PrivacyGuard: Extreme SDN Framework for IoT and Mobile Applications Flexible Privacy at the Edge

Mostafa Uddin

Nokia Bell Labs
USA

mostafa.uddin@nokia-bell-labs.com

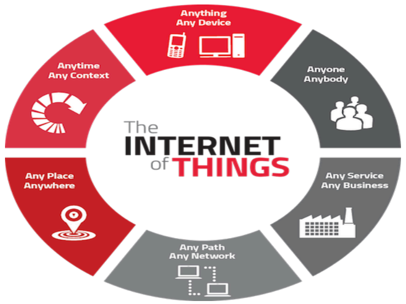
Tamer Nadeem, Santosh Nukavarapu

Virginia Commonwealth University
USA

{tnadeem, nukavarapuskk}@vcu.edu

IEEE PERCOM, Kyoto, Japan, March 11-15, 2019

Rapid growth of mobile data traffic



- Number of smart device users expected to exceed 6 billion by 2020
- IoT connected objects are expected to reach 18 billion by 2022

Mobile devices runs numerous and wide variety of applications

Source: Cisco VNI, 2014
The percentages in parentheses next to the legend refer to traffic share in 2013 and 2018, respectively.

High volume of wireless traffic

Wi-Fi networks are expected to carry almost 60% of smartphones and tablets data traffic by 2019

Growth of Sensitive Apps

- **Sensitive applications communicate sensitive data over internet**

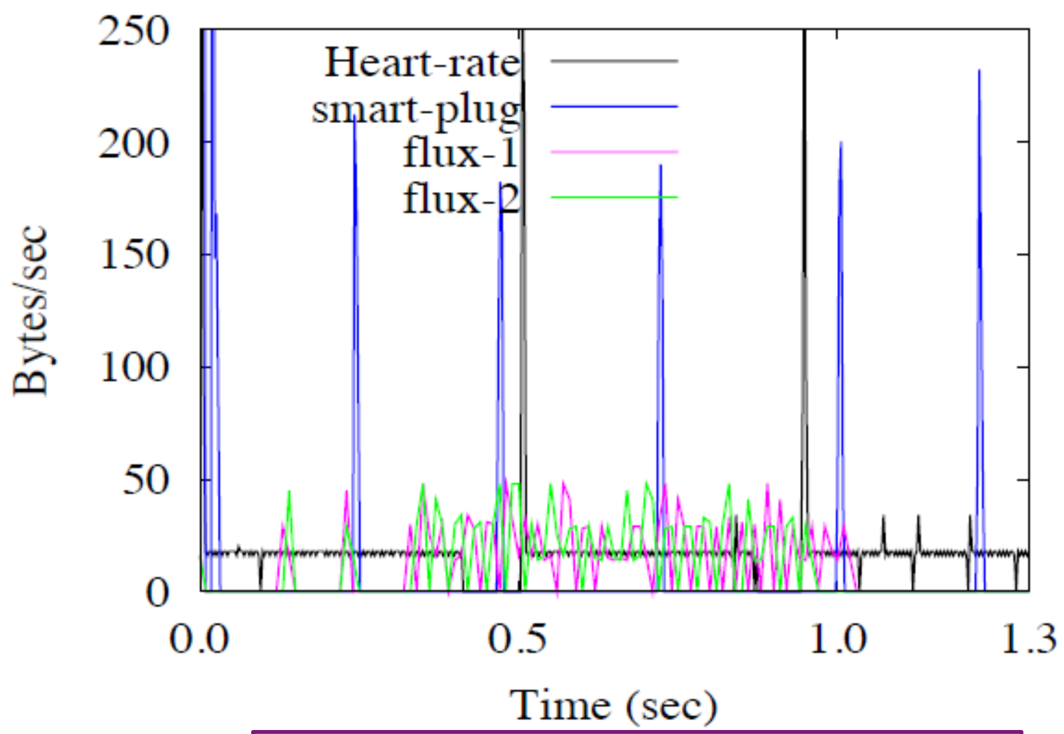
Medical Information:
Blood Pressure Monitoring , Diabetes.

Activity Tracking:
Sleeping Patterns , Exercise Routines.

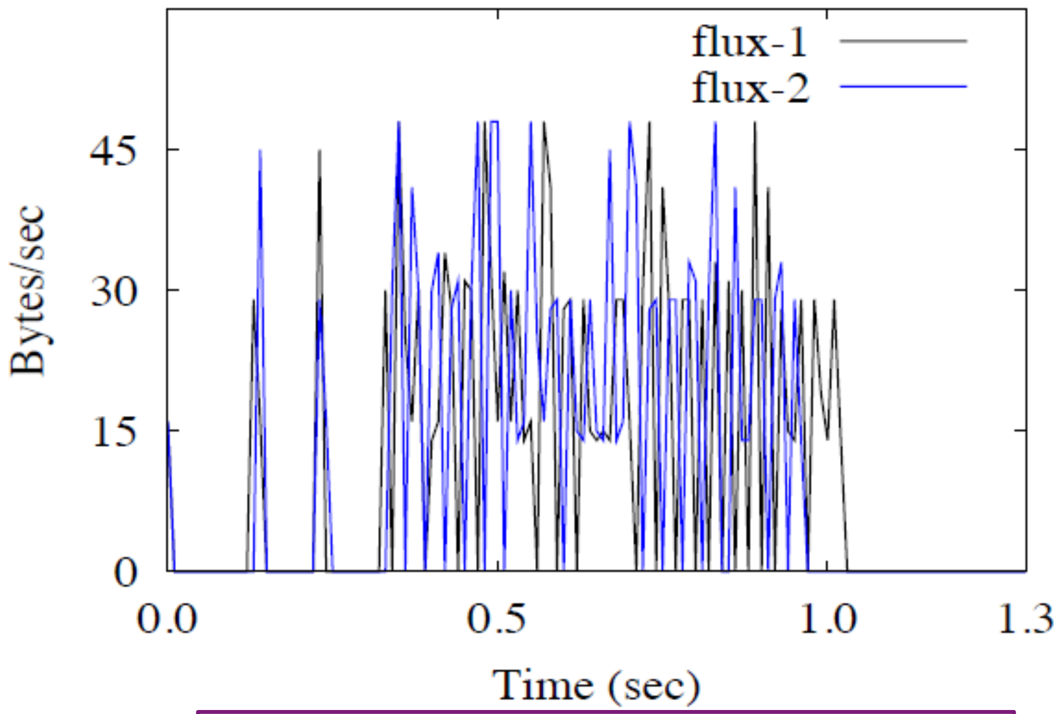
VCU

MuSIClab

Traffic Patterns Of IoT Apps



Traffic Patterns of four IoT devices operating at different times



Zooming into Traffic of the two Flux-lightbulb devices shows high similarity

Most of the IoT mobile apps show unique traffic patterns that are easily distinguishable and consistent over time

VCU

MuSIClab

Problem & Related Work

Privacy Threat Model

IoT Gateway

IoT Devices

BLE

Mobile and IOT Apps

Statistical analysis based on Packet sizes, Inter arrival time, size of first N packets and burst pattern etc.

