Building a Secure and Privacy-Preserving Smart Camera System

PennState

Systems and Internet Infrastructure Security Laboratory

Yohan BEUGIN

A Thesis in Computer Science and Engineering

The Pennsylvania State University

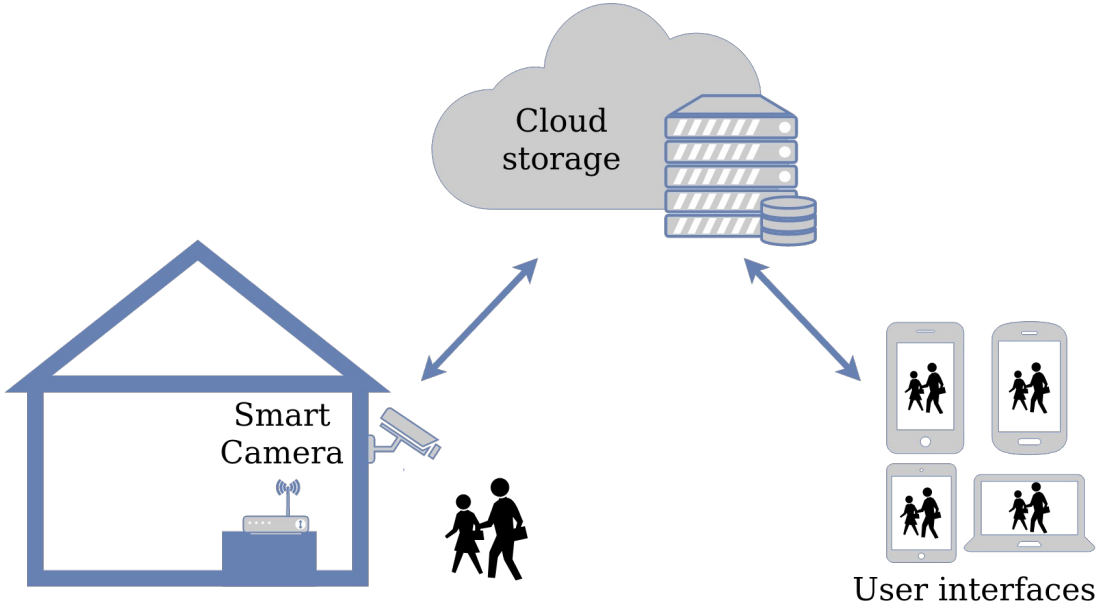
Advised by Dr. Patrick McDaniel

March 17, 2021

Smart Camera System

PennState

Smart Camera: camera connected to the internet (surveillance, doorbell...)



Privacy

Privacy:

- Right to be let alone without interference or intrusion
- Ability for people to keep themselves and information about themselves secret

In a privacy-preserving smart camera system:

- **Control** how the data is recorded
- **Control** who has access to live streaming
- **Control** who can see recorded videos

Motivation

PennState

Senator blasts Amazon's Ring doorbell as an 'open door for privacy and civil liberty violations'

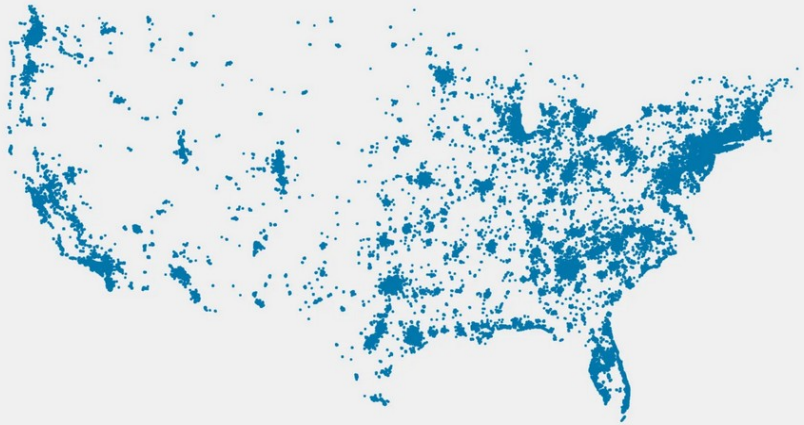
Sen. Ed Markey, D-Mass., called the lack of privacy protections "chilling."

Privacy violations because users have **no control**

Ring partnerships with police: Solving crimes or invading privacy?

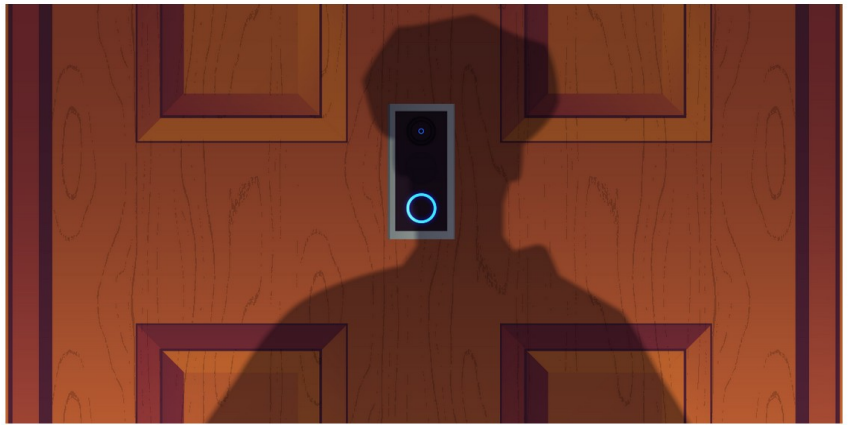
Ring Camera locations in the U.S.

Includes every camera that posted to the Ring Neighbors app since November 2016



Ring Updates Device Security and Privacy—But Ignores Larger Concerns

BY MATTHEW GUARIGLIA AND BILL BUDINGTON | FEBRUARY 18, 2020



Ring and AdHoc Group Against Crime giving away doorbell cameras in KCMO

U.S.

Police Are Monitoring Black Lives Matter Protests With Ring Doorbell Data and Drones

BY KHALEDA RAHMAN ON 8/9/20 AT 10:46 AM EDT

Thesis Statement

Users do not need to give up their privacy to get all the advantages of a smart camera system.

