2015 44th International Conference on Parallel Processing

DISCS: a DIStributed Collaboration System for Inter-AS Spoofing Defense

Bingyang Liu and Jun Bi
Tsinghua University,
liubingyang, junbi@tsinghua.edu.cn

***Abstract*—IP spoofing is prevalently used in DDoS attacks for anonymity and amplification, making them harder to prevent. Combating spoofing attacks requires the collaboration of different autonomous systems (ASes). Existing methods either lack flexibility in collaboration or require centralized control in the inter-AS environment. In this paper, we propose a DIStributed Collaboration System (`DISCS`) for inter-AS spoofing defense, which allows ASes to flexibly collaborate in spoofing defense in a distributed manner. Each `DISCS`-enabled AS implements four defense functions. When a victim AS is under a spoofing attack, it can request other ASes to execute the most appropriate defense functions. We present the distributed and flexible control plane design and the backward compatible and incrementally deployable data plane design for both IPv4 and IPv6. We evaluate `DISCS` with theoretical proof and simulations using real Internet data. The results show that `DISCS` has strong deployment incentives, high effectiveness, minimal false positives, modest resource consumption and strong security.**

***Index Terms*—IP spoofing, spoofing defense, DDoS defense, inter-AS collaboration**

I. Introduction

Distributed denial of service (DDoS) attacks have been the top operational threat of Internet service providers (ISPs) for many years [1] [2] [3]. In brute-force bandwidth DDoS attacks, the agents (also known as botnets) controlled by attackers send high-volume and seemingly legitimate traffic to overwhelm the uplink of victim networks [4]. Since the victim cannot handle the high volume on its own, upstream providers must help it block the traffic, typically by blacklisting the agents' source addresses [4].

With IP spoofing, the attack traffic is even harder to filter. Since the packets' source addresses are forged, blacklisting them may not only fail to block subsequent packets, but also incorrectly block legitimate users. There are two types of IP spoofing based DDoS attacks, *d-DDoS* and *s-DDoS*[1]. In d-DDoS, the agents directly send packets to the victim with arbitrary source addresses for anonymity. In s-DDoS, IP spoofing is used for reflection, i.e., agents use the victim's source addresses to send requests to innocent destination hosts, whose replies then flood the victim.

Conventional wisdom may suggest that IP spoofing is no longer widely used in DDoS attacks due to the large farms of botnets. However, it is not true for two reasons. First, in d-DDoS, IP spoofing helps conceal the agents' locations, making the botnets less likely to be cracked [5]. Second, in s-DDoS, IP spoofing can be used to amplify the attack volume. For example, in a DNS amplification attack, a 60-byte DNS request can trigger a 4000-byte response, yielding an amplification factor of 73 [6]. Recent measurement shows that spoofing based DDoS attacks are truly prevalent [7] [8], and the highest attack rate reached 400Gbps, which is the most intensive DDoS attack ever observed [9].

Combating these spoofing attacks requires the collaboration between different ASes. Inter-AS collaboration has three advantages in spoofing defense. First, it enables spoofing traffic to be filtered far from the victim AS, which alleviates the victim AS's bandwidth pressure and saves intermediate network bandwidth. Second, by exchanging spoofing identification information, ASes' ability for identifying spoofing traffic becomes higher [10] [11]. Third, by providing better filtering effect for collaborators, ASes have higher deployment incentives [12] [13].

However, most of the existing methods in collaborative spoofing defense have a common drawback, i.e., they require the spoofing filtering functions to be executed on all the traffic all the times. This does not only increase the operational costs and routers' work loads, but also causes operational risks since the filtering functions may drop legitimate packets. To deal with the problem, a recent research proposes mutual egress filtering (MEF), a collaboration system that allows on-demand invocation of the filtering functions, which greatly reduces cost and risks and further increases deployment incentives [13]. However, MEF still has flaws. First, it requires a central registration server on the Internet to maintain all the ASes' information, which does not conform with the distributed and autonomous gene of Internet inter-AS environment. Second, although MEF proposes on-demand filtering mechanisms, its filtering functions have intrinsic limitations, i.e., the victim AS cannot determine whether an inbound packet is spoofed or not no matter what source address it carries, so it cannot enforce prioritized queues in case the bandwidth is overwhelmed.

In this paper, we propose a DIStributed Collaboration System (`DISCS`) for inter-AS spoofing defense, which has three main contributions.

