DoCile: Taming Denial-of-Capability Attacks in Inter-Domain Communications

Marc Wyss, Giacomo Giuliari, Markus Legner, and Adrian Perrig
Department of Computer Science, ETH Zurich, Switzerland
{marc.wyss, giacomog, markus.legner, adrian.perrig}@inf.ethz.ch

***Abstract*—In recent years, much progress has been made in the field of Internet bandwidth reservation systems. While early designs were neither secure nor scalable, newer proposals promise attack resilience and Internet-wide scalability by using cryptographic access tokens (*capabilities*) that represent permissions to send at a guaranteed rate. Once a capability-based bandwidth reservation is established, the corresponding traffic is protected from both naturally occurring congestion and distributed denial-of-service attacks, with positive consequences on the end-to-end quality of service (QoS) of the communication. However, high network utilization—possibly caused by adversaries—can still preclude the initial unprotected establishment of capabilities. To prevent such denial-of-capability (DoC) attacks, we present DoCile, a framework for the protection of capability establishment on Internet paths, irrespective of network utilization. We believe that DoCile, deployed alongside a capability-based bandwidth reservation system, can be the foundation of the next generation of secure and scalable QoS protocols.**

***Index Terms*—Denial-of-Capability Attack, DDoS Resilience, Bandwidth Reservation**

I. Introduction

With the continuously increasing dependence of our society on Internet communication, denial-of-service (DoS) attacks represent an ever-growing menace. From the first reports of a DDoS attack in 1996 [3], to the daily incidents we witness today, networking researchers and practitioners alike have tried to find a way to put an end to DDoS attacks. Unfortunately, despite the vast literature on DDoS protection [36], the quest for a definitive solution continues.

One line of research—which has seen renewed interest in recent years—proposes to use *network capabilities* to achieve bandwidth reservations with strict delivery guarantees. Capability-based protocols authorize communication by means of cryptographic tokens (capabilities) that are included in each packet and can be efficiently checked throughout the network. These tokens contain information on the size (in terms of bytes or packets per second) of the authorized flows. They thus allow a scalable way to monitor and enforce bandwidth usage: under congestion—be it naturally occurring or generated by a DDoS attack—the flows of legitimate users with an existing capability remain undisturbed, while packets without capabilities are dropped when congestion arises.

The Weakest Link. In the *capability establishment* process, the source sends a capability-request packet toward the destination. The forwarding elements on the path (e.g., the border routers) verify the legitimacy of the request and approve it by adding preliminary tokens to the packet. The destination grants the reservation and sends the packet back to the source, collecting the finalized tokens, which the source then uses to prioritize and protect subsequent packets. These two initial packets are unprotected, and forwarded as best-effort traffic: they thus represent the *weakest link* in the security of capability-based protocols. In a denial-of-capability (DoC) attack an adversary causes congestion in such a way that these establishment packets are dropped, nullifying the effort to protect any subsequent communication. Without addressing DoC attacks, none of the systems in the literature can provide sufficient guarantees to mitigate DDoS.

The Capability Hierarchy. A handful of proposals against DoC are present in the literature. However, as we argue in Section VIII, all these systems provide insufficient DoC protection for one or more of the following reasons:

- They only provide probabilistic guarantees, meaning that capability establishment can be delayed indefinitely.
- They rely on connectivity to external services (CDN, DNS), which can also be interrupted by DDoS attacks.
- They do not consider proper authentication of the parties involved, enabling spoofing attacks.
- They require substantial functional extensions for routers.
- While they cannot sufficiently protect the forward path, protection of the return path was not considered at all.

To try to mitigate these problems, recent publications have introduced another level of capabilities—established between autonomous systems (ASes) instead of hosts—to protect and enhance the scalability of host-to-host (HtH) capabilities: AS-to-AS (AtA) capabilities are created first, and specify the total amount of bandwidth that the two AS-endpoints allocate to the communication between their hosts. HtH capabilities are then derived from this pool of resources.

From a performance and management perspective, layering capabilities is beneficial. First, it greatly reduces the number of parties involved in the capability exchange, from the tens of billions of Internet hosts [10], to the tens of thousands of ASes [23]. Second, it allows the protocol to leverage the centralized control ASes have over their networks and the existing business relationships between neighboring ASes. Finally, AtA capabilities can be longer-lasting than HtH, since AS-level traffic patterns can be structured as long-term connections. All these properties increase the efficiency of capability-based

978-1-6654-6824-4/22/$31.00 ©2022 IEEE

reservations and reduce the surface for DoC attacks.

However, these attacks are far from neutralized. While HtH capability establishment is now protected by AtA reservations, the establishment of AtA capabilities remains vulnerable. This seems to be an unsolvable bootsrapping problem: adding other levels of capabilities will just shift but not resolve the *weakest link* in capability establishment. Further, existing systems assume the availability of pre-shared symmetric keys, and do not consider—nor protect against—attacks on key exchange protocols. We argue that this assumption cannot hold in a public Internet with tens of thousands of ASes, and hence attacks on such protocols are a fatal and yet unaddressed vulnerability in capability-based bandwidth reservation systems.

DoCile. To finally break free of the capability bootstrapping and key-exchange problems, we design DoCile[1], a system that *does not itself rely on capabilities*, and is therefore the missing piece to complete the capability hierarchy presented in Figure 1. DoCile's key idea is that pre-established protected communication channels between neighboring ASes—which have a business relationship and some level of trust—can be stitched together to protect low-rate control traffic between distant ASes. More specifically, DoCile introduces two basic building blocks: (i) setup-less neighbor-based communication (SNC), a system to protect communication between neighboring ASes; and (ii) telescoped reservation setup (TRS), a recursive process to protect key and capability establishment.

The composition of SNC, TRS, and a secure AtA capability-based bandwidth reservation protocol provides every AS a small but guaranteed packet rate over any routable path. This rate is enough to carry an AtA capability request and subsequent response, shielding this crucial first step from congestion and thus preventing DoC. Starting from those few but guaranteed setup packets, the capability hierarchy rapidly enables the establishment of HtH capabilities and thus highly available communication at gigabit rates.

In summary, our main contributions are the following:

- Based on the observations stemming from our analysis of the capability-hierarchy we design DoCile, the first system to completely and effectively mitigate DoC attacks.
- While previous literature makes the simplifying assumption that symmetric keys are pre-shared between ASes, we design DoCile to also protect key exchanges from DDoS. This removes a further avenue for DoC.
- Through our security analysis, we show that DoCile can guarantee the successful establishment of AtA capabilities at Internet scale, is free from circular dependencies, and is not susceptible to DoC-type attacks.

II. The Capability Hierarchy

In this section, we present the capability hierarchy illustrated in Figure 1 and the concepts and mechanisms that are important at each of its levels.

