28TH USENIX SECURITY SYMPOSIUM

EnTrust: Regulating Sensor Access by Cooperating Programs via Delegation Graphs

Giuseppe Petracca
gxp18@cse.psu.edu
The Pennsylvania State University

Yuqiong Sun
yuqiong_sun@symantec.com
Symantec Research Labs, US

Ahmad-Atamli Reineh
atamli@turing.ac.uk
Alan Turing Institute, UK

Patrick McDaniel
mcdaniel@cse.psu.edu
The Pennsylvania State University

Jens Grossklags
jens.grossklags@in.tum.de
Technical University of Munich, DE

Trent Jaeger
tjaeger@cse.psu.edu
The Pennsylvania State University

PennState College of Engineering

The Alan Turing Institute

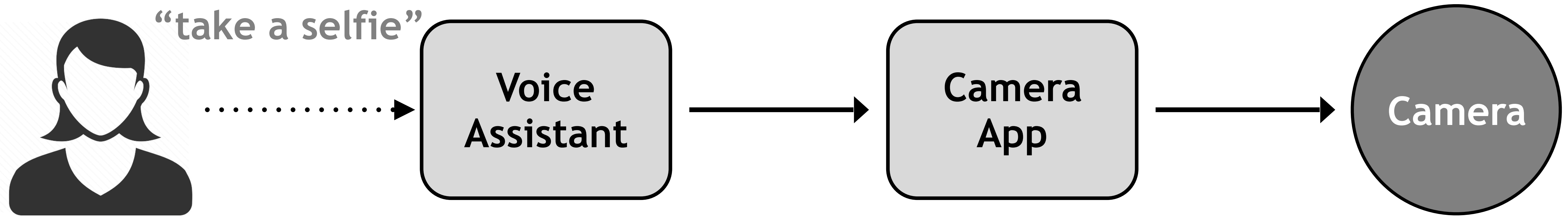
Technische Universität München

Symantec RESEARCH LABS

DISCOVER INNOVATE TRANSITION ARL

Cooperating Program Abstraction

PennState



What Can Go Wrong?

PennState

Researchers uncover new exploits in voice-powered assistants like Amazon Alexa or Google Assistant

McAfee Together is power.

Safeguarding Against Colluding Mobile Apps

ITPRO IT ANALYSIS. BUSINESS INSIGHT.

Researchers show Siri and Alexa can be exploited with 'silent' commands hidden in songs

cnet

SMART HOME

Voice of concern: Smart assistants are creating new openings for hackers

Let's talk about the security of smart speakers.

BY ALFRED NG | AUGUST 8, 2018 5:00 AM PDT

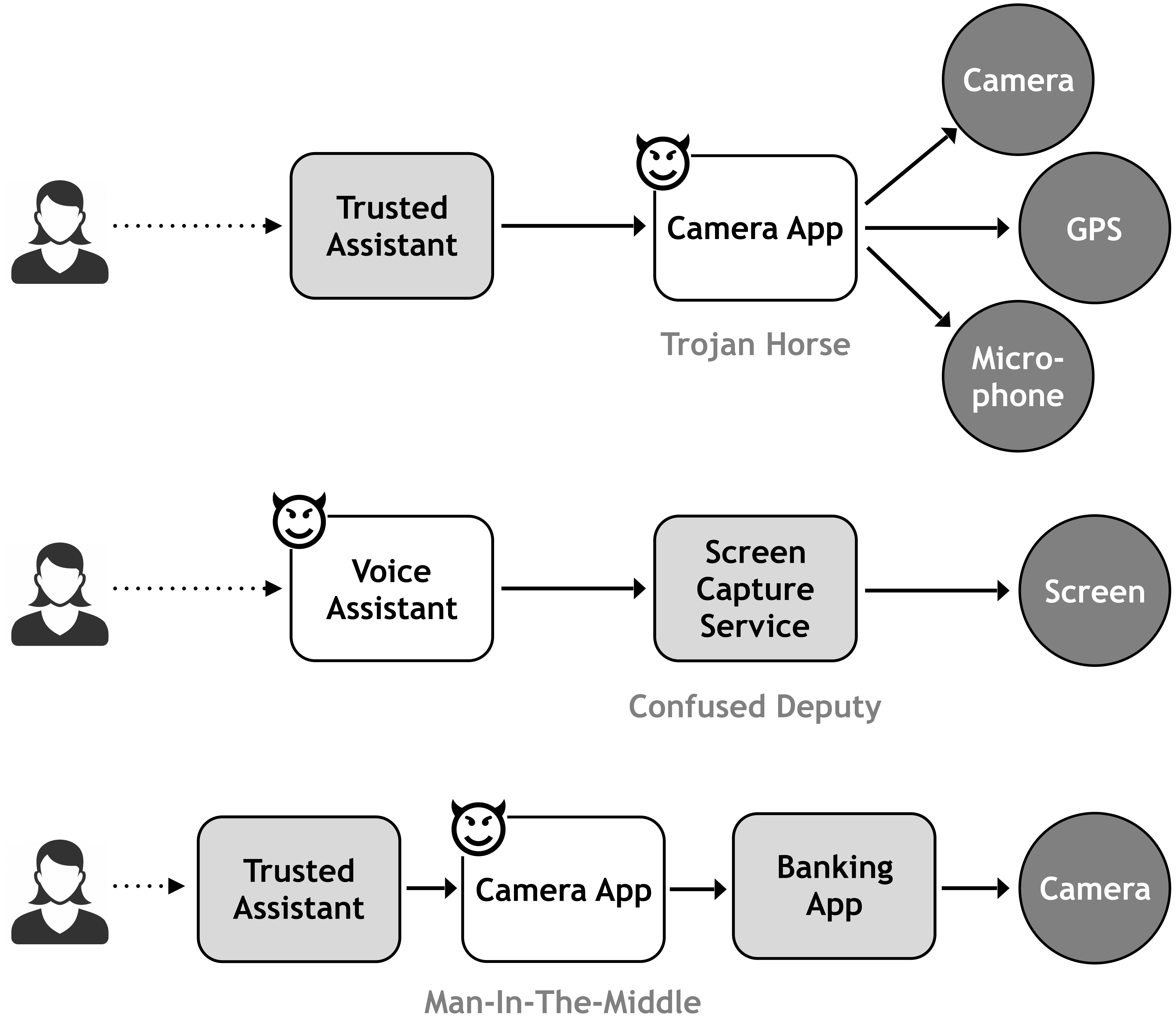
IEEE SECURITY & PRIVACY

Colluding Apps:

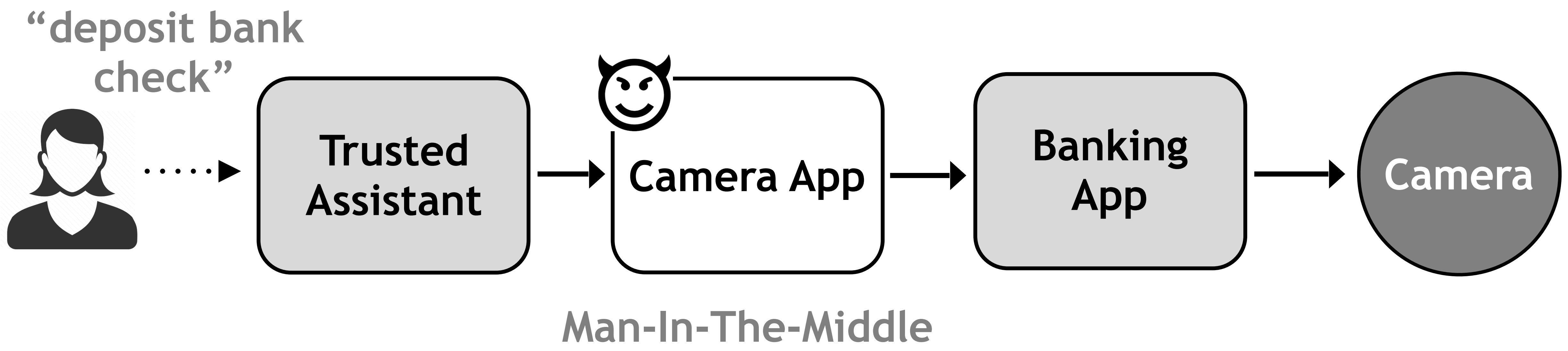
Tomorrow's Mobile Malware Threat

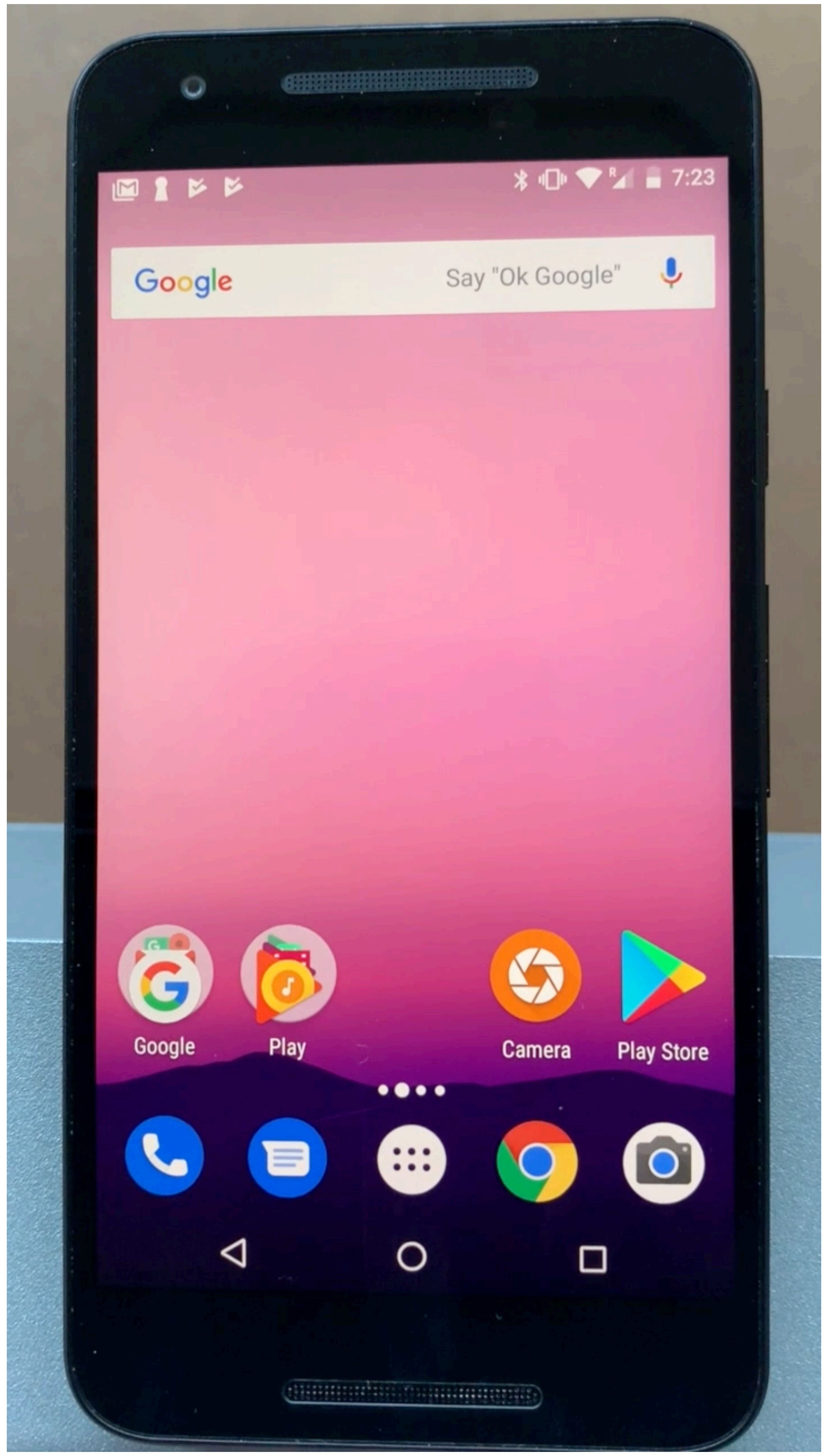
Atif M. Memon | University of Maryland, College Park
Ali Anwar | Montgomery Blair High School

Attack Vectors



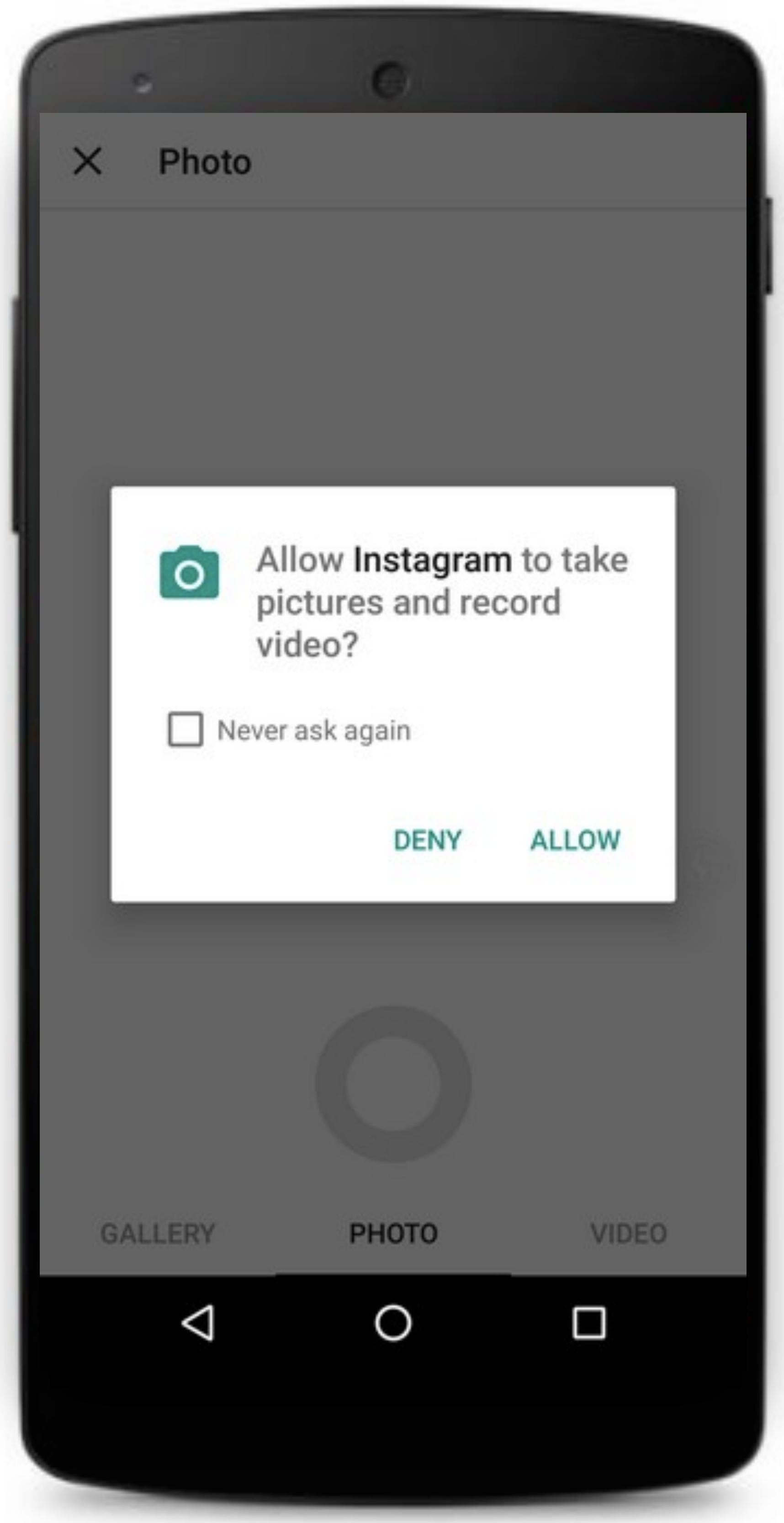
Man-In-The-Middle Attack





First-Use Authorization Model

Ask user for permission **ONLY** the first time **sensor X** is accessed by **program Y**



Proposed Defenses (Related Work)

Bind User Input Events and User Interface To Sensor Access

- Restrict context of use for sensors
- *Do not model input event delegation*

Regulate Inter-Process Communication (IPC)

- Restrict programs interactions
- *Too restrictive (reduce callee's permissions based on caller)*

Enforce Decentralized Information Flow Control (DIFC)

- Restrict how information flows between programs
- *Solve the orthogonal problem of controlling how program share data*

Classify Sensor Access via Machine Learning (ML)

- Model patterns in user decisions
- *Users need the right information to make the right decision (learning depends on users decision)*

Security Model

PennState

Trust Model

- System is booted securely (e.g., kernel, OS, system services, sensor drivers)
- Mandatory Access Control (MAC) enforced from boot time (**no direct access** to sensors for user-level programs)
- User-level programs isolated via sandbox
- Trusted Paths (UI —> OS —> UI)