First, we present in detail a flexible and incrementally deployable design for `DISCS`. On the inter-AS aspect, ASes use BGP for peer discovery, which allows incremental

This work was supported by the National High-tech R&D Program ("863" Program) of China (No.2013AA013505) and the National Science Foundation of China (No.61472213).

Jun Bi is the corresponding author.

[1] "d-" indicates that the victim is the "d"estination addresses of the spoofing packets; "s-" indicates that the victim is the "s"ource addresses.

0190-3918/15 $31.00 © 2015 IEEE
DOI 10.1109/ICPP.2015.25

deployment, and a customized peer-to-peer protocol for on-demand collaboration control, which achieves confidentiality and scalability. Each AS implements four defense functions, which do not only allow ASes to filter spoofing traffic, but also allow the victim AS to determine whether an inbound packet comes from a collaborator ASes. This advanced functionality enables filtering and prioritizing policies.

Second, we propose detailed designs for the four spoofing defense functions, on-demand invocation mechanism, distributed control plane, efficient data plane design, and backward compatible packet format design for IPv4 and IPv6. These designs all together make DISCS easy to be implemented on routers and compatible with legacy infrastructures to support incremental deployment.

Third, we evaluate DISCS against the design requirements (see Section III-A). We theoretically prove that DISCS provides monotonically increasing deployment incentives. By extensive simulations with real Internet data, we show that if 50 largest ASes deploy DISCS, the incentive is 68%, and the effectiveness is 41%. The computation, storage and network cost can scale to the Internet scope and fit in commodity servers and routers with the help of hardware cryptographic modules. We analyze possible attacks against DISCS and show that it provides practical security. Besides being IFP-free, OFP can be avoided by minimizing manual configuration and applying alarm mode.

This paper is organized as follows. We summarize related work in Section II. Then we introduce the design requirements and system overview in III. We present the system design and control plane design in Section IV, and the data plane design in Section V. Section VI evaluates DISCS against the design requirements. Section VII concludes the paper.

II. Related Work

According to the information and principles used for spoofing identification, inter-AS spoofing defense methods can be classified into 3 categories, i.e., end based, end-to-end (e2e) based and path based ones, where an end is an AS and a path is an AS path.

An end based method only requires the information of the local AS to identify spoofing packets. Ingress filtering (IF) [14] is an end based method. IF works at AS borders. It checks the source addresses of the packets leaving an AS against an access list, and drops any packet that does not match the list. In MEF, ASes collaboratively defend against spoofing attacks with an end based method (egress filtering) [13].

In e2e based methods, each ordered pair of ASes (s, d) shares a secret key. For each packet from s to d, the border router of s generates a mark using the key and stamps the mark into the packet. The mark is then verified by the border router of d. If the mark is invalid, the packet is identified as spoofed. Spoofing prevention method (SPM) directly uses the keys as marks [10], while Passport [11] uses keyed message authentication codes (MACs) as marks. Another difference is that Passport also generates marks for intermediate ASes on the forwarding paths so that the intermediate ASes can also identify spoofing packets and demote their forwarding priority. E2e based methods typically have built-in IF to enhance source address validity.

Path based methods associate a source with some information of the valid forwarding paths, where the information can be 1) previous hops on the paths, 2) the length of the paths, and 3) per path marks. Distributed packet filter (DPF) [15] shows that previous hop information can be very effective in filtering spoofing traffic. Unicast reverse packet forwarding (uRPF) [16], inter-AS packet filter (IDPF) [17] and source address validation enforcement (SAVE) [18] specify how to infer valid previous hops in practice. uRPF uses the next hops toward the source address in the forwarding table as valid previous hops. In IDPF, the ASes from whom the BGP updates regarding the source address are received are considered feasible previous hops. In SAVE, a source router probes destination address space so that the routers en route can learn the valid previous hops for the source prefixes associated with the source router upon receipt of the probes. Hop-count filtering (HCF) [19] infers the length of valid paths and the length of the path that the packet traverses from the packet's TTL, and the packet is identified as spoofed if the two lengths do not match. Path identification (PiIP) [20] routers push self-generated bits into the packet's mark stack, and the integral mark stack learned during peace time is considered valid by the destination.

