More than a Fair Share: Network Data Remanence Attacks against Secret Sharing-based Schemes

By

Leila Rashidi, Postdoctoral Associate, Department of Computer Science

In collaboration with

Daniel Kostecki, Alexander James, Anthony Perterson, Majid Ghaderi,

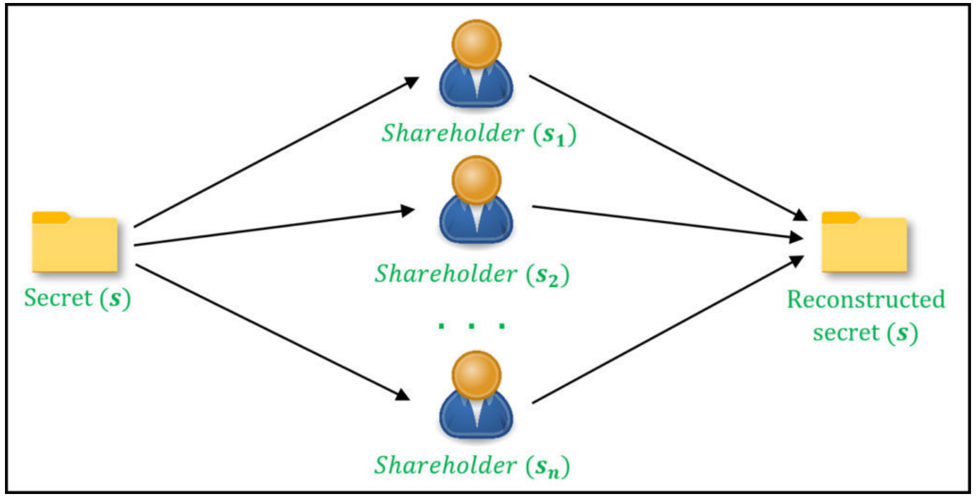
Samuel Jero, Cristina Nita-Rotaru, Hamed Okhravi, and Reihaneh Savafi-Naini

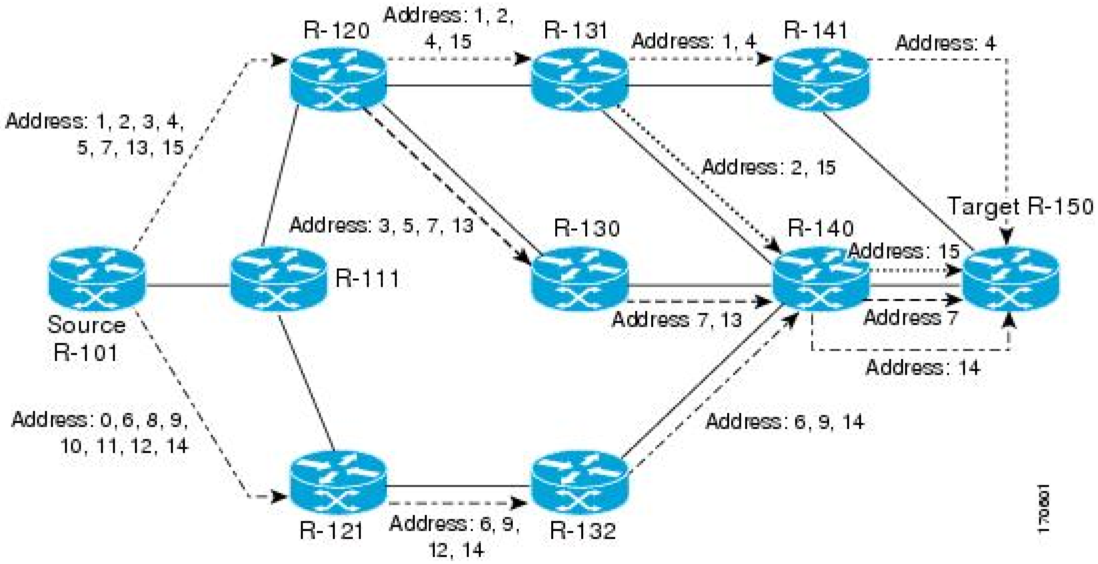
In Proceedings of

Network and Distributed System Security Symposium 2021

Outline

- Introduction
- Background
- Network Data Remanence Attack
- Initial Evidence
- Threat Model
- Considered Attacker Types
- Analytical Results
- Experimental Results
- Proposed Countermeasure
- Effectiveness of the Countermeasure
- Conclusion





Introduction

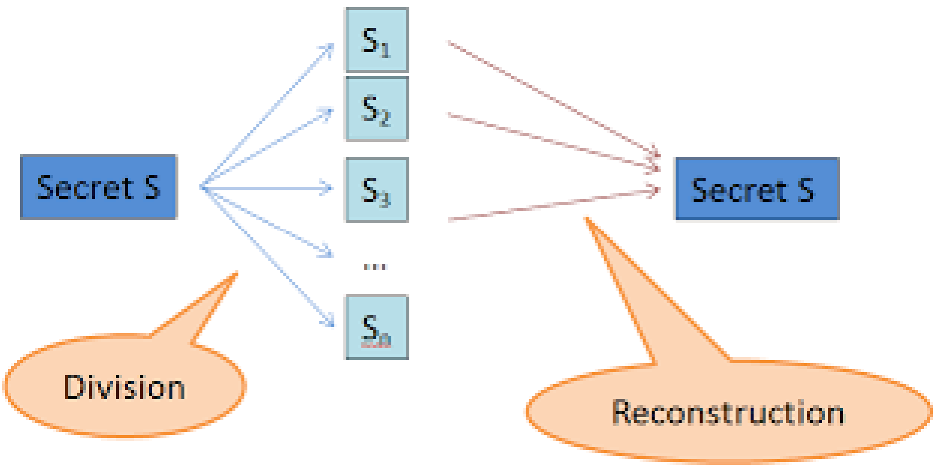
- **Untrusted network:** Improper access to sensitive or personal data may be possible.
- **Q:** How to achieve secrecy and integrity over an untrusted network?
  - **Common Approach:** Using standard protocols such as ***TLS*** to establish a secure and authenticated communication channel.

  What's the problem?

  - ➢ Although the security of such standard protocols is predicated on several assumptions, but the **validity** of these assumptions in real-world have been **undermined** by several challenges.
  - ➢ **Example:** Low-resourced devices (*e.g.*, IoT devices) often do not have the computational power to implement the standard protocols
- Thus, novel communication protocols have been proposed that
  - Use **physical properties** such as ***existence of multiple network paths*** *between sender and receiver*
  - Introduce **dynamism** in the system to stay ahead of an adversary which is trying to guess what paths are used for communication
- ***Goal of this talk:*** Examination of the real-world security of schemes that combine Secret Sharing with Multi-Path Routing.

Background

- Secret Sharing: A fundamental building block in
  - secure multiparty computation
  - distributed storage
  - side channel protection
- *(t,n) threshold secret sharing scheme* uses
  1. A **randomized share generation algorithm:** Takes a message *m* and generates *n* shares
  2. A **deterministic reconstruction algorithm:** Takes any *t* shares and reconstructs the message *m*



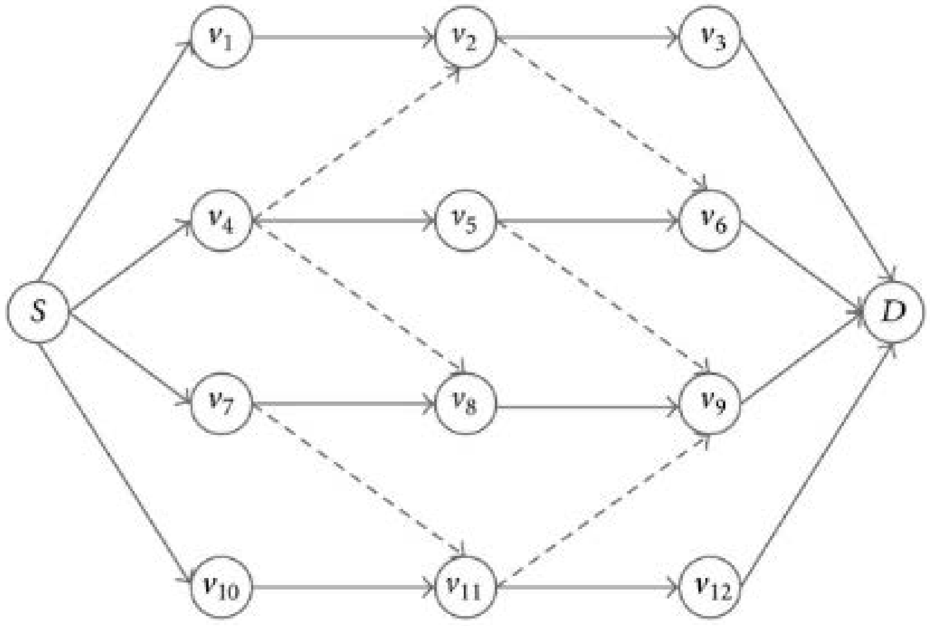
- **Security property** of (*t*,*n*)-Secret Sharing:

  Any $t-1$ shares do not reveal any information about the message. That is, the message will be ***perfectly (information theoretically) secure*** if the adversary can have access to at most $t-1$ shares.

Background: Multipath Routing and Path Switching

- **Multipath routing:** Using multiple paths rather than sending whole traffic along a single path
- **Related work:**
  - Approaches using both Secret Sharing and Multipath Routing (sending each share along a distinct node-disjoint path)
    - **Vulnerability:** Fixed set of paths

      The adversary can **infer** the set of paths used for long flows by monitoring network activity, enabling them to *break* the security of the communication.
  - **Path Switching:** Sending each message on a random path

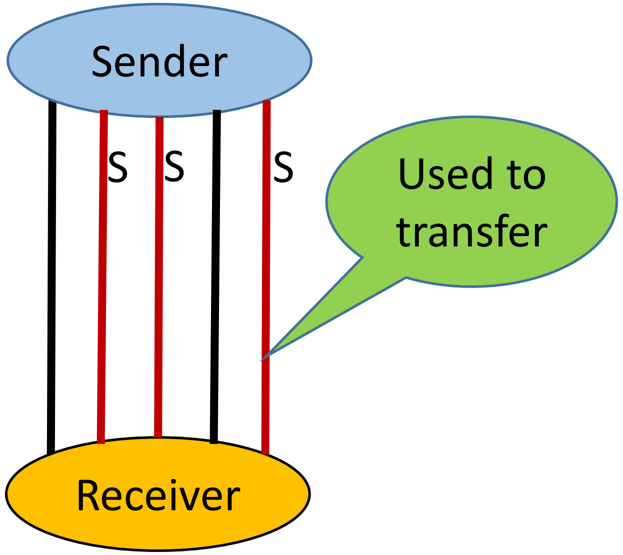


Four Node-Disjoint Paths from *S* to *D*

Multipath Switching with Secret Sharing (MSSS) Scheme

- **Why to choose MSSS?**
  - Since it was shown to have perfect information theoretic security
- **Assumptions**
  - The sender and the receiver are connected by *N* wires.
  - *K* paths can be observed by the adversary at any given time ($K < N$).
  - The adversary is mobile, and can change the paths to which listens.
- **MSSS**
  1. Generating *K* shares for each message using (*K*,*K*)-secret sharing
  2. Random selection of *K* paths
  3. Sending each share of a message along a distinct selected path

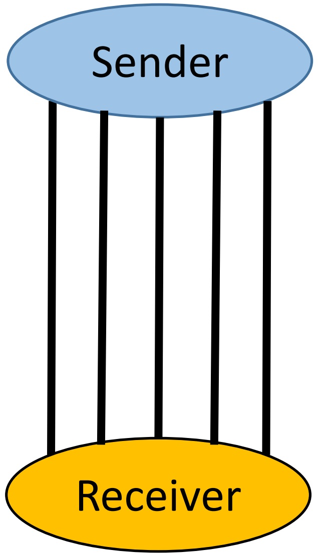
Example:
N=5, K=3



Reference: 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, pp. 111–114.