PennState

Roadmap

PennState

Main features of a smart camera system

Threat model

Privacy Challenges

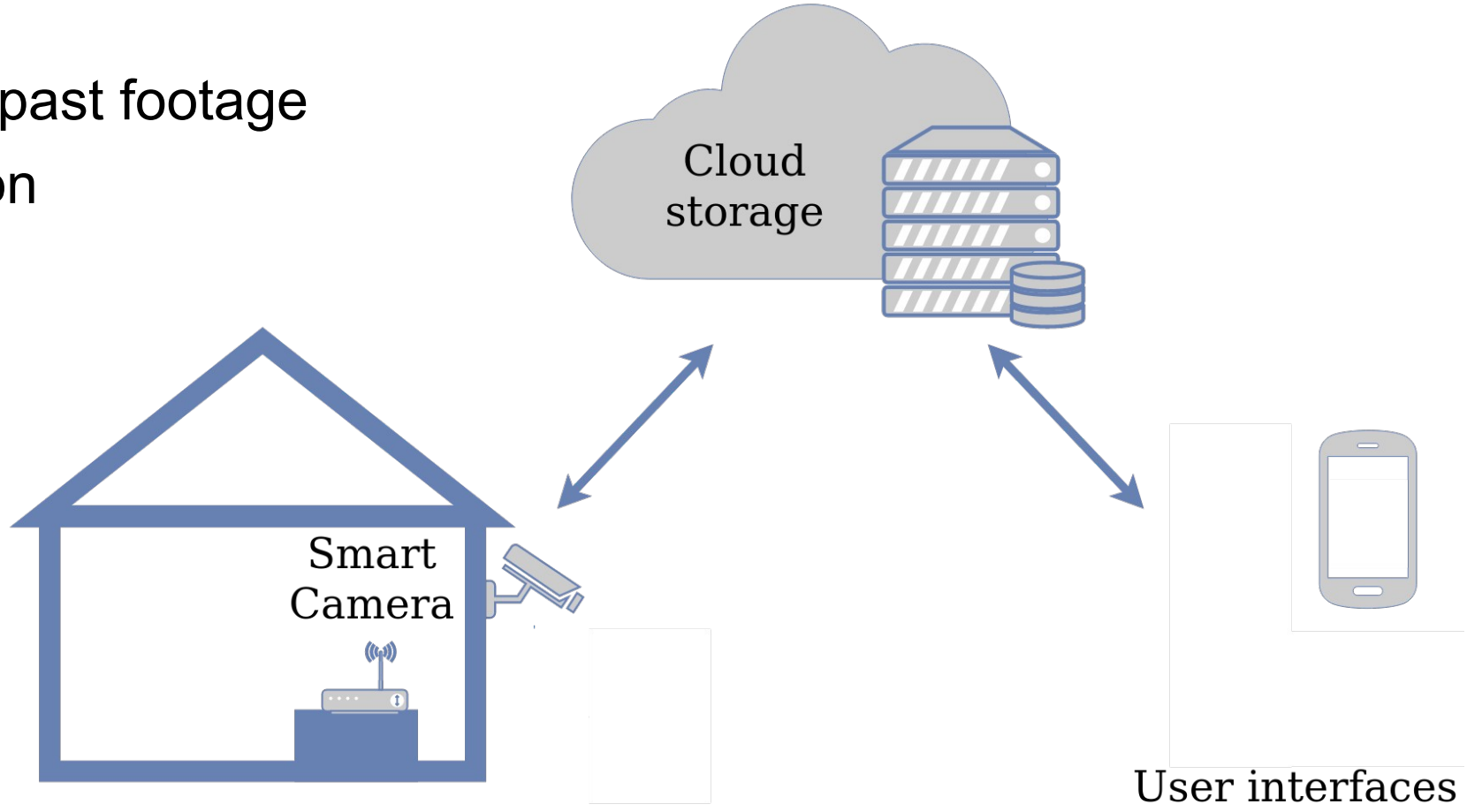
Implementation and demonstration

Evaluation

Main Features of a Smart Camera System

PennState

- Setup
- Live stream and past footage
- Access delegation
- Access recovery
- Factory reset