Among these methods, IF and uRPF are most widely deployed in practice [2] due to their low cost. However, according to a long-term measurement, the global deployment of IF and uRPF was not been improved in four years [21], and about 41.6% ASes on the Internet are still spoofable [22]. This is because IF lacks deployment incentives since an AS can gain little additional self-protection by deploying it. uRPF causes false positives when paths are asymmetric. Hence the prevalent route asymmetry on the Internet impedes its universal deployment. Actually, all path based methods have IFP, especially under partial deployment and route change, let alone some methods have poor incentives and effectiveness. E2e based methods are generally IFP-free. However, SPM and Passport have weak incentives against s-DDoS. SPM has much lower cost than Passport by using deterministic e2e marks, but it loses security. In summary, balancing the design requirements is still an open problem.

III. Design Requirements and System Overview

A. Design Requirements

A recent comparative evaluation study summarizes main performance measures of spoofing defense methods, i.e., effectiveness, cost, security and false positives as the [23] [2]. While the first three are generic measures for network techniques, the fourth one (false positives) is specific to detecting and filtering techniques. Besides, as we focus on the inter-AS scenario, where ASes autonomously make deployment decisions according to their individual interests, a method should provide

[2] We omit the fifth measure (false negatives) since it is not an independent measure, i.e., it can be deduced from effectiveness, as shown in [23].

ASes with strong deployment incentives [21] [12]. Detailed explanations are as follows.

Strong deployment incentives: By deploying a method, an AS should gain significant additional protection, i.e., additional reduction of the spoofing traffic whose destination and source addresses lie in the AS in d-DDoS and s-DDoS, respectively [11].

High effectiveness: The deployment of a method should efficiently reduce the global spoofing traffic on the Internet [15].

Low cost: A method should consume as few resources as possible, including computation and storage cost on routers and extra cost in network bandwidth.

Low false positives: False positives (FP) represent the possibility of a method to drop genuine packets. There are two sources of FP. One is operational FP (OFP), which is caused by operational faults. The other one is inherent FP (IFP), meaning a method has inherent flaws to drop genuine packets no matter how well it is operated. While OFP can be minimized by careful implementation and operation, IFP can only be minimized with proper design.

High security: If a method has security vulnerabilities, a sophisticated attacker can bypass the method to launch spoofing attacks. Thus, a method should avoid possible vulnerabilities.

B. *System Overview*

Each DISCS-enabled AS (*DAS*) can establish mutual collaboration relationship with other DASes, so called DISCS peers. When a DAS is under spoofing attacks, it can request its peers to collaboratively execute spoofing defense functions to protect it.

DISCS employs four end based and e2e based functions to **avoid false positives**. *Destination protection (DP)* and *cryptographic destination protection (CDP)* protect the victim DAS from d-DDoS. *Source protection (SP)* and *cryptographic source protection (CSP)* protect the victim DAS from s-DDoS. DP and SP are end based, and CDP and CSP are e2e based. The e2e marks are cryptographic, which are **securer** than the deterministic marks used by SPM. For an outbound packet, the source DAS only generates one mark for the destination DAS, instead of multiple marks for all the intermediate DASes en-route. Thus, DISCS has much **lower cost** than Passport.

We introduce novel on-demand function invocation mechanism in the design of DISCS: 1) functions are invoked by the victim only "when" an attack is detected, 2) the victim can choose "which" function(s) to invoke according to the type of the attack (DP and CDP for d-DDoS, and SP and CSP for s-DDoS), and 3) functions are executed only for the subnetworks "who" are under attack instead of the whole AS. On-demand invocation avoids all functions from executing all the time for all traffic so that **resource consumption is minimized**. The design of on-demand invocation is lightweight and Internet-wide scalable.

Another key difference from SPM and Passport is that DISCS enforces collaborative defense against both d-DDoS and s-DDoS. DASes can gain equivalent protection against the two types of attacks. However, in SPM and Passport, the defense is mainly designed for d-DDoS, i.e., an outbound packet is filtered only when its destination belongs to a peer AS. Inter-AS collaboration and on-demand invocation only protect DASes, while a legacy AS (*LAS*) cannot get any protection from DISCS since it does not have peers. This provides ASes with **strong deployment incentives**. We will also show that under optimal deployment strategy, DISCS can be **effective** in reducing global spoofing traffic with a small set of early deployers.