Passively Capture WLAN traffic

Analyzing WLAN traffic (encrypted or unencrypted)

Side channel Information

1. User Activities
2. Health conditions
3. Track user occupancy

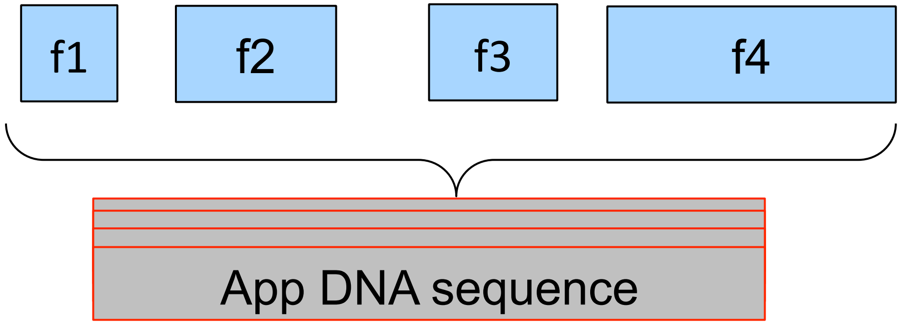
(Threat to individual safety!)

VCU

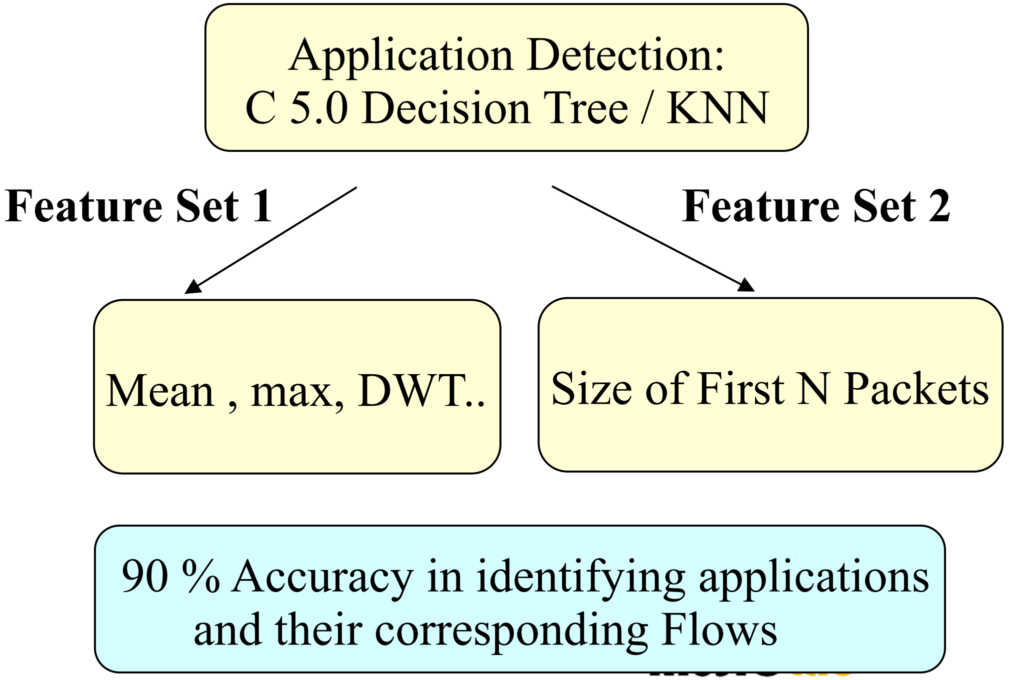
MuSIClab

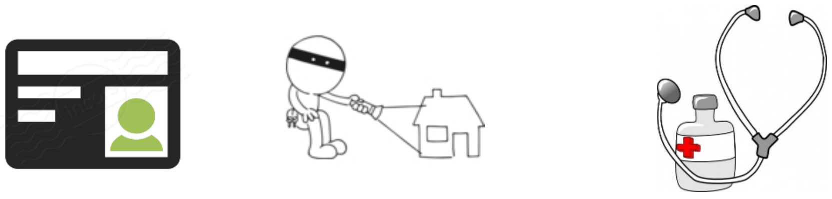
Proof-of-Concept Threat Model

- Two sets of statistics:
  - **Lower-order statistics:** number of packets, number of bytes, protocol, and mean, median, minimum, maximum, and variance of the packet sizes and IPTs,
  - **Higher-order statistic:** Discrete Wavelet Transform (DWT) capturing both the global and the local variations of the time-series data.
- The initial packet-size sequence of an app is unique
  - Application *DNA sequence*
  - Application *genome database*
- Wi-Fi encryption (i.e., 802.11i WPA2)
  - Add a constant number of bytes (16 bytes)
  - Encrypt data part of Wi-Fi frame and not Wi-Fi header



App genome database (build offline)





VCU

Existing Solutions

Infrastructure based solutions

- ❑ Managing network wide devices from network infrastructure
- ❑ Isolate network traffic between sensitive and non-sensitive applications
- ❑ Not well suited for dynamic devices, and do not support client-side solution

Anonymous/Randomization Systems (Virtual MAC interfaces)

- ❑ MAC Layer Management between mobile devices and APs
- ❑ Supporting the multiple virtual interfaces and distributing the traffic over those interfaces
- ❑ Expensive and require device driver modification

Traffic Shaping

- ❑ Traffic Padding, faking superfluous packets and chopping packets
- ❑ Traffic Morphing
- ❑ Efficiency and Overhead varies based on configuration parameters

What is Missing ?

Coarse-grained privacy policies

- ❑ Application-aware or context-aware privacy policy is not possible

User's are not in control of their traffic

- ❑ No flexible and user-friendly tools to meet their requirement
- ❑ Not transparent to the application

Limited work on addressing the privacy inference of side channel attacks

Objectives

Flexible per application privacy preserving Schemes (e.g., traffic shaping)

- ❑ Different applications and even different flows of the same application would have different traffic characteristics.

Programmable privacy preserving policies

- ❑ Support programmable APIs to define and configure different schemes dynamically.

Context aware privacy preserving policies

- ❑ Different application requirements , user objectives , device characteristics and network conditions contexts, require different performance levels of applied privacy schemes.