Roadmap

PennState

Main features of a smart camera system

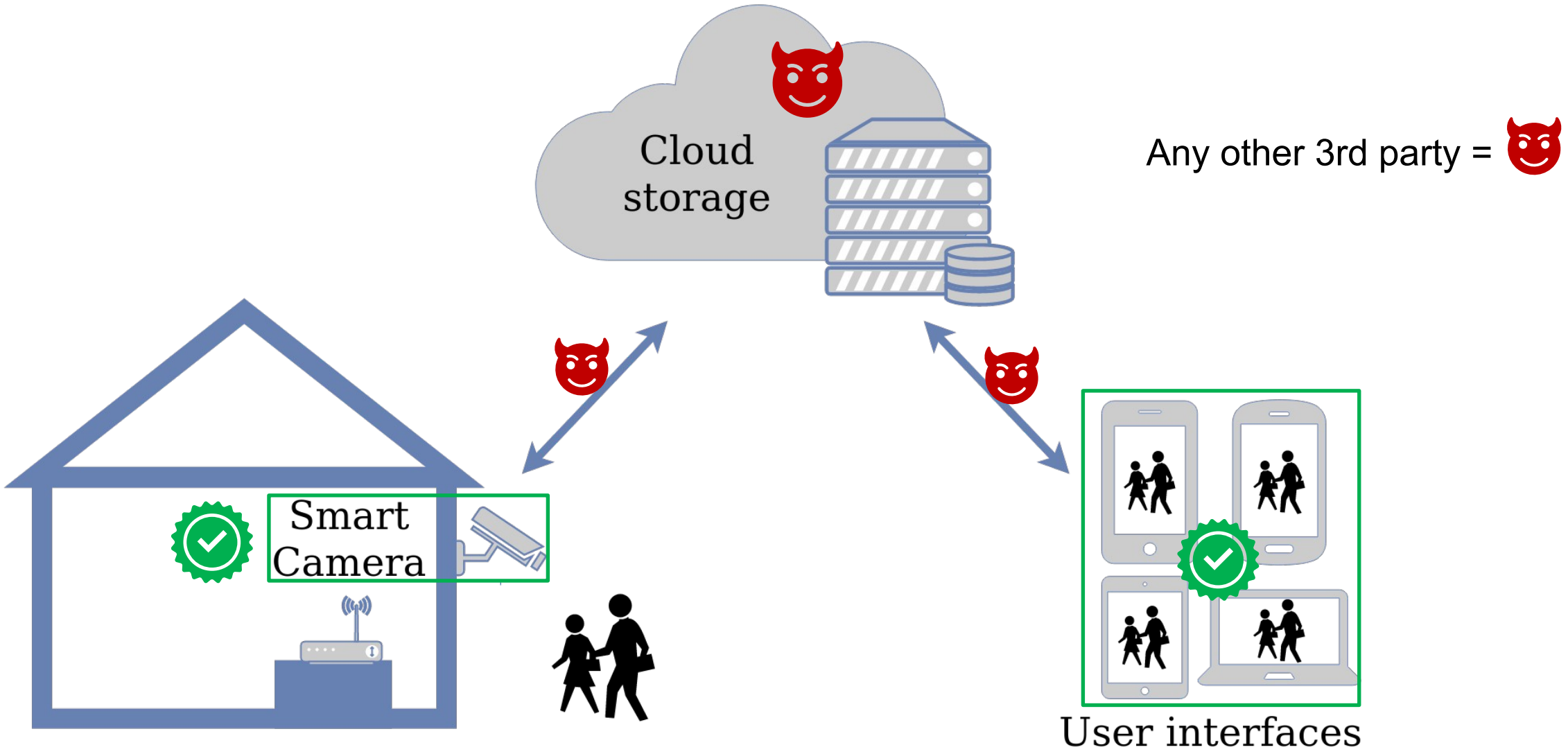
Threat model

Privacy Challenges

Implementation and demonstration

Evaluation

Threat Model



Roadmap

PennState

Main features of a smart camera system

Threat model

Privacy Challenges

Implementation and demonstration

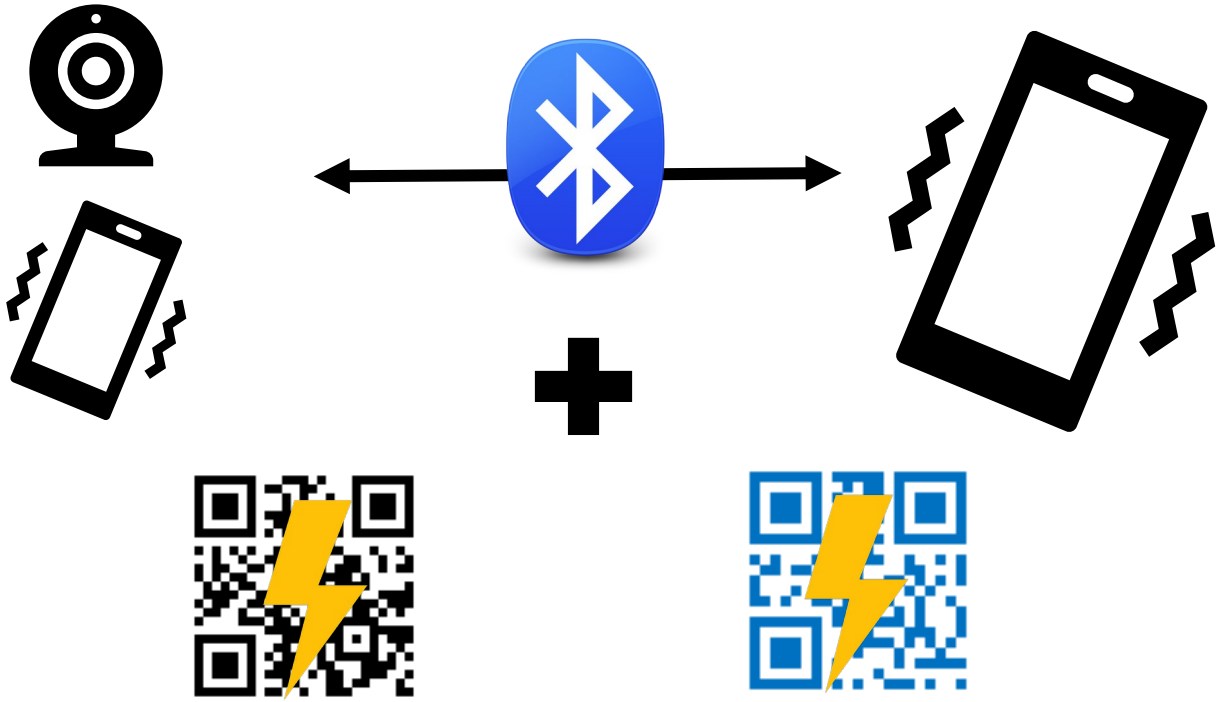
Evaluation

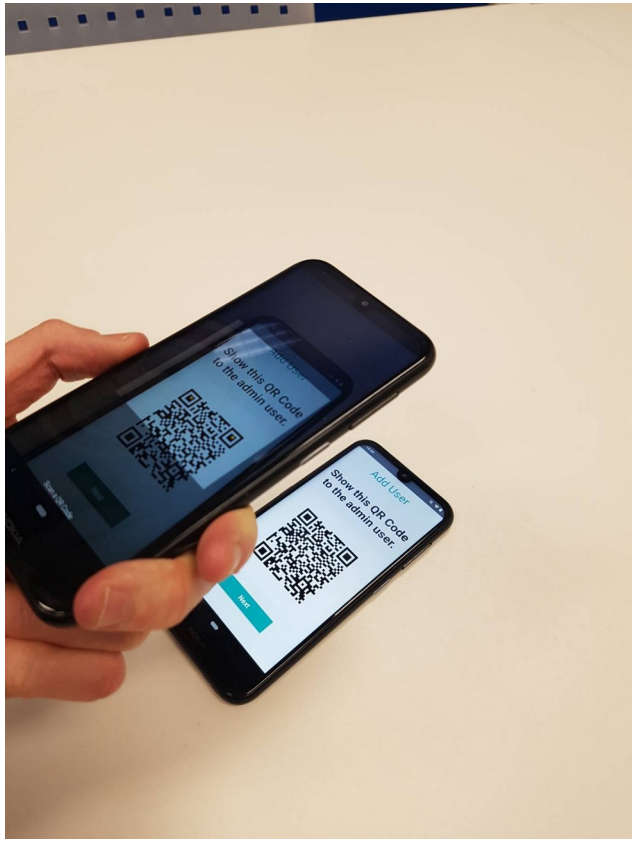
Assuming Control

How to establish an **authenticated and secure** pairing between **unassociated devices without** third party involvement?

PennState

Assuming control





J. M. McCune, A. Perrig, and M. K. Reiter, "Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication"

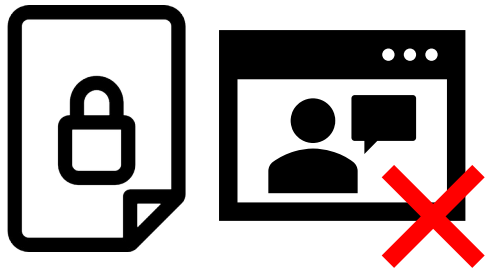
Protecting Content

How to **efficiently** encrypt the video footage to ensure **confidentiality**, **integrity**, **authenticity**, and **freshness** while supporting **fine-grained access delegation** and **recovery**?

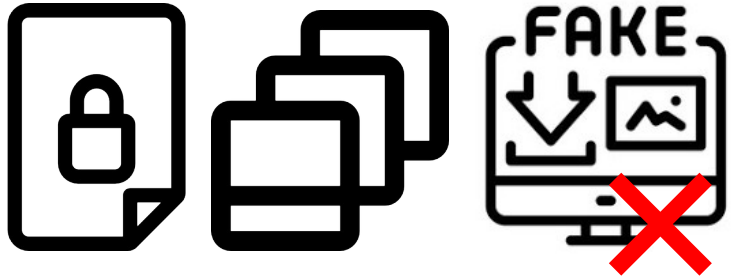
PennState

Protecting Content

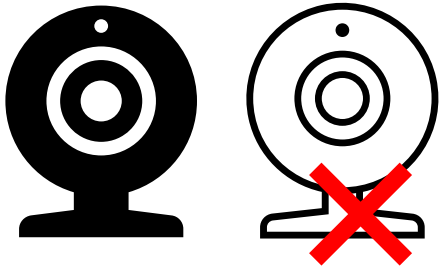
PennState



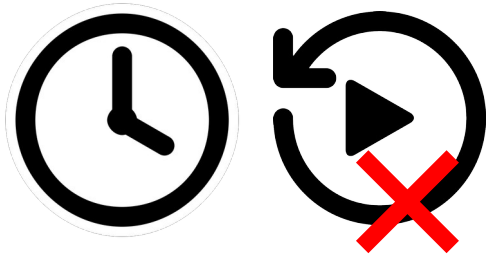
Confidentiality



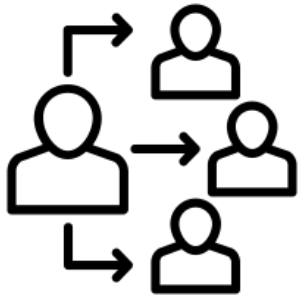
Integrity



Authenticity



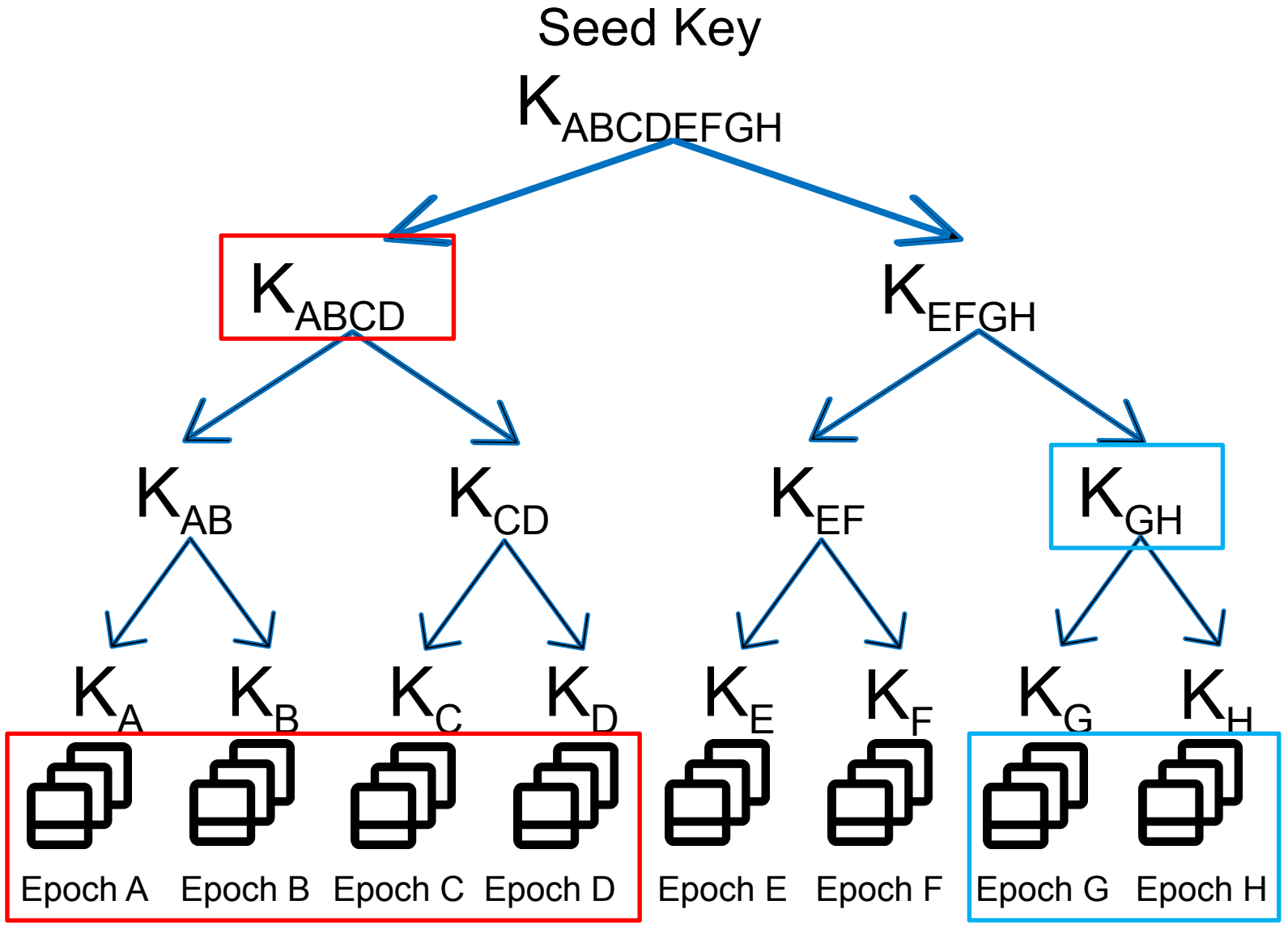
Freshness



Fine-grained delegation

Recovery

Key Tree


Seed Key
$K_{ABCDEFGH}$
K_{ABCD}
K_{EFGH}
K_{AB}
K_{CD}
K_{EF}
K_{GH}
K_A
K_B
K_C
K_D
K_E
K_F
K_G
K_H
Epoch A
Epoch B
Epoch C
Epoch D
Epoch E
Epoch F
Epoch G
Epoch H