[1]DoCile is a portmanteau of the acronym DoC and the English word *docile*, meaning tame, submissive.

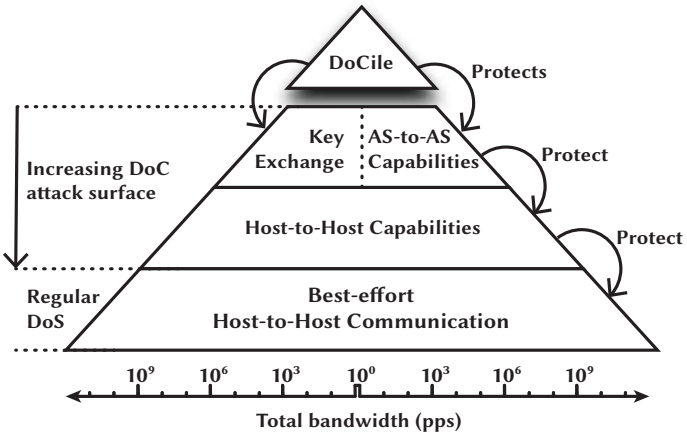

Fig. 1. **The Capability Hierarchy.** Every level is protected by the level above: DoCile protects the key exchange from DDoS and the AS-level capability setup from DoC attacks. The AS-level capabilities form a bandwidth reservation over which host-level capabilities can be exchanged. The host-level capabilities constitute the final bandwidth reservation protecting host-to-host communication from DDoS attacks. The width of each level approximately represents its aggregate bandwidth requirements in packets per second (pps).

A. Level 1: Host-to-Host Capabilities

While network capabilities can be used to protect various network functions, we focus specifically on *capability-based bandwidth reservation systems*.

Operating Principles. Despite the heterogeneity of proposed capability-based reservation systems [4], [11], [21], [32], [34], [35], we emphasize some of their common traits:

T1 *Goal of Capabilities*: As noted by Anderson et al. [1], the goal of these systems is to prioritize desired packets, thus preserving communication in case of congestion and DDoS attacks. This is achieved by extending packets with capabilities, i.e., tokens that represent an authorization to communicate over a certain path.

T2 *Involvement of Intermediate Entities*: All forwarding entities on the path between source and destination must participate in the monitoring of capabilities to achieve the availability guarantees. Otherwise, an adversary could target and congest any intermediate hop that does not prioritize capability-bearing packets. Therefore, all of the above protocols require endpoints to communicate with intermediate systems to disseminate and collect the capability tokens. In most protocols, this communication takes the form of a *capability request* packet, from source to destination, aggregating preliminary information on the path—e.g., the maximum bandwidth available—and a *capability confirmation* packet, which travels from the destination back to the source, collecting the capability tokens from intermediate entities. The protocols differ in the choice of *who* the intermediate entities are: Some systems require all on-path routers to support the validation of capabilities, while others consider entire ASes as one intermediate unit. The advantage of this latter system is that monitoring and enforcement need to be deployed only at border routers, improving scalability.

T3 *Unforgeability*: Capabilities must be unforgeable to prevent adversaries from constructing bogus capabilities. They therefore include the output of cryptographic operations—e.g., signatures, hashes, or message authentication codes (MACs)—over information that identifies the communication such as the source and destination addresses and the maximum allowed rate of communication. An entity with the appropriate key material can then verify that each packet is authorized, and monitor that the sender does not exceed the allowed bandwidth. Only the owner of a capability (the end point who requested it) is allowed to use it for sending packets. To prevent adversaries who can observe network traffic from misusing capabilities, these capability-based systems either include freshness mechanisms or rely on an additional source-authentication mechanism.

T4 *Efficient and Stateless Capability Verification*: As capabilities must be verified at each intermediate forwarding node, the verification must be extremely fast to allow line-rate processing. Therefore, most proposals define capabilities in terms of MACs or hashes, which are highly efficient and can be computed in dedicated hardware on most modern CPUs. Moreover, the state at intermediate nodes must be kept contained in order for the system to scale and to avoid state-exhaustion attacks. To achieve this, either (i) the length and frequency of transmission of tokens are such that the state overhead is small [1], or (ii) the whole token can be generated starting from the contents of the packet headers [4], [11], [32], [34], [35].

T5 *Direction of Communication*: Capability-based reservations can be unidirectional [11], [21], [32], [34], [35], meaning that only the communication from the source to the destination is protected, or bidirectional [4], where the reverse traffic is not forwarded using best-effort, but also prioritized based on the capabilities. In principle, some unidirectional protocols, such as Colibri and GLWP, can be extended with support for bidirectional reservations.

T6 *Protected Renewal*: An existing reservation can itself be used to renew the capabilities (i.e., request new capabilities) if the current ones are expiring in the near future. As soon as a reservation is established, it is thus protected from DDoS over an arbitrary time window.

Real-world Deployment. Early HtH capability-based systems were hardly deployed in practice due to several limitations. In contrast, modern protocols have improved security, scalability, and manageability. Colibri [11] for example, a reservation protocol based on SCION [26], ensures that the allocated bandwidth of a legitimate flow does not diminish below the minimum allocation when the number of bots increases—a property called *botnet-size independence*. The legitimate flow is thus protected against an arbitrary number of bots. Colibri is actively being deployed in the SCIONLab global research network [20].

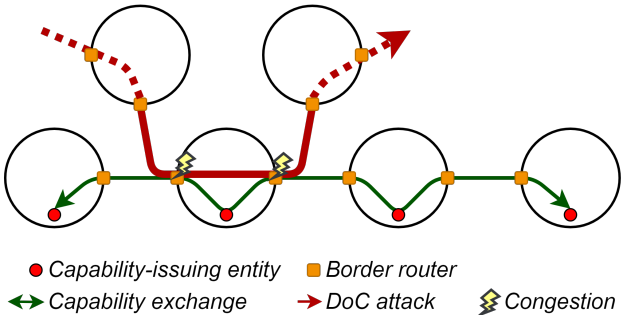

Fig. 2. **The DoC Attack.** During the capability setup, routers are flooded with excessive amounts of traffic, leading to the capability request or response being dropped. Autonomous systems (ASes) are depicted as black circles.

B. Denial of Capability

A capability-based protocol is only as resilient as its *capability establishment* phase: under a DDoS attack, the unprotected request and confirmation packets are dropped, preventing the establishment of a capability (see Figure 2).

Such a DoC attack thus limits the usefulness and applicability of capability-based protocols. However, as noticed by Parno et al. [25], this does not mean that protecting against DoC is exactly the same as protecting against DDoS:[2] With a capability protocol in place, only two packets have to be protected to achieve strict guarantees for the whole communication. This intuition is at the core of the motivation for the second level in the capability hierarchy.