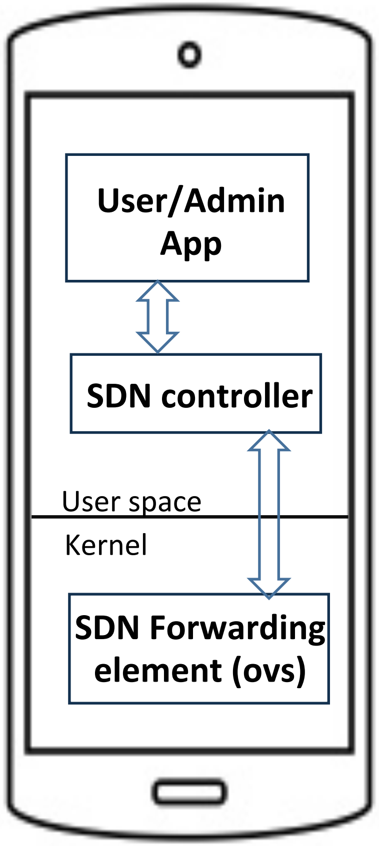
Policies are transparent to application

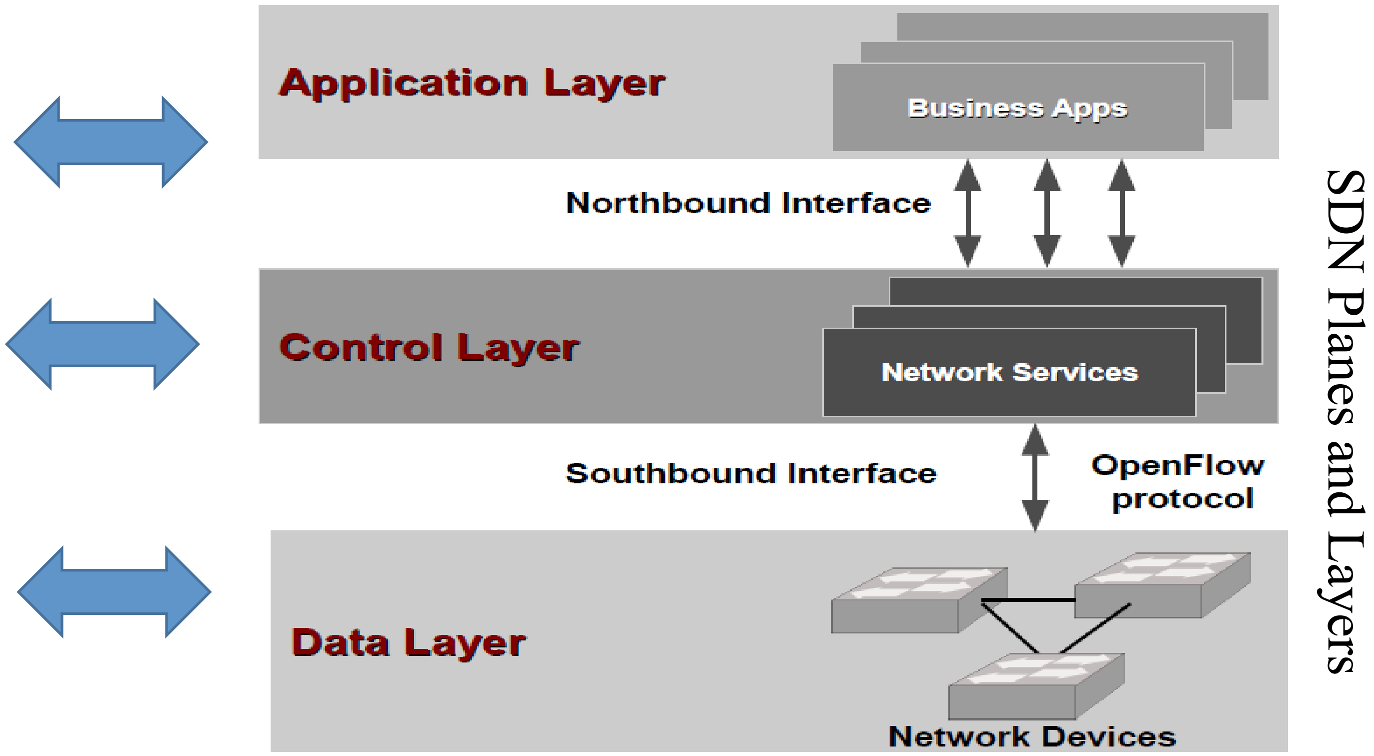
- ❑ Support any application without requiring any modification on either client or server of the application.

Our Solution: PrivacyGuard

- Leverage SDN-based framework on end devices (*Extreme SDN*).

Applying flexible privacy policies using SDN components on an end device.



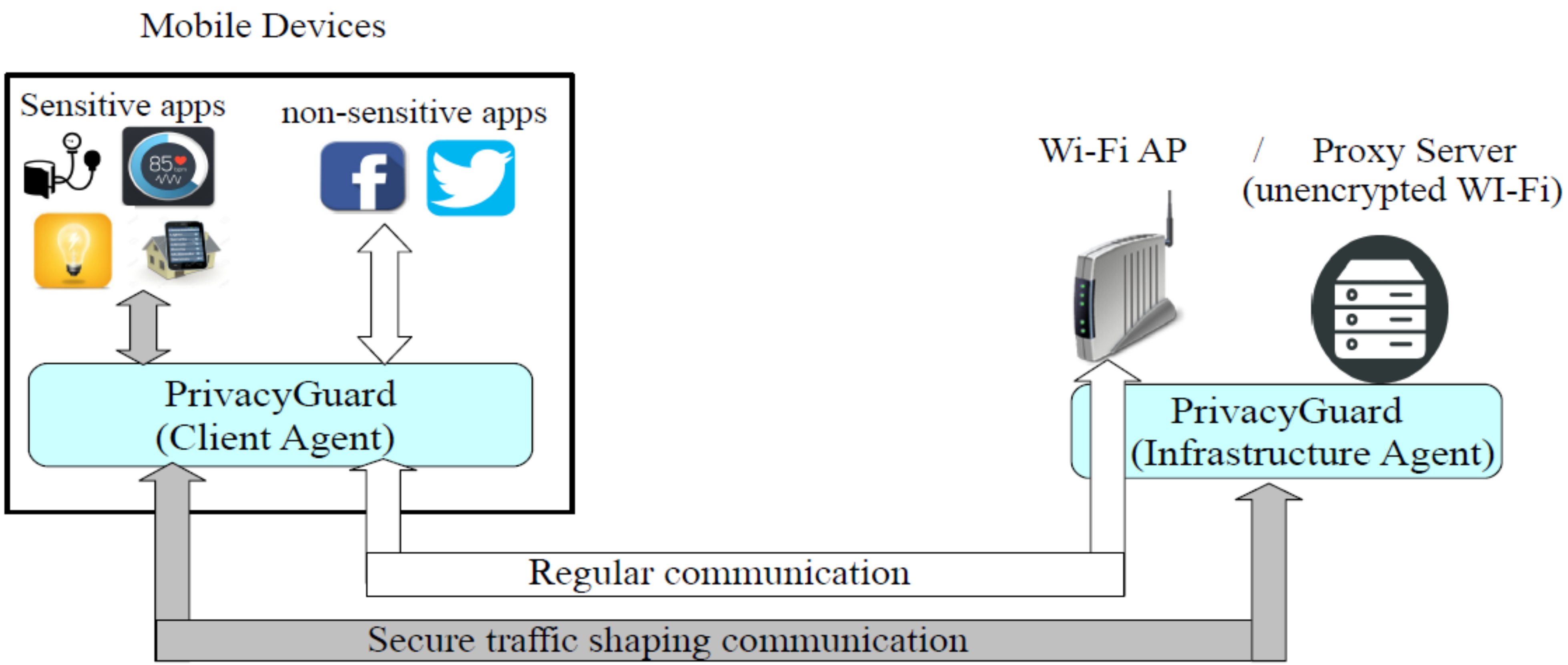


PrivacyGuard: Benefits

- Offloads intrusive or resource-demanding tasks from the network to end devices.
- Fine-grained and intelligent management of privacy-preserving schemes based on real time context awareness.
- Flexible implementation of network privacy policies.
- Offers universal approach to work across network technologies, WiFi and cellular.
- Has no dependency on the internal network support.
- Improves user's privacy with very low overhead.

