Dynamic Security with SDN: Opportunities, Challenges, and Lessons Learned

Cristina Nita-Rotaru

Khoury College of Computer Science

Northeastern University

Castle model of cybersecurity

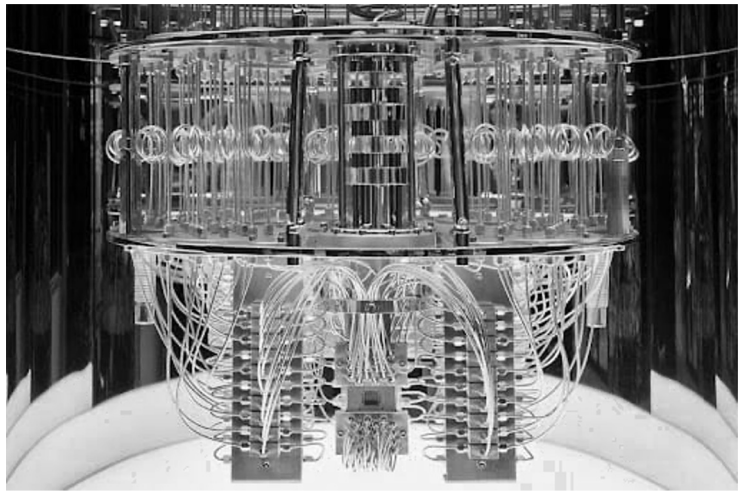
- Secure perimeter defined
  - Attackers are outsiders
  - Protection enforced at the gates
  - Coarse-granularity access control
- End-to-end secure communication
  - Based on cryptographic primitives that assume computationally bounded adversary

Model challenged by disruptions

Perimeter-based security no longer effective

Increased computational power

- Quantum computers will become mainstream by 2030, 2040
- Quantum computing breaks assumptions needed for the security of existing cryptographic primitives, e.g.
  - discrete log problem
  - factorization of large numbers



IBM Quantum System One (CES 2020), IBM Research

What does is mean for security

- Design security mechanisms that do not rely on network perimeter to enforce security
  - Federal government Zero Trust memo
- Design secure communication without relying on computational assumptions about the adversary
  - NIST Post-Quantum Cryptography (PQC), July 2022, four finalists announced

Zero Trust

- *"Zero trust…became the term used to describe various cybersecurity solutions that moved security away from implied trust based on network location and instead focused on evaluating trust on a per-transaction basis." NIST*
- *Zero trust does not mean no trust but*

  ``Narrow and specific trust after authentication"

  Bruce Davie

CONTINUALLY CHECK AUTHORIZATION

Other approaches to secure communication

- No computational assumptions about the attacker but …
- Limited compromise of distributed locations
  - Secret sharing
    - Split data in multiple pieces
- Limited network observability
  - Multi-path
    - Send the message on multiple, possibly disjoint paths
  - Path switching
    - Change randomly the path on which each message is sent

PERIODICALLY CHANGE PATHS DATA FLOWS ON

Challenges for dynamic network security

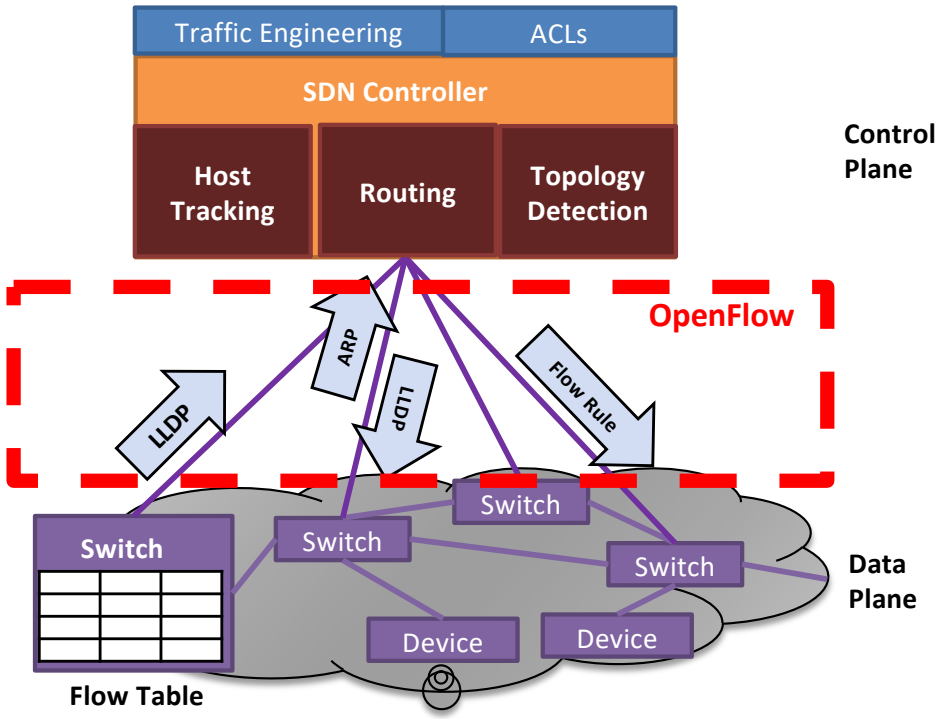
- Supporting low-granularity of enforcement
- Reducing the overhead of checking and enforcing policies
- Managing policy changes
- Scaling with number of users and devices
- Handling geographically distributed enclaves
- Handling mobility
- Reconciling the semantic gap between organizational structures and network-level enforcement

Software-defined programmable security

- Abstractions of physical capabilities are made available to applications or higher-level services in a way that is decoupled from the underlying physical device or infrastructure
- Software-defined approaches have been realized in the context of datacenters which may simultaneously deploy software-defined network, storage, and compute stacks
- Programmable frameworks for emerging computing eco-systems such as IoT, edge computing

Software-defined networking

- SDN switches
  - Forward traffic based on flow rules in a table
  - Send unmatched traffic to controller
- SDN controller
  - Contains network control logic
  - Detects network topology (via LLDP) and hosts (via ARP)
  - Pluggable apps expand functionality
- Standardized protocol
  - OpenFlow
  - Configuration protocol: specifies how to communicate, but not what commands to send



A word of caution

- New opportunities to redesign the security mechanisms and services
- Increased and new attack surfaces that deserve new research investigation
  - SDN exploits – BEADS [RAID 2018]
  - Identity-binding on weaker identities – Persona [Usenix 2017]
  - Vulnerabilities in SDN Apps – Cross-app poisoning [CCS 2018]

  - And many more …

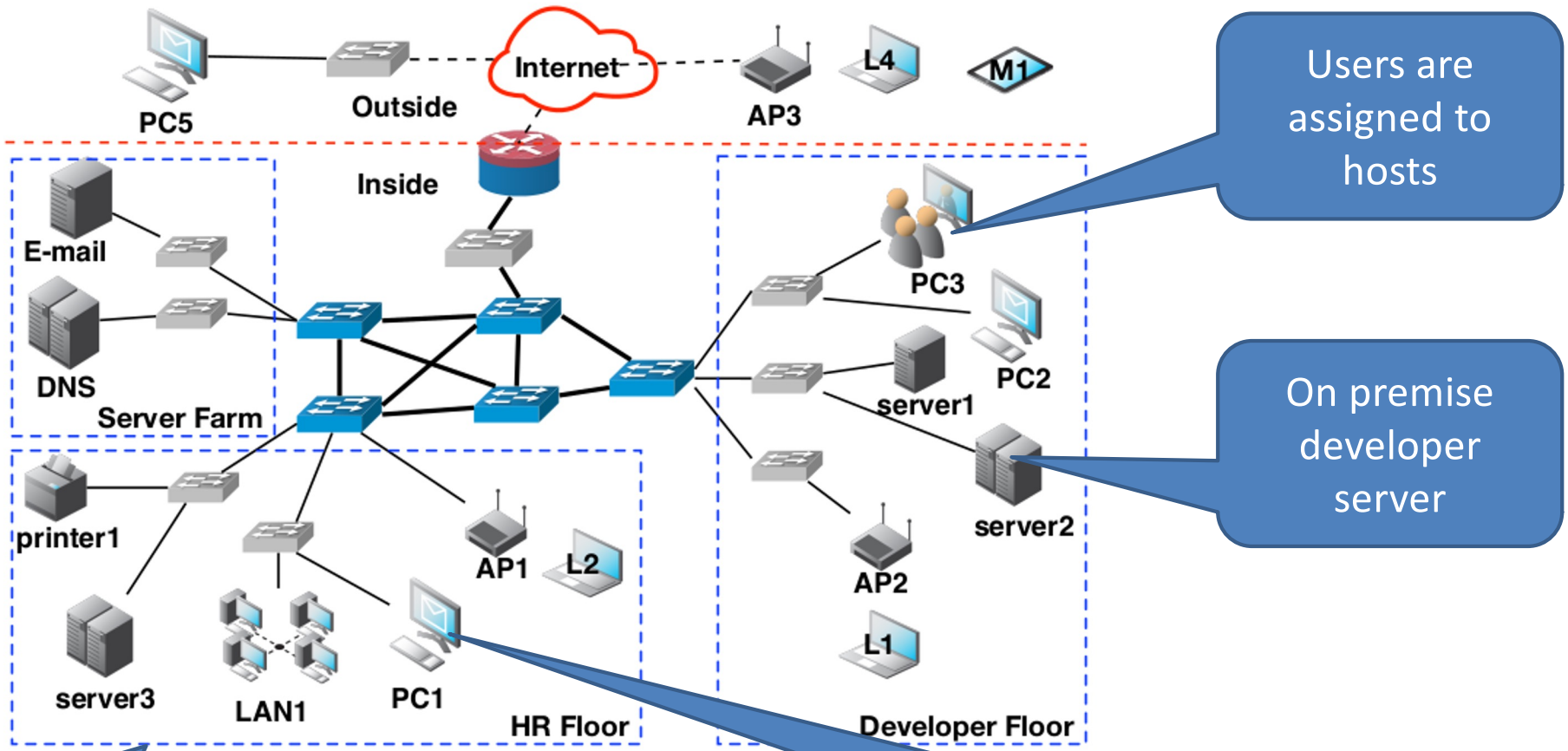
This talk

How to implement least privilege access control on network enclaves

Leverage software-defined networking

How to design secure communication primitives that do not assume computationally-bounded adversaries

Example enterprise network



Zero Trust in enterprise settings

- Emerging Zero Trust Models are **application-focused**
  - Move business applications to the cloud
  - Perform enhanced identity and access control checks within a web gateway
- What about the security needs of **on-premises** workstations, development/file servers, and device management interfaces?

