Side-Channel Attacks on Open vSwitch

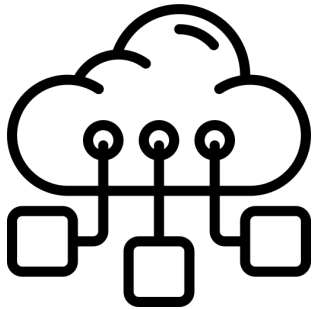
Daewoo Kim and Sihang Liu

University of Waterloo



Explosion of Cloud

- Cloud is foundational in today's computing
  - Scalable compute and storage
  - Elastic resource allocation
  - On-demand services
- Networking is especially critical to the cloud
  - Achieve high-speed interconnect



New Security Concerns

- Sharing is common in cloud environments
- Resource sharing introduces new security risks, e.g.,
  - Network interconnect
  - Compute and storage systems
- Virtualization and isolation can mitigate leakages but not always

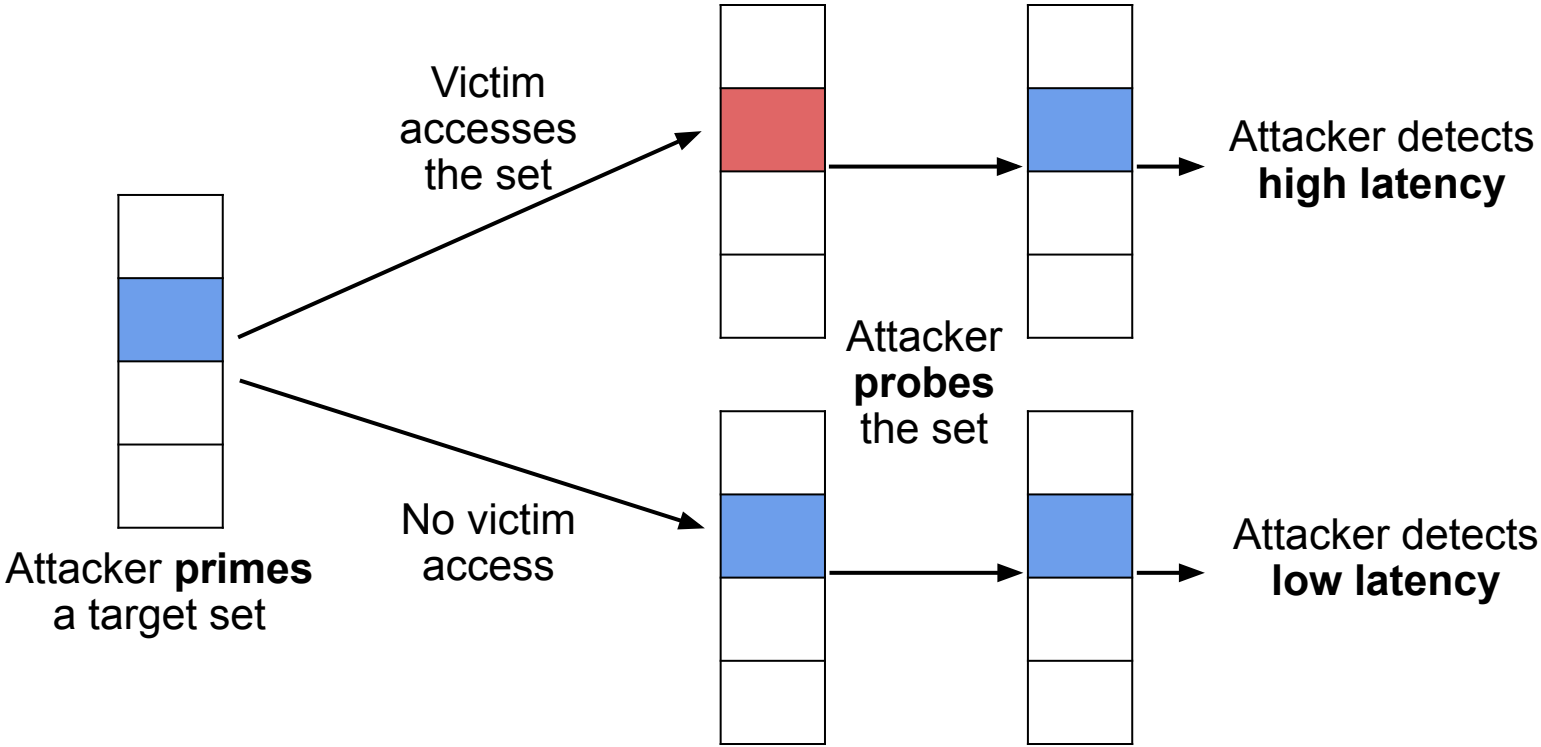
What Are Side-Channel Attacks?

