ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Practical Work

Network Security Group, Department of Computer Science, ETH Zurich

Setup of a national SCION testbed for cyberdefence

by Silvan Niederer

Autumn 2022

ETH student ID:	15-941-958
E-mail address:	nsilvan@student.ethz.ch
Supervisors:	Llorenç Roma Fabio Streun Prof. Dr. Adrian Perrig
Date of submission:	January 20, 2023

Abstract

As SCION networking has recently become available to business customers of major Internet service providers (ISPs) in Switzerland, armasuisse CYD Campus has installed SCION network access to three different ISPs at different locations for a real-world test network. SCION provides path control and isolation to entities and avoids the security issues present in modern BGP environments. In this work, we analyse and accompany the installation of commercial Anapaya EDGE devices and we establish an experimental setup that can facilitate the development of future work. We also evaluate the performance characteristics of currently available commercial solutions and show that packet reordering can adversely affect performance of the SCION IP Gateway.

Acknowledgments

I thank Llorenç Roma and Fabio Streun for their supervision and insights during this work. I thank Lucas Crijns from armasuisse for his support, guidance and time configuring and explaining the Tofino switch and providing the P4 tap program. I would also like to thank Ivo Stragiotti for his continued support and supply in all matters of hardware and late-night network setup and debugging. I extend my gratitude to Prof. Perrig for the possibility to work on this project under his supervision. A hat tip goes to the Anapaya Engineers entertaining my questions and unconventional setups. Last but not least, I'd like to thank Dr. Vincent Lenders, Colin Barschel and Giorgio Tresoldi for fruitful discussions and brainstorming.

Contents

1 Introduction

Traditional IP/BGP routing is inherently best-effort. While destination-based routing allowed for a simple and quick growth of the internet, it lacks guarantees in availability and security. SCION [29] describes a newly designed internet and new protocols that allow for additional authentication of network paths. In recent years, SCION has become available in production [23] and is being sold commercially by internet service providers (ISPs). SCIONlab[16], a lab environment similar to Planetlab[6], already provides an experimental platform. However, SCIONLab can leverage ideal circumstances and not all elements will function the same way in commercial deployments, where stability trumps experimentation. To fill the gap between the experimental nature of the academic deployment and the stability of the commercial setting, the armasuisse Cyber-Defense Campus (CYD Campus) aims to be able to test SCION in the commercial space. To further explore the potential of this new technology, this unique testbed will be available to researchers, governments and partners in Switzerland and abroad. This can help to establish Switzerland as a leader in the experimentation of new internet technologies and attract international attention and cooperation. Finally, it can be used to test and evaluate new security and reliability features of SCION that can be useful for critical infrastructure and government communication networks that are sensitive to failures and breaches. In this practical work, we evaluate the SCION functionality and capabilities and how they are deployed in a productive environment.

1.1 Contributions

We create an independent test infrastructure for SCION. For packet analysis and modification, we create and install a network tap device that allows for the simulation of a subset of the Dolev-Yao attacker model. We also run performance tests that show the currently available real-world performance of a SCION network as provided by Anapaya, and show how malicious or involuntary packet reordering can negatively affect performance.

1.2 Organization

This practical work is split into 2 parts:

1. Analyzing, assisting and documenting the deployment of SCION at armasuisse CYD Campus
2. Implementation of a test environment for network testing and experimentation

In the first part, the focus is on operational stabilities, ease of deployment, and the perspective of an application developer. In the second part, we create an infrastructure that allows for future experiments. We test the feasibility of this infrastructure in an experiment requiring modification of the packet stream.

1.3 Testbed design goals

The goals of this work are to

1. Setup and configure the SCION network access at the armasuisse CYD Campus locations in Lausanne, Thun and Zürich
2. Implement scripts facilitating configuration and rollback
3. Setup a test infrastructure to monitor and modify SCION traffic for future work
4. Conduct an experiment using the created test infrastructure

This work should mark the beginning of the efforts of armasuisse to test and experiment with SCION networks outside of academic boundaries.

1.4 Related work

The SCION deployment for armasuisse is primarily intended for experimentation within a commercial network, but not as a part of a production environment within armasuisse. In contrast to SCIONLab[16], it is not isolated to the lab environment and uses a public SCION-only ASN instead of reserved experimentation SCION ASNs. SIDNLabs also operates a test environment for SCION in the Netherlands, focusing on a P4 implementation[8]. This program is part of the evaluation of emerging internet architectures named 2STiC[1]. Productive deployments include SCI-ED (SCION for ETH Domain)[37] and the SSFN (Secure Swiss Finance Network)[39] as well as the recently created Swiss Secure Health Network[3], which at the time of writing is not yet in operation. The CYD Campus SCION access combines the tools and services available in production environments with the possibility to modify incoming and outgoing packets. This installation allows for testing on the same infrastructure that is also used by commercial customers.

2 Background

2.1 Internet architecture

SCION [5] is a next-generation path-aware internet architecture. Unlike the existing internet architecture, it enables Autonomous Systems (ASes) to create and participate in virtually separated networks (IPVPN) by design. This provides resilience against DDoS and BGP hijacking attacks through isolation. Among other features, it enables network nodes to send data over multiple paths for both increased bandwidth and faster connection recovery.

2.1.1 A short history on traditional networking

Nowadays, internet connectivity is omnipresent. In 2021, 65% of the worldwide population was using the internet, pushing a combined 932 Tbit/s on average [15]. Modern end-user internet access is relatively cheap even for high speeds of 1 Gbit/s and more, but the underlying technologies are often evolutions of decade-old concepts. The famous ARPANET, envisioned in 1967 and first demonstrated in 1972, is today accepted as the first attempt at an interconnected network (internet)[34]. TCP/IP became the standard protocol within ARPANET in 1983, where it replaced the Network Control Protocol (NCP)[31]. This step established the Internet Protocol version 4 (IPv4)[30], which is still being used for more than 70% of HTTP requests to the Cloudflare CDN[7] despite successors existing with IPv6 since 1998[9]. However, the continued growth of the ARPANET required a change from the first inter-domain routing protocol, which had many shortcomings[22], to more scalable solutions. With the early internet developing, the Gateway-to-Gateway protocol (GGP, 1982)[13] and the more advanced Exterior Gateway Protocol (EGP, 1984)[24] were introduced. Between 1989 and 1991, the Border Gateway Protocol (BGP) was established and evolved in 3 different versions[18][19][20]. Famously called the "three-napkin protocol" for its inception at the lunch table at a conference in 1989[41], the protocol saw various revisions until BGP-4 was published in 1994[43]. BGP-4 is currently specified by RFC4271[33] which was published in 2006. Additional features have been integrated as optional extensions to BGP-4. Even though BGP-4 extends previous versions, it employs the same methods as earlier versions of BGP.

As history shows, there is considerable inertia in changing network protocols, especially when a change affects a large user base. In general, stability by employing trusted and proven protocols is desirable. However, BGP is described

by its creators as a "hack" that was meant to ensure functionality and usability of the growing internet, with security not being a concern at all[41].

2.1.2 BGP

The main assumption of BGP is that connecting AS trust each other. BGP runs over TCP connections and operators must explicitly configure information exchange. However, BGP announcement messages only contain information over what ASes an IP address is reachable and a cost indicator without cryptographic authentication. ASes can selectively announce paths depending on their contracts with other ASes or internal routing capacities. When multiple announcements exist, a single best path, usually based on hop count, is selected and forwarded.

Announcements can propagate through neighboring AS. As shown in Figure 2.1, a malicious or honest misconfigured ASN can announce a wrong network to a neighbor. Because network paths are changing, even new routes are not a good indicator for faulty announcements. While propagating correct announcements is desirable, also faulty announcements can propagate through the internet and even local misconfigurations can disrupt worldwide networking. Several approaches to securing BGP via extensions exist, for example RPKI / BGPSec. However, even though they reduce the attack surface considerably, because of various issues like incomplete validation and scalability issues, among others, their security guarantees are generally considered incomplete[28]. Also, the additional cryptographic overhead requires network operators to upgrade their hardware, which slows down adoption despite its intended compatibility.

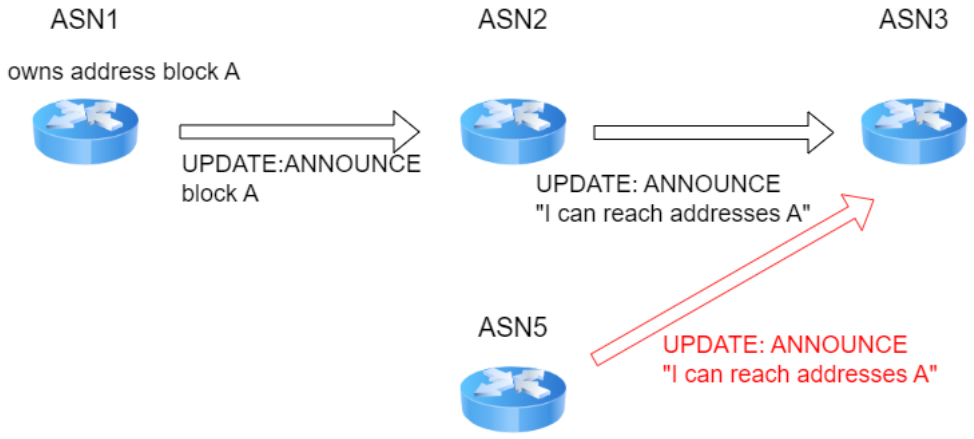

Figure 2.1: ASN1 legitimately owns an address block A and announces it to a neighbor, which announces the reachability to its neighbor. ASN3 cannot distinguish legitimate from malicious announcements

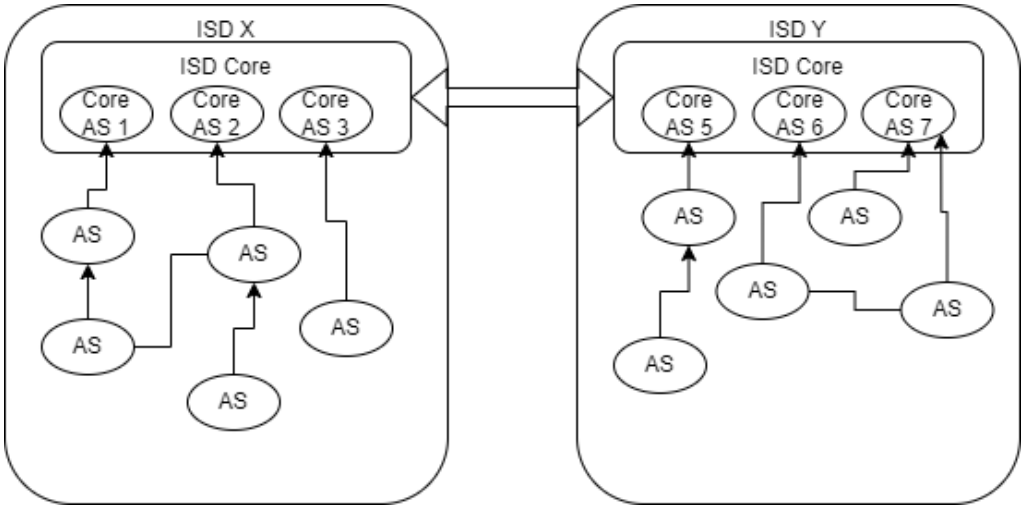

Figure 2.2: SCION network structure: Each isolation domain is a sovereign group of systems. The ISD core AS certify ASes that wish to join the ISD and accept only valid announcements. Non-core ASes can peer to ASes in other ISDs, but cannot announce transit paths to other ISDs

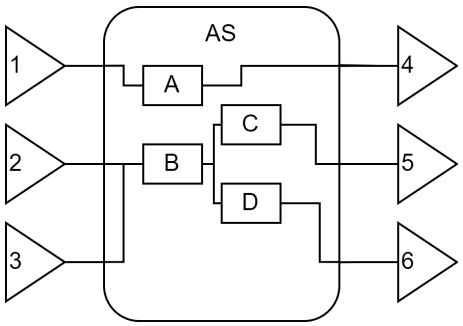

Figure 2.3: The SCION paths are an overlay. In this example, SCION interfaces 1-6 can be announced, but are not indicative of underlying hops A-D

2.1.3 SCION

An Isolation Domain (ISD) contains ASes that share mutual interests and trust assumptions, as shown in Figure 2.2. Currently, general-purpose ISDs are defined within legislation boundaries, i.e. countries. The AS in the core are publicly known by a trust root configuration (TRC). AS can verify the authenticity of other AS and paths via the TRC, and unauthenticated or unauthorized ASes are not eligible for traffic forwarding. The paths are specified by hops of AS ingress and egress interface IDs. These IDs are an overlay of a physical or virtual underlying network, i.e. an AS can route the packets in some underlay internally, allowing for use of existing infrastructure and scaling, as depicted in Figure 2.3. As such, a packet may even leave a geographic region without the overlay noticing. If an AS violates the trust assumptions within the underlay, the ISD Core can revoke the

certificates of the offending AS, removing it from the ISD. Even if the AS stays in the ISD, individual ASes can specify to no longer use paths over this AS.