C. Level 2: AS-to-AS Capabilities

Recent publications propose to protect the first level of capabilities with a second one [4], [11], [31]. Level 2 capabilities operate similarly to HtH capabilities described in Section II-A, but are established exclusively between ASes, and are used to protect the level 1 capability-establishment phase. They introduce the following improvements over level 1 protocols:

- *Reduced DoC Surface*: AtA reservations can be longer-lasting (on the time-scale of multiple minutes or hours), as the aggregate of hosts within ASes continuously generate traffic. In contrast, HtH reservations are short-lived connections that last only tens of seconds to support dynamically changing requirements. These long-lasting level 2 reservations partially mitigate DoC against the level 2 capability exchange, as they require the adversary to plan the attack ahead of time.
- *Scalability*: The efficiency of the whole system improves thanks to the layered approach. In SIBRA [4], for example, the level 2 capability (called *steady paths*) precomputes parts of the tokens for the subsequent level 1 capabilities, such that the overhead for intermediate ASes to admit additional HtH (*ephemeral paths*) reservations is negligible. Moreover, the bandwidth requirements for establishing level 2 capabilities is minimal—as low as two packets, one to request and one to return the tokens.

[2]This is the original claim by Argyraki and Cheriton [2].

- *Simplified Billing*: AtA systems can leverage pre-existing contracts between ASes to manage billing for the bandwidth reserved. In these aspects, level 2 capabilities are superior to level 1 capabilities, for which it is often unclear how billing should be performed.

Key Establishment. The exchange of shared symmetric keys is a central part of most secure AtA reservation protocols. Reservation protocols need to authenticate the source and content of setup packets to every on-path AS, and some of them securely (i.e., authentically and confidentially) send back capabilities to the reservation source. This can only be done efficiently using symmetric cryptography, because setup or renewal requests can happen frequently. Therefore, the source needs to establish shared symmetric keys between itself and every on-path AS. Key-establishment protocols such as PISKES[29] or IKEv2[18] protect their communication using signatures based on a PKI. Passport[22] does not need to actively exchange keys between ASes, but generates them by attaching Diffie–Hellman key values to routing advertisements.

Open Attack Surface. Despite the benefits highlighted in the previous section, the DoC threat is present at level 2 as well. Although attacks against level 2 capability establishment are harder to achieve than attacks on HtH capabilities, they are also more effective: Denying AtA capabilities implicitly means denying all underlying HtH capabilities. Ideally, ASes would set up level 2 reservations in advance, at a point where network utilization is low. As reservations come at a cost (on-path ASes are not expected to provide this reservation service free of charge), there is little motivation for a source AS to do so. Keeping reservations to many destinations at times when they are not actually needed results in high charges, which is even amplified when multi-path routing is used. Hence ASes want to get protected traffic ideally only at times when it is necessary. At this point, however, the high network utilization—and possible adversarial action—means that AtA capabilities cannot be reliably established anymore.

Moreover, a new type of DoC, directed at preventing key exchange, is introduced with the need for efficient source authentication. Our goal is therefore to protect the setup of keys and level 2 capabilities—without adding additional avenues for DoC—finally mitigating this attack.

III. Problem Statement

The DoC Challenge. While capabilities are an effective measure to protect traffic from DDoS, the threat of DoC attacks limits the full potential of such systems. The core of the problem can be summarized in terms of the following challenge, which we address with DoCile:

How can we protect AtA capability setup and key exchange from DoC attacks without exposing further surface for DoC?

No system so far is capable of fully solving this problem, i.e., to achieve a *100% success guarantee* (no benign request or response packet is ever dropped, despite malicious interference) on the capability setup and key exchange in an inter-domain setting. A comprehensive overview of existing systems and their shortcomings can be found in Section VIII.

Assumptions. We build DoCile atop a generic level 2 capability-based bandwidth-reservation protocol (T1). The exact specification of this protocol is not important for DoCile, but we do require that it (i) provides bidirectional reservations (T5), (ii) uses unforgeable tokens (T3), and (iii) authenticates setup and renewal requests (T6). In compliance with the description in Section II, this protocol is assumed to send a request packet traversing a *capability-issuing entity* of each on-path AS, where capabilities are added to the packet either on the forward path, on the backward path, or both (T2). This entity could, for example, be the ingress or egress border router or a dedicated service inside the intra-AS network (or a combination of multiple of these). We assume that the source AS already knows the path for which it wants to create a reservation. Furthermore, it is important that the communication and computation overhead of the reservation protocol for setup traffic is low (T4). For the key establishment protocol, we assume that the key request and response are authenticated between the requesting and the issuing AS.

Finally, we note that *path stability*—i.e., the property that network paths do not change rapidly and unpredictably in time, and cannot be maliciously altered—is required by capability-based reservation systems to achieve strong availability guarantees. Otherwise, irrespective of the capability system in use, an attacker may vanquish bandwidth reservations by hijacking traffic away from the reservation paths. Secure next-generation network architectures such as SCION [26] provide better path stability properties than to the current BGP-based Internet.

Attacker Model. We consider an attacker who can modify, drop, and inject packets anywhere in the network, except for the path for which a legitimate source AS wants to create a reservation—an *on-path* attacker can simply drop all packets, and it is inherently impossible to achieve availability guarantees in this case. [3] In our model, *off-path* attackers try to create *congestion at routers* that are part of the path on which the capabilities are exchanged by flooding them with excessive amounts of traffic (see Figure 2). We only consider such volumetric network-level DDoS and DoC attacks. In particular, DoS attacks such a packet triggering a memory corruption are out of scope and must be addressed separately through techniques ensuring software reliability and system security. An off-path attacker could be an end host, router, or any other entity including entire ASes. The attacker's objective is to prevent the source from obtaining the requested capabilities.

IV. DoCile

In this section we introduce DoCile, our framework to solve the DoC challenge from Section III.

[3] Through multi-path communication an on-path attacker can be avoided by selecting alternative routes to the destination.

A. DoCile Protocol Overview

DoCile has two main concepts: *setup-less neighbor-based communication (SNC)* and *telescoped reservation setup (TRS)*. SNC refers to pre-established bandwidth reservations between two neighboring ASes, and enables bootstrapping new reservations and extending existing ones. Assembling multiple such reservations protects the HtH or AtA capability exchange during reservation setup also on longer paths. This assembled protection mechanism works based on the assumption that the capability exchange messages are authenticated between the source and all on-path ASes.

