DefRec: Establishing Physical Function Virtualization to Disrupt Reconnaissance of Power Grids' Cyber-Physical Infrastructures

Hui Lin[1], Jianing Zhuang[1], Yih-Chun Hu[2], Huayu Zhou[1]
[1]University of Nevada, Reno
[2]University of Illinois, Urbana-Champaign

THE DAILY SIGNAL

SEARCH

Ukraine Goes Dark: Russia-Attributed Hackers Take Down Power Grid

Riley Walters / January 13, 2016 / 1 comments

NATIONAL SECURITY

Stuxnet Raises 'Blowback' Risk In Cyberwar

WSJ.com - U.S. regulator says knocking out nine key substations could cause nationwide blackout

Energy sector tops list of US industries under cyber attack, says Homeland Security report

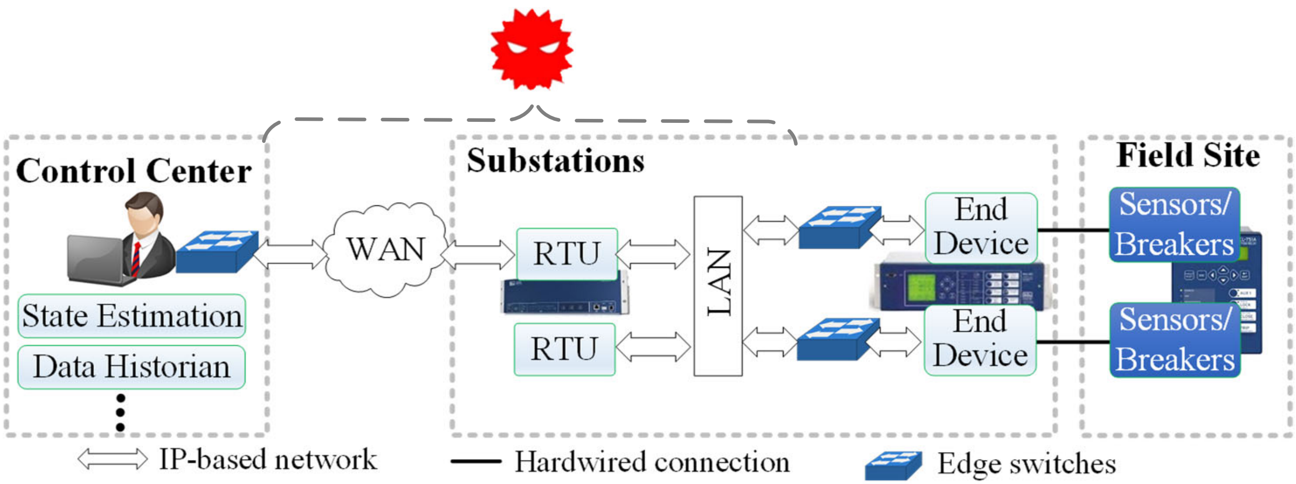
Researchers uncover holes that open power stations to hacking

Hacks could cause power outages and don't need physical access to substations.

From Passive Detection to Preemptive Prevention

- Preemptive approaches disrupting reconnaissance before an adversary starts to inflict physical damage are highly desirable
  - Preventing reconnaissance on a critical set of physical data can cover more attacks, including unknown ones
- Research gap to design practical and efficient anti-reconnaissance approaches
  - Mimicking system behaviors can be easily detected
  - Simulations (used in honeypots) are based on a static specification
    - E.g., inconsistent to proprietary implementation
  - No not model physical processes

Threat Model

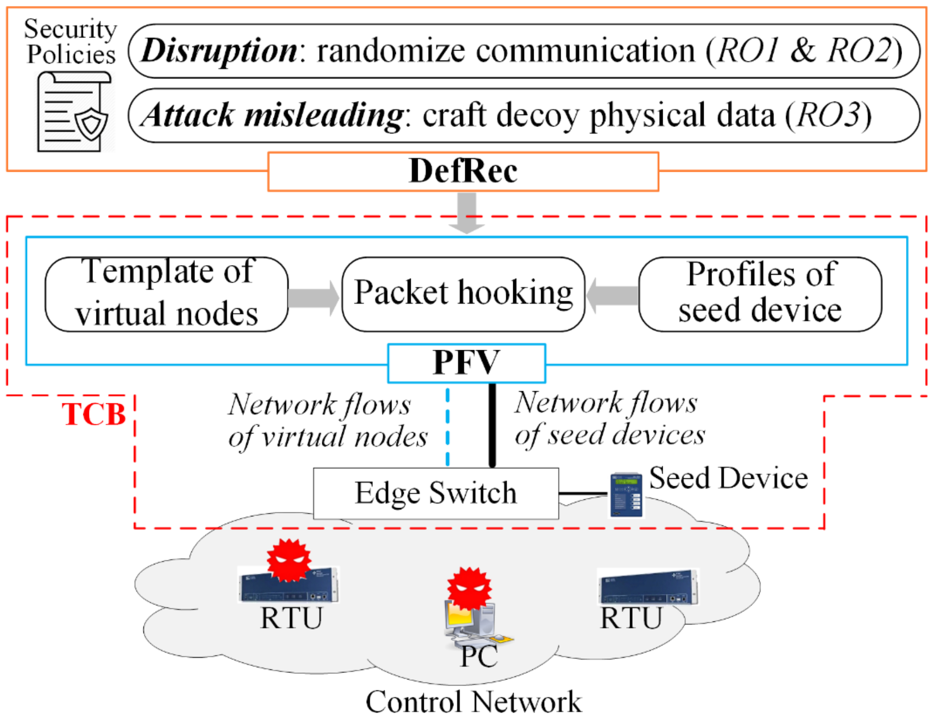


- We assume that adversaries can compromise any computing devices connected to the control network
  - ***Passive attacks*** monitor network traffic to obtain the knowledge of power grids' cyber-physical infrastructures
  - ***Proactive attacks*** achieve the same goal by using probing messages
  - ***Active attacks*** manipulate network traffic, including dropping, delaying, compromising existing network packets, or injecting new packets
- ***Passive*** and ***proactive*** attacks are common techniques used in reconnaissance, while ***active*** attacks are used to issue attack-concept operations and cause physical damage

