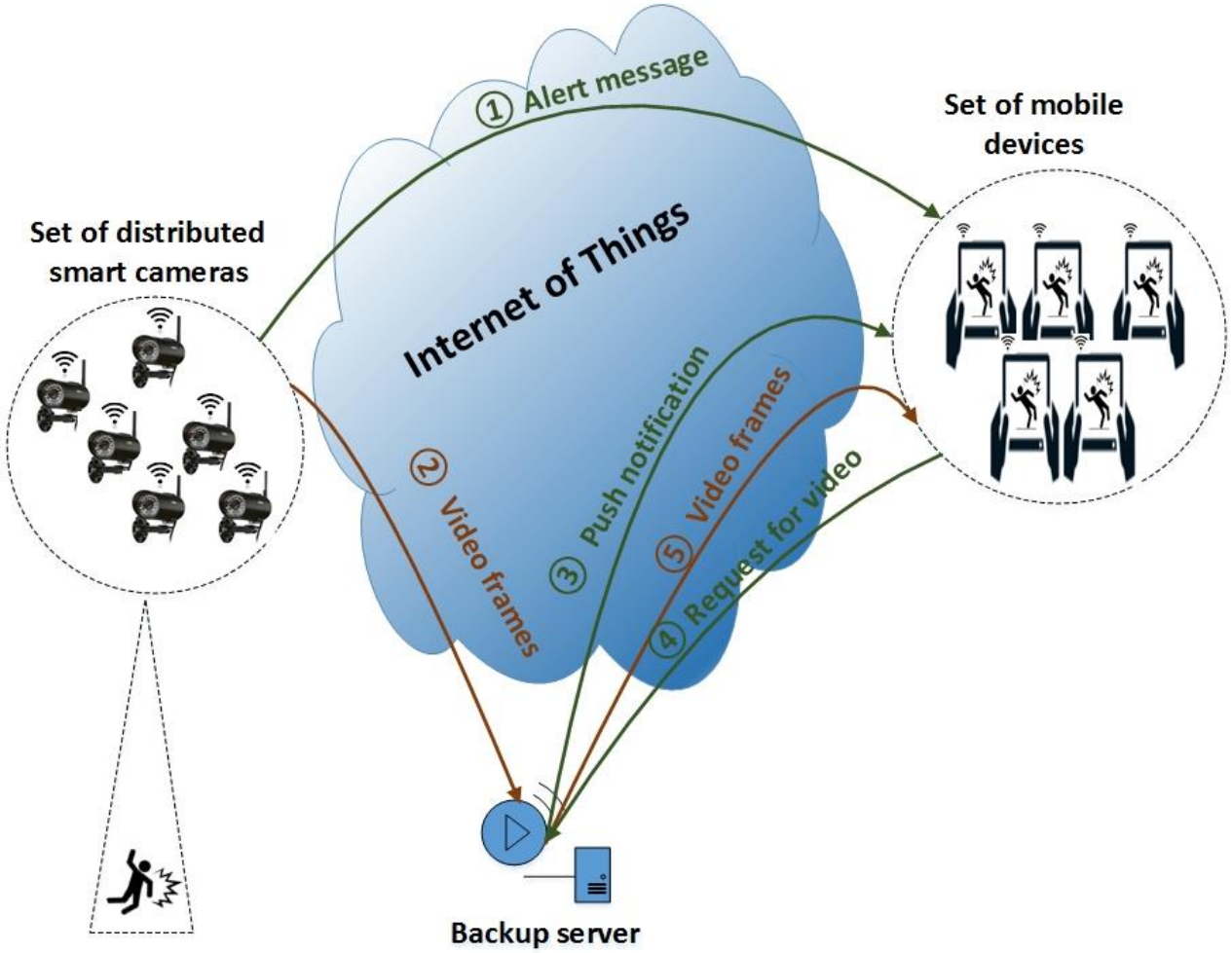
Smart Cameras with onboard Signcryption for securing IoT Applications