As secure reservation protocols often assume shared symmetric keys (Section II-C), and because the preceding key exchange can also be attacked, DoCile additionally protects key exchanges from DDoS. While requesting shared keys between two *key services* may appear to be similar to requesting capabilities between *capability-issuing entities*, and that both could be protected in the same way, key-exchange protocols only need to communicate information between two end points, and on-path entities are not involved. This means that we cannot make the assumption that intermediate ASes will be able to check the authenticity of key requests, and hence assembling multiple SNCs is not possible for those protocols. This problem is solved with TRS. In TRS, the source AS iteratively fetches symmetric keys from all ASes on the path. The shared key for the source AS' neighbor is requested based on SNC, where the received key subsequently enables the source AS to create a level 2 bandwidth reservation with its neighbor. Based on this reservation, a further key request is issued, which the neighbor forwards to its own neighbor (the next on-path AS) based on SNC. The resulting key is returned using SNC and the bandwidth reservation in the backward direction. The key can then be used further to create a longer reservation. By iteratively repeating this process, the source AS ultimately has shared symmetric keys with all on-path ASes, and is in possession of all the capabilities for this path. Through this process, every capability- and key-exchange packet is guaranteed to always reach its destination.

B. Setup-Less Neighbor-Based Communication (SNC)

The idea behind SNC is to have a static reservation between two neighboring ASes, which exists without having to explicitly request or renew it. SNC is a promise of an AS to its neighbor accept and forward a certain class and amount of traffic from the neighbor with strict priority. Its purpose is to bootstrap a basic level of quality of service without already requiring any reservations, i.e., without the need of pre-distributed capabilities. We will present SNC as a unidirectional static reservation; it however becomes bidirectional when both neighboring ASes provide such a reservation to each other. To use SNC from some AS A towards its neighboring AS B, AS A marks certain outgoing packets destined to AS B. The border router of AS B, which is responsible for the communication with AS A, then limits the throughput of marked packets (AS A knows and adheres to this limit), and forwards them prioritized to its local interface, i.e., to the intra-AS network. An illustration is shown in Figure 3. Importantly, the amount of bandwidth reserved for SNC traffic has to be chosen large enough to carry setup packets even when all ASes in the Internet want to issue a capability request simultaneously; therefore, this bandwidth depends on the Internet topology. This requirement constitutes a basis for the strict forwarding guarantees that DoCile can provide. As we show in Section VII, this bandwidth is fairly low in practice. The process of marking packets serves as an abstraction, the concrete implementation of how packets are identified and policed can vary. Packet labeling can be realized without further encapsulation, for example using the DSCP (Differentiated Services Code Point), which is part of the IPv4 header and also of the traffic class field in IPv6 packets. The policing at the border router of AS B can be implemented efficiently by means of the leaky bucket or token bucket algorithm [24]. Optionally, SNC traffic can be authenticated using pre-established symmetric keys or by means of persistent TLS sessions between the neighboring ASes. All of those mechanisms are widely used in practice.

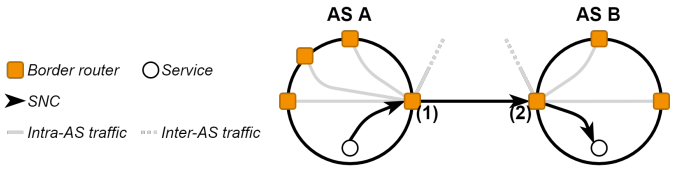

Fig. 3. **Illustration of SNC**. Grey lines denote traffic originating from or destined to other border routers of the same (solid line) or different (dotted line) ASes. To protect SNC from such traffic, border router (1) forwards SNC packets from the capability-issuing entity with strict priority, and border router (2) also forwards them with priority but only after policing them, thereby dropping packets that would exceed the SNC threshold.

C. Protecting Capability Setup

Capability requests and responses are protected in DoCile by assembling multiple SNCs (Figure 4). Thereby the source AS sends a capability request to the second AS, where the request is prioritized over all other traffic due to SNC. The capability-issuing entity of the second AS subsequently validates the authenticity of the request and its source based on shared symmetric keys, where unauthentic packets get dropped and authentic ones are extended with the granted capability. Furthermore, all requests have to be *per-source rate-limited* by this entity, where *source* refers to the AS from which the request originates, to enforce that egress SNC traffic will never exceed the SNC bandwidth threshold. An adequate choice for the capability request rate $\rho_{\text{Cap}}^{\text{Req}}$ and response rate $\rho_{\text{Cap}}^{\text{Resp}}$ is in the order of one up to a few packets per second. As the SNC bandwidth threshold is required to be high enough to carry setup packets even when all ASes at the same time issue requests at these rates (see Section IV-B), per-source rate-limited traffic guarantees that this bandwidth is never exceeded. After adding the capability to the request packet, the capability-issuing entity of the second AS forwards the packet using SNC to the third AS. This process continues until the request arrives at the last AS, which ultimately returns the response

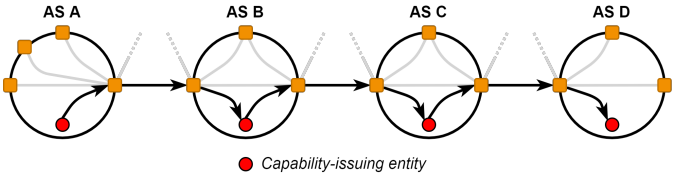

Fig. 4. **Assembling Multiple SNCs**. Every capability-issuing entity checks the authenticity of the requests. Authentic requests are per-source rate-limited, marked, and forwarded to the egress border router. The response packet is sent back following the same steps but in the opposite direction. The capability-issuing entity is illustrated as an AS-internal service, but depending on the underlying reservation protocol, it could also be directly integrated into the ingress or egress border router.

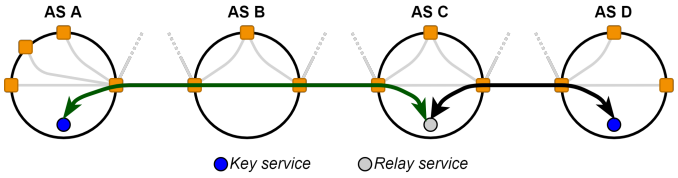

Fig. 5. **TRS Steps 2 to 4**. AS A wants to fetch a symmetric key from AS D, for which it sends a key request over the existing reservation (green line) to AS C, which forwards the request protected by SNC to AS D. The reply is again sent back over SNC to AS C, and then over the reservation to AS A.

packet back to the source, again using the assembled SNCs, but this time in the other direction. This simple mechanism thus allows every AS in the Internet to fetch from every other AS capabilities at a rate of $\rho_{\text{Cap}}^{\text{Req}}$ or $\rho_{\text{Cap}}^{\text{Resp}}$, respectively, where this exchange cannot be disturbed by DoC attacks: Malicious ASes overusing the SNC bandwidth towards a neighboring AS are prevented through traffic policing at the neighbor's router, and any attempt to exhaust a large fraction of the SNC bandwidth on remote inter-domain links through request flooding is prevented through per-source rate-limiting.