Can we remove the network altogether?

How to support organizational structure?

Available solutions and limitations

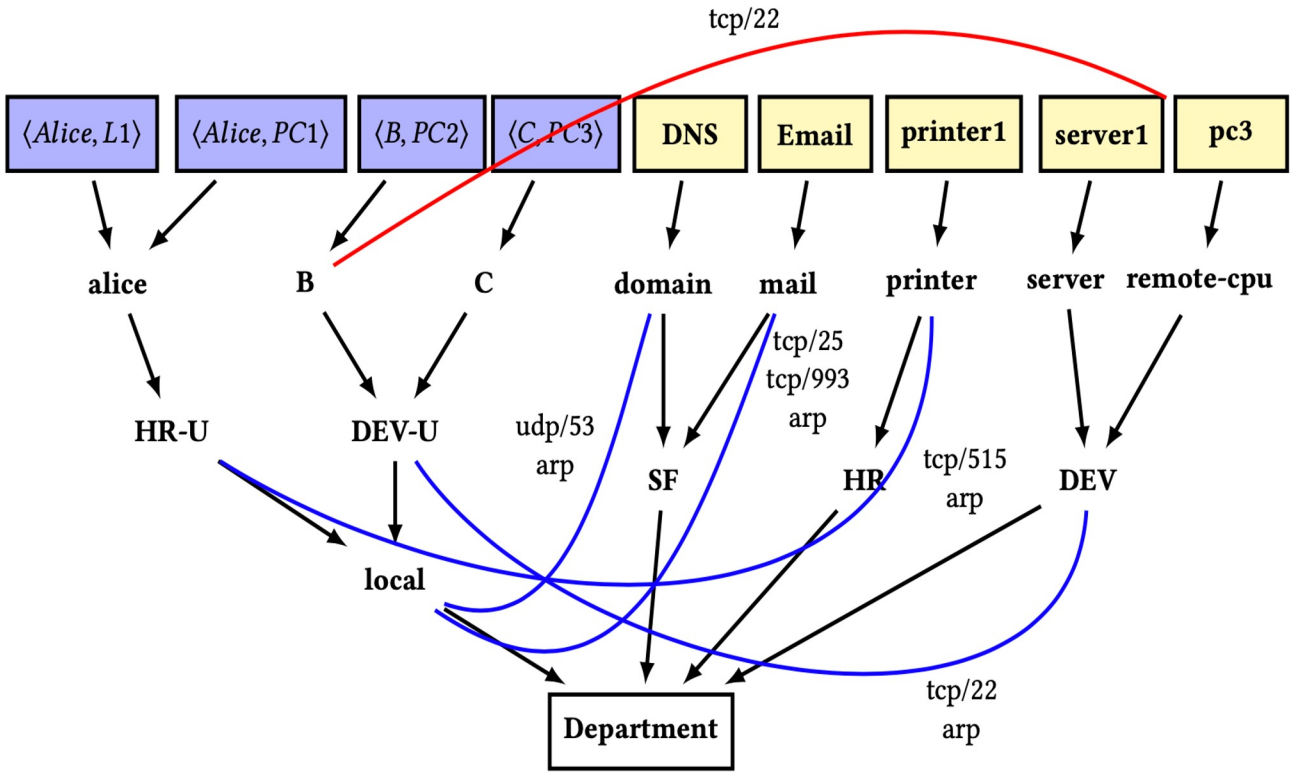
- **Host-based firewalls** can limit access of every other host by IP address
  - Managing host-based firewalls in mass deployments is complicated and may lead to lock-out
- **Microsegmentation** enables detailed implementation of security policies for specific application segments
  - It still relies on a perimeter for security
  - Can be deployed only within cloud; completely isolated segments not appropriate for end-user hosts

There is a semantic gap between networking primitives and an enterprise's organizational structure!

Network Views (abbrev. NetViews)

- **Goal:** secure the on-premises network environment
- **Intuition:** each host has a different "view" of what other hosts and services exist in the network
  - **Least privilege access control**
  - Embrace organizational needs
  - Fine-grained enforcement
- Design decisions:
  - How should NGAC policy concepts capture network primitives while bridging the semantics?
  - What are the semantics of an allow decision and also how should the networking infrastructure respond to an allow decision?

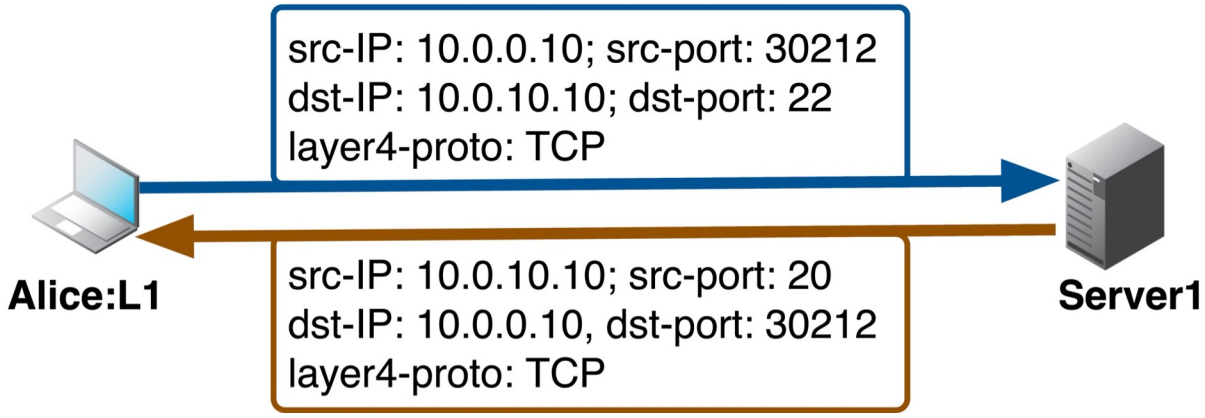
NetViews policy; Based on NIST's Next Generation Access Control (NGAC)



- **Users** are real-world ⟨*user*,*device*⟩ pair
- **Objects** are network servers
- **Actions** are Layer-4 identifiers (e.g., tcp/22)

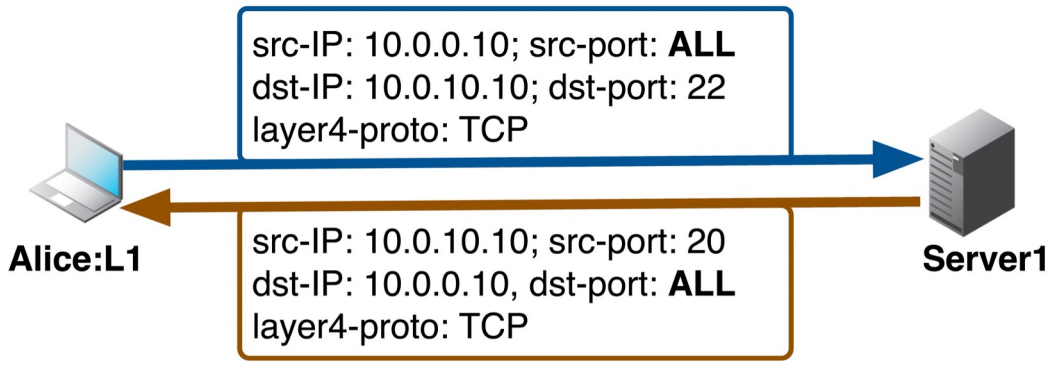
Access control semantics

- Traditional firewalls are either **stateful** or **stateless**
  - Stateful firewalls are more secure: limit probing (e.g., ACK scan)
  - OpenFlow cannot enforce a stateful firewall policy
- **But …** the more fine-grained the Flow-Rules, the more "state" that is stored within the network forwarding rules
  - E.g., react to Packet-In with 5-tuple for **both directions**

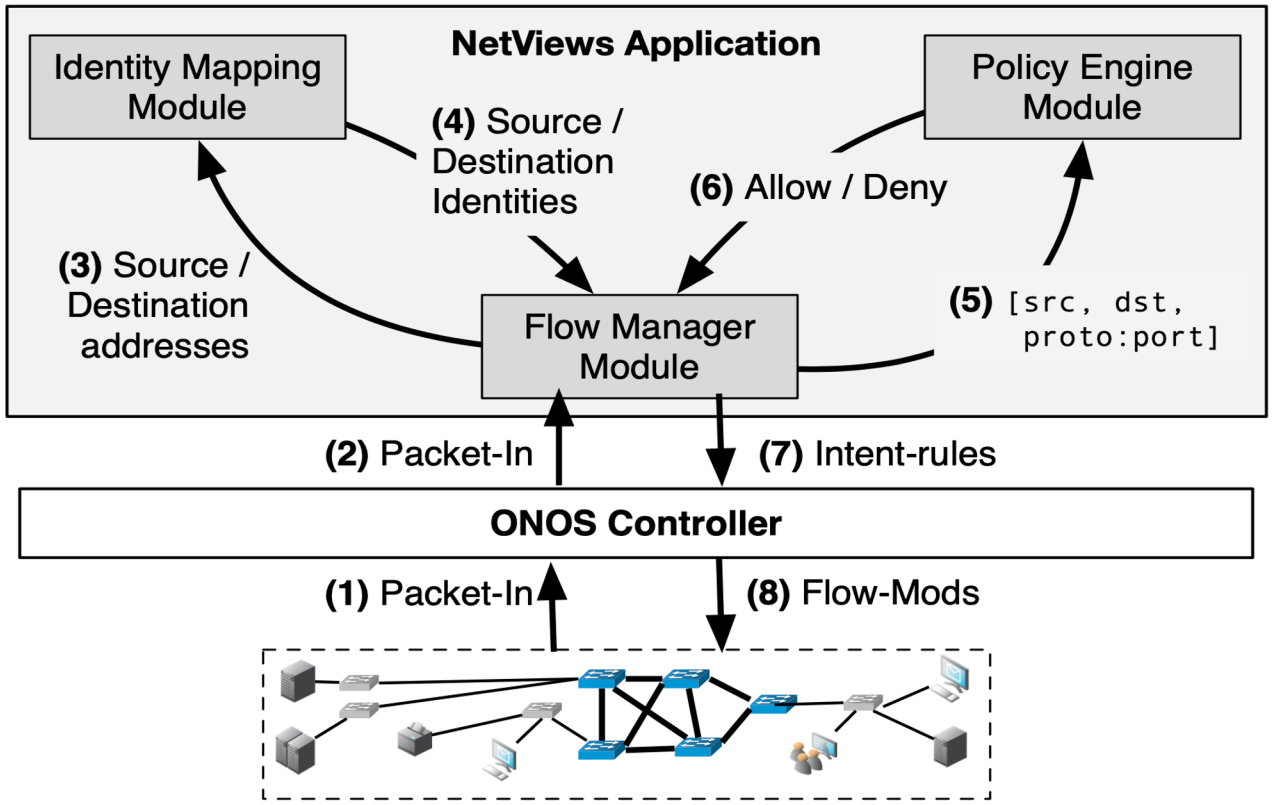


Multi-connection optimization

- Enforcement semantics are more strict than required
- Network probing does not matter if I connect to you first (and the policy allows it)
  - **Result:** reverse flow rule can allow **any** client port
  - Significantly reduces access checks **and** TCAM needed in switches