Threat Model

- Users may install programs from unknown sources (grant access “at first-use”)
- Programs communicate via Inter-Process Communications (e.g, intents, broadcast messages)
- Programs may leverage IPC to exploit the three attack vectors mentioned

Focus —> How programs access sensors

Out Of Scope —> How programs share collected data (solutions exist)

Our Insight

- Track how an input event is delegated among cooperating programs
- Expose delegation information to users (informed authorization decisions)
- Allow users to restrict the set of permissions of the delegated program

Challenges

- Track the input event delegation (from the user input to the sensor operation)
- Resolve ambiguities with multiple (concurrent) events
- Authorize the right set of permissions given the input event

Our Approach: Track Input Event Delegation

Input Event

$e = (c, s, p_i, t_0)$

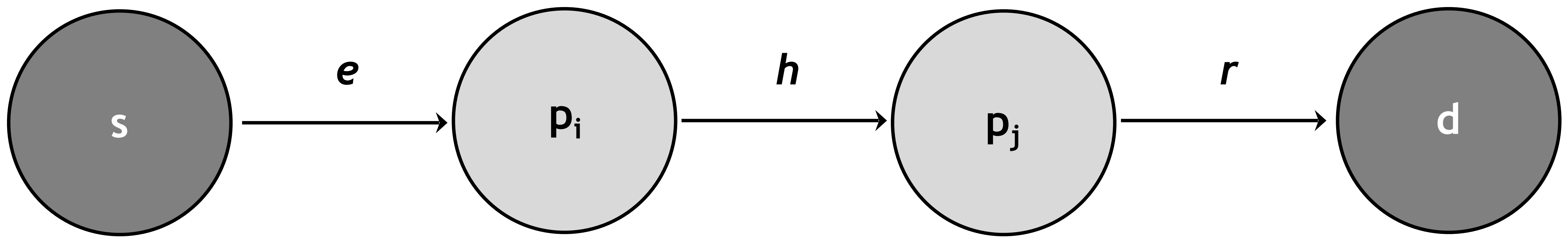
Handoff Event

$h = (p_i, p_j, t_i)$

Sensor Request

$r = (p_j, o, d, t_j)$

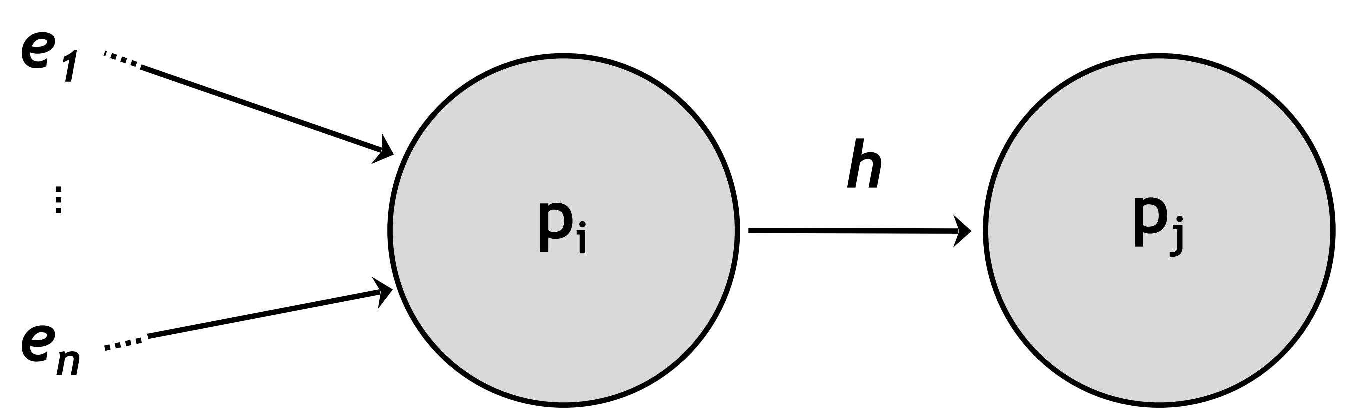
Delegation Graph

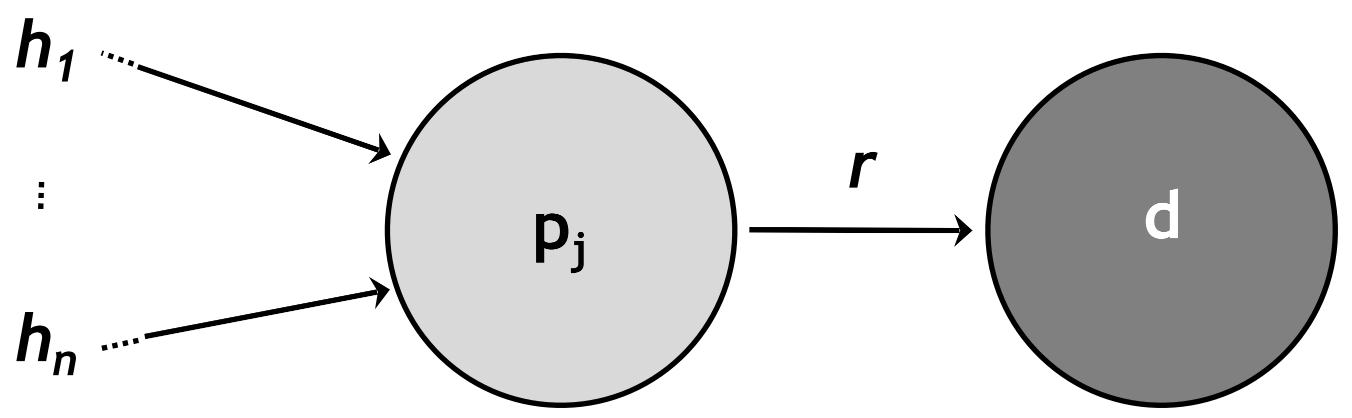


c = context s = source sensor p = program t = timestamp *o = sensor operation* d = destination sensor

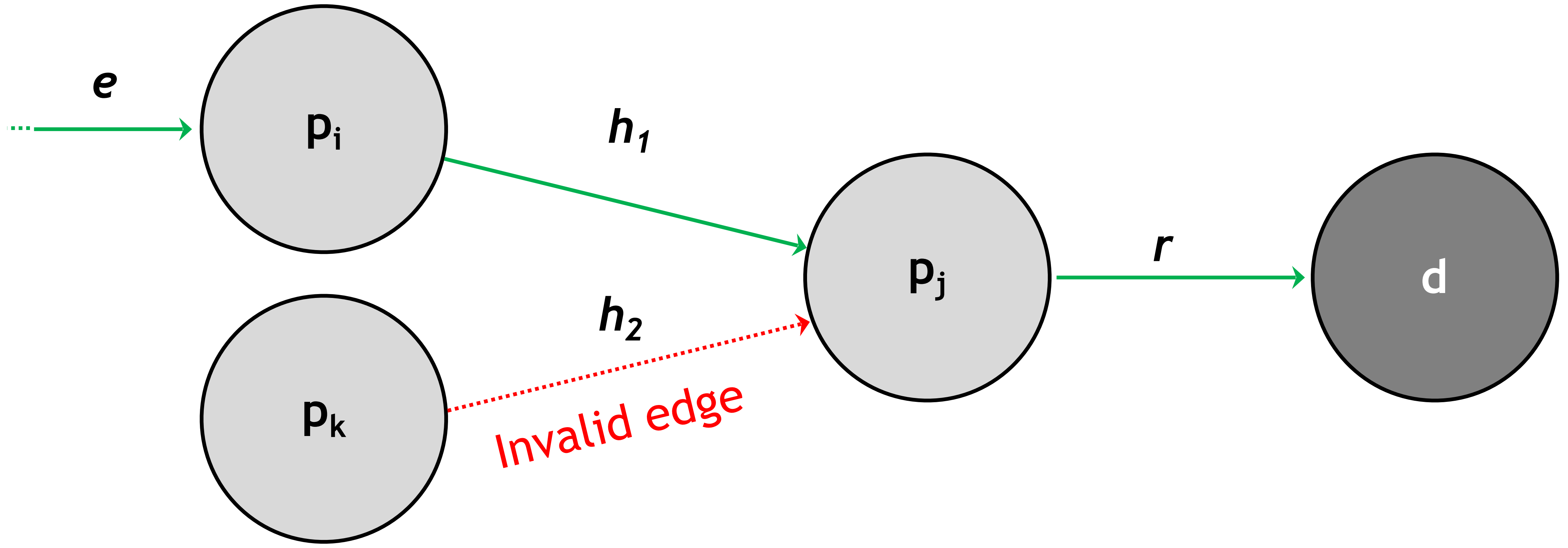
Our Approach: Resolve Ambiguities (Delegation Path)