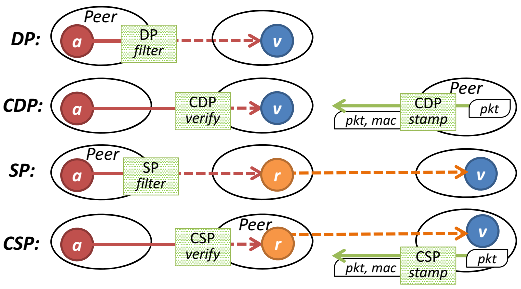


Fig. 1: DISCS spoofing defense functions.

IV. System Design

In this section, we first describe the detailed design of the defense functions. Then we present the control plane design by following the whole process from initial deployment to collaboration in spoofing defense. The process includes 4 steps.

[Step 1] **DAS Discovery:** A DAS must discover other DASes before collaborating with them.

[Step 2] **Peer Relationship Setup:** Two DASes become DISCS peers only if both agree to peer with the other.

[Step 3] **Key Negotiation:** A pair of peers must negotiate the keys for the cryptographic functions (CDP and CSP).

[Step 4] **Function Invocation:** The DAS under attack can request its peers to execute defense functions for it.

A. *Spoofing Defense Functions*

The functions are implemented on AS border routers. DP and SP are end based, and CDP and CSP are e2e based. An e2e based function has two primary operations, stamping and verification. *Stamping:* when forwarding an outbound packet, the border router of the source DAS writes a mark in the packet. *Verification:* upon receipt of an inbound packet, the border router of the destination DAS validates the mark – if the mark is valid, it is erased and the packet is forwarded; otherwise the packet is identified as spoofed and dropped. The marks are per packet cryptographic keyed MACs generated with symmetric key algorithms, which consume far less computation resource than the asymmetric algorithms with equivalent security [11].

Figure 1 roughly illustrates how the functions work. When a subnetwork v in a DAS is under attack, the DAS asks its peers to execute corresponding defense function(s). DP and CDP protect v from being the destination of d-DDoS attacks. SP and CSP protect v from being the source of s-DDoS attacks.

TABLE I: The Anatomy of DISCS Functions

FUNCTION	DIR.	COND.	ACTION
DP-filter	out	$dst \in v$	if $src \notin local$, drop
CDP-stamp	out	$dst \in v$	stamp
CDP-verify	in	$dst \in v$	if $src \in peer$, verify
SP-filter	out	$src \in v$	drop
CSP-stamp	out	$src \in v$	if $dst \in peer$, stamp
CSP-verify	in	$src \in v$	verify

The functions filter different sets of spoofing packets, as shown below.

Destination Protection (DP): A peer DAS filters outbound packets targeting v. If the source address of a packet does not belong to the local AS, the packet is identified as spoofed and dropped.

Cryptographic Destination Protection (CDP): A peer DAS stamps each outbound packet targeting v. Upon receipt of an inbound packet targeting v, the victim DAS verifies the packet if the source address belongs to a peer,

Source Protection (SP): A peer DAS drops the outbound packets whose source addresses belong to v.

Cryptographic Source Protection (CSP): The victim DAS stamps each outbound packet whose source address belongs to v and destination address belongs to peer DASes. Upon receipt of an inbound packet whose source address belongs to v, peer DASes verify the packet.

Table I anatomizes the functions. The functions in bold are executed by peer DASes, and the others are executed by the victim DAS. We leave the MAC generation algorithms, packet formats and other data plane details in Section V.

B. DAS Discovery

Figure 2 shows the inter-AS structure of the DISCS system. Each DAS has a controller, which connects with local border routers via iBGP like a route reflector [24]. To be discovered by other DASes on the Internet, once an AS deploys DISCS, its controller initializes an advertisement (namely DISCS-Ad). The DISCS-Ad is carried in a BGP update, spread from the controller to local border routers via iBGP, then to the border routers of other DASes via eBGP, and finally to the other controllers via iBGP. A DISCS-Ad is formatted as an optional transitive BGP path attribute so that LASes can retain it without knowing what it means [11]. To be spread by the border routers, the update must make modifications to their Loc-RIBs [25]. We suggest prepending or de-prepending the origin AS to the AS path since these updates can modify Loc-RIBs without affecting the reachability of prefixes.

The DISCS-Ad contains the AS number of the origin DAS i and the domain name (or IP address) of its controller. Upon receipt of the DISCS-Ad, the controller of DAS j can learn the existence of DAS i and communicate with i's controller by its domain name. The communication channel between the controllers (*con-con channel*) is secured with SSL. The SSL overhead will be evaluated in Section VI-C.