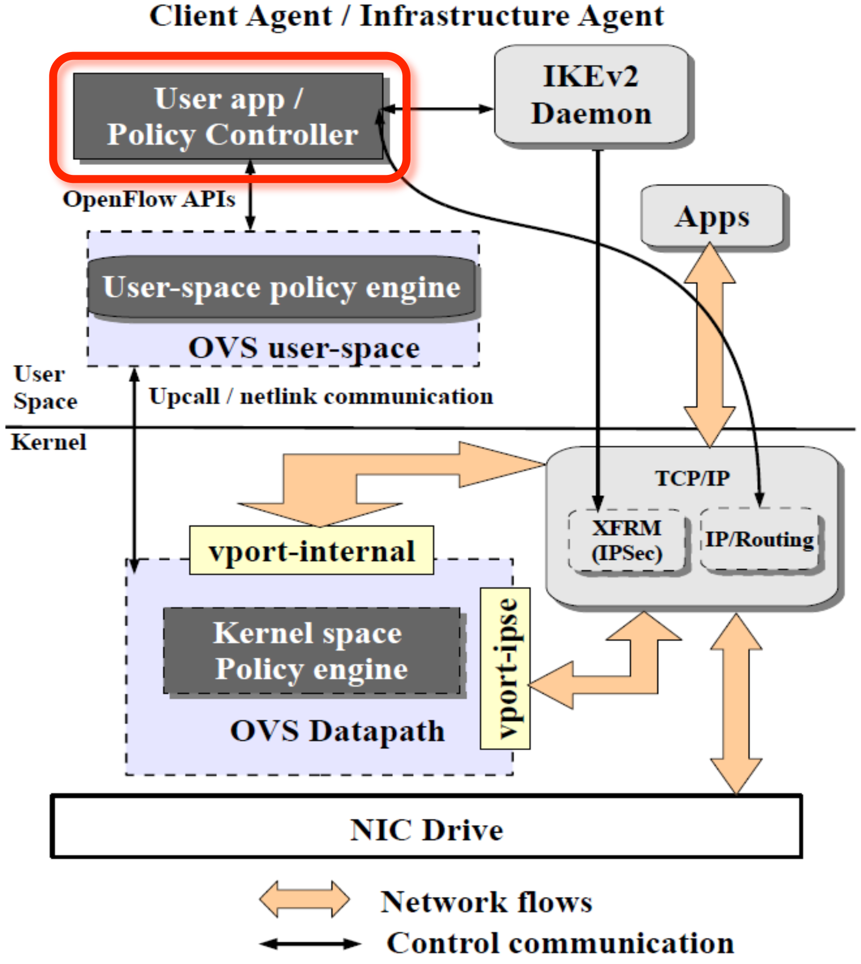
System Overview and Architecture

PrivacyGuard: Overview



PrivacyGuard: Architecture

Overall architecture on both client and infrastructure agents

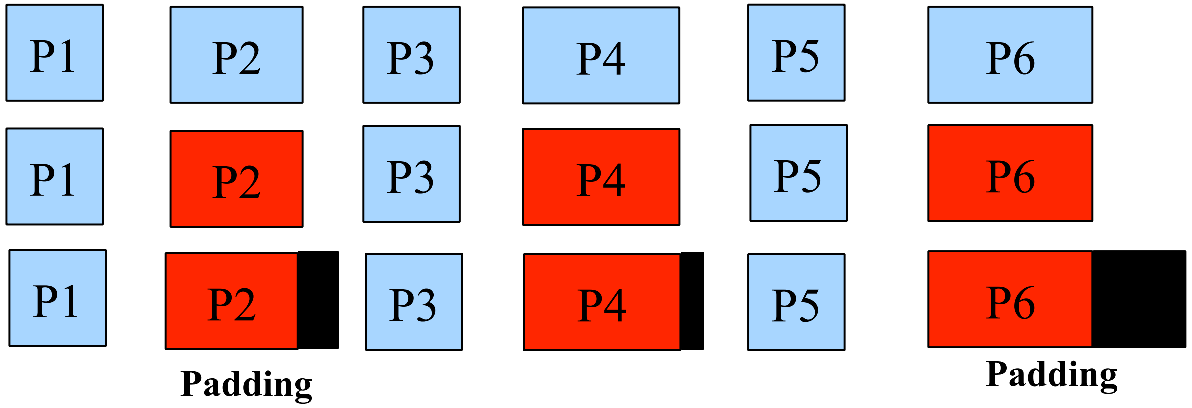


User-app: Handle user interface and track active applications

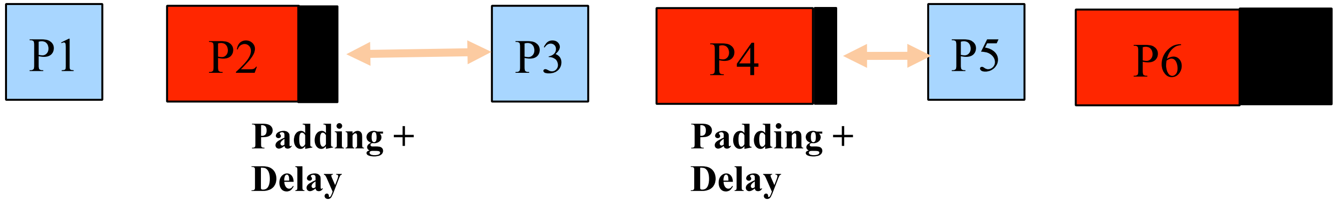
- user interface - categorize applications, define privacy preserving schemes.
- flow-to-application mappings
- configure the IPSec tunneling module
- release allocated resources at the end

PrivacyGuard: Privacy-Preserving Schemes

- PrivacyGuard can programmatically apply different privacy preserving schemes
- Packet Padding
  - Original application traffic flow
  - Uniform distribution based packet selection
  - Size of padding follows selected distribution and Configuration parameters
  - Norm_Pad [Gaussian, mean = 200 , stddev = 100]



- Packet Delaying
  - Inter arrival time based on uniform distribution from Min-Max range