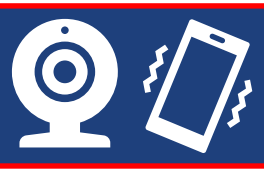
Roadmap

PennState

Main features of a smart camera system

Threat model

Privacy Challenges



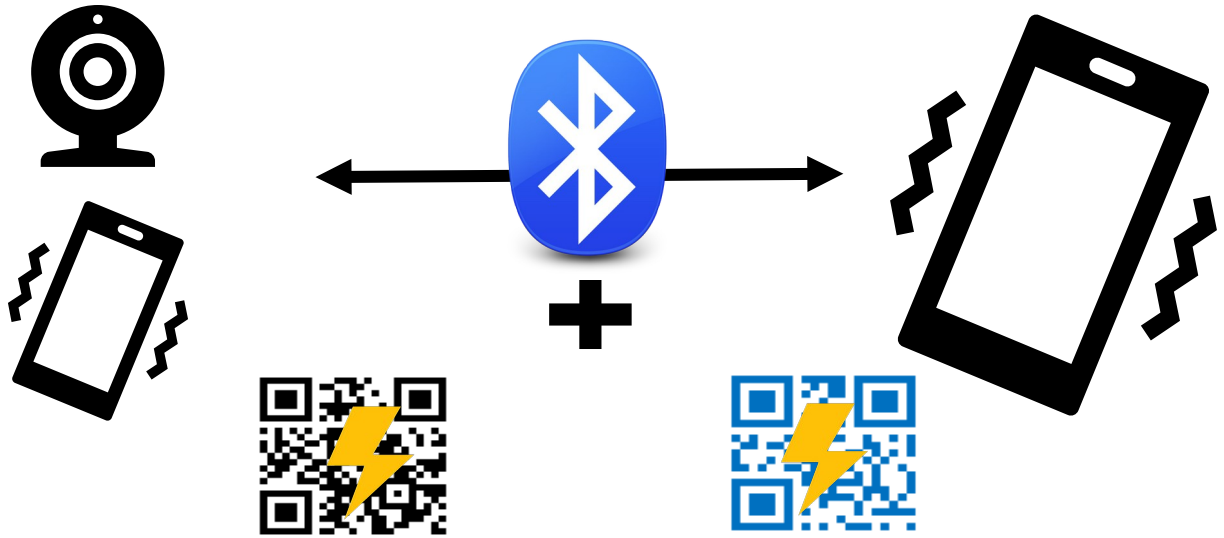
Implementation and demonstration

Evaluation

Setup, Delegation, Recovery, Reset

PennState

1. Pairing between devices

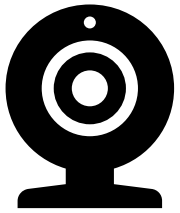


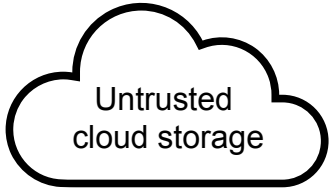
2. Exchange of corresponding material: keys, configs, escrow material...

3. Performing operation: initialization, recovery, reset

Streaming

PennState





Each frame F_i is recorded at timestamp t_i.

Capture frame by frame (v4l2)

$\{k_{t_i} \leftarrow Extract(\mathcal{K}, t_i) | i \in [\![1, N]\!]\}$

$\{IV_i \leftarrow RandBytes(16) | i \in [\![1, N]\!]\}$

Key Extraction

$\{C_i = AES256Enc(IV_i, k_{t_i}, F_i) | i \in [\![1, N]\!]\}$

$\{h_i = HMAC(k_{t_i}, C_i || IV_i || t_i) | i \in [\![1, N]\!]\}$

$\sigma = Sign(SK_{camera}, h_1 || h_2 || ... || h_N)$

Encryption

$< \{C_i, IV_i, t_i | i \in [\![1, N]\!]\}, \sigma >$

Upload

Download

$< \{C_i, IV_i, t_i | i \in [\![1, N]\!]\}, \sigma >$

Key Extraction

$\{k_{t_i} \leftarrow Extract(\mathcal{K}, t_i) | i \in [\![1, N]\!]\}$

Decryption

$\{h_i = HMAC(k_{t_i}, C_i || IV_i || t_i) | i \in [\![1, N]\!]\}$

$1 \stackrel{?}{=} Verify(PK_{camera}, \sigma_i, h_1 || h_2 || ... || h_N)$

$\{F_i = AES256Dec(IV_i, k_{t_i}, C_i) | i \in [\![1, N]\!]\}$

Frames Buffer (MediaCodec)

