Demo: TrustEYE.M4 – A novel Platform for Secure Visual Sensor Network Applications

Thomas Winkler and Bernhard Rinner
Institute of Networked and Embedded Systems and Lakeside Labs
Alpen-Adria-Universität Klagenfurt
Lakeside Park B02b, 9020 Klagenfurt
thomas.winkler@aau.at, bernhard.rinner@aau.at

ABSTRACT

Designers of Visual Sensor Network (VSN) platforms face many challenges. Image sensors deliver large amounts of data and substantial computing power and memory are required for processing. At the same time power consumption should be kept low to facilitate battery-powered operation. The increasing deployment of VSNs also raises privacy and security related questions. In this work we present a new VSN platform called TrustEYE.M4 which is designed as a state-of-the-art research platform for the development of secure VSN applications. It integrates an ARM Cortex M4 processor, a WiFi radio, a high-performance image sensor and a dedicated security chip. We present the architecture of TrustEYE.M4 and demonstrate its capabilities with a secure video streaming application.

Categories and Subject Descriptors

K.6.5 [**Management of computing and information systems**]: Security and Protection—*Authentication, Unauthorized access*; I.4.9 [**Image processing and computer vision**]: Applications; C.3 [**Computer Systems Organization**]: Special-purpose and application-based systems—*Real-time and embedded systems*

General Terms

Security, Privacy

Keywords

Visual sensors, Embedded smart cameras, Security, Privacy

1. MOTIVATION AND RELATED WORK

Application domains for Visual Sensor Networks (VSNs) are manifold ranging from surveillance [2] over environmental monitoring to smart homes [1] and assisted living [5]. Depending on the application, data security and privacy protection are important issues [11].

Over the past years, several different VSN platforms have been developed including the low-performance Cyclops by Rahimi et al. [7] or CITRIC by Chen et al. [4] which is equipped with high-performance ARM CPU. Security and privacy protection were addressed only by very few platforms. PrivacyCam by Chattopadhyay and Boult [3] is a camera based on a Blackfin DSP which identifies regions of interest using background subtraction and encrypts them using AES. Mohanty and Adamo [6] describe an FPGA-based camera that provides integrity, authenticity and ownership guarantees for digital video via watermarking and encryption. In our preliminary work on TrustCAM [10] we implement authenticity and integrity guarantees via digital signatures and provide secure timestamping for videos using a dedicated hardware security IC.

Based on our previous experience we take this approach a step further and present TrustEYE.M4 – a modular and extensible VSN platform which is based on a custom-designed circuit board. With its ARM Cortex M4 processor and 4 MB of additional SRAM it provides sufficient resources for many computer vision applications while maintaining a small power budget and enabling standalone, battery-powered scenarios. Subsequently, we present the system architecture of TrustEYE.M4 and demonstrate its capabilities with a secure, privacy-preserving streaming application.

2. SYSTEM ARCHITECTURE

This section presents the hard- and software architecture of the TrustEYE.M4 platform. Initially, we describe the goals and requirements we followed in the design phase:

On-board image analysis. The system provides sufficient computing power for on-board image processing and analysis at frame rates of 10 fps or more. Image resolutions for on-board processing are between 160×120 and 320×240 pixels.

Platform security support. TrustEYE.M4 is designed as a low-cost platform. Nevertheless, components are chosen to support security features such as secure boot and hardware acceleration for bulk data encryption. Furthermore, it includes a dedicated Trusted Platform Module [8] (TPM) which provides a unique platform ID, secure timestamping as well as support for asymmetric cryptography and secure storage for cryptographic keys.

Video streaming capabilities. VSNs are typically designed to deliver object descriptors and high-level events instead of video streams. However, this high-level data may not always be reliable and therefore TrustEYE.M4 can deliver live video streams for verification and monitoring. Resolutions for streaming are higher than those for on-board processing.