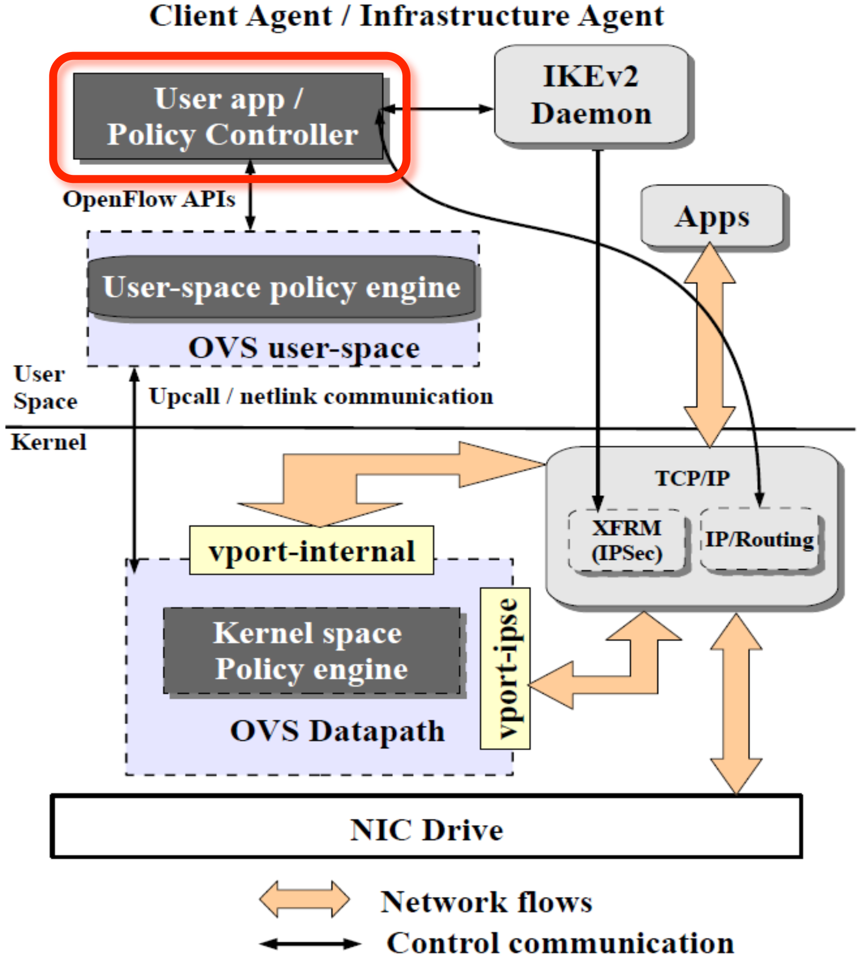
  - Norm_Pad_Delay [Gaussian , mean = 200 , stddev= 100 , IPT = {Gaussian , min =0 , max =20ms}]
  - Max_Pad_Delay [Gaussian , mean = 1500 (MTU) , stddev= 10 , IPT = {Gaussian , min =0 , max =20ms}]

VCU

MuSIClab

PrivacyGuard: Architecture

Overall architecture on both client and infrastructure agents



User-app: Handle user interface and track active applications

- user interface - categorize applications, define privacy preserving schemes.
- flow-to-application mappings
- configure the IPSec tunneling module
- release allocated resources at the end

Policy Controller: Convert application privacy preserving schemes to the flow-level policies.

- create and maintain the flow-policy table entries
- periodically estimate the current contexts

VCU

MuSIClab

PrivacyGuard: Context Information

Application Context

- High Sensitive Applications or Flows (revealing medical, activity information etc.) should use high obfuscation scheme.
- Low sensitive applications should not use any scheme or low overhead scheme.

User Context

- User location , time.
- Secure location (e.g., home) can have less efficient scheme for sensitive applications.
- Unsecure location (e.g., coffee shop or hotspot) can have high efficient scheme.

Device Context

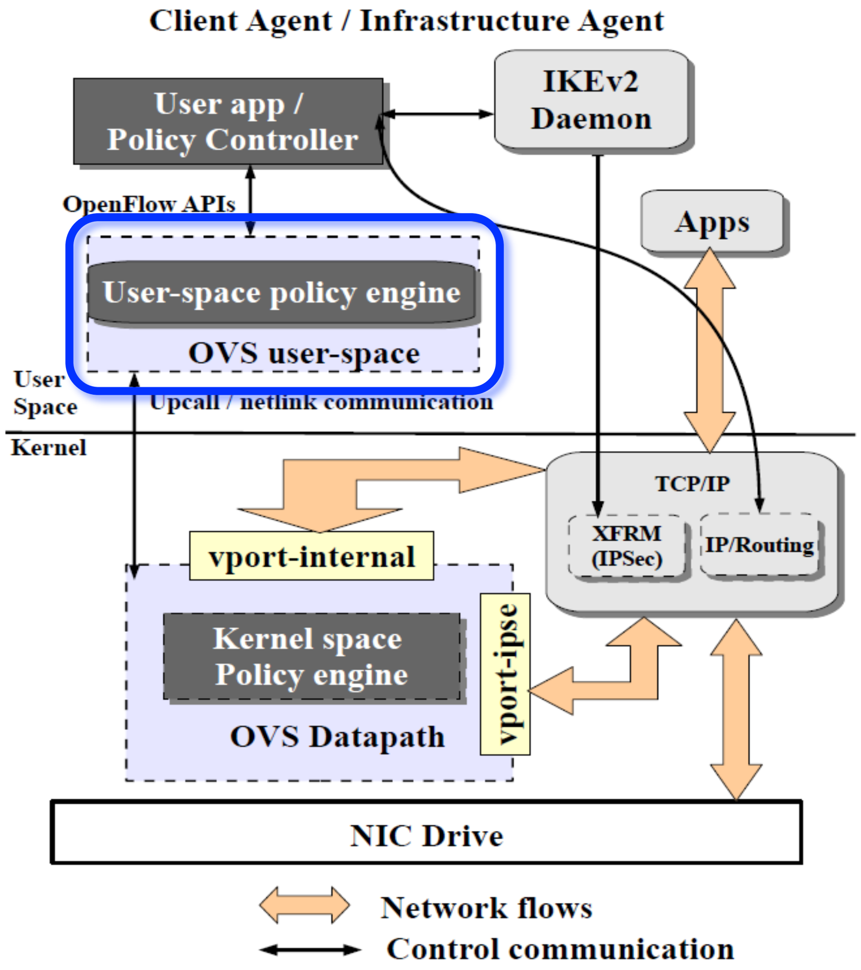
- Battery Level, Computing power.
- High battery Level , more suitable to apply high efficient scheme.
- Battery Level drops below certain threshold , switch to low power consumption and less efficient scheme.

Network Context

- Unencrypted Wi-Fi Hotpsot or Train station.
- High Load traffic , privacy schemes with low network bandwidth overhead would be preferable.

PrivacyGuard: Architecture

Overall architecture on both client and infrastructure agents



User-space Policy Engine: Maintain the flow policy table and use it in configuring the OVS forwarding element.

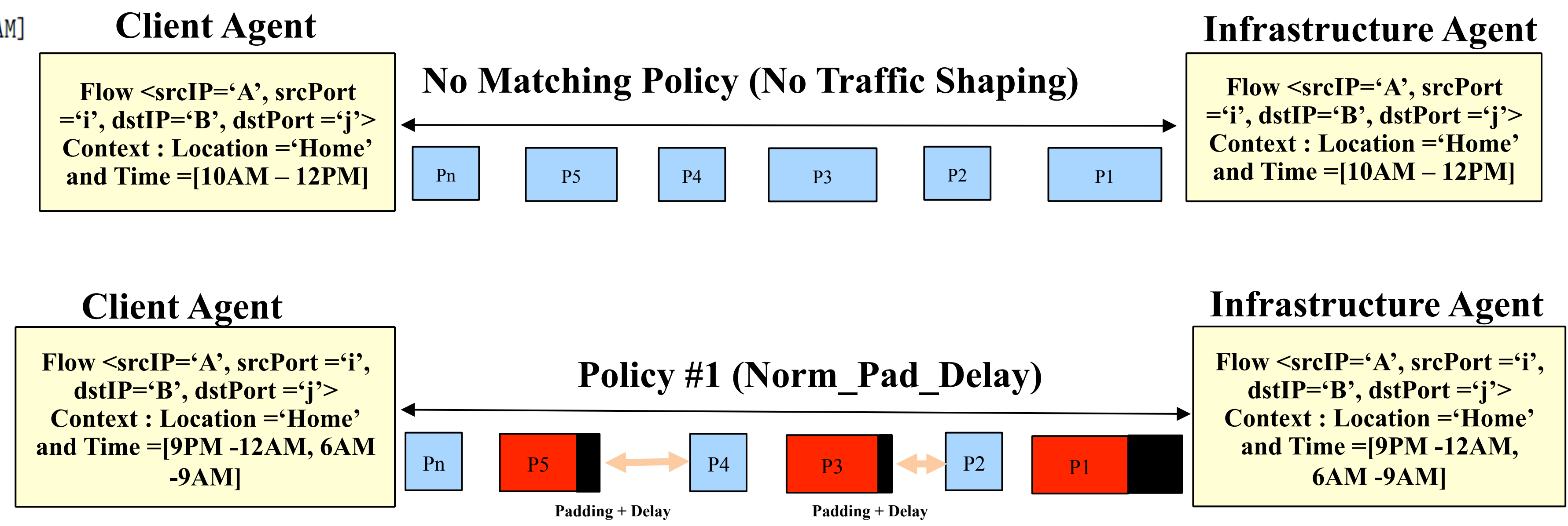
- extend OpenFlow APIs in OVS
- maintain and utilize the entries of flow-policy table
- search the flow-policy table entries to find the policy entry for new starting flows