PennState





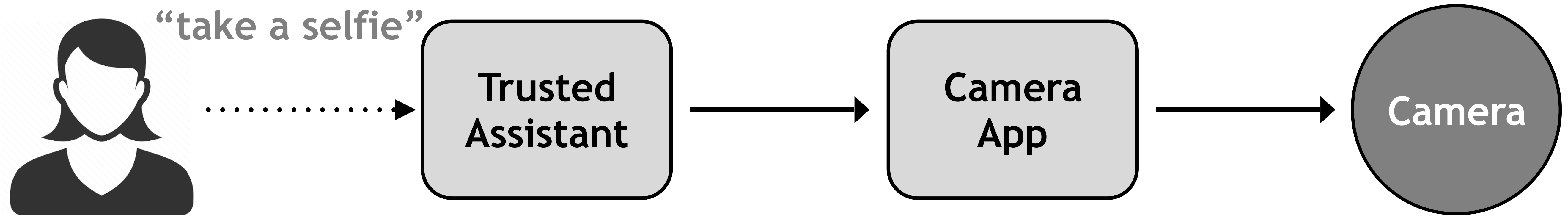
- Queue and deliver sequentially (events are consumed faster than produced)

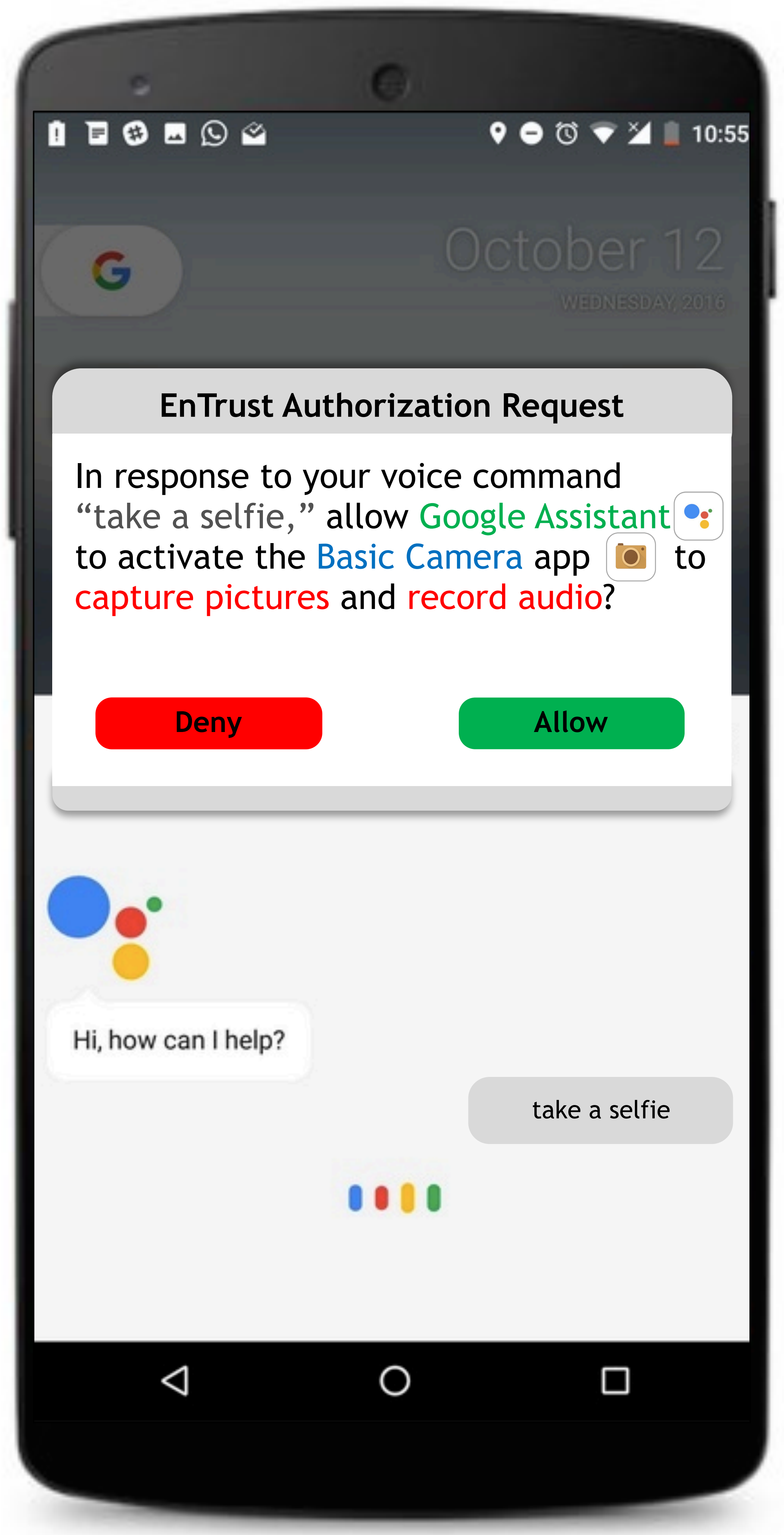


- Prioritize handoff events deriving from input event

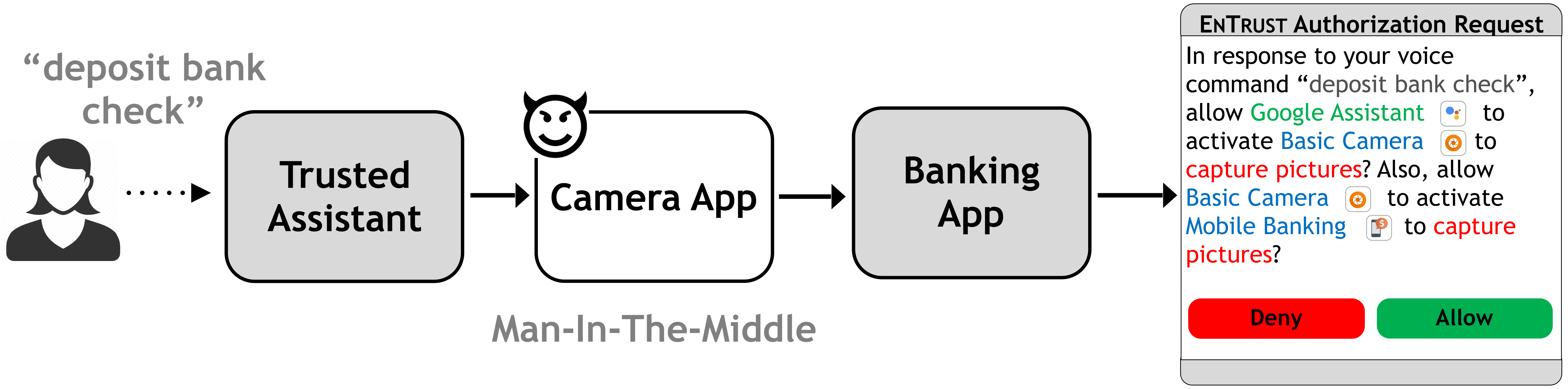
e = input event   *h* = handoff event   *p* = program   r = *operation request*   *d* = destination sensor