Flexibility and modularity. As a platform for research and development, TrustEYE.M4 is modular and supports extension modules (e.g., WiFi, auxiliary sensors) as well as different image sensors. The system is flexible enough to be deployed either in standalone mode or as a secure sensing component in a larger, more powerful camera system [9].

Power-aware system design. Low power consumption is an important requirement in many VSN applications. TrustEYE.M4 is designed such that all major components support low-power operation modes.

2.1 Hardware Architecture

The TrustEYE.M4 CPU board shown in Figure 1 is based on a 50×50 mm custom-designed printed circuit board. It uses an STM32F417 ARM Cortex M4 microcontroller from STMicroelectronics with 192 kB on-chip SRAM and 1 MB on-chip program Flash memory. The controller supports SIMD instructions for parallel data processing as well as DSP instructions. Since the on-chip SRAM is insufficient to hold multiple images as required by many computer vision algorithms, an additional 2×2 MB of external SRAM are added. Figure 2 presents an overview of the core components of the processing board. The system is powered either via a Micro-USB connector or a single-cell lithium polymer (LiPo) battery. An on-board BQ24074 IC is used for system power management and LiPo charging.

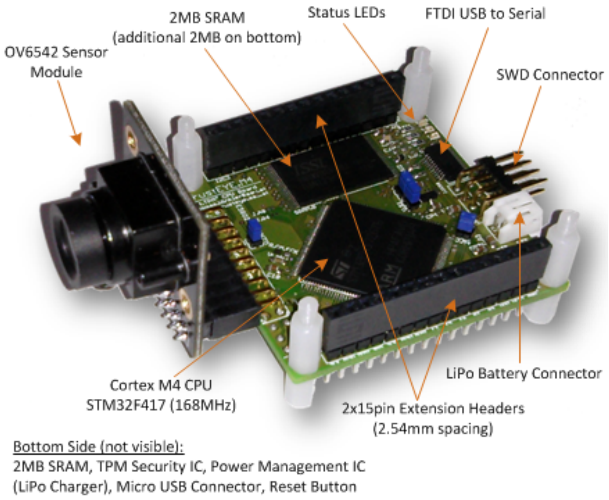
2MB SRAM
(additional 2MB on bottom)
Status LEDs
FTDI USB to Serial
OV6542 Sensor
Module
SWD Connector
Cortex M4 CPU
STM32F417 (168MHz)
LiPo Battery Connector
2x15pin Extension Headers
(2.54mm spacing)
Bottom Side (not visible):
2MB SRAM, TPM Security IC, Power Management IC
(LiPo Charger), Micro USB Connector, Reset Button

Figure 1: The 50×50 mm TrustEYE.M4 CPU board with an OmniVision OV5642 image sensor module.

Programming and debugging is supported via the Serial Wire Debug/Viewer (SWD/SWV) connector and the MCU's serial bootloader accessible via the Micro-USB port. The ability to power, charge, program and debug TrustEYE.M4 via a single Micro-USB port turns it into a convenient platform for both research and education. Figure 2 illustrates the modular design of TrustEYE.M4. The image sensor is connected via a dedicated port and can be easily exchanged. Two 15 pin headers provide access to on-board buses such as I2C and SPI as well as to GPIO pins of the CPU.

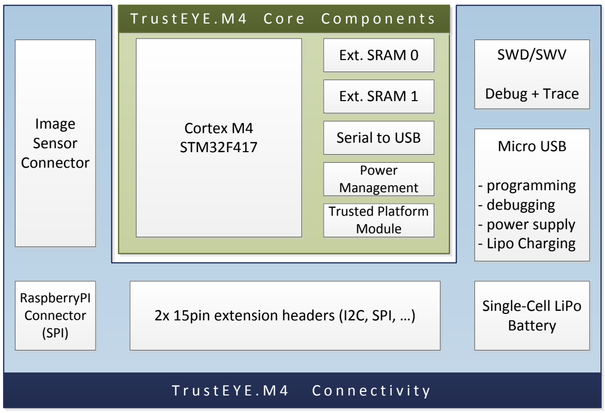

Figure 2: TrustEYE.M4 contains a Cortex M4 CPU (STM32F417, 168 MHz), 4 MB SRAM, an TPM chip, a BQ24074 power management IC and an FT230xs USB to serial converter. External connectivity is implemented via breakout headers.