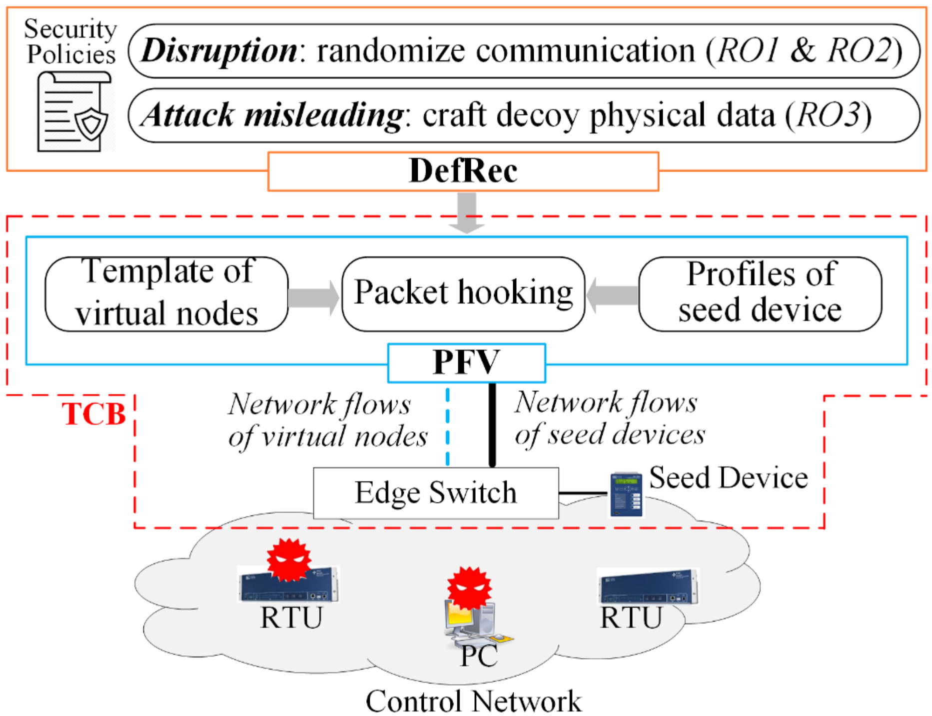
Design Objective

- Disrupt and mislead attackers' reconnaissance based on ***passive*** and ***proactive*** attacks, such that their ***active*** attacks become ineffective
  - RO1 & RO2: significantly delay passive and proactive attacks for obtaining the knowledge of the control network
  - RO3: leverage intelligently crafted decoy data to mislead adversaries into designing ineffective attacks

Design Overview of DefRec based on PFV



Design Overview of DefRec based on PFV

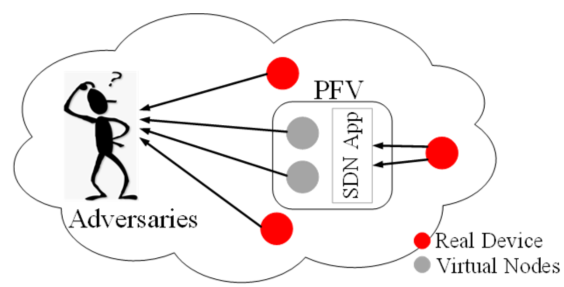


PFV (physical function virtualization): construct virtual nodes that follow the actual implementation of real devices

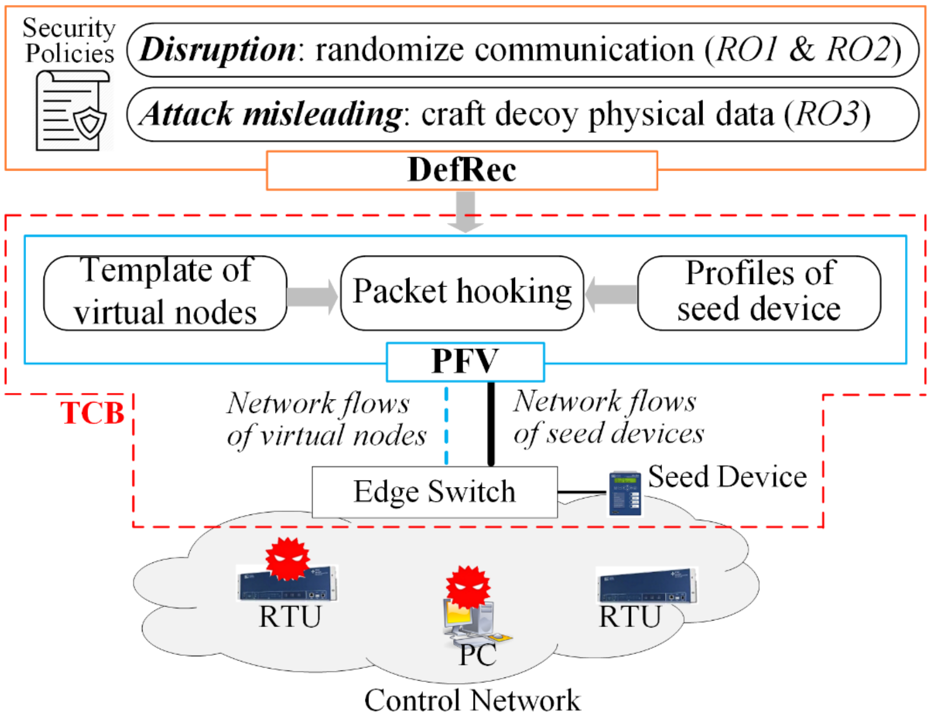
- Complementary to existing security approaches

Trusted computing base (TCB):

- Network controller application
- Edge switches
- A few end devices (used as seed devices)
- Communication channels connecting them



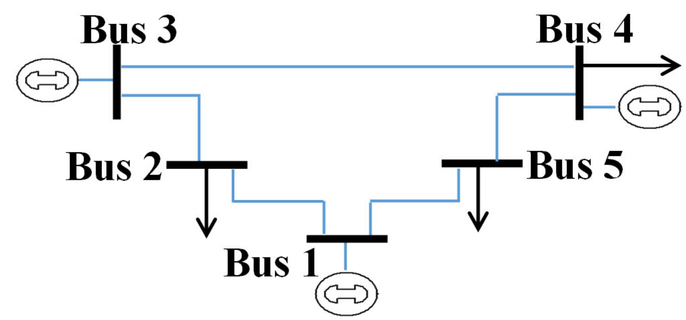
Design Overview of DefRec based on PFV



Trusted computing base (TCB):

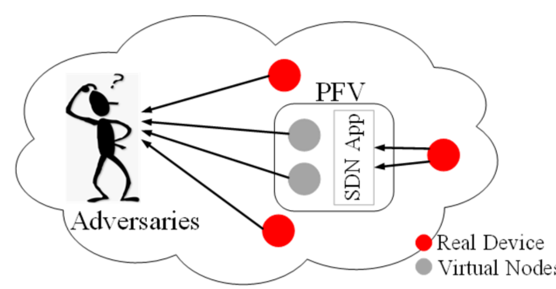
- Network controller application
- Edge switches
- A few end devices (used as seed devices)
- Communication channels connecting them