D. Telescoped Reservation Setup (TRS)

As described in Section II-C, all secure reservation protocols depend on previously established symmetric keys. Thus, also the key-exchange protocol needs to be protected from DDoS attacks. Unfortunately, reusing the approach from Section IV-C to protect the key-exchange channel by assembling multiple SNCs is not possible: key-exchange protocols are end-to-end protocols that generally do not involve intermediate entities and we cannot assume that every on-path AS authenticates the key requests and responses. Therefore, we rely on a different procedure: the telescoped reservation setup (TRS). With TRS, a source AS can reliably exchange keys and establish a reservation on the path to the destination AS. Specifically, the key exchange and reservation setup with TRS consists of multiple rounds of alternating sub-path reservation setups plus SNC extensions for the key exchange, and assembled SNCs for the reservation setup. It comprises the following steps:

1) Through SNC, the key service of the source AS fetches a symmetric key from the key service of the neighboring, i.e., second, AS on the path. Using that key, the source AS subsequently establishes a reservation to that neighbor.
2) The source AS prepares a key-exchange request message destined to the neighbor of the reservation endpoint AS. It sends this request not directly to the foreseen destination, but to the AS to which it previously established a reservation instead, where this reservation protects the key request on that path segment.
3) A *relay service* inside the last AS of the reservation receives and authenticates the request and its source, performs per-source rate limiting, and relays it to the specified neighbor. It performs this forwarding based on SNC, such that the key request is guaranteed to arrive at the key service of the destination AS; the key service then also authenticates the request and prepares the key.
4) To return the key back to the source AS, the last AS sends a reply packet back to its neighboring AS based on SNC, and the neighboring AS again uses the reservation from step 2—this time in the opposite direction.
5) Based on assembled SNCs, the source creates a reservation for the prolonged path by fetching the corresponding capabilities, which is now possible because the source is in possession of all necessary symmetric keys.
6) Repeat steps 2–5 until the source has received the keys of all on-path ASes and has established a reservation to the destination AS, i.e., fetched all necessary capabilities.

The authentication verification in step 3 is necessary to protect the per-source rate limiting against spoofing attacks. By our assumption, this check is already implemented by the bandwidth reservation protocol in use. In case the reservations on the sub-paths are no longer needed, they can either be explicitly closed through an authenticated notification, or simply not be renewed so that they will expire at the end of their validity period. A visualization of TRS steps 2–4 can be found in Figure 5.

Because we now have two protocols using SNC (for capability setup and key exchange), we either need to increase its allocated bandwidth to also take into account a key request rate of $\rho_{\text{Key}}^{\text{Req}}$ and key response rate of $\rho_{\text{Key}}^{\text{Resp}}$, or implement the per-AS rate-limiting for both protocols together, so that the rate of $\rho_{\text{Cap}}^{\text{Req}}$ is comprised of reservation requests and key-exchange requests together (and similarly for the response messages).

V. Security Analysis

Based on the attacker model described in Section III, we now analyze DoCile's resistance to DoC attacks.

Resistance to DoC. DoCile prevents DoC attacks as every step of the capability and key setup is guaranteed to succeed irrespective of any malicious interference. For the capability setup, every request is protected by the assembled SNCs. The key-exchange requests in every round of TRS are first protected by the previously established reservation, and then by SNC for the last inter-domain link. This means that every capability and key setup request always reaches its destination, and the corresponding response is again returned reliably to the requesting AS. This guarantee is independent of the attacker's traffic pattern, and no distributed attack can prevent capability establishment in DoCile. In previous systems, the probability that a packet successfully arrives is the product of

the success probabilities of all links, and therefore decreases exponentially with the length of the path. This implies that, in expectation, exponentially many request packets need to be sent to reach the destination (this applies also for the reply packets on the return path). With TRS, in every round the request and response packets are guaranteed to be forwarded to their destination. Since this is repeated N times for an N-hop path, the communication complexity of TRS is linear.

Inter-Domain Attacks against SNC. Border routers usually handle a multitude of inter-domain connections (see Figure 3). If ingress traffic from more than one connection is destined to the intra-AS network, congestion can arise at the corresponding router interface. Due to strict prioritization, SNC packets of one connection are guaranteed to be forwarded irrespective of the traffic pattern of other inter-domain connections. This might lead to ASes tagging all their traffic (destined to the intra-AS network of its neighbors) with the high-priority flag, aiming at better delivery guarantees also for illegitimate best-effort traffic. However, each border routers limit the throughput of SNC traffic for every neighbor, non-compliant ASes cannot abuse this system and can be punished for misbehavior.

Intra-Domain Attacks against SNC. If untrusted or malicious end hosts inside an AS try to mark their own packets as SNC, authorized packet marking services can be authenticated for the egress border router using MACs based on pre-shared symmetric keys. Marked unauthentic packets are then dropped. Markers could also be removed earlier at the end-host's access switch. Furthermore, an AS can internally use DiffServ [6] to avoid prioritized packets being dropped in the local network due to congestion. This measure is effective because the intra-AS network is a controlled and therefore trusted environment, which needs to be only protected from misbehaving end hosts.

Sybil Attacks. In DoCile, every AS provides a guaranteed request-rate to every other AS in the Internet. An attacker that is able to create a large number of ASes could therefore create congestion among the prioritized SNC traffic. However, this is infeasible in practice: AS numbers are issued by regional Internet registries following guidelines specified in RFC 1930 [14] through a (semi-)manual process [27]; an attacker trying to obtain thousands of new AS numbers would be detected and sanctioned. Furthermore, BGP speakers would need to be deployed and keys would need to be set up by the attacker in order to support BGP and RPKI.

Security Proof of TRS. To show the security of the TRS construction, we resort to an algebraic proof. We model the Internet as an undirected graph (V, E), representing (malicious or benign) ASes connected by inter-domain links. For some node v, $n(v)$ refers to its direct neighbors, i.e., all nodes u for which $(v, u) \in E$. We define a path $\pi = [v_0, \ldots, v_N]$ as a list of nodes, and call a path benign, if every of its nodes is benign. Furthermore, we introduce two predicates: $K(u, v)$ means that nodes u and v have shared keys, and $R(\pi)$ refers to a guarantee that a bandwidth reservation on path π can be established with certainty in bounded time. We formalize our assumptions made throughout the paper as follows:

H1 $\forall v \in V, \forall u \in n(v) : R([v, u])$
H2 Let π be a benign path. $(\forall v_i \in \pi \backslash \{v_0\} : R([v_{i-1}, v_i]) \wedge K(v_0, v_i)) \implies R(\pi)$
H3 Let path $\pi = \tilde{\pi} + [v_N]$ be benign. Then $(R(\tilde{\pi}) \wedge R([v_{N-1}, v_N])) \implies K(v_0, v_N)$

Here, H1 is the assumption that neighboring nodes deploy SNC, where we model SNC as unconditional bidirectional bandwidth reservation between two neighbors. H2 states that assembling SNCs on a benign path allows to establish a reservation if the source v_0 has shared keys with all on-path ASes. Lastly, H3 refers to a key request protected through an existing reservation and SNC on the last hop.

