Using Visual Challenges to Verify the Integrity of Security Cameras

Junia Valente
Alvaro Cárdenas

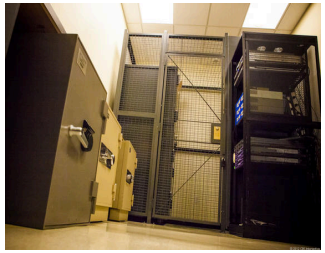
The University of Texas at Dallas

Supported in part by:
NIST 70NANB14H236,
AFOSR FA9550-12-1-0077

Problem: Security Cameras Used in Surveillance Sensitive Settings are Attractive to Attacks

- Security cameras are used in sensitive settings, e.g., to monitor:
  - Uranium enrichment facilities
  - Access to certificate vaults protecting secret keys, and computer generating random numbers for lottery
  - Electricity substations

Infrastructure surveillance camera

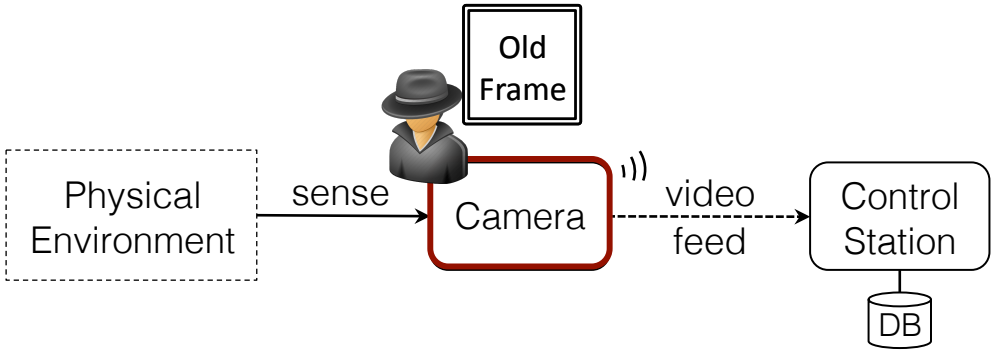


Camera monitors the entrance to Symantec's SSL certificate vaults

- "Hollywood-style" attacks are becoming a reality:
  - Camera monitoring access to computer generating random numbers for a lottery company recorded only a frame per minute, instead of a frame per second
  - Security expert Craig Heffner demonstrated how simple it is to exploit surveillance cameras by freezing the current frame on admin panel

Our Goal is to Detect a Compromised Camera in the Presence of An Attacker

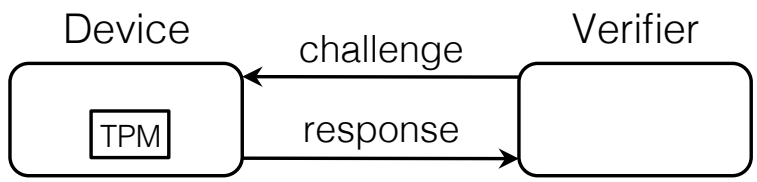
- Detect when a camera is reporting incorrect information or old data
- Goal: minimize the chances the attacker can use previously recorded footage and present it as new



Attacker is able to record old image frames and present them as new.

Attacker can Bypass Traditional Integrity Mechanisms when Secret Keys or Roots-of-Trust are Compromised

- *Attestation protocols* can improve device trustworthiness
  - Verifier sends a random challenge to the device and device replies with a response to prove its authenticity and freshness
  - Assumption: keys and root-of-trust have not been compromised



