Data and Network Security

(Master Degree in Computer Science and Cybersecurity)

Lecture 7

Outline for today

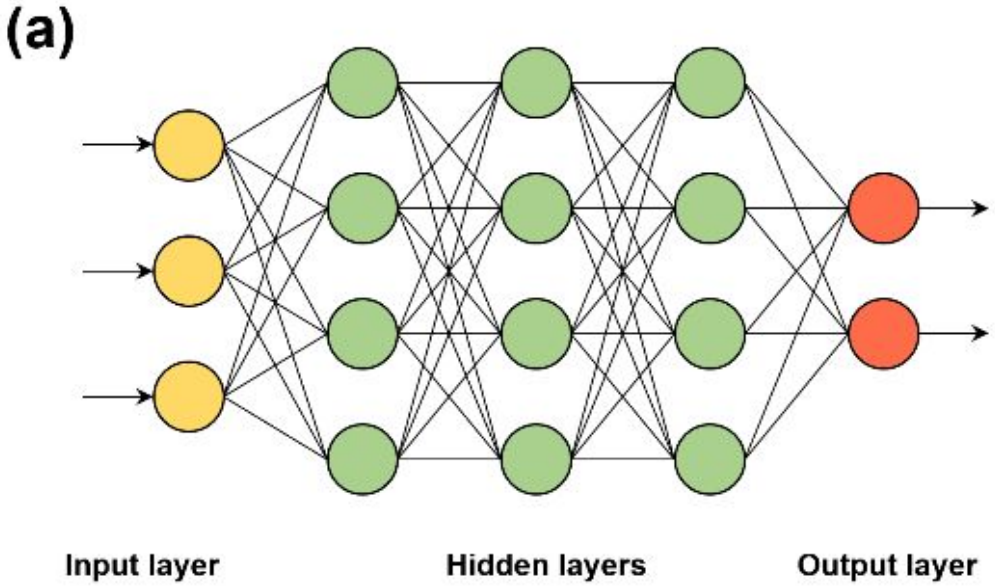
- Recap last lecture
- Internet of Things
- Remote attestation

Information leakage from ML models

Can I infer some (sensitive) property of the dataset used to train an ML model?

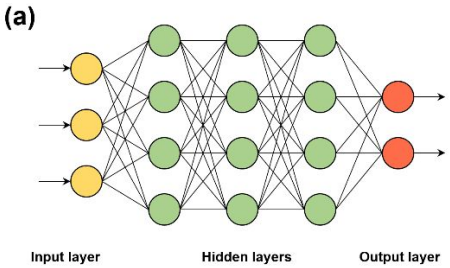
DATASET

Does this dataset have this property?

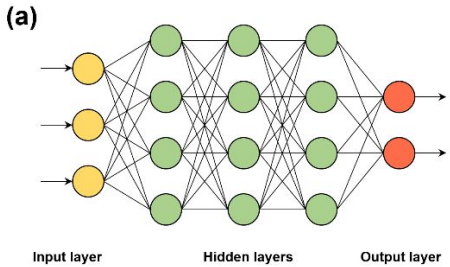


But you have only the model…

Does this dataset have this property?

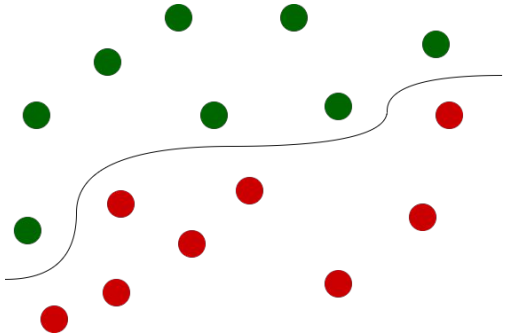


[f1,f2,f3,...fx]



[f1,f2,f3,...fx]

Train a binary classifier on these features



Information leakage from ML models

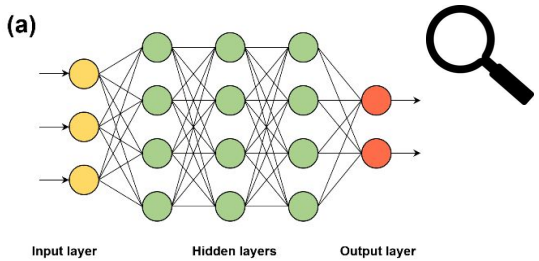
Was this datapoint part of this dataset?

DATASET

Information leakage from ML models

When we query the model with input data, we observe the model's responses:

- **predicted labels,**
- **probabilities, or scores assigned to different classes.**

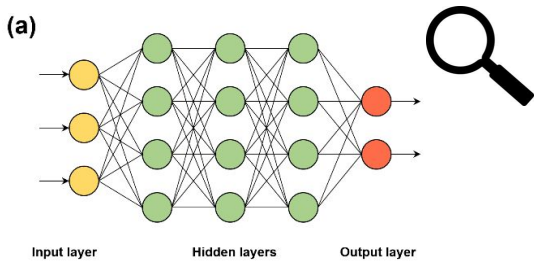


*distinguishing features in these responses that can indicate whether the input data was likely part of the training dataset

Distinguishing features?

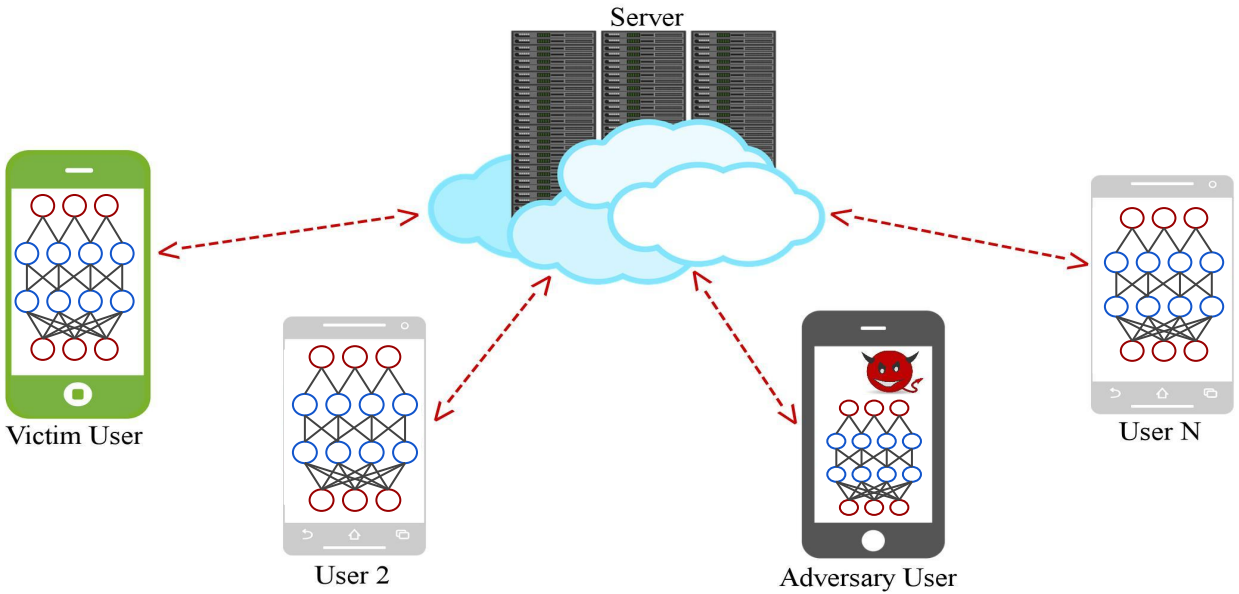
One key indicator that a data point was part of the training dataset is overfitting.

Overfitting occurs when a model learns to memorize specific examples from the training data rather than capturing general patterns.



If a model exhibits overfitting, it may produce responses that are overly confident or precise for data points seen during training but less accurate for unseen data.

Collaborative Learning Scheme



Adversary's goal?
Reconstruct private samples from the dataset of the victim indirectly influencing the learning of other participants

