Sensor-level Security and Privacy Protection by embedding Video Content Analysis

Thomas Winkler
Institute of Networked and Embedded Systems
Alpen-Adria Universitaet Klagenfurt
Lakeside B02b, 9020 Klagenfurt
Email: thomas.winkler@aau.at

Bernhard Rinner
Institute of Networked and Embedded Systems
Alpen-Adria Universitaet Klagenfurt
Lakeside B02b, 9020 Klagenfurt
Email: bernhard.rinner@aau.at

***Abstract*—In traditional video camera networks privacy protection is performed, if at all, at the data center after the video has been streamed over the network. With the on-board processing capabilities of today's visual sensor network devices, image processing and privacy protection can be performed now on the cameras before any data is sent over the network. While moving privacy protection closer to the data source is a big step forward, it raises also new challenges. The growing software stack of today's embedded devices and the widespread use of public wired and wireless network infrastructure entail substantial effort to secure this new class of devices. In this position paper we describe a novel approach that aims to tightly integrate security and privacy protection with the image sensor itself and thereby eliminating many security issues of today's embedded camera systems. Resource-efficient, online video content analysis plays a vital role in this endeavor towards making privacy protection an inherent property of future image acquisition units. This paper outlines the requirements and challenges of such an approach and discusses potential implementation strategies.**

***Index Terms*—security; privacy; visual sensor networks**

I. Motivation and Goals

Visual sensor networks (VSNs) bring together concepts from wireless sensor networks, embedded computing and computer vision [1], [2]. In contrast to traditional multi-camera networks, video processing and analysis are performed in-network instead of on central servers. Furthermore, nodes of a VSN collaborate spontaneously to solve complex tasks such as tracking of persons over long distances. Regardless of the application scenario, images captured by VSNs contain potentially sensitive data. Privacy and data security are therefore critical issues which have long been neglected by designers of camera networks. Privacy becomes especially important when considering that VSNs are deployed not only in public places but also in private environments as in assisted living [3], [4] or home monitoring applications.

From a technical perspective, VSN devices are computing systems based typically on ARM or x86 platforms which run mostly off-the-shelf operating systems such as Embedded Linux or Windows Embedded. On top of these base systems, domain- and application-specific libraries and middleware systems and finally the actual applications are deployed. But it is not only the large software stack which sets VSN devices apart from traditional CCTV systems. While CCTV installations rely on dedicated, closed-circuit communication networks, VSNs additionally use publicly available wired and wireless network infrastructure such as the Internet. Open networks and the large amount of software make VSN devices more vulnerable and more attractive for potential attackers.

To prevent attackers from getting access to sensitive image data that contains not only the identities of persons but also additional information such as behavior, habits or personal preferences, VSN devices have to be secured. In a holistic security approach not only the application level must be addressed but the entire camera down to the operating system and the hardware. We have followed this approach in our preliminary research on TrustCAM [5], [6] where we integrated a Trusted Platform Module (TPM) into a camera prototype where it serves as the anchor for a number of security features including trusted boot, secure image authentication, integrity protection and timestamping. Furthermore, an integral feature of TrustCAM is confidentiality for all outgoing data as well as multi-level privacy protection implemented via separately secured video sub-streams that contain the regions of interest with different levels of protection applied.

The major limitation of the "secure camera" approach of TrustCAM is that large system components such as the operating system, the network stack and a substantial number of system libraries are part of the implicitly trusted software base. While the TPM allows to securely report the exact software versions running on a VSN node it is impossible to provide assurance about the actual security level or security flaws potentially contained in this software. Furthermore, privacy protection and security measures are performed at the application level as part of the computer vision tasks executed on the camera. Consequentially, security and privacy protection are tightly interwoven with the application logic and eventually they are left in the responsibility of the application developers.

The TrustEYE [7] project is designed to overcome these limitations by making security and privacy protection inherent properties of the image sensing unit. The key idea which we present in this position paper is to "protect" access to the sensor similar to the human eye which is a well encapsulated sensory organ. Figure 1 sketches our trustworthy sensing approach with the TrustEYE sensing unit as the key component. The TrustEYE has exclusive access to the sensor's raw data.