NetViews implementation



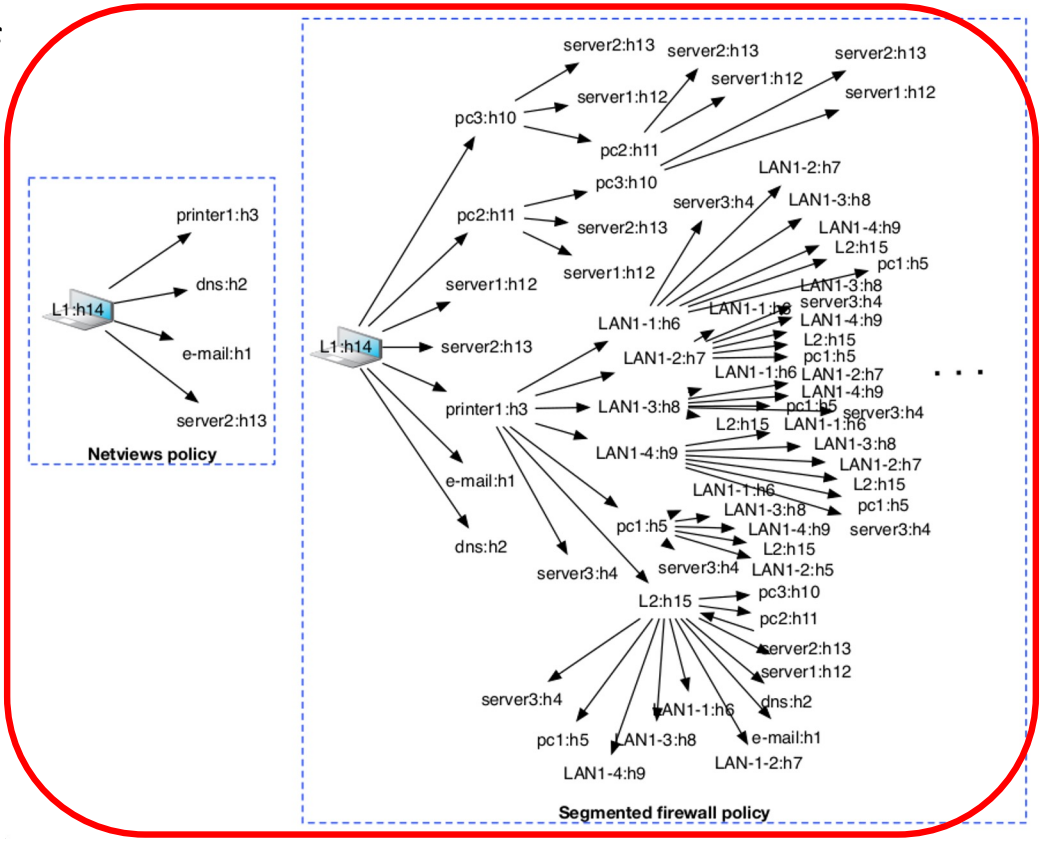
- Built as an ONOS application on top of "Intent" abstraction
- Policy engine follows the reference implementation of NGAC
- Uses a static identity mapping

Security analysis

- Used a reachability-based attack graphs (Lippmann et al.) as basis of security analysis.
- NetViews **drastically** reduces the attack surface

TABLE I: Number of hosts reachable in hop-counts 1 to 5 for the reference topology (Figure 1) based on policy type

Policy Type	hop-count	server1	server2	server3
NetViews	1	3	2	6
	2	2	0	0
	3	0	0	0
	4	0	0	0
	5	0	0	0
Segmented Firewall	1	4	4	8
	2	9	9	10
	3	9	9	10
	4	9	9	10
	5	9	9	10

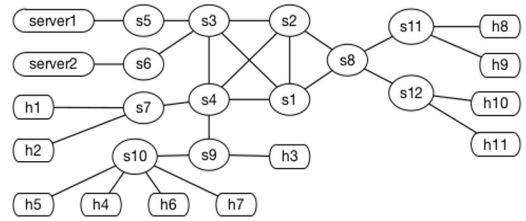


[Richard Lippmann, Kyle Ingols, Chris Scott, Keith Piwowarski, Kendra Kratkiewicz, Mike Artz, and Robert Cunningham. 2006. Validating and Restor ing Defense in Depth Using Attack Graphs. In Proceedings of the IEEE Military Communications conference (MILCOM).]

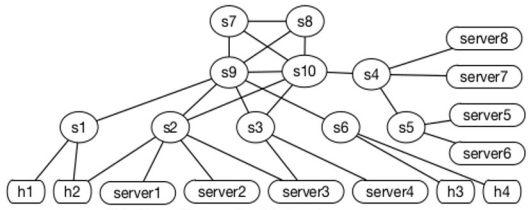
Performance evaluation

- Three ONOS applications
  - Baseline (ONOS fwd)
  - Intent Forwarding (ONOS ifwd)
  - NetViews

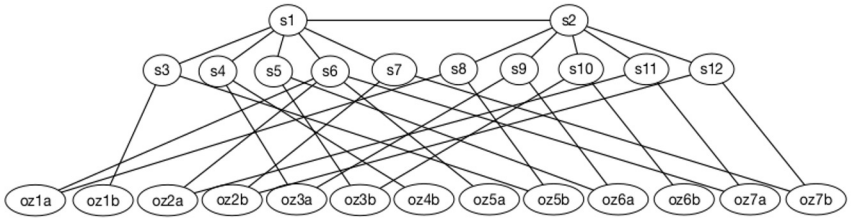
Topology	Devices	Switches	Details
Reference	13	12	Sample topology seen in Figure 1
Cisco [82]	12	10	Enterprise network with Cisco PIX firewall
MiniStanford [82], [41]	100	25	Stanford backbone network



(a) Reference Enterprise Topology



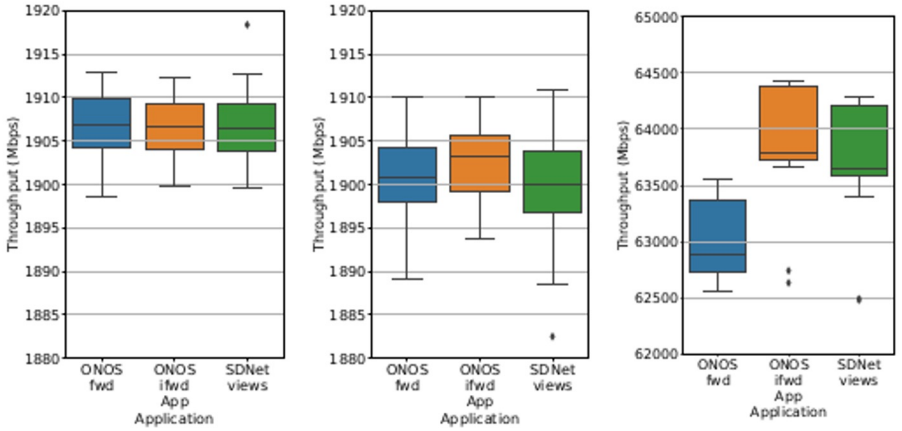
(b) Cisco Enterprise Network



(c) Stanford Backbone Network (Hosts and servers are assigned randomly with the leaf nodes)

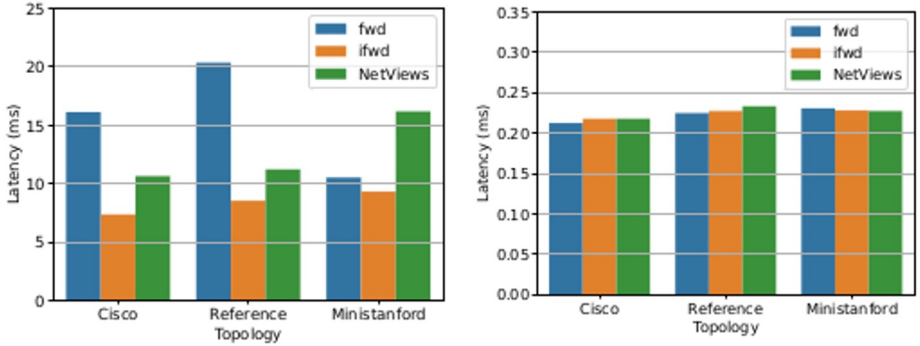
Performance overhead

Throughput



(a) Cisco (b) Reference (c) MiniStanford

Latency

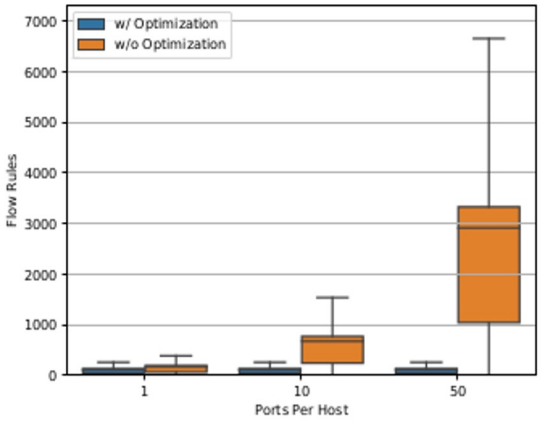


(a) Average initial packet latency (b) Average n^{th}-packet latency

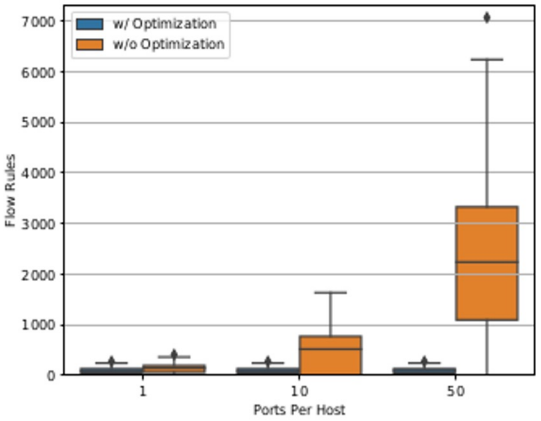
NetViews does not show any significant throughput overhead over the fwd or ifwd applications.