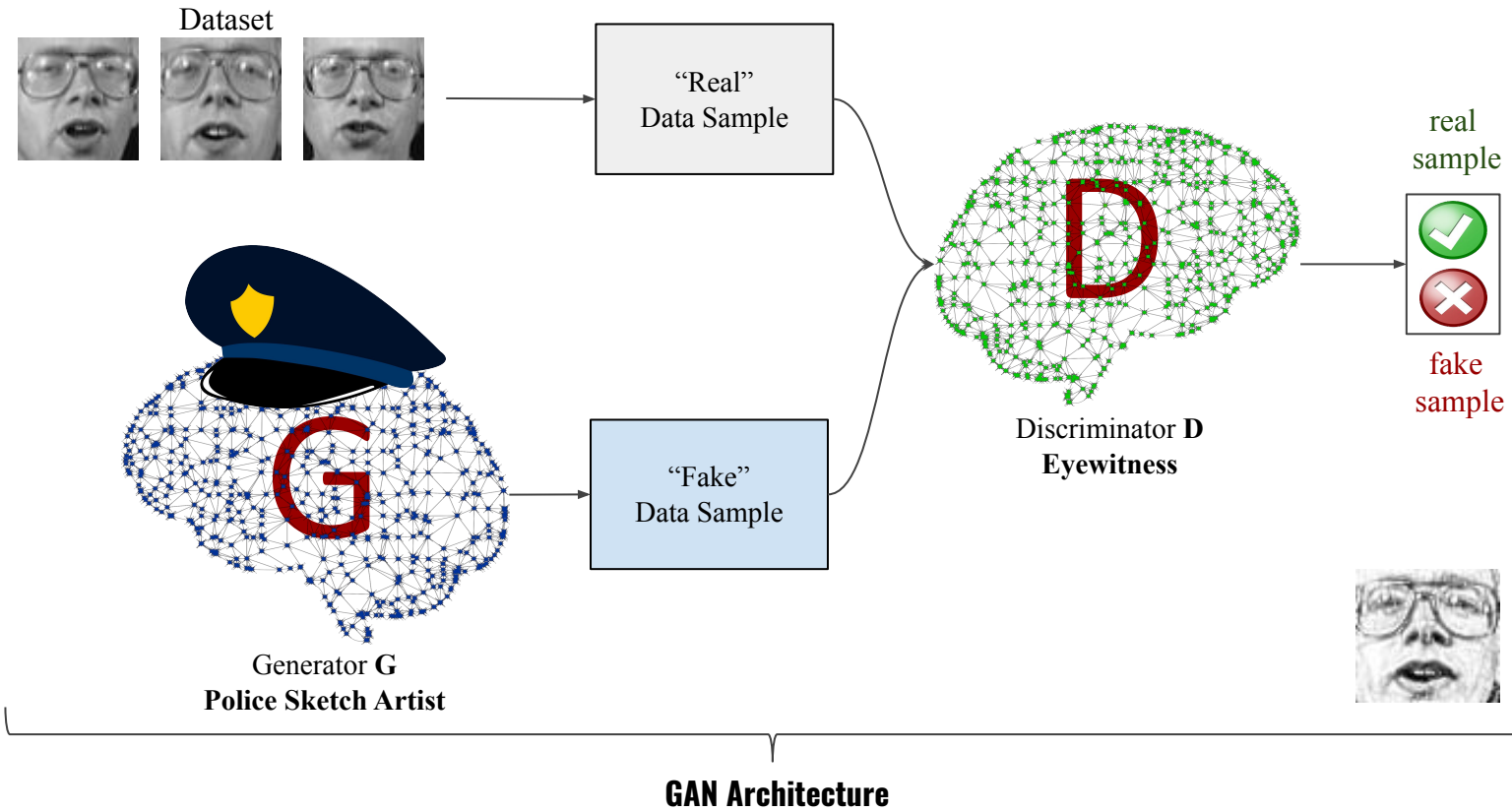
How should the adversary behave?

- **The adversary should operate as an participant within the privacy-preserving collaborative deep learning protocol.**

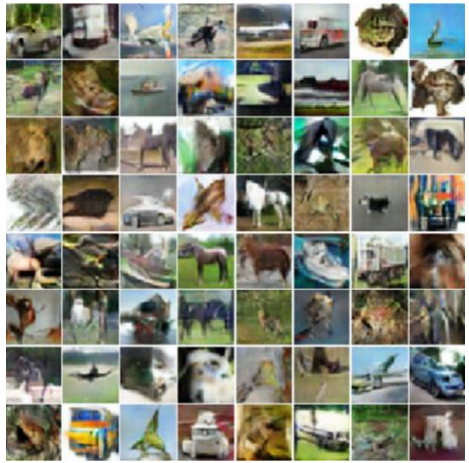
- **The objective of the adversary is to infer meaningful information about a label that he does not own.**

- **The adversary does not compromise the global parameter server that collects and distributes parameters to the participants.**

Generative Adversarial Network



MNIST images



CIFAR-10 images

faces

album covers

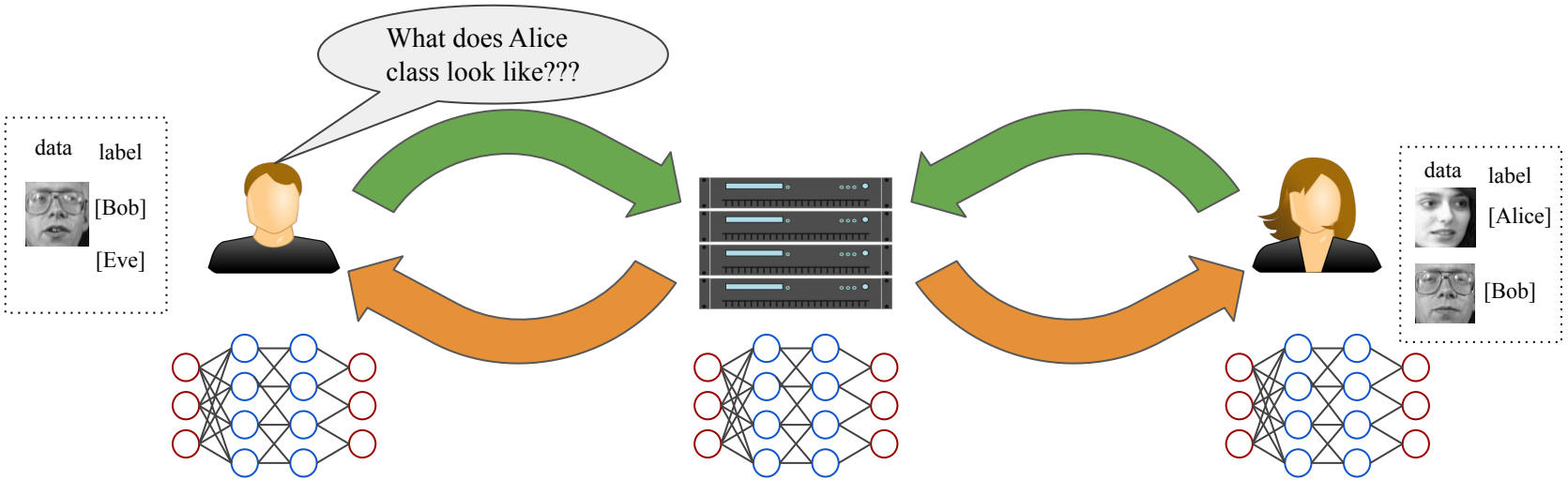
bedrooms

GAN results in the literature

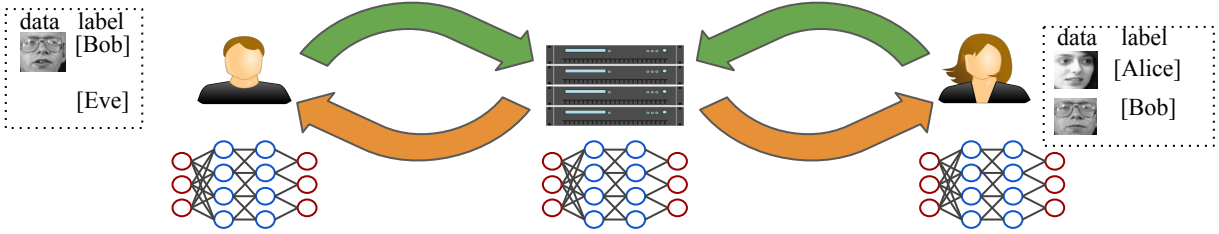
images from:

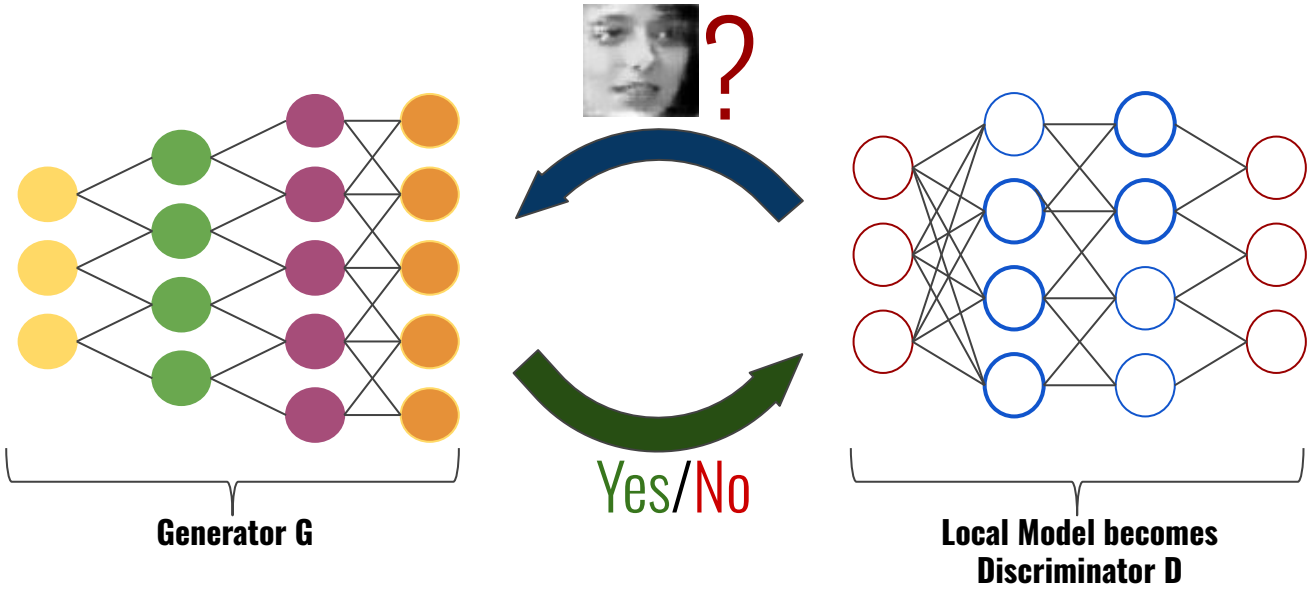
- https://blog.openai.com/generative-models/
- Goodfellow et al. Generative Adversarial Networks
- Radford et al. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Violating the privacy…