- Side channels are based on indirect, unintended behaviors or features
- Secretly leak information about the target system
- Examples of side channels:
  - Chip power
  - Thermal signal
  - Electromagnetic signal
  - Timing difference

Cache is one of the most typical examples that lead to timing differences

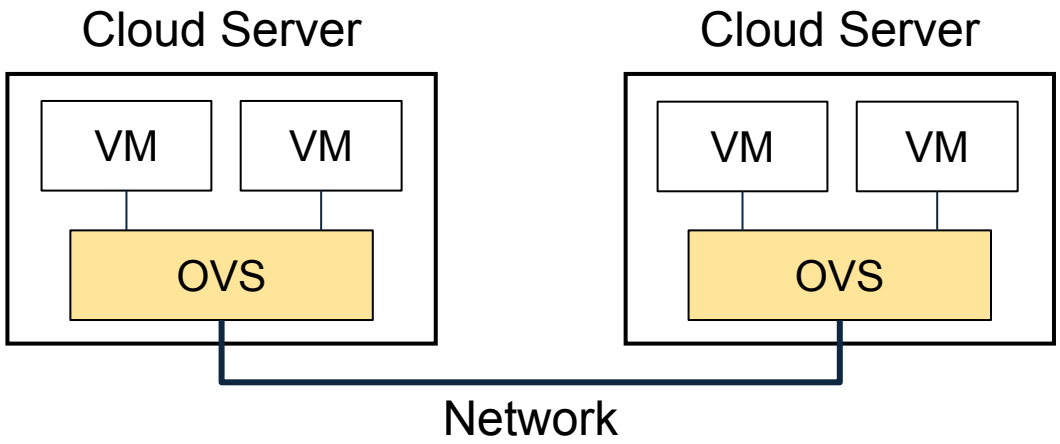
Example of Cache Side Channel

A Prime+Probe attack can infer bits of a memory address through cache



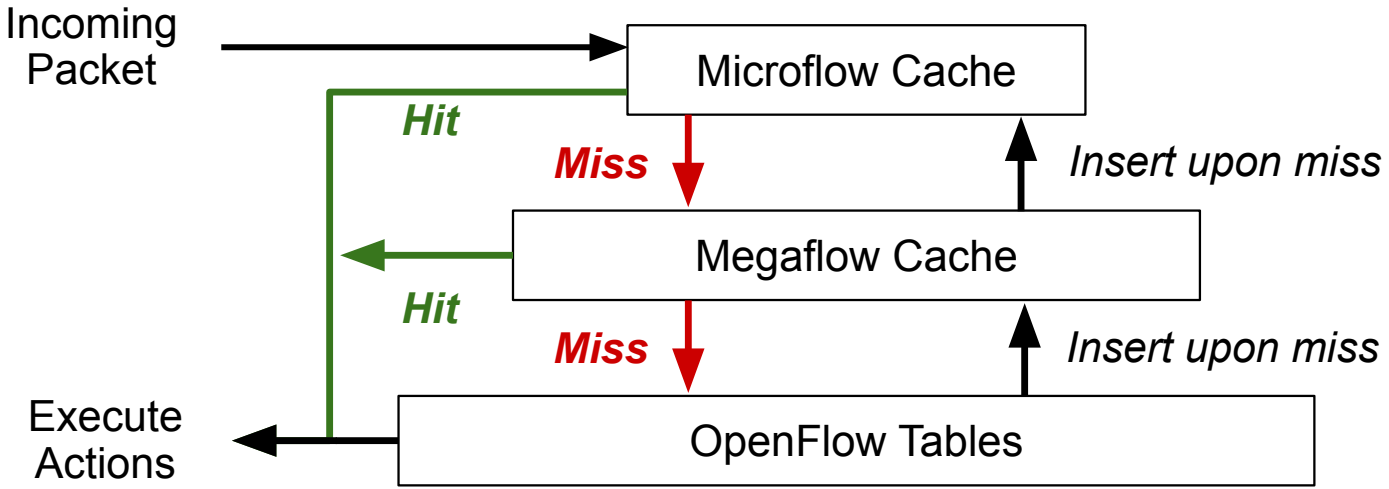
Open vSwitch (OVS)

- Cloud environments usually share resources via VMs
- OVS is one way to accelerate the network in cloud environment
  - It is a virtual switch that connects VMs or containers
  - Works with SDN and enables efficient and flexible network management