Subhan Ullah✶• *Bernhard Rinner*✶ *Lucio Marcenaro*•

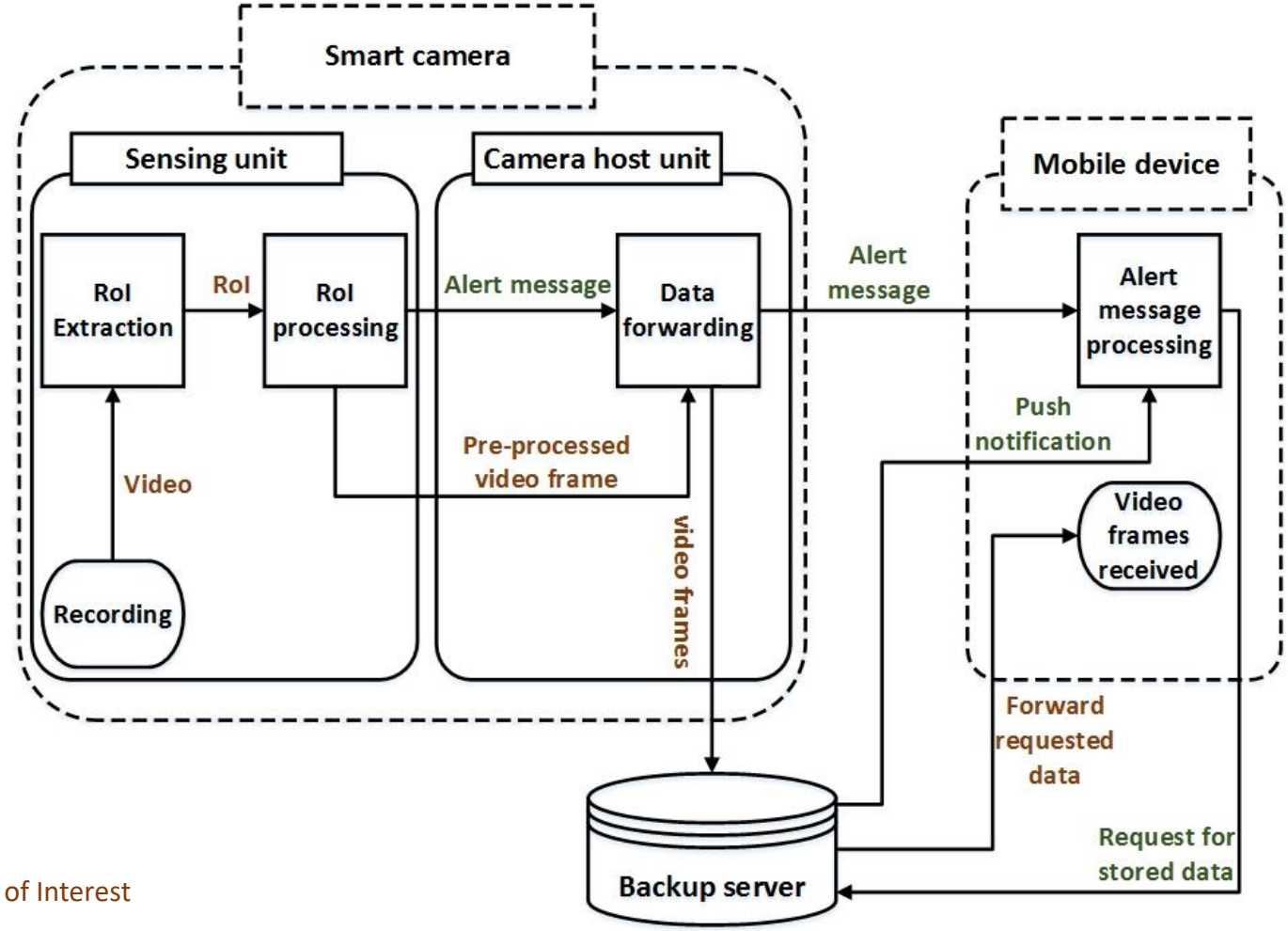
✶Institute of Networked and Embedded Systems,
Alpen-Adria-Universität Klagenfurt, Klagenfurt Austria
{subhan.ullah, bernhard.rinner}@aau.at
•Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture, University of Genova, Genova Italy
lucio.Marcenaro@unige.it

Introduction

- Smart cameras in IoT applications
  - Sensing, processing, communication on a single platform
- Event-triggered monitoring
- Smart cameras captures personal data
- Security mechanisms to ensure confidentiality, authenticity, integrity and freshness of data
- Challenges
  - Real-time performance
  - Resource limitations
  - High volume of data
  - Open infrastructure