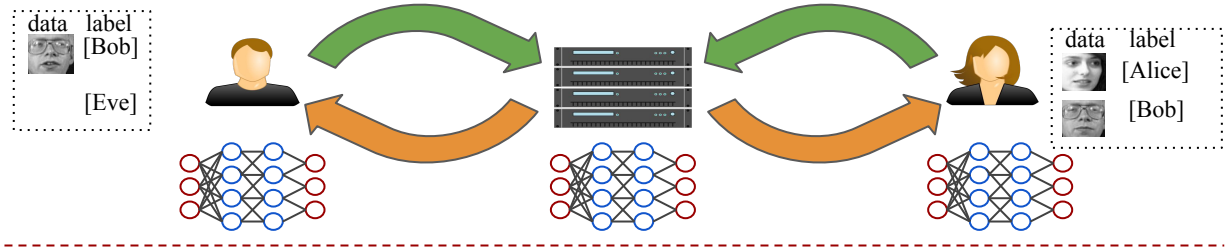


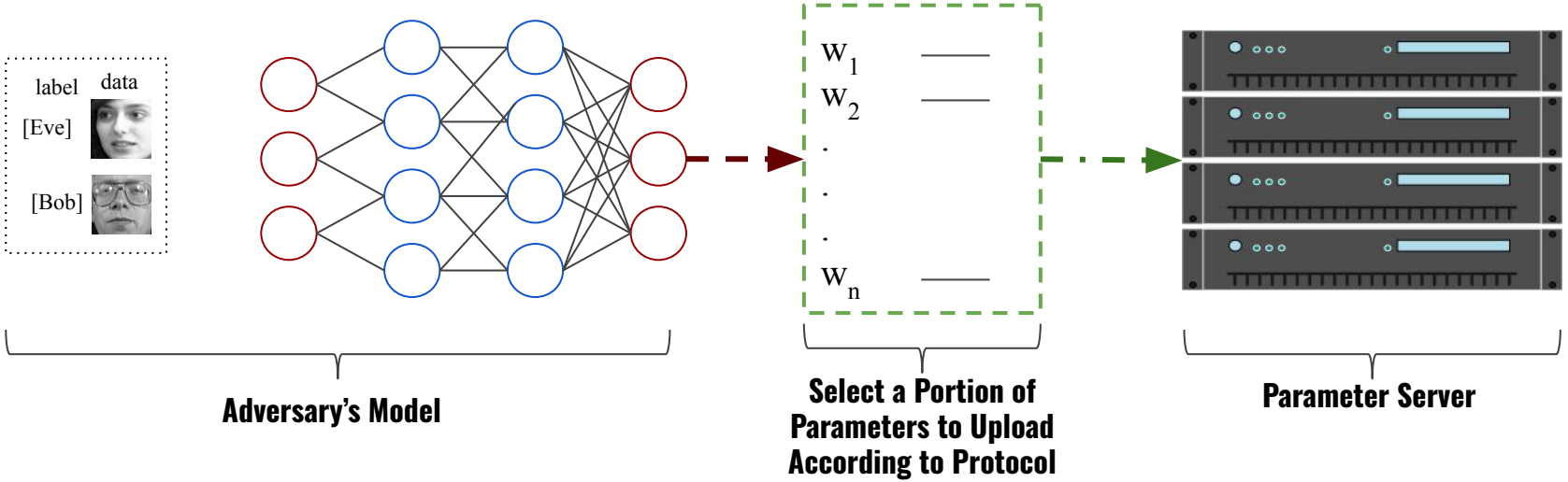
Adversary's Turn





Adversary's Turn





Experiments without Differential Privacy

Actual Images

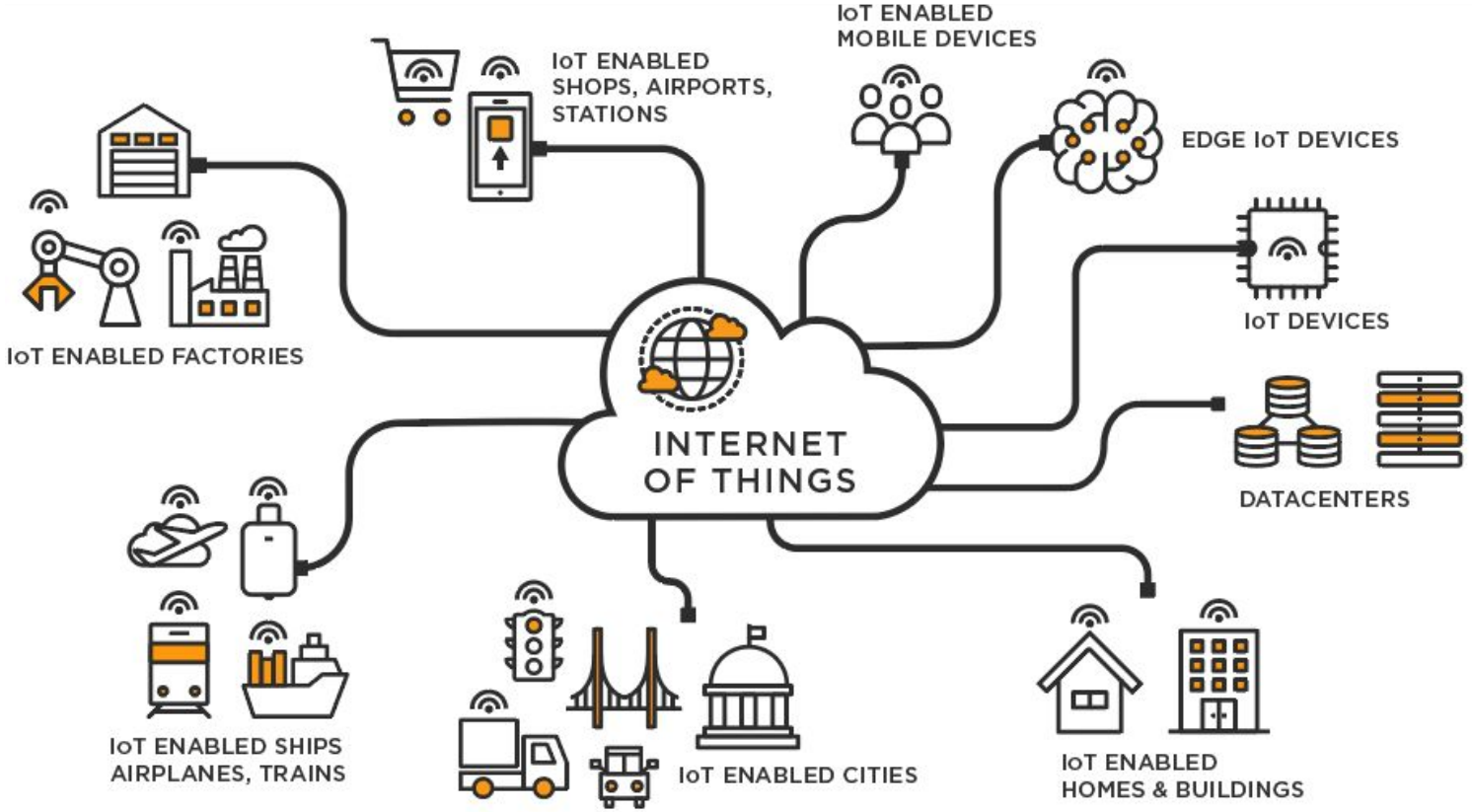
Generated Data

Original vs Generated

Outline for today

- Recap last lecture
- **Internet of Things**
- Remote attestation

Internet of Things (IoT)



Internet of Things (IoT)

2008 – number of things connected to the internet was greater than the number of people living on earth.

Internet of Things (IoT)

2020 – number of things connected to the internet was over **50 billion.**

Internet of Things (IoT) - Definition

Network of physical objects or "things" embedded with sensors, software, and other technologies that enable them to **connect and exchange data** with other devices and systems over the internet.

Internet of Things (IoT) - Purpose

Creation of a seamless network where devices can communicate and interact with each other autonomously, leading to increased:

- efficiency,
- automation,
- improved decision-making.

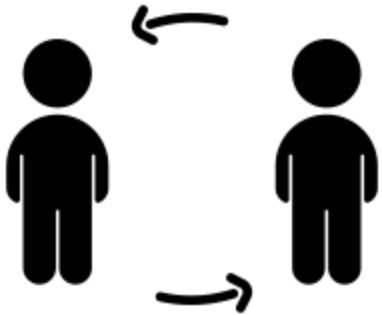
Internet of Things (IoT) - Purpose

By collecting and analyzing data from various sources, IoT systems can:

- provide valuable insights,
- optimize processes,
- enable new services and applications across numerous domains (healthcare, transportation, agriculture, manufacturing, and smart cities).

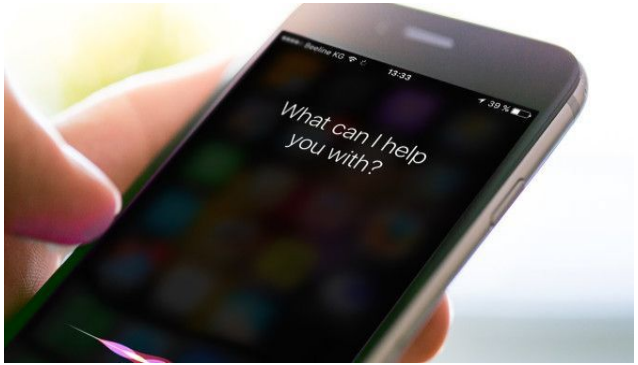
Internet of Things (IoT) - Trust

- **Trust**
  - to believe that someone is good and honest and will not harm you.



Internet of Things (IoT) - Trust

- **Trust**
  - to believe something is safe and reliable



Do you trust your mobile?

Does the door trust the camera?