NetViews has acceptable latency compared to ifwd for both the initial and n^{th}-packet

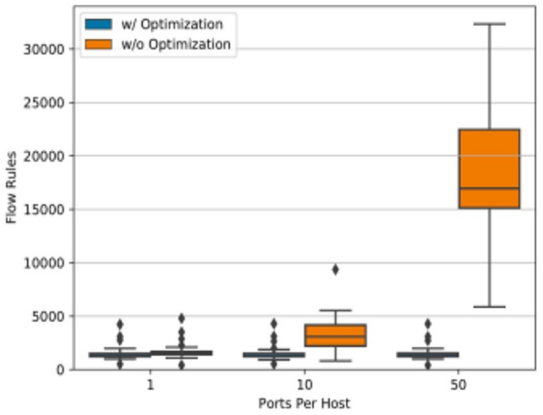
Scaling with number of flows



(a) Cisco



(b) Reference



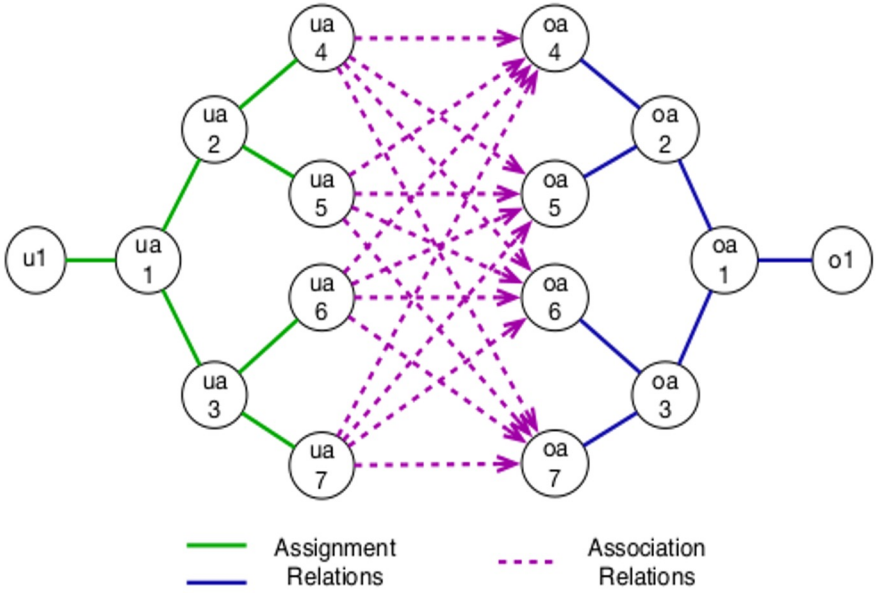
(c) MiniStanford

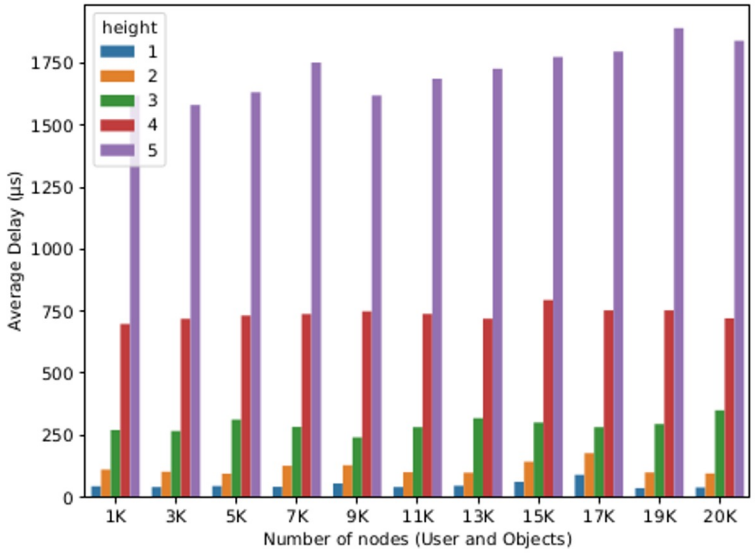
(scales differ for readability)

The multi-connection optimization can result in significantly fewer Flow Rule

(e.g, with 50 connections per session, the number of Flow Rules per switch without optimization is 2,916, while with optimization, its 116)

Performance of policy engine





- Used random policy graph generation algorithm from Basnet et al.
- Netviews overall average delay is minimal, even for the 20,000 node (u and o) benchmark, with 1,280,000 graph vertices (u, o, ua, and oa).

R. Basnet, S. Mukherjee, V. M. Pagadala, and I. Ray, "An efficient implementation of next generation access control for the mobile health cloud," in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, 2018, pp. 131–138.

Summary so far

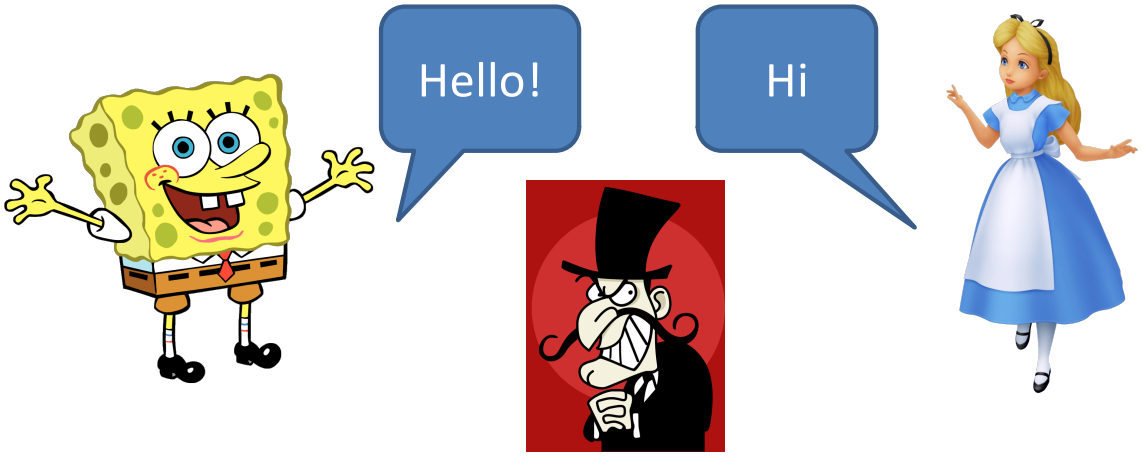
- **Proposed Network Views (NetViews for short)** as an abstraction and model for access control within enterprise networks that provides a fine-grained least-privilege network access control.
- NetViews is not a replacement for firewalls at the network edge, NetViews can be seen as a **building block** to enable Zero Trust
  - Zero Trust for on-premises network components
- Significantly reduces attack reachability graph
- Performance comparable to reactive SDN

This talk

How to implement least priviledge access control on network enclaves

How to design secure communication primitives that do not assume computationally-bounded adversaries

Secure communication



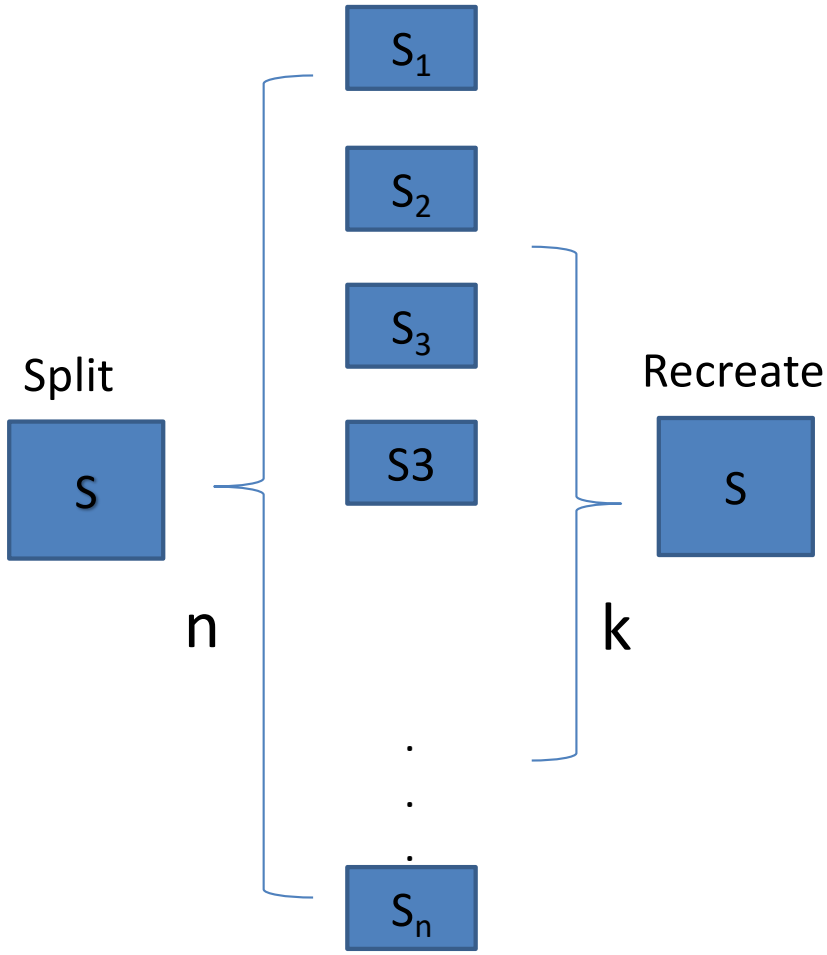
- Establish a secure and authenticated ...nication channel using standard protocols such a... QUIC
- Security guaranteed by cr...phic primitives that assume computationally-bo...dversary