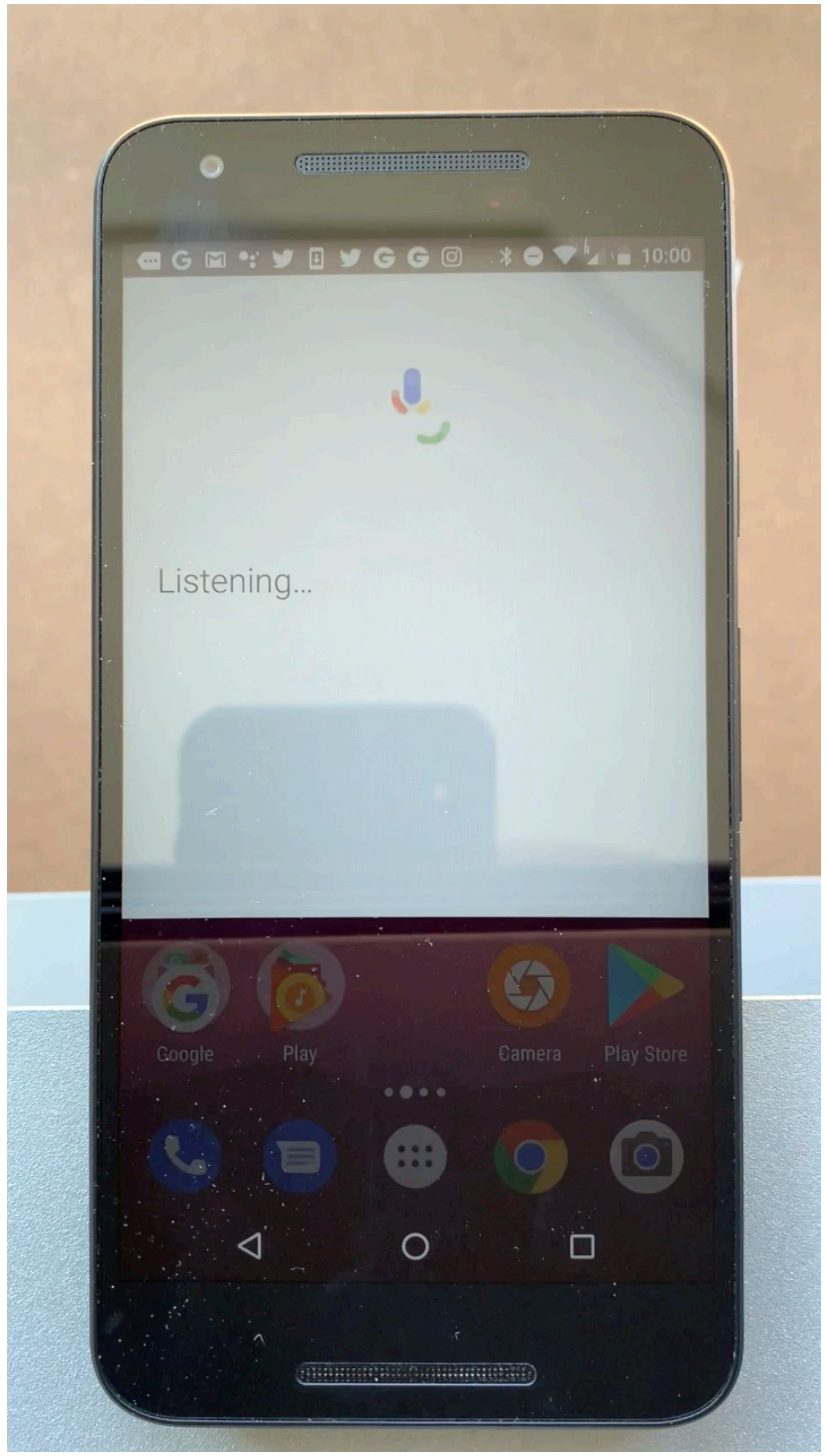
Our Approach: Authorize The Right Set Of Permissions





Man-In-The-Middle Attack (Prevented by EnTrust)





EnTrust: Evaluation

Prototyped (Android OS 7.1.1_r3)

Tested (Nexus 5X smartphones)

Research Questions:

- What is the decision overhead imposed by **EnTrust** on users due to explicit authorization of constructed delegation graphs? (**Field Study**, 9 Subjects, 7 Days, 10 Apps, 5 Voice Assistant)
- Is **EnTrust** backward-compatible with existing programs? How many operations from legitimate programs are incorrectly blocked by **EnTrust**? (Android **Compatibility Test Suite** (CTS), 1k Apps, 5 Augmented Reality Gaming Apps)
- What is the performance overhead imposed by **EnTrust** for delegation graph construction and enforcement? (Graph Construction, Graph Caching, Graph Enforcement, Ambiguity Prevention, Memory Requirements)
- To what degree is the **EnTrust** authorization assisting users in avoiding *Confused Deputy*, *Trojan Horse*, and *Man-In-The-Middle* attacks? (**Laboratory Study**, 60 subjects, 4 Groups, 3 Attacks)

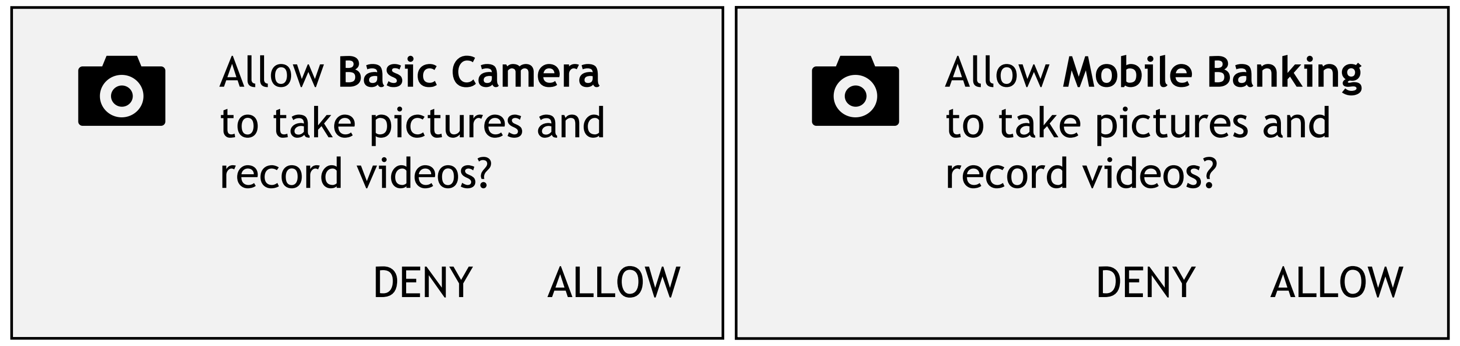
EnTrust: Laboratory Study

PennState

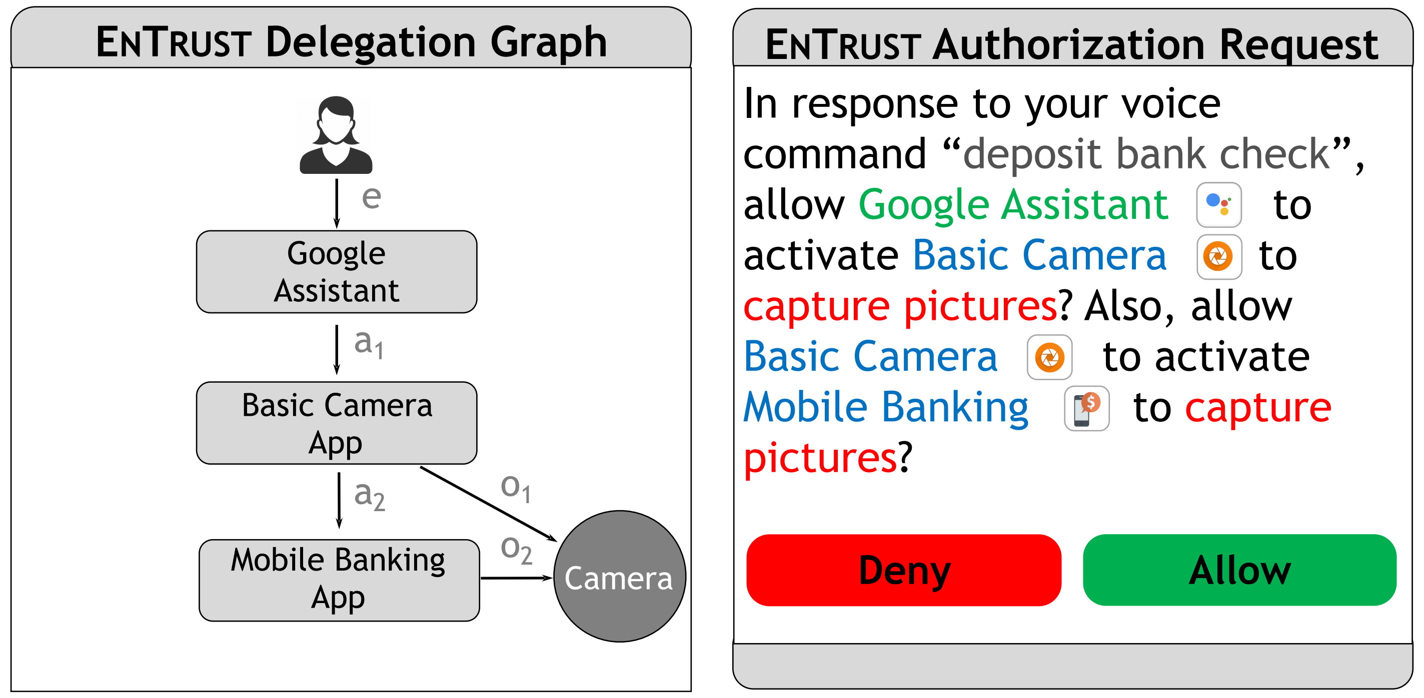
Directive: Ask *Google Assistant* to "deposit bank check" (After logging into *Mobile Banking* with the provided credentials, deposit the provided check)

Attack Scenario: (**Man-In-The-Middle**) *Google Assistant* launches *Basic Camera* registered for the voice intent "deposit bank check". The *Basic Camera* runs in the background, captures a picture of the check and - via a spoofed intent - launches the *Mobile Banking* app registered for the voice intent "deposit check". The collected data is sent to the remote service controlled by the adversary.