Once constructed and verified, the path in the SCION header does not have to be modified for subsequent packets within a specified path lifetime. Legner et al[17] show that this leads to brute-forceable paths. A proposal to fix this is to authenticate every packet with EPIC (an acronym for "Every Packet Is Checked").

2.2 SCION packets

While SCION can wrap any higher-level protocol data, most SCION-enabled applications use UDP as a transport protocol. While TCP, despite it being slowly replaced with QUIC in the web domain, is still used for roughly 90% of internet traffic[36], it is not present in SCION and has to be tunneled through UDP. The rationale for this is that TCP traditionally does not support multiple paths and the multipath TCP extension (MPTCP) is less flexible than SCION multipath. Furthermore, QUIC is seen as a better replacement for TCP.

SCION packets, as described in Figure 2.4, have varying header lengths depending on the path. The host address size within a SCION AS is flexible up to 16 bytes and can differ between sender and receiver for flexibility. While IPv4 (4 bytes) and IPv6 (16 Bytes) are likely the most common host addresses, senders can set the address of any protocol. The Info and Hop fields contain information and authentication codes of the constructed path. In special cases, e.g. for packets that should be delivered in the local AS, the packet header structure differs from the most common structure shown in Figure 2.4.

2.3 Operations

Early versions of SCION have been proposed back in 2011[44]. While there was a testing environment called SCIONlab as early as 2017[12][16], these early deployments require a VPN to access and are thus only reachable via existing infrastructure. In late 2021, this changed with the first deployments that are offered over a direct connection[23]. Currently, the most prominent network making use of the SCION protocol is the Secure Swiss Finance Network (SSFN) which is operational and planned to replace the older MPLS-based Finance IPNet by 2024[39]. SIX provides this service for customers specific to the financial market. Additionally, SCION in the ETH Domain (SCI-ED)[37] connects research entities and partners of federal universities.

At the time of writing, SCION is available for commercial customers of Sunrise, Swisscom and SWITCH in Switzerland. The ISPs provide a link to the network. All of them rely on Anapaya Systems for SCION-related management. Currently, there are 2 major Isolation Domains in Switzerland: The open domain 64 as well as the strongly restricted SSFN domain for organizations in the financial sector.

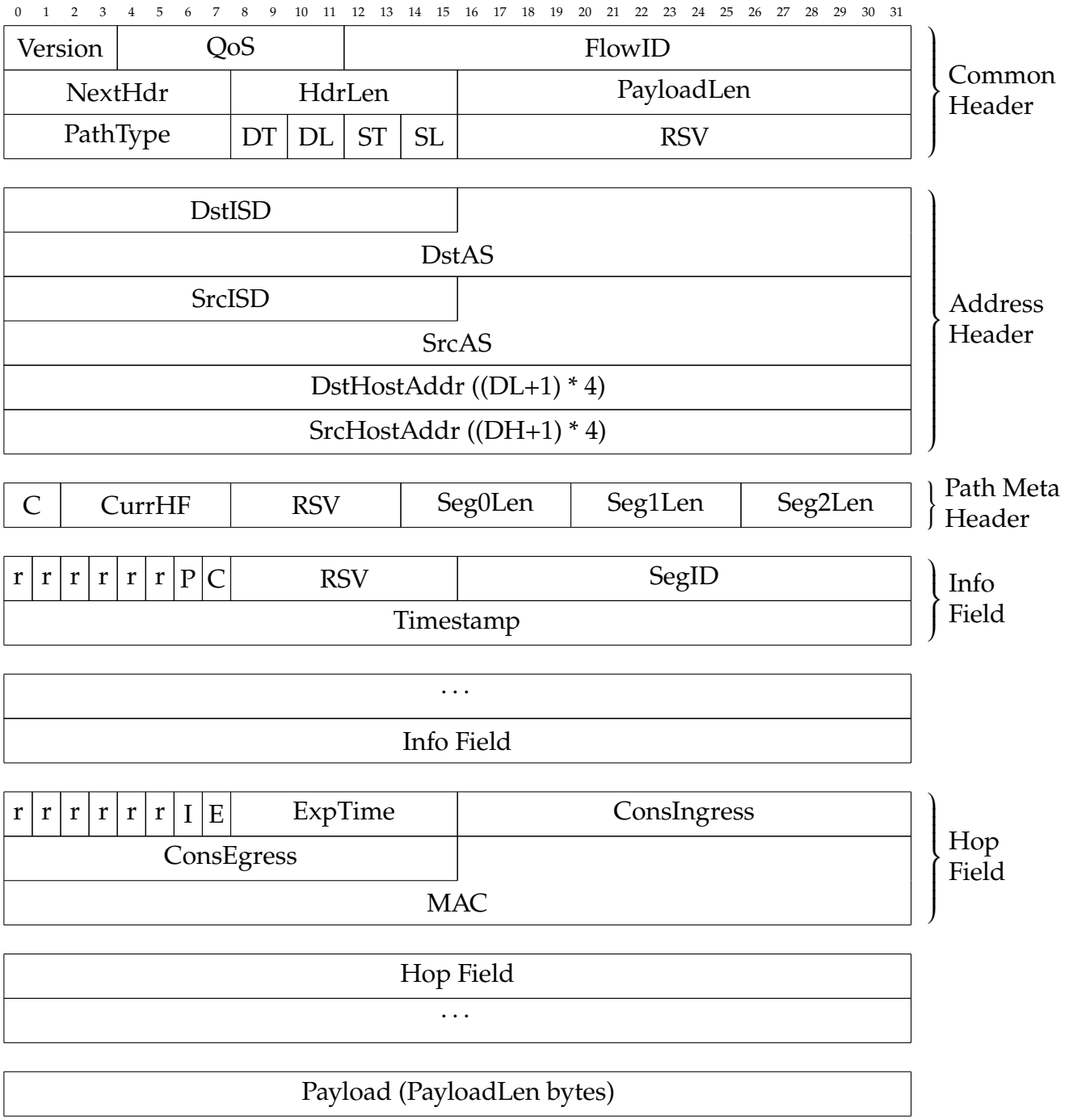

Figure 2.4: SCION packet structure. The number of Info- and Hop Fields depends on the selected path. The structure differs in special cases and depending on the enabled extensions

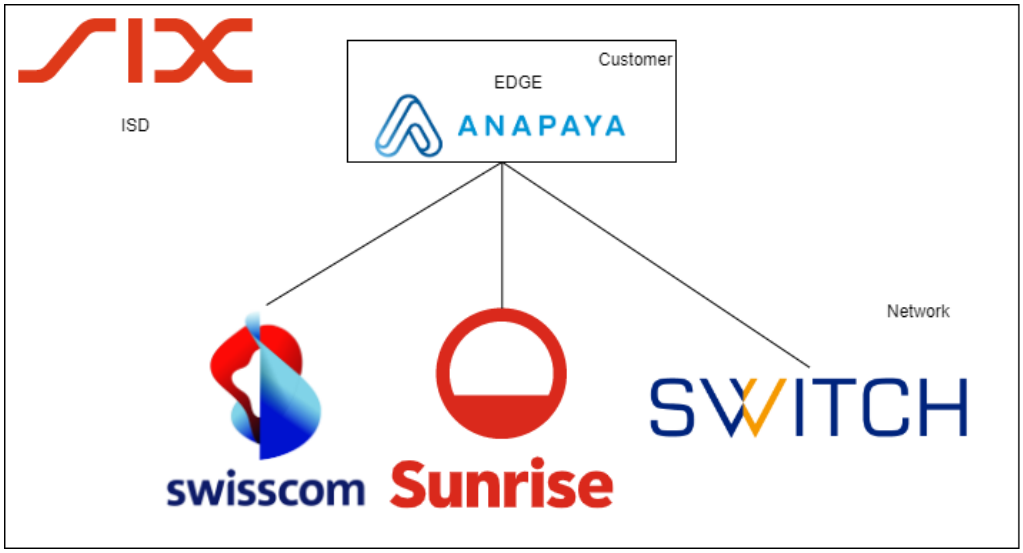

Figure 2.5: SSFN logical overview. Participation requires both network access by one of the approved ISPs and an Anapaya EDGE device. SIX manages the access to the SSFN isolation domain

For the SSFN, SIX authorizes participants, with connections provided by the ISPs, shown in Figure 2.5. For the open ISD64, Anapaya is the main certificate authority, and the ISPs authenticate their clients, i.e. guarantee that an entity is a customer within Switzerland. It is possible to opt-out and manage the private key of an AS within the customer's systems. In the managed environment, Anapaya creates the required certificates and has management access to the Anapaya EDGE device and therefore to the key of the certificate. All providers mention a SCION license and bill it in addition to standard business IP access and management services. This monthly fee is required for the certification of an AS and therefore is always part of a SCION network access contract. AS numbers are assigned in a public SCION-only range or to match RIPE ASN entries. For example, Swisscom is assigned the ISD-AS (IA) 64-3303, matching Isolation Domain 64 (Switzerland) and the RIPE ASN 3303 (Swisscom). ETH uses a public SCION-only IA 64-2:0:9, matching ISD 64 (Switzerland) with a SCION-only, i.e. not RIPE-registered identifier. At the time of writing, no ISP offers SCION-enabled residential internet access. Anapaya, Swisscom, Cyberlink and SwissIX sometimes use the terminology "B2B-focusing" network or internet for SCION.

Swisscom and SWITCH position the SCION service as an alternative to leased-line and other forms of IPVPN (e.g. MPLS-VPN). SCION does not depend on expensive private lines and allows for easy cross-ISP connections without significant coordination efforts that are inherent to MPLS-based systems.

Proposals for a name resolution service to resolve names into addresses exist with RHINE and its predecessor, RAINS, which was implemented in SCIONLab[4]. RHINE is not currently available in the SCION ISD64. Instead, all SCION-related

configuration is done via static addresses or SCION-specific host file entries. When using the SCION IP Gateway provided with the Anapaya EDGE, hosts can continue to use existing IP-based services like DNS within their networks.

3 SCION components overview

In this chapter, we provide an overview of the individual parts of the SCION deployment in the armasuisse CYD Campus.

armasuisse CYD Campus has ordered SCION network access for experimental usage at three of their office locations in Switzerland: At their main office in Thun as well as their satellite locations near EPFL in Lausanne and ETHZ in Zürich. The setup should be as diverse as possible to see different implementations and to leverage the features of SCION as a cross-ISP IPVPN solution. This also allows for pinpointing eventual issues within the networks to providers, and seeing if providers handle different anomalies differently. An overview of the entire setup is shown in 3.1. To match the most common deployments, armasuisse has ordered the managed SCION network access.