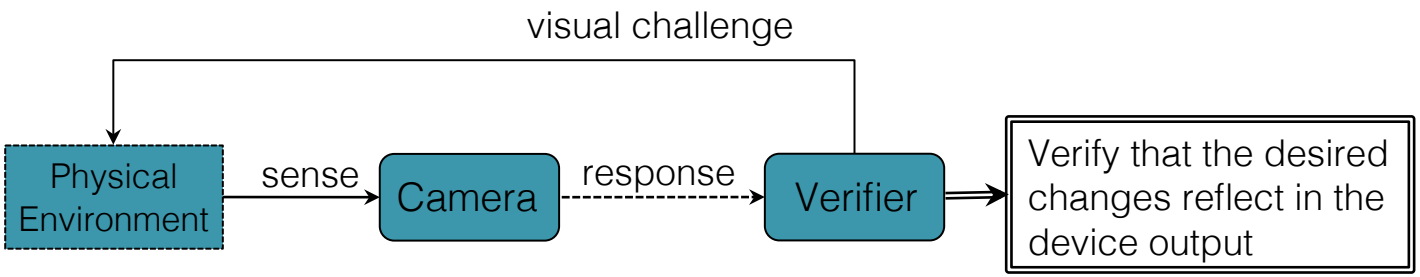
Can we still trust the device when this assumption does not hold?

We Propose a New Defense that Leverages the Unique Properties of Sensing Devices

- Cameras visually perceive the physical world
- They use sensors to create a digital form of a real world scenery

Proposed Solution: Send Visual Challenges to the Physical World the Sensing Device is Sensing, Instead to the Device

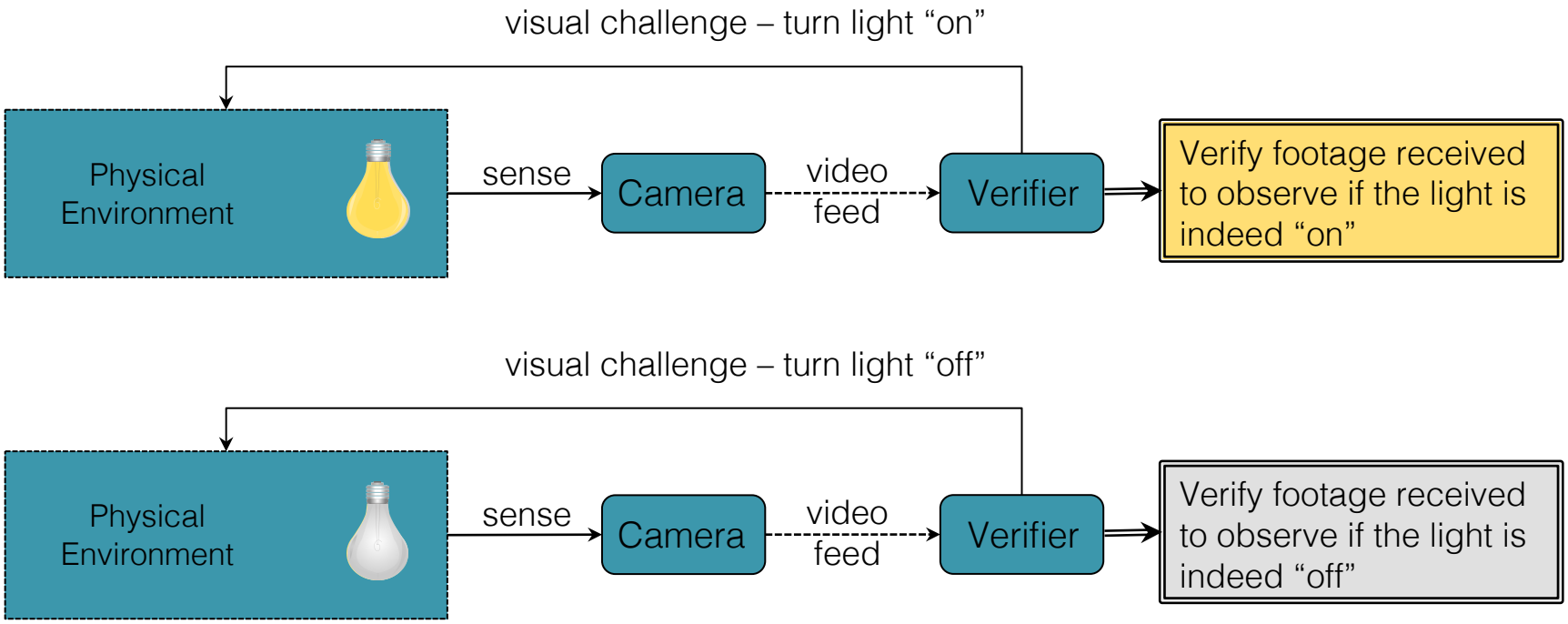
- To mitigate this problem we propose an independent verification channel which:
  - Detects integrity violation of camera feed even when secret information has been compromised or camera has been physically attacked
- We propose a new defense that leverages unique properties of cameras:



Outline

- Overview
- System Design
- Detection Mechanism
- Attacker Model
- Security Analysis

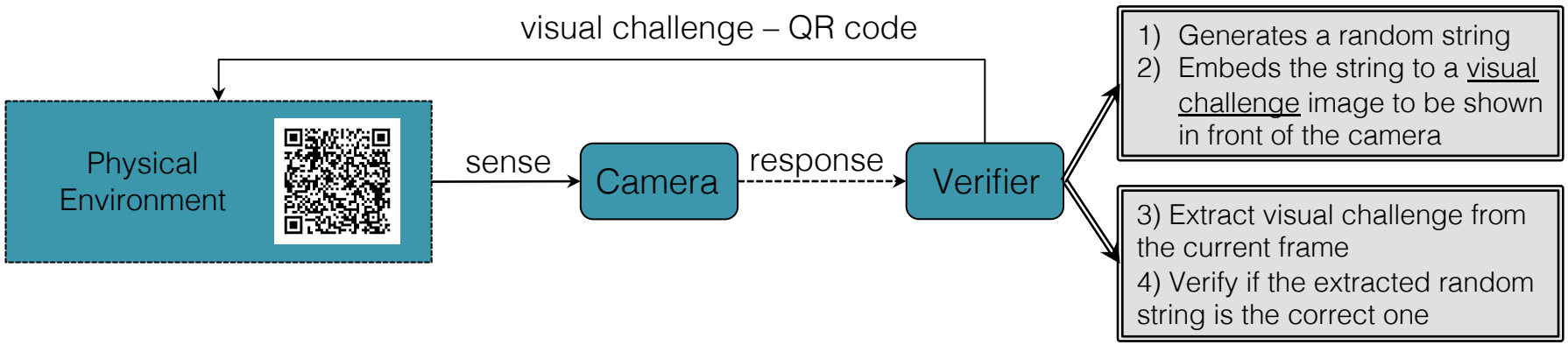
Intuitive Idea for Visual Challenges: Have a Light on Sight of Camera that Turns "On" or "Off" at Discretion of Verifier



This approach is vulnerable to replay attacks!