Theorem 1. *Let π be a benign path. Then running TRS over π results in $R(\pi)$.*

Proof. We show the statement by induction. Let $\pi_i = [v_0, \ldots, v_i]$ be a sub-path of π. For the base case, i.e., for π_1, we know through H1 that $R([v_0, v_1])$ holds trivially. Moreover, we derive $K(v_0, v_1)$ from H3. For the inductive step, our hypothesis for a specific sub-path π_i is $R(\pi_i)$ and $\forall v_j \in \pi_i \backslash \{v_0\} : K(v_0, v_j)$. We want to show that $K(v_0, v_{i+1})$ and $R(\pi_{i+1})$. The former result we get through H3 combined with H1. Together with the induction hypothesis, this gives us $\forall v_j \in \pi_{i+1} \backslash \{v_0\} : K(v_0, v_j)$. Applying this result together with H1 to H2 shows that also $R(\pi_{i+1})$ holds. □

VI. Deployment

The implementation of DoCile is simple, efficient, and scalable, and is compliant with core principles of the Internet.

A. Overhead

Extending Services and Routers. The reservation, key, and relay services need to implement per-source rate-limiting. Ideally this is integrated into the corresponding service itself, directly after the service checks the authenticity of the request, but before processing its content. Otherwise, the rate-limiting can also be performed on the processed packets, so that the service does not need to be changed. The latter approach also allows per-source rate-limiting for both the reservation and key protocols together (by forwarding traffic from both protocols to a dedicated rate-limiter), which is an alternative to increasing the SNC allocation size (Section IV-D). Rate-limiting can for example be implemented efficiently using a hash table, which maps AS identifiers to timestamps, where a timestamp denoets the last time the corresponding AS sent a request. After rate-limiting, the processed packets need to be marked and sent to the right border router, which only causes negligible overhead. At the border router, a packet marked as SNC from a neighboring AS has to be dropped in case it is not destined to the local interface. Furthermore, the border router has to forward the policed SNC traffic with priority. Policing can, for example, be implemented using a single token bucket, and queuing disciplines such as strict priority scheduling can be used for the packet prioritization.

An SNC packet originating from the intra-AS network and destined towards a neighboring AS can also be policed at the border router, but the service sending the packets has already shaped the traffic according to the SNC bandwidth size of its neighbor anyway, so this additional policing is redundant.

Assembled SNCs and TRS. A reservation setup request sent over multiple assembled SNCs will experience no significant overhead compared to a request being sent over best-effort in a congestion-free setting. But due to rate-limiting of requests at the different services, the source AS can only issue few requests at once, and hence the TRS process can take multiple seconds. However, this overhead is also negligible compared to the common key validity periods, which are in the order of hours [29]. While our description of DoCile focuses on the setup of capabilities over a single path, it is important to note that an AS may want to request capabilities over many paths. If those paths are intersecting each other, so that they share at least one on-path AS, the setup requests cannot be sent at the same time, but need to be scheduled one after the other. To reduce the overall setup duration, an AS can first try to fetch keys and capabilities by requesting them over best-effort traffic. DoCile can then be used to re-issue the requests that fail, which takes more time, but guarantees a successful setup.

Scalability of SNC. In the past, the number of ASes in the Internet has been linearly rising [5], and we do not expect that trend to come to a stop in the near future. However, as the network capacity is increasing even exponentially [19], we therefore argue that DoCile will also in the future be able to provide at least the guaranteed packet-rate that it does today, by also scaling up the SNC bandwidth accordingly.

Practicality. Apart from a secure bandwidth reservation system, DoCile only relies on packet prioritization (scheduling), policing, and rate limiting as its building blocks. These concepts are well-known and widely used in practice. Furthermore, every reservation protocol relies on packet prioritization and policing itself. Therefore rate limiting is the only feature that needs to be introduced additionally by DoCile.

B. Deployment Considerations

Incremental Deployment. Not all ASes need to switch to DoCile at the same time in order to improve resistance against DoC attacks. Even if for a certain path only a subset of ASes supports DoCile, the reservation request is already partially protected. To deploy DoCile, an AS can implement SNC for each border router independently, which also holds for the corresponding contracts with its neighbors. An AS does not need to upgrade the routers or re-negotiate its peering agreements at once.

Compliance with Provider Agreements. The Internet is based on business agreements between neighboring ASes. Also the SNC bandwidth could therefore be a part of those contracts, as no other parties need to be involved. If an AS does not want to have this (minimal) collaboration overhead, it can still implement SNC without coordinating with its neighbors. If the allocation sizes are chosen large enough, the guarantees of DoCile still hold. If they are too small so that congestion can still occur among the prioritized SNC packets, the protection against DoC attacks is still better than without DoCile in place.

Deployment Incentives. Inter-domain reservations are expected to become a business where every AS on the path receives money for its preferential treatment of reservation traffic. DoCile thus offers a positively-reinforcing revenue cycle to ASes and ISPs, which can sell reserved-traffic services to their direct customers and receive compensations from neighboring ASes to forward reservation traffic. A negative incentive is present as well: customers will move away from ASes that do not provide highly-guaranteed communication, while reservation traffic will not transit through them. This negates both direct and indirect revenue streams, making non-adoption an unsustainable business strategy.

VII. Discussion

SNC Allocation Size. An important parameter in DoCile is the size of the bandwidth allocated to SNC traffic. To provide worst-case guarantees, it has to be large enough to support a scenario where every AS sends a reservation setup request over the corresponding inter-domain link at the same time. Consequently, with around $100\,000$ ASes in today's Internet [23], reservation and key setup packets that are at most $1500\,\mathrm{B}$, and rate limiting to two requests per second (one setup request and one response, i.e., $\rho_{\mathrm{Cap}}^{\mathrm{Req}} = \rho_{\mathrm{Cap}}^{\mathrm{Resp}} = 1/s$), the necessary bandwidth amounts to only $2/\mathrm{s} \times (100\,000 \times 1500\,\mathrm{B} \times 8\,\mathrm{bit/B}) = 2.4\,\mathrm{Gbps}$. Because Internet topology and provider policies restrict the set of ASes that can actually send traffic over a specific inter-domain link, and if the size of this set is known, the corresponding SNC allocation can be further reduced. We emphasize that the allocated capacity is only completely utilized in a worst-case scenario, which is unlikely to occur in practice. In the average case, only a fraction of the reserved SNC bandwidth is actually utilized. The remaining bandwidth is not wasted however, as it can be used for best-effort traffic. Still, the SNC bandwidth between two neighbors must be chosen to be relatively low. Allocating a large fraction of the bandwidth for SNC—which is intended for low-bandwidth bootstrapping operations—will starve the bandwidth available to HtH reservations, which should be the ones carrying the bulk of traffic.