3.1 Installation

Within the duration of this work, hardware supply shortage led to an unavailability of 10 Gbit/s Anapaya EDGE devices. Therefore, SWITCH installed a whitebox

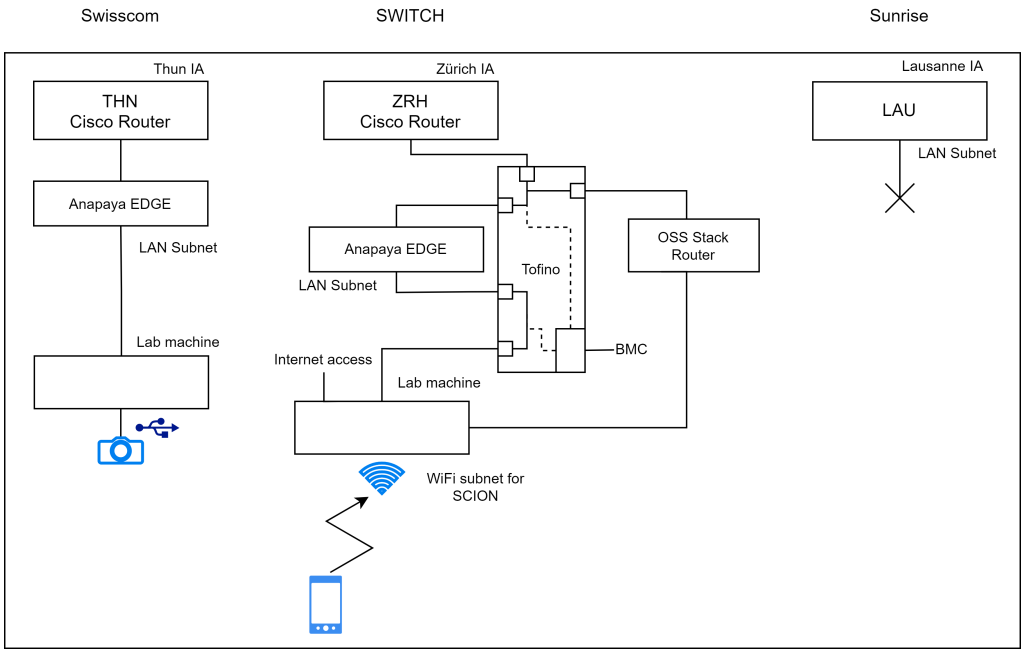

Figure 3.1: Full network setup used in this work

Location	Thun	Zürich	Lausanne	ETH
ISP	Swisscom	SWITCH	Sunrise	ETH
Link Speed	10G [1]	10G [2]	1G	1G (VM)
Management	Fully managed	Fully managed	Fully managed	ETH ISD

[1] Due to limited hardware availability, a 1G EDGE appliance is installed

[2] Due to limited hardware availability, the Anapaya SIG is running on a whitebox device

Table 3.1: armasuisse experimental SCION access

machine at the location in Zürich and Anapaya configured it, while Swisscom installed a 1 Gbit/s Anapaya EDGE device in Thun. We used the setup described in 3.1. Due to cabling issues in the building in Lausanne, no connectivity was available for testing within the duration of this work. To allow for more flexibility, ETH provided an additional access in two different machines provided by the NetSec group. In the service offer to armasuisse, SWITCH explicitly states that the existing IP connection and the new SCION connection are physically separated. Sunrise and Swisscom guarantee access to their internet backbone without further details on the connection. Sunrise provided the most details about the service offering. In Zürich and Thun, the SCION links are connected to Cisco routers on site, from where they are connected to the appliance, taking a total of 2 standard height units of rack space. This is the standard deployment of SCION services provided by SWITCH and Swisscom.

3.2 Configuration

Because the SCION service is managed, Anapaya engineers configured the appliances. By default, the ISPs deliver the managed appliances in a user-inaccessible state, with only Prometheus and Secure Shell (SSH) services exposed. The SSH service is restricted with public-key authentication only for security reasons to enable Anapaya to do configuration and maintenance remotely. Administrators can unlock the system only by accessing it via the Baseboard Management Controller (BMC) or physical presence and the credentials which Anapaya made available on request. The Anapaya EDGE devices run an Ubuntu 18.04.6 LTS operating system which is supported until April of 2023 and is provided with security updates (Extended Security Maintenance) until April 2028[42]. In addition to the In-Band (via WAN) management, Anapaya requests Out-of-Band access over traditional internet connections to the machines for management purposes in case the WAN side fails. This management access is realized using a state-of-the-art WireGuard VPN tunnel.

The system services handling SCION network packets, IP-to-SCION tunneling and management are contained in separate Docker containers. Table 3.2 contains a list of the containers running within the EDGE. The individual containers have the respective SCION configuration files and directories bind-mounted. The

name	purpose category
dataplane-control	SCION service
dataplane	SCION service
router	SCION service
gateway	SCION service
daemon-<IA>	SCION service
telemetry	management
appliance-cron	management
node-exporter	management
control-<IA>	SCION service
promtail	management
dispatcher	SCION service
appliance-controller	management

Table 3.2: Docker containers running on the Anapaya EDGE device as reported by `docker ps`

configuration is modified using a REST API. The appliance also provides a Web GUI, enabling both human (Figure 3.2) and computer configuration changes. The service exposing access to the REST API and the web interface is accessible only from localhost without password protection by default, but a configuration change can expose this service also to external interfaces and create password-protected accounts. The configuration itself is stored in a single JSON file. Configuration files for the individual subservices are derived from this one file, rendering configuration backup and restore operations very easy to implement. The REST API enables users to change settings with simple cURL commands. Alternatively, the web interface provides a Visual Studio Code-based JSON editor with autocomplete to directly edit the settings, as shown in Figure 3.3. However, in the managed SCION setup, it is intended that only Anapaya itself changes the configuration. While clients can modify the settings using the API or web interface, there is no incremental or merging change system. Changes will therefore be overwritten without warning whenever Anapaya reapplies or changes the configuration, including SSH access permissions. This means that desired changes should be communicated to the technical support directly, which will then implement them. This can be limiting in experimental setups but is not an issue in production environments where clients do not want to handle configurations themselves. After the initial configuration, no further interaction is generally required.

In most use cases, the SCION IP Gateway (SIG) functionality (Figure 3.4) is desired. This service acts as a tunnel for IP packets that are wrapped into SCION packets and delivered to specified SCION addresses, enabling existing IP-based networks to be extended over SCION. To establish a tunnel between two SIGs, the operator must specify the addresses to announce to which other AS and

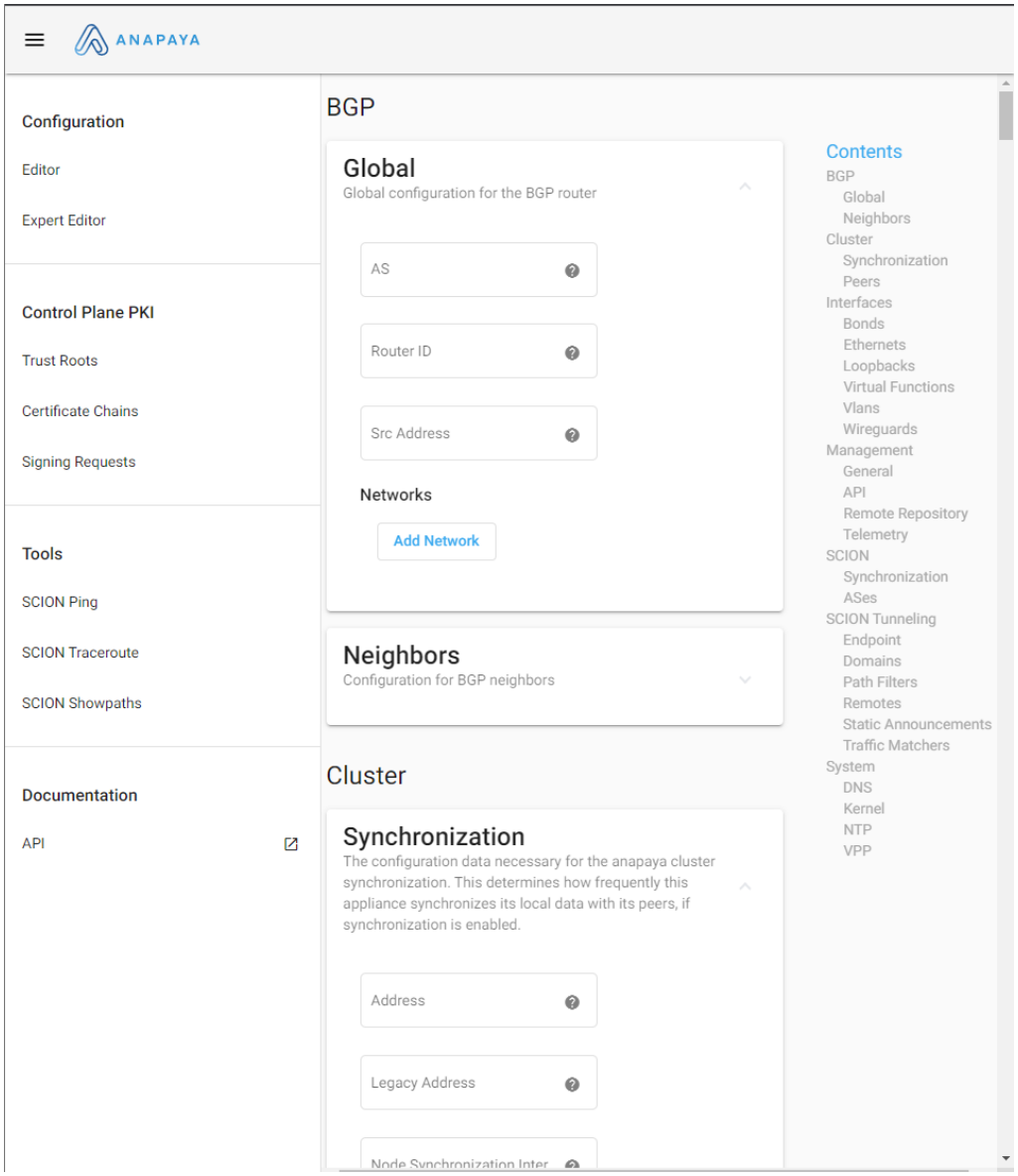

Figure 3.2: Anapaya EDGE configuration

which addresses to accept from which other AS in the gateway configuration, and the corresponding ASes must enter the matching configuration. The SIG does handle both IPv4 and IPv6 packets but may fragment and reassemble them when transitioning between IP and SCION and vice-versa, respectively. SCION/UDP packets carry data to and from SIGs. By default, the SIG does not checksum (the UDP checksum for SIG packets is zeroed), authenticate, or encrypt payload data. These properties need to be ensured by upper-layer protocols if desired. Anapaya provides a WireGuard configuration option.

Figure 3.3: Anapaya config file web editor (some information blanked)

3.3 SCION end host connections

The SCION IP Gateway does not require adjustment to individual software. This facilitates deployment in existing environments and is the currently only officially supported (and managed) way of using SCION. However, SCION can also be deployed on endhosts to give every application direct path selection control.

The source code for the endhosts is openly available at [25]. The NetSec group maintains an apt repository containing packaged binaries for SCIONLab for different architectures. The endhost stack runs in userspace and requires two services to run: A dispatcher and a daemon.