It separates sensitive from non-sensitive data by applying dedicated image and video content analysis and ensures that only non-sensitive data is made available to the camera host system. The non-secure camera host system performs further processing and finally transfers camera data via the networking interface to the VSN. The TrustEYE approach clearly separates privacy protection and security functionality from application code. The protection functionality that is integrated into the sensing unit can be kept rather lightweight since neither an OS nor a network stack or middleware libraries are required. Compared to the "secure camera" approach the trustworthy sensing unit significantly reduced the number of implicitly trusted components. The camera host system with its large operating system, the network stack and the system libraries do no longer have to be trusted. Applications executed on the camera host system get access only to pre-processed and protected data. Thus, security and privacy protection remain no longer in the sole responsibility of application developers.

The major contribution of the TrustEYE approach presented in this paper is the tight integration of security and privacy protection with the image acquisition unit. This is a clear advantage over existing solutions since (1) protection can no longer be bypassed, (2) the number of implicitly trusted system components is considerably reduced and (3) application developers are relieved from the burden of integrating security features into their applications.

The rest of this paper is structured as follows: In Section II we summarize related work. Subsequently, in Section III we discuss requirements, challenges and potential implementation approaches for a trustworthy sensing unit. Thereafter, Section IV describes an early TrustEYE prototype implementation. Finally, Section V gives an outlook to future work.

II. Related Work

The main security goal in VSN applications must be the protection of the captured images. Fundamental security considerations have previously been discussed by, e.g., Serpanos and Papalambrou [8] and Senior et al. [9]. In our own work [10], we presented a classification of related work on VSN security.

A. VSN Security and Privacy

This section gives an overview of the basic VSN security and privacy requirements and presents related work.

Image Integrity Protection. Images delivered by a camera can be modified by an attacker during transmission or while stored in a database. Often overlooked is that integrity protection is important not only for single frames but also for sequences. Re-ordering of images can substantially change the meaning of a video. Common integrity protection techniques are checksums and digital signatures [11], [12] as well as watermarks [13], [14].

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

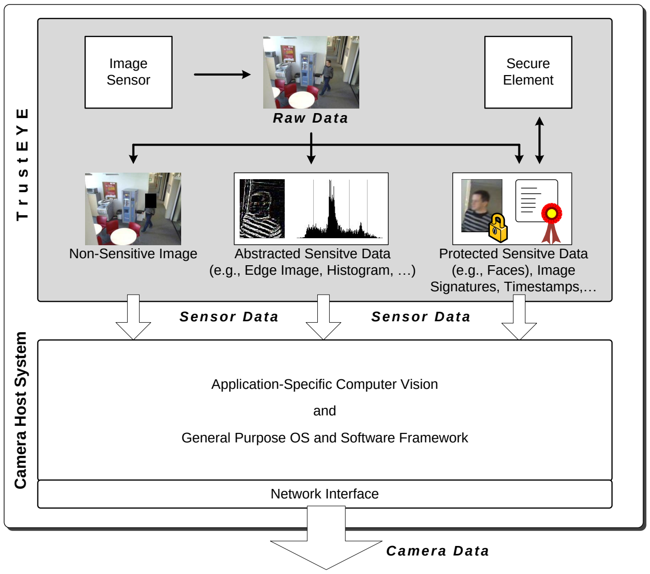
TrustEYE
Image Sensor
Raw Data
Secure Element
Non-Sensitive Image
Abstracted Sensitve Data (e.g., Edge Image, Histogram, …)
Protected Sensitve Data (e.g., Faces), Image Signatures, Timestamps,…
Sensor Data
Sensor Data
Camera Host System
Application-Specific Computer Vision
and
General Purpose OS and Software Framework
Network Interface
Camera Data
Embedded Smart Camera System

Fig. 1. The data flow in a TrustEYE-based smart camera. Raw data from the image sensor is pre-processed within TrustEYE and security as well as privacy protection techniques are applied. The resulting "sensor data" is further processed and analyzed by the camera host system and delivered as "camera data" over the network.