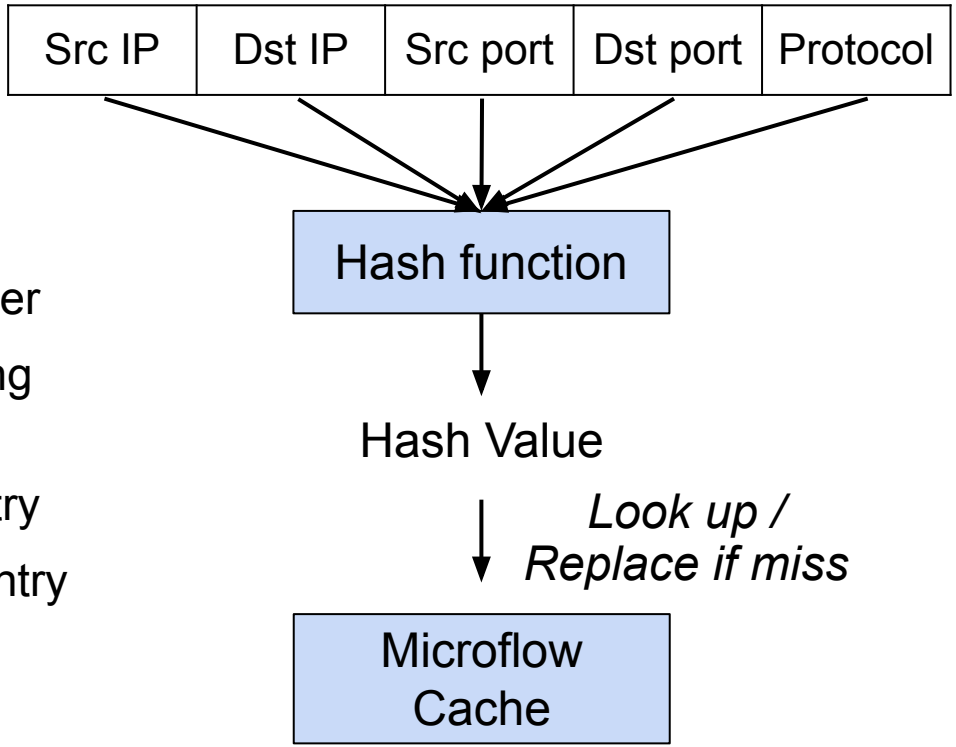
Flow Caching in OVS

- OVS caches network flows to accelerate flow actions
- It has a two-level cache: **microflow cache** and **megaflow cache**
  - The incoming packet first looks up microflow cache
  - If **hit**, execute actions
  - If **miss**, look up megaflow cache and insert flow back to microflow

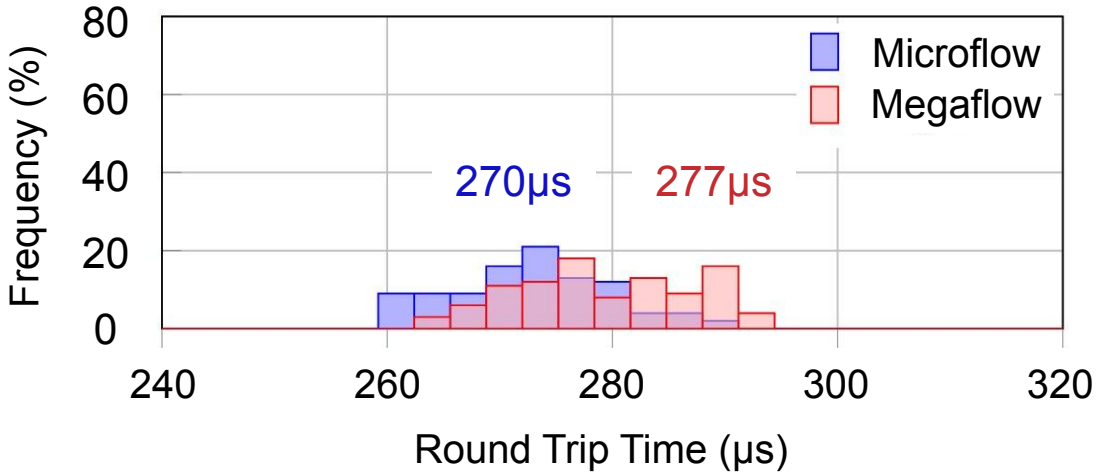


Microflow Cache

- Microflow cache is based on a hash table
- Lookup process:
  1. Generate hash from packet header
  2. Look up the microflow cache using the hash value
     - **Hit**: Use the cached flow entry
     - **Miss**: Replace the conflict entry