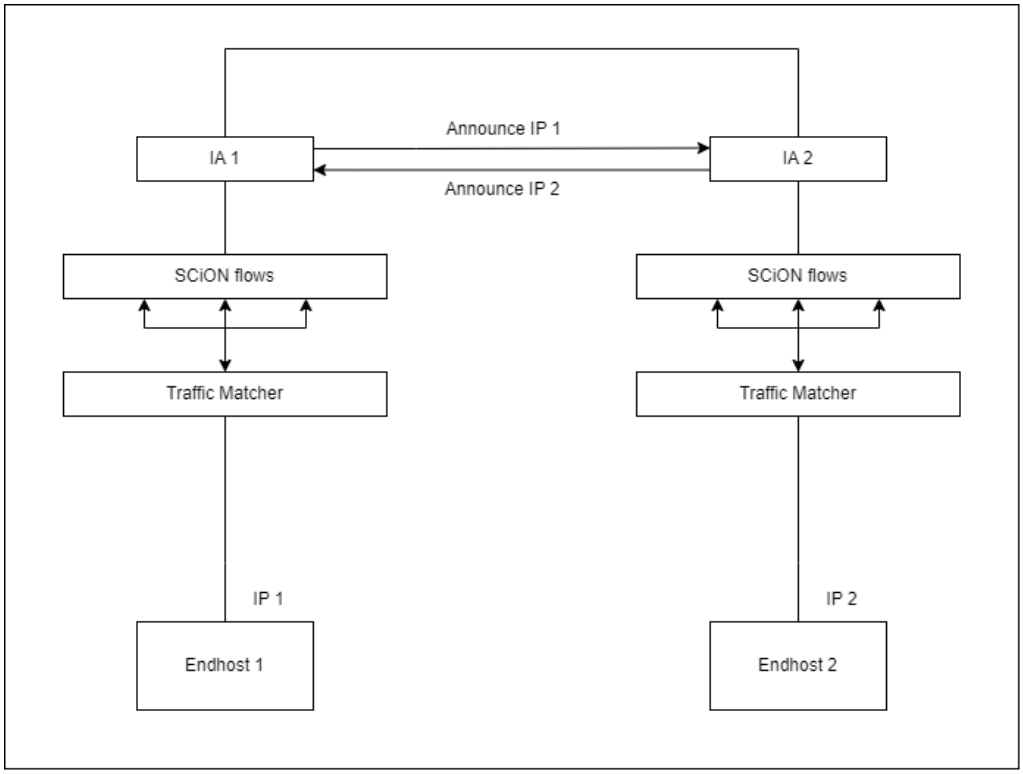

Figure 3.4: SIG functionality

3.3.1 SCION dispatcher

SCION does not carry specific layer 4 protocol data. However, to stay compatible with existing network infrastructure and to facilitate deployment, SCION packets are wrapped into a UDP underlay. This is achieved using a service program that listens on a Unix Domain Socket (UDS) and wraps and unwraps packets going to and coming from a network interface, respectively. The dispatcher service reads its configuration from `/etc/scion/topology.json` by default to determine the target addresses to wrap. In turn, the SCION Path Aware Networking library (PAN) uses an Environment variable (`SCION_DISPATCHER_SOCKET`) to determine the dispatcher to use, which removes the need for individual applications to specify the next hop themselves and allows for different dispatchers to run simultaneously. This is similar to gateway entries in routing tables. To enable the transmission of packets to an application, the dispatcher also maintains a port map to deliver incoming packets to the correct UDS.

3.3.2 SCION daemon

A SCION sending application needs to include the path of the packet at the point of sending. Because the paths do not change very frequently, but path information must be fetched by every new socket, the daemon provides a way to

cache and periodically retrieve path and key information and provide this to a single or multiple SCION-enabled applications. The daemon requires trust root configuration (TRC) information to be able to verify the authenticity of gathered network paths. The TRC search path is `/etc/scion/certs` by default. Currently, the TRCs cannot be obtained individually without a SCION network connection as there is no global openly available registry.

3.3.3 SCION bootstrapper

Both the daemon and the dispatcher require information on the network channel to use to transfer SCION packets (underlay). The daemon additionally requires cryptographic materials for authentication. While the required files can be manually created or copied to a machine, this does not scale. The scion-bootstrapper utility, provided by the NetSec group, is a program that checks different announcement methods for the presence of a configuration server and then downloads the corresponding configuration files: `topology.json` and required TRCs. The functionality is shown in Figure 3.5.

For example, to announce the presence of a bootstrap server to the bootstrap client over DHCP, DHCP option 72 is used. Option 72 specifies the "Default World Wide Web" DHCP server option, which is rarely used and can be specified as a list, which is why it can be used as an announcement option for the SCION bootstrapper.

3.4 SCION reference router

The deployment at armasuisse CYD campus uses the Anapaya EDGE as a gateway and SCION router. In SCIONLab, the open-source reference implementation is used instead. The `scion-control-service` package in the NetSec Debian repository provides the required binaries to run a border router. A production SCION deployment may use the reference router instead of relying on Anapaya for a specific router device.

3.5 Features currently not available

All features for a working SCION network are available in the commercial SCION. The commercial SCION offerings are limited to connections to static addresses of multiple participants. This works well for the specific use case of an organization aiming to connect to other organizations that also are their own AS but has limited scaling potential for residential and consumer-level use. Most prominently, there is no name resolution (like DNS) available. There have been proposals for this with RAINS and RHINE[4], but there is no existing service that would allow registering addresses similar to DNS registrars. Essentially, a user would have to remember

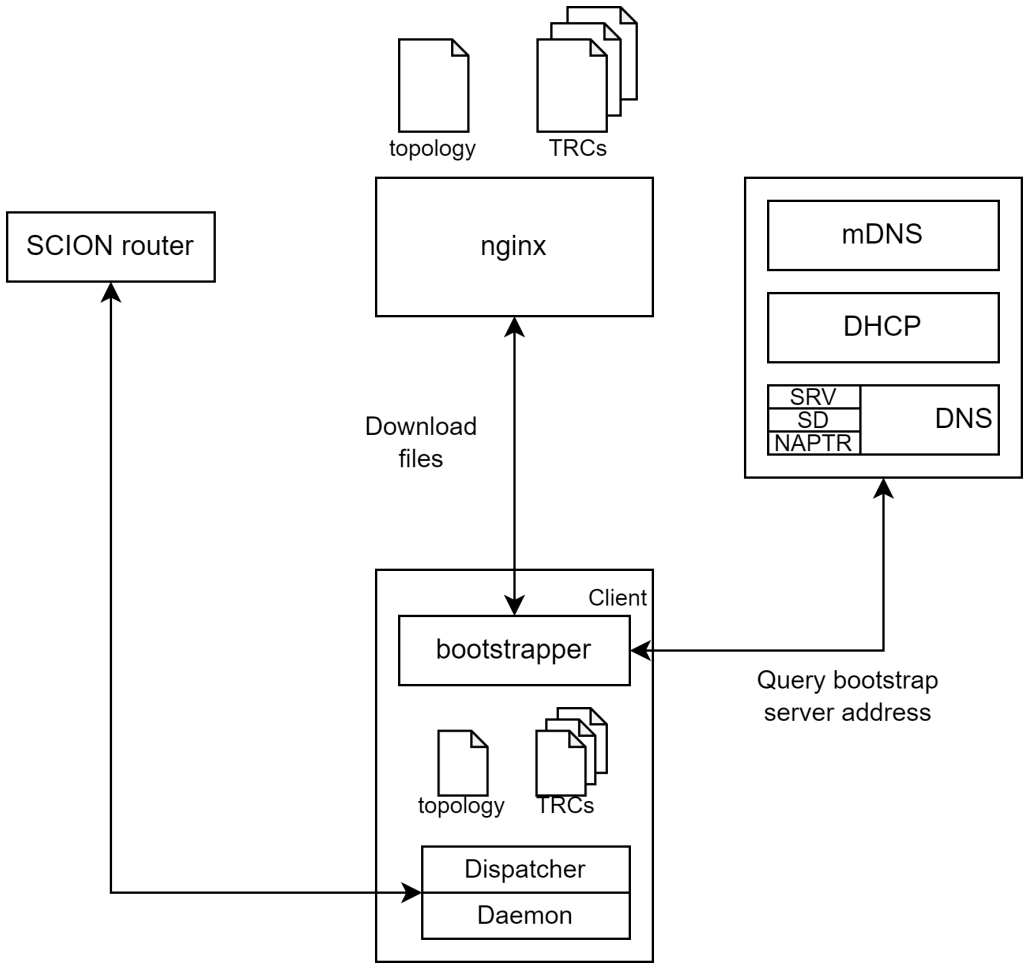

Figure 3.5: SCION bootstrapper functionality. The bootstrap client queries the bootstrap server address from one or multiple announcement channels and downloads and installs the configuration to enable dispatcher and daemon

IA and an IPv4 or IPv6 address to access a SCION service, or a client program would have to hard-code the addresses. Similarly to name resolution, there is (as of Q4 2022) no global listing of services and AS. In particular, only public ASes that are listed in ASN databases can be resolved. Currently, we observe only a few SCION-only ASes appearing on paths, but it's not possible to find out who this AS belongs to. A software tool named "OrgMan" was once announced but was not available for our testing. On the side of security, hardening measures like EPIC are not currently available.

These issues are not protocol-related. CAs and DNS registrars are third-party services that facilitate using a network technology and also were introduced late in the traditional internet. However, the absence of these services limits the applications that can currently be transitioned to SCION.

4 Setup for operation

In this chapter, we show the details of the configurations tested in this work.

4.1 SCION Endhosts

While using SCION natively within applications is the primary use of the SCIONLab, the commercial setup focuses on the use of the IP to SCION translation. This only leaves limited control to individual applications, which limits the tests that can be performed, especially as modifying the Anapaya EDGE configuration is subject to changes in the underlying network. To facilitate future use and experimentation, we automate the configuration of endhosts using the NetSec group's bootstrapper.

4.1.1 SCION bootstrapper

We install the bootstrapper server directly on the Anapaya EDGE as this provides the most direct access to the required configuration files. The TRC files can be obtained from a configured Anapaya EDGE device in path `/etc/scion/crypto/trc`. Each of the files is named `ISD_-B_-S_.trc`, where the number following "ISD" denotes the isolation domain (64 for Switzerland), the number following "B" denotes the revision (if Core AS have changed), and "S" denotes the serial (increased when extending the lifetime of a previous revision). Some of the TRC files were already expired, but they show the evolution of the ISD, as ASes joined the core. Alongside the TRC files, the bootstrapper should also distribute the `topology.json` file, containing information about the service and underlay addresses to use. This file is not present on the Anapaya EDGE, but the similarly named file `local_topology.json` contains the required information, albeit in a different structure. The different structures are provided in Figures 4.1 and 4.2. The Anapaya topology is designed to allow for multiple IA in the same file.

When the Anapaya EDGE is used as a gateway only (and not as a SCION router), these addresses are not required. While the file could be created manually, we provide a script to automatically translate the files for reproducibility even if the configuration is changed by Anapaya in the future. We test and verify that the script works in two different locations in different networks and different underlay addresses.

We did test the setup using multiple different machines. The SCION configuration was working on machines with NICs from Intel and Marvell Aquantia,

```
{
    "isd_as": {
        IA: {
            "shard": 1,
            "internal_scion_mtu": 1472,
            "control": {
                "address": Address:Port
            },
            "discovery": {
                "address": Address:Port
            },
            "router": {
                "internal_address": Address:Port,
                "interfaces": {
                "1": {
                    "address": Address:Port,
                    "remote_address": Address:Port,
                    "remote_isd_as": PARENT_IA,
                    "link_type": "parent",
                    "scion_mtu": 1452,
                    "bfd": {
                        "detect_mult": 3,
                        "desired_min_tx_interval": "200ms",
                        "required_min_rx_interval": "200ms"
                    }
                }
                }
            },
            "gateway": {
                "control_address": Address:Port,
                "data_address": Address:Port,
                "probe_address": Address:Port
            }
        }
    }
}
```

Figure 4.1: Anapaya local topology file structure

and even on a Raspberry Pi 3B+. All systems reached more than 100 Mbit/s of SCION bandwidth. The time-to-first-packet was in the order of a few seconds after installation, except on systems with very slow system drives, like the Raspberry Pi, where setting up the internal databases is slowing down the system. Once paths have been fetched, performance is largely unaffected by slow system drives.

```
{
    "isd_as": IA,
    "mtu": 1472,
    "attributes": [],
    "control_service": {
        IA: {
            "addr": Address:Port
        }
    },
    "discovery_service": {
        IA: {
            "addr": Address:Port
        }
    },
    "border_routers": {
        IA: {
            "internal_addr": Address:Port,
            "ctrl_addr": Address:Port,
            "interfaces": {
                "1": {
                    "underlay": {
                        "public": Address:Port,
                        "remote": Address:Port
                    },
                    "bandwidth": 1000,
                    "isd_as": PARENT_IA,
                    "link_to": "PARENT",
                    "mtu": 1452
                }
            }
        }
    }
}
```

Figure 4.2: SCIONLab topology file structure

4.2 SCION application development

The SCION IP Gateway provides a solution that allows using SCION without modifying any node in the local network other than redirecting traffic to the SIG. This is one of the simplest forms of enabling SCION at a site, but not all features of SCION can be used this way. For example, while traffic can be directed to different links by the SIG based on QoS flags, source- and destination addresses or similar, an application cannot know which paths will be taken, what the expected latency will be and it has no guarantees whether or not traffic is directed over a SCION link. For future technologies and new applications, and to make full use of path selection and SCION extensions, SCION-native connections are desirable. Currently, there is no socket family for SCION or another binding in an operating system kernel or network stack openly available. Instead, there is the SCION

Path-Aware Networking (PAN) library written in Go that interfaces with the SCION daemon and dispatcher. While it is possible to access the socket without the PAN, the non-trivial path resolution would then also be up to the user program.