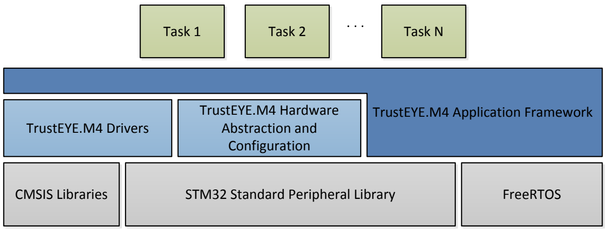
Task 1
Task 2
...
Task N
TrustEYE.M4 Drivers
TrustEYE.M4 Hardware Abstraction and Configuration
TrustEYE.M4 Application Framework
CMSIS Libraries
STM32 Standard Peripheral Library
FreeRTOS

Figure 3: The TrustEYE.M4 software architecture includes drivers, a hardware abstraction and configuration layer and an application framework.

We support two image sensor modules – one with an OmniVision OV7725 sensor (640×480) and one with an OmniVision OV5642 sensor (5 megapixels). Both sensors are configured via the I2C bus and deliver images via a parallel 8-bit bus. Optionally, the OV5642 can deliver JPEG compressed images with embedded thumbnail images at 320×240 in YUV422 format. Data transfers from the image sensor module to SRAM and from SRAM to other peripherals are implemented via the microcontroller's DMA engines such that the CPU itself is available for image processing tasks.

2.2 Software Architecture

Figure 3 presents the software architecture of TrustEYE.M4. The lowest layer consist of the CMSIS (Cortex Microcontroller Software Interface Standard) library from ARM Ltd., the STM32 Standard Peripheral Library from STMicroelectronics which provides access to on-chip peripherals and the FreeRTOS real-time operating system. On the second layer, there is a hardware abstraction layer together with device specific drivers for, e.g., the image sensors, the Trusted Platform Module or the WiFi radio.

The TrustEYE.M4 application framework provides a runtime environment for applications composed of individual tasks scheduled by FreeRTOS. Event tough TrustEYE.M4 uses a single-core processor, it supports a certain degree parallelism via its DMA engines. To ensure that tasks waiting for DMA completion do not block other tasks, a synchronized double-buffering mechanism for data handover between tasks is included in the software framework. Devel-

Operation	Runtime
Meanshift	62 ms
Roberts Cross	11 ms
HMAC-SHA1	1.2 ms
AES256 Encryption	1.7 ms
Total	75.9 ms

Table 1: Runtimes (avg. over 100 frames) for the processing steps of the secure streaming application.

opment is entirely based on Open-Source tools including Eclipse/CDT as IDE, the GNU ARM GCC toolchain and custom tools for USB-based programming.

3. APPLICATION EXAMPLE AND DEMO

To demonstrate the capabilities of TrustEYE.M4 we use a privacy-preserving video streaming application which applies a cartoon-like (see Fig. 4) filter to captured images. This effect is based on a modified version of meanshift filtering which has been specifically adapted to the limited resources of embedded platforms [9]. While the cartoon-like effect is primarily intended as a protection against misuse by insiders (e.g., system operators) how have legitimate access to streamed video, we additionally perform AES encryption and a digital signature (HMAC-SHA1) to ensure confidentiality, integrity and authenticity for the transmitted data.