Internet of Things

- Many interconnected devices
- Resource constrained devices (i.e. small memory, CPU…)
- Deployed in safety-critical environments.

Internet of Things - Threats

ANDY GREENBERG SECURITY 07.21.15 06:00 AM

HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

About 90% of Smart TVs Vulnerable to Remote Hacking via Rogue TV Signals

How Israel Caught Russian Hackers Scouring the World for U.S. Secrets

Exploiting the popular Kaspersky antivirus software, Russian hackers searched millions of computers for American intelligence keywords. Israeli intelligence tipped off American officials.

By NICOLE PERLROTH and SCOTT SHANE

Why do these malicious attacks happen?

- Default passwords
- Cleartext communication
- Software bugs
- Non-appropriate hardware protection

IoT vulnerabilities - a study

- **25** vulnerabilities per device
- Over **80%** did not require sufficiently complex and long passwords
- Over **70%** did not encrypt local and remote traffic comm.
- Over **60%** contained vulnerable user interfaces and/or vulnerable firmware.

IoT vulnerabilities

The **S** in **IoT** stands for **Security**

Tim Kadlec

IoT devices - a summary from security perspective

Attractive target

- Contain sensitive and private information

IoT devices - a summary from security perspective

Attractive target
- Contain sensitive and private information

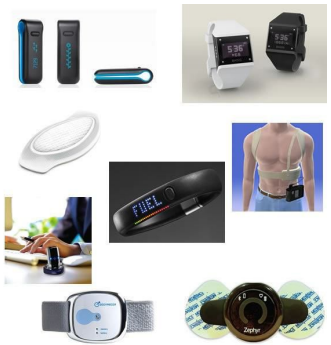
Easy to exploit
- Do not support complex security techniques

IoT devices - a summary from security perspective

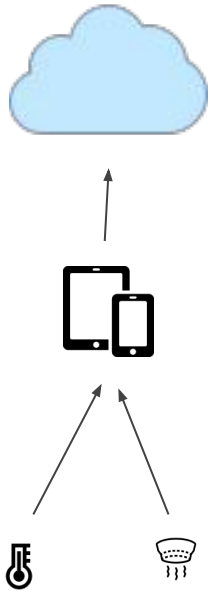
Attractive target
- Contain sensitive and private information

Easy to exploit
- Do not support complex security techniques

Amplify the attack impact
- Control physical environment
- Access personal information
- Spread to other interconnected devices



Scenario: Uploading the data from an Iot device



Cloud

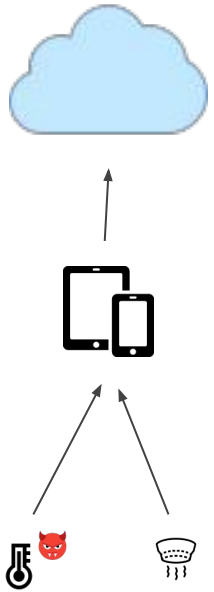
Gateway

IoT device

Data reported from the temperature sensor

```
{
  "sensor": "dht11",
  "data": {
    "temperature": "30.1",
    "humidity": "70.1"
  }
}
```

Scenario: Uploading the data from an Iot device



Cloud

Gateway

IoT device

Data reported from the temperature sensor

```
{
  "sensor": "dht11",
  "data": {
    "temperature": "30.1",
    "humidity": "70.1"
  }
}
```

Malware can:
- Extract secrets
- Modify software
- Alter behaviour

Are these data trustworthy?

What can we put as a requirement?

One could be able to decide whether or not to trust the data reported by sensors.

What can we do?

- **Prevent device compromise?**

What can we do?

- **Prevent device compromise? - Difficult**

What can we do?

- **Prevent device compromise? - Difficult**

- **Detect compromised node (IoT device) to isolate from the network**

What can we do?

- **Prevent device compromise? - Difficult**

- **Detect compromised node (IoT device) to isolate from the network - Detection tools**

How to detect malicious presence?

Guarantee that Prover (the device) is “telling the truth” even when it is infected by malware.

How to detect malicious presence?

Verify that each device is NOW running the initial application.

- **Verifier: A trusted entity, not always present, not possible to physically reach a device**
- **Prover: An untrusted platform**

Outline for today

- Recap last lecture
- Internet of Things
- **Remote attestation**

Remote attestation

Remote attestation:

- Security mechanism used to verify the integrity and trustworthiness of a device's software and hardware configuration remotely.

- This process allows a trusted entity, such as a server or a cloud-based service, to **assess whether an IoT device is running the expected software and hasn't been tampered with or compromised.**

Remote attestation

Remote attestation:

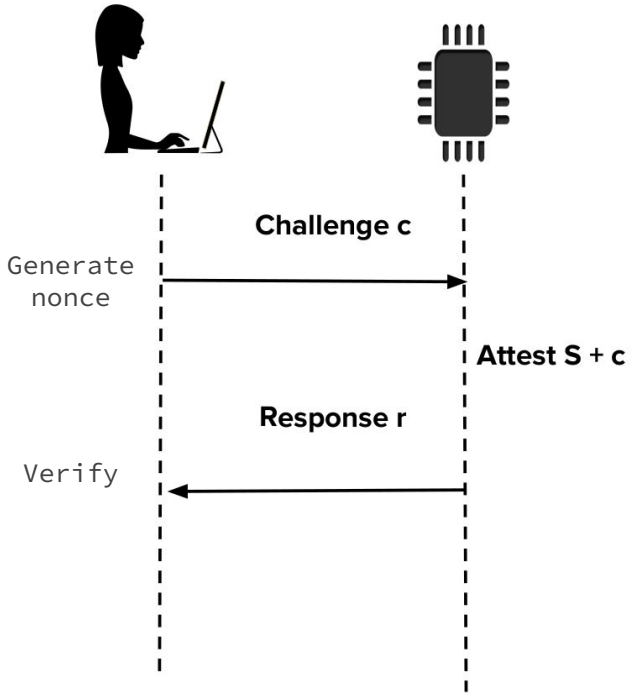
- Security protocol
- Two parties (trusted verifier, untrusted prover)
- Remote verification of the untrusted prover

Remote attestation - Overview

Challenge (Executed by Verifier)
- Outputs a random Challenge (nonce, timestamp, memory addresses, attestation routine)

Attest (Executed by Prover)
- Computes a small attestation response based on internal state S and challenge c

Verify (Executed by Verifier)
- Compared the response received from Prover with the expected state



Adversary models

- Software Adversary
- Hardware Adversary

Adversary models

- Software Adversary
  - Remote: Infect device(s) with malware
  - Local: Learn device secret, impersonate or clone, can launch side channel attack
  - Mobile adversary: Relocates or deletes itself

Adversary models

- Hardware Adversary
  - Stealthy Physical Intrusive: Capture device and physically extract secrets, clone device(s)
  - Physical Intrusive: Capture device and modify contents/components

Requirements of Remote attestation

Challenge (Executed by Verifier)

- Authentic, fresh, unpredictable

Attest (Executed by Prover)

- Authentic, Atomic, Unforgeable, Dynamic, Deterministic

Verify (Executed by Verifier)

- deterministic

Approaches of Remote attestation

Memory
- Static vs. Dynamic

Number of device
- Single vs. swarms

Design
- Software based, Hybrid, or Hardware

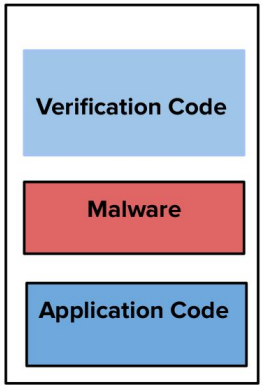
Software-based attestation

- No hardware features to support attestation
  - No secrets on prover (e.g., no Attestation Key)

- Relies on two pillars:
  - Tight time constraints
  - Lack of free space to store malicious code

Software-based attestation

- Attestation protocol reads memory and computes a checksum
- Attacker must offset the memory reads to avoid detection: delay
- Verifier measures execution time to detect the malware



Desired security property

Verifier’s check is successful if and only if:

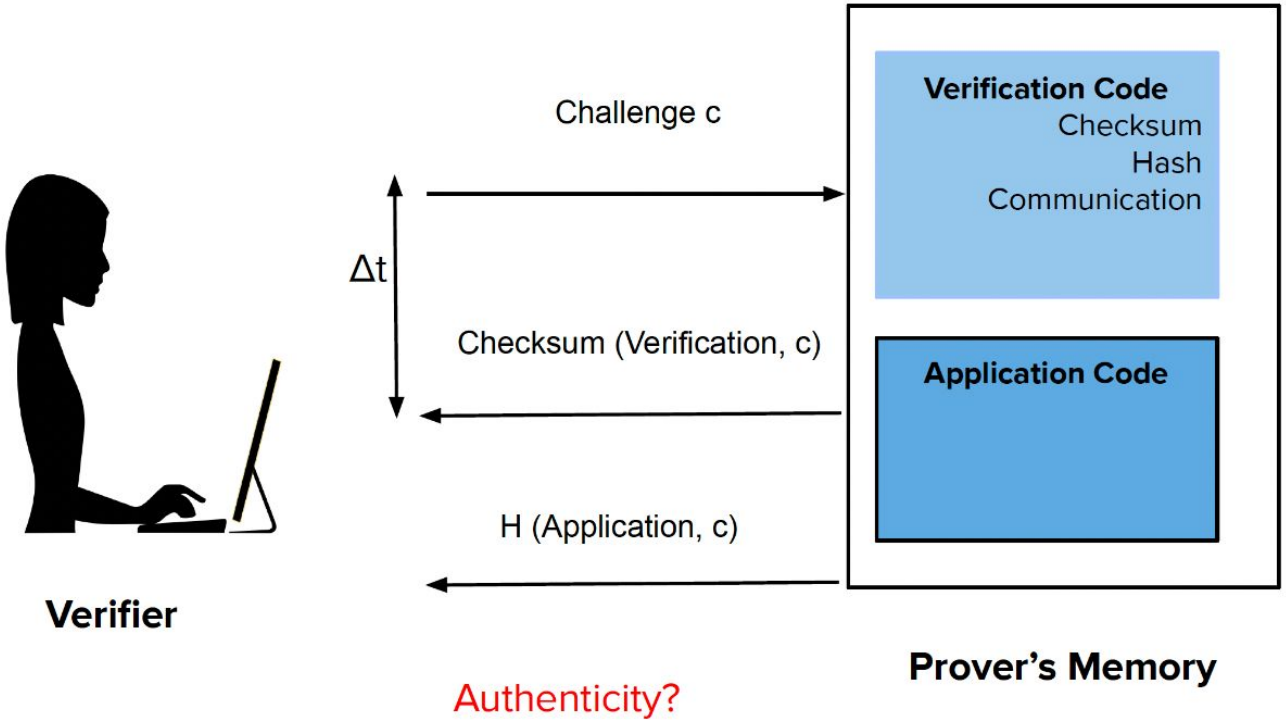
- Verification function is unmodified
- Untampered execution environment is established