Video

Demonstration

PennState

Roadmap

PennState

Main features of a smart camera system

Threat model

Privacy Challenges

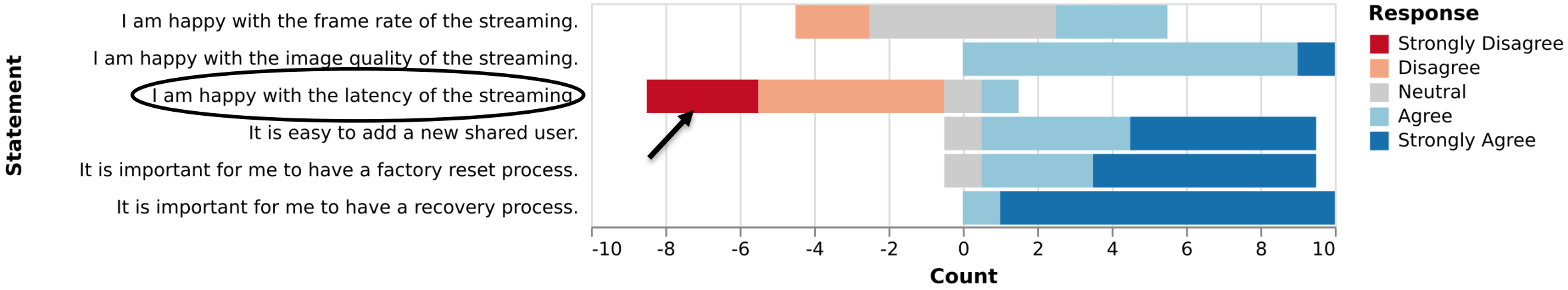
Implementation and demonstration

Evaluation

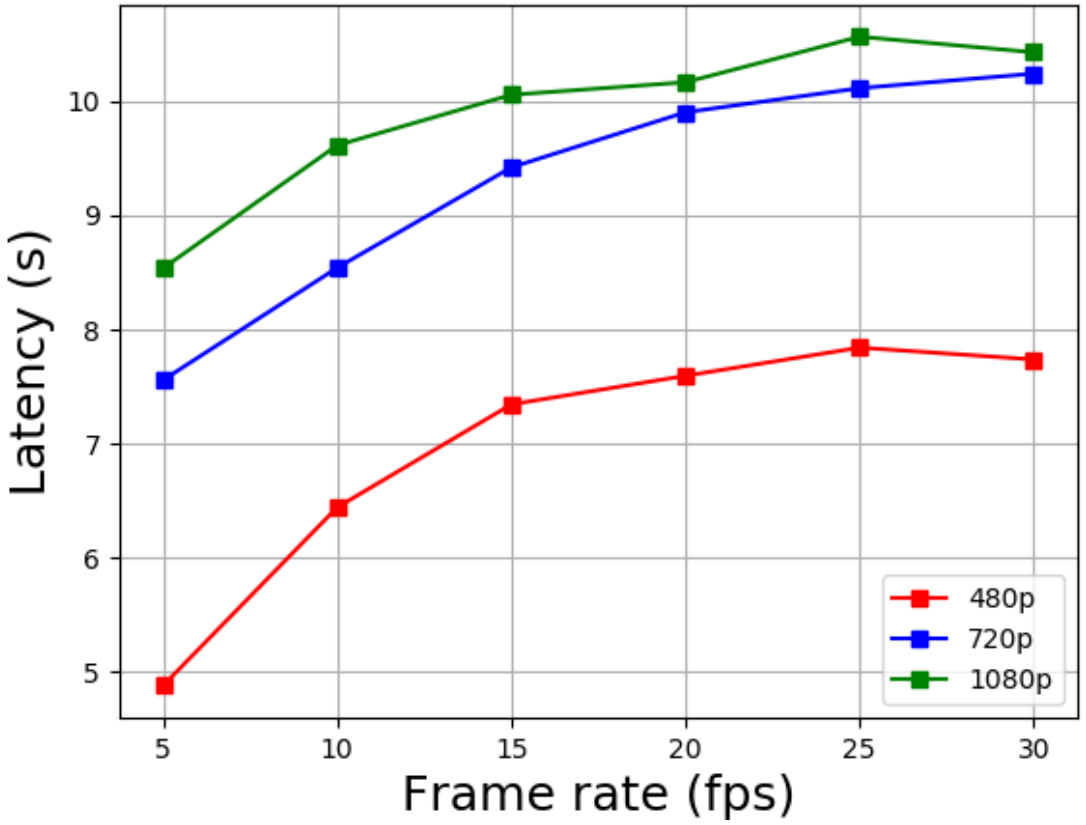
User study

Highlights:

- Self-explanatory and straightforward
- Pairing found simple to perform
- Latency to improve



Performance Evaluation



Key Extraction	Frame Encryption	Hash	Signature	Upload
0.05ms	2.755ms	1.908ms	8.749ms	509.20ms

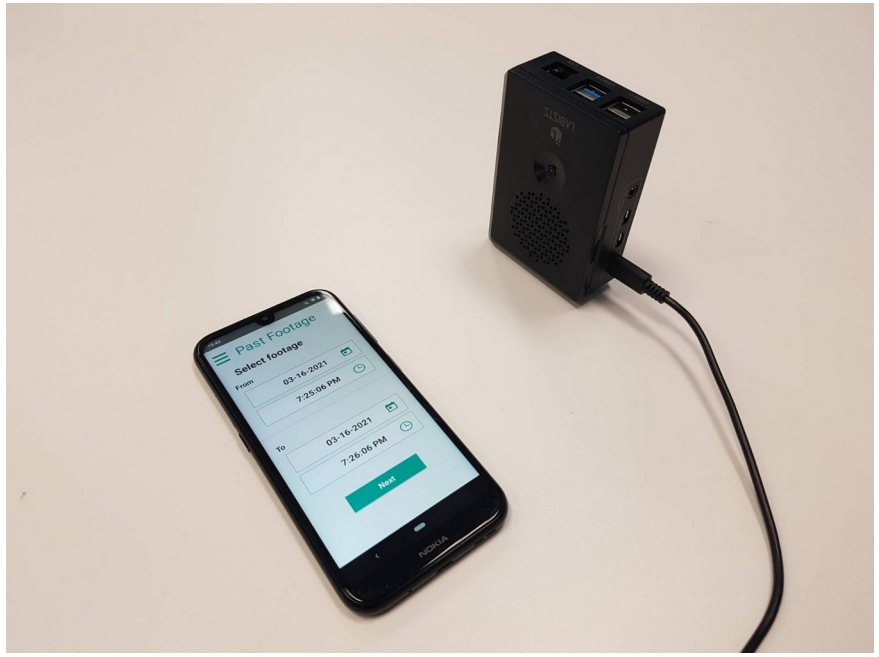
Download	Key Extraction	Frame Decryption	Hash Verification	Signature Verification
421.76ms	0.023ms	0.371ms	1.936ms	0.534ms

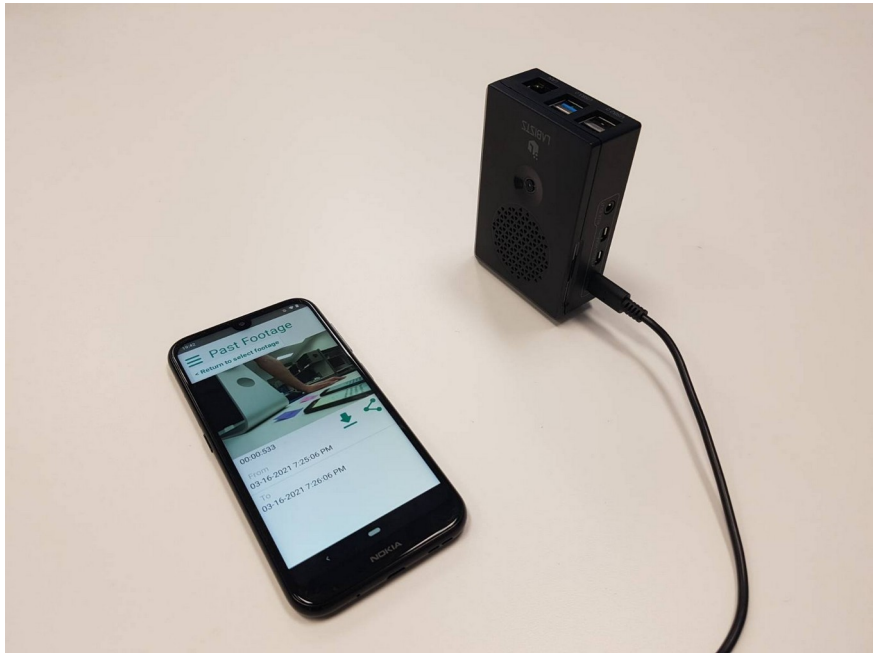
Table Time delay for each operation averaged over 1,000 frames

Conclusion

Privacy-preserving smart camera system **fully controlled** by the user:

- **No need** to trust any third party
- **Fine-grained** delegation
- **Same features** as commercially available systems





PennState

Thank you! Questions?

ykb5060@psu.edu

Additional slides

- If it were a real system...
- Seeing is Believing
- Eufy Doorbell
- OpenSSL Benchmarking
- Key derivation
- Android MediaCodec
- Process details

PennState

If it were a real system...

PennState

- Specialized hardware to improve the performances
- IOS implementation (+ web?)
- Movement detection
- Audio support
- Physical packaging to consider
- Trust the cloud storage to simplify deletion, delegation, and revocation.

Seeing is Believing (2005)

PennState

Channel	COTS	Resists MITM	Convenient
Ultrasound	○	○	●
Audible ("beeps")	○	◐	●
Radio	●	○	●
Physical Contact	○	●	●
Wired Link	●	●	○
Spoken Passwords	N/A	●	○
Written Passwords	N/A	●	○
Visual Hash Verif.	●	●	◐
Infrared	●	◐	◐
Seeing-Is-Believing	●	●	●

Figure 7. Characteristics of various channels proposed for authentication. We acknowledge that rating the convenience of a channel is subjective; however, we believe it is useful to compare various channels in this way. COTS indicates that the necessary hardware is already present in Commercial Off-The-Shelf products. Symbols: yes (●), partial (◐), no (○).

X \ Y	CD	C	D	N
CD	✓	✓*	✓	✓*
C	✓	✓*	✓	✓*
D	presence	presence	×	×
N	×	×	×	×

Legend	
✓	Strong authentication possible
✓*	Barcode label required on housing
presence	Confirm presence only
×	No authentication possible

Figure 2. Can a device of type X authenticate a device of type Y? We consider devices with cameras and displays (CD), cameras only (C), displays only (D), and neither (N).