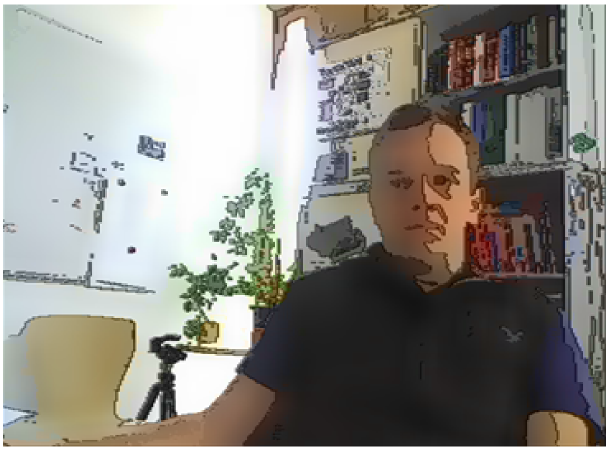

Figure 4: Cartooning effect applied on TrustEYE.M4 before data leaves the camera device.

Table 1 presents average runtimes for the processing steps of the cartooning application. The meanshift filter takes the majority of the runtime followed by Roberts Cross edge detection which is used to emphasize the edges of color regions. Data encryption and SHA1 computation are performed using the hardware accelerators of the STM32F417 CPU and consume relatively little time. Including all overheads, the cartooning application achieves a frame rate of 11 fps.

The demonstration showcases the battery-powered TrustEYE.M4 prototype platform running the cartooning application in realtime. The privacy-protected video stream is transmitted to a tablet device via WiFi for viewing. In the demonstration, the audience gets a hands-on experience with the TrustEYE.M4 platform and the cartooning effect. For comparison, the unmodified video stream is displayed alongside the cartoonized version. An example video is presented in the media section of the TrustEYE website[1].

Acknowledgments

This work was performed as part of project TrustEYE and received funding from the the European Regional Development Fund (ERDF) and the Carinthian Economic PromotionFund (KWF) under grant KWF-3520/23312/35521.

4. REFERENCES

[1] M. Brezovan and C. Badica. A Review on Vision Surveillance Techniques in Smart Home Environments. In *Proceedings of the International Conference on Control Systems and Computer Science*, pages 471–478, 2013.

[2] A. Cavoukian. Surveillance, Then and now: Securing Privacy in Public Spaces. Technical report, 2013.

[3] 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*, pages 1–8, 2007.

[4] P. W.-C. Chen, P. Ahammad, C. Boyer, S.-I. Huang, L. Lin, E. J. Lobaton, M. L. Meingast, S. Oh, S. Wang, P. Yan, A. Yang, C. Yeo, L.-C. Chang, D. Tygar, and S. S. Sastry. CITRIC: A Low-Bandwidth Wireless Camera Network Platform. In *Proceedings of the International Conference on Distributed Smart Cameras*, page 10, 2008.

[5] S. Fleck and W. Straßer. Smart Camera Based Monitoring System and its Application to Assisted Living. *Proceedings of the IEEE*, 96(10):1698–1714, 2008.

[6] S. P. Mohanty. A Secure Digital Camera Architecture for Integrated Real-Time Digital Rights Management. *Journal of Systems Architecture*, 55(10-12):468–480, Oct. 2009.

[7] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. B. Srivastava. Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems*, page 13, 2005.

[8] Trusted Computing Group. TPM Main Specification 1.2, Level 2, Revision 116. `http://www.trustedcomputinggroup.org/resources/tpm\_main\_specification`, July 2011. last visited: July 2014.

[9] T. Winkler, A. Erdélyi, and B. Rinner. TrustEYE.M4: Protecting the Sensor - not the Camera. In *Proceedings of the International Conference on Advanced Video and Signal Based Surveillance*, page 6, 2014. (to appear).

[10] T. Winkler and B. Rinner. Securing Embedded Smart Cameras with Trusted Computing. *EURASIP Journal on Wireless Communications and Networking*, 2011:20, 2011.

[11] T. Winkler and B. Rinner. Security and Privacy Protection in Visual Sensor Networks: A Survey. *ACM Computing Surveys*, 47(1):42, 2014. (in print).

[1] TrustEYE Website: `http://trusteye.aau.at`