Security Analysis of MSSS

- MSSS provides information-theoretic security and remains secure against an adversary with access to a *quantum computer* if following assumptions hold
    1. Time is divided into fixed consecutive intervals such that in each interval, both sender and adversary change their sets of paths.
    2. All paths have the same end-to-end delay.
    3. Path delays are **negligible** (*i.e.*, transmissions are instantaneous).



- The second and third assumptions imply that

  the adversary have one chance to capture a share on a path.

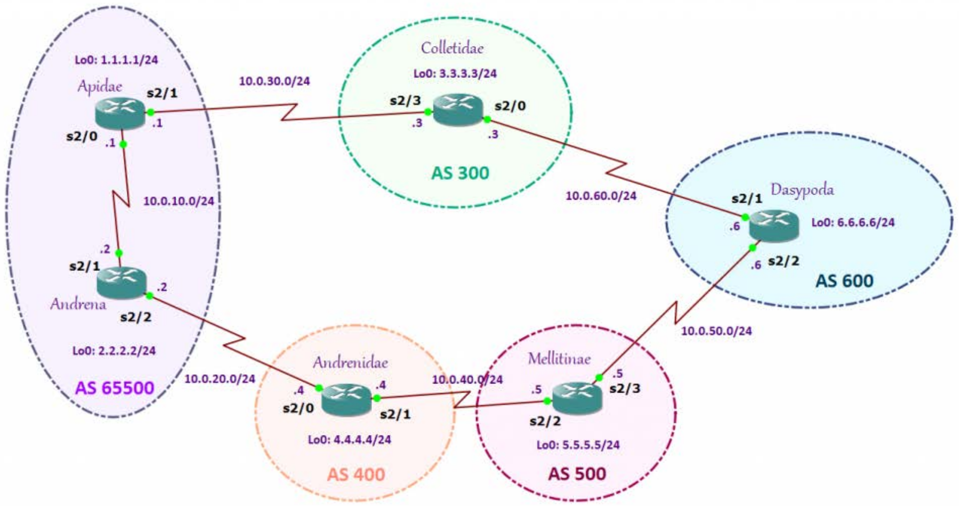
Assumptions for Security Analysis of MSSS

- Two Aforementioned assumptions
  - All paths have the same end-to-end delay
  - Path delays are negligible (*i.e.*, transmissions are instantaneous)

➢ These assumptions do not hold in real networks due to the following properties:

  ➢ Paths with multiple hops

  ➢ Hops and paths can have a different delays.



Network Data Remanence Attack (NDR)

Real Networks

- Multi-hop paths
- Different delays

→

Lingering of data in the network

→

Breaking confidentiality guarantees of Secret Sharing-based schemes

Warning:
Network Data Remanence Attack,

Attacker has more chance to collect enough shares

Note:
This name is chosen for this attack as we were inspired by data remanence side channels in storage context

Data Remanence

- **Origin:** Storage Context
- **Definition:**

  The residual physical representation of data that has been in some way erased

  (**Ref:** NSA/NCSC Rainbow Series)

- Does anyone know an example of Data Remanence?

Note:

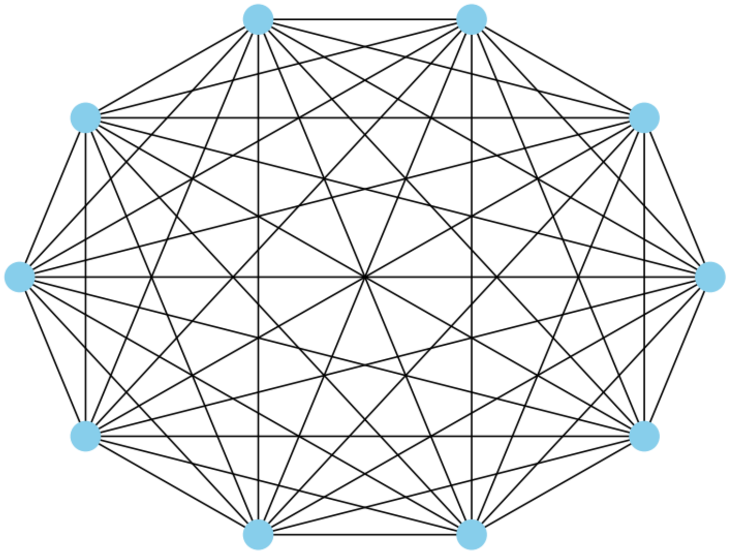
- While data remanence has been studied extensively in the context of storage media, it has received **very little attention** in the **context of networking**.
- This is the first time that **a data remanence sidechannel** has been considered outside storage systems.

What’s Next?

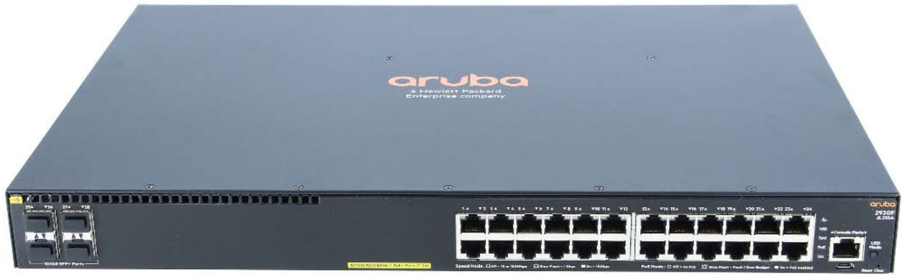
- Evidence of the NDR side-channel in real networks
- How an attacker can exploit the vulnerability introduced by NDR?
  - We identified two new attacks
- Introducing a model that captures the multi-hop nature of paths and analysing different attack strategies against MSSS
- The impact of these attacks in practical settings (Mininet)
- Countermeasure

Experiments: Testbed Setup

- A complete graph topology with 10 nodes
- ONOS SDN controller via OpenFlow 1.3
- Four Aruba 2930F switches
    - Each of the physical switches can host up to 16 distinct OpenFlow agent instances.
    - From the perspective of the controller, each OpenFlow agent instance appears as a distinct OpenFlow switch
    - Each Aruba 2930F switch includes 24 ports, each at 1 *Gbps*.

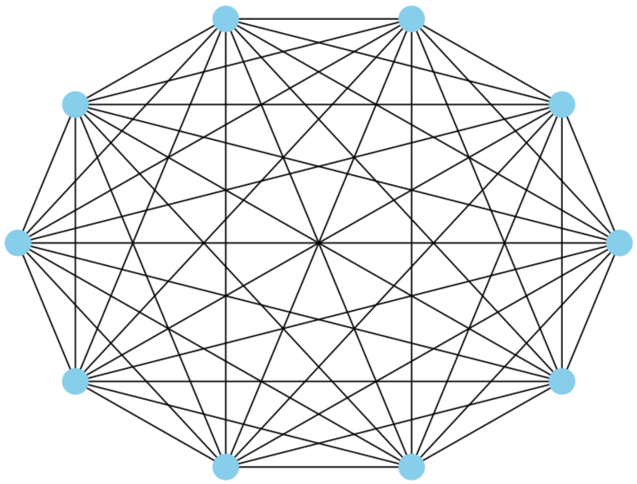


Topology



Experiments: Testbed Setup

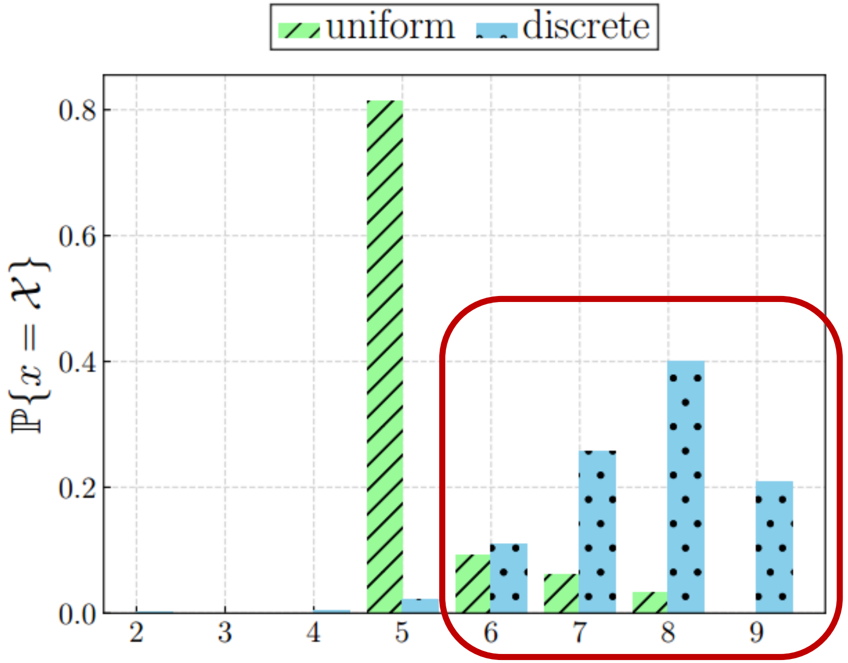
- In each experiments, there is a data transfer between two nodes using MSSS scheme.
- Bulk data transfer size: 20 *MB*
  - Divided to messages of size 256 *B*
- *N=9, K=5*
- Length of switching interval is 100 *ms*
- Two scenarios are considered for path delays:
  - **Continuous:** Uniformly selected from the range [0, 250] *ms*
  - **Discrete:** Uniformly selected from set {0, 100, 200} *ms*
- **Jitter:** The latency on an individual path could randomly vary by up to 50 *ms* for each packet transmitted over the path.



Topology: A Complete graph with 10 nodes

Initial Evidence from Testbed

Number of shares per each message, *K=5*



PDF of the **number of active paths** per switching interval where shares of any packet were present

PDF of the **number of switching intervals** where shares of the same packet were present.

Lingering of shares in the network

Threat Model:: Assumptions

1) The attacker captures packets at nodes/hops
2) the attacker has access to all switches and can redirect a copy of the traffic to their machine.
3) While the attacker has access to all of the switches, they cannot capture traffic from all of them at all times.

- ***Answer:*** that would require an unreasonably fast machine with significant resources and bandwidth (ev[illegible] speeds become an issue in that scenario), and such an attac[illegible] identifiable.