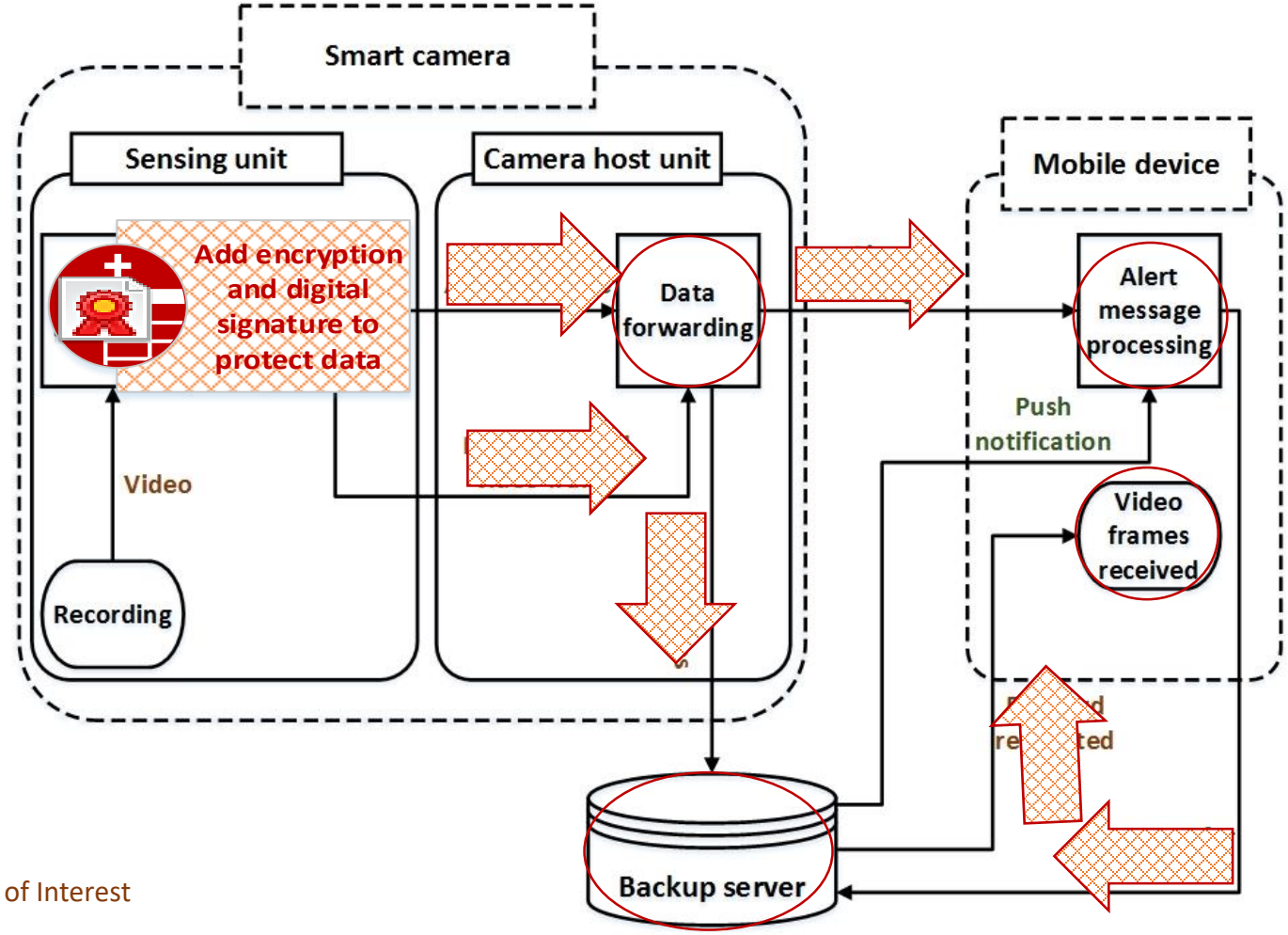


System architecture



ROI: Region of Interest

Security approach



ROI: Region of Interest

Threats model

- Eavesdropping
  - Eavesdropping is a passive attack
  - Can compromises the confidentiality of image data
  - Encryption of images on sensing unit ensure confidentiality
  - Attackers needed decryption key to eavesdropped encrypted data during transmission
- Data modification
  - Attackers can change, inject or delete images on camera host or during the transmission
  - Digital signatures of images on sensing unit ensure integrity
  - Attackers required private key for modification of signed images/video frames

Threats model

- Impersonation
  - Attackers using the identity of a sensing unit to transmit its own images
  - Digital signatures applying to images on sensing unit ensure authentication
  - Attackers required private key of sensing unit to impersonate the data
- Replay attack
  - Correct timestamping provides freshness
  - Attackers can transmit the same valid information repeatedly
  - Or delivers outdated information as fresh one

State-of-the-art (protection of images/videos)