The NetSec group has a few example applications which use the PAN library. For a demonstration of SCION networking at the inauguration of the armasuisse CYD Campus in Zürich, an application has been written that streams a webcam feed using SCION. While it would have been possible to read the webcam feed directly from a video device driver, it became evident that this required adjustments based on different cameras and systems. Because the main requirement was to ensure reliable operation and existing software written in a language different from Go would have required extensive development efforts due to missing bindings, the plan was made to simply relay UDP messages of existing video stream programs like VLC. VLC supports UDP MPEGTS streaming to a UDP port, as depicted in Figure 4.3.

The PAN uses mainly two different modes of communication: QUIC and raw UDP, with QUIC providing a TCP+TLS-like behavior by handling a TLS state

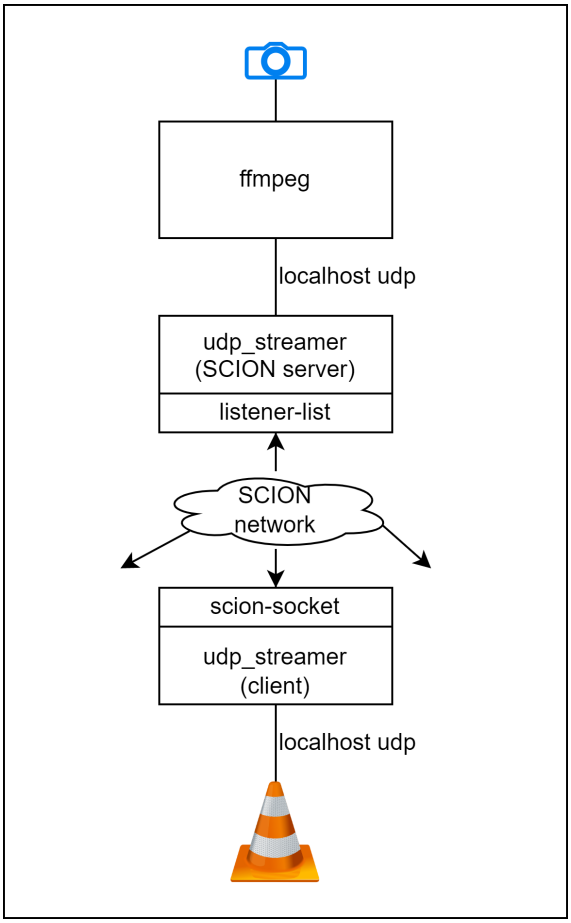

Figure 4.3: UDP Streamer as implemented for a demonstrative video stream

machine and taking care of retransmissions and raw UDP working as known from UDP/IP. For development, various dependencies limit flexibility in choice of version, with Go1.16 working as mentioned in the documentation.

Because ffmpeg will stream to a single UDP address, a system was implemented that required a client to send a single arbitrary SCION packet to the UDP relay server to be added to the receivers, and the relay would then copy all incoming UDP packets from ffmpeg to registered listeners indefinitely. Because of this design, only SCION packets leave the system.

4.3 SCION on mobile devices

Within armasuisse, there are multiple active and planned projects about and including SCION. One of them includes research on the possibility of SCION-enabled mobile phones in cooperation with Noser engineering. Mobile (roaming) devices impose additional requirements on a networking system, as local addresses as well as the routes between endpoints may change arbitrarily when moving between locations. For this work, we tested if SCION could be enabled and used on a Wifi connection. The general setup is largely equivalent to creating a normal IP-based hotspot:

1. Install and configure the `hostapd` service program to control the wireless NIC and move it to AP mode
2. Install and configure `dnsmasq` as a DHCP server (optionally, enable the integrated DNS server)
3. Set up routing as desired.

This results in a working setup for normal IP traffic. Enabling SCION on top requires a few additional steps:

1. Add DHCP option 72 as described in Secion 3.3.3 to the `dnsmasq` configuration: `dhcp-option=72,<Bootstrap-server>`
2. Ensure correct routes from the SCION router

SCION wraps source addresses twice: Once in the underlay and once in the SCION header. This works in routed setups but will break in setups that employ simple NAT, as shown in Figure 4.4. It is still possible to use a NAT by adding static routes to the subnet behind the NAT on the router, however, this effectively defeats the purpose of a NAT. It is not possible to change the source addresses because this would interfere with other SCION functionality, like source address authentication[5]. In general, a smart routing scheme and IPv6 can make NAT obsolete. While it would be possible to keep a mapping of translated addresses in the SCION router as the router will see both underlay and SCION address information, this would require additional state tables.

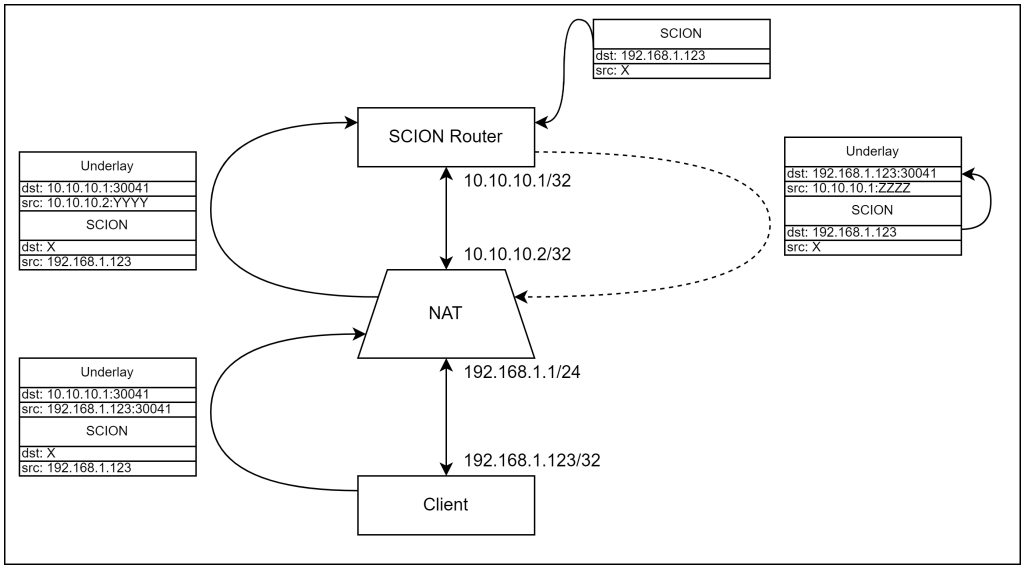

Figure 4.4: SCION Endhosts behind a NAT Access Point

Overall, SCION over WiFi works as expected when the NAT issues are worked around. Switching and moving between networks required us to restart the SCION bootstrapper, daemon and dispatcher manually in our tests, but this is likely a bug, as the bootstrapper should trigger whenever the system handles DHCP events. Operating system vendors could trivially integrate more advanced switching into network manager software. Roaming between AS has not been examined. Chuat et al[5] outline different options for widespread SCION deployment, and it will be interesting to see if (Carrier-Grade) NATs will be present in these systems. While NATs can be considered a hacky solution to the problem of IP address space exhaustion which is partially alleviated by the spread of IPv6, NATs are still very widespread[21], and Anapaya has implemented NAT in their appliances as well[2].

4.3.1 SCION on Android

The text in this subsection is only descriptive. Noser engineering did implement and debug the SCION endhost stack on Android, while armasuisse provided a SCION WiFi network to test in and the SCION application described in 4.2.

SCIONLab is available on Android phones[26]. However, it works only over a VPN connection to SCIONLab. At the time of writing no ISP offers SCION on cellular networks. For testing SCION on Android, Noser moved the SCION binaries to a Pixel 6a Android device running AOSP (Android Open Source Project). This version of Android does not include Google services, has no vendor locks and provides access to a root account. The SCION dispatcher and daemon binaries are started from the Android Debug Bridge (ADB) shell. Using `nohup`, the processes keep running even when disconnecting the USB cable. Because AOSP

is based on Linux but is not fully compatible, configuration file paths need to be adjusted using environment variables. For a demonstration, Noser compiled the UDP video stream example for Android, which worked as expected. Just like on the desktop, the application sent incoming SCION/UDP packets to a localhost UDP socket which the Android version of VLC was listening on.

4.4 SCION reference router implementation

Using our tap implementation, we can quickly switch connections similar to changing cables plugged into an L2 network switch. We set up a machine with the `scion-control-service` as well as a daemon and dispatcher with the configuration file and TRCs that we extracted from the Anapaya EDGE. Additionally, we created the required directory structure and copied keys and certificates from the EDGE. This means that we did not have to configure a system to use the Control Plane Public Key Infrastructure (CPPKI) to create Certificate Signing Requests, but as the certificates are refreshed every 18 hours, no long-term deployment would be possible in this setup. While there are install scripts specific to SCIONLab, there is little documentation on an installation and configuration of a border router from scratch and the configuration files are slightly different between the Anapaya and the NetSec implementation. As a last quirk, the SWITCH upstream router did announce its IPv6 link-local address using Unsolicited Neighbor Advertisements to a fixed MAC address. We could establish connectivity after spoofing the MAC address to the expected value. We then confirm that the setup is working by running a SCION bandwidth test on the machine running the reference SCION border router.

5 Experiments

5.1 Overview

For the second part of this work, we investigate the details of SCION packets and describe the experimental setup. The setup is designed to allow for a subset of the Dolev-Yao attacker model, i.e. provide tools to read, drop, modify or insert packets[10].

5.2 Experimental setup

5.2.1 Overview

We installed an Intel Tofino P4 programmable switch at the location in Zürich. While an implementation of SCION on P4 does exist[8], we use the P4 switch solely as a simple tap device. Armasuisse CYD Campus provided the P4 code, concept and an introduction to the inner workings of the machine. The tap works as depicted in Figure 5.1: Incoming packets are tagged with their ingress port in P4, and then passed to the CPU for further processing. The host then reads the packet, extracts the ingress port, looks up a dynamically programmed output port, updates the packet data and returns it to the switch. If the switch receives a packet with the custom *Port Protocol* header, it removes the header and sends the modified packet to the specified egress port. The received packet is recorded to a `.pcap` file for later offline analysis.

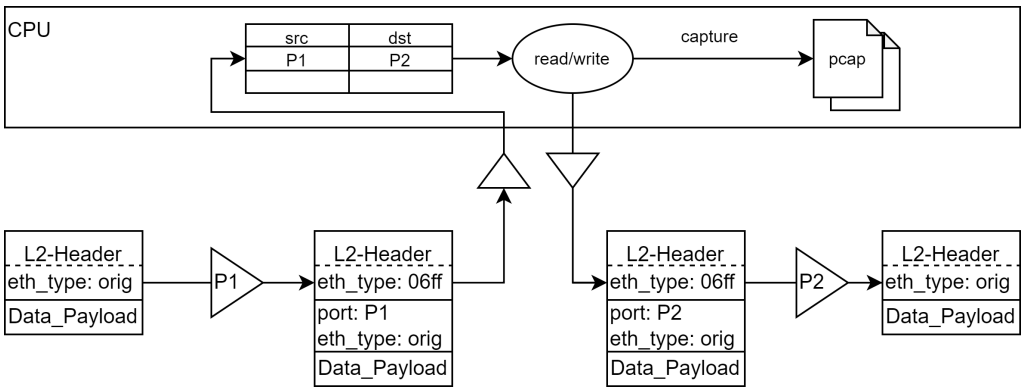

Figure 5.1: Tap device functionality

5.2.2 Host logic

A Tofino P4 programmable switch could handle most required operations in programmable hardware, in particular dropping packets, modifying packets headers, and simple forwarding based on ingress ports and packet addresses. However, while there is still potential for optimization, using a less sophisticated, but more flexible approach was chosen. For example, the switch copy could simply copy packets to the host interface for recording, it would then be more complex to implement a logic to intercept or modify packets. Because the link speeds are currently limited to 1 Gbit/s, the 10 Gbit/s switch-CPU interface is more than capable of processing the expected input bandwidth, even on more than one ports.