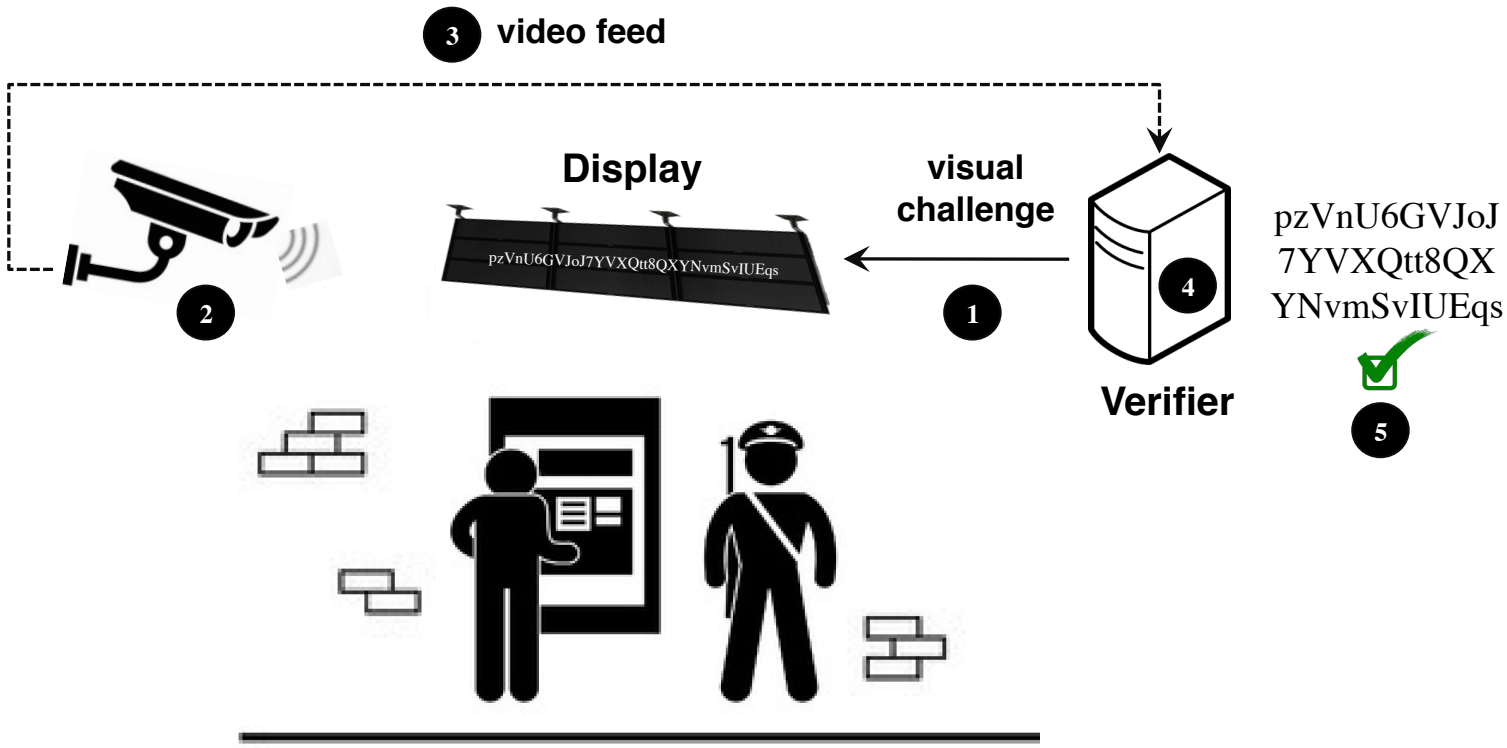
QR codes and Plain Text: Visual Challenges with High Entropy to Prevent Replay Attacks

- Our proposed visual challenges prevent an attacker from recording frames with all possible challenges and then replaying whenever a challenge is repeated



Attacker cannot record all possible frames containing our visual challenges

Two New Devices are Added to a Currently Deployed System: a Display and a Verifier



Verifier refreshes the random challenge periodically

Lab Setup for Experiments

QR Code on a digital monitor

Camera

We Found Vulnerabilities on the Camera System used in our Experiments and Reported Them to US CERT

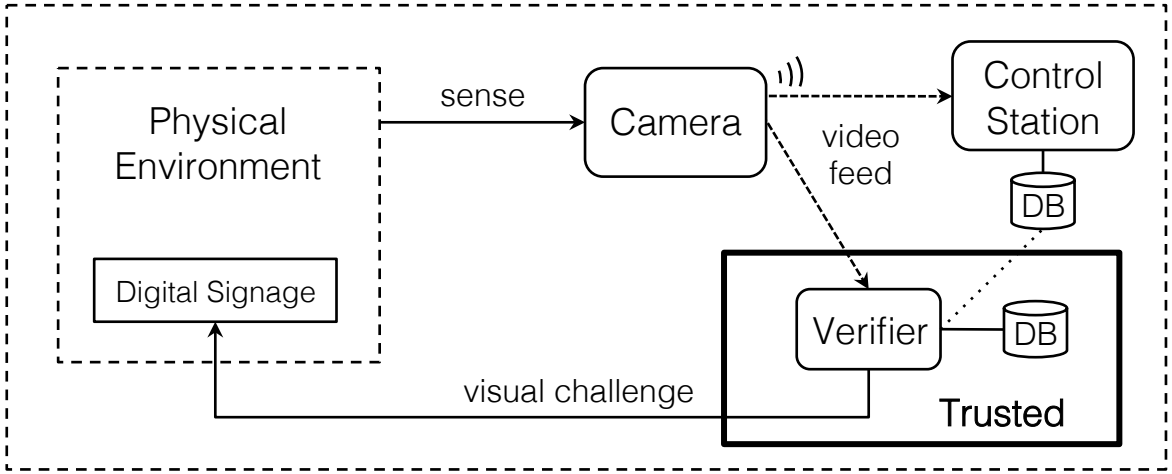
- Note: We have omitted the slides showing the vulnerabilities since we are still working with the vendor to fix the vulnerabilities, and the report is not public yet.

