Workshop on Advances in Automated Multimedia Surveillance for Public Safety*, page 6, 2011.
47. Mukesh Saini, Pradeep K. Atrey, Sharad Mehrotra, and Mohan S. Kankanhalli. Hiding Identity Leakage Channels for Publication of Surveillance Video. *Transactions on Data Hiding and Multimedia Security*, 2011.

48. Mukesh Saini, Pradeep K. Atrey, Sharad Mehrotra, and Mohan S. Kankanhalli. Wˆ3 -Privacy: Understanding What , When , and Where Inference Channels in Multi-Camera Surveillance Video. *Multimedia Tools and Applications: Special Issue on Multimedia Communications, Services Security*, page 22, 2012.
49. Martin Schaffer and Peter Schartner. Video Surveillance: A Distributed Approach to protect Privacy. In *Proceedings of the International Conference on Communications and Multimedia Security*, pages 140–149, 2007.
50. Jeremy Schiff, Marci Meingast, Deirdre K Mulligan, Shankar Sastry, and Kenneth Y Goldberg. Respectful Cameras: Selecting Visual Markers in Real-Time to Address Privacy Concerns. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 971–978, 2007.
51. Andrew Senior, Sharath Pankanti, Arun Hampapur, Lisa Brown, Ying-Li Tian, Ahmet Ekin, Jonathan Connell, Chiao Fe Shu, and Max Lu. Enabling Video Privacy through Computer Vision. *IEEE Security & Privacy Magazine*, 3(3):50–57, 2005.
52. Dimitrios N Serpanos and Andreas Papalambrou. Security and Privacy in Distributed Smart Cameras. *Proceedings of the IEEE*, 96(10):1678–1687, October 2008.
53. Torsten Spindler, Christoph Wartmann, Ludger Hovestadt, Daniel Roth, Luc van Gool, and Andreas Steffen. Privacy in Video Surveilled Areas. In *Proceedings of the International Conference on Privacy, Security and Trust*, page 10, 2006.
54. Suriyon Tansuriyavong and Shinichi Hanaki. Privacy Protection by Concealing Persons in Circumstantial Video Image. In *Proceedings of the Workshop on Perceptive User Interfaces*, page 4, 2001.
55. Texas Instruments. BeagleBoard Website. http://www.beagleboard.org. last visited: April 2011.
56. Juan R. Troncoso-Pastoriza, Luis Pérez-Freire, and Fernando Pérez-González. Videosurveillance and Privacy: Covering the Two Sides of the Mirror with DRM. In *Proceedings of the Workshop on Digital Rights Management*, pages 83–94, 2009.
57. Trusted Computing Group. TCG Software Stack (TSS) Specification Version 1.2 Level 1 Errata A. http://www.trustedcomputinggroup.org/resources/tcg_software_stack_tss_specification, March 2007. last visited: April 2011.
58. Trusted Computing Group. TPM Main Specification 1.2, Level 2, Revision 116. http://www.trustedcomputinggroup.org/developers/trusted_platform_module, July 2007. last visited: March 2012.
59. Hauke Vagts and Alexander Bauer. Privacy-Aware Object Representation for Surveillance Systems. In *Proceedings of the International Conference on Advanced Video and Signal Based Surveillance*, pages 601–608, 2010.
60. Hauke Vagts and Jürgen Beyerer. Security and Privacy Challenges in modern Surveillance Systems. In *Proceedings of the Future Security Research Conference*, pages 94–116, 2009.
61. Jehan Wickramasuriya, Mahesh Datt, Sharad Mehrotra, and Nalini Venkatasubramanian. Privacy Protecting Data Collection in Media Spaces. In *Proceedings of the International Conference on Multimedia*, pages 48–55, 2004.
62. Adam Williams, Deepak Ganesan, and Allen Hanson. Aging in Place: Fall Detection and Localization in a Distributed Smart Camera Network. In *Proceedings of the International Conference on Multimedia*, pages 892–901, 2007.
63. Thomas Winkler and Bernhard Rinner. A Systematic Approach Towards User-Centric Privacy and Security for Smart Camera Networks. In *Proceedings of the International Conference on Distributed Smart Cameras*, page 8, 2010.
64. Thomas Winkler and Bernhard Rinner. 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*, pages 593–600, 2010.
65. Thomas Winkler and Bernhard Rinner. Securing Embedded Smart Cameras with Trusted Computing. *EURASIP Journal on Wireless Communications and Networking*, 2011:20, 2011.
66. Thomas Winkler and Bernhard Rinner. User Centric Privacy Awareness in Video Surveillance. *Multimedia Systems Journal*, 18(2):99–121, 2012.

67. Chung Kei Wong and Simon S. Lam. Digital Signatures for Flows and Multicasts. *IEEE/ACM Transactions on Networking*, 7(4):502–513, 1999.
68. Kenichi Yabuta, Hitoshi Kitazawa, and Toshihisa Tanaka. A New Concept of Security Camera Monitoring with Privacy Protection by Masking Moving Objects. In *Proceedings of the International Pacific-Rim Conference on Multimedia*, pages 831–842, 2005.
69. Shuiming Ye, Ying Luo, Jian Zhao, and Sen-Ching Samson Cheung. Anonymous Biometric Access Control. *EURASIP Journal on Information Security*, 2009:18, 2009.