First-Use



EnTrust



	Group-FR Unprimed	Group-FR Primed	Group-EN Unprimed	Group-EN Primed
(Prompted)	47%	47%	100%	100%
(Explicit Allows)	13%	0%	7%	0%
(Attack Success)	67%	53%	7%	0%

Findings and Takeaway

PennState

- **Improved Attack Vectors Prevention:**

  Can reach 47-67% improvement compared to first-use authorization (**delegation path authorization**)

- **Compatible With Existing Programs:**

  No discernible slowdown, glitch, crashes, or responsiveness issues (no apps **modification** required)

- **Low User Decision Overhead:**

  No more than 4 explicit authorizations per program (**caching** of authorized delegation paths)

- **Negligible Performance Slowdown and Memory Overhead:**

  Less than 1% performance slowdown and 5.5 KB of cache per program

Thank You
For Your Attention!

Giuseppe Petracca

gxp18@cse.psu.edu
https://sites.psu.edu/petracca/

Backup Slides for Q&A

Study Demographics

PennState

- 69 **recruited subjects**, 34 (49%) were **female**.
- 36 (52%) were in the **18-25 years** old range, 27 (39%) in the **26-50 range**, and 6 (9%) were in above the **51 range**.
- 33 (48%) were **students** from our Institution, 9 of them (13%) were **undergraduate** and 24 (35%) were **graduate** students, 2 (3%) were Computer Science Majors.
- 11 (16%) worked in **Public Administration**, 9 (13%) worked in **Hospitality**, 6 (9%) in **Human Services**, 6 (9%) in **Manufacturing**, and 4 (6%) worked in **Science** or **Engineering**.
- All participants reported being **active smartphone users** (1-5 hours/day).
- 42 (61%) of the subjects were **long-term Android users** (3-5 years), others were **long-term iOS users**.
- Available participants as evenly as possible for both laboratory and field study.
- Each lab group had 9 long-term Android users, the remaining 6 long-term Android users participated in our field study.

Laboratory Study (Experimental Procedures)

PennState

Two Phases: (Users where not aware of the two phases)

Preliminary Phase:

- No attacks
- Meant to avoid “cold start”
- Users interacted with voice assistants
- Users authorized sensor operations at first-use

Attack Phase:

- Users interacted with programs performing 3 attacks

Randomized Order: In each phase, tasks were presented to users in a different randomized order

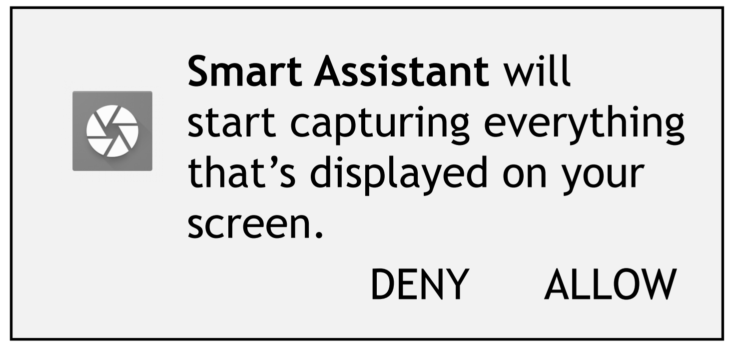
Laboratory Study (Confuse Deputy)

PennState

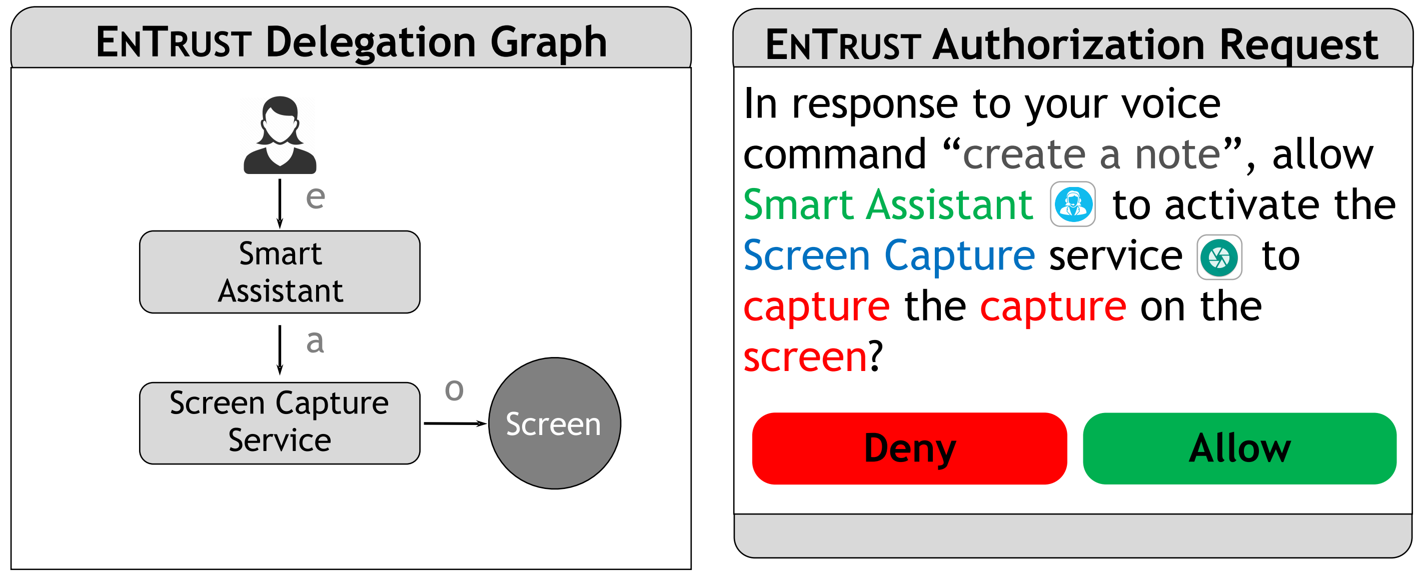
Directive: Ask *Smart Assistant* to “create a note”. Dictate a voice note to *Notes.* For example, “remind me to buy milk on the way home.”

Attack Scenario: (**Confused Deputy**) *Smart Assistant* launches *Notes* and adds the specified note, however, it also requests the *Screen Capture* service to capture the content on the screen. Credit card information and passwords, visible in the notes summary, are captured and sent to a remote server controlled by the adversary

First-Use



EnTrust



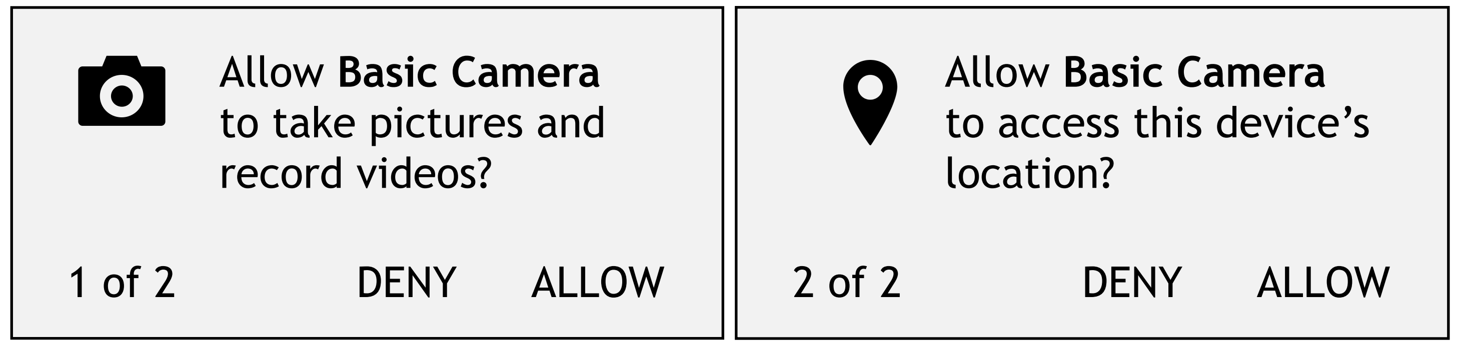
Group-FR Unprimed	Group-FR Primed	Group-EN Unprimed	Group-EN Primed	
40%	47%	100%	100%	(Prompted)
27%	0%	20%	0%	(Explicit Allows)
87%	53%	20%	0%	(Attack Success)

Laboratory Study (Trojan Horse)