Note that the security of DAS discovery process relies on the security of BGP. If BGP is insecure, man-in-the-middle attacks can be easily launched. For example, an attacker in the middle can modify the domain name of the controller and hijack the con-con channel. Fortunately, with the standardization and steady deployment of RPKI [26] [27], accompanied with the secure inter-AS routing protocols such as secure BGP [28] and secure origin BGP [29], we assume that BGP will become secure in the future.

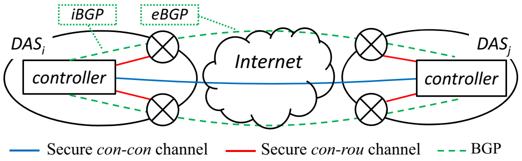


Fig. 2: The inter-AS structure of DISCS.

C. Peer Relationship Setup

A DAS can selectively peer with other DASes based on its DISCS peering policy, e.g., it can make a local blacklist containing the DASes who have a conflict of interest. Upon receipt of the DISCS-Ad of DAS i, the controller of DAS j checks i against the blacklist. If i is not blacklisted, the controller of j sends a peering request to the controller of i with a delay of random duration, which prevents all DASes from sending requests simultaneously. Upon receipt of the peering request, i's controller decides whether to accept the request by checking j against its local blacklist. The peer relationship is set up once i accepts the peering request.

D. Key Negotiation

Each ordered pair of DASes (i, j) shares a symmetric key $key_{i,j}$, which is used by i and j to stamp and verify the MACs, respectively. $key_{i,j}$ is locally generated by the controller of i, and sent to the controller of j via the secure con-con channel. The two controllers then deploy the key to local border routers. We assume that the communication between the controller and the routers is secure since today's remote access to the routers has to be secured in any case.

For better security, a DAS may want to re-key periodically. To avoid the verification failure caused by temporary asynchronization, the re-keying party first sends the new key to its peer, but does not use the new key for stamping until its peer finishes deploying the new key. For verification, a MAC is considered valid if it conforms with either the old key or the new key.

E. Function Invocation

DISCS functions are invoked on demand. The victim DASes should decide "when" to invoke, "which" functions to invoke, and "who" to protect. Since most networks are not under attack for most of the time, on-demand invocation avoids the functions from consuming resources in vain.

1) "When" to invoke: At the upstream of the DDoS defense tool chain are the attack detection modules [4, AL-2:ADS], which detect attacks in real time and invoke the DISCS functions automatically. If the detection module is not deployed, DISCS itself can be used for attack detection, which we will present in Section IV-F. In the worst case, network operators can invoke the functions manually.

Every invocation is associated with duration. When the duration expires, the victim DAS and its peers stop executing the functions. The duration can be a predefined value, e.g., 24 hours[3]. If attacks are still in progress when the duration ends, the victim DAS can re-invoke the functions with a longer duration.

If CDP or CSP is invoked, the asynchronization between the DASes may cause temporary verification failure since verification may start earlier or stop later than stamping. Two short intervals at the beginning and the end of the duration are used to tolerate the asynchronization. Within the intervals, the verification end only erases MACs without verifying the MACs or dropping invalid packets.

2) "Which" function(s) to invoke: The victim DAS chooses which functions to invoke according to the type of the spoofing attack. Generally speaking, DP and CDP are used to combat d-DDoS while SP and CSP are used to combat s-DDoS. Since e2e cryptographic functions consume much more resource than end based functions, we suggest that DP/SP is invoked whenever CDP/CSP is invoked so that a portion of spoofing traffic is dropped by DP/SP before being processed by CDP/CSP. If the attack type is unknown or the attack is highly destructive, the victim can invoke all the four functions.

3) "Who" to protect: The victim DAS can specify the IP prefixes of the victim subnetwork in the invocation request. The peer DASes check the ownership of the prefixes, and accept the request only if they belong to the victim DAS.

Now we have described all the factors related with on-demand function invocation. The complete formation of an invocation is a triple $(v, f, duration)$, where v is the prefixes to be protected, f is the function to be executed on v, and $duration$ is the duration associated with f. The victim DAS can specify multiple triples to invoke multiple functions and for multiple victims.

F. Alarm Mode

DISCS can be applied in alarm mode, where identified spoofing packets are not dropped immediately, but sampled and sent to the controller using NetFlow [31] or sFlow [32] for further analysis. Once the controller detects an attack, it requests the peers to quit the alarm mode, i.e. to drop the identified spoofing packets. Alarm mode is especially useful for the DASes lacking attack detection modules.