DefRec: specify security policies to disrupt reconnaissance

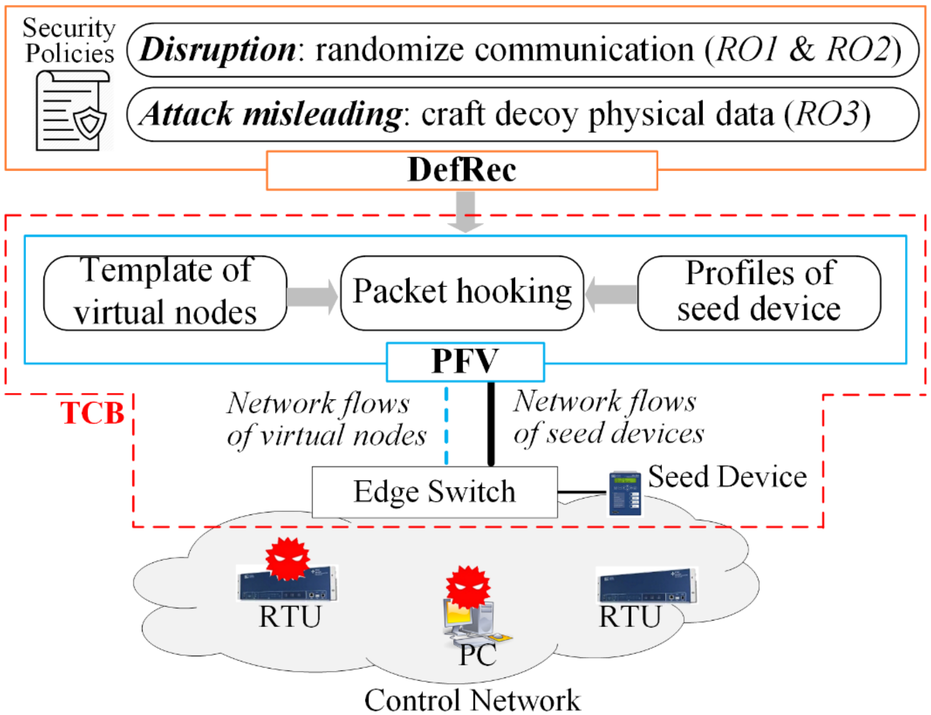


PFV (physical function virtualization): construct virtual nodes that follow the actual implementation of real devices

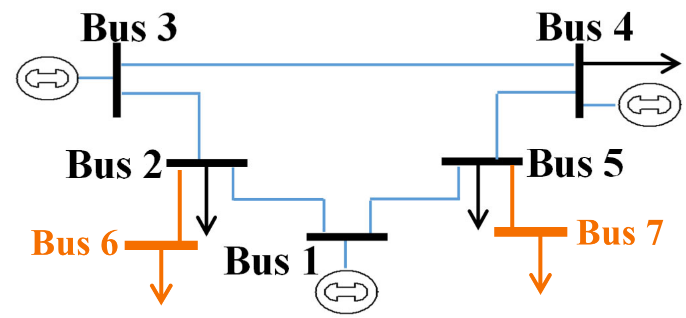
- Complementary to existing security approaches



Design Overview of DefRec based on PFV



DefRec: specify security policies to disrupt reconnaissance

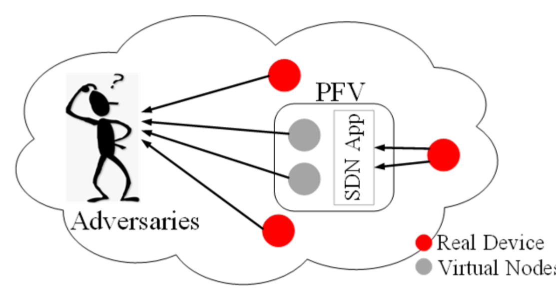


PFV (physical function virtualization): construct virtual nodes that follow the actual implementation of real devices

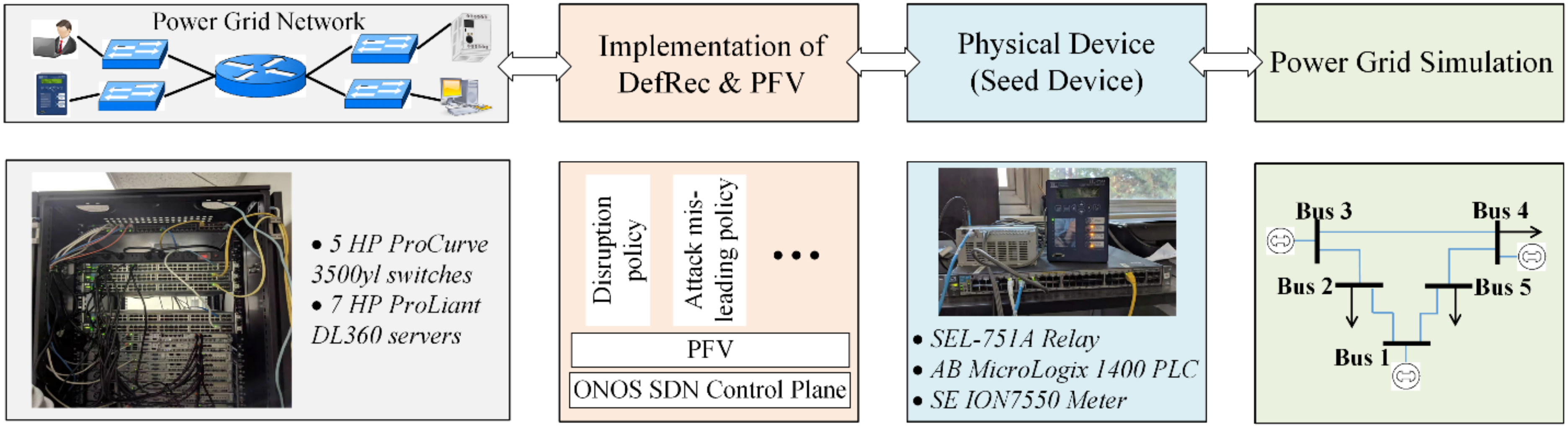
- Complementary to existing security approaches

Trusted computing base (TCB):

- Network controller application
- Edge switches
- A few end devices (used as seed devices)
- Communication channels connecting them



Implementation



- **Communication networks**
- **Implementation of PFV & DefRec**
- **Physical device**
- **Power grid simulation**