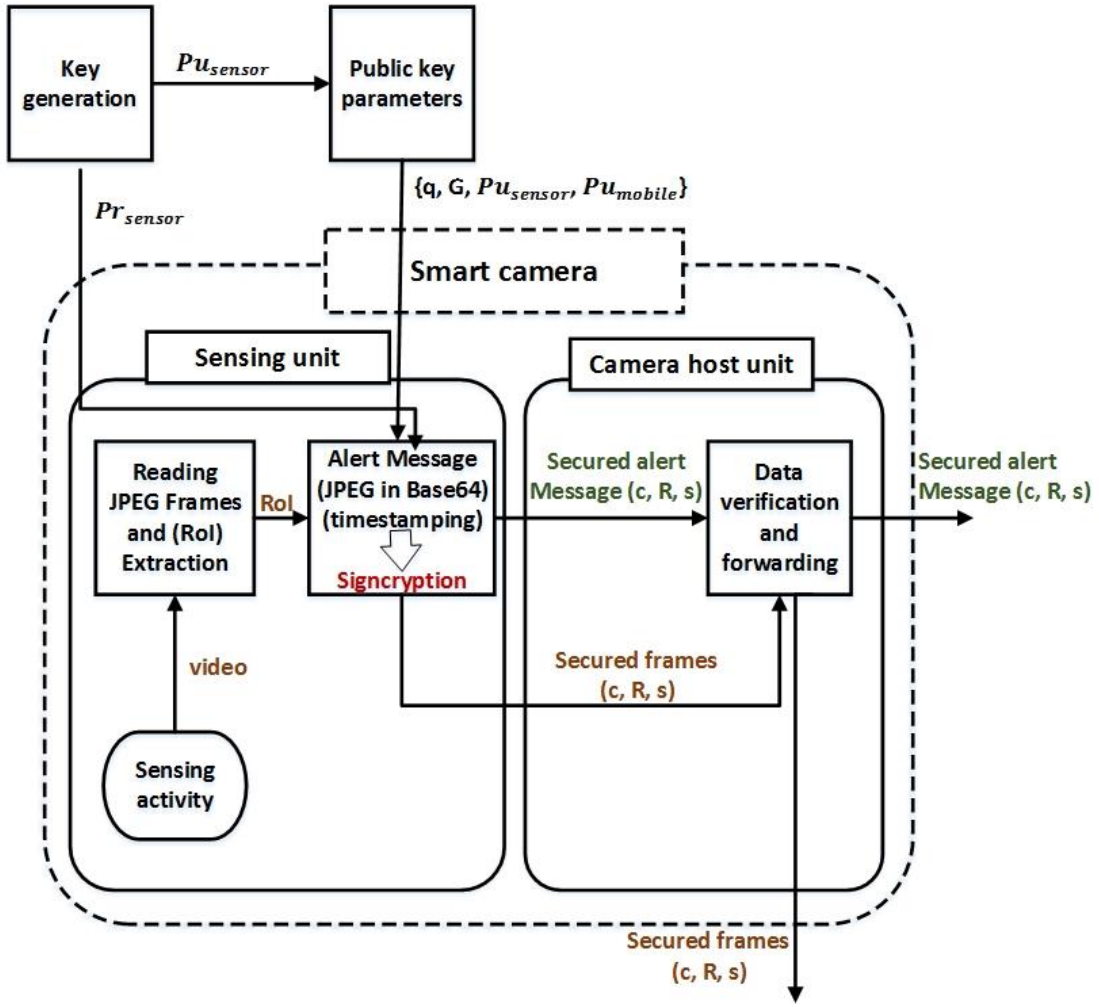
- Image or video security and protection
  - Digital Watermarking approach [V. M. Potdar_2005] [P. W. Wong_1998]
    - Integrity verification
    - Detection of changes in size or pixel values
    - Watermarking computationally expensive for IoT devices
  - Watermarking then AES encryption [S. P. Mohanty_2009]
    - Integrity and confidentiality
    - Computationally expensive for IoT devices
  - Digital signature [P. K. Atrey_2007]
    - Provides authentication and integrity but no confidentiality
  - RSA based digital signature and AES encryption [T. Winkler_2014], [T. Winkler_2015]
    - Less efficiency due to sign-then-encryption way of implementation
    - Large key size required for RSA based signatures

State-of-the-art (security close to sensing unit)

- Security close to visual sensing unit
  - CMOS active pixel sensor (APS) imager [G. R.Nelson_2005]
    - On chip watermarking
    - Pervasive image authentication
    - Authentication and integrity only
  - On-chip cryptographic unit [P. Stifter_2006]
    - Image sensor with EEPROM to uniquely identify the imager
    - Authentication and integrity of image data only
  - Trust EYE.M4 platform [T. Winkler_2015]
    - Hardware based trusted platform module (TPM)
    - Provides onboard security and privacy
  - CMOS image sensor based on PUFs [Y. Cao_2015]
    - Exploiting the dark signal noise uniformity of fixed pattern noise
    - On-chip authentication and identification

Signcryption process

- Signcryption based on Elliptic Curve Discrete Logarithm Problem (ECDLP) [E. Mohamed_2009]
  - $Pu = Pr \,.\, G$
  - Digital signature (ECDSA)
  - Encryption (AES)
- Signcrypted packet (C, R, S)
- Advantages of Signcryption
  - Lightweight and provides equal security as “sign-then-encryption”
  - Public verifiability



Signcryption algorithm

- Key pairs
  - Sensor: Pr_{sensor} , Pu_{sensor}
  - Mobile: Pr_{mobile} , Pu_{mobile}
- Signcryption algorithm