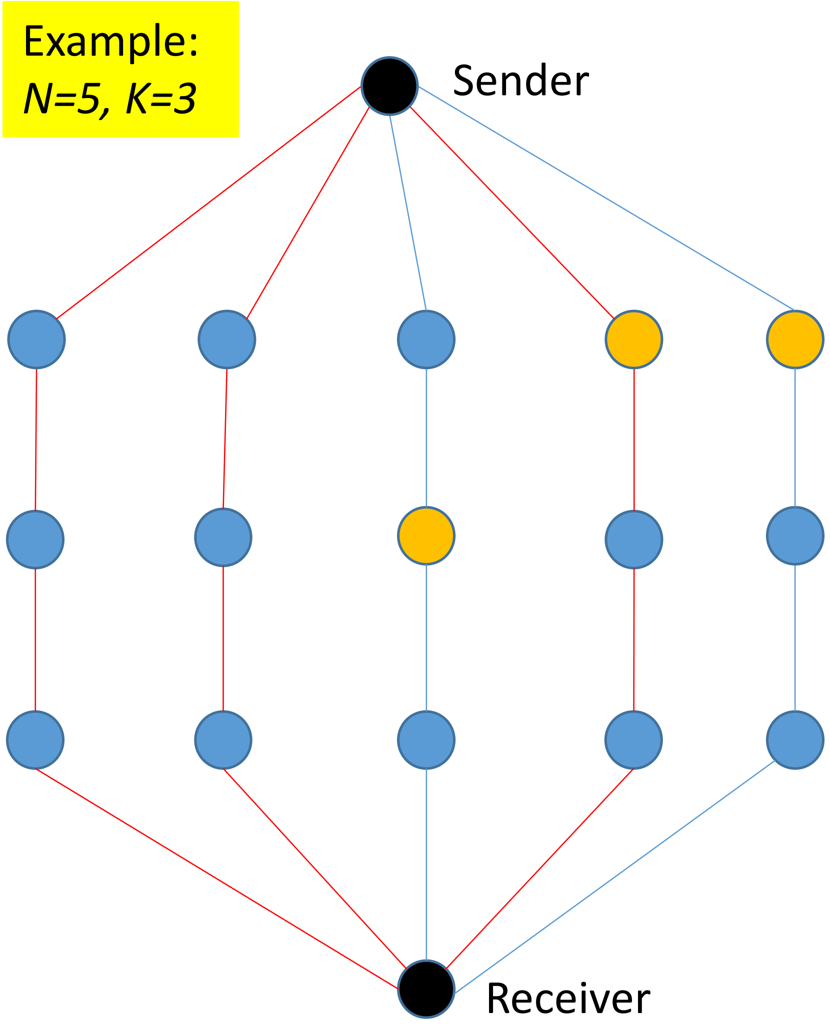
Why?

- Therefore, a realistic attacker can only
  - Listen to a fraction of switches at each time (say 10%)
  - And as a result, capture a fraction of traffic from each switch (say 10%)

Threat Model:: Assumptions (Cont.)

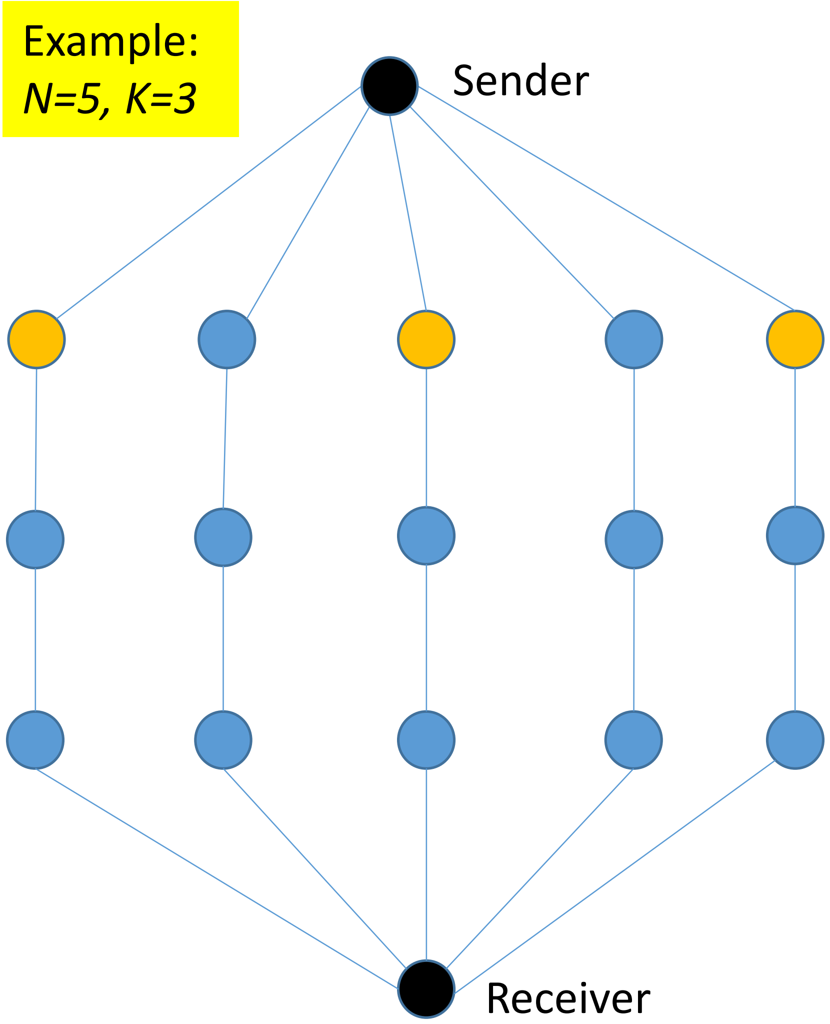
4) The attacker is able to listen to at most *K* hops simultaneously, where *K* is equal to the number of paths used to send shares of a message in *MSSS*.

5) Based on its resources, the attacker can switch what paths they are listening to and at what intermediate nodes.



Attackers

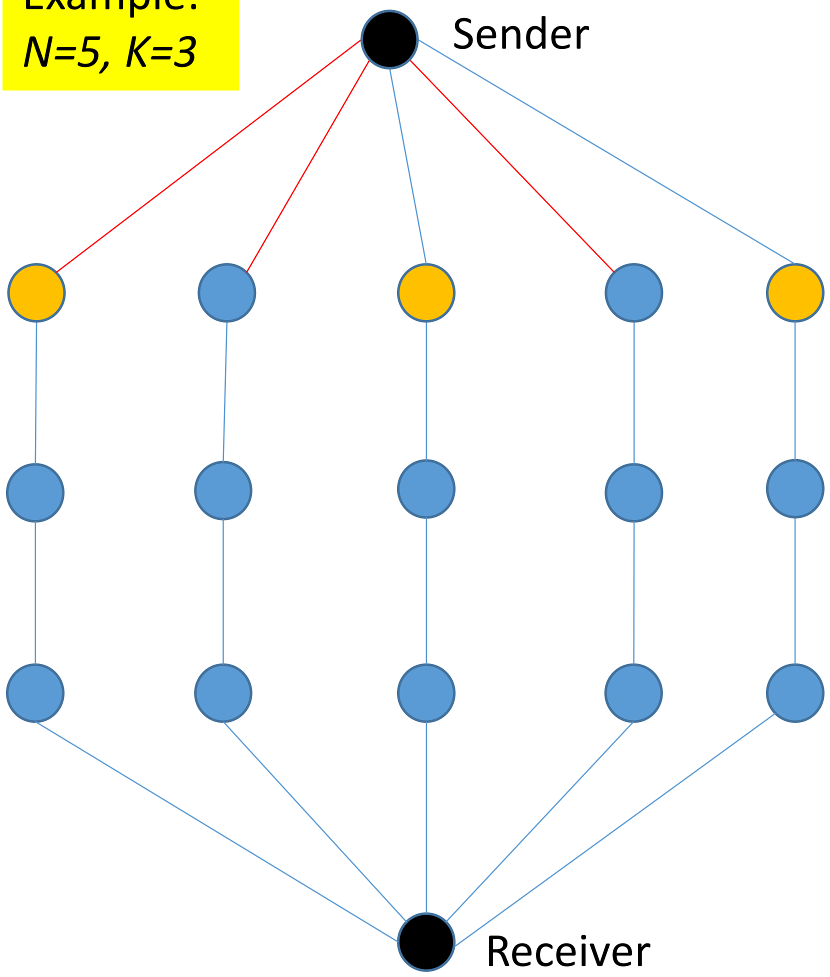
- **Basic Attackers:** listen to K distinct paths
  - **Fixed**: listens to a fixed set of K paths



Attackers

- **Basic Attackers:** listen to K distinct paths
  - **Fixed:** listens to a fixed set of K paths
  - **Synchronized:** Switches hops synchronously with the sender

Example:
N=5, K=3



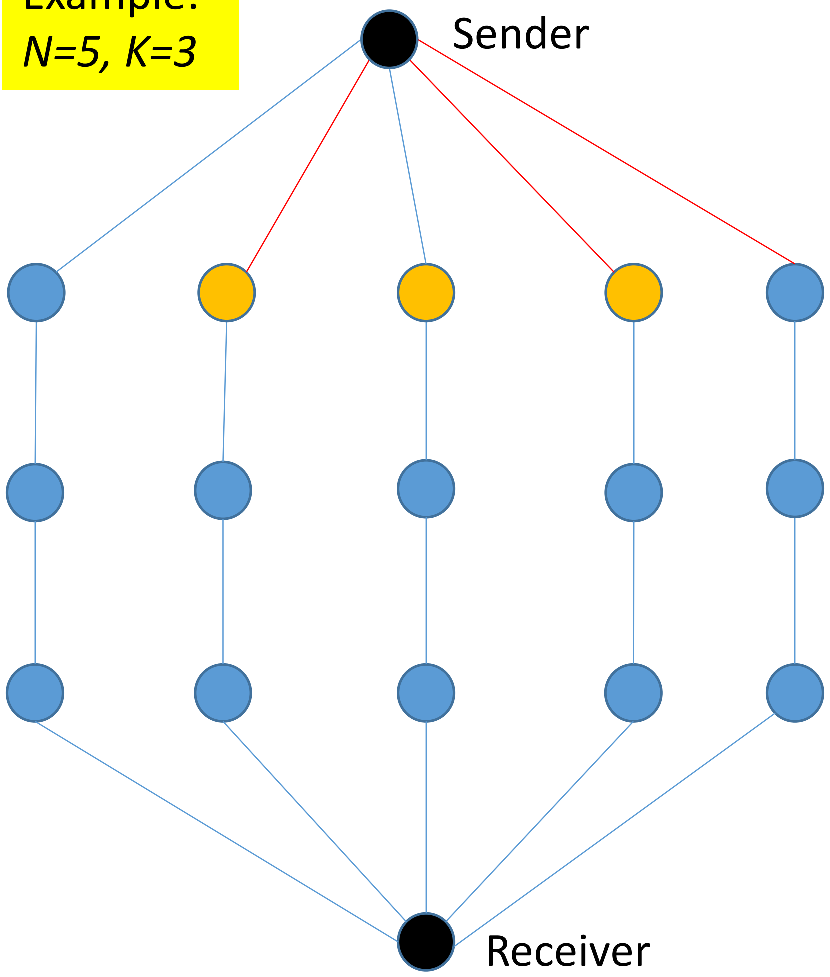
The *first* switching interval

Attackers

- **Basic Attackers:** listen to K distinct paths
  - Fixed
  - **Synchronized:** Switches hops synchronously with the sender