Implementation – Communication Network

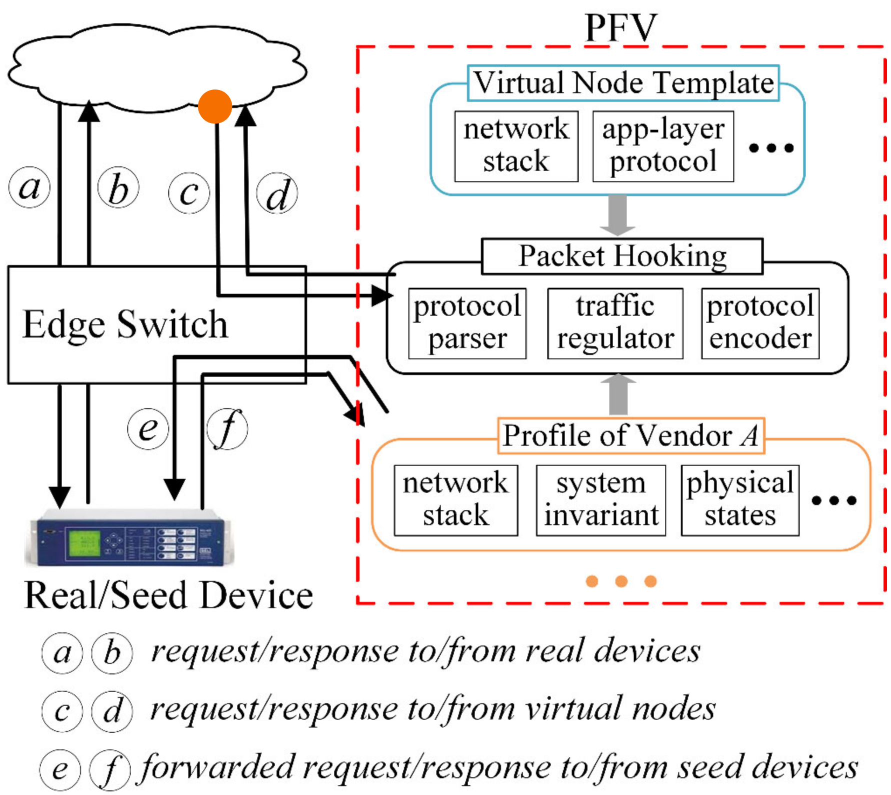
- Follow implementation presented in a NSDI paper [1]
  - Obtained the logical topology of six different communication networks from TopologyZoo dataset
  - Implemented each network in five HP SDN-compatible switches
  - In each switch, we grouped physical ports into VLANs (virtual local area network), each of which represents a logical switch; connect VLANs by Ethernet cables
  - Built Docker instances in seven HP servers as end hosts
    - Need to enhance each server with Ethernet ports
  - Implemented DNP3 master and slaves based on opendnp3 library

- Alternative approach: use cloud infrastructure, e.g., NSF Geni testbed
  - Need to configure virtual switches manually
  - The number of hardware switches are very limited

[1] W. Zhou et al., “Enforcing customizable consistency properties in software-defined networks,” in 12th USENIX NSDI, 2015.

Implementation – PFV

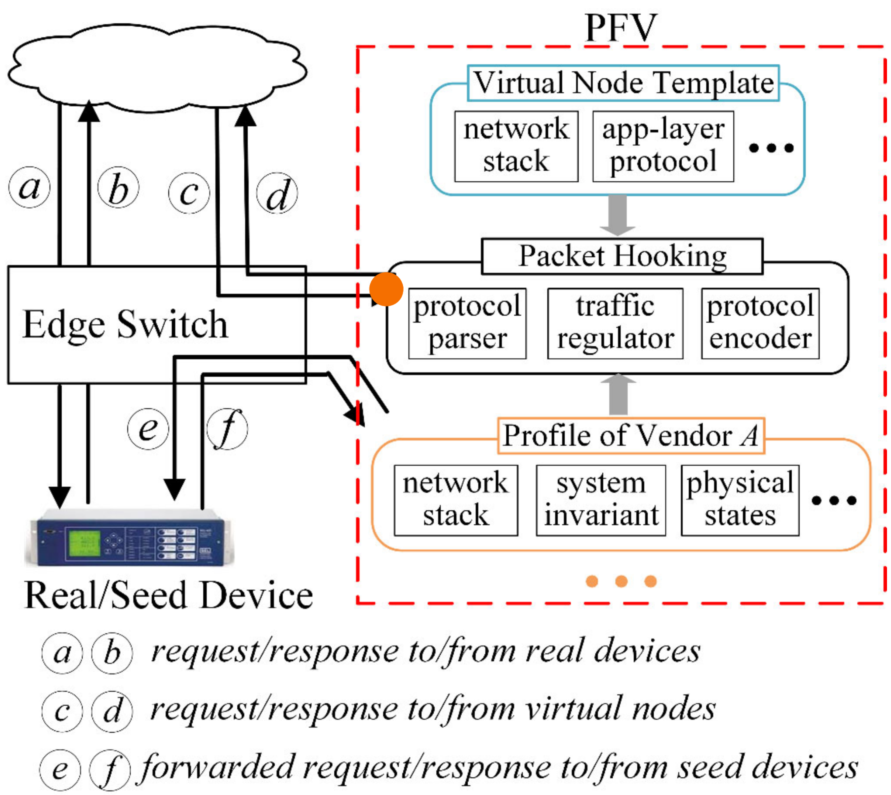
- PFV: use interaction of real devices to build virtual nodes
  - Virtual node template
  - Profile of seed devices
  - Packet hooking component



- Virtual node template
  - Static configuration of target network
- Profile of physical devices
  - Dynamic behavior at network-layer
- Packet hooking component
  - Construct the outbound packets of virtual nodes
  - Follow the probabilistic behavior of real devices

Implementation – PFV

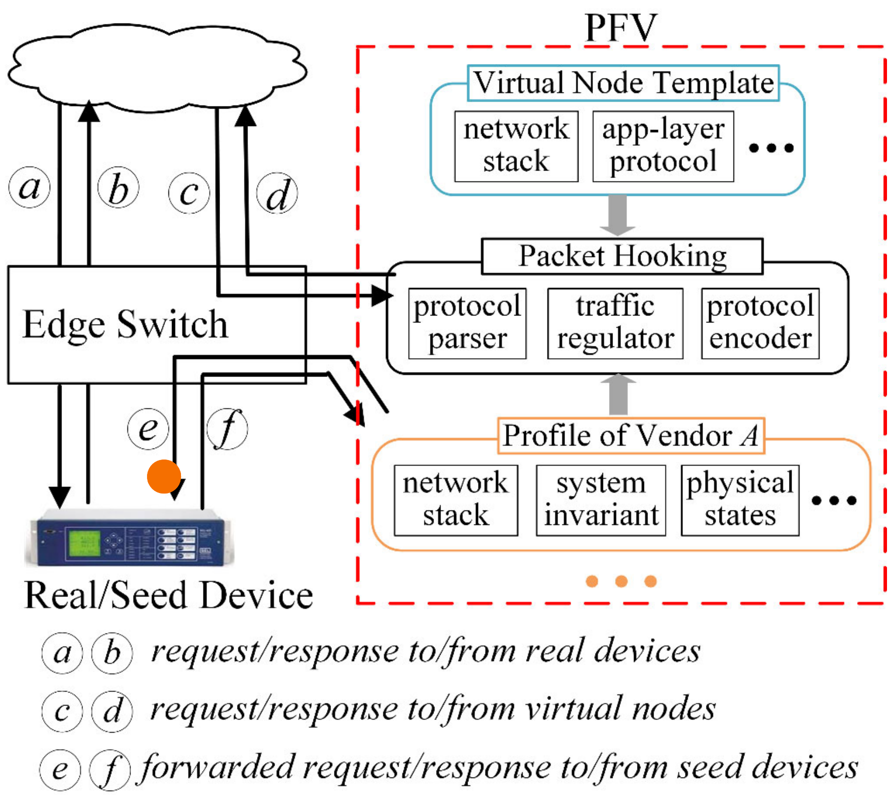
- PFV: use interaction of real devices to build virtual nodes
  - Virtual node template
  - Profile of seed devices
  - Packet hooking component