$$v \in \{1, 2, \dots, q-1\}$$

$$k_1 = hash(vG)$$

$$k_2 = hash(vPu_{mobile})$$

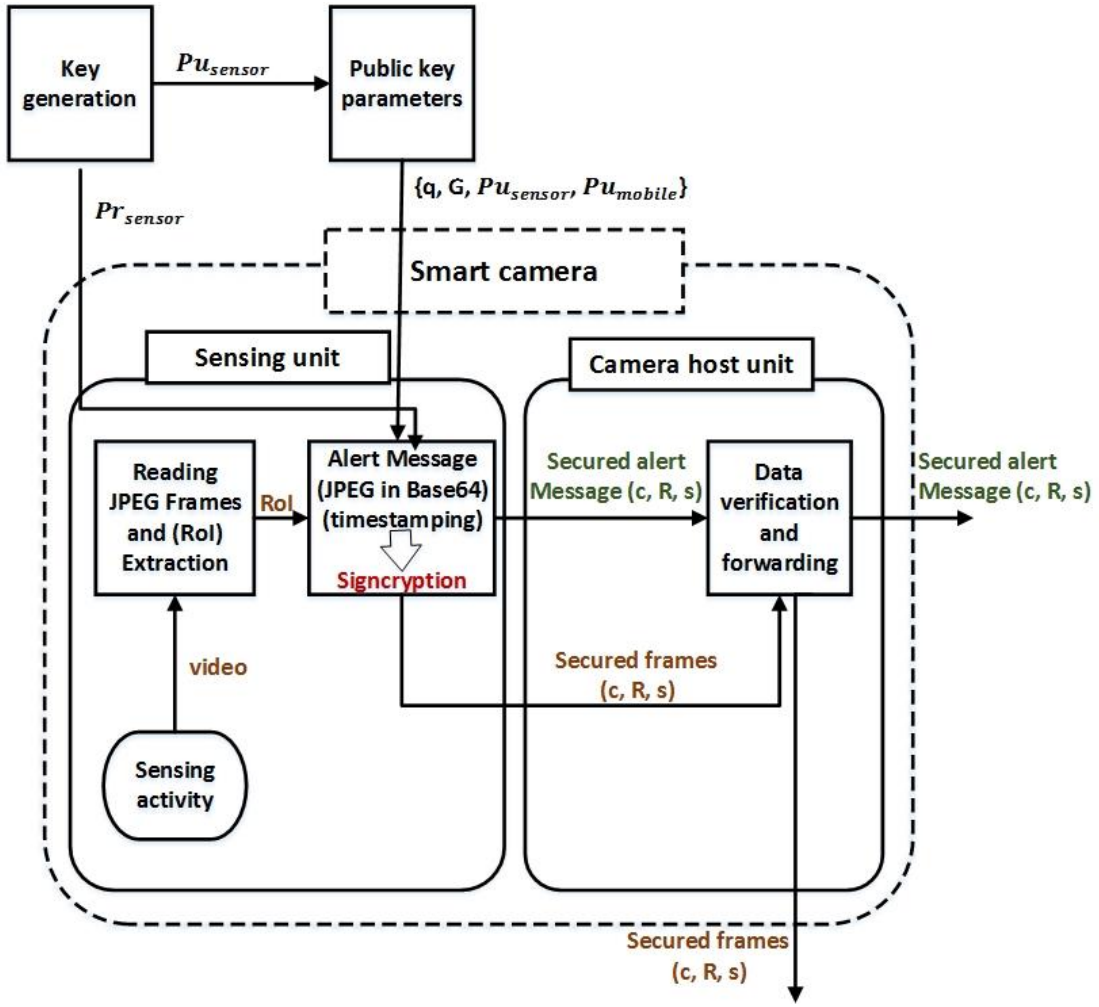
$$c = Ek_2(frames_t)$$

$$r = hash(c, k_1)$$

$$s = \frac{v}{(r + Pr_{sensor})} \; mod \; q$$

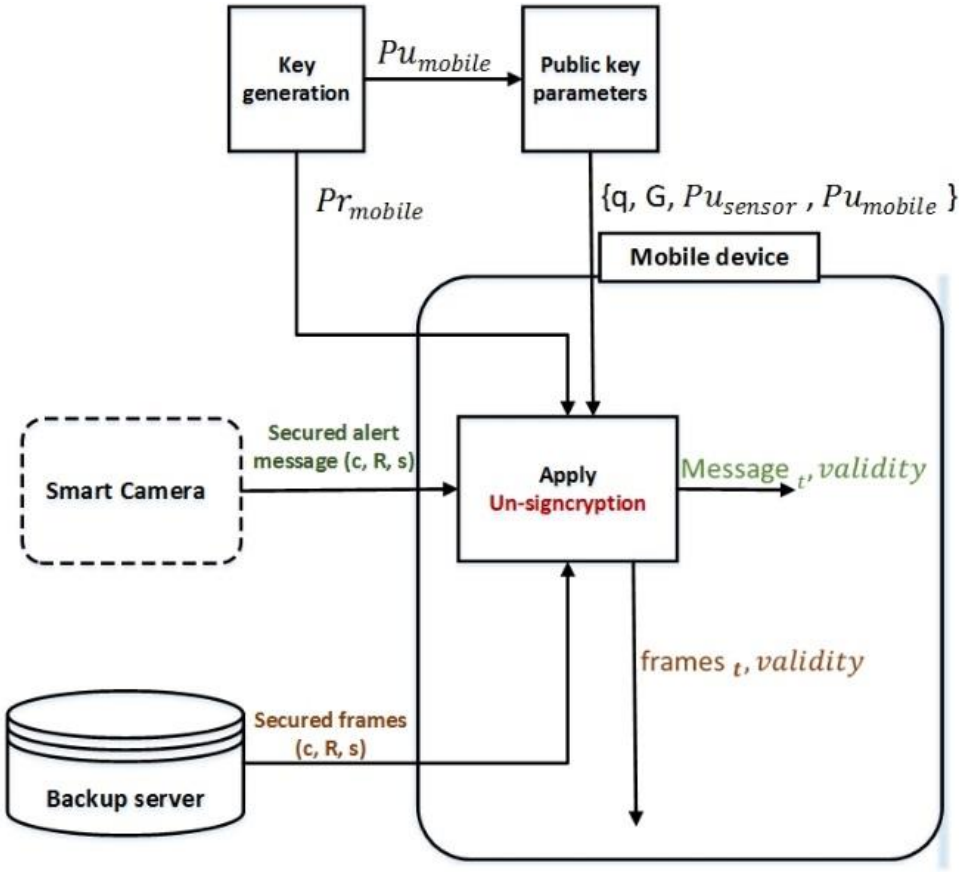
$$R = r\,G$$

Signcryption output $= (c, R, s)$



Unsigncryption process

- Un-signcryption of the image/video frames
  - Visual sensor (Public key)
  - Mobile device (Private key)
  - Public parameters
- Proof of security