Request Rate. If the source uses the assembled SNCs to protect multiple reservation setups, it needs to wait some time before it can issue the next request in case the reservations will share a common on-path AS. If the paths are independent of each other, a reservation can be established over all of them at the same time. Fortunately, small ASes with few neighbors (e.g., non-core ASes in COLIBRI [11]) only manage capabilities for a small number of paths while large ASes are well connected and thus have access to many disjoint paths over which to send requests. Furthermore, the per-source rate limiting could be adapted to allow higher rates for direct

neighbors to not overly restrict ASes that only have a single provider. For a reservation setup over TRS for N on-path ASes ($N \geq 2$), i.e., when the source AS did not establish symmetric keys with any of the ASes yet, the reservation takes $2 \times N - 3$ seconds for the $N - 1$ key requests and $N - 1$ setup requests for (intermediate) reservations. However, this overhead is only required in the absolute worst case if all links on the intended path are congested, and the source does not have shared keys with any on-path AS. By proactively fetching keys in advance and renewing reservations, such a situation can be avoided.

Protecting Services. DoCile is primarily designed to protect the network itself, and not the capability-issuing entities or key services. In case a service is located at the border router, the router should be anyway fast enough to handle the requests at line rate. If it is deployed at a dedicated server, the service can prioritize requests received over SNC (which are quantitatively upper-bounded) over requests received as best-effort. Through efficient source authentication and subsequent per-source rate-limiting, the capability issuing entity for instance only has to handle at most $\alpha \times (\rho_{\text{Cap}}^{\text{Req}} + \rho_{\text{Cap}}^{\text{Resp}})$ packets per second (worst case), where α refers to the number of ASes in the Internet. In general, we rely on the services being fast enough (or to be parallelizable) to handle this request rate, which is significantly less than the terabits of attack traffic they would potentially have to handle in the worst case without DoCile deployed.

VIII. Related Work

A. DoC Protection

We provide a survey of other systems that tackle the DoC problem, and discuss their shortcomings. The survey also includes capability-based bandwidth reservation protocols (Section II) that claim to be resistant against DoC attacks.

Portcullis [25] protects capability setup using a proof-of-work scheme. This results in a *probabilistic* per-computation fair division of the bandwidth along the setup path: the guarantees are weaker than in DoCile, as packets could still be dropped. Further, Portcullis relies on a connection to a globally trusted third party, and uses CDNs or DNS to coordinate information, whose availability can also be attacked. **SIFF** [32] sends multiple capability requests, until one finally reaches the destination. As the system guarantees that the probability of a successful request is non-negligible, the combined probability of success rapidly approaches 1. This analysis, however, does not consider the presence of multiple congestion points along a path, nor DoC on the return path. These vulnerabilities might delay the establishment of capabilities indefinitely. **SIBRA** [4] evaluates that the level 1 capabilities in the protocol (ephemeral reservations) are $100\,\%$ resistant to DoC attacks. However, the system does not protect level 2 capabilities (steady reservations), opening up to level 2 DoC attacks as described in Section II-C. **TVA** [34] uses path identifiers [33] to fair-queue capability requests (*request-to-send* packets, or RTS) based on the ingress interface of each AS. Legitimate requests can nevertheless be blocked by an RTS flooding attack at the ingress border router of an intermediate AS. **Enhanced TVA (ETVA)** [16] tries to address this shortcoming by introducing a challenge-reply mechanism at the border router, and thus prevent source spoofing attacks. However, the ERTS packets (i) are an additional source of congestion in the network, and (ii) are not protected from congestion of other ERTS packets; thus, they can still be targeted by DoS both on the forward and backward path. Hence these two systems offer limited protection against DoC.

B. Other Bandwidth-Allocation Systems

For completeness, we also mention other bandwidth-allocation systems that are *not based on network capabilities*. Unfortunately, these systems do not provide any or only very limited guarantees under DDoS attacks.

Congestion control (CC) has been the primary means for resource allocation in the Internet since it was introduced in the 1980s [15]. As it cannot achieve strong communication guarantees, especially in case of adversaries, CC is insufficient for protecting the capability setup. Recently, Brown et al. [9] proposed *recursive congestion shares (RCS)*: every AS would have a contract with its neighbor specifying a guaranteed share of its egress capacity in case of congestion. Data traffic will therefore be protected recursively, i.e., by all congestion shares of its path. While this limits the impact of DDoS attacks, the probability of a packet reaching its destination still decreases exponentially in the path length. In another direction, most systems that want to achieve **QoS without relying on capabilities** do either not scale to the Internet [7], [8] because of excessive in-network state, or cannot provide any bandwidth guarantees [6]. Many widely used traffic-engineering protocols are limited to intra-AS deployment [6], [17], [28]. One of the most limiting aspects of such systems is that they have not been designed with security in mind.

C. Other Means for DDoS mitigation

Instead of providing bandwidth allocations, some systems rely on other means for DDoS attack mitigation. As such, they are unable to provide any communication guarantees.

Software-defined networking (SDN) can detect and deal with anomalous activities efficiently due to its centralized design and programmable configuration. However, SDN is typically limited to intra-domain contexts and the controller is a single point of failure. Furthermore, communication between the control- and data plane also introduces a new surface for volumetric DDoS attacks [30]. **Content delivery networks (CDNs)** can absorb massive amounts of attack traffic[13] by providing vast network capacities in conjunction with globally distributed servers. They thus protect services such as websites from DDoS, which is however a different scenario compared to undisturbed host-to-host communication that bandwidth reservation protocols try to achieve. **Out-of-band (OOB)** communication is used in practice as a backup channel that allows to reach remote nodes despite the main network connection being interrupted, and can help mitigating DDoS attacks [12]. We emphasize that DoCile does not require OOB channels, although SNC can be considered a

form of OOB communication. However, OOB communication between direct neighbors as in SNC is much easier to achieve than OOB channels between any two ASes in the Internet.

IX. Conclusion

Protecting the exchange of capabilities from network congestion and DoC attacks is the missing piece to finally achieve availability guarantees for QoS on a public Internet. In our analysis of existing capability-based reservation protocols and DoC defenses we observe that layering capabilities—i.e., using AS-to-AS (AtA) capabilities to protect Host-to-Host (HtH) capabilities—is beneficial, as it reduces the DoC attack surface. Guaranteeing the transmission of a few packets per (distinct) AS path then suffices to initiate a cascade of reservations—first AtA, then HtH—resulting in the full protection of all authorized communications.