- Virtual node template
  - Static configuration of target network
- Profile of physical devices
  - Dynamic behavior at network-layer
- Packet hooking component
  - Construct the outbound packets of virtual nodes
  - Follow the probabilistic behavior of real devices

Implementation – PFV

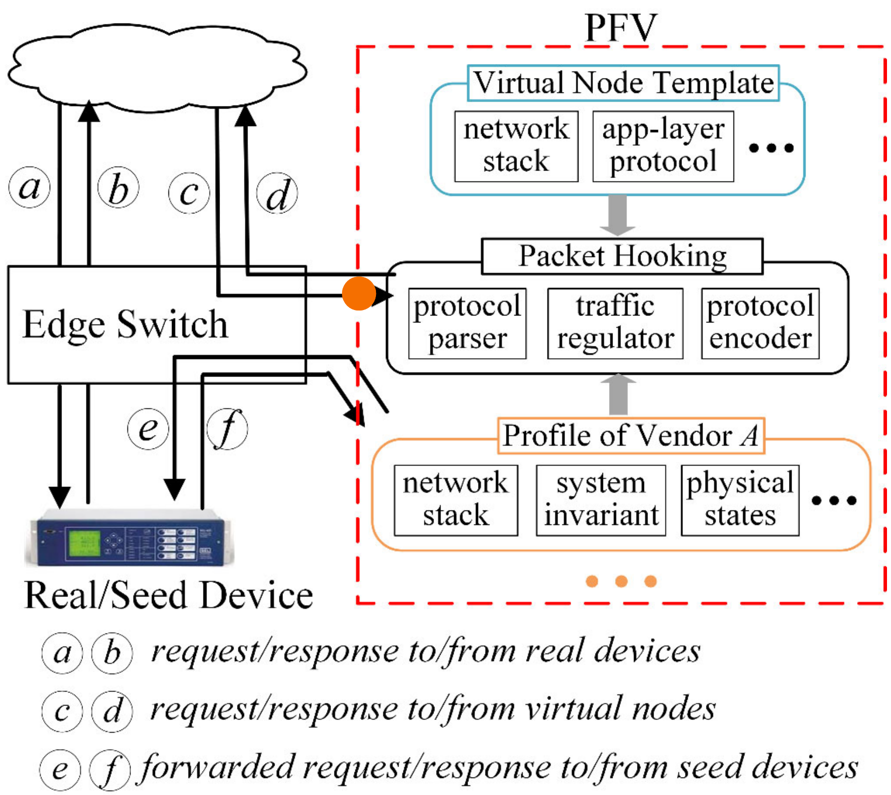
- PFV: use interaction of real devices to build virtual nodes
  - Virtual node template
  - Profile of seed devices
  - Packet hooking component



- Virtual node template
  - Static configuration of target network
- Profile of physical devices
  - Dynamic behavior at network-layer
- Packet hooking component
  - Construct the outbound packets of virtual nodes
  - Follow the probabilistic behavior of real devices

Implementation – PFV

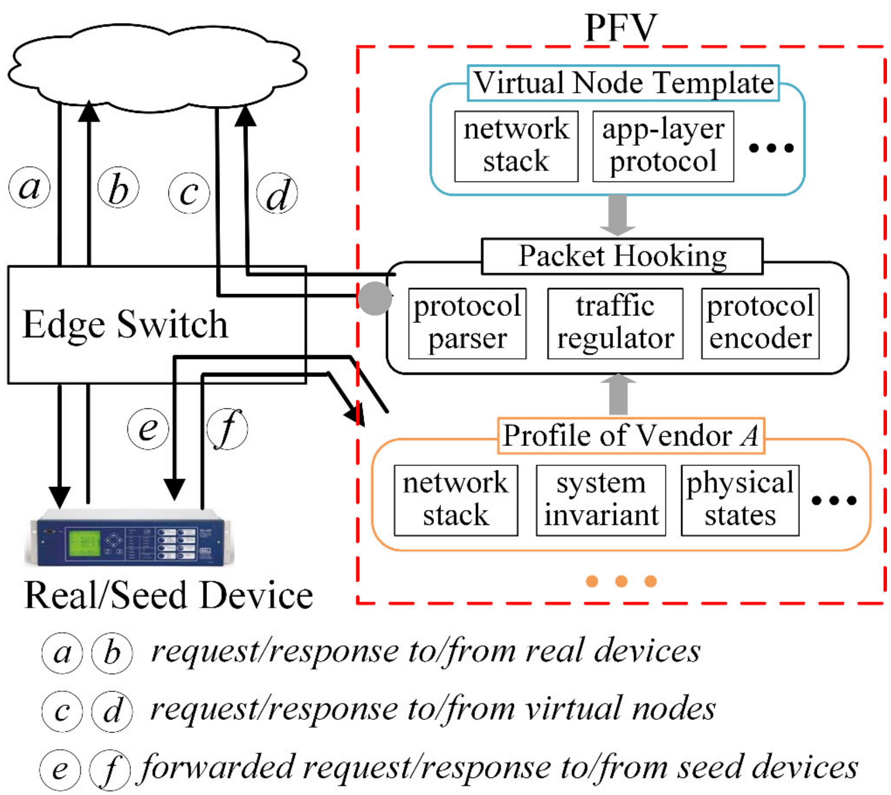
- PFV: use interaction of real devices to build virtual nodes
  - Virtual node template
  - Profile of seed devices
  - Packet hooking component



- Virtual node template
  - Static configuration of target network
- Profile of physical devices
  - Dynamic behavior at network-layer
- Packet hooking component
  - Construct the outbound packets of virtual nodes
  - Follow the probabilistic behavior of real devices