Outline

- Overview
- System Design
- DETECTION MECHANISM
- Attacker Model
- Security Analysis

Trust Assumptions: The Only Trusted Entity in our System is the Verifier

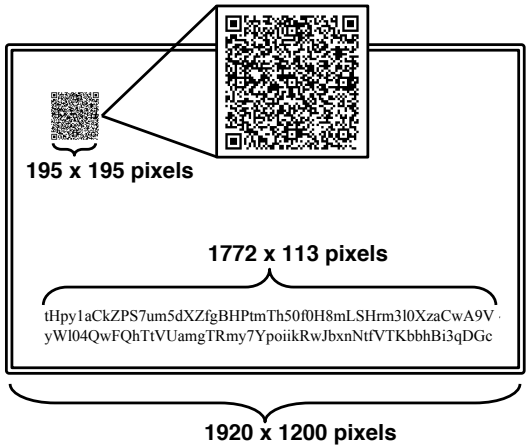
- We assume that everything else can be compromised



Background: Important Consideration that Differentiate the Use of QR Codes and Plain Text as Visual Challenges

- QR codes are decoded with barcode readers such as **`zbar`** and plain text are recognized with optical character recognition (OCR) such as **`tesseract`**.

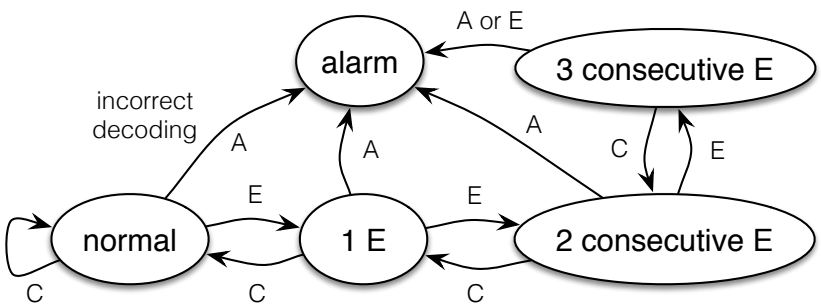
- QR codes produce a binary detection score (either correctly detect or produce decoding error) and OCR algorithms produce "soft" detection scores (0 to 100%)



Attack Detection: We Use the Accumulation of Errors over a Period of Time to Detect Attacks

- If our detection mechanism does not recognize a particular correct response to the challenge (a false alarm), the accumulative errors will still be low when no attacks
- Consistently high mismatch over time should be an indication of an attack

For QR codes:



C = correct | A = incorrect | E = cannot decode

For OCR, we use the CUSUM Algorithm:

$$S_0 = 0$$

$$S_k = \max(0, S_{k-1} + \varepsilon_k - \delta)$$

ε_k is the error difference between reconstructed and real challenge

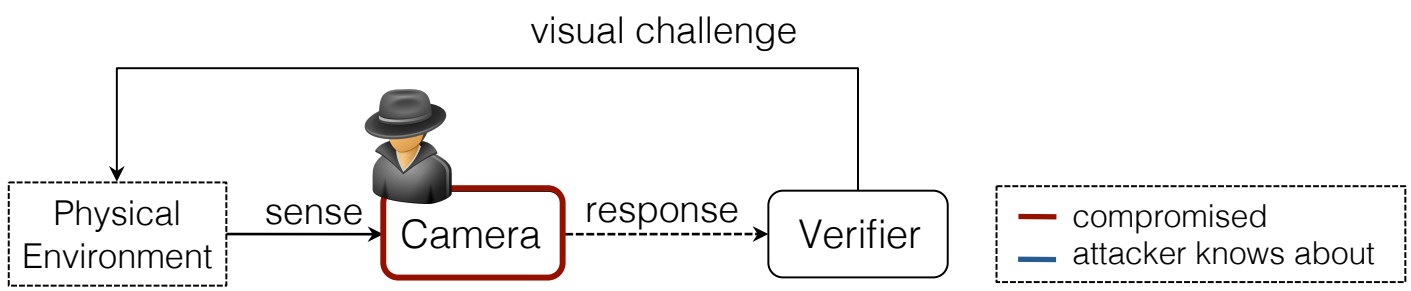
δ is a parameter chosen such that $\varepsilon_k - \delta$ is smaller than zero when no attack