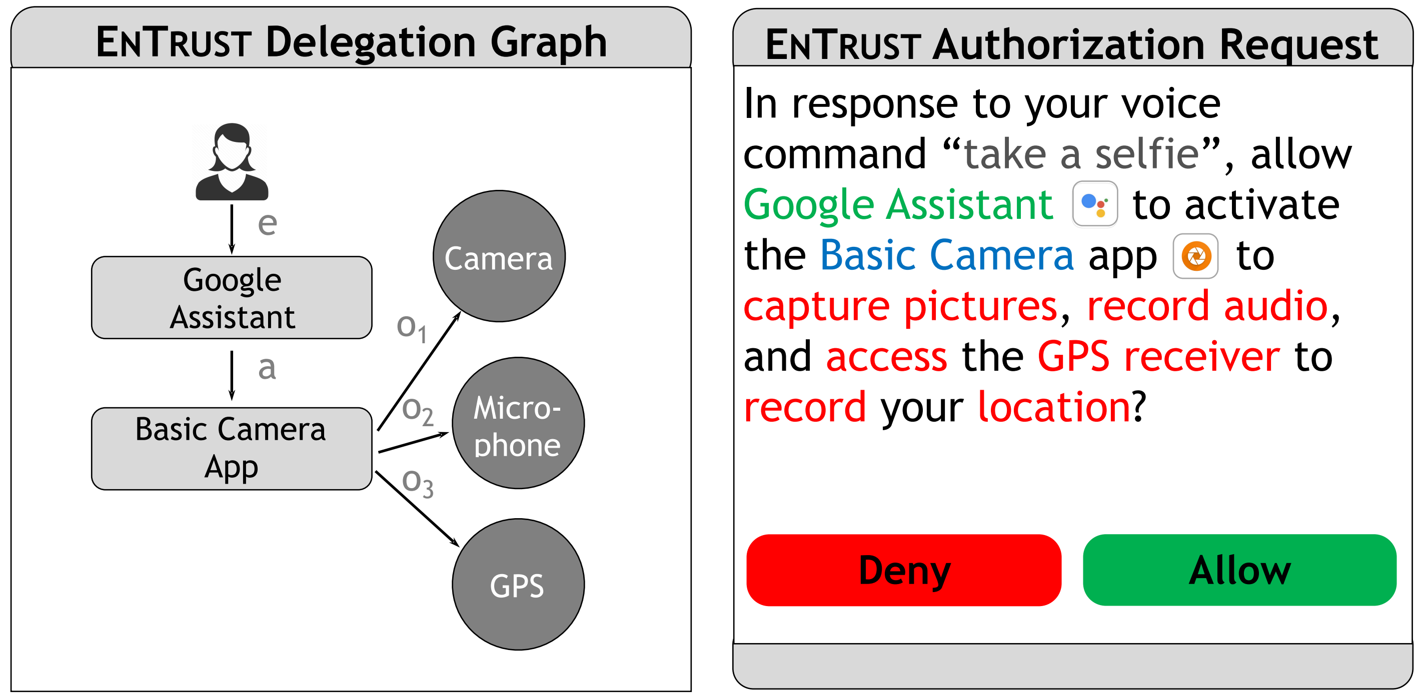
Directive: Ask *Google Assistant* to “take a selfie”

Attack Scenario: (**Trojan Horse**) *Google Assistant* activates the *Basic Camera* app, which is a Trojan app that takes a selfie but also records a short audio and the user’s location. The collected data is then sent to a remote server controlled by the adversary.

First-Use



EnTrust



Group-FR Unprimed	Group-FR Primed	Group-EN Unprimed	Group-EN Primed	
40%	53%	100%	100%	(Prompted)
20%	0%	13%	0%	(Explicit Allows)
80%	47%	13%	0%	(Attack Success)

Statistical Significance

Hypothesis: The information in *EnTrust* authorizations helps unprimed users identify attacks

Calculated the difference in explicit allows, across the three experimental tasks, for subjects in **Group-FR-U** versus subjects in **Group-EN-U**.

Result: Statistically significant difference ($\chi 2 = 19.3966$; $p = 0.000011$)

Hypothesis: *EnTrust* better helps primed and unprimed users in preventing attacks than first-use

Calculated the difference in successful attacks, across the three experimental tasks, for subjects in **Group-FR-U** and **Group-FR-P**, versus subjects in **Group-EN-U** and **Group-EN-P**.

Result: Statistically significant difference ($\chi 2 = 65.5603$; $p = 0.00001$)

Standard Bonferroni correction would be applied for multiple testing, but not necessary due to the small p-values.

Field Study (Experimental Procedures)

PennState

Loan Device:

- Pre-installed 5 voice assistants and 10 apps
- Mock accounts for apps requiring log-in
- Transferred participants' SIM card, data and apps (no data collected from such apps)

Required Actions:

- Everyday tasks for 7 days
- Pre-specified voice commands for pre-installed voice assistants
- Pre-specified action for pre-installed apps
- Free interaction with pre-installed apps

IRB Approved:

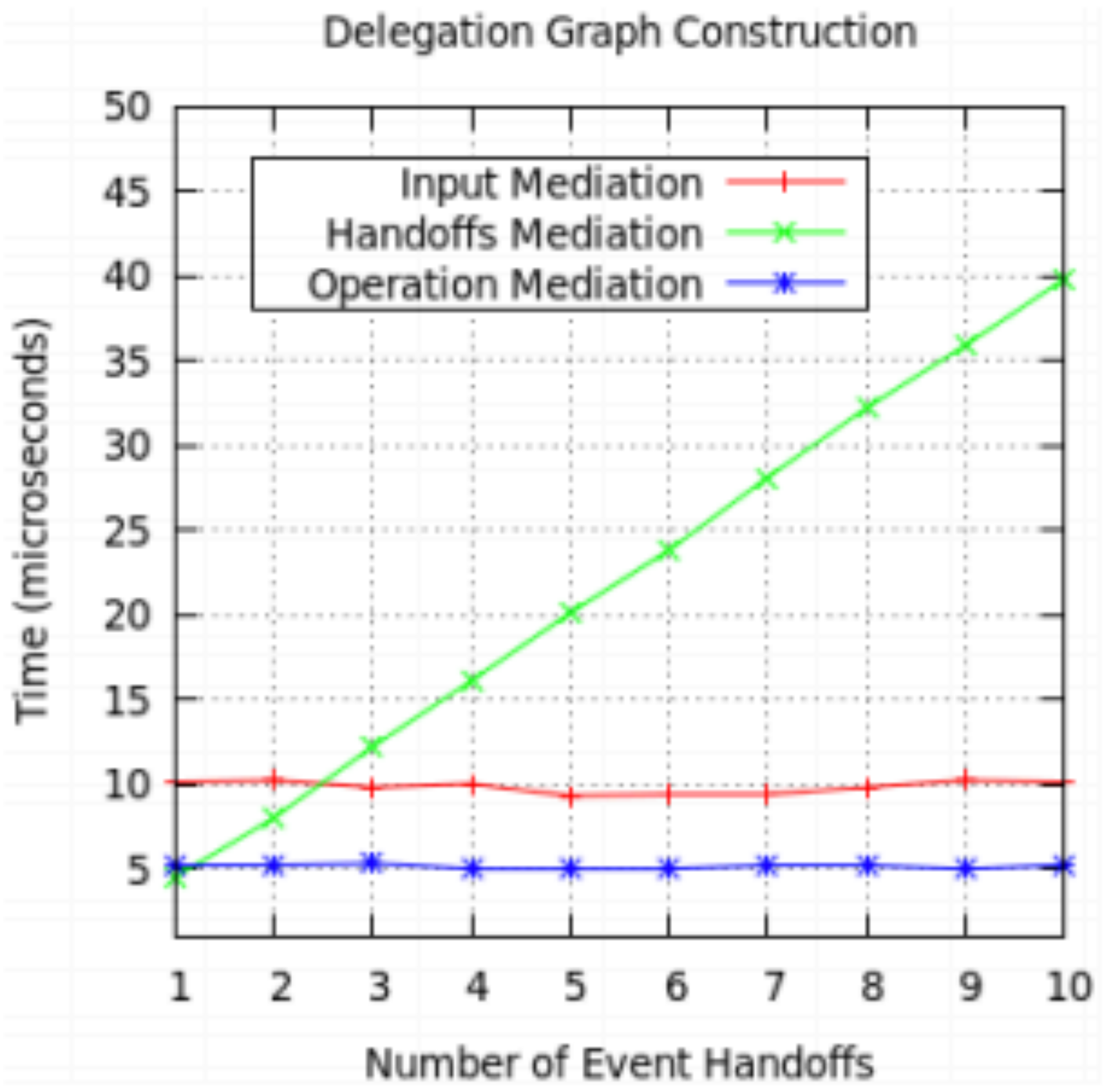
- Advertised as a generic "voice assistants and apps testing"
- No mentioning of security implications (mere propose was to measure decision overhead)