Implementation – PFV

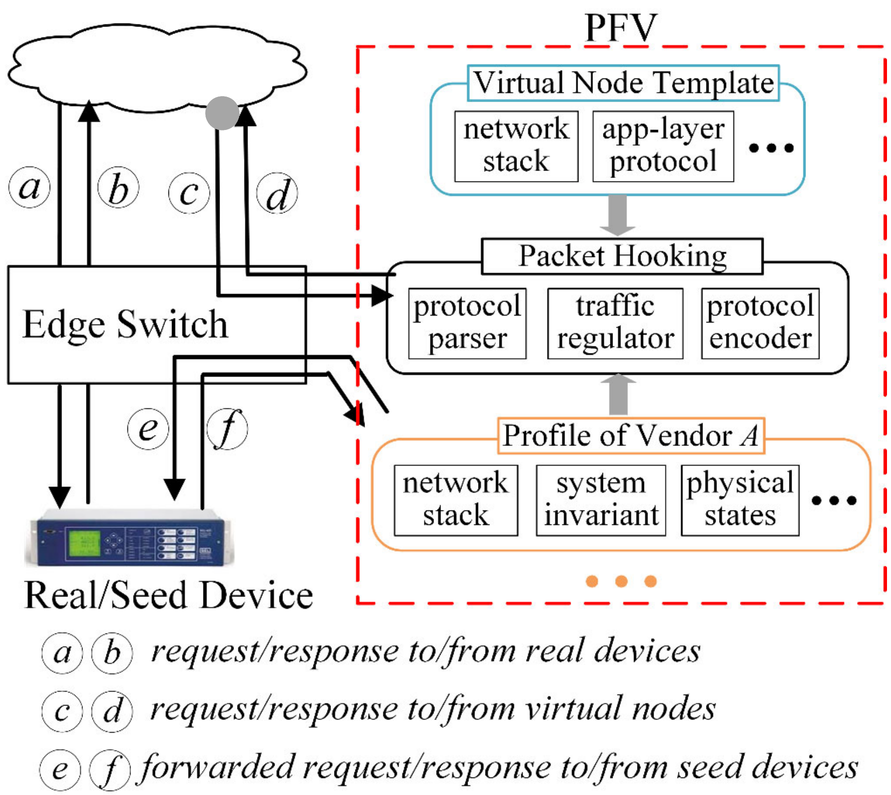
- PFV: use interaction of real devices to build virtual nodes
  - Virtual node template
  - Profile of seed devices
  - Packet hooking component



- Virtual node template
  - Static configuration of target network
- Profile of physical devices
  - Dynamic behavior at network-layer
- Packet hooking component
  - Construct the outbound packets of virtual nodes
  - Follow the probabilistic behavior of real devices

Implementation – PFV

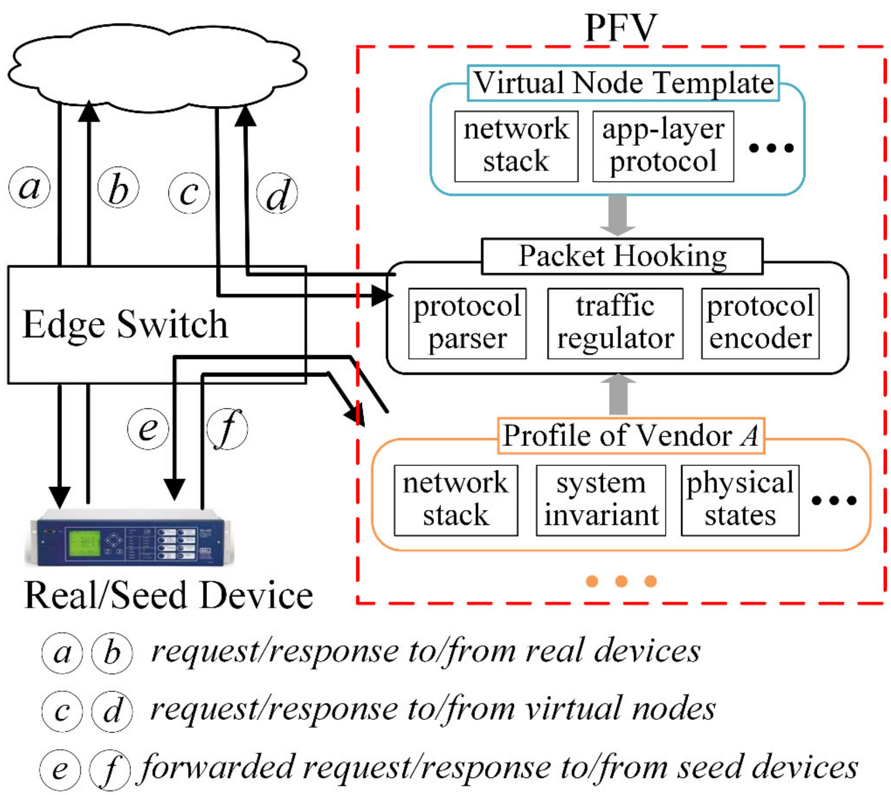
- PFV: use interaction of real devices to build virtual nodes
  - Virtual node template
  - Profile of seed devices
  - Packet hooking component



- Virtual node template
  - Static configuration of target network
- Profile of physical devices
  - Dynamic behavior at network-layer
- Packet hooking component
  - Construct the outbound packets of virtual nodes
  - Follow the probabilistic behavior of real devices

Implementation – PFV

- PFV: use interaction of real devices to build virtual nodes
  - Virtual node template
  - Profile of seed devices
  - Packet hooking component



- Implemented based on SDN (software-defined networking)
  - Follow implementation found in both security and network communities
  - ONOS, open source network operating system used in commercial networks
  - Implemented an encoder/decoder of DNP3 in ONOS core services
  - Implemented software modules loaded by ONOS core services

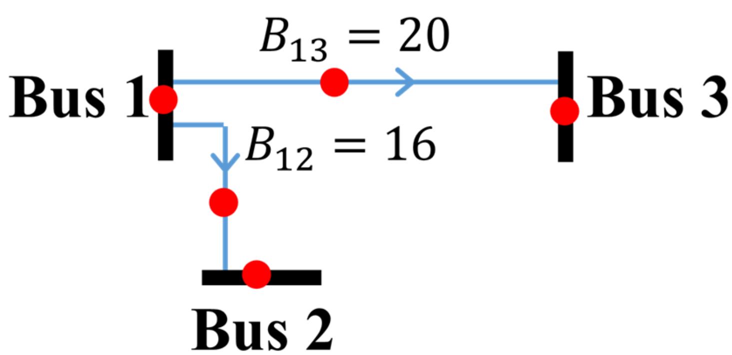
Attack Misleading Policy for Physical Infrastructure