Desired security property

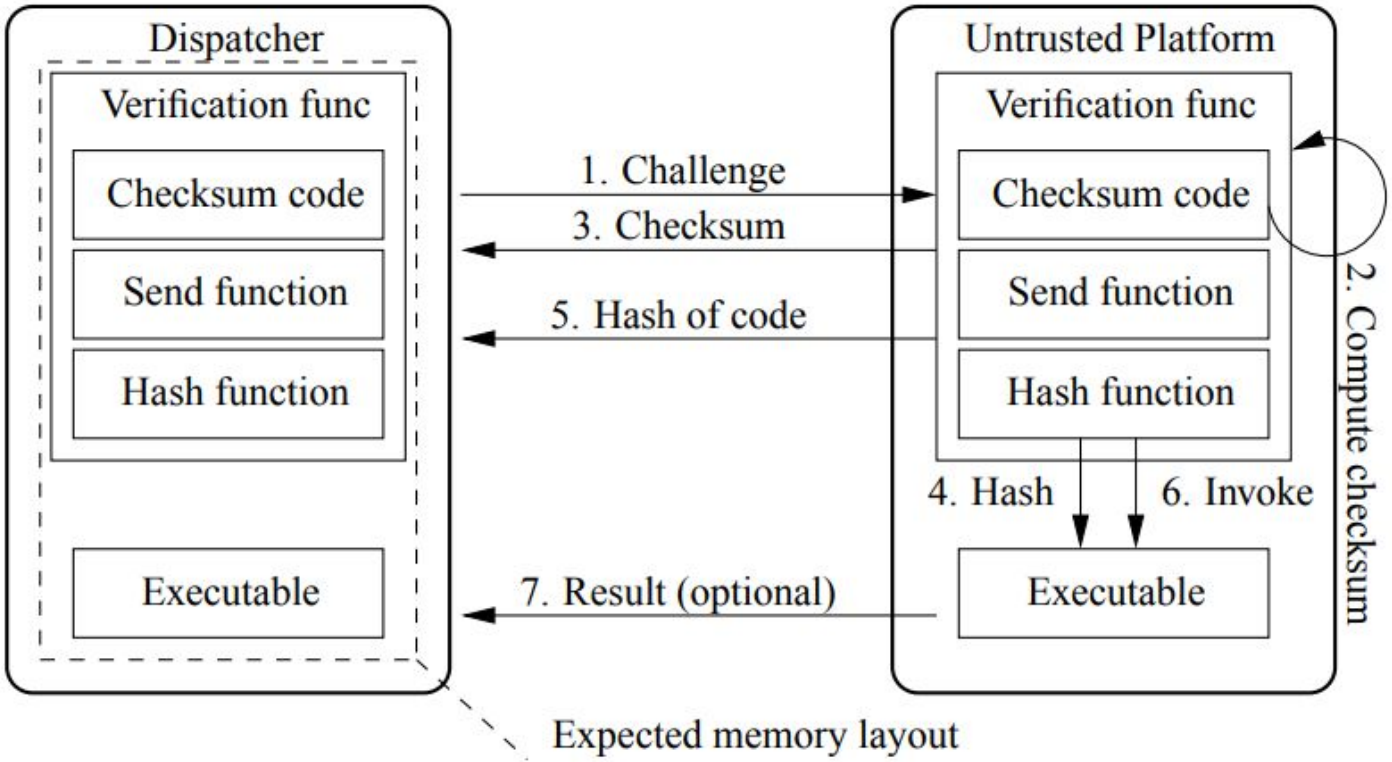
Verifier's check is successful if and only if
- Verification function is unmodified
- Untampered execution environment is established

Intuition: Checksum is incorrect or checksum computation slows down if the attacker:
- Modifies verification function and forges correct checksum, or
- Fakes the creation of untampered code execution environment

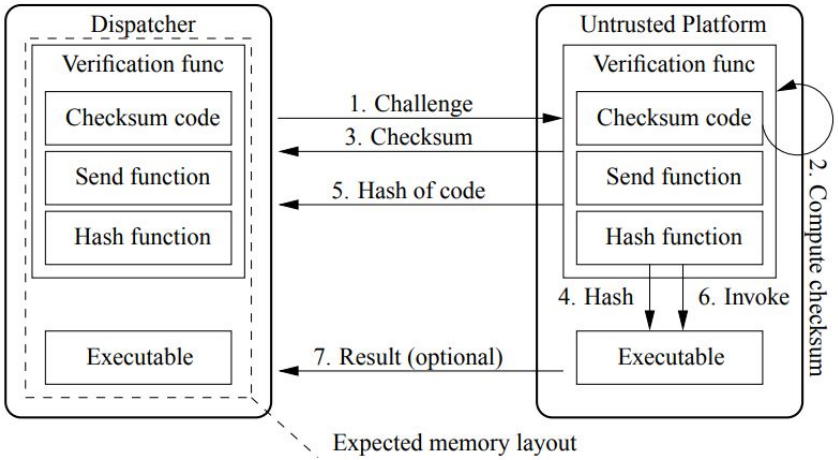
Software based attestation - Pioneer



Software based attestation - Pioneer



Software based attestation - Pioneer

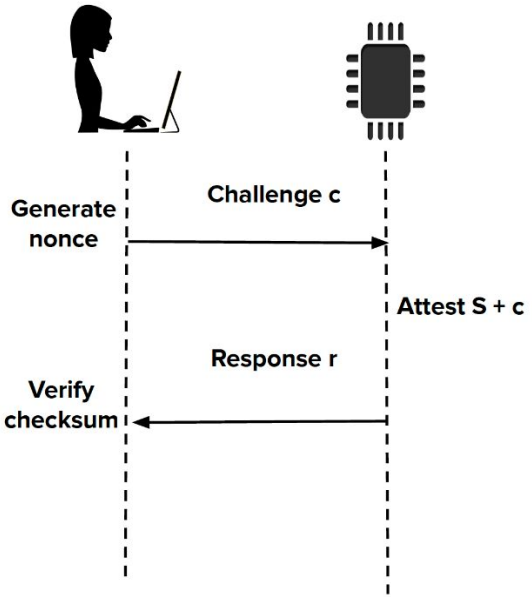


1. D: $t_1 \leftarrow$ current time, $nonce \xleftarrow{R} \{0,1\}^n$
   $D \rightarrow P$: $\langle nonce \rangle$
2. P: $c \leftarrow \text{Checksum}(nonce, P)$
3. $P \rightarrow D$: $\langle c \rangle$
   D: $t_2 \leftarrow$ current time
   if $(t_2 - t_1 > \Delta t)$ then exit with failure
   else verify checksum c
4. P: $h \leftarrow \text{Hash}(nonce, E)$
5. $P \rightarrow D$: $\langle h \rangle$
   D: verify measurement result h
6. P: transfer control to E
7. $E \rightarrow D$: $\langle$result (optional)$\rangle$

Software based attestation

V: c ←$ Challenge()
V→P: c
V: Tc←current time
P: r←Attest(S, c)
P→V: r
V: Tr ←current time
V: Verify(S, c , r, Ta, Tc, Tr)

S- internal state	Ta- attestation time
c- challenge	Tc- challenge time
r- response	Tr- response time



Attacks against software-based attestation

- Compression attack
  - Compress code to make space for attack code
  - Decompressed on-the-fly during attestation

Attacks against software-based attestation

- Compression attack
  - Compress code to make space for attack code
  - Decompressed on-the-fly during attestation

- Return-oriented rootkit
  - Install a rootkit that hides itself in non-executable memories
  - Use ROP (Return-Oriented Programming) to implement this attack.