  - Authentication
  - Integrity
  - Confidentiality
  - Freshness



Unsigncryption algorithm

- Verification by camera host

  $k_1 = hash\left(s(R + Pu_{sensor})\right)$

  $r = hash\ (c, k_1)$

  $rG = R$ (Public verifiability)

- Un-signcryption by mobile device

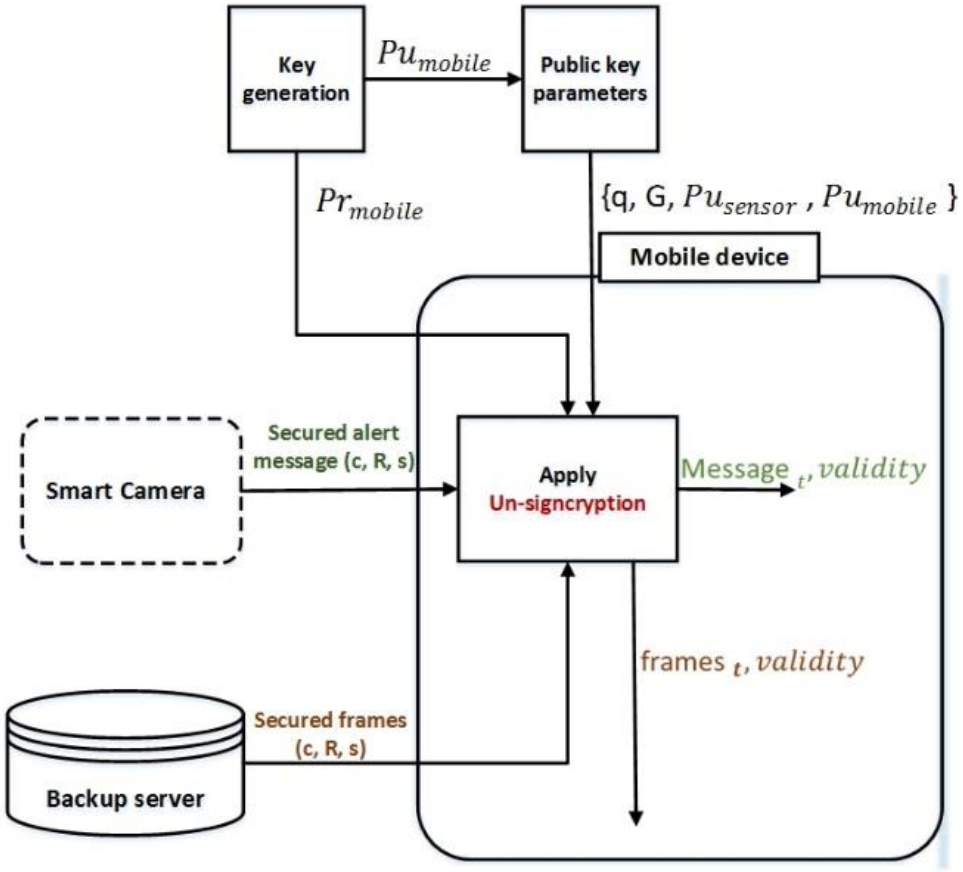
  $k_1 = hash(\mathrm{s}(\mathrm{R}+Pu_{sensor}))$

  $r = hash\ (c, k_1)$

  $k_2 = hash(Pr_{\mathrm{mobile}}\ s(R + Pu_{sensor}))$

  $frames_t = Dec\ k_2\ (c)$

  $rG = R$ (Validated)