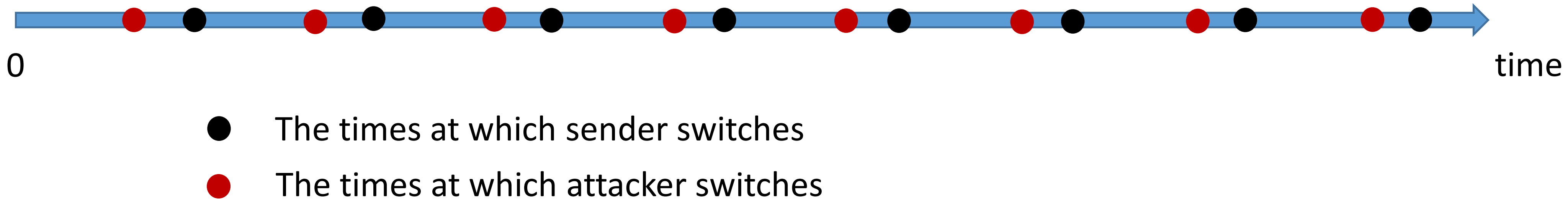
Example:
N=5, K=3

The *second* switching interval



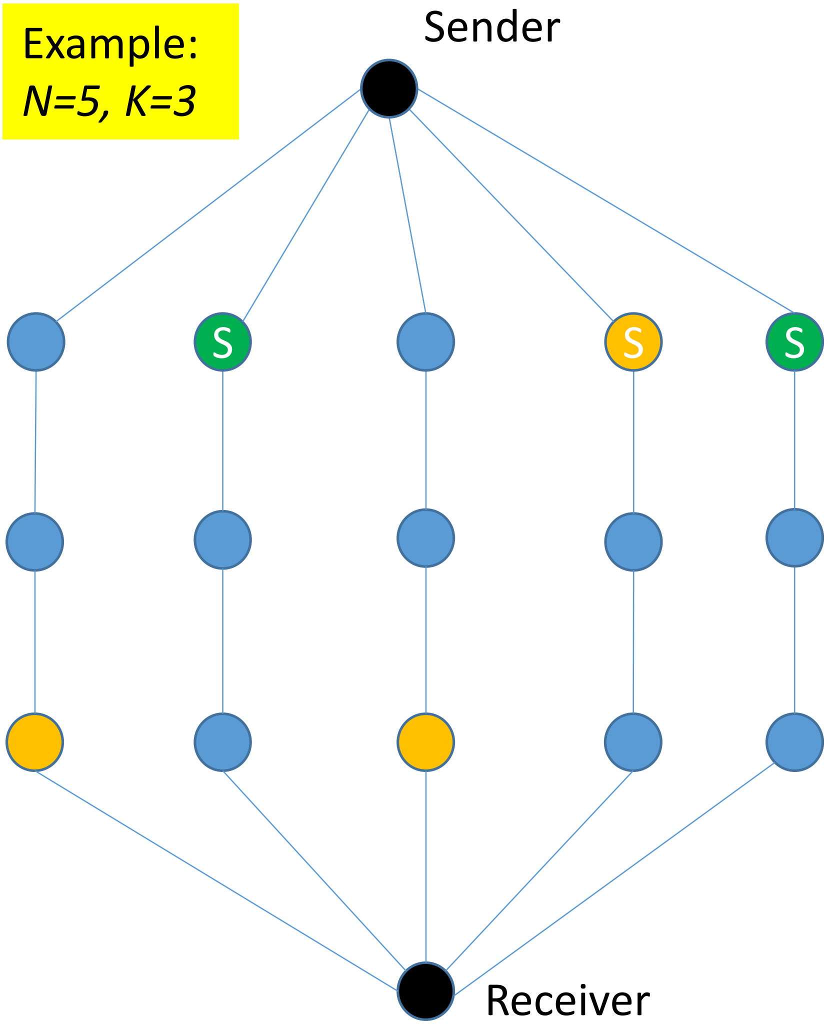
Attackers

- **Basic Attackers:** listen to K distinct paths
  - Fixed
  - Synchronized
  - **Independent:** Switches hops, but does not know when the sender switches



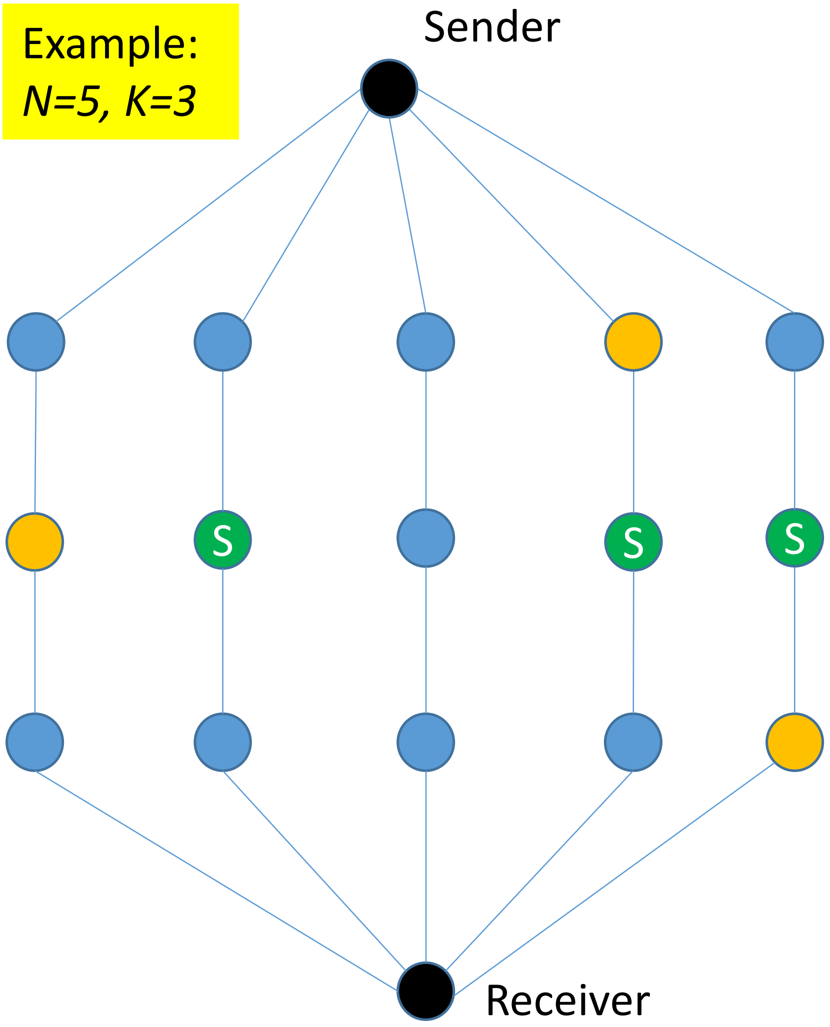
Attackers

- **Basic Attackers:** listen to K distinct paths
  - Fixed
  - Synchronized
  - Independent: Switches hops, but does not know when the sender switches
- **NDR Attackers:** Synched with sender, Deliberately want to exploit the NDR side channel,
  - **NDR Blind:** listens to K random hops
  - The number of choices in the example is $\binom{15}{3}$



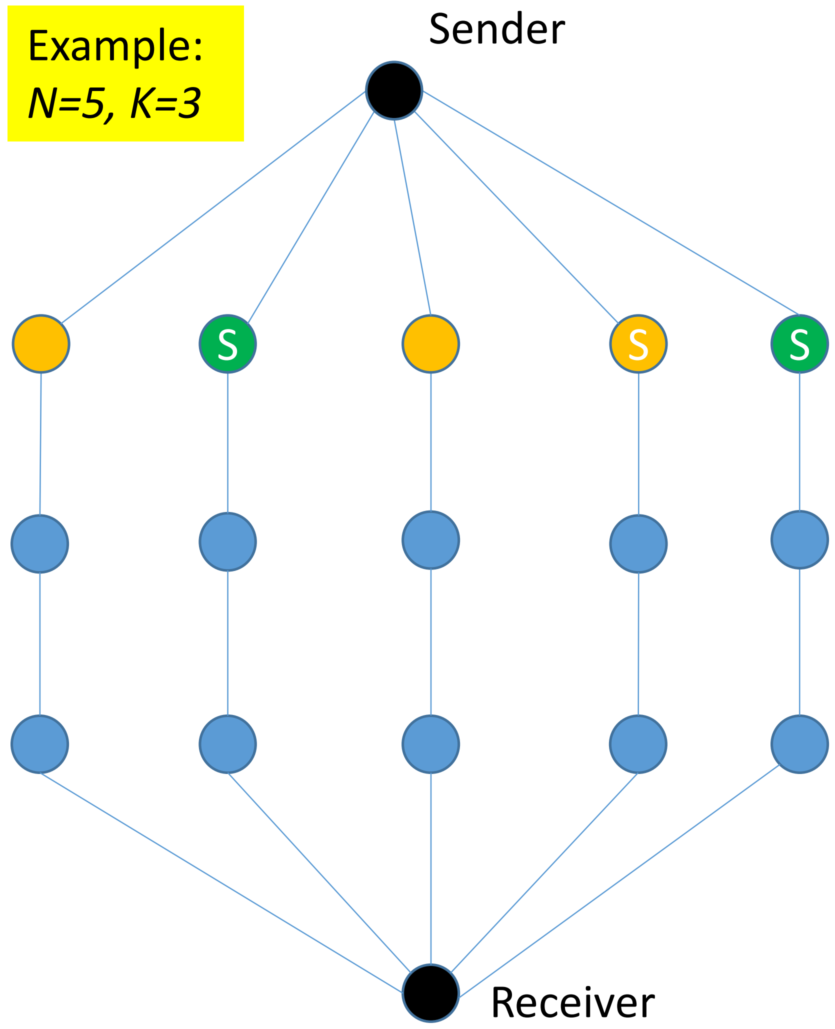
Attackers

- **Basic Attackers:** listen to K distinct paths
  - Fixed
  - Synchronized
  - Independent: Switches hops, but does not know when the sender switches
- **NDR Attackers:** Synched with sender, Deliberately want to exploit the NDR side channel,
  - **NDR Blind:** listens to K random hops



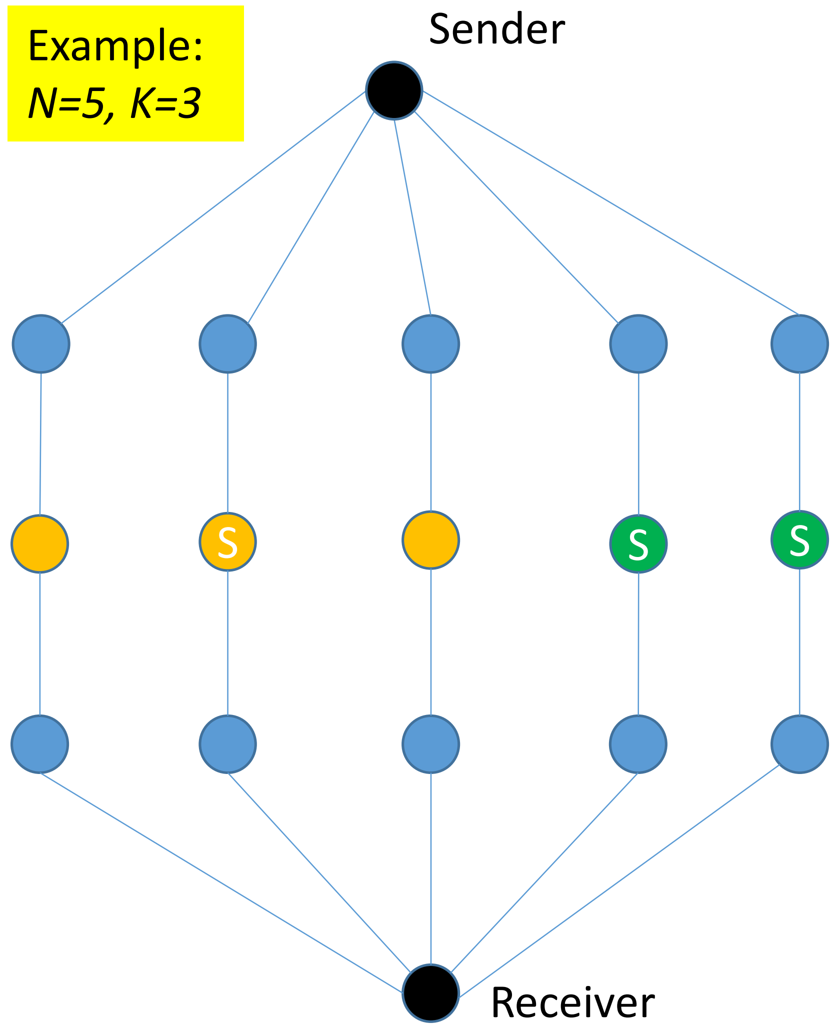
Attackers

- **Basic Attackers:** listen to K distinct paths
  - Fixed
  - Synchronized
  - Independent: Switches hops, but does not know when the sender switches
- **NDR Attackers:** Synched with sender, Deliberately want to exploit the NDR side channel,
  - NDR Blind: listens to K random hops
  - **NDR Planned:** Chase the shares