- RO3: craft decoy data as the application-layer payload of network packets from virtual nodes
  - Mislead adversaries into designing ineffective attacks
  - Satisfy physical model of power grids

Attack Misleading Policy for Physical Infrastructure

- RO3: craft decoy data as the application-layer payload of network packets from virtual nodes
  - Mislead adversaries into designing ineffective attacks
  - Satisfy physical model of power grids
- We use the theoretical model of false data injection attack (FDIAs) as a case study

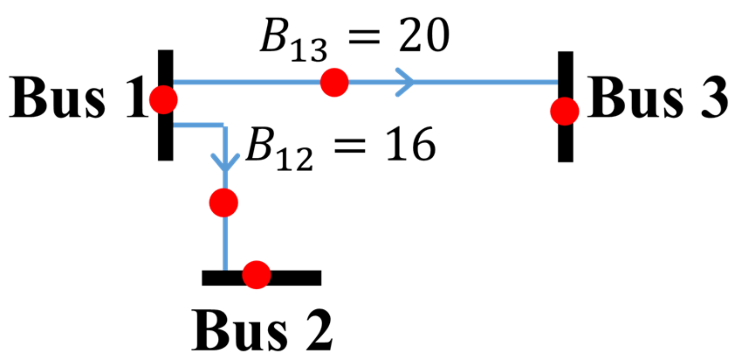
Attack Misleading Policy for Physical Infrastructure



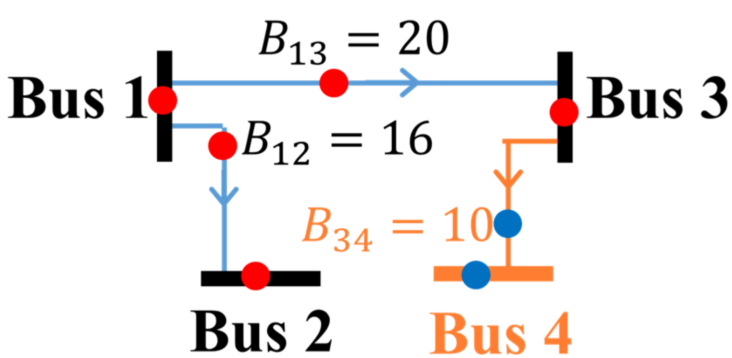
An example power grid

- RO3: craft decoy data as the application-layer payload of network packets from virtual nodes
  - Mislead adversaries into designing ineffective attacks
  - Satisfy physical model of power grids
- We use the theoretical model of false data injection attack (FDIAs) as a case study
  - With accurate knowledge of power grids' topology, ***active*** attacks can compromise measurements without raising alerts in state estimation
    - Measurement errors are less than a detection threshold

Attack Misleading Policy for Physical Infrastructure



An example power grid



The power grid with decoy data observed by adversaries

- RO3: craft decoy data as the application-layer payload of network packets from virtual nodes
  - Mislead adversaries into designing ineffective attacks
  - Satisfy physical model of power grids
- We use the theoretical model of false data injection attack (FDIAs) as a case study
  - With accurate knowledge of power grids' topology, ***active*** attacks can compromise measurements without raising alerts in state estimation
    - Measurement errors are less than a detection threshold
  - With misleading knowledge of power grids' topology, ***active*** attacks raise alerts in state estimation
    - Measurement errors are 5,000 times of the detection threshold

Implementation – DefRec

$$H = \begin{bmatrix} H_1 \\ \hline H_2 \end{bmatrix} \begin{bmatrix} 16 & -16 & 0 \\ 20 & 0 & -20 \\ 36 & -16 & -20 \\ -16 & 16 & 0 \\ \hline -20 & 0 & 20 \end{bmatrix} \begin{matrix} \text{line } (1,2) \\ \text{line } (1,3) \\ \text{bus } 1 \\ \text{bus } 2 \\ \text{bus } 3 \end{matrix}$$

$$H' = \begin{bmatrix} H_1' \\ \hline H_2' \end{bmatrix} = \begin{bmatrix} H_1 & 0 \\ \hline H_{21}' & H_{22}' \\ \hline H_{23}' & H_{24}' \end{bmatrix}$$

$$= \begin{bmatrix} 16 & -16 & 0 & 0 \\ 20 & 0 & -20 & 0 \\ 36 & -16 & -20 & 0 \\ -16 & 16 & 0 & 0 \\ \hline -20 & 0 & 30 & -10 \\ \hline 0 & 0 & -10 & 10 \\ 0 & 0 & 10 & -10 \end{bmatrix} \begin{matrix} \text{line } (1,2) \\ \text{line } (1,3) \\ \text{bus } 1 \\ \text{bus } 2 \\ \text{bus } 3 \\ \text{bus } 4 \\ \text{line } (3,4) \end{matrix}$$

- Followed the theoretical model presented in the first paper about FDIA [2] to "prove" the effectiveness of decoy data
  - The proof follows common procedure in literatures from IEEE Transactions on Smart Grid

[2] Y. Liu et al., "False data injection attacks against state estimation in electric power grids," in 17th CCS, 2010.

Implementation – DefRec

$$H = \begin{bmatrix} H_1 \\ \hdashline H_2 \end{bmatrix} \begin{bmatrix} 16 & -16 & 0 \\ 20 & 0 & -20 \\ 36 & -16 & -20 \\ -16 & 16 & 0 \\ \hdashline -20 & 0 & 20 \end{bmatrix} \begin{matrix} \text{line (1,2)} \\ \text{line (1,3)} \\ \text{bus 1} \\ \text{bus 2} \\ \text{bus 3} \end{matrix}$$

$$H' = \begin{bmatrix} H'_1 \\ \hdashline H'_2 \end{bmatrix} = \begin{bmatrix} H_1 & 0 \\ \hdashline H'_{21} & H'_{22} \\ \hdashline H'_{23} & H'_{24} \end{bmatrix}$$

$$= \begin{bmatrix} 16 & -16 & 0 & 0 \\ 20 & 0 & -20 & 0 \\ 36 & -16 & -20 & 0 \\ -16 & 16 & 0 & 0 \\ \hdashline -20 & 0 & 30 & -10 \\ \hdashline 0 & 0 & -10 & 10 \\ 0 & 0 & 10 & -10 \end{bmatrix} \begin{matrix} \text{line (1,2)} \\ \text{line (1,3)} \\ \text{bus 1} \\ \text{bus 2} \\ \text{bus 3} \\ \text{bus 4} \\ \text{line (3,4)} \end{matrix}$$

- Followed the theoretical model presented in the first paper about FDIA [2] to "prove" the effectiveness of decoy data
  - The proof follows common procedure in literatures from IEEE Transactions on Smart Grid
- Implemented in MATPOWER
  - The state-of-the-art power system analysis tools
  - Commonly used in both power engineering and security communities

[2] Y. Liu et al., "False data injection attacks against state estimation in electric power grids," in 17th CCS, 2010.