Security analysis and countermeasures

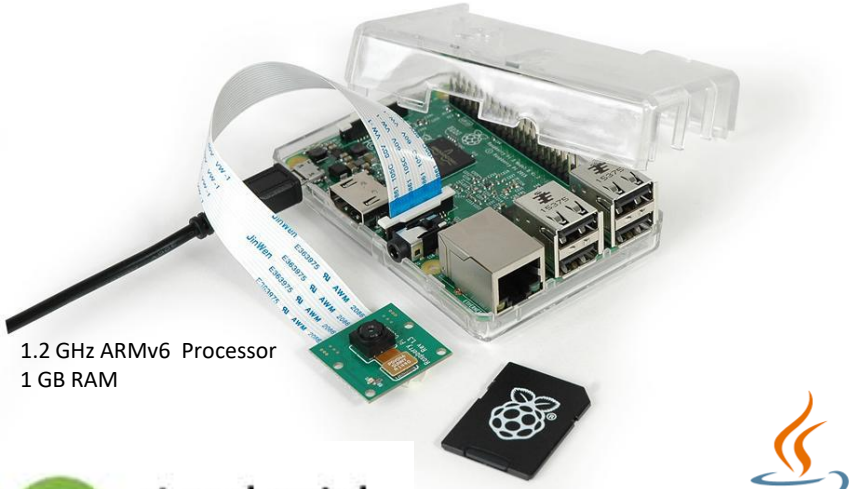
- Security analysis of signcryption with respect to system architecture
- Security of the signcryption technique is based on the computational hardness of ECDLP
- Countermeasures
  - Confidentiality
    - Confidentiality provides by AES encryption in signcryption model
    - Attackers need private key (Pr_{mobile}) of the mobile device to derive AES key (k_2) in the proposed security model
    - Private key of mobile device is secured under the assumption of computational hardness of ECDLP

Security analysis and countermeasures

- Countermeasures
  - Authenticity and Integrity
    - Private key (Pr_{sensor}) of sensing unit provides authentication and integrity by generating digital signature
    - Public key (Pu_{sensor}) of sensing unit is used to verify signed images
    - Attackers need private key (Pr_{sensor}) of sensing unit to modify the signed images
    - Private key of sensing unit is also secured on the assumption of computational hardness of ECDLP
  - Freshness
    - Timestamping of data before signcryption provides freshness to the images

Experimental setup

- Raspberry Pi-3 platform
  - Pi camera captures images
  - Java package of EC-based signcryption is used for security
  - Implementation performed on Raspberry Pi-3 platform
- Experiments
  - Two different experiments has performed by varying image and key sizes
- Measured the efficiency of EC-signcryption and unsigncryption

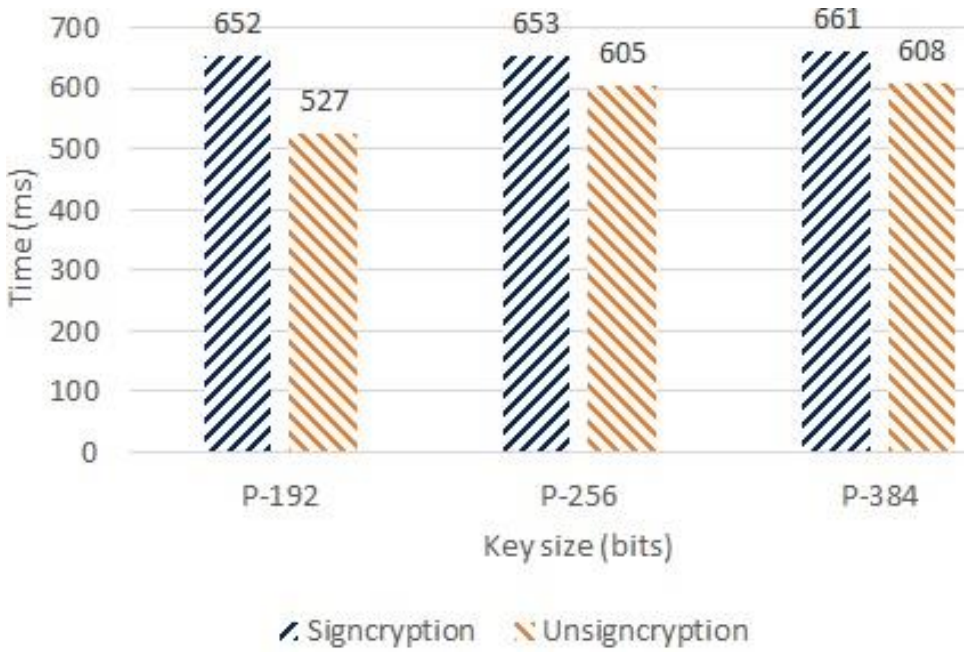


Android Studio

Java

Results (experiment-1)