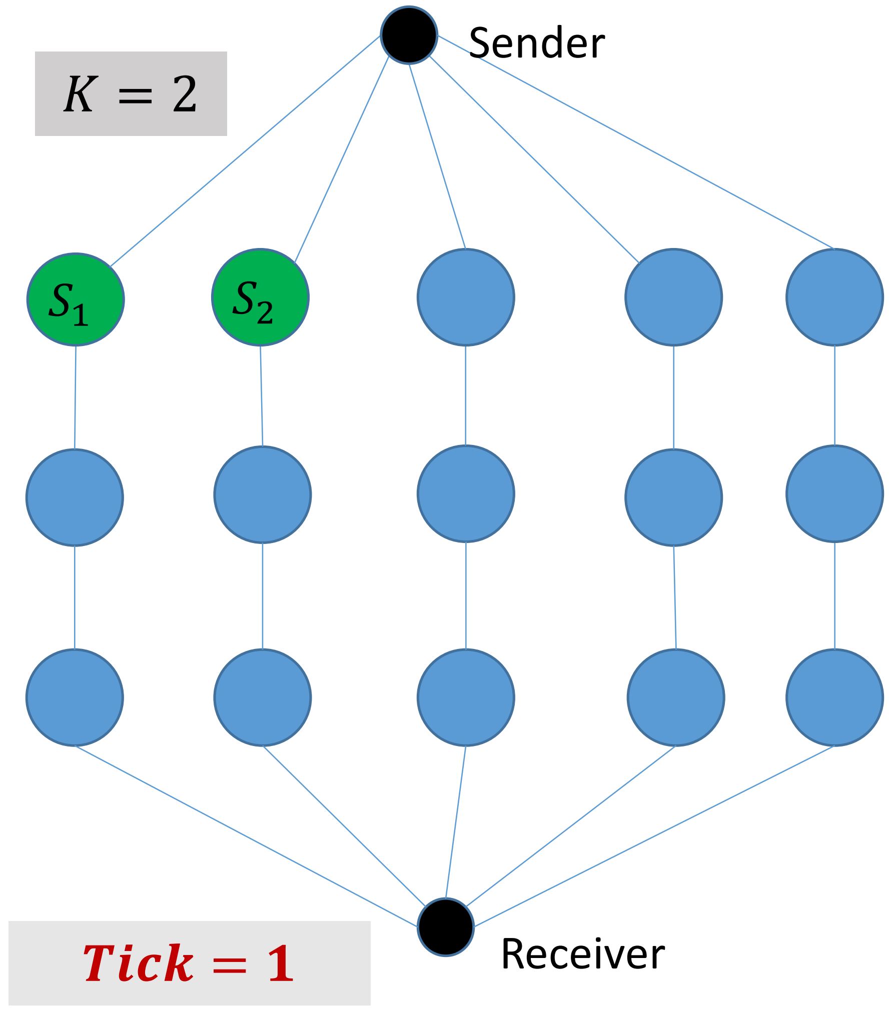
Attackers

- **Basic:** listen to K distinct paths
  - Fixed
  - Synchronized
  - Independent: Switches hops, but does not know when the sender switches
- **NDR:** Synched with sender, Deliberately want to exploit the NDR side channel,
  - NDR Blind: listens to K random hops
  - **NDR Planned:** Chases the shares



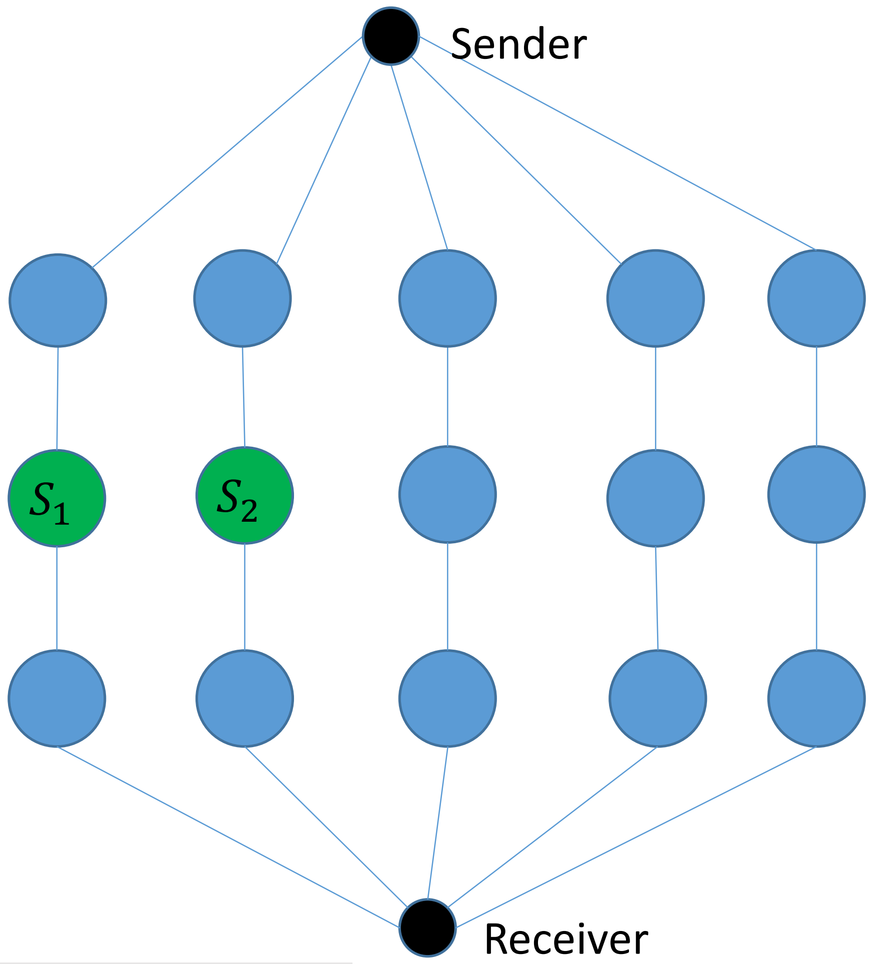
An Abstract Model to Do Analysis

- Assumptions:
  - There are N node-disjoint paths from the sender to receiver, which have the same length
  - We consider time as clock ticks
  - It takes one clock tick for each share to traverse each link of the network.
  - At clock tick $t = 0$, the sender sends shares of the information along K random paths.
  - At each subsequent tick it selects a new set of K paths.



An Abstract Model to Do Analysis (Cont.)

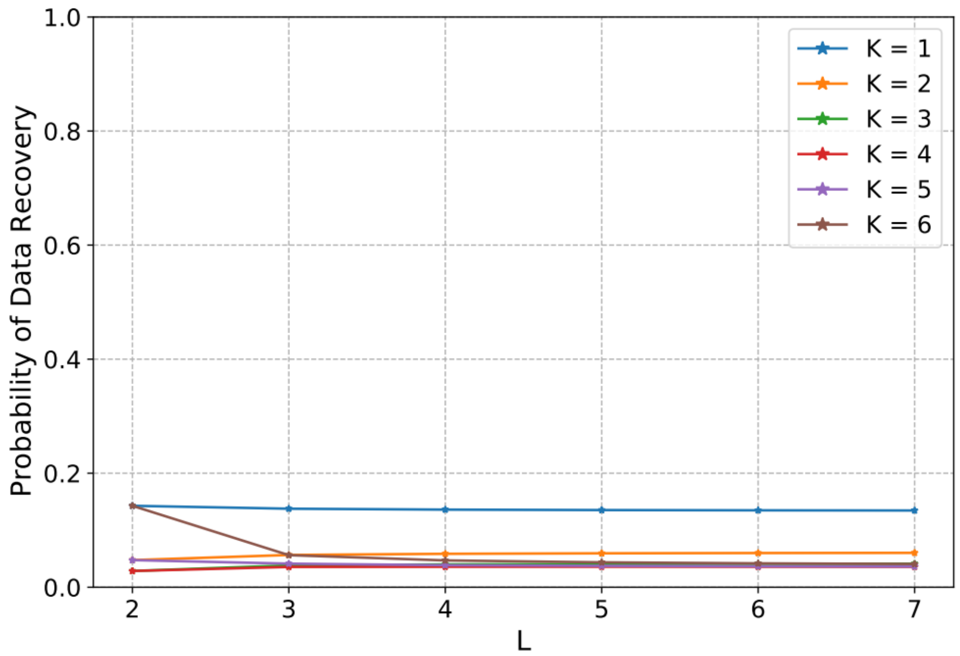
- Assumptions:
  - All disjoint paths have the same path length
  - We consider time as clock ticks
  - It takes one clock tick for each share to traverse each link of the network.
  - At clock tick t = 0, the sender sends shares of the information along K random paths.
  - At each subsequent tick it selects a new set of K paths.



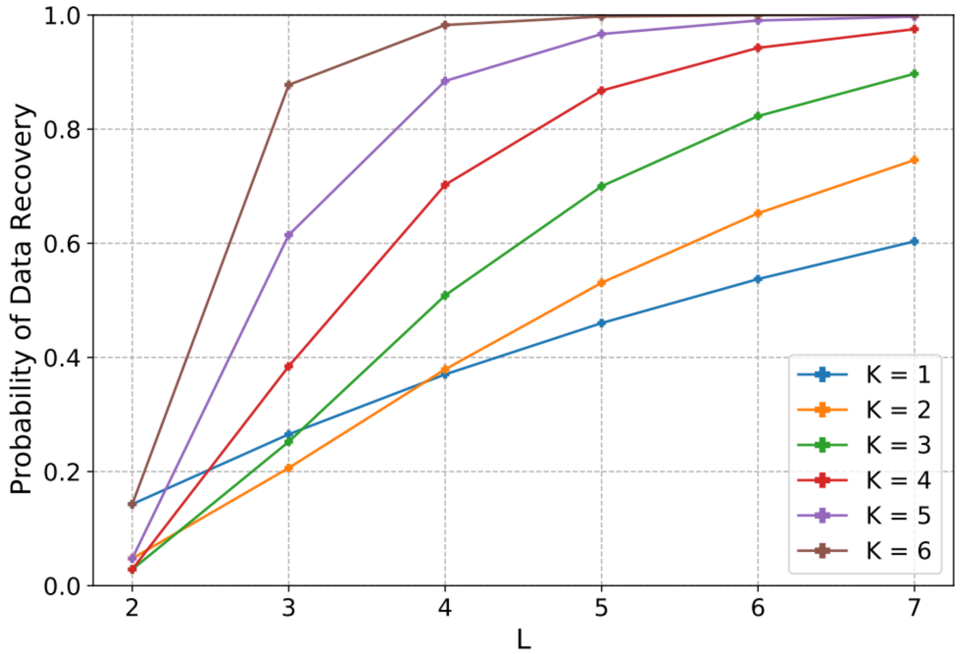
$\boldsymbol{Tick = 2}$

Analytics: Effectiveness of NDR Attackers

- A **single message**, which is sent by the sender, was considered.
- **Measure of Interest:** Probability of Data Recovery (probability of capturing all *K* shares of the message)
- Seven disjoint path from the sender to the receiver *(N=7)*