Image Freshness and Timestamping. To prevent replay attacks where recorded videos are injected into the network to replace the live 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 answers not only the question when an image was taken, but at the same time also provides evidence of freshness.

Confidentiality. Confidentiality usually denotes the protection of all data against eavesdropping without differentiation between sensitive and non-sensitive video and image data. Confidentiality is usually achieved by encryption of the communication channel.

Privacy. A topic that is closely related to confidentiality is privacy. The main difference is that most related work on privacy protection focuses on identifying and protecting sensitive image regions such as faces [15]. This approach stems from the desire to protect privacy and, at the same time, keep enough image data visible to facilitate behavioral monitoring. As a consequence, selective protection of image regions is a mechanism that protects sensitive data against misuse be insiders such as system operators or security guards. In contrast to that, the confidentiality requirement denotes protection of all data delivered by the camera against illegitimate access by outsiders.

The most basic form of privacy protection is blanking where sensitive regions are simply removed from captured images. To avoid blank areas, removed regions can be filled with background as proposed by Cheung et al. [16]. Alternatively, obfuscation and scrambling techniques can be applied to sensitive regions. Examples include mosaicking, pixelation or blurring [17], [18]. Protection can be performed also as part of image compression as suggested by Dufaux and Ebrahimi [19] who scramble sensitive regions in the transform domain. Abstraction is another approach for protecting sensitive image data. Bounding boxes, stick figures, silhouettes [9] or avatars are used to replace detected persons. Simple privacy protection techniques usually go hand in hand with the loss of identities. Boult [20], [21] argues that, depending on the application domain, the identities of persons should be recoverable under controlled conditions. He suggests to combine privacy protection with encryption of sensitive data such that identities can be recovered by, e.g., the police. Similar ideas are suggested by Cavallaro [22] and Cheung [23].

The choice of a protection technique depends on the application and the involved goals. Blanking protects both behavior and identity – only the presence of persons remains perceptible. Obfuscation and scrambling allow to monitor the behavior of persons and are therefore more suitable for safety applications where not only presence of persons but also detection of unusual behavior is important. Regardless of the chosen protection technique two key question remain the same: (1) is privacy adequately protected by the chosen technique and (2) what is the impact on the utility of the visual sensor network. Work by Gross et al. [24] indicates that the overall protection capabilities ob pixelation and blurring are relatively low. In more recent work, Dufaux and Ebrahimi [25] present a framework for the evaluation of privacy protection mechanisms. Their results show also that simple pixelation and blurring offer only limited protection. Blurred or pixelated human faces can often still be identified with standard face recognition algorithms. In contrast to that, scrambling mechanisms perform much better. A study by Boyle et al. [26] on the effects of filtered video on awareness and privacy shows that pixelation provides better privacy protection than blurring. Studies by Korshunov et al.[27], [28] indicates that pixelation yields best performance in terms of balance between privacy protection and intelligibility of the video content. Best privacy protection and least intelligibility was achieved with masking filters. Blurring filters resulted in exactly the opposite performance – best intelligibility and least privacy protection.

Image 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 [29], [20], [8]. If a video stream contains different levels of information (e.g., full video, annotations, etc.), access should be managed separately for each level.

B. Sensor-level Security

Sensor-level security mechanisms have been presented by various researchers. In contrast to TrustEYE, the focus so far was mainly on providing integrity and authenticity guarantees.

Early work on real-time image watermarking has been presented by De Strycker et al. [30]. In their approach, the authors use a TriMedia digital signal processor to embed an invisible, digital watermark into video frames in real-time. The watermark consists of a pseudo-noise pattern that depends on a secret key. The system is evaluated in the context of a video broadcasting application where it provides authenticity guarantees for delivered video streams. Nelson et al. [31] take these ideas one step further and propose a CMOS image sensor with built-in watermarking capabilities. In their concept, every image sensor is equipped with a unique, secret key which is used to generate pseudo-random noise serving as watermark. To verify image authenticity, a recipient has to know the sensor's secret key.