J. M. McCune, A. Perrig, and M. K. Reiter, "Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication"

Eufy Doorbell

- New commercially available smart doorbell
- Security standards supposedly built in
- No current academic evaluation
- Few details about the key management, generation, distribution, and rotation

OpenSSL benchmarking

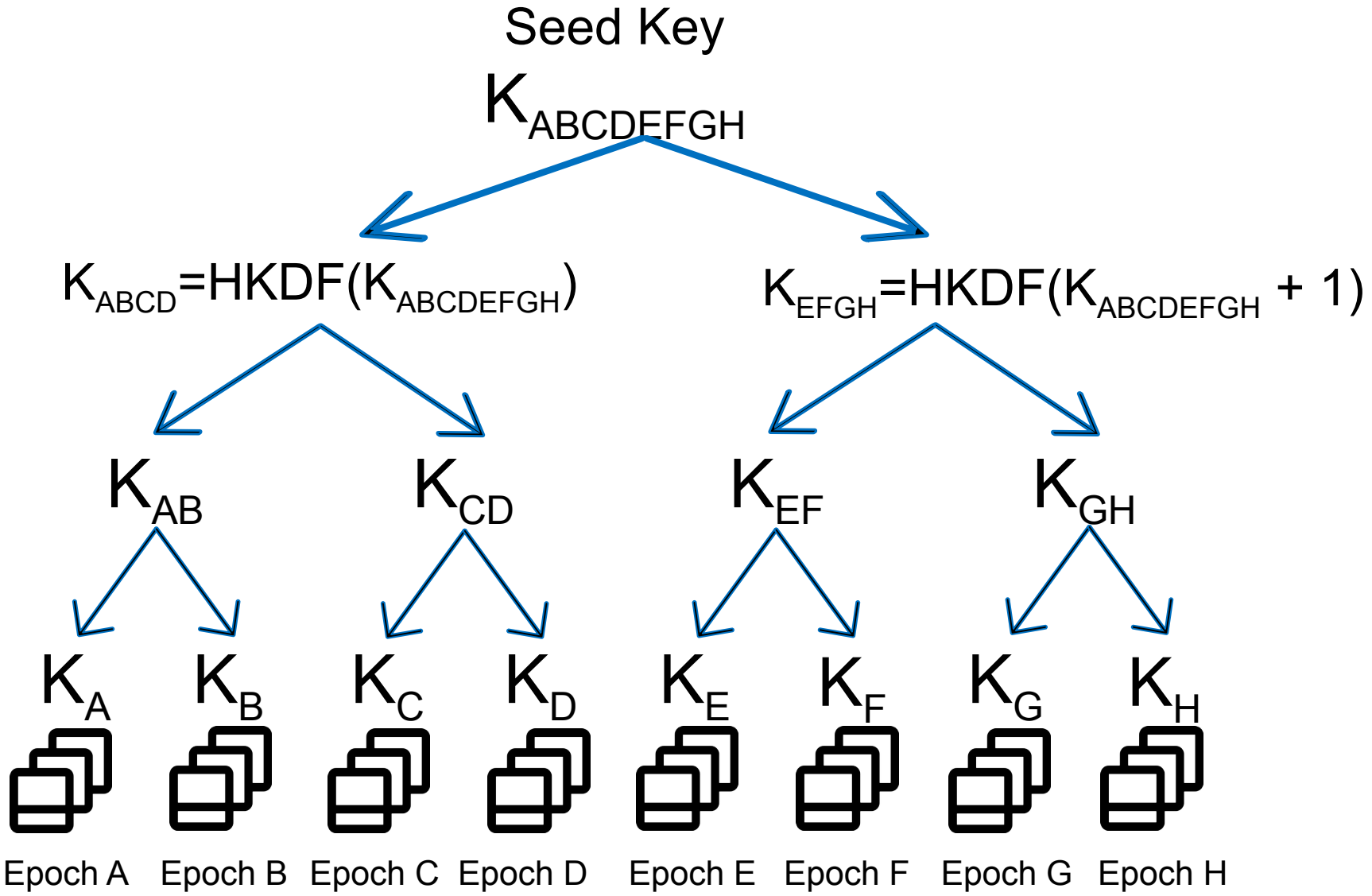
PennState

Algorithm	On 128 bytes blocks
aes-128-cbc	90,010 kB.s^{-1}
aes-192-cbc	78,780 kB.s^{-1}
aes-256-cbc	68,820 kB.s^{-1}
sha256	93,870 kB.s^{-1}
sha512	43,800 kB.s^{-1}
HMAC(sha256)	15,690 kB.s^{-1}
HMAC(sha512)	10,200 kB.s^{-1}

RSA signature size	Signing	Verifying
2048-bits	145.7 sign.s^{-1}	6621.3 verify.s^{-1}

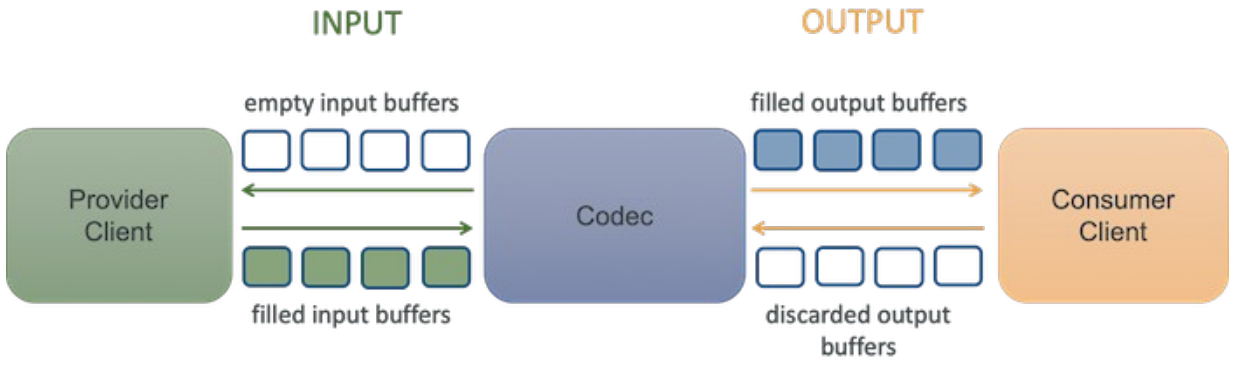
Raspberry Pi 4 Model B – 2GB SDRAM
Quad core Cortex-A72 (ARM v8) - 1.5GHz