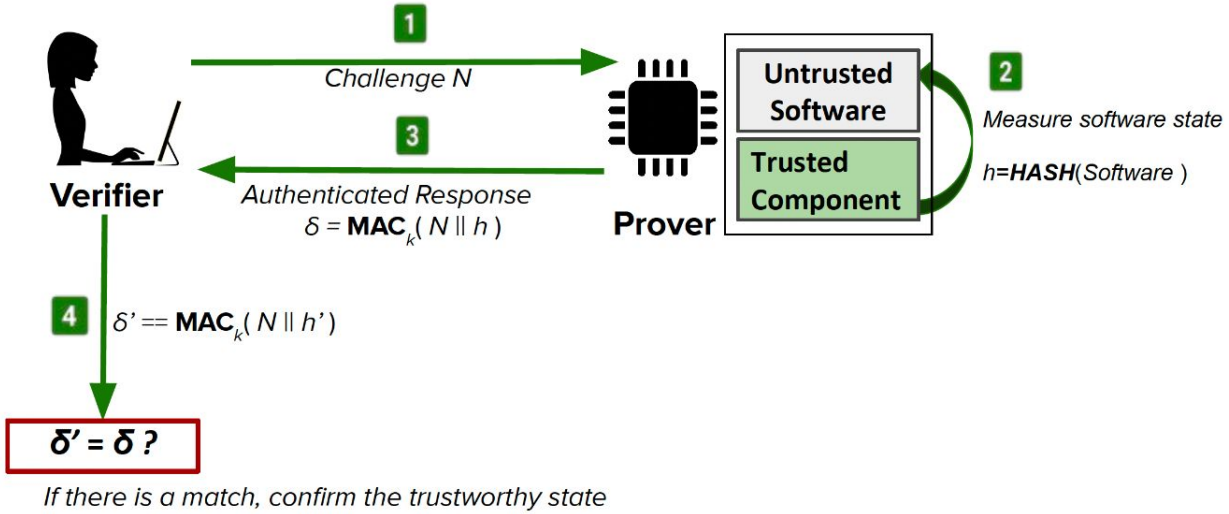
Limitations of software-based attestation

- Limitations:
  - verifier must know exact hardware configuration
  - difficult to prove time optimality
  - assumes “adversarial silence” during attestation
  - limited to “one-hop” networks

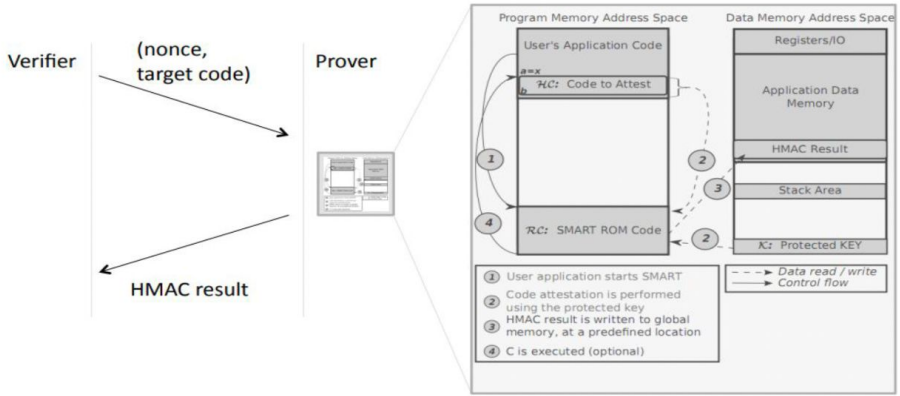
- Advantages:
  - No hardware requirements

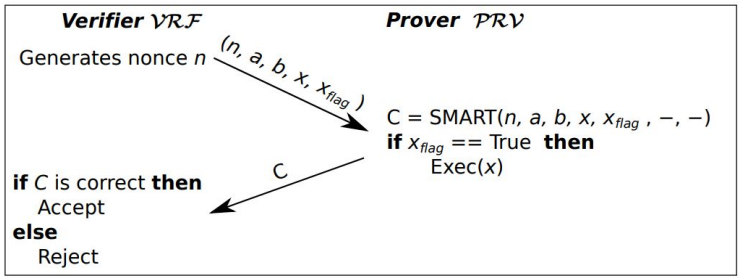
Hybrid attestation

Prover and Verifier share a key k
- N is a random nonce
- Verifier expects configuration h'



Hybrid attestation - SMART protocol



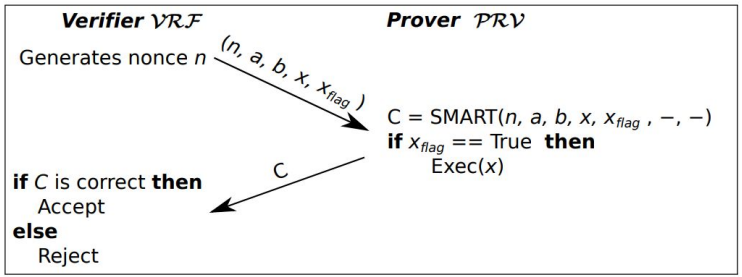


Hybrid attestation - SMART protocol

- Prover and Verifier share a secret key K.

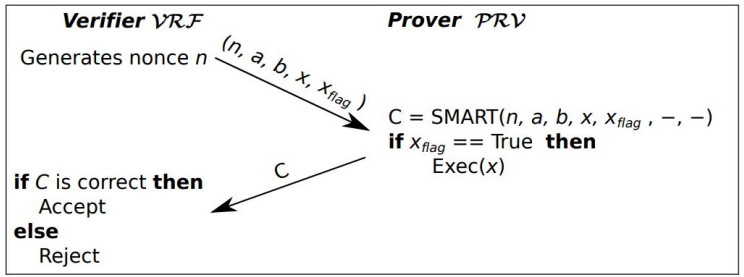
- Verifier sends several parameters to Prover: attestation region boundaries a and b; address x where Prover optionally passes control after attestation if x_{flag} is set; and nonce *n* to prevent replay attacks.

- A ROM-resident code segment on Prover computes a cryptographic checksum C of a region [a, b] in Provers memory (using nonce n) and then passes control to x.



Hybrid attestation - SMART protocol

- Prover and Verifier share a secret key K.
- Verifier sends several parameters to Prover: attestation region boundaries a and b; address x where PRV optionally passes control after attestation if x_{flag} is set; and nonce n to prevent replay attacks.
- A ROM-resident code segment on Prover computes a cryptographic checksum C of a region [a, b] in Provers memory (using nonce n) and then passes control to x.



- After execution of code starting at x, Prover returns C to Verifier.
- Verifier verifies correctness of C by re-computing it using the same parameters and K.

Hybrid attestation

Advantages

- Can be used across a network/over an untrusted channel
- Verifier does not need to know exact hardware configuration of the Prover

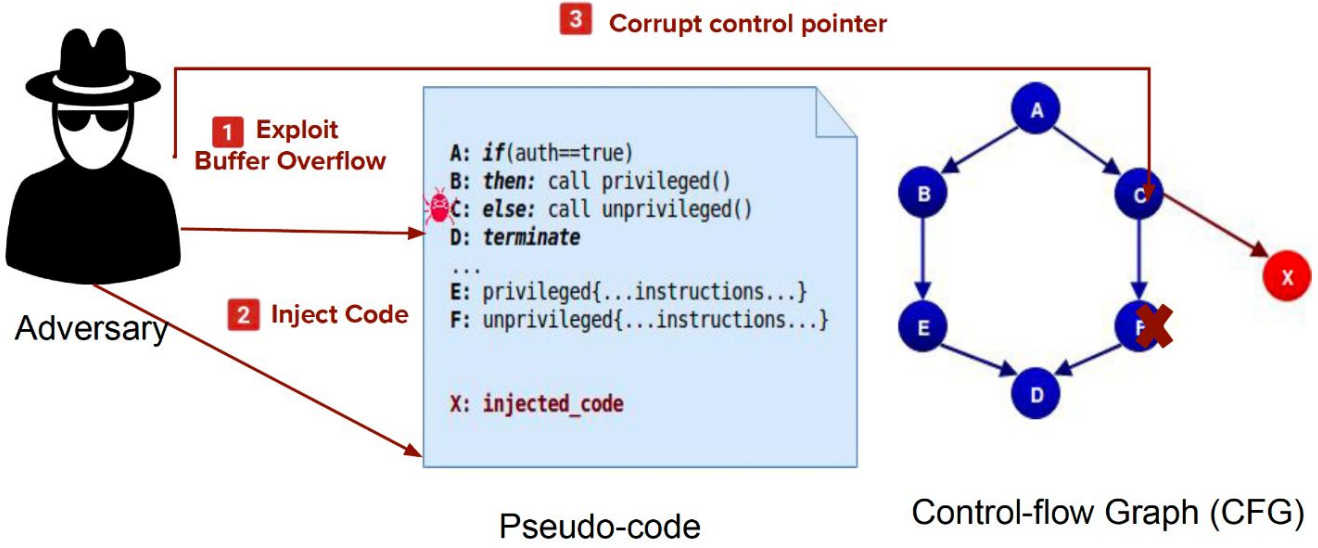
Disadvantages

- Needs additional hardware support

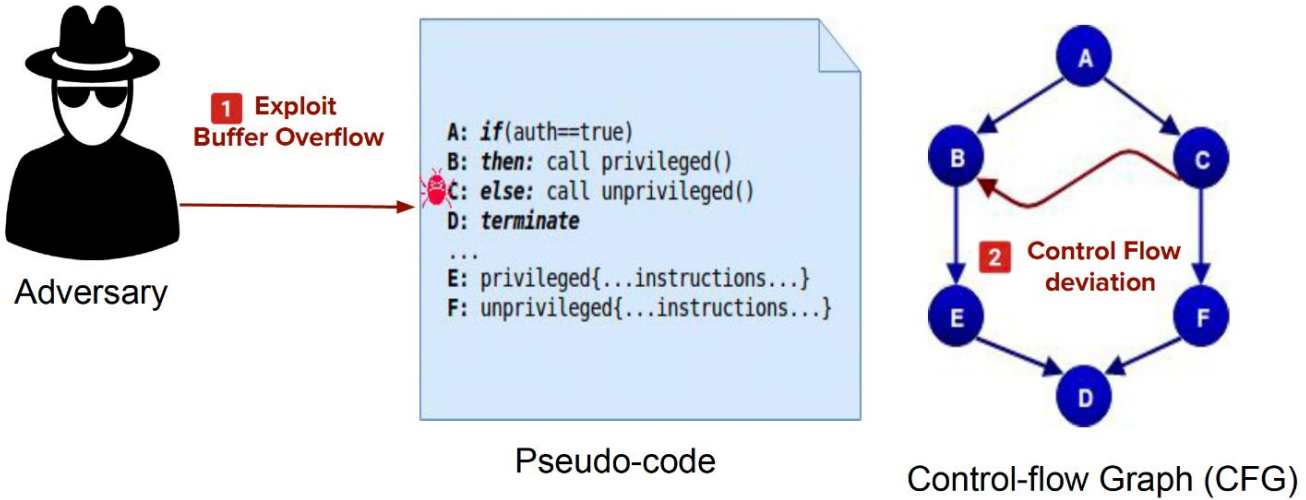
Attestation...