Stifter et al. [32] suggest to integrate a secure storage for a symmetric, cryptographic key into the image sensor. This key is used in an on-chip crypto unit as part of message authentication code (MAC) computations. With this setup, the authors are able to provide integrity and authenticity guarantees for delivered data.

Mohanty and Adamo [33], [34] describe a secure digital camera system that provides integrity, authenticity and ownership guarantees for digital video content. This is achieved by using a combination of watermarking and encryption techniques. A binary watermark image is encrypted with a user-supplied key before it is embedded into the image. A custom hardware prototype based on an FPGA demonstrates the feasibility of the approach while meeting the real-time performance requirements. Karthigaikumar and Baskaran [35] focus not only on real-time performance but also on low power consumption of their ASIC implementation. These requirements are met with their custom watermarking algorithm design.

Most work on securing image data at the sensor level has focused so far on providing integrity protection and authenticity guarantees. The most common implementation form is by embedding watermarks into the captured data [36].

TrustEYE advances the state of the art by additionally incorporating strong confidentiality and privacy protection techniques. Only pre-processed data is made available to applications on the non-secure camera host system. At the same time, TrustEYE maintains the flexibility of an embedded smart camera system. Applications on the camera host system can be easily modified and updated by developers without the need for re-integration of security and privacy protection mechanisms since they are inherently provided by the TrustEYE sensing unit.

III. Sensor-level Security and Privacy Protection

The main idea of TrustEYE is to separate system components that have access to raw, unprotected image data from those components that do not need this low-level access. To bring protection as close to the sensitive data as possible, the vision of TrustEYE is to incorporate security and privacy protection into the image sensor itself. Specifically, we defined the following requirements for the TrustEYE sensing unit:

- *Integration of security functionality into the image sensing unit.* Image acquisition and data security (i.e., timestamping (freshness), authenticity and integrity protection for image data) should be integrated as one single inseparable unit. To provide meaningful authenticity guarantees a unique ID will be required per sensing unit.

- *Privacy protection as a feature of the image sensing unit.* Similar to security also privacy protection should be an inherent feature of the image sensing unit which can not be bypassed. Only anonymized data is passed on to the camera host system for further processing. In addition to the anonymized image data, also statistical or abstracted data could be provided by the sensing unit to the camera host system.
- *Strong boundary protection.* A strong requirement of the TrustEYE concept is that the boundary between the sensing unit and the camera host system is strictly protected. It must be ensured by design that the camera host system can not get access to potentially sensitive, unprotected data stored temporarily within the TrustEYE sensing unit.
- *Controlled flexibility.* The requirements for the security and privacy protection techniques that are applied inside the TrustEYE sensing unit might differ depending on application context and environment (e.g., local legislative regulations). Therefore, the behavior of the TrustEYE sensing unit should be adaptable to these requirements. However, adaptation must be technically limited to legitimate parties (e.g., governmental institutions).

To satisfy these requirements, a holistic concept for a trustworthy sensing unit has to address a number of issues. Specifically, we see the following major challenges:

- *Protection of primary and secondary identifiers.* Privacy in VSNs and video surveillance applications is still a relatively vague term. A common approach to achieve privacy protection is anonymization of image and video data by removing or distorting image regions that contain personally identifiable information. While the face of a person is the most obvious identifier also other aspects such as gender, race, gait or items carried by the person could lead to identification and hence a breach of privacy. We call identifiers directly related to an individual primary identifiers. But it is not only these primary identifiers that are important for privacy protection. Secondary identifiers are related to the environment of a person and include location (where), performed actions (what) and time (when) [37]. A person entering a particular office every day at the same time might very well be the owner of the office. Even if primary identifiers are well protected, secondary identifiers might give away sufficient information to reliably identify a person. While protection of secondary identifiers is already a complex task, it becomes even more difficult in multi-camera scenarios where correlation of secondary identifiers across a number of spatially distributed cameras might lead to even more reliable identification of individuals.
- *Privacy vs. system utility tradeoff.* A fundamental goal of a VSN system is to be able to observe behavior of monitored individuals. Applied anonymization and the need for behavior monitoring must be balanced such that the overall utility of the VSN does not degrade severely. Due to specific application requirements, different regional laws and different cultural attitudes there will be no single best approach but a continuum of solutions.
- *Protection techniques for resource-limited devices.* To be economically feasible, a secure sensing unit will have to be very lightweight with respect to required resources. At the same time, realtime requirements must be met. Protection techniques must be chosen and designed to meet these constraints.
- *Correlation of sensor and camera data.* The camera host system serves as the execution environment for user-designed applications. Depending on the specific application, data generated by user applications (e.g., detected events) will have to be securely linked to the original image data.
- *Privacy and security in multi-camera scenarios.* Joint tasks such as person tracking across multiple cameras raise special security challenges. In these scenarios information such as features of tracked persons have to be provided to other nodes of the network. In these situations the sender requires guarantees that the receiver provides at least the same level of security. On the other hand, the receiver needs assurance that received data is authentic and unmodified and comes from a trustworthy source.

We see different technical approaches towards fulfilling the previously outlined requirements and addressing the involved challenges. For the secure sensing unit potential approaches range from an ASIC over custom SoC solutions to software-based techniques such as virtualization. Subsequently, we discuss these approaches in more detail.

- *Secure Sensor (ASIC).* An application specific integrated circuit (ASIC) that combines an image sensor, memory, and logic for security and privacy protection in a single chip has many advantages for the implementation of a secure sensing unit. With all components in a single IC and no interconnects to, e.g., external RAM, it becomes difficult for an attacker to access potentially sensitive raw sensor data. Using custom logic, an ASIC can be optimized for high performance and the external interface of the IC can be designed to mimic those of existing sensor interface standards such that it can be used as a direct “drop-in” sensor replacement. Disadvantages of an ASIC are the high development effort and the limited flexibility. Once the design is complete, implemented algorithms can not be changed or updated. Only limited adaptation is possible via parameters that configure integrated hardware blocks.
- *Secure Sensing Unit (SoC).* A system on chip (SoC) design of a secure sensor unit could be built around an existing SoC which is augmented with custom firmware and external peripherals. In contrast to an ASIC, inte-

gration of the sensor and the attached protection unit is less tight. Interconnects between individual components need special protection which could be achieved, e.g., by encrypting all data that is stored in external RAM. Moreover precautions during PCB design can be taken which make attacks more difficult. These include the use of ball grid components and traces that are not routed on top or bottom layers of the PCB. Furthermore, package on package techniques where, e.g., RAM is stacked directly on top of the SoC make attacks more difficult. Mechanical protection of the sensing unit can be achieved by sealing the unit with, e.g., epoxy resin. Another important aspect is that security mechanisms must be integrated that ensure that only pre-certified, authentic and unmodified firmware can be loaded into the SoC. Similar to an ASIC, also a SoC implementation of a secure sensing unit can expose the same interfaces as an image sensor and can therefore be used as a direct sensor replacement. A major advantage of an SoC stems from its high flexibility. Even after the hardware design is completed, the behavior of the SoC can be updated via firmware.

- *Software-based Separation (Virtualization).* The approach with the highest flexibility is based on virtualization. While virtualization techniques were developed originally for servers and desktop computers, virtualization is now also available for embedded devices. With virtualization, the system is logically partitioned into isolated execution domains with different privileges. The concept of trustworthy sensing could be implemented by having one domain that has exclusive access to the image sensor and is responsible for security and privacy protection measures. The second domain does not have access to the image sensor but only to pre-filtered data provided by the secure domain. Strict separation of the domains can be enforced via the underlying hypervisor and available hardware security extensions such as ARM TrustZone [38]. A virtualization-based solution requires more setup effort and knowledge from developers than a ASIC or a SoC-based "drop-in" solution. The performance of a virtualized secure sensing unit is equivalent to that to the camera host system.