VCU

MuSIClab

PrivacyGuard: Flow Policy Table

```
Policy #1
  ID: srcIP='A', srcPort='i', dstIP='B', dstPort='j'
  CONTEXT: Location='Home' AND Time=[9PM-12AM, 6AM-9AM]
  ACTION: Padding='Normal:μ=1500,σ=10, p=1.0'
          Delay='Uniform:min=0,max=20ms'
Policy #2
  ID: srcIP='A', srcPort='k', dstIP='B', dstPort='l'
  CONTEXT: Location='Home'
  ACTION: Padding='Normal:μ=400,σ=100, p=0.6'
Policy #3
  ID: srcIP='A', srcPort='m', dstIP='D', dstPort='n'
  CONTEXT: Battery=High AND Location=HotSpot
  ACTION: Padding='Normal:μ=1500,σ=10, p=1.0'
          Delay='Uniform:min=0,max=20ms', IPSec
Policy #4
  ID: srcIP='A', srcPort='m', dstIP='D', dstPort='n'
  CONTEXT: Battery=High OR WiFi Load=Low
  ACTION: Padding='Normal:μ=1500,σ=10, p=1.0'
          Delay='Uniform:min=0,max=20ms'
Policy #5
  ID: srcIP='A', srcPort='m', dstIP='D', dstPort='n'
  CONTEXT: Battery=Low OR WiFi Load=High
  ACTION: Padding='Normal:μ=1500,σ=10, p=0.6'
          Delay='Uniform:min=0,max=20ms'
```