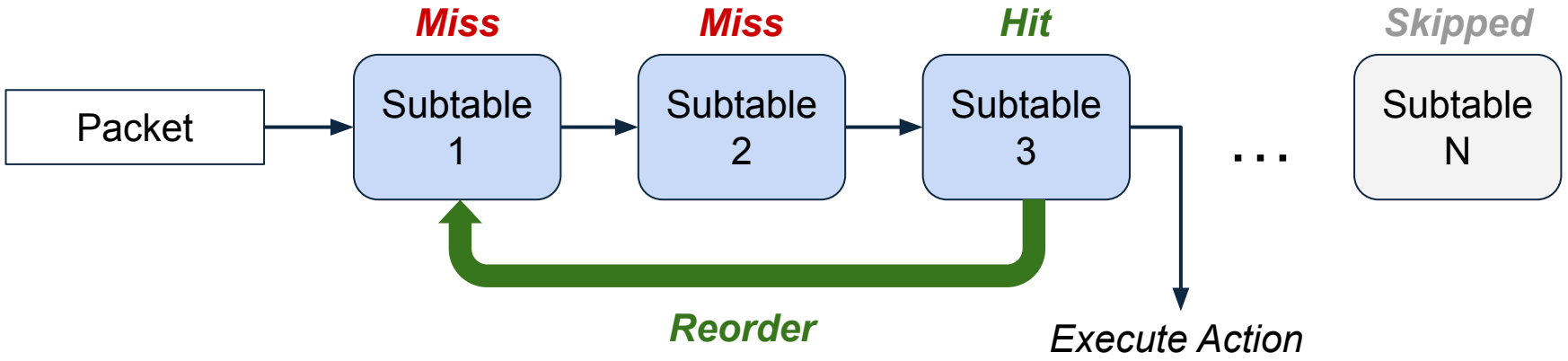
Microflow Cache Latency



OVS microflow and megaflow have distinguishable latencies

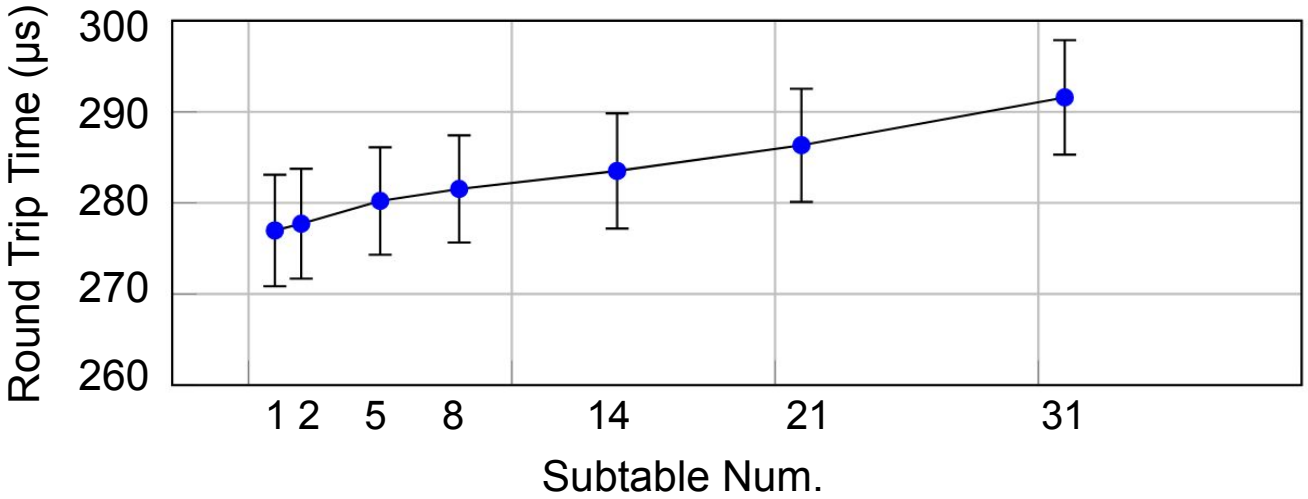
Megaflow Cache

- Megaflow cache consists of a number of subtables
  - Look up subtables sequentially until hit
  - Latency depends on num. subtable lookups
  - Subtables are reordered to the front if accessed frequently



Megaflow Cache Latency

- Evaluate access latency vs. subtable location



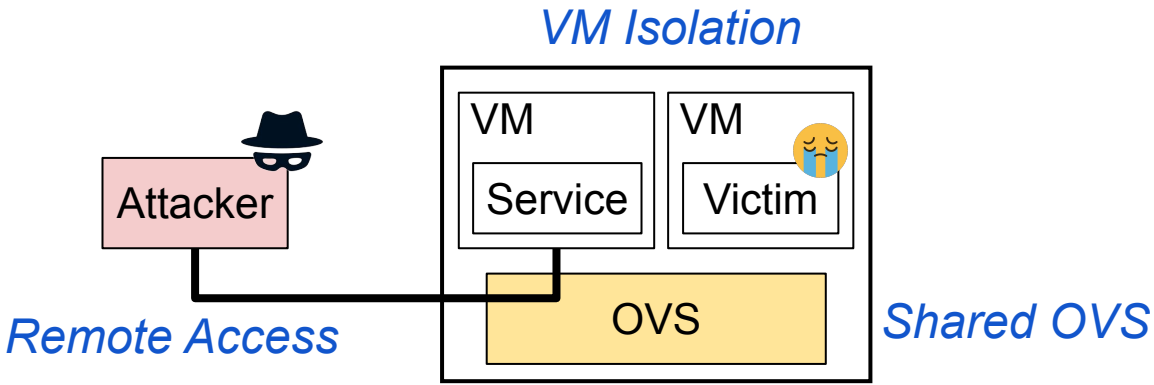
Megaflow has distinguishable levels of latencies, corresponding to subtables order