Quantum computing

Secret sharing

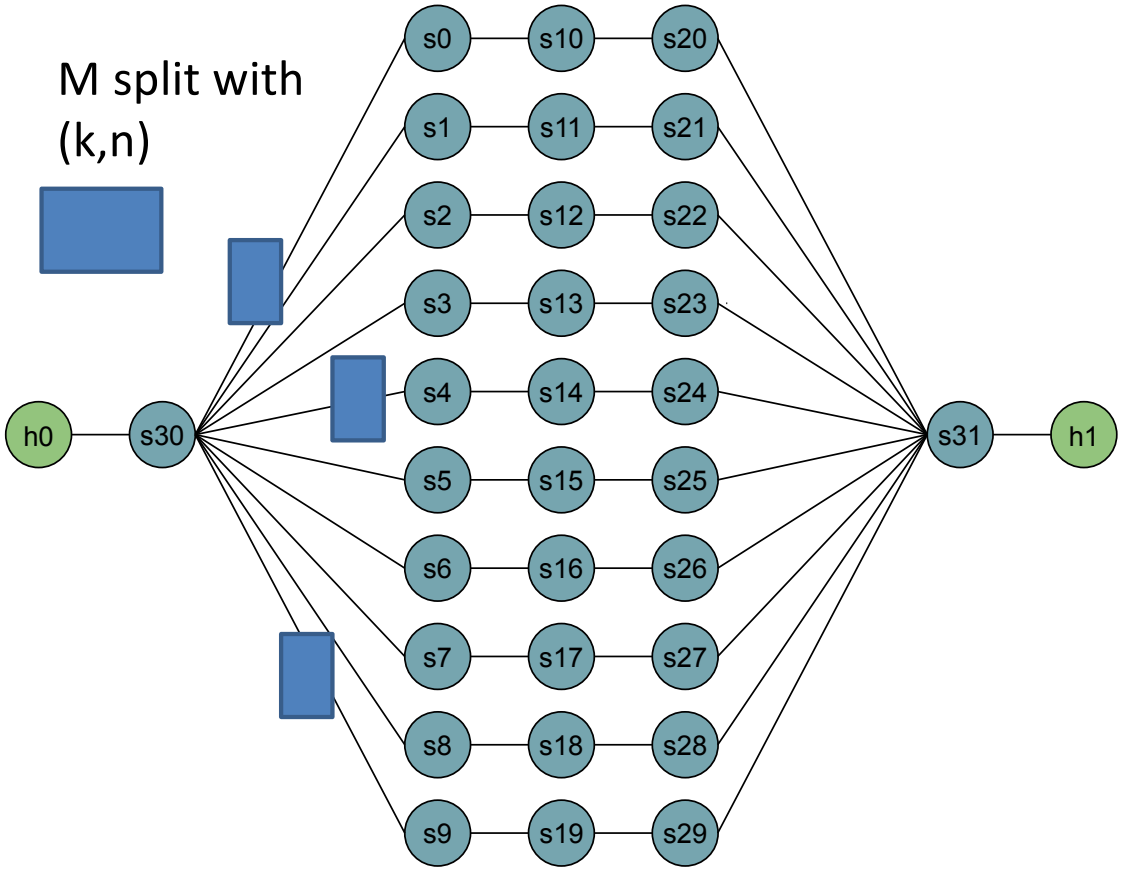
A. Shamir. *How to Share a Secret. 1979*

- Split and recreate a secret between participants that do not trust each other
- A (k, n) scheme for secret S:
  - Divide S into n pieces $s_1, \ldots s_n$
  - Any group of k or more users can jointly obtain S
  - Any group of k-1 or less users can not jointly obtain S
- Security: Secure as long as the adversary does not capture more than k-1 shares



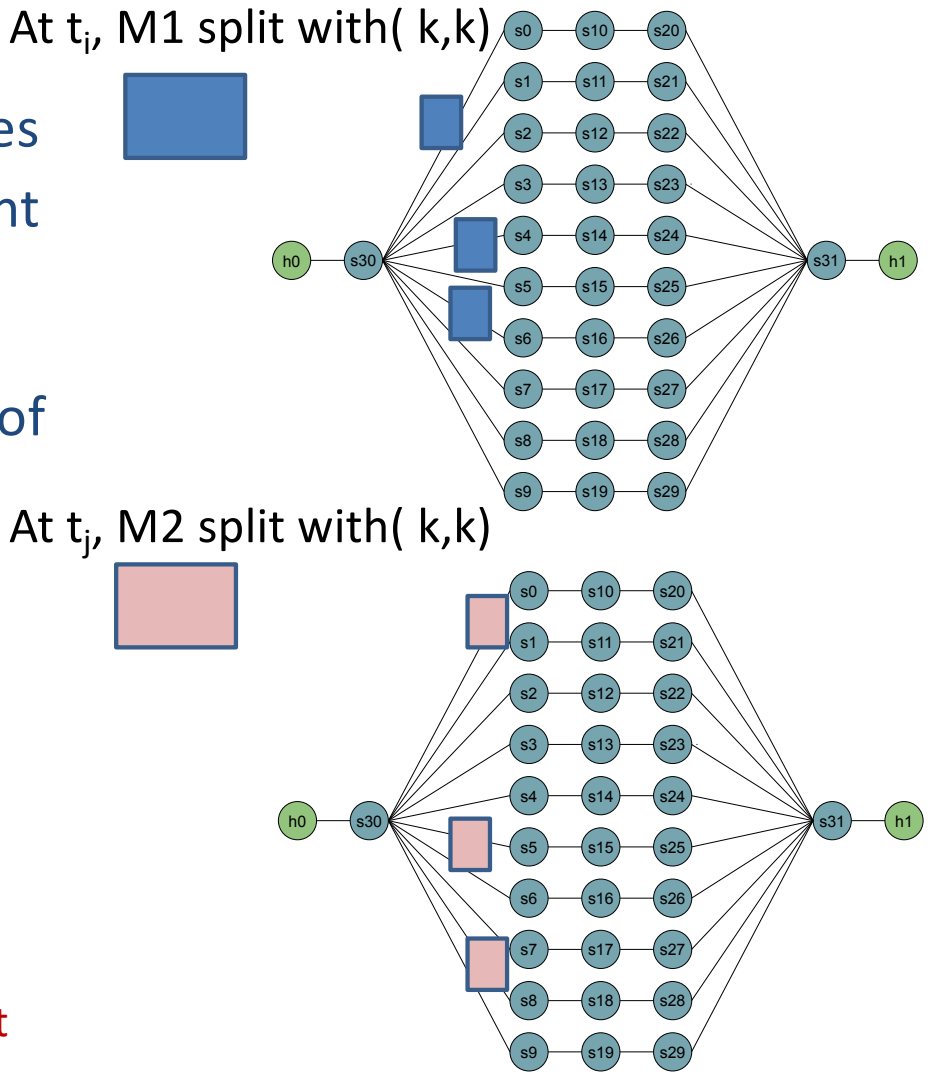
Multi-path and secret sharing

- Message split with secret sharing and sent the pieces on disjoint paths
- Security:
  - The message remains perfectly secret as long as the adversary can access at most k – 1 paths
  - Adversary bounded in terms of network access; does not know/observe ALL the paths



Multi-path switching with secret sharing (MSSS)

- MSSS (k,n):
  - Sender splits the message in k shares
  - Sender sends the shares on k disjoint paths (out of the possible n)
  - Sender and receiver *switch* to a randomly selected set of paths out of the total set of n paths
- Security: It provides information-theoretic security against an adversary with access to a quantum computer



R. Safavi-Naini, A. Poostindouz, and V. Lisy, "Path hopping: An MTD strategy for quantum-safe communication," in ACM Workshop on Moving Target Defense, 2017

The problem

~~Are implementations of multi-path switching practical (what is the cost of randomization)?~~

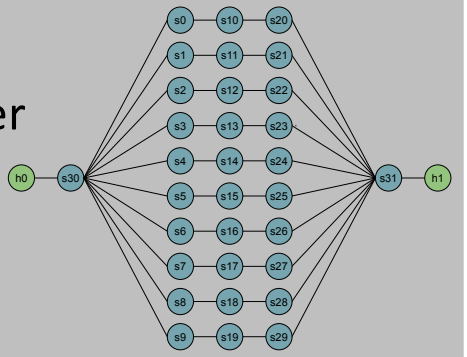
Are implementations of multi-path switching with secret sharing schemes secure?

Identify a side-channel (**Network Data Remanence**) and attacks exploiting it (**NDR Blind and NDR Planned**)

Multi-path switching with secret sharing

System

Network:
There are n disjoint paths known by sender and receiver and connecting them



Sender:
Each clock tick i:
Selects set K_i = {k paths our of n}
Splits M using (k,k) secret sharing
Sends them on the set of paths K_i

Receiver:
Listens to all paths; thus no need for secret key

Attacker

Can not observe/access all paths
Each clock tick j
Selects set K_j = {k paths out of n}
Accesses K_j to recover shares