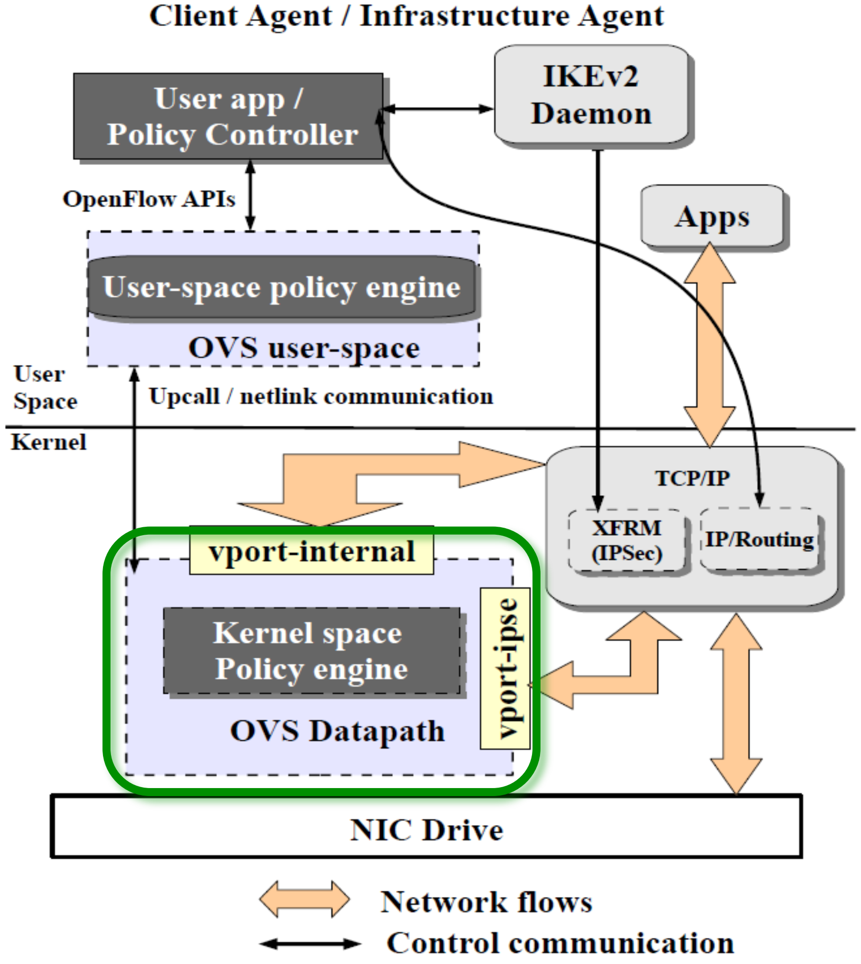


VCU

MuSIClab

PrivacyGuard: Architecture

Overall architecture on both client and infrastructure agents

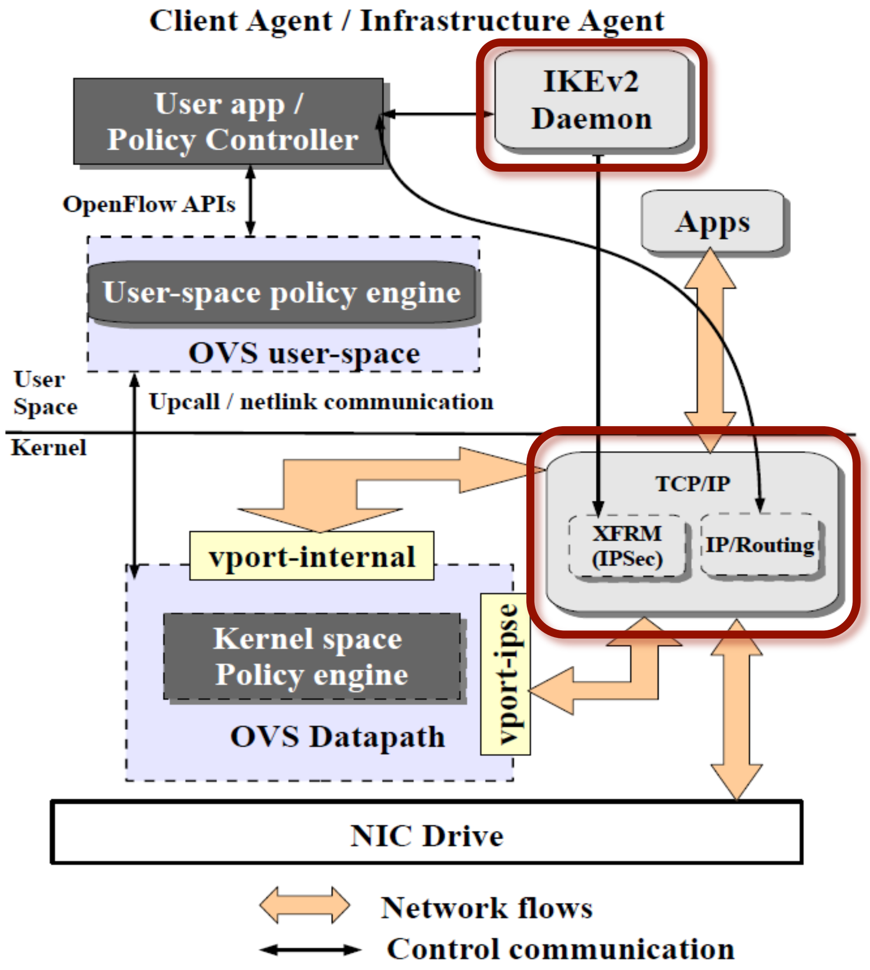


Kernel-space Policy Engine: Apply traffic shaping policy on flow packets

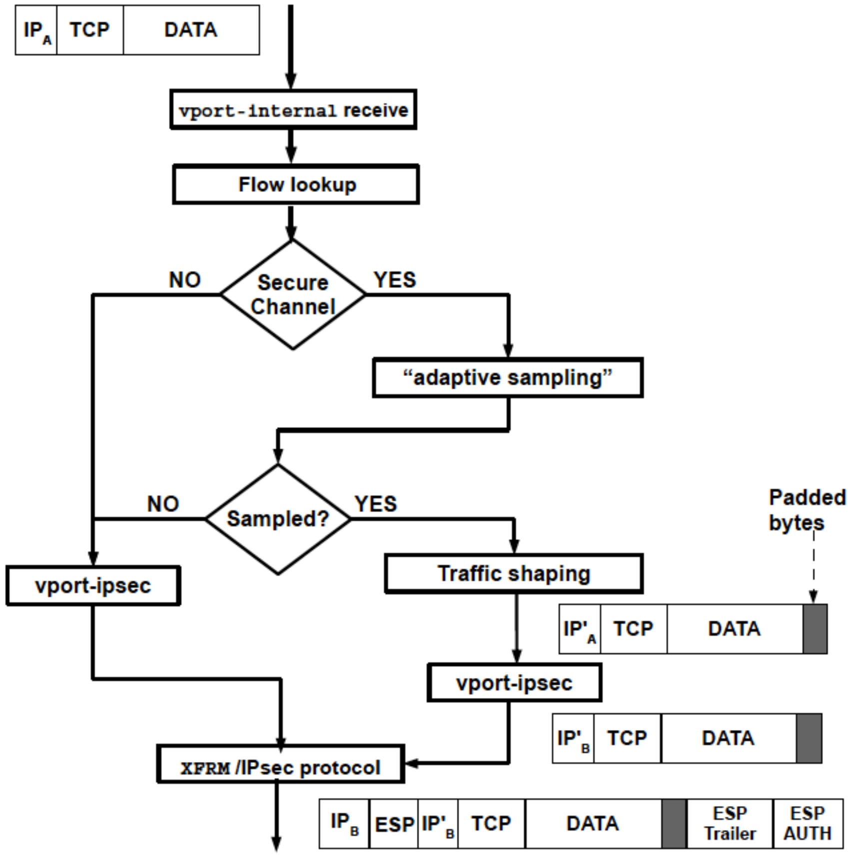
- trace new flows to get corresponding policies
- introduced new data path actions corresponding to the privacy preserving schemes
  - "adaptive sampling", "padding", and "reverse padding" for the packet-padding policies.
  - "delay" for the packet-delaying policies
- implement a new qdisc scheduler for Linux Traffic Control (tc)
- utilize the unused reserved bits of the "ToS" & "Options" fields in the IP header to mark the padded packets and corresponding parameters

PrivacyGuard: Architecture

Overall architecture on both client and infrastructure agents



IPSec Tunneling: Details in the paper



Performance Evaluation

Experiment Setup

Testbed

- Client agent – Nexus 4 Smartphones with Android 4.4 running
- Infrastructure agent – Ubuntu Laptop (access point)
- Installed 8 commercially available IoT device applications on the Nexus Device (acting as gateway)
  - Applications span different domains including home appliance, medical , activity fitness.

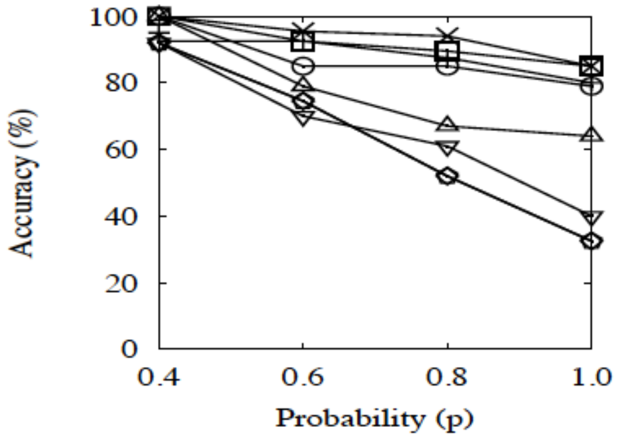
Traffic Shaping

- Three different traffic shaping schemes based on packet padding and packet delaying
- Norm_Pad , Norm_Pad_Delay and Max_Pad_Delay