Summary of OVS Attack Primitives

1. **Timing difference in OVS**
   - Microflow, megaflow, and cache miss have distinguishable latencies
2. **Microflow cache hash collisions**
   - The hash for the microflow cache is generated from packet header fields
   - Collisions can leak information about packet header fields
3. **Megaflow subtable ordering**
   - The megaflow subtable ordering is based on access frequency
   - Subtable latency can leak traffic rate

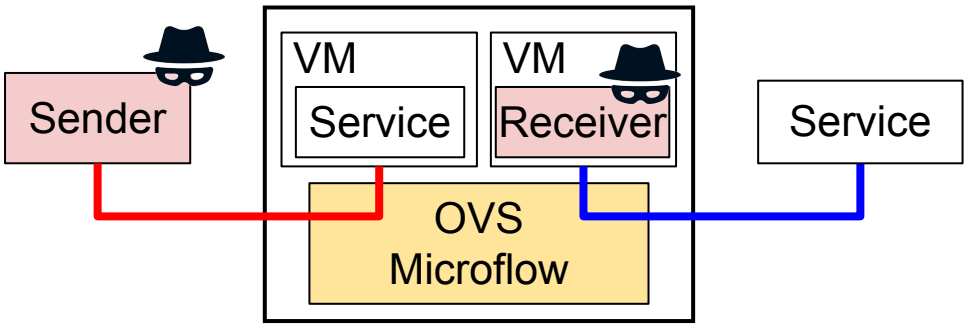
Attack Model

- Assume a cloud environment
  - Attacker and victim have no direct co-location
  - Attacker may access services co-located with the victim's server, but are isolated by VMs
- Only OVS is shared

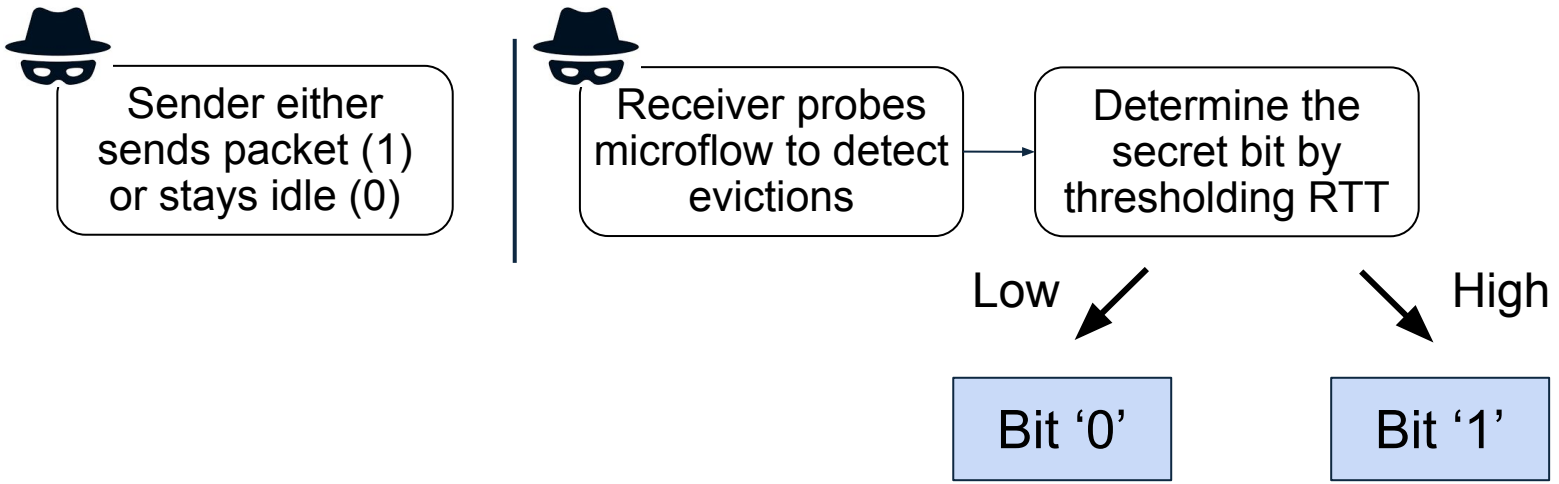


Remote Covert Channel

- Receiver knows the microflow cache entry that sender uses
  - By probing all microflow cache entries
- Receiver periodically probes the shared entry
  - Decide whether a transmitted secret bit is ‘0’ or ‘1’
- Use Attack Primitives 1 & 2 to detect the sender's transmitted bits