IV. Early Feasibility Study

From the potential design approaches outlined in Section III we have chosen an SoC-based approach for an early TrustEYE feasibility study. The prototype is based on an ARM Cortex M4 SoC with custom firmware. The chosen STM32F417 chip is clocked at 168MHz, offers 192kB of onboard SRAM and 1MB of flash memory. The SoC comes with a variety of interfaces including a dedicated camera interface, SPI, I2C, several UARTs and an external memory interface. Furthermore, the chip is equipped with an on-board crypto unit and supports the implementation of a secure boot procedure which is essential for the realization of a firmware-based TrustEYE sensor unit. A block diagram of the TrustEYE sensing unit prototype is shown in Figure 3.

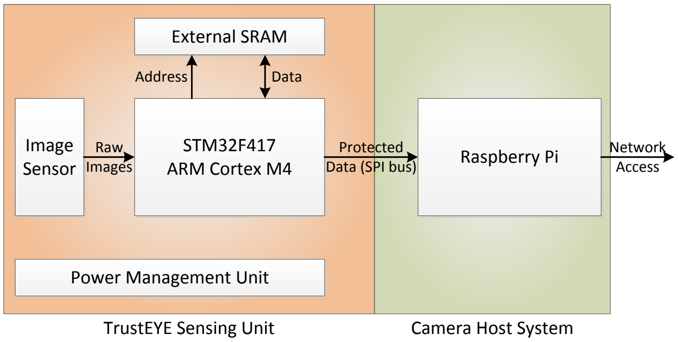

Fig. 2. Block diagram of the early TrustEYE hardware feasibility study. The TrustEYE sensing unit consists of a STM32F417 Cortex M4 microcontroller, external SRAM and an image sensor. The sensing unit delivers data to the camera host system via the SPI bus.

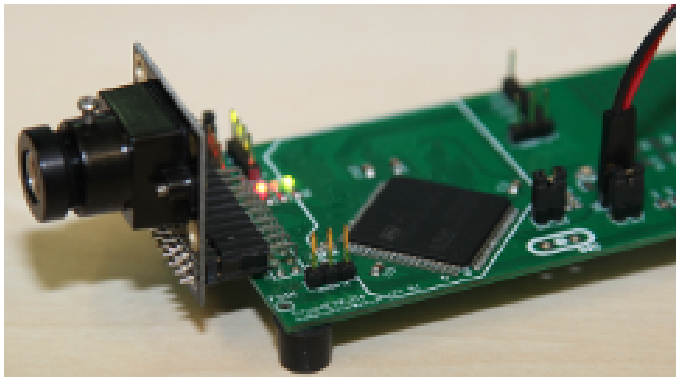

Fig. 3. Photo of an early TrustEYE feasibility study. The custom-designed circuit board carries an STM32F417 ARM Cortex M4 microcontroller (168 MHz, 1 MB Flash, 192 kB SRAM) external SRAM and a camera module. In this feasibility study the connection to the camera host system is established via SPI operating at 32 MHz.

Figure 4 shows a picture of the first assembled prototype of the TrustEYE sensing unit which is currently used for design validation and firmware prototyping tasks. At this early development stage, the camera host system is implemented with a Raspberry Pi[1] evaluation board. The Raspberry Pi is running a standard Linux distribution and is used to deliver data received from the TrustEYE sensing unit via the Ethernet to information consumers. The connection between the TrustEYE prototype system and the Raspberry Pi is implemented via the Serial Peripheral Interconnect (SPI) which is operated at 32 MHz. Data flow is unidirectional from TrustEYE to Raspberry Pi which ensures that the Raspberry Pi camera host system can not randomly access raw data temporarily stored within the TrustEYE sensing unit.

V. Outlook

Work on the secure TrustEYE sensing unit is still at an early stage. A first SoC-based prototype system has been developed which will serve as the basis for the evaluation of the secure sensing unit concept. Important next steps are the integration

[1] Raspberry Pi Website: http://www.raspberrypi.org

of security and privacy protection techniques into the SoC's software framework. This work will go hand in hand with the development of a concept for securing the SoC's firmware and the implementation of a faster connection to camera host system that replaces the currently used SPI bus.