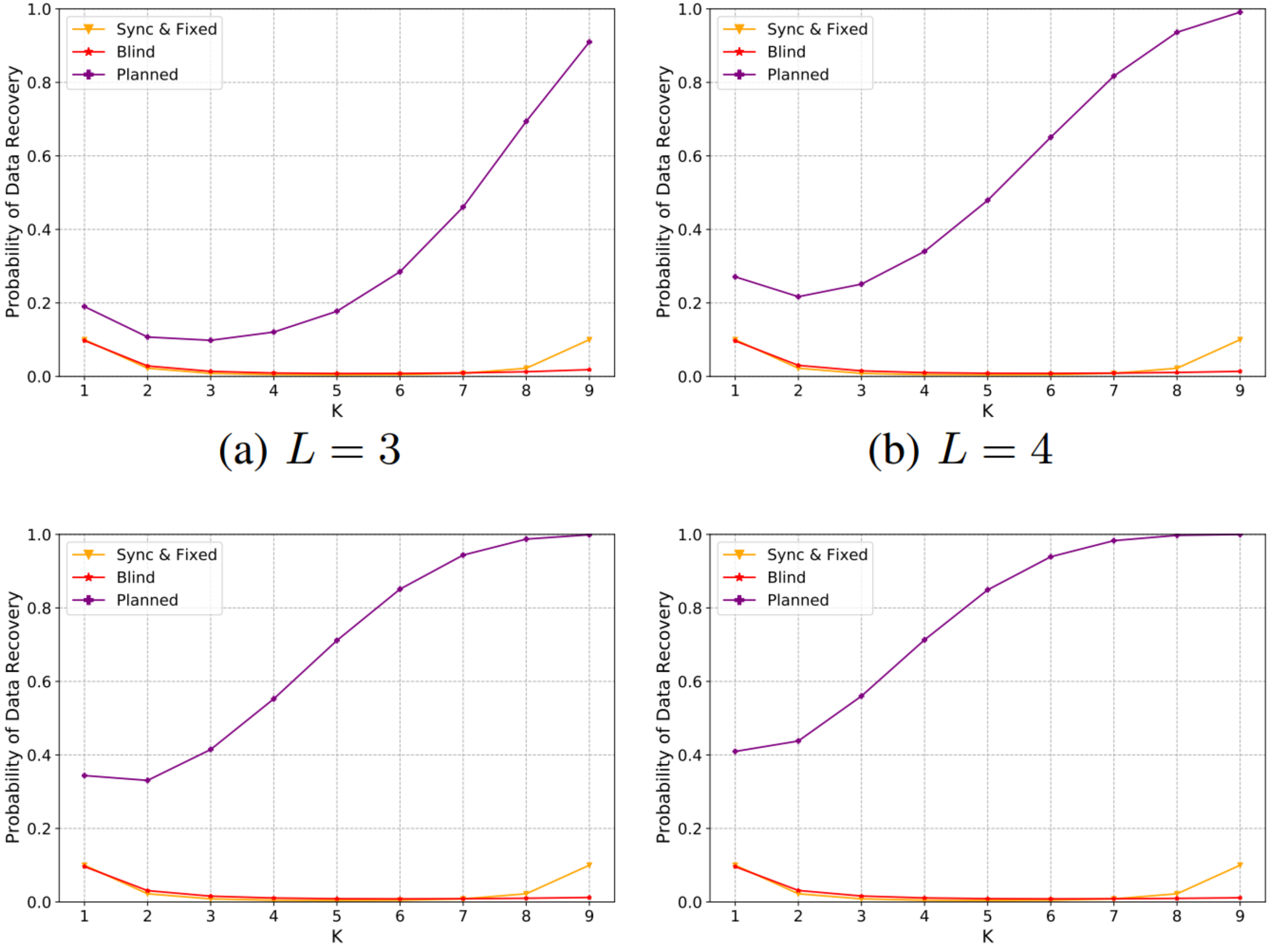
NDR Blind Attacker



NDR Planned Attacker

Analytics: Impact of Path Length and Number of Shares

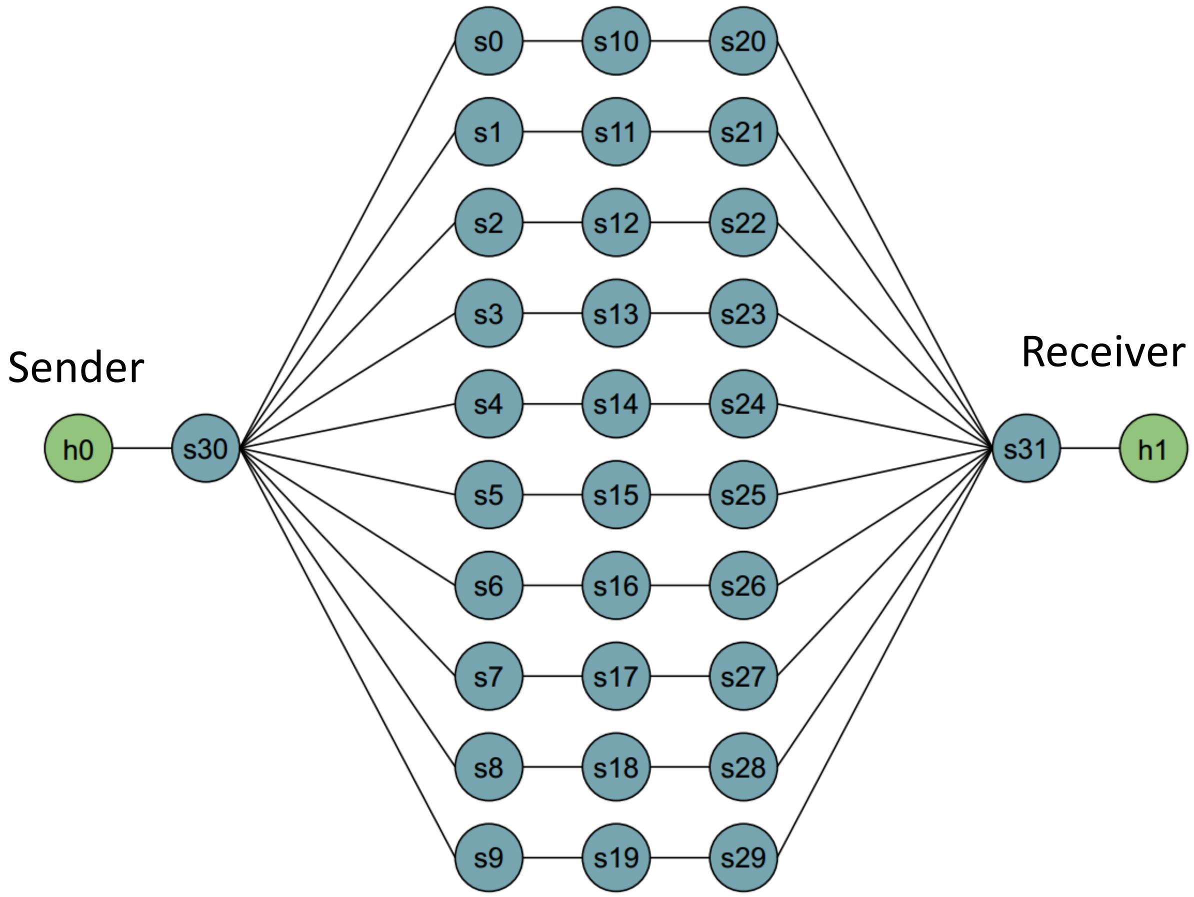
- **Setting:** 10 disjoint paths *(N=10)*
- ***Note:*** Since the Fixed and Sync attackers probe only the nodes at distance one from the sender, their probability of recovery does not change with path length.
- ***Important Observations:***
  - The Fixed, Sync, and Blind attackers, that do not intelligently attempt to exploit the side-channel, are not very effective.
  - The Planned attacker that strategically exploits the side-channel is increasingly effective at capturing all *K* shares as the path length increases.



(a) $L = 3$ (b) $L = 4$ (c) $L = 5$ (d) $L = 6$

Settings for Mininet Experiments

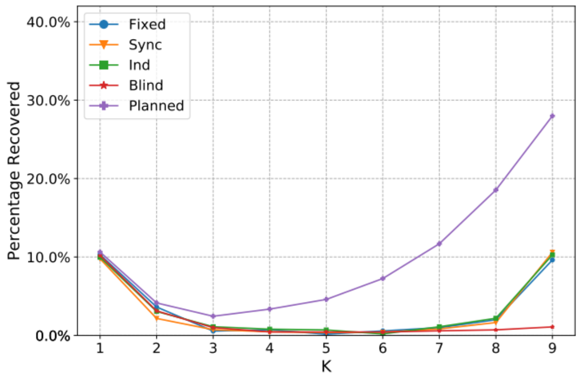
- N = 10 (ten paths)
- The capacity of links was not restricted.
- **Server:** Intel Xeon Silver 4114 CPUs running at 2.20 *GHz*
- **Virtual Machine:** CentOS VM in QEMU with 6 Cores and 8 *GB* of RAM
- **Controller:** ONOS 1.14.0-SNAPSHOT
- **Switch:** Open vSwitch 2.9.2 supporting OpenFlow 1.4
- File size = 10 *MB*, message size = 512 *B*
- Length of switching intervals:
  - **Default:** 100 *ms*
  - Independent Attacker: 200 *ms*



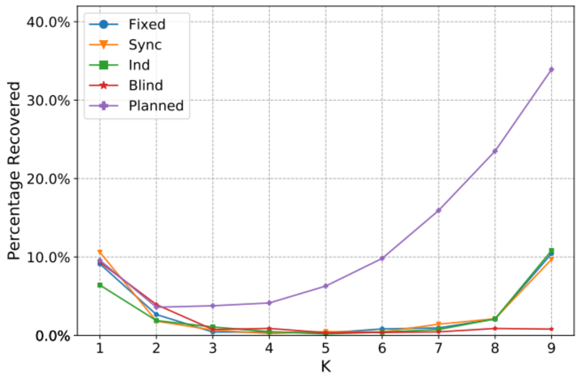
Measure of Interest and Scenarios

- **Measure:**
  - Percentage Recovered
- **Fixed Delay Scenario:** Each link has the same constant delay of 50 *ms.*
  - ***Issue:*** All paths had the **same delay**, but in real networks, each link, and, in turn, each path, has a **different delay**.
- How to consider a more realistic scenario?
- We applied the following actions:
  1. A **random delay** is added to the first link of each path.
     - **Range:** [0,100] ***ms***
     - Sampled per each path
  2. Applying **jitter** to each message to emulate the small variations in delay, which is common in real networks.
     - **Range:** [0,100] μs

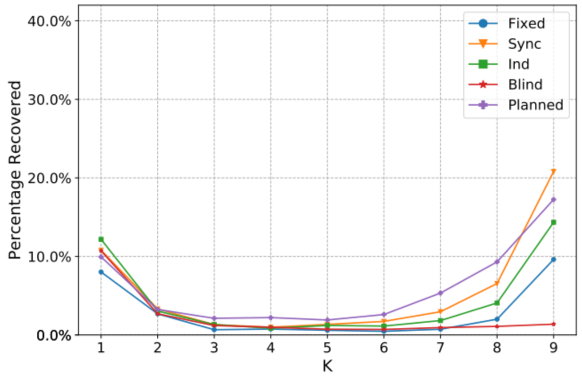
Experiments: Comparing Results of Two Scenarios