As a first test, a Python Scapy[35]-based program was used, but the performance was far too low at multiple hundred milliseconds of delay and few kilobits per second of bandwidth. A second iteration was implemented in an `AF_PACKET` C program, The third iteration used `recvmsg`/`sendmsg` system calls. However, even though this avoided the overhead in the kernel of setting up new structures for `sendmsg` on every call, it still was bound by the syscall performance. To fix this bottleneck, the work of both receiving and sending is distributed to multiple threads, resulting in a sufficient performance for full duplex 1 Gbit/s. However, without any synchronization, this would lead to packet reordering. Adding locks for synchronization resolves this issue, but also prevents concurrent calls to the `recv` system call, reducing performance slightly. Balancing the requirements of performance and ordering mandates a solution that provides strict ordering of packets as well as the possibility to send multiple packets per syscall. This is implemented in the last revision, which uses `PACKET_RX_RING` and `PACKET_TX_RING` to handle packets in userspace and get packets from and to the kernel in bulk.

While the tap service can record packets to a PCAP file, the latency is too high to drop a packet and send a modified version when reading from the PCAP. For a fast implementation, the tap source code is modified and headers can be parsed in C. This is a tradeoff between usability and performance, as filtering has to be implemented directly in the tap source code, reducing modularity.

5.2.3 Possible improvements

When a specific traffic flow should be examined, filtering can be offloaded to P4 directly to reduce CPU load. Care must be taken to only redirect based on flows and not individual packets to avoid reordering issues. The link speed is currently limited to a single 10G X520 internal NIC. Packets could be balanced to a second 10G X520 internal NIC in case a single one is not sufficient. The packet parsing code can be extended for more flexibility and dynamic filter inputs.

5.3 Throughput performance

As the SIG tunnels packets by wrapping them and does not otherwise coalesce or rewrite packets, the raw IP performance is an upper bound. For a 1500 Byte default Maximum Transmission Unit (MTU), we expect a payload efficiency of around 95% for UDP/IP, which can drop to 84% when carried over a SIG, as shown in Listing 1. As such, we would expect an average performance of around 840 Mbit/s on an empty 1 Gbit/s link. In our deployment, there is additional traffic for link health checks and metrics, so we would expect a slightly lower sustainable bandwidth. The documentation of SCIONLab recommends an MTU setting of as low as 1200 Bytes to ensure the best operation when running over a VPN connection that further reduces usable payload[27]. Because the SIG is designed

$$
\begin{aligned}
\eta_{UDP/IP} &= \frac{s_{MTU} - s_{IPv4} - s_{UDP}}{s_{MTU} + s_{eth}} \\
&= \frac{1500 - 20 - 8}{1500 + 38} = \frac{1472}{1538} = 95\% \\
\eta_{UDP/SCION} &= \frac{1472 - s_{SCH} - s_{SAH} - s_{PMH} - x \cdot s_{IF} - y \cdot s_{HF} - s_{UDP}}{s_{MTU} + s_{eth}} \\
&= \frac{1472 - 12 - 24 - 4 - x \cdot 8 - y \cdot 12 - 8}{1500 + 38} \\
&= \frac{1424 - x \cdot 8 - y \cdot 12}{1538} \overset{\text{x=3,y=5}}{=} \frac{1340}{1538} = 87\% \\
\eta_{SIG} &\overset{\text{x=3,y=5}}{=} \frac{1340 - s_{SIG} - s_{IPv4} - s_{UDP}}{s_{MTU} + s_{eth}} \\
&= \frac{1340 - 16 - 20 - 8}{1538} = \frac{1296}{1538} = 84\% \\
s_{MTU} &= \texttt{ETH\_MTU} = 1500\ \texttt{bytes} \\
s_{eth} &= \texttt{sizeof(Ethernet\_frame)} = 38\ \texttt{bytes} \\
s_{IPv4} &= \texttt{sizeof(IPv4)} = 20\ \texttt{bytes} \\
s_{UDP} &= \texttt{sizeof(UDP\_Header)} = 8\ \texttt{bytes} \\
s_{SCH} &= \texttt{sizeof(SCION\_Common\_Header)} = 12\ \texttt{bytes} \\
s_{SAH} &= \texttt{sizeof(SCION\_Address\_Header)} = 24\ \texttt{bytes} \\
s_{PMH} &= \texttt{sizeof(Path\_Meta\_Header)} = 4\ \texttt{bytes} \\
s_{SIG} &= \texttt{sizeof(SIG\_Header)} = 16\ \texttt{bytes} \\
s_{IF} &= \texttt{sizeof(InfoField)} = 8\ \texttt{bytes} \\
s_{HF} &= \texttt{sizeof(HopField)} = 12\ \texttt{bytes}
\end{aligned}
$$

Listing 1: Theoretical limits on payload efficiency

to integrate into existing networks with likely a default MTU of 1500 bytes, it also fragments packets by splitting them into multiple SIG-packets, which the receiving SIG then reassembles. Predicting an MTU can be difficult for applications, as changing path selections can have an impact on the header length and therefore maximum payload size.

Overall, we have the following assumptions going into the test:

1. We expect around 840 Mbit/s of usable bandwidth on a 1 Gbit/s link
2. We expect fragmentation to have an impact when sending packets that are larger than the calculated maximum payload
3. We expect UDP to achieve a higher raw bandwidth than TCP similar to what can be expected in traditional IP networks.

To avoid any interference and to avoid bottlenecks in raw hardware performance, we use a machine with a 10G NIC connected directly to the Anapaya EDGE in Zürich. The other test-endpoint is connected to the EDGE device over a 1G copper interface. The tests are done bidirectionally, to test the full performance, using iperf3. In addition, we do a cross-check with the scion-bwtester, an application provided by the NetSec group, for a SCION-native comparison. Here, we expect an efficiency of 87%.

Overall, the SCION SIG can deliver about 800 Mbit/s of goodput on a 1 Gbit/s link. However, not all tests perform as expected. Most prominently, we can see in Figure 5.2 that the SIG has a payload capacity between 1200 and 1250 bytes as there is a clear drop in performance when using larger packets. Overall, we have found an upper bound of 81% efficiency. Also, we can see in Figure 5.3 that at there is a sporadic loss at 600 to 650 bytes payload in very high bandwidths, which is likely caused by SIG packets containing two payload packets dropping in the network or causing a high fragmentation through bad alignment. One anomaly is that in one direction, a few packets would still be received despite the size exceeding the MTU. We could not pinpoint this phenomenon during our testing. The SIG can sustain a high bandwidth at low jitter, but other streams will suffer from increased latency of up to 0.5 seconds at bandwidths close to the limit of 800 Mbit/s, as determined with concurrent ICMP pings. With a link that is not fully loaded, latency as reported by ping is within the expected range of 8-12 ms, depending on the selected path. This is hinting towards a large FIFO buffering in the SIG. The results also show that the fragmentation behavior of the SIG is deterministic, but predicting a good MTU to use for SIG-paths is not easy. We can, for example, observe a clear staircase in bidirectional TCP in Figure 5.4. We suspect that different segment sizes allow for good integration for TCP ACK messages by the SIG into SIG frames containing payload, while others do not. This is highly dependent on the individual TCP implementation and timing. We think the asymmetric performance is caused by stalled ACKs in the FIFO, as noted above. Because we only set the Maximum Segment Size and a TCP implementation can

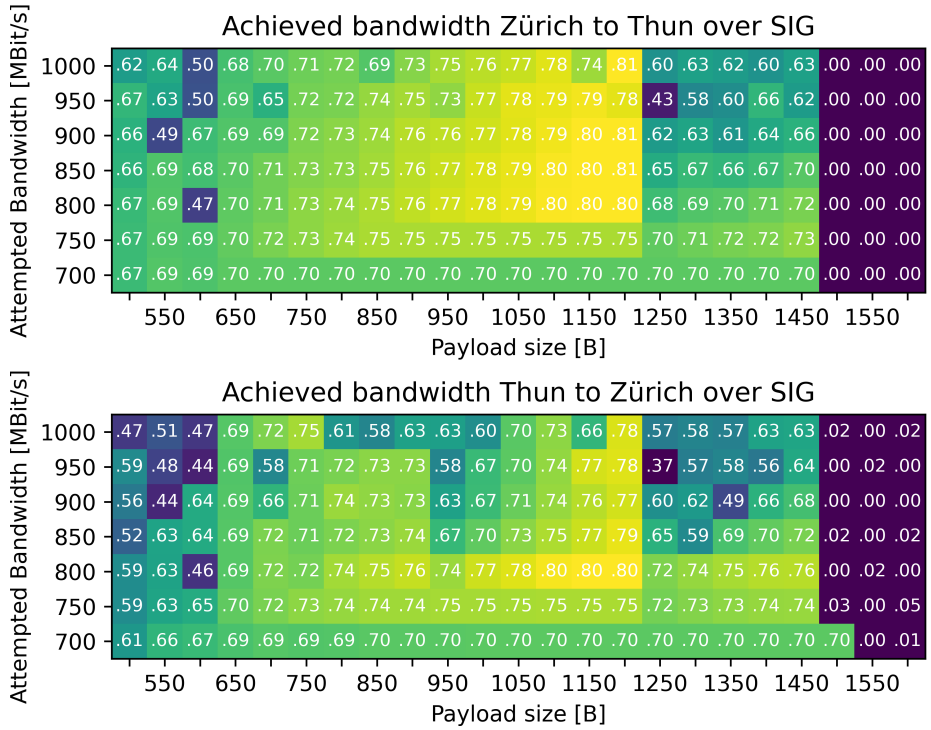

Figure 5.2: SIG performance in Gbit/s for UDP payload with varying sending bandwidth and payload size

freely choose to send smaller messages, we do still have a connection even with MSS beyond the MTU. We did not configure any traffic shaping within the EDGE.

The results from the SCION bandwidth tester do resemble traditional performance much closer: Figure 5.5 shows that increasing the payload size will also increase throughput. This indicates that individual packet processing is a bottleneck, and not the underlying network. The usable payload size is larger than with the SIG, as we would expect, as there is no SIG header to wrap and no additional IP header within the payload. As expected, packets exceeding the MTU are lost, showing that there is no support for packet fragmentation when not using the SIG. There is an anomaly where a bandwidth of 840 Mbit/s is reached according to the bandwidth tester when sending 1 Gbit/s, but not when sending at a target bandwidth of 850 Mbit/s. Looking at the source code, there seems to be a possibility for packets that are still in a buffer to be counted outside of the measurement period, causing slightly overcounting results when the target bandwidth is set much higher than the theoretical maximum. We can see in Figure 5.6 that the loss rate is rather high for all sizes, which hints to a bottleneck within the software or reference implementation, and not network forwarding capacity.