Acknowledgment

This work was performed in the project *TustEYE: Trustworthy Sensing and Cooperation in Visual Sensor Networks* [7]. It was partly funded by the European Regional Development Fund (ERDF) and the Carinthian Economic Promotion Fund (KWF) under grant KWF-3520/23312/35521.

References

[1] S. Soro and W. B. Heinzelman, "A Survey of Visual Sensor Networks," *Advances in Multimedia*, vol. 2009, p. 21, May 2009.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.

[3] H. Aghajan, J. C. Augusto, C. Wu, P. Mccullagh, and J.-A. Walkden, "Distributed Vision-Based Accident Management for Assisted Living," in *Proceedings of the International Conference on Smart Homes and Health Telematics*, 2007, pp. 196–205.

[4] S. Fleck and W. Straßer, "Smart Camera Based Monitoring System and its Application to Assisted Living," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1698–1714, 2008.

[5] T. Winkler and B. 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*, 2010, pp. 593–600.

[6] ——, "Securing Embedded Smart Cameras with Trusted Computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, p. 20, 2011.

[7] T. Winkler, "Website of Project TrustEYE: Trustworthy Sensing and Collaboration in Visual Sensor Networks," http://trusteye.aau.at, 2012, last visited: February 2013.

[8] D. N. Serpanos and A. Papalambrou, "Security and Privacy in Distributed Smart Cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1678–1687, Oct. 2008.

[9] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, A. Ekin, J. Connell, C. F. Shu, and M. Lu, "Enabling Video Privacy through Computer Vision," *IEEE Security & Privacy Magazine*, vol. 3, no. 3, pp. 50–57, 2005.

[10] T. Winkler and B. Rinner, "A Systematic Approach Towards User-Centric Privacy and Security for Smart Camera Networks," in *Proceedings of the International Conference on Distributed Smart Cameras*, 2010, p. 8.

[11] P. K. Atrey, W.-Q. Yan, and M. S. Kankanhalli, "A Scalable Signature Scheme for Video Authentication," *Multimedia Tools and Applications*, vol. 34, no. 1, pp. 107–135, 2006.

[12] M. Hefeeda and K. Mokhtarian, "Authentication Schemes for Multimedia Streams: Quantitative Analysis and Comparison," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 6, no. 1, pp. 1–24, Feb. 2010.

[13] N. Memon and P. W. Wong, "Protecting Digital Media Content," *Communications of the ACM*, vol. 41, no. 7, pp. 35–43, Jul. 1998.

[14] S. P. Mohanty and E. Kougianos, "Real-Time Perceptual Watermarking Architectures for Video Broadcasting," *Journal of Systems and Software*, vol. 84, no. 5, pp. 724–738, May 2011.

[15] I. Martínez-Ponte, X. Desurmont, J. Meessen, and J.-F. Delaigle, "Robust Human Face Hiding Ensuring Privacy," in *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services*, 2005, p. 4.

[16] S.-C. S. Cheung, J. Zhao, and M. V. Venkatesh, "Efficient Object-Based Video Inpainting," in *Proceedings of the International Conference on Image Processing*, 2006, pp. 705–708.

[17] K. Chinomi, N. Nitta, Y. Ito, and N. Babaguchi, "PriSurv: Privacy Protected Video Surveillance System Using Adaptive Visual Abstraction," in *Proceedings of the International Multimedia Modeling Conference*, 2008, pp. 144–154.

[18] J. Wickramasuriya, M. Datt, S. Mehrotra, and N. Venkatasubramanian, "Privacy Protecting Data Collection in Media Spaces," in *Proceedings of the International Conference on Multimedia*, 2004, pp. 48–55.

[19] F. Dufaux and T. Ebrahimi, "Scrambling for Video Surveillance with Privacy," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshop*, 2006, pp. 160–166.

[20] T. E. Boult, "PICO: Privacy through Invertible Cryptographic Obscuration," in *Proceedings of the Workshop on Computer Vision for Interactive and Intelligent Environments*, 2005, pp. 27–38.