Outline

- Overview
- System Design
- Detection Mechanism
- ATTACKER MODEL
- Security Analysis

Adversary Model: We Consider Three Types of Attackers

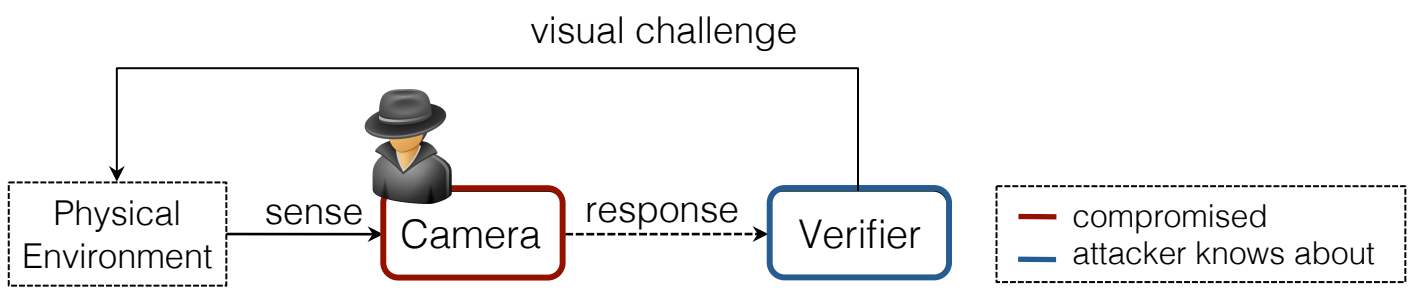
- Naive attacker: launches replay attacks (Hollywood style)
- Timing attacker: knows about our detection mechanism and tries to launch attacks without raising an alarm
- Forgery attack: attacker is aware of detection mechanism and has access to the visual challenge



Naive attacker

Adversary Model: We Consider Three Types of Attackers

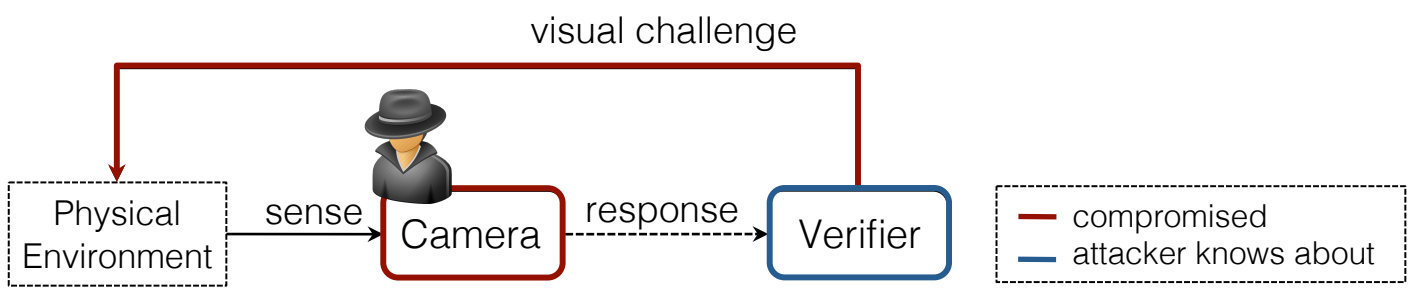
- Naive attacker: launches replay attacks (Hollywood style)
- Timing attacker: knows about our detection mechanism and tries to launch attacks without raising an alarm
- Forgery attack: attacker is aware of detection mechanism and has access to the visual challenge