We did also repeat the measurements using the open source router (c.f. Section 4.4). We found that performance was similar when doing unidirectional

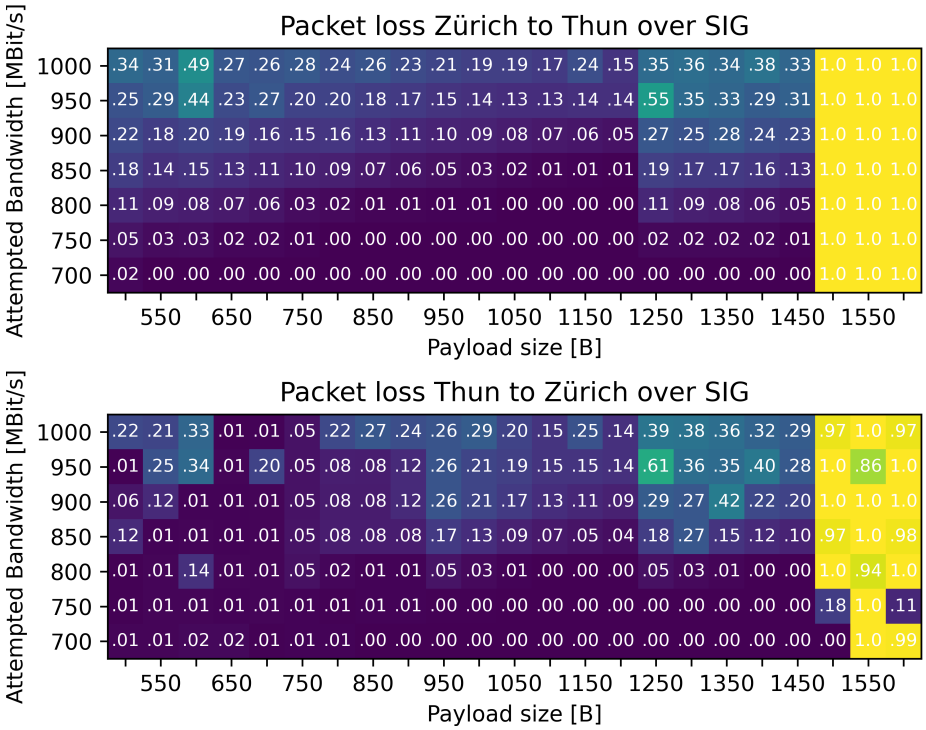

Figure 5.3: SIG performance in ratio of lost packets for UDP payload with varying sending bandwidth and payload size

tests, with 740 Mbit/s from Zürich to Thun, and 650 Mbit/s in reverse direction. In bidirectional tests, lossless communication was possible up to 400 Mbit/s bidirectionally. We did note that sending much higher bandwidths above 600 Mbit/s bidirectionally would cause subsequent paths lookups to fail. We did not observe this with the Anapaya EDGE. We believe this is caused by the daemon becoming overloaded and unable to resolve paths. The system would recover from this state automatically after a few seconds.

The Anapaya EDGE provides Intel Vector Packet Processing (VPP) optionally. Intel VPP builds on the Data Plane Development Kit (DPDK), which bypasses the kernel packet processing for lower latency and higher throughput[14]. This optimization was active during our testing on the Anapaya EDGE. Earlier tests where VPP was not enabled would yield results similar to the ones we observed in the reference implementation.

5.4 Packet reordering on the SIG

During early experiments, bandwidth over the SIG would sometimes drop as low as 1 Mbit/s for TCP iperf3 tests. UDP did not suffer such a strong performance degradation, but did lose between 10 and 20% in goodput. After a investigation by

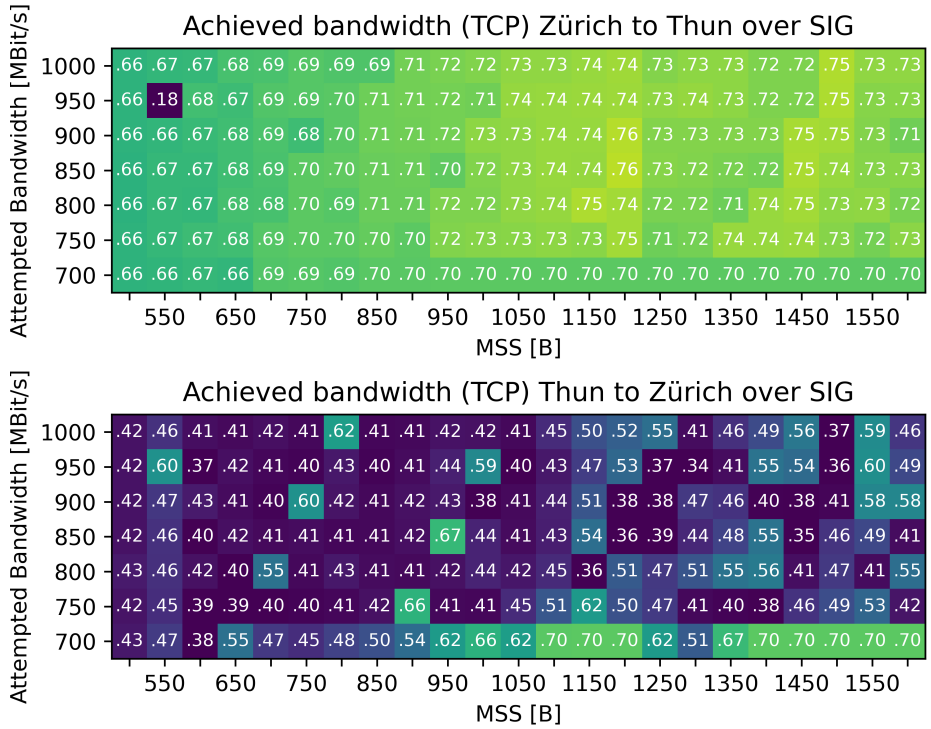

Figure 5.4: SIG performance in Gbit/s with varying sending bandwidth and MSS

Anapaya engineers, the cause was found in reordered packets on the SIG caused by an unsynchronized network tap implementation. When implementing early versions the tap software, we did not think about packet ordering requirements as generally, IP networks enforce no ordering guarantees. This can still be considered a bug for a network tap implementation, as network taps should be virtually unnoticable and have no side effects. We did fix this in later revisions as explained in Section 5.2.2, but we did continue to investigate the performance impacts. The reordering did not notably affect TCP bandwidth benchmarks from host to host over the network tap device, i.e. without passing the SIG. The conclusion for this anomaly is that the SIG does not employ a (large enough) reorder buffer or reception window and drops out-of-order packets. Local TCP passing the tap working nearly unimpeded shows that the reordering does not incur with a high delay and reordering is happening within a region small enough to fit into the TCP window. Looking at the open source implementation shown in Figure 5.7, the fragmentation reassembly buffer does only accept packets with strictly monotonically increasing sequence numbers.

Investigations in the Open Source SIG code lead to the following part in the code: As shown in the code excerpt in Listing 5.7, fragmented frames are discarded if any part arrives out of order. This only applies to frame that do not carry a single complete packet, i.e. frames that carry fragments of packets, because the program will flush all complete packets as soon as possible. As such, the likelihood of

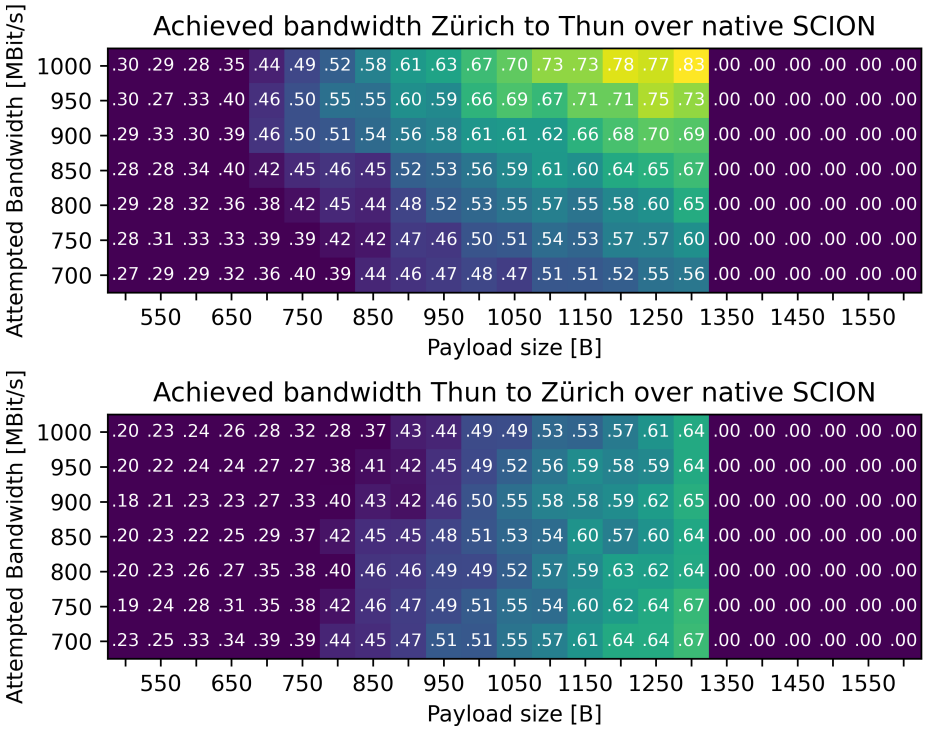

Figure 5.5: Native reference implementation performance in Gbit/s with varying sending bandwidth and payload size

this conflict increases at high bandwidths of fragmented packets. Bandwidth tests showed that reordering taps drop performance of TCP connections carried over a SIG to approximately 1 Mbit/s, which is consistent with TCP congestion control behavior. UDP performance is similarily severely impacted with a direct linear relation of reordered packets to loss in performance. We do note that even in high-load scenarios, we only saw tiny fractions of packet reorderings in our setup with comparatively short paths under normal circumstances, i.e. without a middlebox causing reordering. Therefore, we also did not see performance degradation because of reordering after eliminating the local sources of reordering.

In general, IP networks are allowed to reorder packets, even though most reordering is flow-aware to avoid disruptions[45]. While it can be argued that reordering is not to happen on links intended for SCION, adding a reception window with reordering capabilities to the SIG could increase its robustness.

To show the feasibility of the tap implementation, we provide code that executes a DoS attack simply by reordering select packets to verify this phenomenon. We confirm that we can selectively drop packets to adversely affect SIG performance without affecting other SCION traffic. We achieve this by reading the SIG packet structure and selectively exchanging the position of packets in the sending queue if for any two packets p_1, p_2 $p1[Index] \notin [\texttt{0x0}, \texttt{0xffff}] : p2[SeqNr] > p1[SeqNr]$. Index `0x0` indicates the start of a packet, and Index `0xffff` indicates the last

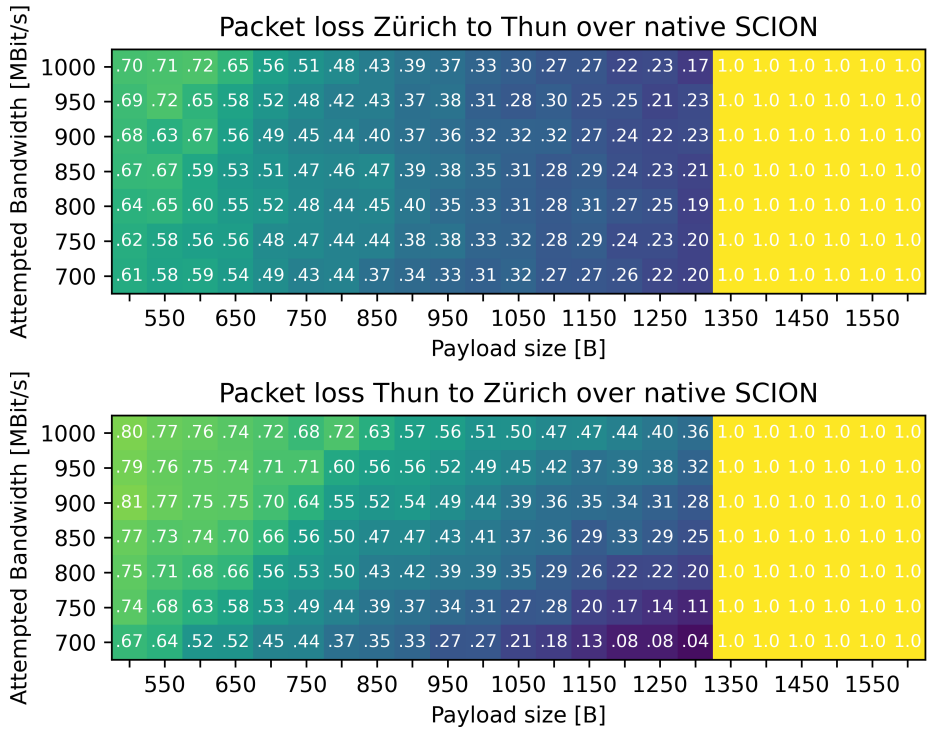

Figure 5.6: Native reference implementation performance in ratio of lost packets with varying sending bandwidth and payload size

```
// github.com/scionproto/scion/blob/5823...9e79/gateway/dataplane/rlist.go#L98
// If there is a gap between this frame and the last in the reassembly list,
// remove all packets from the reassembly list and only add this frame.
if frame.seqNr > lastFrame.seqNr+1 {
    logger.Debug(fmt.Sprintf("Detected dropped frame(s). Discarding %d frames.",
        l.entries.Len()), "epoch", l.epoch, "segNr", frame.seqNr,
        "currentNewest", lastFrame.seqNr)
    increaseCounterMetric(l.evicted, float64(l.entries.Len()))
    l.removeAll()
    l.insertFirst(ctx, frame)
    return
}
```

Figure 5.7: Code excerpt from the open source SIG. The reassembly buffer is an ordered list of pointers to a packet ring. The current section shows that packets are discarded if they do not arrive in strictly monotonic ordering

frame in a fragmented list. The code therefore exchanges any frame containing a fragment with a later-occurring frame, leading to packets in a stream being logically interleaved by one packet in the sequence if they contain a fragment. Note that that check is overly restrictive, as we could allow reordering for packets with index `0x0` as well. However, not all packets with index `0x0` are part of a fragment, and we wanted to restrict to only reordering fragments.