[21] A. Chattopadhyay and T. E. Boult, "PrivacyCam: A Privacy Preserving Camera Using uClinux on the Blackfin DSP," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[22] A. Cavallaro, "Privacy in Video Surveillance," *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 168–169, 2007.

[23] S.-C. S. Cheung, J. K. Paruchuri, and T. P. Nguyen, "Managing Privacy Data in Pervasive Camera Networks," in *Proceedings of the International Conference on Image Processing*, 2008, pp. 1676–1679.

[24] R. Gross, L. Sweeney, F. D. Torre, and S. Baker, "Model-Based Face De-Identification," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshop*, 2006, p. 8.

[25] F. Dufaux and T. Ebrahimi, "A Framework for the Validation of Privacy Protection Solutions in Video Surveillance," in *Proceedings of the International Conference on Multimedia and Expo*, 2010, pp. 66–71.

[26] M. Boyle, C. Edwards, and S. Greenberg, "The Effects of Filtered Video on Awareness and Privacy," in *Proceedings of the Conference on Computer Supported Cooperative Work*, 2000, pp. 1–10.

[27] P. Korshunov, C. Araimo, F. D. Simone, C. Velardo, J.-L. Dugelay, and T. Ebrahimi, "Subjective Study of Privacy Filters in Video Surveillance," in *Proceedings of the International Workshop on Multimedia Signal Processing*, 2012, p. 5.

[28] P. Korshunov, S. Cai, and T. Ebrahimi, "Crowdsourcing Approach for Evaluation of Privacy Filters in Video Surveillance," in *Proceedings of the International Workshop on Crowdsourcing for Multimedia*, 2012, p. 6.

[29] M. Schaffer and P. Schartner, "Video Surveillance: A Distributed Approach to protect Privacy," in *Proceedings of the International Conference on Communications and Multimedia Security*, 2007, pp. 140–149.

[30] L. De Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes, and G. Depovere, "Implementation of a Real-time Digital Watermarking Process for Broadcast Monitoring on a TriMedia VLIW Processor," *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 147, no. 4, p. 371, 2000.

[31] G. Nelson, G. Jullien, and O. Yadid-Pecht, "CMOS Image Sensor with Watermarking Capabilities," in *International Symposium on Circuits and Systems*, 2005, pp. 5326–5329.

[32] P. Stifter, K. Eberhardt, A. Erni, and K. Hoffmann, "Image Sensor for Security Applications with On-chip Data Authentication," *Proceedings of the Society of Photo-Optical Instrumentation Engineers*, vol. 6241, p. 8, 2006.

[33] S. P. Mohanty, "A Secure Digital Camera Architecture for Integrated Real-Time Digital Rights Management," *Journal of Systems Architecture*, vol. 55, no. 10-12, pp. 468–480, Oct. 2009.

[34] O. Adamo, S. Mohanty, E. Kougianos, and M. Varanasi, "VLSI Architecture for Encryption and Watermarking Units Towards the Making of a Secure Camera," in *Proceedings of the International System-on-Chip Conference*, Sep. 2006, pp. 141–144.

[35] P. Karthigaikumar and K. Baskaran, "FPGA and ASIC Implementation of Robust Invisible Binary Image Watermarking Algorithm Using Connectivity Preserving Criteria," *Microelectronics Journal*, vol. 42, no. 1, pp. 82–88, Jan. 2011.

[36] E. Kougianos, S. P. Mohanty, and R. N. Mahapatra, "Hardware Assisted Watermarking for Multimedia," *Computers & Electrical Engineering*, vol. 35, no. 2, pp. 339–358, Mar. 2009.

[37] M. Saini, P. K. Atrey, S. Mehrotra, and M. S. Kankanhalli, "W3-Privacy: Understanding what, when, and where inference channels in multi-camera surveillance video," *Springer International Journal on Multimedia Tools and Applications*, no. August, p. 24, 2012.

[38] ARM Limited, "ARM Security Technology Building a Secure System using TrustZone Technology," Tech. Rep., 2009.