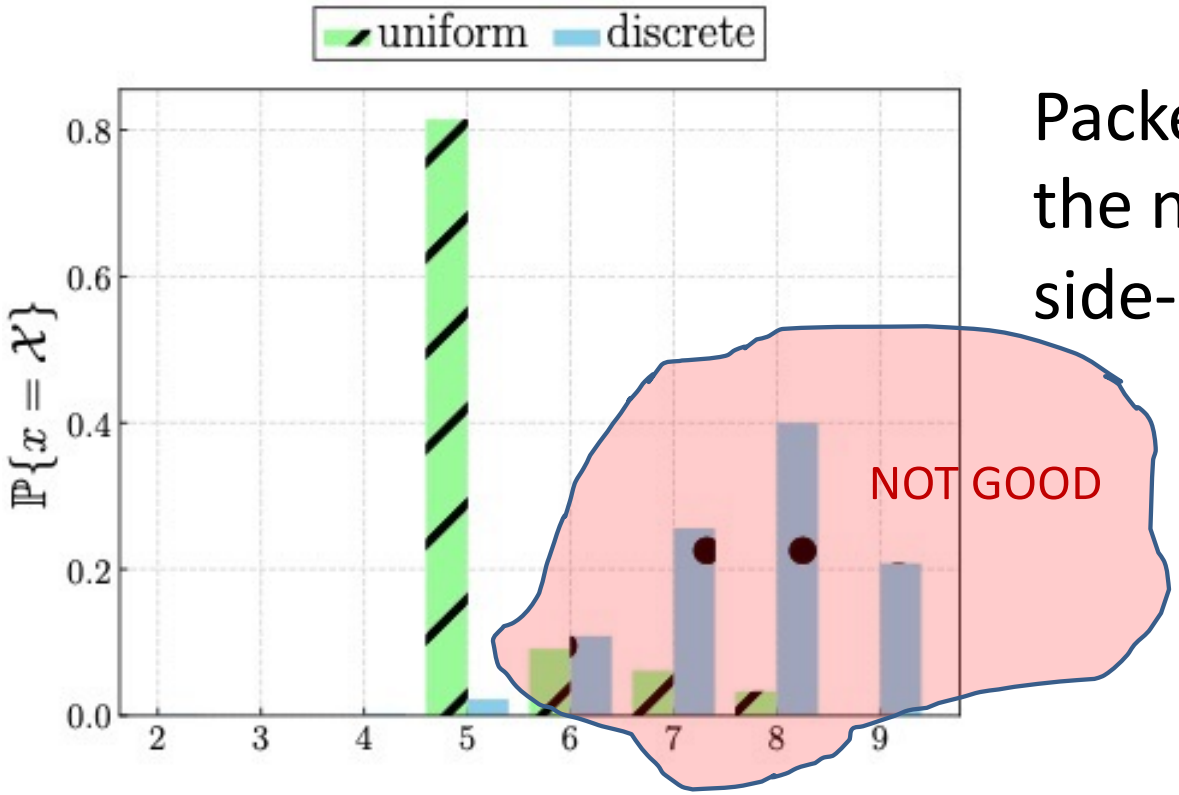
Attacker switch clock can be the same or not with the one of the sender

Security

It provides information-theoretic security and remains secure against an adversary with access to a quantum computer

Network Data Remanence Side-Channel (NDR)

(5, 9) scheme, showing active paths – paths that have ongoing packets



Packets linger longer in the network creating a side-channel

Can an attacker exploit this side-channel?

Attacker capability

- Attacker captures packets at nodes
  - Has access to all of the nodes, but they cannot possibly capture traffic from all of them at all times.
  - can only capture traffic at a fraction of nodes at each time.
- Attacker is able to listen to at most K nodes simultaneously (K is number of paths used by MSSS)
- Attacker can switch what paths they are listening to and at what intermediate nodes
- **Attacker chooses nodes, and can decide to stay on same path and select a node on the same path**

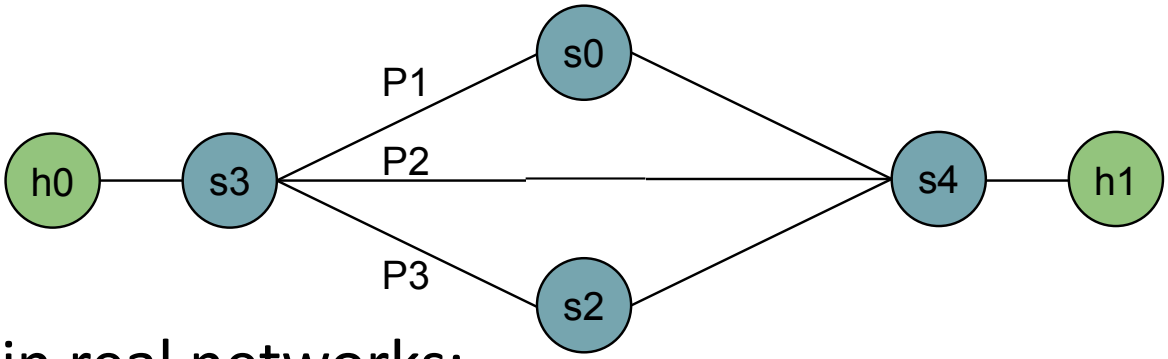
Network Data Remanence attacks

Attacker strength

- **NDR Blind**: selects K nodes from all nodes on all paths
- **NDR Planned**: follows shares as they travel along the paths in the network
  - Listens to K random nodes of distance 1 from the sender
  - Probes K random nodes of distance 2 from the sender during the second switching interval
  - and so on ….
- **NDR Planned Opt**: checks at each step to see if all shares needed to reconstruct a message are captured
  - Starts at distance 1, instead of continuing with next hop

Assumptions not met by real networks

- Theoretical security based on well-known physical layer model which assumes that paths have same length and delay



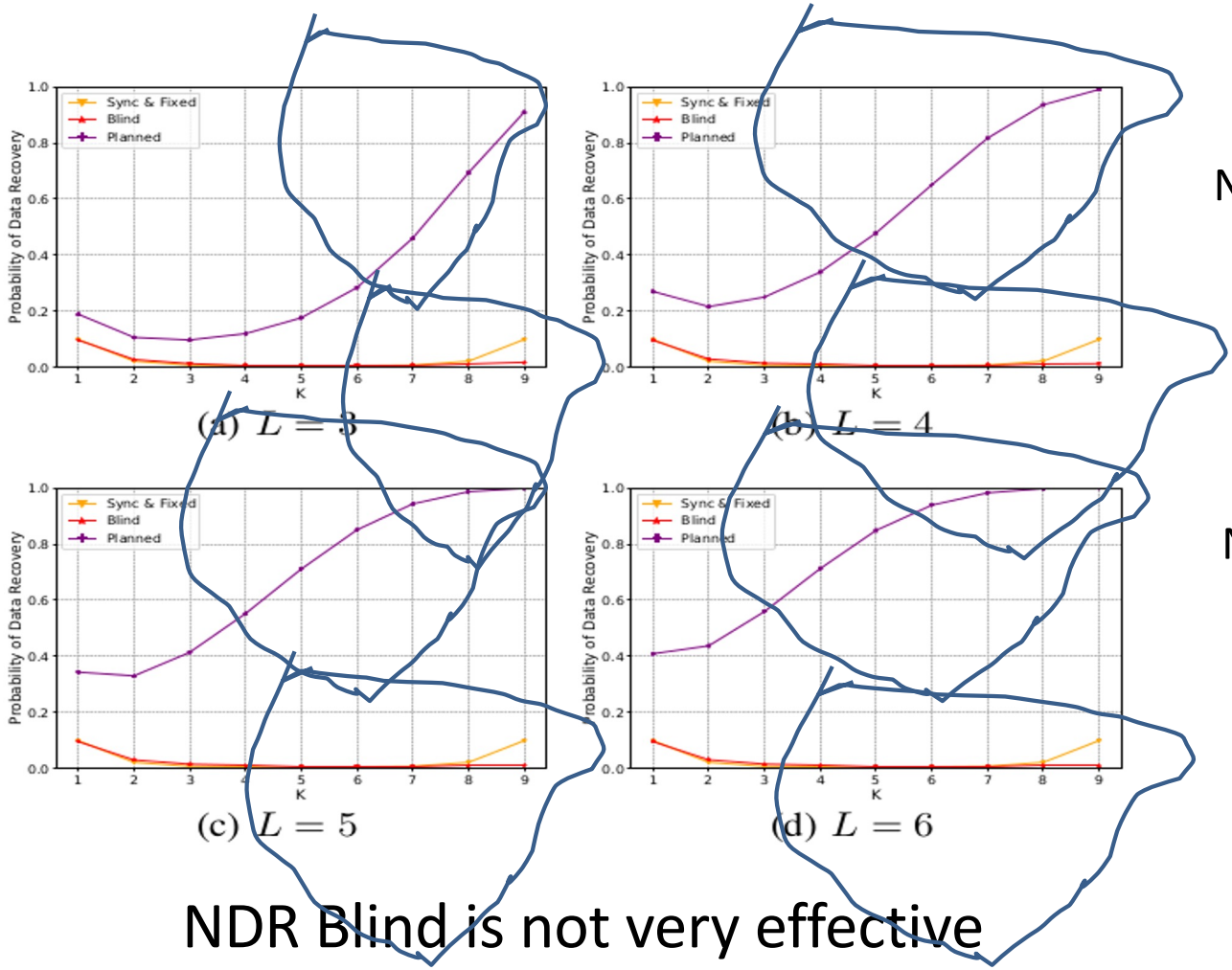
However … in real networks:

- Paths do not have the same number of hops
- Links (and paths) do not have the same delay

Attacker gets more chances at capturing a share on a path (than assumed by the model)