There are no further security implications. A reordering attacker would have to be on the same path. Another possibility is for an attacker to craft packets and inject them at the right time, sending duplicate sequence numbers in advance to cause a drop of legitimate packets with the same sequence number at the SIG, or to cause symptoms similar to reordering. This requires knowledge of the currently active sequence number. As the sequence numbers reset to 0 when a new path is chosen after a connection loss, we tried to create a fault situation with the SIG by sending packets from within an AS, masquerading as the SIG, by forging the source address to the address of the gateway and crafting a SIG packet manually. A short test of a sending application in the same AS to mimic SIG packets did not succeed, although we could not pin-point the reason for this failure. Overall, this DoS attack requires a very strong threat model that would enable many other attacks both in SCION and IP networks and cannot be considered to have direct real-world implications. However, as IP does not guarantee any order on the packets, an attacker on the underlay would be within specifications and localization could be difficult. This is not an attack on the protocol itself, but rather on a (primary) application running on SCION. Nevertheless, allowing for a small reordering window could avoid spurious packet drops and increase goodput at a small latency cost in reordering networks. Setting the sequence number to a random starting value could limit the attack potential of actors that can force a SIG to restart.

5.5 Future work

The testbed is intended to be used for futher experiments. The setup with a Tofino P4 programmable switch allows to also test P4-native SCION implementations with little modification of the current setup. The setup allows for verification of both LAN and WAN side traffic for testing of SCION extensions like the SCION Packet Authentication Option (SPAO) or EPIC. The ability to record traffic enables traffic pattern analyses for encrypted data streams. In addition, the environment provides options to develop and test SCION-native applications, for example web servers. Currently, the installation is limited to a 1 Gbit/s path bandwidth, future work can test higher bandwidths and investigate potential throughput or reliability gains provided by path reservation systems like Colibri[11]. Similarly, it could be evaluated if a single host can effectively perform temporal lensing attacks[32] by utilizing the latency difference between different paths.

6 Conclusions

6.1 Experimental testbed

We installed SCION at two out of three locations. At the site in Zürich we verified host setup and mobile device feasibility. In a demonstration, we have shown that SCION can be used to carry specific data over selected paths. We provide an analysis of the performance characteristics when using the SIG and verify the achievable goodput. We setup a Tofino-based test environment and have shown that it can be used to successfully verify the performance impact of select packet reorderings in the SIG protocol. While we did not conduct tests testing the security mechanisms, further work can build on the provided documentation and installation. We provide this testbed to test both SCION-native and SIG-based use cases in a production network.

6.2 Final thoughts

In the time of this work, we could investigate the current state of the commercial SCION service currently available for purchase. Currently, these services fit into a category of Business-to-Business (B2B) networks on a similar level to SD-WAN and MPLS-based IPVPN solutions while providing ISP-agnostic connectivity. The Anapaya support staff was quick to respond and outages caused by experimentation were quickly reported in their managed service. As other users report, e.g. [40], the product is still new. This in particular affects knowledge resources and applications that can actively profit from SCION properties like path selection beyond a SIG. Similary, security applications like SCION-specific firewalls are not yet available. The SCION protocol already has different optional extensions with different levels of support in existing networks, and different configuration file structures as shown in this work. This early fragmentation of the ecosystem can lead to confusion, and some initiatives like the SCION association are trying to counteract this and ensure standardization and support interested parties in transitioning to a SCION network[38]. Not all ways of connecting to a SCION network that are offered particularily when transitioning provide the same security guarantees, and depend heavily on the underlay. Once the efforts of standardization currently ongoing are completed, the most prominent inaccessibilities will resolve.

With some features currently still in development, the armasuisse CYD Campus SCION test environment will allow to test developments and changes that are still to come.

Bibliography

[1] 2STiC. Welcome to the 2stic programme. https://web.archive.org/web/20230109064922/https://www.2stic.nl/, 2023.

[2] Anapaya. Appliance release v0.32. http://web.archive.org/web/20230118122625/https://docs.anapaya.net/en/latest/release-notes/v0.32/, 2023.

[3] Anapaya. Leading swiss organizations are adopting scion at a rapid rate. https://web.archive.org/web/20230115190121/https://www.anapaya.net/swiss-cyber-security-days-2022, 2023.

[4] L. Chuat, M. Legner, D. Basin, D. Hausheer, S. Hitz, P. Müller, and A. Perrig. Rhine: Secure and reliable internet naming service. In *The Complete Guide to SCION*, pages 431–459. Springer, 2022.

[5] L. Chuat, M. Legner, D. A. Basin, D. Hausheer, S. Hitz, P. Müller, and A. Perrig. The complete guide to scion-from design principles to formal verification, 2022.

[6] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[7] Cloudflare. Cloudflare radar. https://radar.cloudflare.com/.

[8] J. de Ruiter and C. Schutijser. Next-generation internet at terabit speed: Scion in p4. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, pages 119–125, 2021.

[9] S. Deering and R. Hinden. Rfc2460: Internet protocol, version 6 (ipv6) specification, 1998.

[10] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983.

[11] G. Giuliari, D. Roos, M. Wyss, J. A. García-Pardo, M. Legner, and A. Perrig. Colibri: a cooperative lightweight inter-domain bandwidth-reservation infrastructure. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, pages 104–118, 2021.

[12] C. Hähni. Design and implementation of functionality, security and maintainability enhancements for scionlab coordination service. *BSc Thesis*, 2017.

[13] R. Hinden and A. Sheltzer. DARPA Internet gateway. RFC 823, Sept. 1982.

[14] Intel. Intel vpp. `https://www.intel.com/content/www/us/en/developer/articles/technical/build-a-fast-network-stack-with-vpp-on-an-intel-architecture-server.html`, 2022.

[15] International Telecommunication Union. Measuring digital development. `http://web.archive.org/web/20230116131215/https://www.itu.int/en/ITU-D/Statistics/Documents/facts/FactsFigures2021.pdf`.

[16] J. Kwon, J. A. García-Pardo, M. Legner, F. Wirz, M. Frei, D. Hausheer, and A. Perrig. Scionlab: A next-generation internet testbed. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pages 1–12. IEEE, 2020.

[17] M. Legner, T. Klenze, M. Wyss, C. Sprenger, and A. Perrig. {EPIC}: Every packet is checked in the data plane of a {Path-Aware} internet. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 541–558, 2020.

[18] K. Lougheed and Y. Rekhter. Border Gateway Protocol (BGP). RFC 1105, June 1989.

[19] K. Lougheed and R. Y. Border Gateway Protocol (BGP). RFC 1163, June 1990.

[20] K. Lougheed and R. Y. Border Gateway Protocol 3 (BGP-3). RFC 1267, Oct. 1991.

[21] G. Maier, F. Schneider, and A. Feldmann. Nat usage in residential broadband networks. In *International Conference on Passive and Active Network Measurement*, pages 32–41. Springer, 2011.

[22] J. McQuillan, G. Falk, and I. Richer. A review of the development and performance of the arpanet routing algorithm. *IEEE Transactions on Communications*, 26(12):1802–1811, 1978.

[23] F. Meyer. SCION secure internet enters everyday service, 2022.

[24] D. L. Mills. Exterior Gateway Protocol formal specification. RFC 904, Apr. 1984.

[25] netsec group. Scion. `https://github.com/scionproto/scion`.

[26] netsec group. Scion. `https://web.archive.org/web/20230116163153/https://play.google.com/store/apps/details?id=org.scionlab.scion`, 2023.

[27] netsec group. Scionlab troubleshooting. `http://web.archive.org/web/20230117082812/https://docs.scionlab.org/content/faq/troubleshooting.html`, 2023.

[28] OECD. Routing security. *OECD*, OECD Digital Economy Papers(330), 2022.

[29] A. Perrig, P. Szalachowski, R. M. Reischuk, and L. Chuat. *SCION: A Secure Internet Architecture*. Springer, 2017.

[30] J. Postel. Internet protocol. Technical Report 791, 1981.

[31] J. Postel. NCP/TCP transition plan. RFC 801, Nov. 1981.

[32] R. Rasti, M. Murthy, N. Weaver, and V. Paxson. Temporal lensing and its application in pulsing denial-of-service attacks. In *2015 IEEE Symposium on Security and Privacy*, pages 187–198. IEEE, 2015.

[33] Y. Rekhter, S. Hares, and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 4271, Jan. 2006.

[34] L. Roberts. The arpanet and computer networks. In *A history of personal workstations*, pages 141–172. 1988.

[35] R. Rohith, M. Moharir, G. Shobha, et al. Scapy-a powerful interactive packet manipulation program. In *2018 international conference on networking, embedded and wireless systems (ICNEWS)*, pages 1–5. IEEE, 2018.

[36] L. Schumann, T. V. Doan, T. Shreedhar, R. Mok, and V. Bajpai. Impact of evolving protocols and covid-19 on internet traffic shares. *arXiv preprint arXiv:2201.00142*, 2022.

[37] SCIED Team. Scion infrastructure for the eth domain. `https://web.archive.org/web/20230109065451/https://scied.scion-architecture.net/`, 2023.

[38] SCION Association. Scion standardization. `https://web.archive.org/web/20221129155018/https://www.scion.org/standardization`, 2022.

[39] SIX. SIX info-center. `https://web.archive.org/web/20221229124742/https://www.six-group.com/de/products-services/banking-services/interbank-clearing/info-center.html`, 2022.

[40] B. Stump. Introduction to the secure swiss finance network. `https://web.archive.org/web/20230103075554/https://www.six-group.com/dam/download/banking-services/events/2022/ssfn.pdf`, 2023.

[41] C. Timberg. The long life of a quick "fix". `https://web.archive.org/web/20150601035758/http://www.washingtonpost.com/sf/business/2015/05/31/net-of-insecurity-part-2/`, 2015.

[42] Ubuntu documentation. Releases. `https://web.archive.org/web/20230101204252/https://wiki.ubuntu.com/Releases`, 2023.

[43] R. Y. and L. T. A Border Gateway Protocol 4 (BGP-4). RFC 1771, Mar. 1995.

[44] X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen. Scion: Scalability, control, and isolation on next-generation networks. In *2011 IEEE Symposium on Security and Privacy*, pages 212–227. IEEE, 2011.

[45] X. Zhou and P. Van Mieghem. Reordering of ip packets in internet. In *International workshop on passive and active network measurement*, pages 237–246. Springer, 2004.

A An Appendix

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Setup of a national SCION testbed for cyberdefence

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Niederer	Silvan

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 20.01.2022	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.