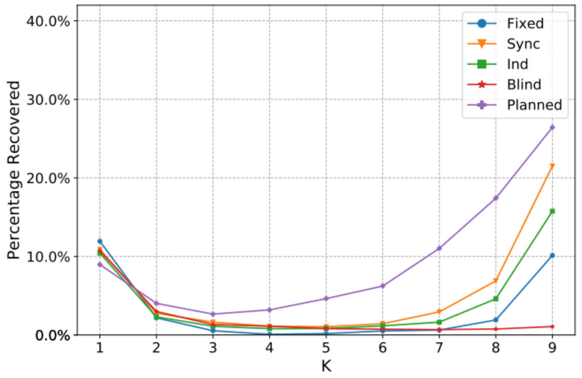
(a) $L = 3$



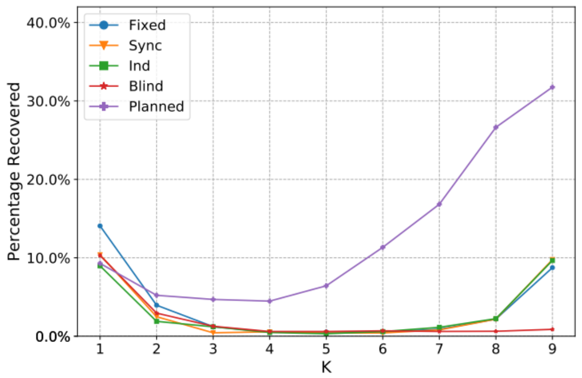
(b) $L = 4$



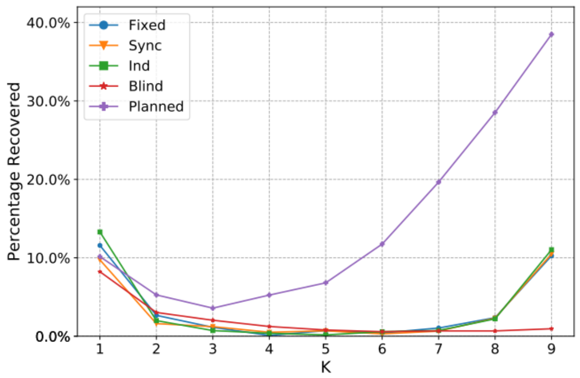
(a) $L = 3$



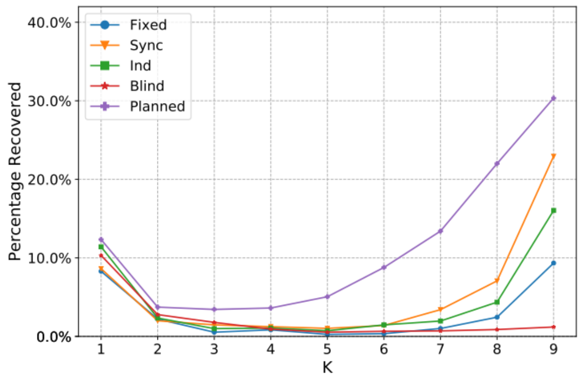
(b) $L = 4$



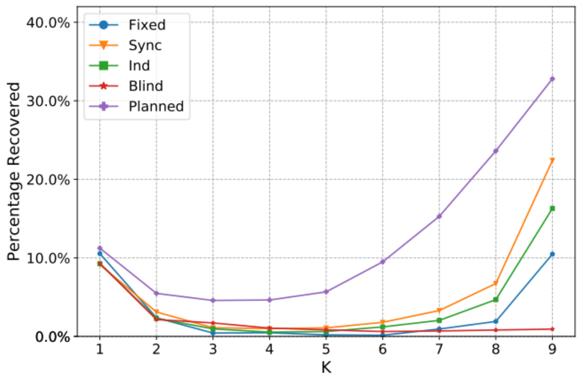
(c) $L = 5$



(d) $L = 6$



(c) $L = 5$



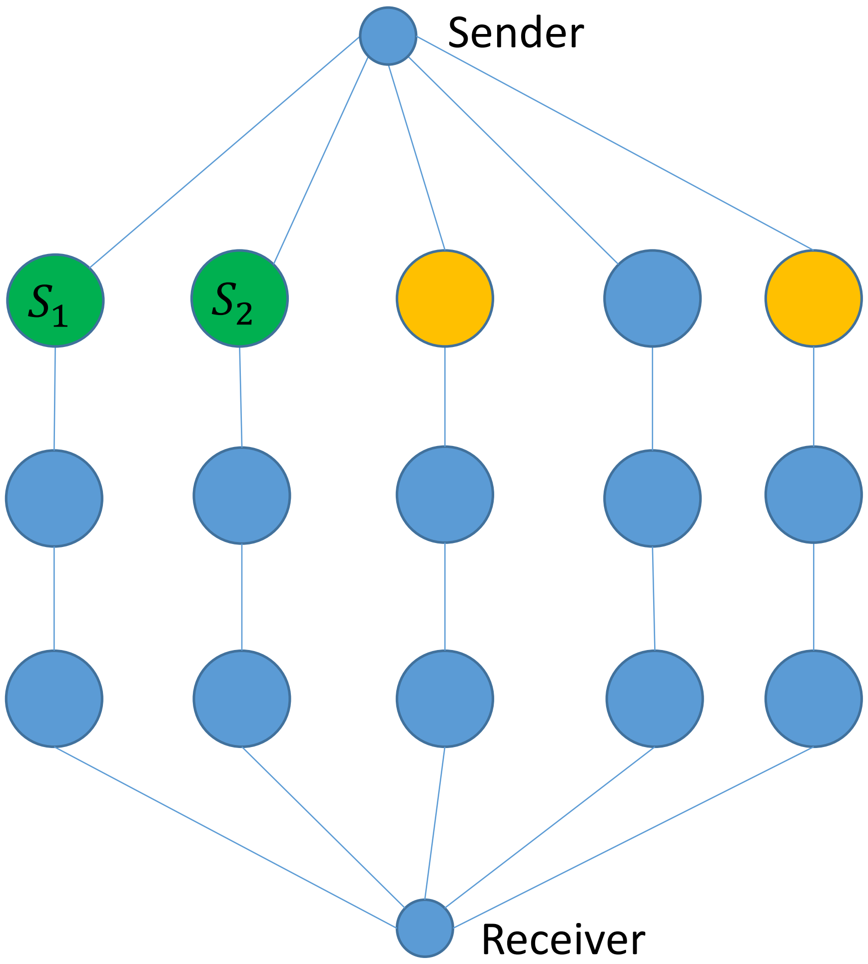
(d) $L = 6$

Fixed Delay for Each Link

Added Random Delay and Jitter

Proposed Countermeasure

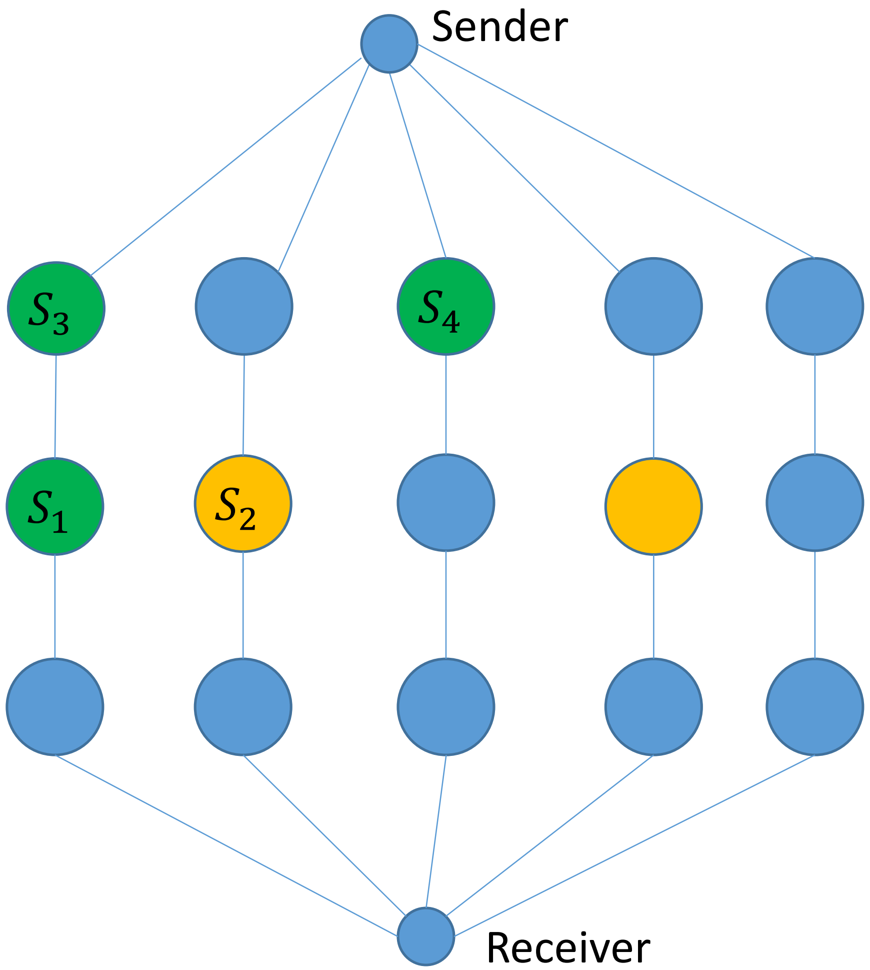
- The proposed countermeasure is based on
  - Generating more shares:
    - Splitting information to KH shares rather than K shares (H>1)
  - Spreading shares across both space and time:
    - Sender sends shares over multiple switching intervals
    - For example, in the abstract model, the sender sends K shares at the ticks $0, 1, \ldots, H-1$ along K paths which are selected uniformly.
- Example:
  - $K = 2, H = 3$
  - NDR Planned Attacker



$t = 1$
Captured shares={}

Proposed Countermeasure

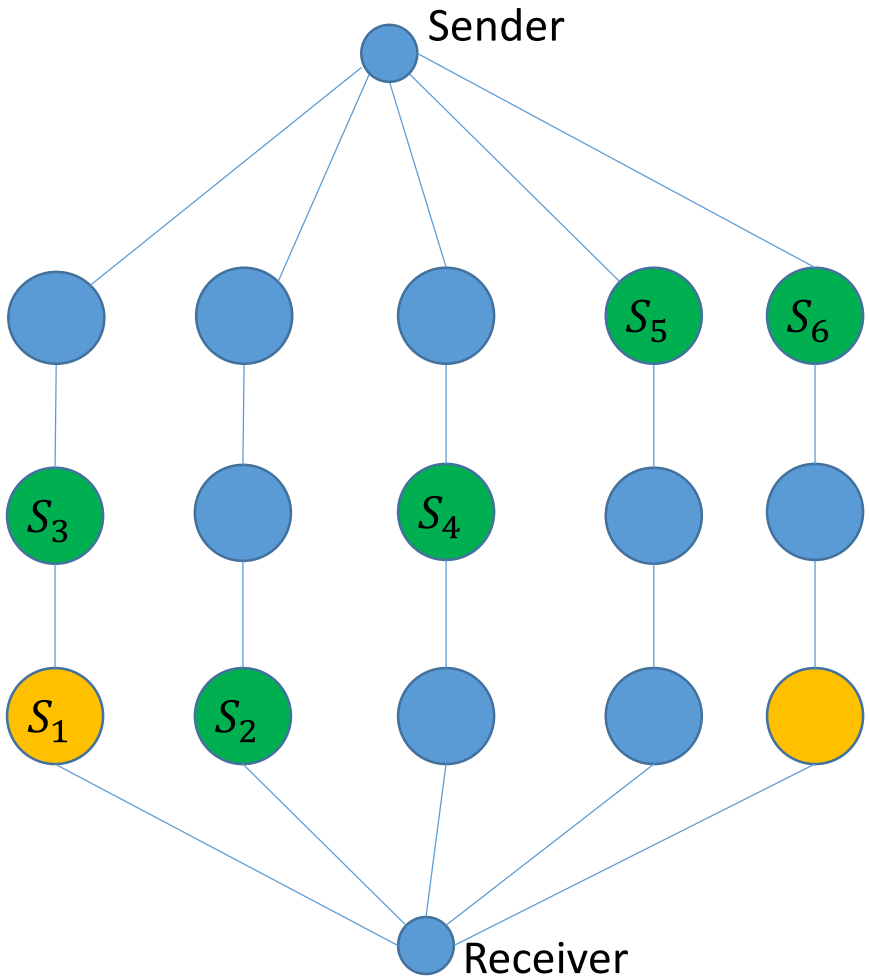
- The proposed countermeasure is based on
  - Generating more shares:
    - Splitting information to *KH* shares rather than *K* shares (*H*>1)
  - Spreading shares across both space and time:
    - Sender sends shares over multiple switching intervals
    - For example, in the abstract model, the sender sends K shares at the ticks $0, 1, \ldots, H-1$ along K paths which are selected uniformly.
- Example:
  - $K = 2, H = 3$
  - NDR Planned Attacker