Probability of data recovery by the attacker

L: path length
K: # shares
N: # paths,10



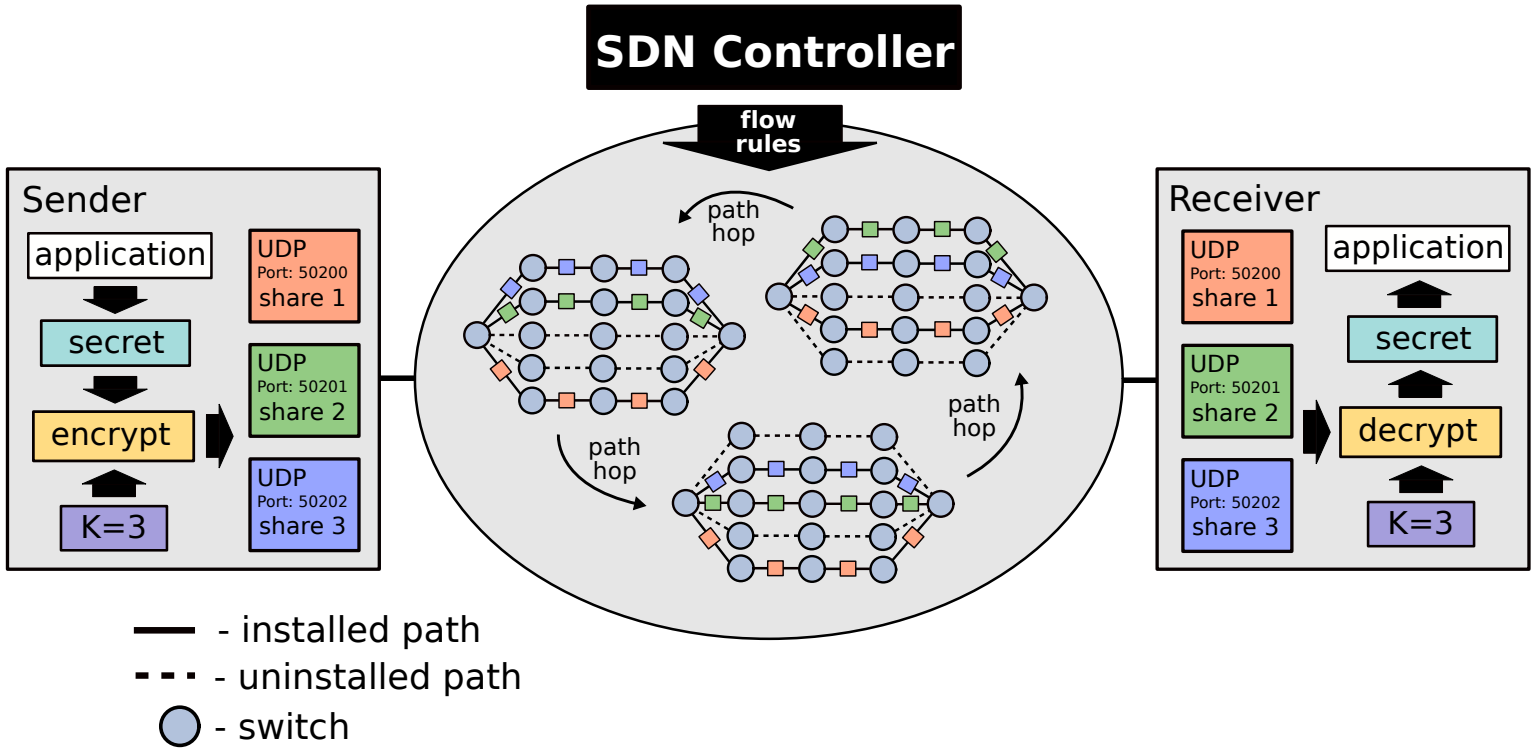
NDR Planned

NDR Blind

NDR Blind is not very effective

NDR Planned attacker is very effective

MSSS SDN-based design

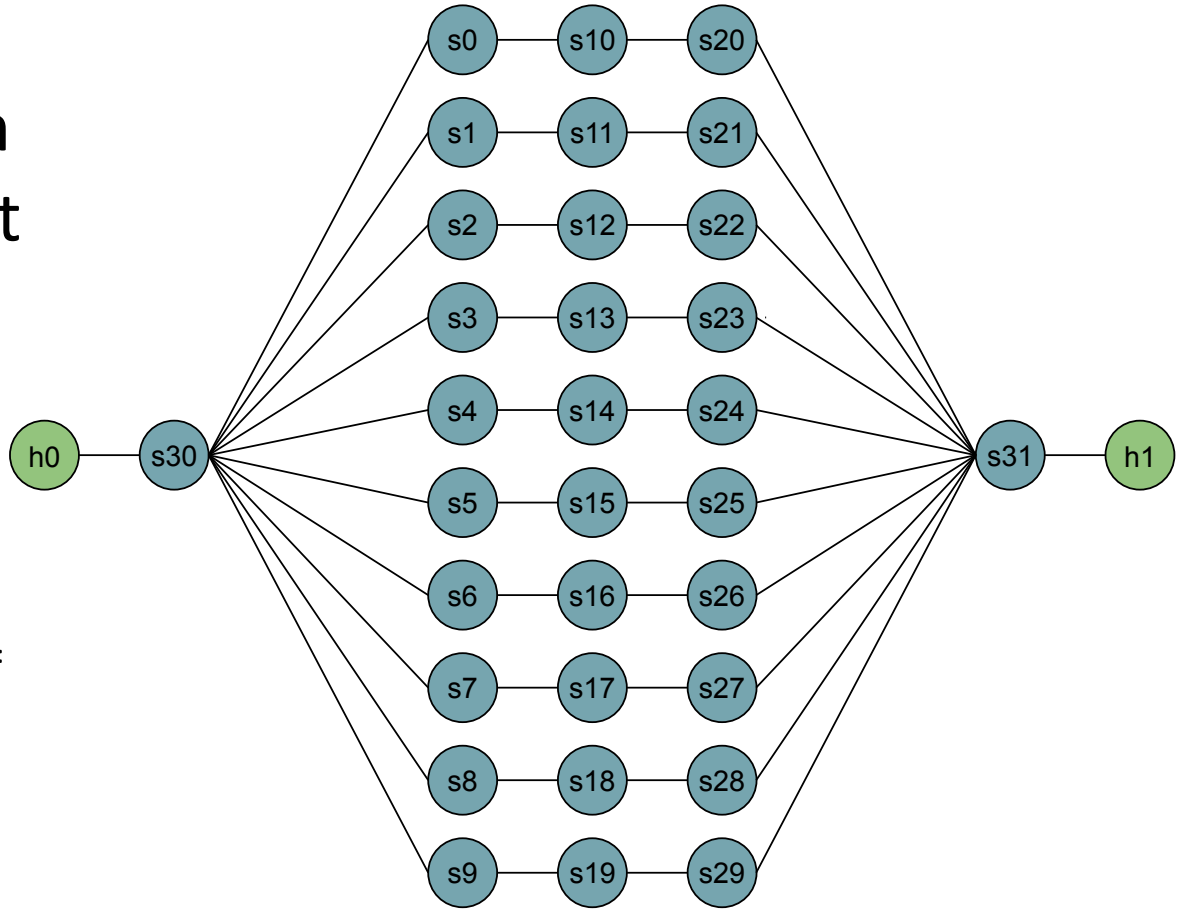


- UDP ports are used to distinguish between paths
- Receiver listens to all paths

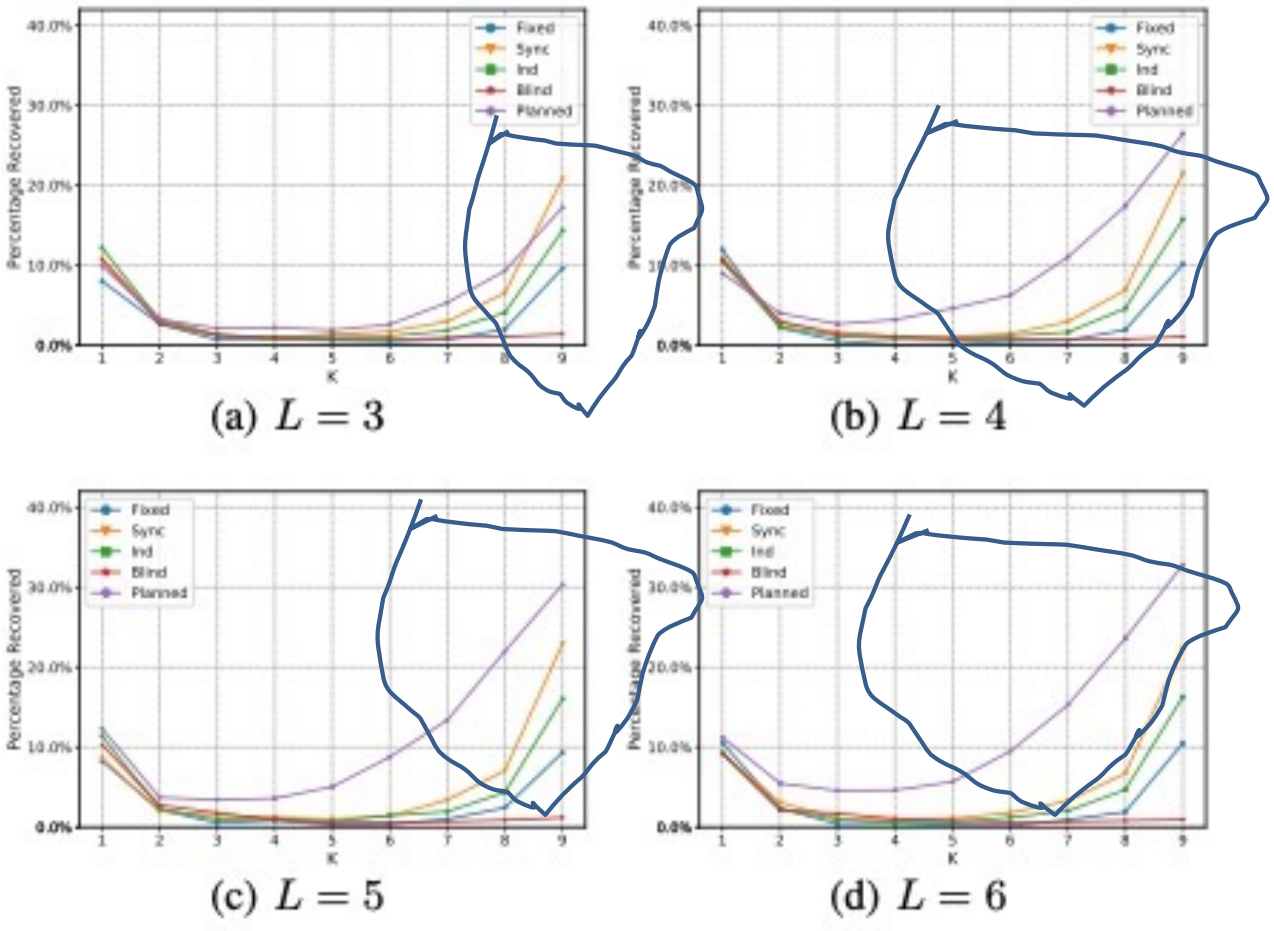
Experimental results

- MSSS implemented with ONOS and Mininet

- N = 10
- L = 4
- K = 3
- Path switching interval δ = 100 ms
- File size = 10MB
- M = 512B



Impact of path delay



NDR Planned attacker is very effective in SDN –based implementation

How to mitigate the attacks?