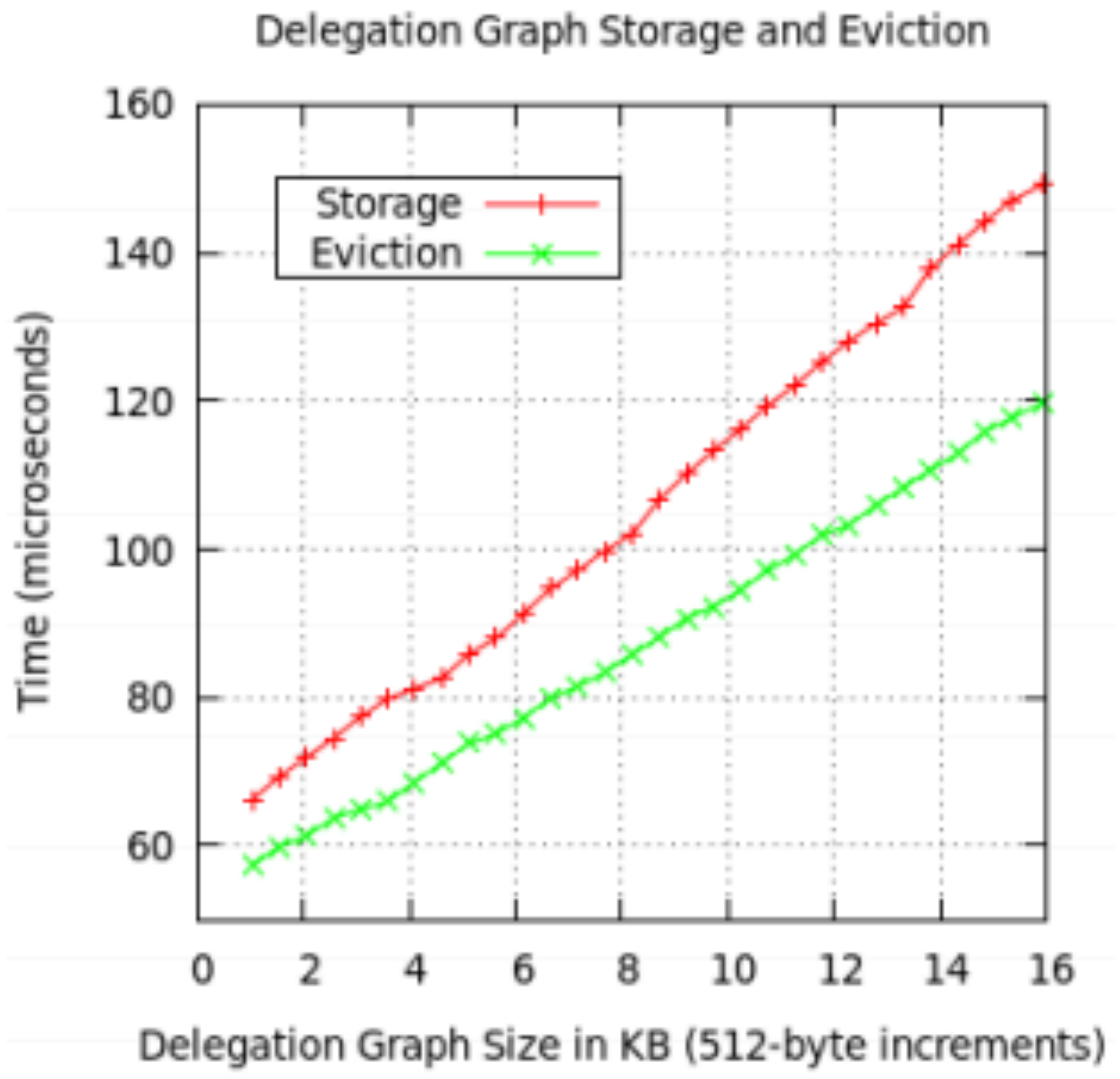
Field Study (Findings)

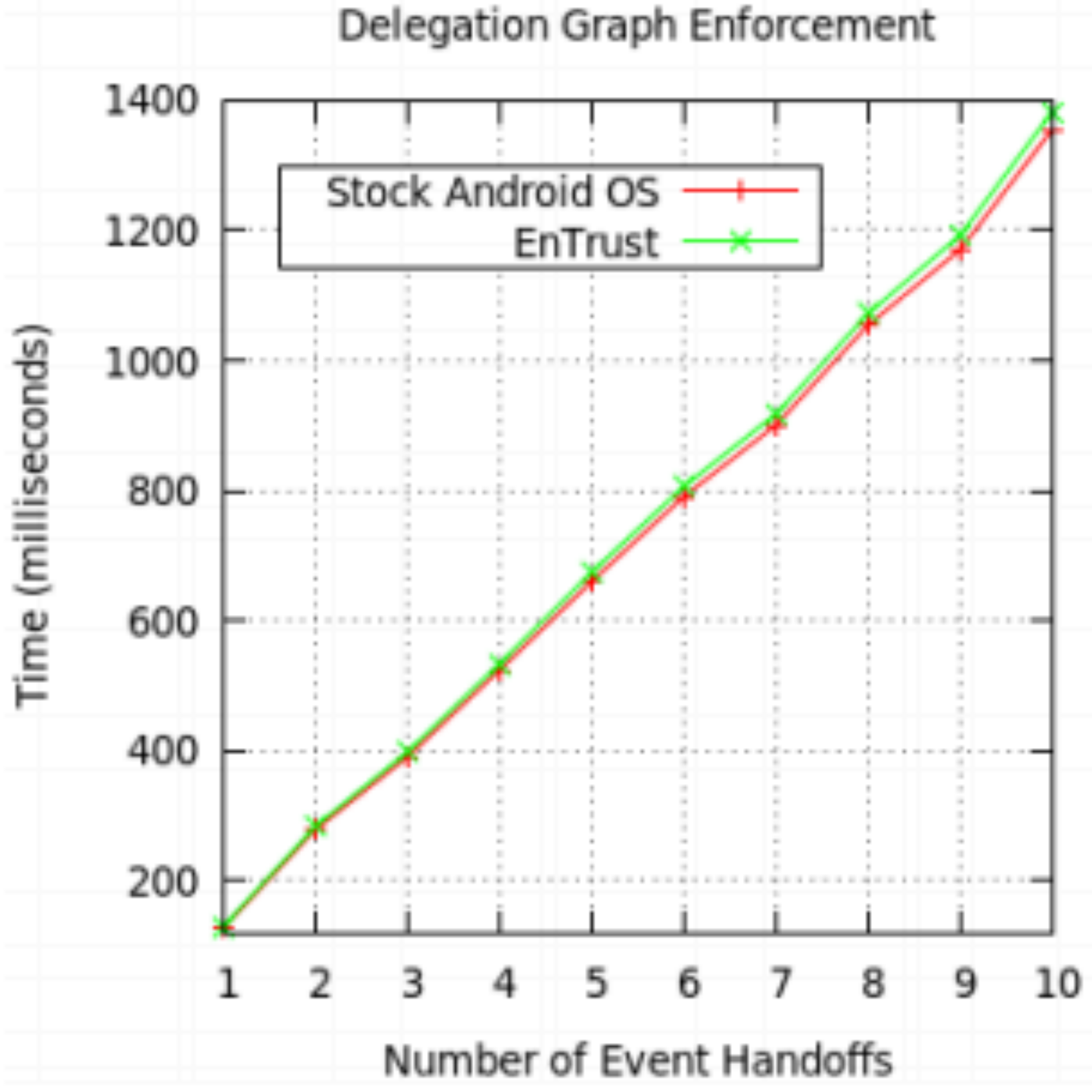
	Expl. Authorizations First-Use	Expl. Authorizations EnTrust	Impl. Authorizations in s 7 Days Period
Snapchat	3	3	276
YouTube	3	3	84
Facebook Messenger	2	2	93
Instagram	3	3	393
Facebook	3	3	117
Whatsapp	2	2	76
Skype	3	3	100
WeChat	2	2	101
Reddit	1	1	18
Bitmoji	3	3	127
Google Assistant	1	**4**	72
Microsoft Cortana	1	**3**	49
Amazon Alexa	1	**4**	84
Samsung Bixby	1	**4**	63
Lyra Virtual Assistant	1	**3**	56

Performance Measurements

PennState







Time Constraints Analysis