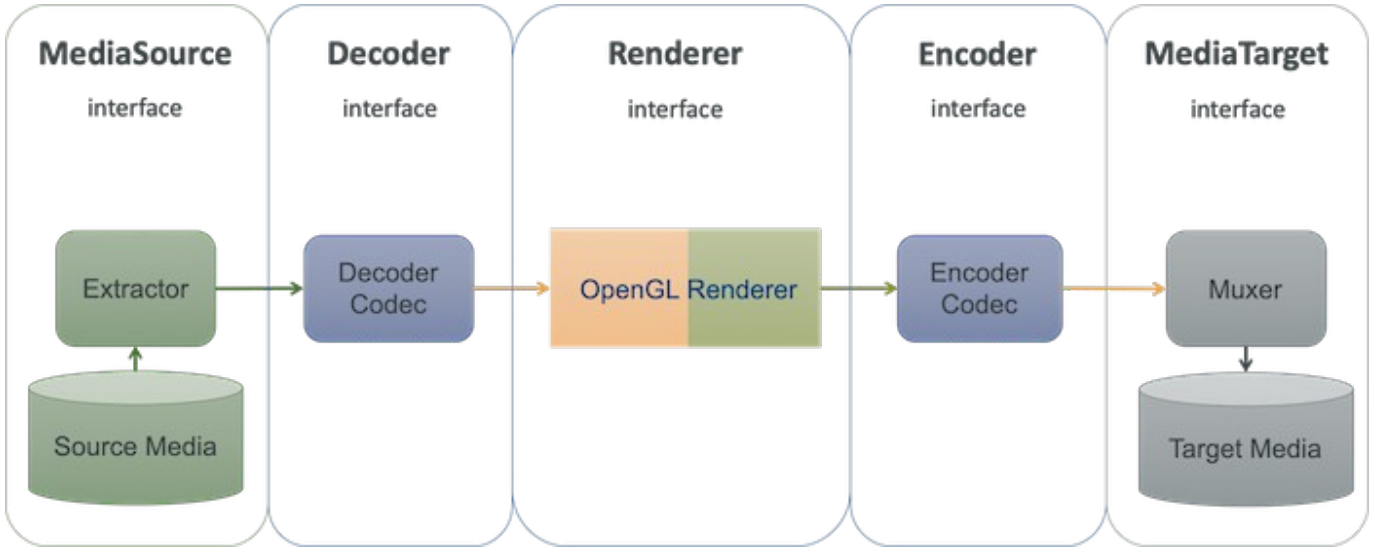
Key Derivation



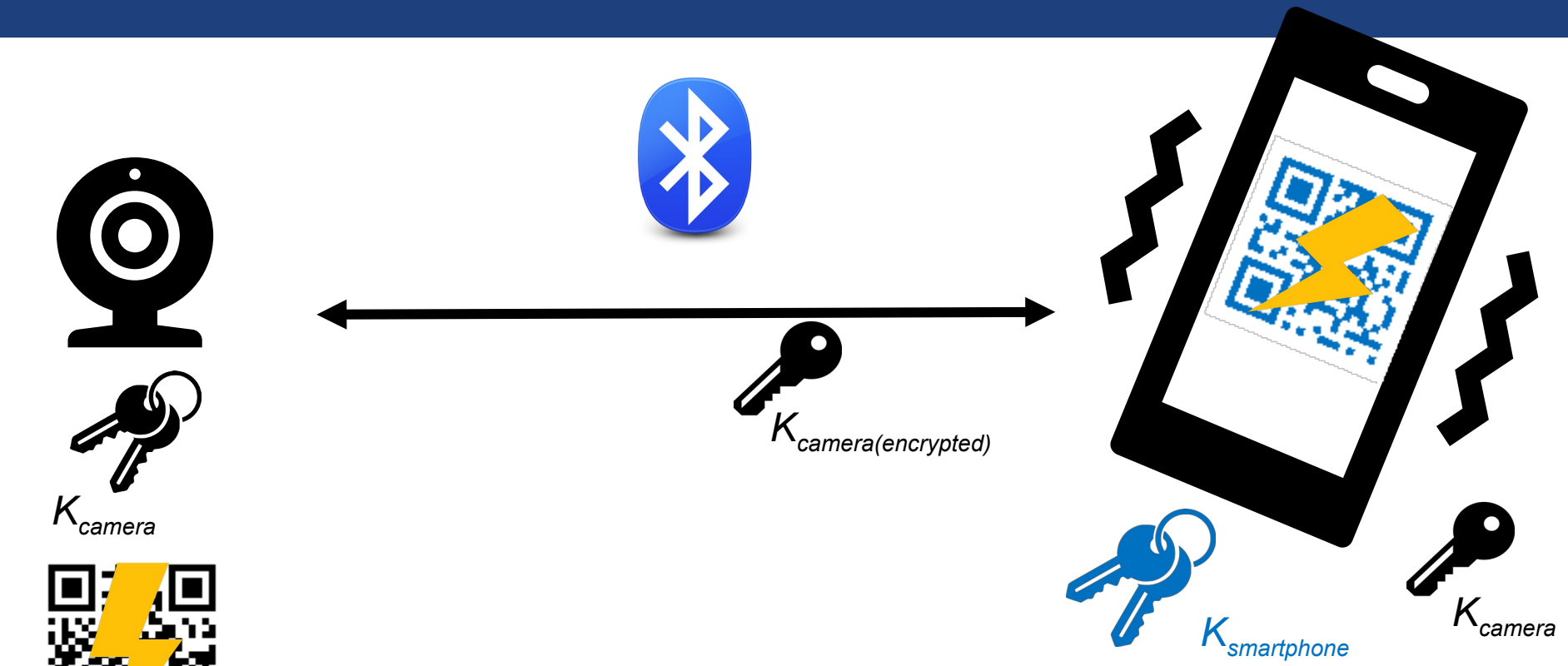
Android MediaCodec

PennState



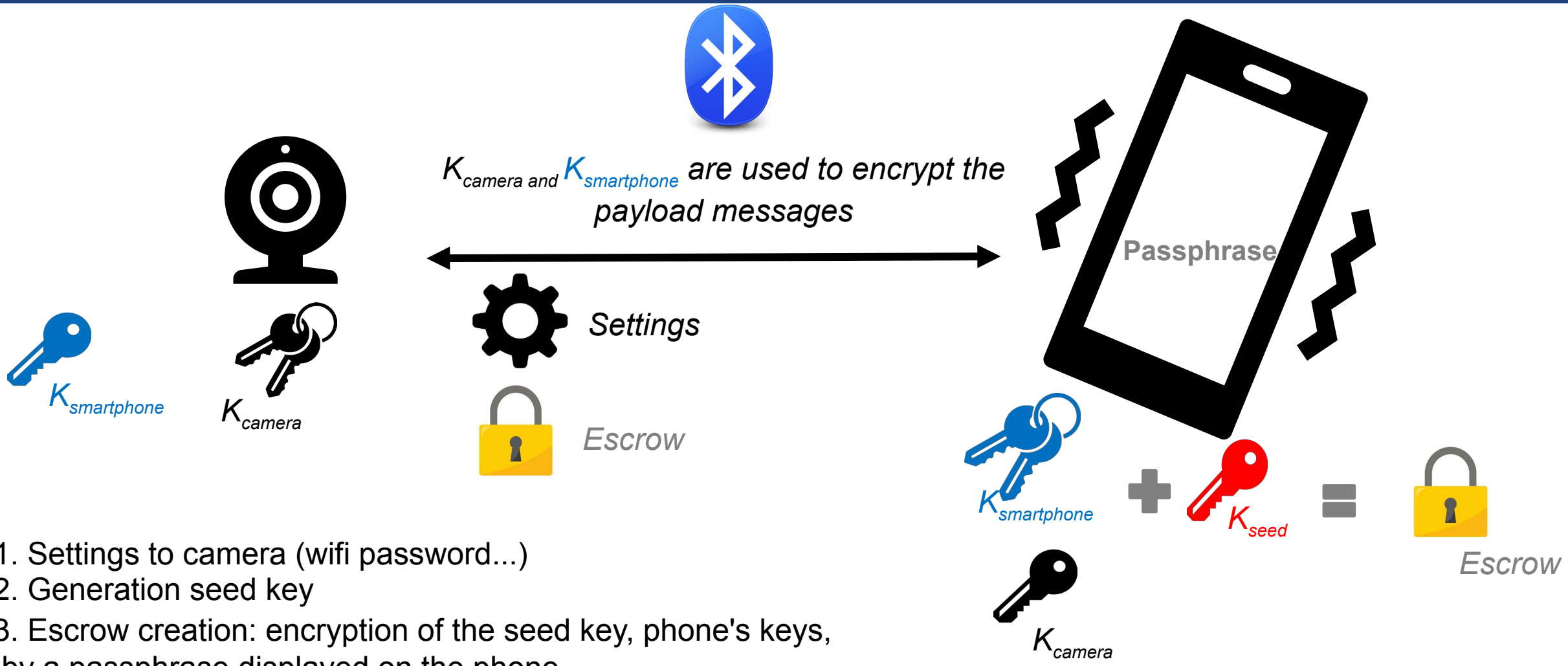


Setup



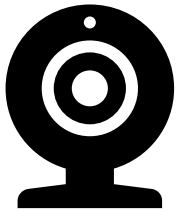
1. Camera and phone both generate key pair, phone creates QR code containing its public key
2. Phone scans camera's QR code, BT connects to camera
3. Camera scans phone's QR code to get phone public key
4. Camera encrypts its public key with the phone public key and sends to the phone

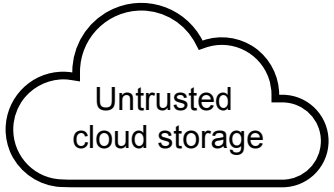
Setup (cont.)



1. Settings to camera (wifi password...)
2. Generation seed key
3. Escrow creation: encryption of the seed key, phone's keys, by a passphrase displayed on the phone
4. Escrow material sent to the camera

Streaming Process





Each frame F_i is recorded at timestamp t_i.