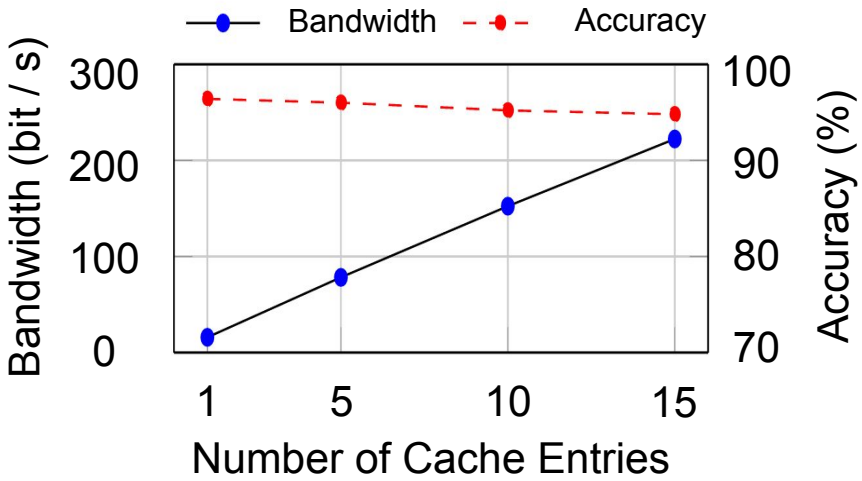


Remote Covert Channel



Covert Channel Accuracy and Bandwidth

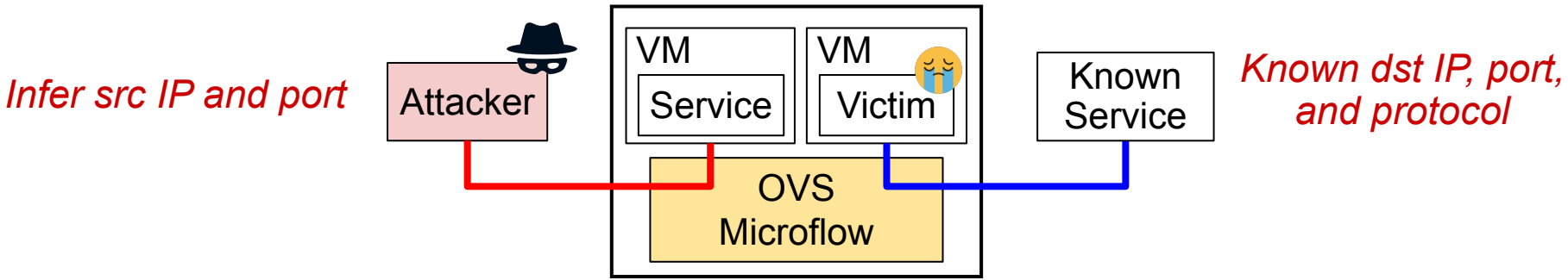
- Receiver measures the RTT 50 times to determine a secret bit
- Using multiple microflow cache entries increases the bandwidth



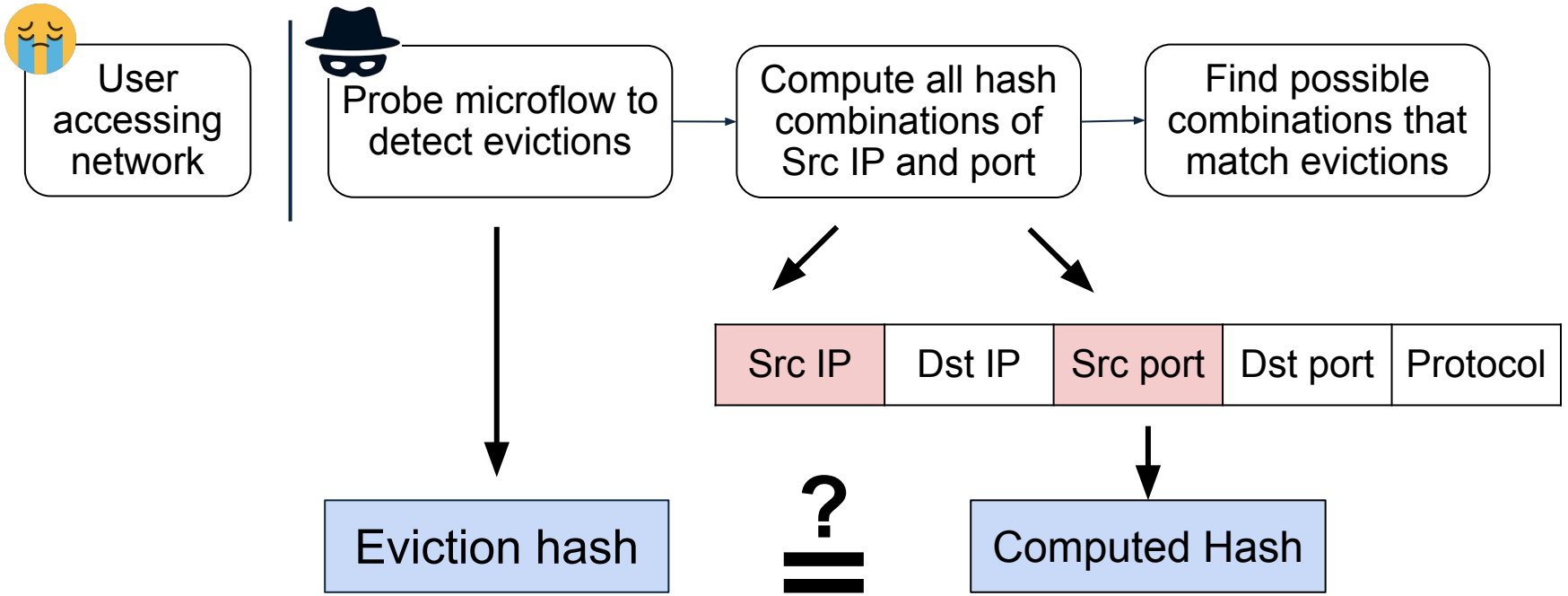
Covert channel can be established by exploiting the microflow cache