To achieve this initial DoC-resilient setup, we design DoCile. Its two subsystems, setup-less neighbor-based communication (SNC) and telescoped reservation setup (TRS), re-purpose existing capability-based reservation protocols and peering contracts between ASes to protect AtA capability establishment and key exchange from DDoS attacks. DoCile can be bootstrapped from existing agreements between neighboring ASes and provides strong incentives for incremental deployment. It is the first system that can guarantee the establishment of keys and capabilities in spite of ongoing attacks. DoCile is therefore an important step towards an Internet free of DoC and—consequently—DDoS.

Acknowledgment

We would like to thank Juan Angel García-Pardo and Jordi Subirà Nieto for their feedback on the system design. We gratefully acknowledge support from ETH Zurich and the Zurich Information Security and Privacy Center (ZISC).

References

[1] Anderson, T., Roscoe, T., Wetherall, D.: Preventing Internet denial-of-service with capabilities. ACM SIGCOMM Computer Communication Review (CCR) (2004)

[2] Argyraki, K., Cheriton, D.: Network capabilities: The good, the bad and the ugly. In: Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets) (2005)

[3] Balaban, D.: The history and evolution of DDoS attacks. https://www.embeddedcomputing.com/technology/security/network-security/the-history-and-evolution-of-ddos-attacks (2020)

[4] Basescu, C., Reischuk, R.M., Szalachowski, P., Perrig, A., Zhang, Y., Hsiao, H.C., Kubota, A., Urakawa, J.: SIBRA: Scalable Internet bandwidth reservation architecture. In: Proceedings of the Network and Distributed System Security Symposium (NDSS) (2016)

[5] Bates, T.: CIDR report. www.cidr-report.org/as2.0/ (2021)

[6] Blake, S., Black, D., Carlson, M., Davies, E.B., Wang, Z., Weiss, W.: An architecture for differentiated services. RFC 2475 (1998)

[7] Braden, R., Clark, D., Shenker, S.: Integrated services in the Internet architecture: an overview. RFC 1633 (1994)

[8] Braden (Ed.), R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205 (1997)

[9] Brown, L., Ananthanarayanan, G., Katz-Bassett, E., Krishnamurthy, A., Ratnasamy, S., Schapira, M., Shenker, S.: On the future of congestion control for the public Internet. In: Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets) (2020)

[10] Cisco: Cisco annual Internet report (2018–2023). https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (2020)

[11] Giuliari, G., Roos, D., Wyss, M., García-Pardo, J.A., Legner, M., Perrig, A.: Colibri: A cooperative lightweight inter-domain bandwidth-reservation infrastructure. In: Conference on Emerging Networking Experiments and Technologies (CoNEXT) (2021)

[12] Gong, D., Tran, M., Shinde, S., Jin, H., Sekar, V., Saxena, P., Kang, M.S.: Practical verifiable in-network filtering for ddos defense. In: 39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, Dallas, TX, USA, July 7-10, 2019

[13] Guo, R., Li, W., Liu, B., Hao, S., Zhang, J., Duan, H., Shen, K., Chen, J., Liu, Y.: Cdn judo: Breaking the cdn dos protection with itself. In: NDSS (2020)

[14] Hawkinson, J., Bates, T.: Guidelines for creation, selection, and registration of an autonomous system (AS). RFC 1930 (1996)

[15] Jacobson, V.: Congestion avoidance and control. ACM Computer Communication Review (CCR) (1988)

[16] Jin, G., Yang, J., Wei, W., Dong, Y.: Mitigating denial of capability with a notification mechanism. In: International Conference on Networking, Architecture, and Storage (NAS) (2007)

[17] Katz, D., Kompella, K., Yeung, D.: Traffic Engineering (TE) Extensions to OSPF Version 2. RFC 3630 (2003)

[18] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P.: Internet Key Exchange Protocol Version 2 (IKEv2). RFC 5996 (2010)

[19] Keslassy, I., Chuang, S., Yu, K., Miller, D., Horowitz, M., Solgaard, O., McKeown, N.: Scaling internet routers using optics. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. SIGCOMM '03 (2003)

[20] Kwon, J., García-Pardo, J.A., Legner, M., Wirz, F., Frei, M., Hausheer, D., Perrig, A.: SCIONLab: A next-generation Internet testbed (2020)

[21] Lee, S.B., Gligor, V.D.: FLoc : Dependable link access for legitimate traffic in flooding attacks. In: IEEE International Conference on Distributed Computing Systems (2010)

[22] Liu, X., Li, A., Yang, X., Wetherall, D.: Passport: Secure and adoptable source authentication (2008)

[23] Maigron, P.: Autonomous system number statistics (2021), https://www-public.imtbs-tsp.eu/~maigron/RIR_Stats/RIR_Delegations/World/ASN-ByNb.html

[24] Medhi, D., Ramasamy, K.: Network Routing: Algorithms, Protocols, and Architectures. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2007)

[25] Parno, B., Wendlandt, D., Shi, E., Perrig, A., Maggs, B., Hu, Y.C.: Portcullis: Protecting connection setup from denial-of-capability attacks. In: Proceedings of the ACM SIGCOMM Conference (2007)

[26] Perrig, A., Szalachowski, P., Reischuk, R.M., Chuat, L.: SCION: A Secure Internet Architecture. Springer (2017)

[27] RIPE NCC: Autonomous System Numbers. https://www.ripe.net/manage-ips-and-asns/as-numbers/request-an-as-number

[28] Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. RFC 3031 (2001)

[29] Rothenberger, B., Roos, D., Legner, M., Perrig, A.: PISKES: Pragmatic Internet-scale key-establishment system. In: Proceedings of the ACM Asia Conference on Computer and Communications Security (ASI-ACCS) (2020)

[30] Swami, R., Dave, M., Ranga, V.: Software-defined networking-based ddos defense mechanisms. ACM Comput. Surv. (Apr 2019)

[31] Wyss, M., Giuliari, G., Legner, M., Perrig, A.: Secure and scalable QoS for critical applications. In: Proceedings of the IEEE/ACM International Symposium on Quality of Service (IWQoS) (2021)

[32] Yaar, A., Perrig, A., Song, D.: SIFF: a stateless Internet flow filter to mitigate DDoS flooding attacks. In: Proceedings of the IEEE Symposium on Security and Privacy (2004)

[33] Yaar, A., Perrig, A., Song, D.: Pi: A path identification mechanism to defend against DDoS attacks. In: Proceedings of the IEEE Symposium on Security and Privacy (2003)

[34] Yang, X., Wetherall, D., Anderson, T.: A DoS-limiting network architecture. ACM SIGCOMM Computer Communication Review (2005)

[35] Yang, X., Wetherall, D., Anderson, T.: Tva: A dos-limiting network architecture. IEEE/ACM Transactions on Networking (2008)

[36] Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Communications Surveys & Tutorials (2013)