Capture frame by frame (v4l2)

$\{k_{t_i} \leftarrow Extract(\mathcal{K}, t_i) | i \in [\![1, N]\!]\}$

$\{IV_i \leftarrow RandBytes(16) | i \in [\![1, N]\!]\}$

Key Extraction

$\{C_i = AES256Enc(IV_i, k_{t_i}, F_i) | i \in [\![1, N]\!]\}$

$\{h_i = HMAC(k_{t_i}, C_i || IV_i || t_i) | i \in [\![1, N]\!]\}$

$\sigma = Sign(SK_{camera}, h_1 || h_2 || ... || h_N)$

Encryption

$< \{C_i, IV_i, t_i | i \in [\![1, N]\!]\}, \sigma >$

Upload

Download

$< \{C_i, IV_i, t_i | i \in [\![1, N]\!]\}, \sigma >$

Key Extraction

$\{k_{t_i} \leftarrow Extract(\mathcal{K}, t_i) | i \in [\![1, N]\!]\}$

Decryption

$\{h_i = HMAC(k_{t_i}, C_i || IV_i || t_i) | i \in [\![1, N]\!]\}$

$1 \stackrel{?}{=} Verify(PK_{camera}, \sigma_i, h_1 || h_2 || ... || h_N)$

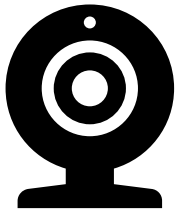
$\{F_i = AES256Dec(IV_i, k_{t_i}, C_i) | i \in [\![1, N]\!]\}$

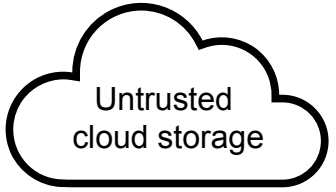
Frames Buffer (MediaCodec)

Video

Streaming Process (OTS)

PennState





Each frame F_i is recorded at timestamp t_i.

$$k_{t_i} \leftarrow Extract(\mathcal{K}, t_i)$$

$$IV_i \leftarrow RandBytes(16)$$

$$(PK_{i+1}, SK_{i+1}) \leftarrow OneTimePair()$$

$$C_i = AES256Enc(IV_i, k_{t_i}, F_i)$$

$$h_i = HMAC(k_{t_i}, C_i||IV_i||t_i||PK_{i+1})$$

$$\sigma_i = \begin{cases} Sign(SK_{camera}, h_1) & \text{if } i = 1 \\ Sign(SK_i, h_i) & \text{otherwise} \end{cases}$$

$$< C_i, IV_i, t_i, \sigma_i >$$

Capture frame by frame

Key Extraction

Encryption

Upload

Download

$$< C_i, IV_i, t_i, \sigma_i >$$

Key Extraction

$$k_{t_i} \leftarrow Extract(\mathcal{K}, t_i)$$

Decryption

$$h_i = HMAC(k_{t_i}, C_i||IV_i||t_i||PK_{i+1})$$

$$1 \stackrel{?}{=} \begin{cases} Verify(PK_{camera}, \sigma_1, h_1) & \text{if } i = 1 \\ Verify(PK_i, \sigma_i, h_i) & \text{otherwise} \end{cases}$$

Frames Buffer

$$F_i = AES256Dec(IV_i, k_{t_i}, C_i)$$

Video

R. Gennaro and P. Rohatgi, "How to Sign Digital Streams"

Delegation

PennState

Shared user's phone

Exchange public keys through QR codes scanning

Main user's phone

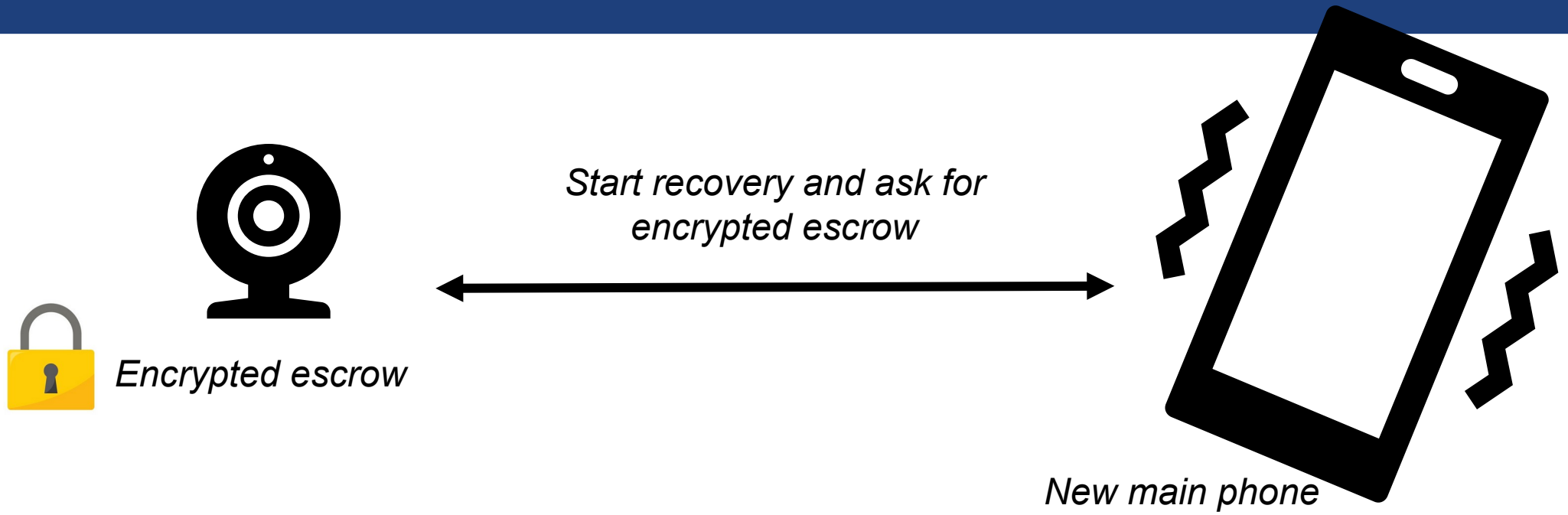
Definition of access control for this shared user (which keys they should have access to)

Retrieve encrypted decryption keys asynchronously

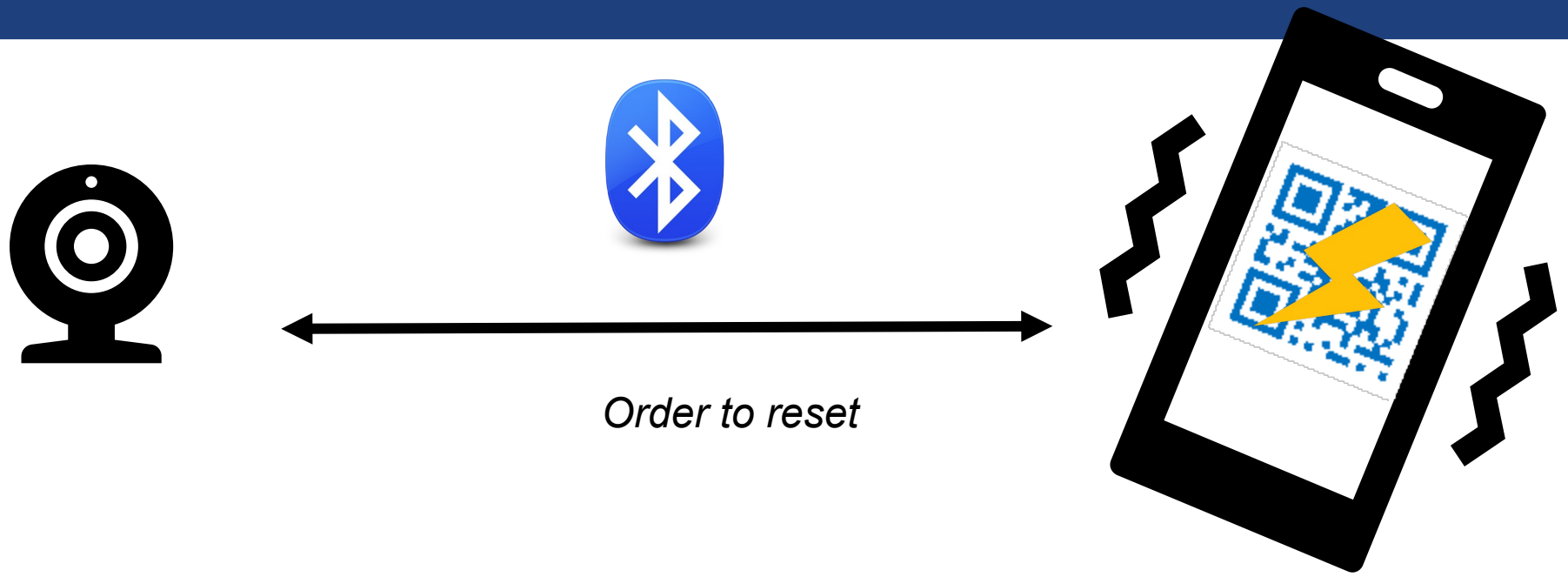
Decryption keys encrypted with shared user key sent asynchronously

Access Recovery

PennState



Factory Reset



1. Secure channel
2. Phone sends order to factory reset
3. Camera verify authenticity of order, and performs operation