Program Memory Attestation schemes do not address runtime attacks

Code injection attack



Code reuse attack



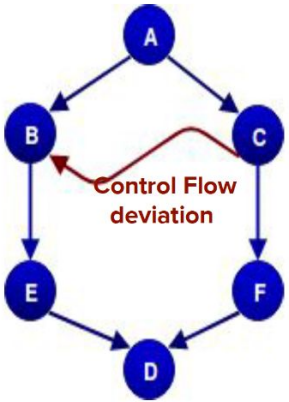
Adversary

Pseudo-code

Control-flow Graph (CFG)

Dynamic attestation

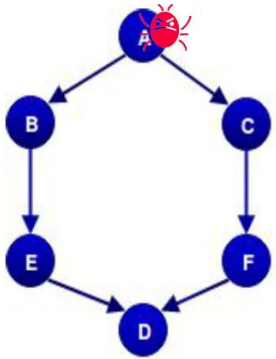
Control-Flow Attack
corrupt code pointer




```
A: if(auth==true)
B: then: call privileged()
C: else: call unprivileged()
D: terminate
...
E: privileged{...instructions...}
F: unprivileged{...instructions...}


X: injected_code
```

Non-Control-Data Attack
corrupt data pointer/variable

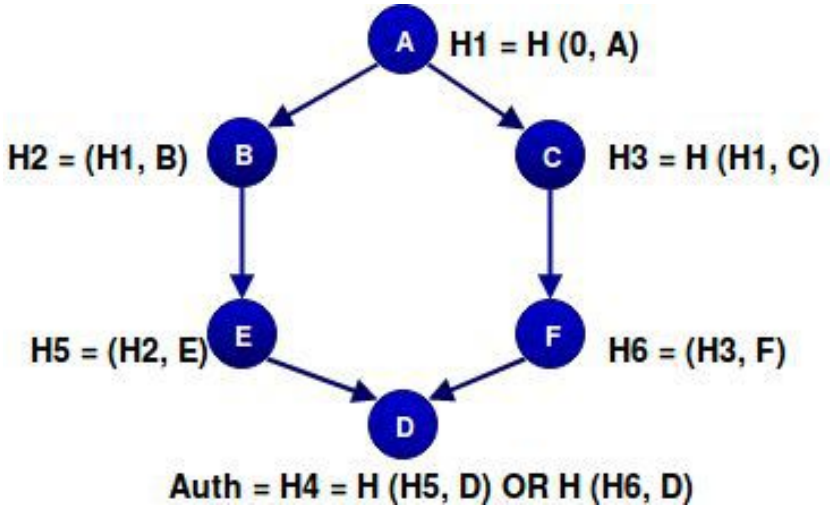


Control - Flow Graph (CFG)

Control flow attestation

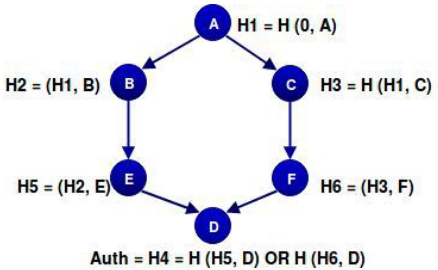
- Want a complete attestation of the run-time state of the Prover
- A single hash value that represents the entire control flow of the Prover's state (for fast and efficient evaluation)

Cumulative Hash Value: $H_i = H(H_{i-1}, N)$
H_{i-1} -- previous hash result
N -- instruction block (node) just executed



Control flow attestation - Challenge

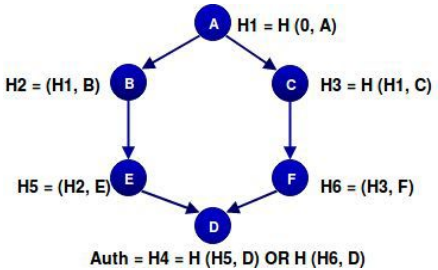
Different loop paths and loop iterations lead to many valid hash values



Control flow attestation - Challenge

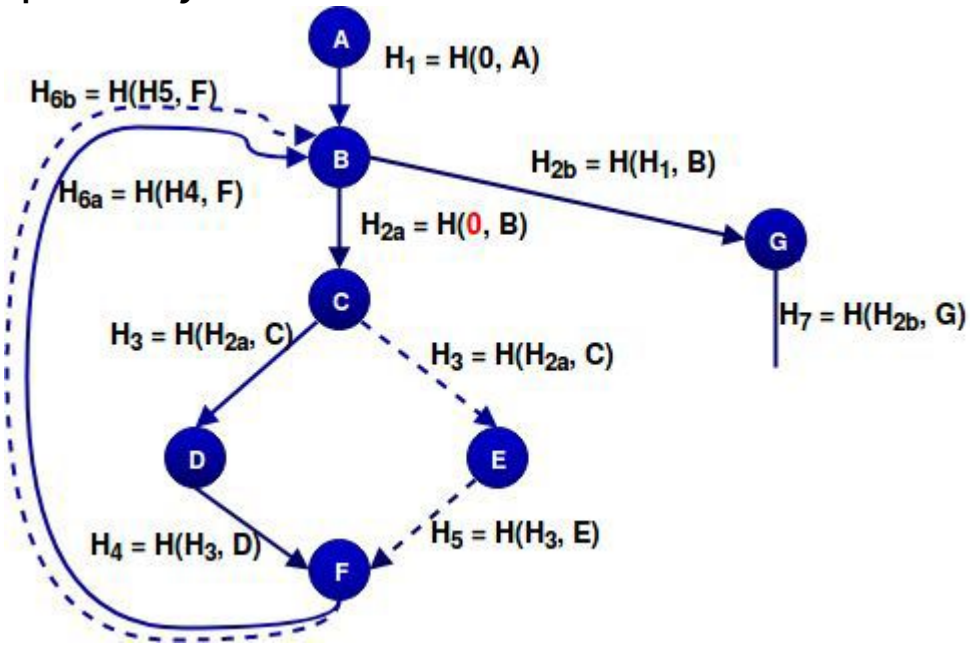
Different loop paths and loop iterations lead to many valid hash values.

What to do: Treat loops as sub-graphs and report their hash values and # of iterations separately



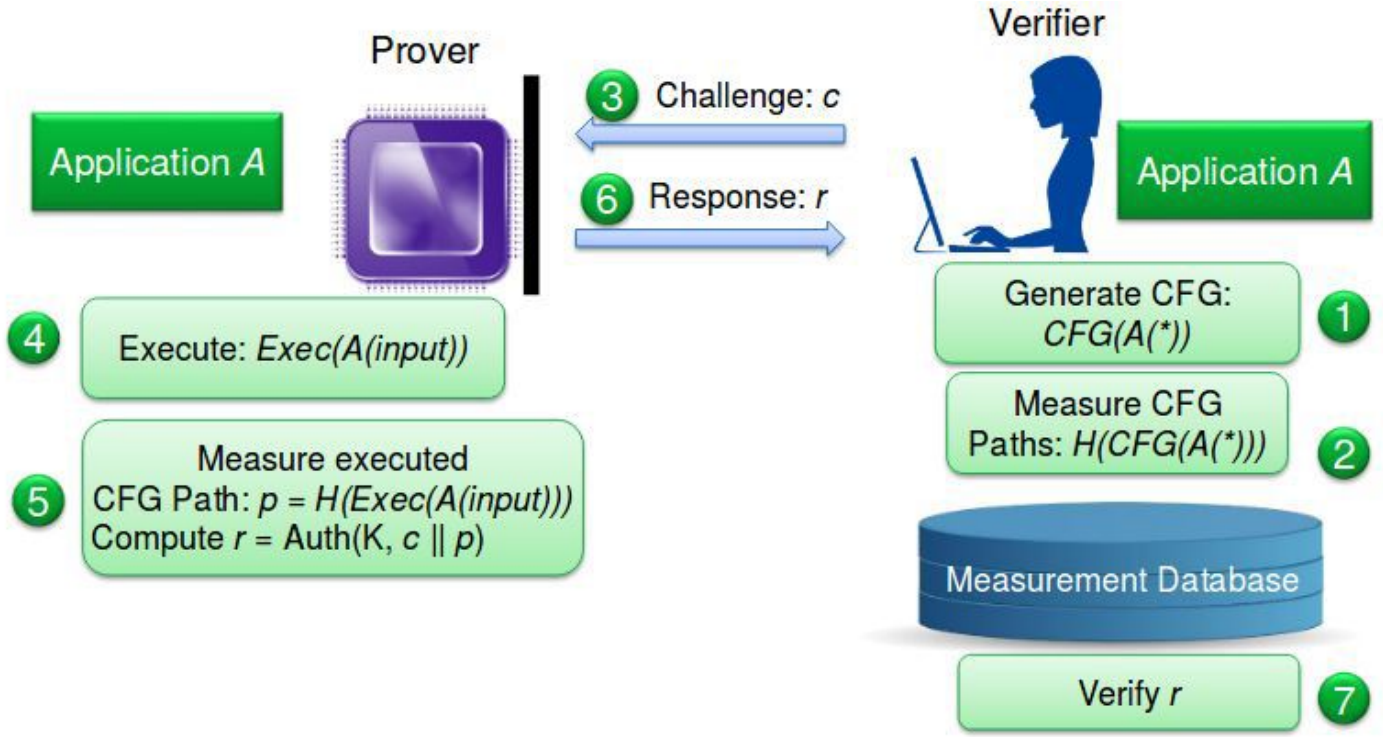
Control flow attestation - Challenge

Different loop paths and loop iterations lead to many valid hash values.
What to do: Treat loops as sub-graphs and report their hash values and # of iterations separately