Timing attacker

Adversary Model: We Consider Three Types of Attackers

- Naive attacker: launches replay attacks (Hollywood style)
- Timing attacker: knows about our detection mechanism and tries to launch attacks without raising an alarm
- Forgery attack: attacker is aware of detection mechanism and has access to the visual challenge



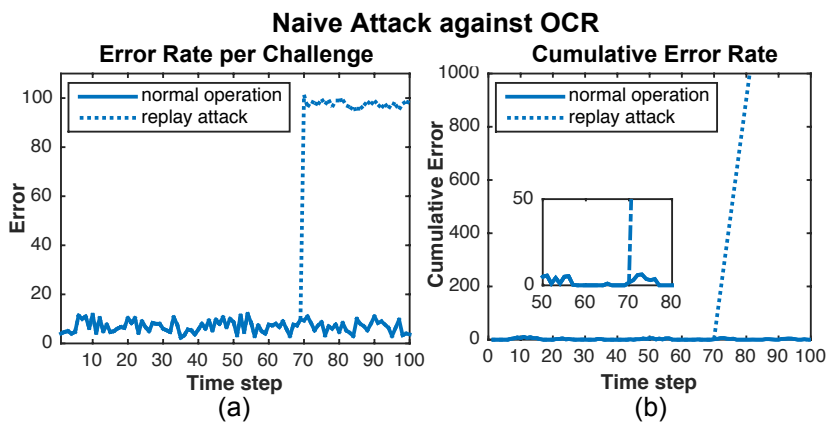
Forgery attack

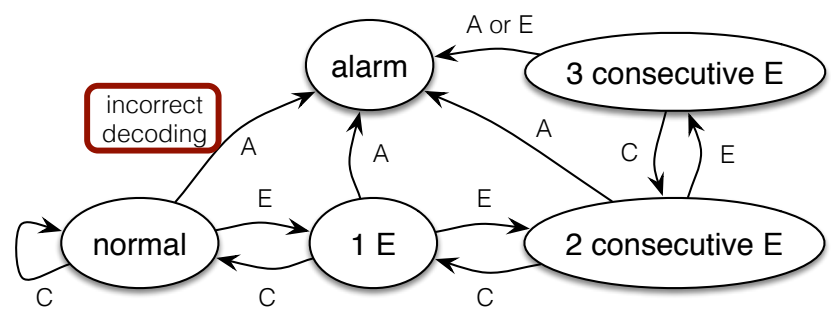
Outline

- Overview
- System Design
- Detection Mechanism
- Attacker Model
- SECURITY ANALYSIS

Naive Attacker: Launches Replay Attacks

- Attacker attempts to use old data (i.e., feed from a previous day) in a replay attack
- Our detection mechanism immediately detects this attack as the old data will not contain the correct visual challenge