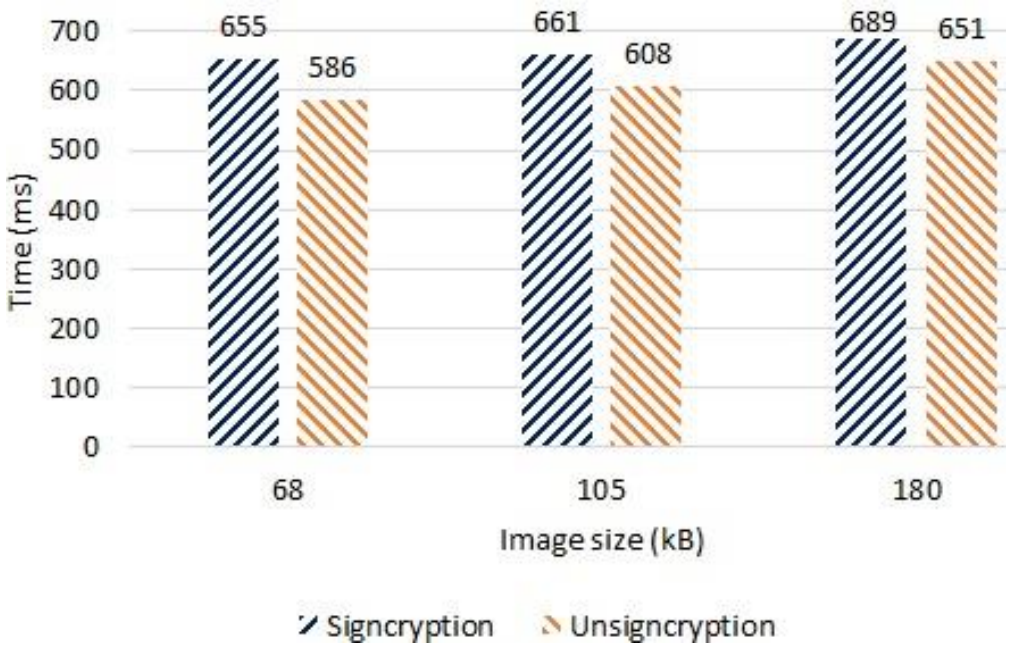
- Different EC-keys of 192, 256, 384 bits are used
- Apply to same image size 105 kB
- AES session key size of 256 bits are used for encryption
- Efficiency of the signcryption and unsigncryption



Running time of signcryption and unsigncryption with different EC keys for an 480 x 320 image with a size of 105 kB.

Results (experiment-2)

- Different image sizes 68, 105, 180 kB are used
- Apply same EC-key of P-384 bits
- AES session key size of 256 bits are used for encryption
- Efficiency results of signcryption and unsigncryption



Running time of signcryption and unsigncryption with different image sizes using an EC P-384 bits key

Conclusion

- Protection of (image/video) data for event triggered monitoring
- EC-based signcryption on a sensing unit
- Identified potential threats and presented countermeasures
- Results shows that EC-based signcryption is resource efficient for implementing on sensing unit

Future directions

- Future directions
  - Physical Unclonable Function (PUFs)
    - Generation of secure and temper proof private keys
  - Extension of the security techniques
    - Safety and security of public premises (city, train-station, airport)
    - Proactive monitoring and the collection of identities and tracking of individuals
- Challenges of future work
    - Privacy of observed people
    - Substantial computation for detection of unusual activities on resource constraint devices

References

Ref.	Full Reference
[V. M. Potdar]	V. M. Potdar, S. Han, and E. Chang, “A survey of digital image watermarking techniques,” in Proc. IEEE International Conference on Industrial Informatics, Aug. 2005, pp. 709–716.
[P. W. Wong]	P. W. Wong, “A public key watermark for image verification and authentication,” in Proc. International Conference on Image Processing, vol. 1, Oct. 1998, pp. 455–459.
[P. K. Atrey]	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, 2007.
[S. P. Mohanty]	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, 2009.
[T. Winkler_2014]	T. Winkler and B. Rinner, “Security and privacy protection in visual sensor networks: A survey,” ACM Comput. Surv., vol. 47, no. 1, pp. 2:1–2:42, May 2014.
[T. Winkler_2015]	T. Winkler and B. Rinner, “Secure embedded visual sensing in end-user applications with TrustEYE.M4,” in Proc. IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Apr. 2015, pp. 1–6.
[P. Stifter_2006]	P. Stifter, K. Eberhardt, A. Erni, and K. Hofmann, “Image sensor for security applications with on-chip data authentication,” in Proc. of the Society of Photo-Optical Instrumentation Engineers, vol. 6241, pp. 8, Apr. 2006
[S. P. Mohanty – 2009]	S. P. Mohanty. A Secure Digital Camera Architecture for Integrated Real-Time Digital Rights Management. Journal of Systems Architecture, pages 468–480, Oct. 2009
[Y. Cao_2015]	Y. Cao, L. Zhang, S. S. Zalivaka, C. H. Chang, and S. Chen, “Cmos image sensor based physical unclonable function for coherent sensor-level authentication,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 11, pp. 2629–2640, Nov. 2015.
[E. Mohamed_2009]	E. Mohamed and H. Elkamchouchi, “Elliptic curve signcryption with encrypted message authentication and forward secrecy,” International Journal of Computer Science and Network Security, vol. 9, no. 1, pp. 395–398, 2009.
[G. R.Nelson_2005]	G. R. Nelson, G. A. Jullien, and O. Yadid-Pecht, “Cmos image sensor with watermarking capabilities,” in Proc. IEEE International Symposium on Circuits and Systems, May. 2005, pp. 5326–5329 Vol. 5.

Thank you!

Questions & Answers