Remote Packet Header Recovery Attack

- Attacker knows victim is accessing some well-known service
  - Victim's Dst IP, port, and protocol are known
- Infer victim's remaining header fields
  - Victim's Src IP and port are targets
- Use Attack Primitives 1 & 2 to detect microflow collisions

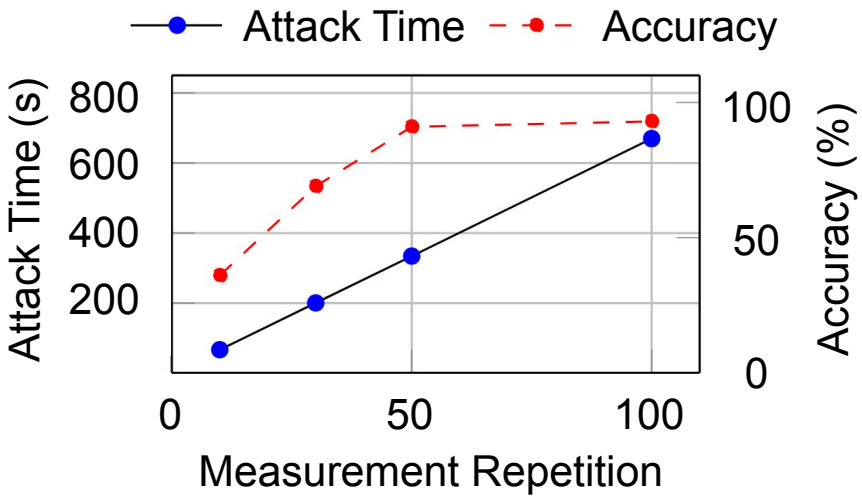


Remote Packet Header Recovery Attack



Recovery Accuracy and Time

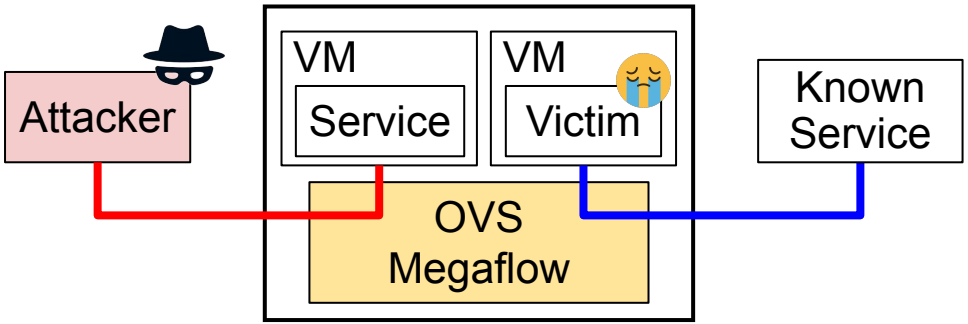
- Measurement repetition increases recovery accuracy
- Probing time increases as attacker repeats probing



Packet header fields can be recovered by probing active entries in microflow cache