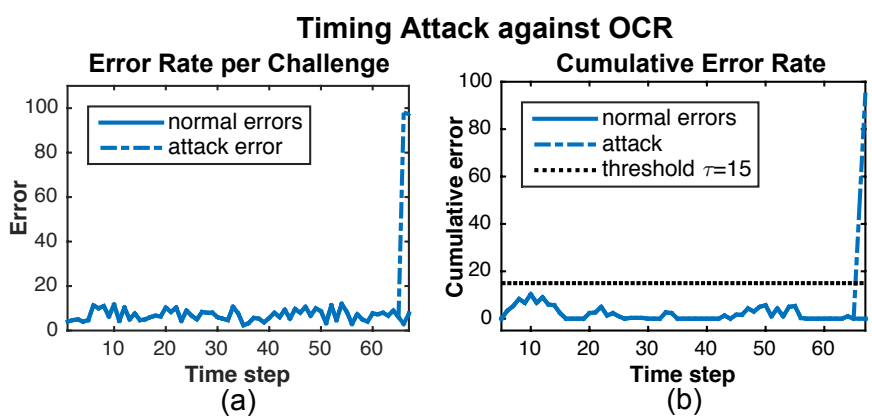


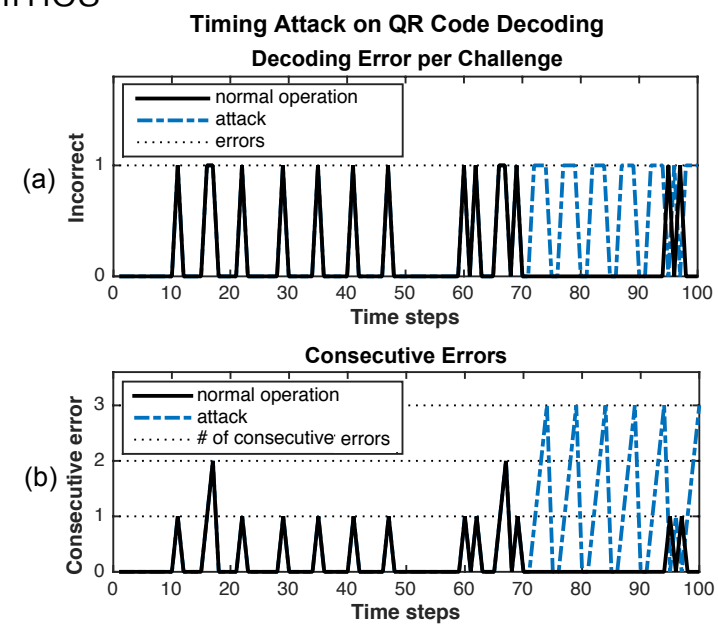


C = correct | A = incorrect | E = cannot decode

Timing Attacker: Tries to Launch Attacks without Raising an Alarm

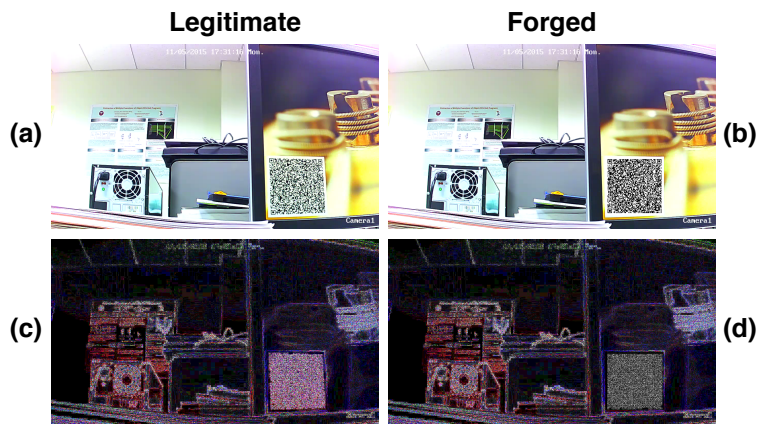
- Attacker goal: send a couple of incorrect frames without raising an alarm
- This is not possible for plain text. For QR codes, an attacker can leverage the fact that QR codes cannot be decodable sometimes





Forgery Attacker: Strongest Attack Against our System

- Attacker attempts to forge a fake image displaying the expected visual challenge
- Attacker goal: fool our detection mechanism to give operators the confidence that the feed is legitimate (because it contains the correct visual challenge)
- <u>Defense</u>: leverage image forensics tools such as Error Level Analysis algorithm



Ongoing / Future work: Usage of Image Forensics Techniques in Real-Time to Detect Forgery Attacks

- Goal: deploy an automatic test that detects attempted forgeries
- Advantages we have:
  - We can significantly reduce the search space for forgeries because the verifier knows the location where it put the visual challenge in the display
  - Anti-forensics tools are computationally expensive, and it is not clear if an attacker is able to finish a forged image (with correct visual challenge) in time to reply to our challenge

Conclusions

- We propose an independent channel for verification that will detect an integrity violation of the video received even when the camera has been compromised
  - Solution: the physical environment is continuously modified in a controlled way
  - Our approach leverages unique properties of sensing devices (i.e., camera)
- We can leverage unique properties from IoT devices to create new security solutions
- Advantages: our approach does not require the active involvement of the prover during the verification process, hence, it can work for legacy systems

Thank you!

Questions?

Junia Valente

(juniavalente@utdallas.edu)