V. Data Plane Design

In this section, we present the detailed data plane design of DISCS, including the tables maintained in routers, how to generate a *tuple* for an incoming packet by looking up the tables, and how to utilize the tuple to perform DISCS operations on the packet. Then we present the MAC generation algorithm and the packet formats for IPv4 and IPv6.

A. Tables in Data Plane

A DISCS border router maintains some extra tables in the data plane. All the tables are firstly constructed by the controller, and then installed on routers.

Prefix-to-AS Mapping Table: The Pfx2AS table maps (longest prefix match) each IP prefix p to the AS i it belongs to, i.e., i = Pfx2AS(p). A controller can obtain accurate mapping by retrieving RPKI certificates.

Key Tables: DAS i maintains two key tables, which map a peer to the keys associated with it, one for stamping (Key-S) and the other for verification (Key-V). For a peer j, $key_{i,j}$ = Key-S(j), and $key_{j,i}$ = Key-V(j).

Function Tables: Function tables map an IP prefix to the functions to be executed on it. There are four function tables, namely In-Src, In-Dst, Out-Src and Out-Dst, which match (longest prefix match) the source and destination addresses of inbound and outbound packets, respectively. From Table I, we can see that the sets of possible functions for In-Src, In-Dst, Out-Src and Out-Dst are $\{$CSP-verify$\}$, $\{$CDP-verify$\}$, $\{$SP, CSP-stamp$\}$ and $\{$DP, CDP-stamp$\}$, respectively. Denote by Func(s, d) the set of functions to be executed on packet (s, d). Func(s, d) = In-Src(s) $\cup$ In-Dst(d) if it is an inbound packet; Func(s, d) = Out-Src(s) $\cup$ Out-Dst(d) if it is an outbound packet.

B. Tuple Generation

A *tuple* is a data structure generated for an incoming packet. The tuple records the DISCS operations to be performed on the packet and the information required by the operations. There are two types of tuples, i.e., in-tuples and out-tuples, which are associated with inbound and outbound packets, respectively. This subsection describes how to generate the tuples by looking up the DISCS tables.

In-tuple: Since the set of functions that can possibly be executed on an inbound packet is $\{$CSP-verify, CDP-verify$\}$, an in-tuple has a format of $(verify?, key_v)$, where $verify?$ is a boolean value indicating whether verification should be performed, and key_v is the key for verification. $verify?$ is set if and only if Func(s, d) $\neq \phi$, which means that if either CSP-verify $\in$ InSrc(s) or CDP-verify $\in$ InDst(d), the packet should be verified. key_v = Key-V(Pfx2AS(s)), where the key is associated with the source AS, which is mapped from s.

Out-tuple: Since the set of functions that can possibly be executed on an outbound packet is $\{$SP, DP, CSP-stamp, CDP-stamp$\}$, the out-tuple has a format of $(drop?, stamp?, key_s)$, where $drop?$ is a boolean value indicating whether the packet should be dropped, $stamp?$ is a boolean value indicating whether the packet should be stamped, and key_s is the key for stamping. $drop?$ is set if and only if Pfx2AS(s) $\neq$ LocalAS and (SP $\in$ Out-Src(s) or DP $\in$ Out-Dst(d)). $stamp?$ is set if and only if (CSP $\in$ Out-Src(s) and Key-S(Pfx2AS(d)) $\neq Null$) or CDP $\in$ Out-Dst(d). key_s = Key-S(Pfx2AS(d)).

C. Processing Flow

With the help of tuples, the DISCS processing flow is quite simple as shown in Figure 3. For an inbound packet,

[3] [30] shows more than 93% of DDoS attacks last less than 24 hours.

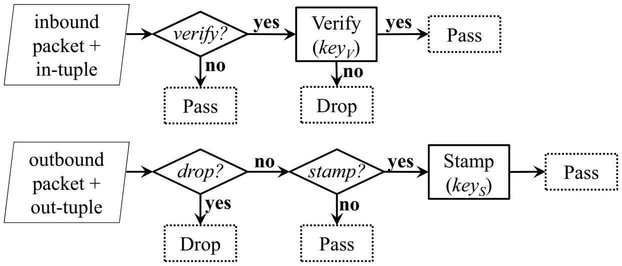


Fig. 3: DISCS processing flow.

if $verify?$ is not set, the packet is directly passed to the forwarding engine of the router. Otherwise, the packet is verified using key_v. Valid packets are passed to the forwarding engine while invalid ones are dropped. For an outbound packet, it is dropped if $drop?$ is set. An un-dropped packet is stamped with if $stamp?$ is set; otherwise it is directly passed to the forwarding engine.