Implementation – Physical Devices

- Selecting devices that passed the conformance test of the DNP3 protocol
- Schweitzer Engineering Laboratories (SEL) 751A relay
  - Used in [3] to study fingerprinting methods for physical devices in power grids
- Allen Bradley (AB) MicroLogix 1400 PLC
  - Lower model of 17xx series used in [4]
  - Support wide control operations used in different cyber-physical systems
- Schneider Electric (SE) ION7550 power meters
  - Comparatively simple functionality
  - Purchased a refurbished device

[3] D. Formby et al., "Who's in control of your control system? device fingerprinting for cyber-physical systems," in 2016 NDSS.
[4] L. A. Garcia et al., "Hey, my malware knows physics! attacking PLCs with physical model aware rootkit," in 2017 NDSS.

Implementation – Power Grid Simulation

- Simulated six power grids, whose configurations are included in MATPOWER
  - The latter two systems represent the biggest two areas of Polish 400-, 220-, and 110-kV national transmission networks
  - Varied operational conditions according to real operational data

Power Grid Simulation	***Network***
IEEE 24-bus	DataX
IEEE 30-bus	Abilene
RTS96 73-bus	Hurricane
IEEE 118-bus	Chinanet
Poland 406-bus	Cesnet
Poland 1153-bus	Forthnet

- New cases
  - More cases are included in MATPOWER after paper submission
  - E.g., an 10,000-bus case to represent U.S. national grid

Evaluation

- Security evaluation
  - Effectiveness of PFV
  - Effectiveness of attack-disruption policy
  - Effectiveness of attack-misleading policy

- Performance evaluation

Effectiveness of PFV

Objective: evaluate whether virtual nodes can follow the runtime behavior of real devices

Original Plan

- We applied fingerprinting methods proposed for CPSs [3] on both real physical devices and virtual nodes
  - Use the time that IEDs execute commands as a system invariant
- We compare the probability density functions (PDFs) of execution time measured for real devices and virtual nodes

Experiment

- **Issue #1**: physical devices support different types of control operation
- **Solution**: measure two common operations
- **Issue #2**: proprietary implementation of TCP/IP stack
  - Some responses integrate ACK message
- **Solution**: use SDN controller to measure the round-trip time behind the swtich

Effectiveness of Disruption Policy

Objective: estimate how long we can delay passive and proactive attacks for obtaining network configuration

Original Plan

- Info-theoretically estimate the probability that passive and proactive attacks can obtain the network configuration

Experiment

- Issue #1: the results are difficult to be interpreted
  - E.g., some false negative rates are as low as 10^{-10}
- Solution: Use the delay time as evaluation metric
  - Assuming that an attacker can *passively* monitor up to 200 network packets every second
  - Assuming that adversaries can *probe* control networks with a throughput of 10 Gigabytes per second

[5] Y. Liu et al., “False data injection attacks against state estimation in electric power grids,” in 17th CCS, 2010.

Effectiveness of Attack-Misleading Policies

- Redefine false positive/false negative for crafted decoy data
  - FN: FDIAs prepared based on decoy data are successful
    - Measurement errors are less than a detection threshold
  - FP: decoy data are not valid, meaning that the combination of decoy and real data does not follow the physical model of a power grid
- Evaluations are performed based on repeated simulation of FDIAs implemented in MATPOWER
  - 1,000 times for small scale power grids and 200 times for big scale power grids

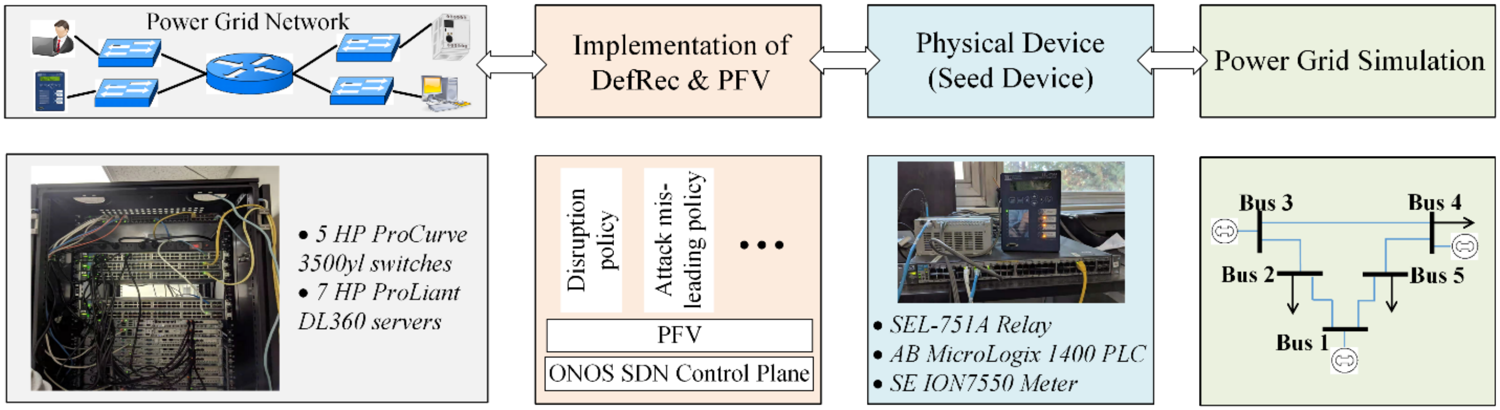
Performance Overhead of Spoofed Network Packets

- **Objective**: measure the impact of spoofed network packets on the round-trip time of real network packets
- Unpublished experiments
  - Evaluate in Mininet, a network emulator in a single desktop
    - Results are affected by the bandwidth of Ethernet card of that desktop
  - Evaluate in NSF Geni testbed
    - Results are affected by limited bandwidth
    - Reserving resources for a large scale communication network (more than 100 switches) is very challenging

Performance Overhead of Crafting Decoy Data

- **Objective**: measure the latency of crafting decoy data
- Unpublished experiments
  - The algorithm to craft decoy data is largely relied on the state estimation, a domain specific analysis method in power grids
  - Scales poorly with the size of power grids
- We adjusted the parameter of the algorithm to speed up
  - E.g., reduce the number of iteration of computation, borrowing experiences developed in our previous projects

Discussions



- The experiment in DefRec is relied on a cyber-physical testbed
  - Communication network relies on hardware SDN-compatible switches
  - Power grids relies on state-of-the-art simulation
- Next step:
  - Upgrade switches to better configure “port-delay”
  - Integrate power grids with a large size
  - Integrate cyber and physical components to construct a hardware-in-the-loop testbed