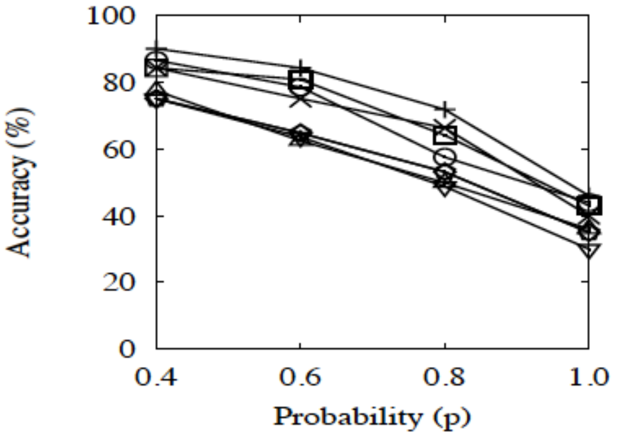
Metrics

- Accuracy, Precision, Energy overhead, Network overhead

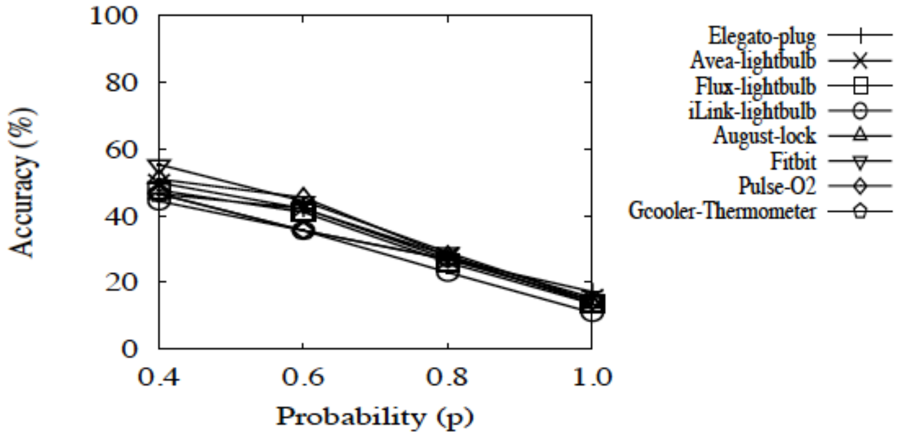
Traffic Shaping Schemes Performance



Accuracy of Norm_Pad scheme for different applications and p values



Accuracy of Norm_Pad_Delay scheme for different applications and p values



Accuracy of Max_Pad scheme for different applications and p values

- Scheme has high efficiency for Fitbit with large values of p , but fails in obfuscating applications such as Flux-lightbulb application.
- Low efficiency with applications that transmit their packets in periodic patterns (Elegato plug, Avea-Lightbulb, Flux Light bulb and ilink-Lightbulb)

- More efficiency for applications that transmit packets at periodic patterns

- Efficiency exceeds other the other two scheme even at low values of p
- Applications such as Elegato Plug iLink-lightbulb and Flux-lightbulb transmit many large size packets which can obfuscated by padding packets to MTU

VCU

MuSIClab

Programmability and Flexibility

Flexibility in setting Policies

Max_Pad
Fitbit
150 % more energy

Saves Battery →

Policy #1
Max_Pad,
Senstive
Time Zone

Ability to adapt to context changes

Unsaturated Network

Policy #4

Network overhead 150 % to 80% but moving to less efficient scheme →

Saturated Network

Policy #5

Low Battery

Policy #4

Moving to High Efficient Scheme but high overhead scheme →

High Battery

Policy #5

Moving to IPSec Enabled Scheme →

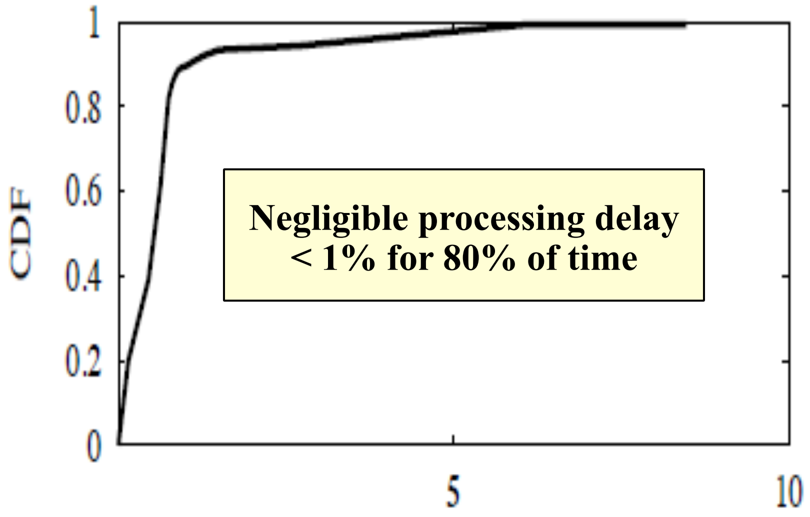
Insecure Location &&
High Battery
Policy #3

```
Policy #1
  ID: srcIP='A', srcPort='i', dstIP='B', dstPort='j'
  CONTEXT: Location='Home' AND Time=[9PM-12AM, 6AM-9AM]
  ACTION: Padding='Normal:μ=1500,σ=10, p=1.0'
          Delay='Uniform:min=0,max=20ms'
Policy #2
  ID: srcIP='A', srcPort='k', dstIP='B', dstPort='l'
  CONTEXT: Location='Home'
  ACTION: Padding='Normal:μ=400,σ=100, p=0.6'
Policy #3
  ID: srcIP='A', srcPort='m', dstIP='D', dstPort='n'
  CONTEXT: Battery=High AND Location=HotSpot
  ACTION: Padding='Normal:μ=1500,σ=10, p=1.0'
          Delay='Uniform:min=0,max=20ms', IPSec
Policy #4
  ID: srcIP='A', srcPort='m', dstIP='D', dstPort='n'
  CONTEXT: Battery=High OR WiFi Load=Low
  ACTION: Padding='Normal:μ=1500,σ=10, p=1.0'
          Delay='Uniform:min=0,max=20ms'
Policy #5
  ID: srcIP='A', srcPort='m', dstIP='D', dstPort='n'
  CONTEXT: Battery=Low OR WiFi Load=High
  ACTION: Padding='Normal:μ=1500,σ=10, p=0.6'
          Delay='Uniform:min=0,max=20ms'
```

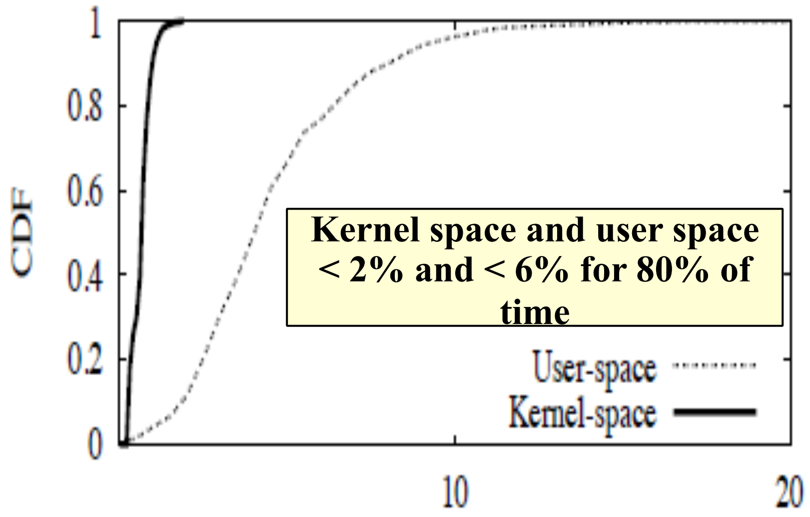
VCU

MuSIClab

PrivacyGuard Overhead



Processing delay overhead of PrivacyGuard(%)



CPU usage overhead of PrivacyGuard (%)

Conclusion / Future Work

Conclusion

- Design and develop PrivacyGuard; a flexible programmable privacy-preserving framework to obfuscate the activities of sensitive IoT and mobile applications
- Realize and implement a prototype of PrivacyGuard on android Mobile devices
- Evaluate and analyze the performance of PrivacyGuard using different commercial IoT based apps.

Future Work

Recommend optimal privacy schemes

- Crowdsourcing
- Reinforcement Based Learning

Other Attack Models schemes

- Understand restriction and impact of different obfuscation schemes
- IoT Device to Access Point attack Model
- ISP attack Model

PrivacyGuard API

- To be utilized by application developers.
- During low battery level, application developer could configure the app to drop less useful functional flows (advertising data).

Thank You!

QUESTIONS

tnadeem@vcu.edu

https://music.lab.vcu.edu/

SmartEdge'19 Workshop

VCU

MuSIC lab