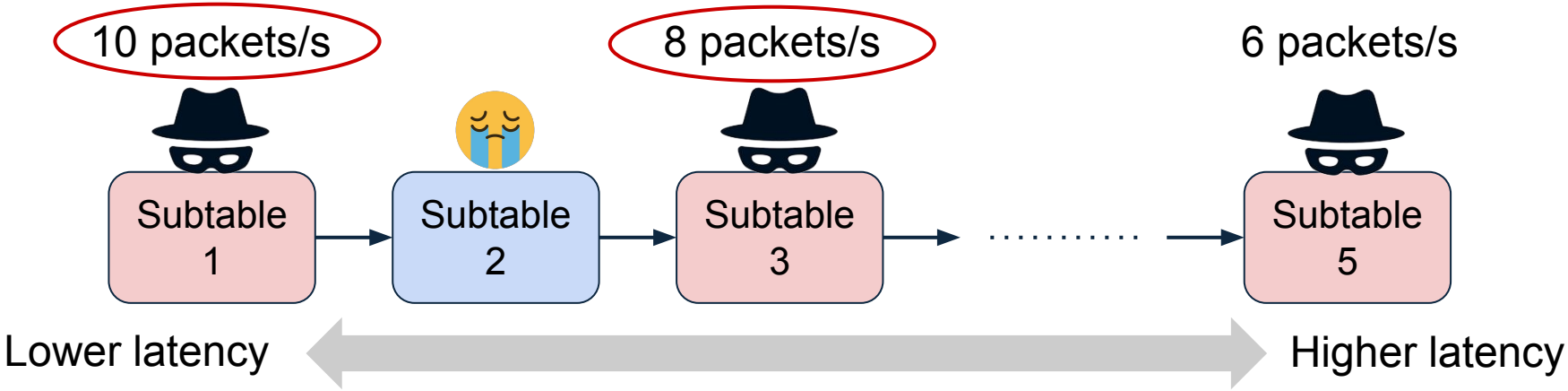
Remote Packet Rate Monitoring Attack

- Attacker has knowledge about victim's packet header
  - Using packet header recovery attack
- Attacker can locate victim flow's subtable and probe it
- Use Attack Primitive 3 to monitor victim's packet rate



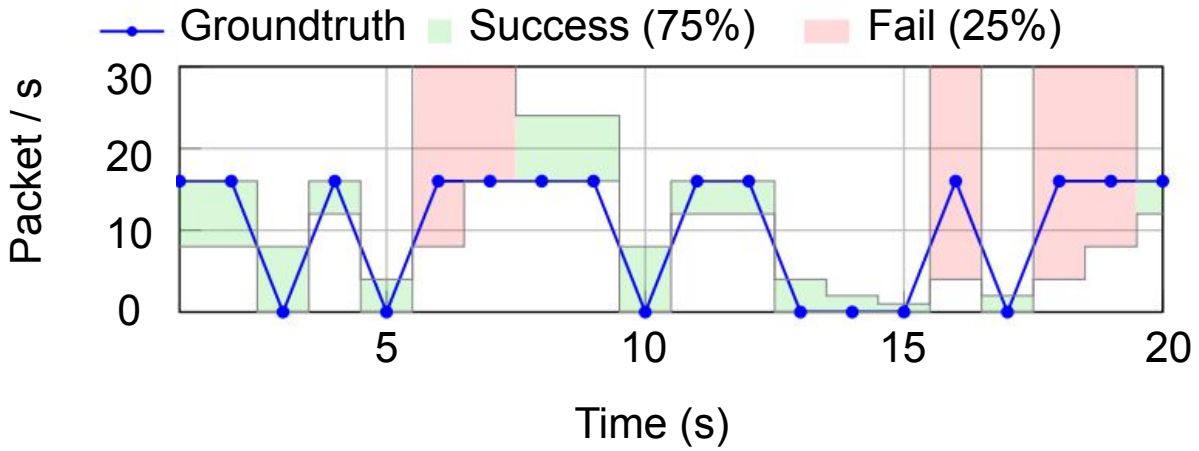
Remote Packet Rate Monitoring Attack

- Attacker measures latency to victim's co-located subtable
- Attacker accesses certain subtables that work as "thresholds"
  - Send packets at fixed rates
  - Compare the victim's subtable latency and determine the relative ordering
- Use neighboring thresholds to identify victim's packet rate



Recovery Accuracy

- Replay packets based on timestamps in UNSW-NB15
- Recovery of an example flow:



Attacker can monitor packet rate at 71.9% accuracy on average

Defense Mechanisms

1. **OVS instance isolation**
   - Eliminates OVS sharing among tenants
   - Defends against all of our proposed attacks
2. **Microflow cache hash randomization**
   - Randomizes the hash function
   - Defends against all of our proposed attacks
3. **Megaflow cache reordering randomization**
   - Randomizes the reordering interval
   - Mitigates only megaflow-cache-based attacks

Summary

- The wide use of cloud introduces a wider attack surface
- We demonstrate three side-channel attacks on OVS
  - Remote covert channel
  - Remote packet header recovery attack
  - Remote packet rate monitoring attack
- We present three defense mechanisms
  - OVS instance isolation
  - Microflow cache hash randomization
  - Megaflow cache reordering randomization