$t = 2$
Captured shares={S_2}

Proposed Countermeasure

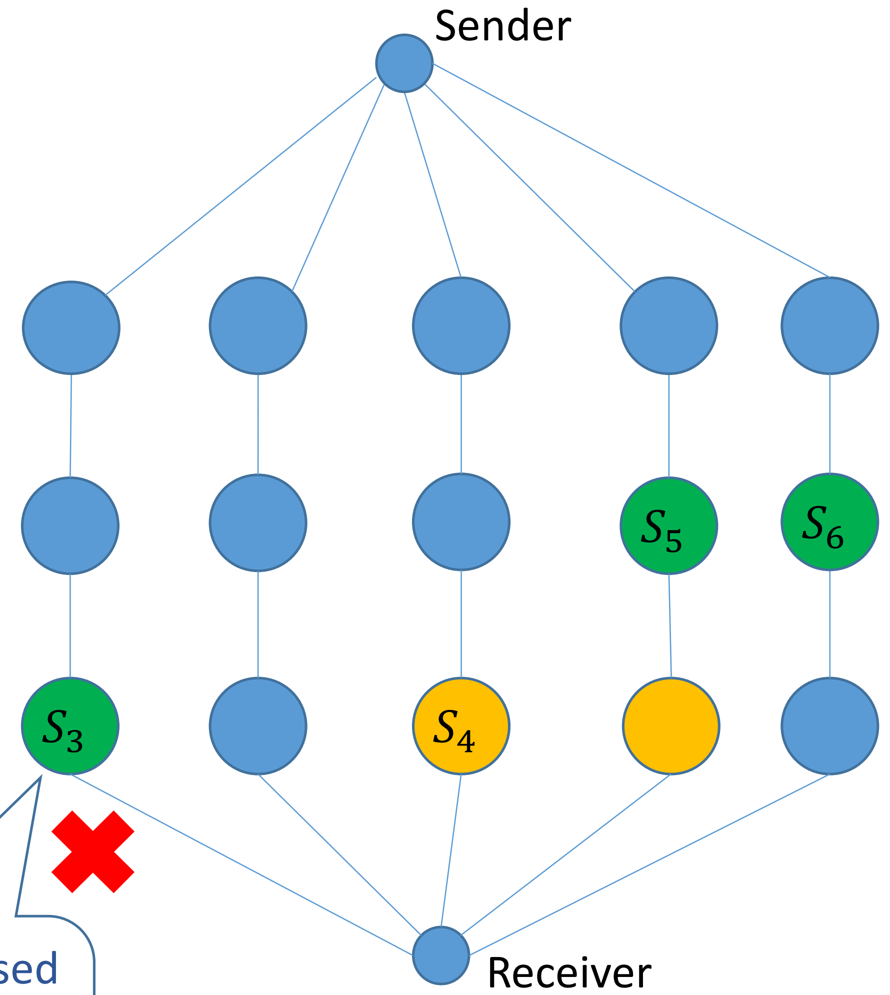
- The proposed countermeasure is based on
  - Generating more shares:
    - Splitting information to *KH* shares rather than *K* shares (*H*>1)
  - Spreading shares across both space and time:
    - Sender sends shares over multiple switching intervals
    - For example, in the abstract model, the sender sends K shares at the ticks $0, 1, \dots, H-1$ along K paths which are selected uniformly.
- Example:
  - $K = 2, H = 3$
  - NDR Planned Attacker



$t = 3$
Captured shares=$\{S_1, S_2\}$

Proposed Countermeasure

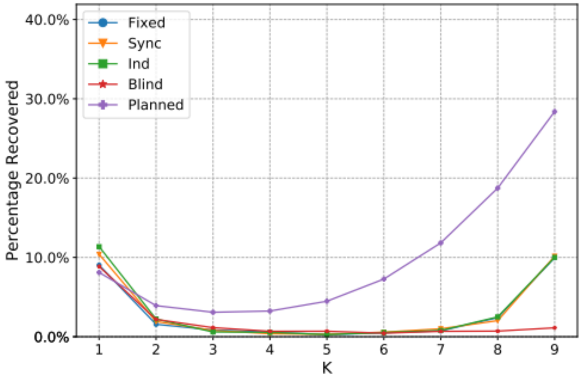
- The proposed countermeasure is based on
  - Generating more shares:
    - Splitting information to *KH* shares rather than *K* shares (*H*>1)
  - Spreading shares across both space and time:
    - Sender sends shares over multiple switching intervals
    - For example, in the abstract model, the sender sends K shares at the ticks $0, 1, \ldots, H-1$ along K paths which are selected uniformly.
- Example:
  - $K = 2, H = 3$
  - NDR Planned Attacker



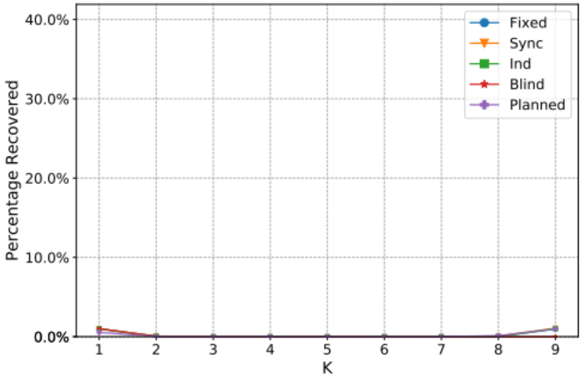
Fail: Attacker missed S_3, it had only one chance to capture S_3 and S_4

$t = 4$
Captured shares=$\{S_1, S_2, S_4\}$

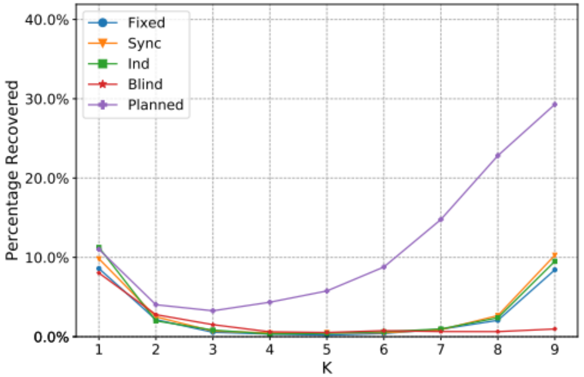
Experiments:: Effectiveness of the Countermeasure on Percentage Recovered



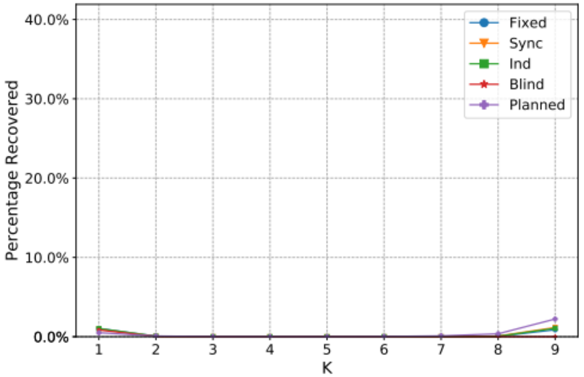
(a) $L = 3$, No Countermeasure



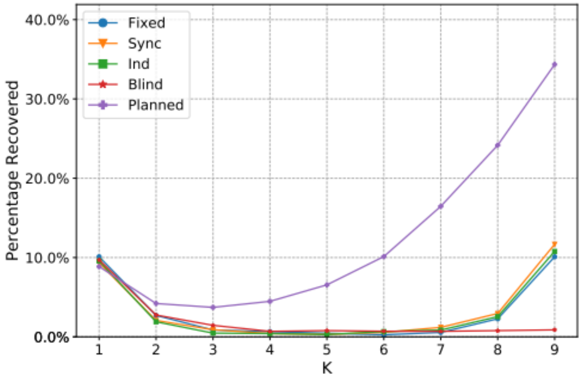
(b) $L = 3$, With Countermeasure



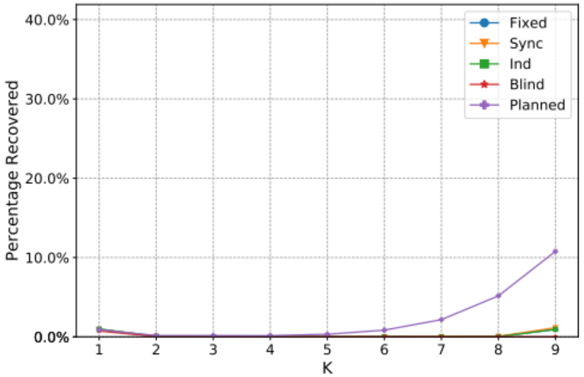
(c) $L = 4$, No Countermeasure



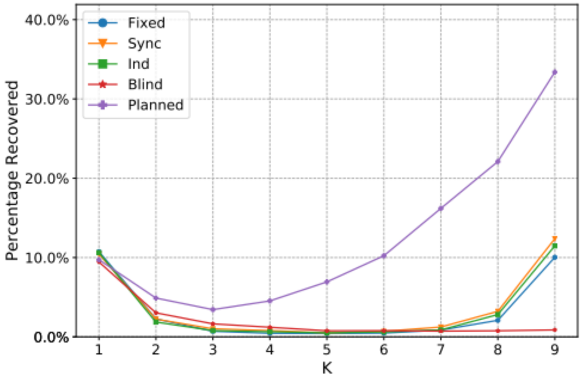
(d) $L = 4$, With Countermeasure



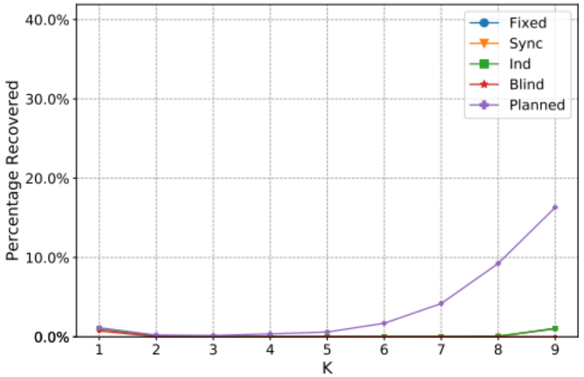
(e) $L = 5$, No Countermeasure



(f) $L = 5$, With Countermeasure



(g) $L = 6$, No Countermeasure



(h) $L = 6$, With Countermeasure

Conclusion

- We uncovered vulnerability of Secret Sharing-based schemes in real networks, introducing Network Data Remanence (NDR) side channel.
- We demonstrated the presence of NDR in a physical SDN testbed.
- We identified five kinds of attacks which exploit NDR side channel to break confidentiality of a recently proposed Secret Sharing-based scheme (MSSS).
- The effectiveness of each attack was analyzed in an abstract model of network.
- Also, Mininet was used to evaluate the success probability of each attacker.
- Finally, a countermeasure was proposed for protection against NDR side-channel.

Thanks for your attention

Any Question?