Auth = H7, <H1, {<H6a, #H6a>, <H6b, #H6b>} >

Control flow attestation



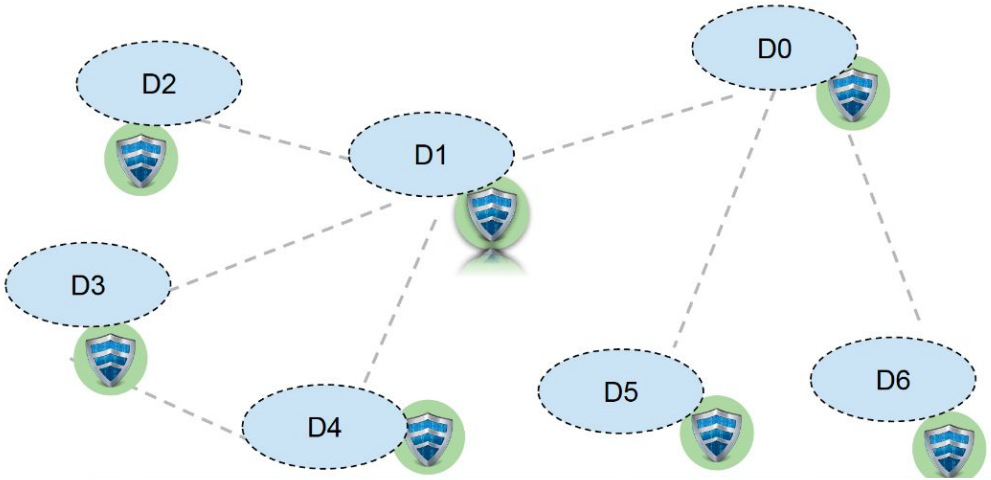
Collective attestation

- Verify the internal state of a large group of devices
- Should be more efficient than attesting each node individually

Collective attestation - one approach

System Model and Assumptions

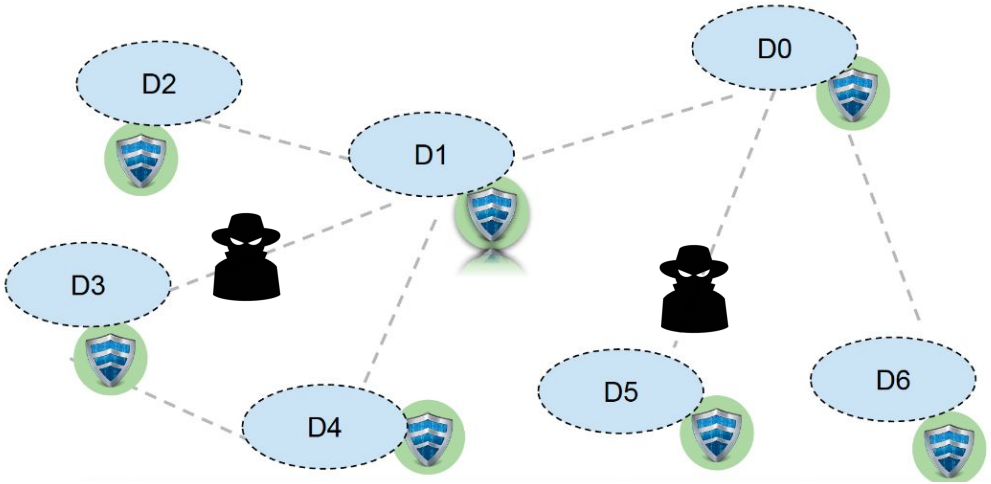
- ALL devices equipped with a trusted component (implementation based on SMART and TrustLite security architectures)
- Devices talk only to their neighbors



Collective attestation - one approach

Adversary can

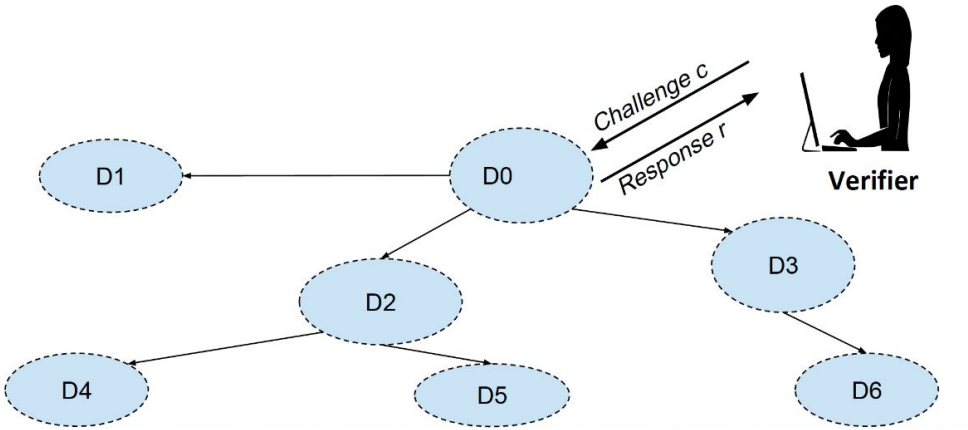
- Control communications (eavesdrop, modifies, insert, deletes messages)
- Software-only attacker (e.g., Malware)
- No physical damage



Collective attestation - one approach

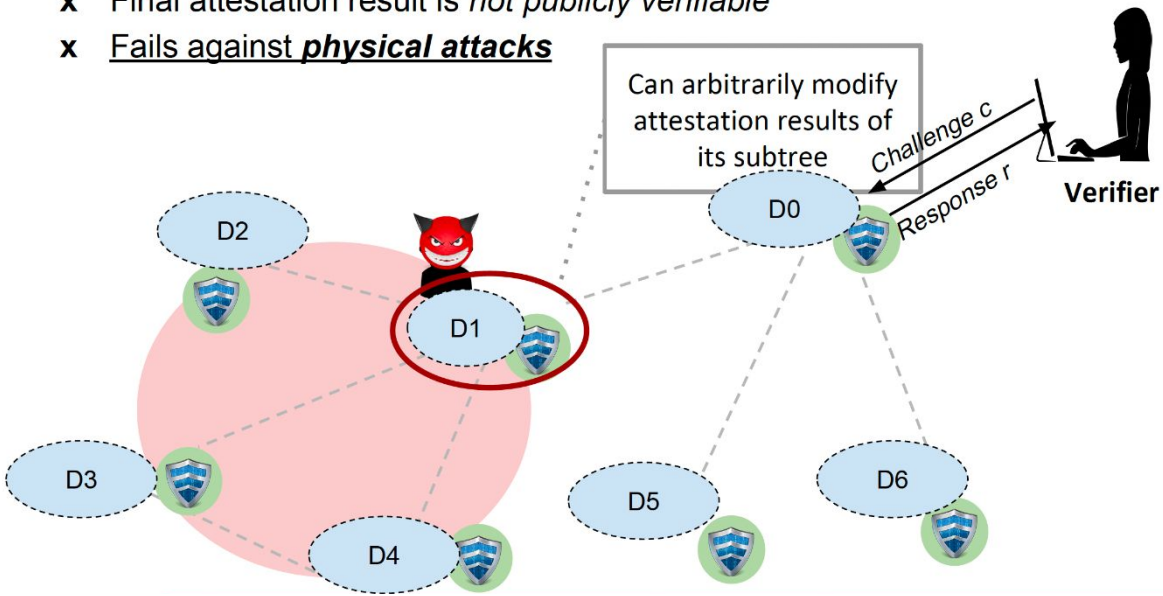
Algorithm logic:

1. Verifier selects random device (D_0) initializes attestation
2. Spanning tree is created rooted at D_0
3. Each devices gets attested by its parent (leaves first)
4. Sub-tree roots accumulate results and reports to their parent
5. D_0 reports overall result to verifier



Collective attestation - one approach

- x Lack of flexibility (ALL devices must participate to attestation)
- x Final attestation result is *not publicly verifiable*
- x Fails against ***physical attacks***



Remote attestation challenges - summary

- There is no single remote attestation protocol that checks all types of the memories in a IoT devices e.g., data variables, CPU registers
- Attestation is an overhead operation
- During the attestation, device stop the usual work
- The results of the attestation are not comprehensive
  - Attestation results are boolean claims (T or F). Do not give insights about the history of the device.
- Design of robust framework for mobile devices which allow them to join or leave the network while preserving the network integrity

Remote attestation - open issues

- Uninterrupted remote attestation e.g., medical device
- Efficient remote attestation for dynamic memory
- Attestation of distributed services
- Attestation of all types of memories
- Attestation of IoT devices in dynamic networks

Reading Material

1. Remote attestation: Link-1, Link-2, Link-3, Link-4, Link-5, Link-6, Link-7, Link-8, Link-9