D. MAC Generation Algorithm

We use AES-CMAC [33] as the MAC generation algorithm for its practical security and extensive high-speed hardware implementations. AES-CMAC takes a 128-bit key and a msg as input, where msg is some fixed fields of a packet, and outputs $cmac$ = AES-CMAC(key, msg). $cmac$ is then truncated to fit in the available space in the packet. See Section V-E and V-F for more details about the generation of msg and the truncation of $cmac$ for IPv4 and IPv6, respectively.

E. IPv4 Packets

For IPv4, msg contains 21 bytes. The first 13 bytes include Version, IHL, Total Length, Flags (padding with 5 '0' bits), Protocol, Source address and Destination address fields in the IP header. The last 8 bytes are the first 8 bytes of the payload (padding '0' bits if less than 8 bytes). The reasons that we choose these fields are twofold. First, they are immutable, i.e., they typically do not change en route. Hence, the destination DAS can produce the same msg as the source DAS. Second, these fields are sufficient to differentiate almost all non-identical packets [34] [4]. This makes DISCS immune to replay attacks (see Section VI-E2).

$cmac$ is truncated to 29 bits to be placed in the 16-bit IPID field and the 13-bit Fragment Offset field, as shown in Figure 4. For stamping, the two fields are replaced with a MAC. After verification, if the MAC is valid, these two fields are replaced with random bits. Of course, the header checksum should be updated accordingly after stamping and verification.

Rewriting IPID and Fragment Offset may cause destination hosts fail to assemble IP fragments. However, since only about 0.06% of IPv4 packets on the Internet are fragmented [35], we think this collateral damage is worth taking for the victim subnetwork who is suffering from an attack. Other networks will not be affected since DISCS functions are only executed for the victim prefixes.

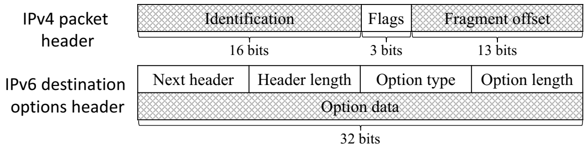


Fig. 4: Packet formats: MAC is embedded in the masked fields.

F. IPv6 Packets

The msg of an IPv6 packet includes source address, destination address and the first 8 bytes of the payload (padding '0' bits if less than 8 bytes). The reason why we do not include Payload Length and Next Header is that we are going to modify these two fields.

An IPv6 packet does not have much infrequently used space in the header [5]. Hence, we place the MAC in the destination options header lying before the routing header as an option, namely DISCS option. The first three bits of the option type are "001", which tell legacy routers to forward the packet even if they cannot correctly process the option. The other five bits of the type need to be allocated by Internet assigned numbers authority (IANA). The option data length is 4, and the option data is the 4-byte MAC. For stamping, if destination options header exists, only the option should be inserted; otherwise the entire header should be inserted. For verification, the option is removed if the MAC is valid, and if no other option exists, the entire options header should be removed. After stamping or verification, the Payload Length and Next Header fields in IPv6 header should be updated accordingly.

Stamping a MAC enlarges an IPv6 packet by at most 8 bytes even when an entire destination options header is inserted. The resulted packet size may exceed the maximum transmission unit (MTU) of the external link of the source DAS. In this case, the border router sends a "packet too big" ICMPv6 message back to the source host, announcing a new MTU value that is 8 bytes smaller than the MTU of the external link.

The MAC is not put in hop-by-hop options header since this header is processed by every router on the path. Legacy routers may process it using general CPU, resulting in low throughput. Instead, destination options header is only processed by the DISCS routers and the routers listed in routing header, which is seldom used. We do not define a new option header since legacy routers cannot recognize it and may even drop the packet due to implementation simplification.

VI. Evaluation

This section evaluates DISCS against the design requirements, including deployment incentives, effectiveness, cost, FP and security.

[4] DISCS slightly differs from [34] in that IPID and Fragment Offset are excluded from msg since DISCS rewrites these fields with a MAC.