We want to keep information theoretic security

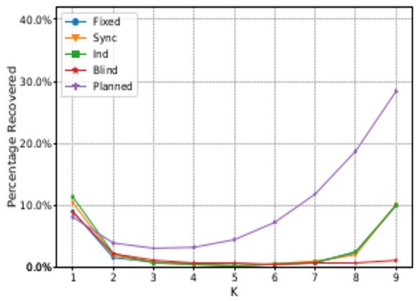
Break the message into more shares

- How to send these shares:
  - Use more disjoint paths – need to also increase the attacker power to be fair
  - Use the same K paths repeatedly -- could result in reduced protection
- Our approach: distribute shares over both ***time*** **and** ***space*** instead of just space using a random set of paths to send a K-sized set of shares

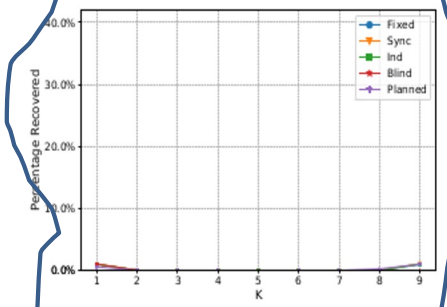
Our mitigation

- Generate more shares and spread them across both space and time
- Instead of (K, K), the sender uses (HK, HK) secret sharing
  - divide the shares into H sets of K shares
  - send these sets of shares, one at each consecutive clock tick
  - at t = 0, 1, . . . , H − 1, the sender chooses K paths uniformly at random, and then sends a share along each chosen path
- We call H *resilience factor*, a system parameter that can be configured by the sender

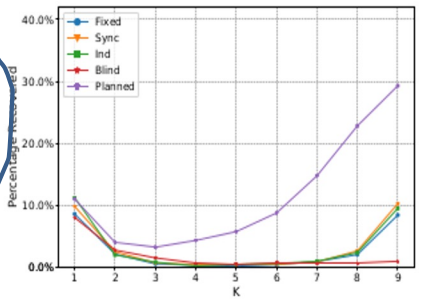
Experimental results: Probability data recovery



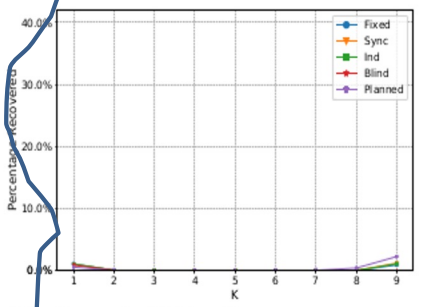
(a) $L = 3$, No Countermeasure



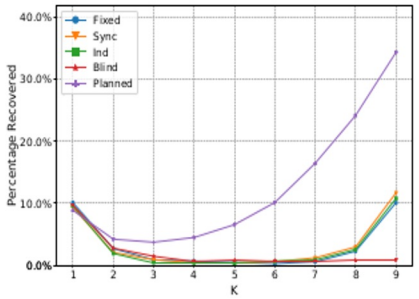
(b) $L = 3$, With Countermeasure



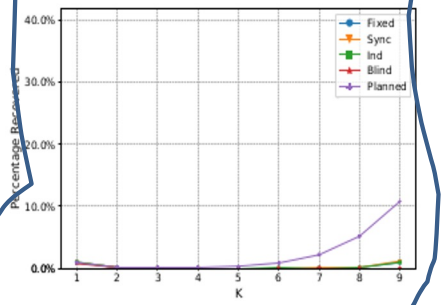
(c) $L = 4$, No Countermeasure



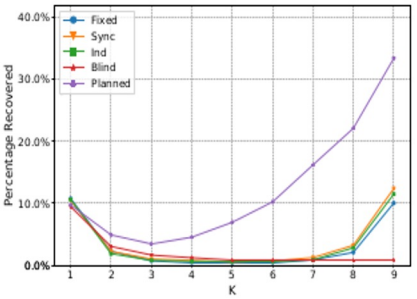
(d) $L = 4$, With Countermeasure



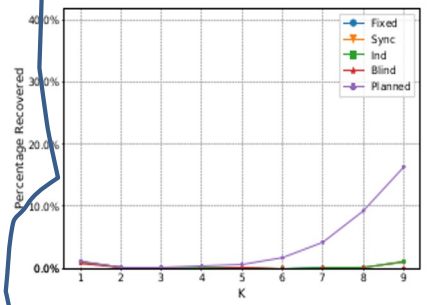
(e) $L = 5$, No Countermeasure



(f) $L = 5$, With Countermeasure



(g) $L = 6$, No Countermeasure



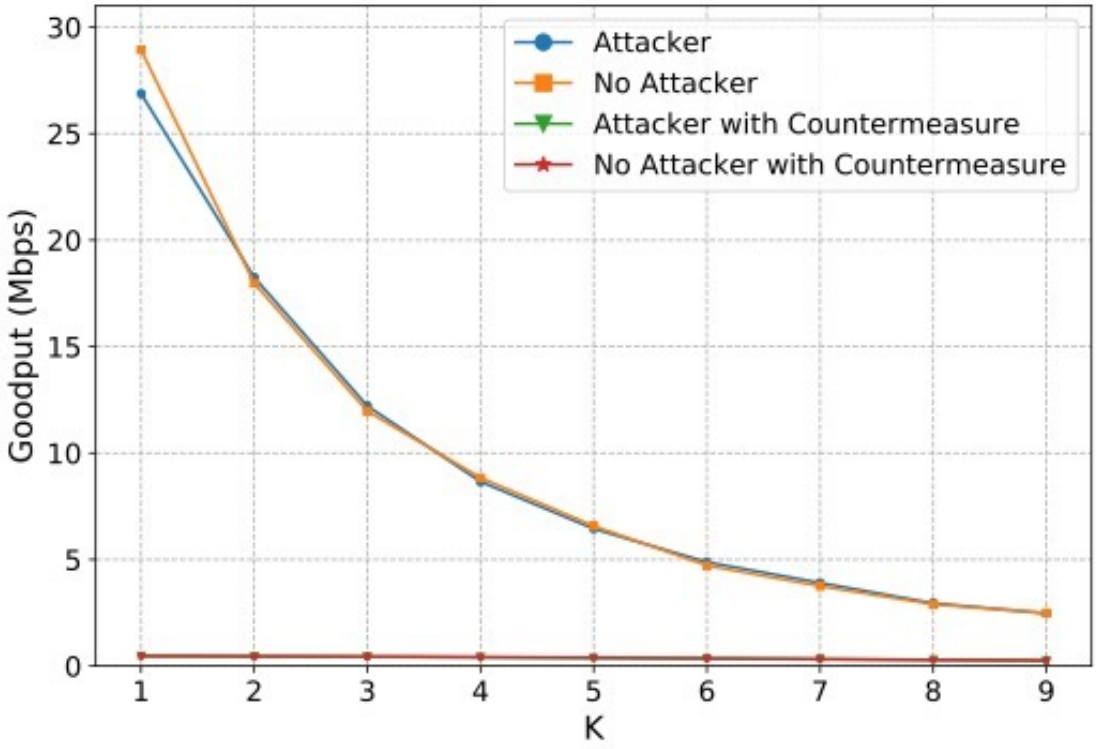
(h) $L = 6$, With Countermeasure

Effect of the countermeasure and number of shares on percentage of recovered data with varying path length. Fixed scenario with 2 *ms* delay between each node. The sender's $\delta = 4$ *ms* and the attacker's $\delta = 8$ *ms*. File size and the resilience factor, H, are set to 1 *MB* and 2, respectively.

Countermeasure mitigates the NDR Planned attack in SDN-based implementation

Experimental results: Goodput

L = 3, H = 2, sender's δ = 4 *ms*, *a*ttacker's δ = 8 *ms*.



Increasing the number of shares, and spreading them through time, has a significant impact on performance

Summary so far

- Analyzed secure communication schemes that do not make computational assumptions about the attacker
- Identified a side-channel Network Data Remanence and analyzed and demonstrated attacks that exploit it in a SND-based implementation of MSSS
- Proposed a countermeasure, analyzed and demonstrated in the same SDN-based implementation

What lessons have we learned from these projects?

Lessons learned (1)

Data for realistic enterprise network remains a challenge

- There are very few available realistic network enterprise topologies, the ones we had access too are limited and relatively small
- Same goes about access control policies, the ones available are small and not very complex

Lessons learned (2)

SDN is not sufficient for complete solutions

- SDN operates in a network eco-system that relies on other protocols for network topology discovery and authentication of devices
- There is a need for secure identity mapping between devices and IP addresses

Lessons learned (3)

Importance of implementation and experiments in the security evaluation of cryptographic systems

- We discovered an implementation side-channel that must be taken into account in the implementation of secure message protocols
- Our work has led to the discovery of NDR side channels in other protocols

Lessons learned (4)

Discovery of side-channels lead to ensuing new requirements

- Implementation of computational and quantum cryptographic systems have led to the discovery of side-channels and ensuing new requirements (e.g. protection against timing channels)
- Our mitigation strategy against the NDR side-channel is the first step towards protecting against this side channel, and is at the cost of significantly lowering the system information rate

Acknowledgments

- University of Calgari:
  - Leila Rashidi, Majid Ghaderi, Reihaneh Safavi-Naini, Alexander James, University of Calgary
- MIT Lincoln Labs:
  - Samuel Jero, Hamed Okhravi
- NC State:
  - Iffat Anjum, Rajit Bharambe, Will Enck, Brad Reeves,
- Northeastern University:
  - Daniel Kostecki, Ethan Leba, Anthony Peterson, Jessica Sokal,

Publications

- **Removing the Reliance on Perimeters for Security using Network Views.** Iffat Anjum, Daniel Kostecki, Ethan Leba, Jessica Sokal, Rajit Bharambe, William Enck, Cristina Nita-Rotaru, and Bradley Reaves. ACM SACMAT 2022. Best student paper award. Code: https://github.com/netviews/ss-netviews.
- **More than a Fair Share: Network Data Remanence Attacks against Secret Sharing-based Schemes.** Leila Rashidi, Daniel Kostecki, Alexander James, Anthony Peterson, Majid Ghaderi, Samuel Jero, Cristina Nita-Rotaru, Hamed Okhravi Reihaneh Safavi-Naini. NDSS 2021.
- **The Tale of Discovering a Side Channel in Secure Message Transmission Systems.** Majid Ghaderi, Samuel Jero, Cristina Nita-Rotaru, Hamed Okhravi, and Reihaneh Safavi-Naini In CFAIL 2022, with Crypto 2022.
- **On Randomization in MTD Systems.** Majid Ghaderi, Samuel Jero, Cristina Nita-Rotaru, and Reihaneh Safavi-Naini. MTD 2022, in conjunction with ACM CCS 2022.