[5] People may suggest using the Flow Label field since its usage is not very clear. It was true; but the latest specification defines its usage and forbids routers from modifying this field [36].

A. *Deployment Incentives*

Deployment incentive describes how much a technique can motivate an LAS to deploy it. Denoting by D the set of DASes, $v \notin D$ is an LAS. For a spoofing defense method, deployment incentive is defined as the expected ratio of the reduced spoofing traffic attacking v when v becomes a DAS to the total spoofing traffic attacking v [11].

Denote by (a, i, v) a spoofing flow sent by agent AS a, attacking victim AS v, and taking AS i as an innocent. In a d-DDoS, v is the destination address of the flow and i is the spoofed source address. In an s-DDoS, v is the intended source address and i is the innocent destination (reflector).

The integral filter $F(D, (a, i, v))$ describes whether the DASes in D can filter out the spoofing flow (a, i, v):

$$F(D,(a,i,v)) = \begin{cases} 1, & \text{if } D \text{ can filter out } (a,i,v) \\ 0, & \text{otherwise} \end{cases}$$

When v becomes a DAS, the integral filter becomes $F(D \cup \{v\}, (a, i, v))$. Define $\Delta(D, (a, i, v))$ as the "delta" between the two filters regarding the spoofing flow (a, i, v):

$$\Delta(D,(a,i,v)) = F(D \cup \{v\},(a,i,v)) - F(D,(a,i,v))$$

Let $\delta = \Delta(D, (a, i, v))$. $\delta = 1$ indicates that v improves its protection by becoming a DAS since the flow that cannot be filtered by D can now be filtered by $D \cup \{v\}$. $\delta = -1$ indicates that becoming a DAS declines v's protection. $\delta = 0$ indicates that the two filters are equivalent, i.e. both can filter the flow or neither can filter it.

Denote by p_j^A, p_j^I, and p_j^V the independent probabilities of AS j being the agent, innocent, and victim of a spoofing flow, respectively, where $\forall j, 0 \leq p_j^A, p_j^I, p_j^V \leq 1$, and $\sum_j p_j^A = \sum_j p_j^I = \sum_j p_j^V = 1$. v's deployment incentive is the accumulated weighted deltas:

$$inc(D,v) = \sum_{a,i} p_a^A p_i^I \Delta(D,(a,i,v))$$

1) Deployment Incentives in Theory: Next, we formalize the deployment incentives of DISCS functions. We assume that all DASes peer with each other. Since all the functions are invoked on demand, an LAS cannot get any protection from DISCS, i.e., $F(D, (a, i, v)) = 0$. Thus, $\Delta(D, (a, i, v)) = F(D \cup \{v\}, (a, i, v))$.

Deployment incentive of DP against d-DDoS: When v becomes a DAS, every j in D filters outbound traffic targeting v, and the spoofing packets whose source addresses do not belong to j are dropped.

$$\Delta(D,(a,i,v)) = \begin{cases} 1, & \text{if } a \in D \text{ and } i \neq a \\ 0, & \text{otherwise} \end{cases}$$

Thus, the deployment incentive is

$$inc(D,v) = \sum_{a \in D, i \neq a} p_a^A p_i^I = \sum_{a \in D} p_a^A (1 - p_a^I)$$

Deployment incentive of CDP against d-DDoS: When v becomes a DAS, each j in D stamps outbound traffic targeting v, and v verifies the inbound traffic whose source address belongs to j. If a packet is not originated from j, v can identify and drop it by verifying its MAC.

$$\Delta(D,(a,i,v)) = \begin{cases} 1, & \text{if } i \in D,\, a \neq v \text{ and } a \neq i \\ 0, & \text{otherwise} \end{cases}$$

Thus, the deployment incentive is

$$inc(D,v) = \sum_{i \in D, a \neq v, a \neq i} p_a^A p_i^I = \sum_{i \in D} p_i^I (1 - p_v^A - p_i^A)$$

Deployment incentive of DP+CDP against d-DDoS: The incentive of DP+CDP is not equal to the sum of the incentives of DP and CDP since the sets of spoofing traffic filtered by the two functions have overlap when $i \in D$, $a \in D$ and $i \neq a$. Filtering the flows in this overlap has an incentive of $\sum_{i \in D, a \in D, a \neq i} p_i^I p_a^A = \sum_{i \in D} p_i^I (p_D^A - p_i^A)$, where $p_D^A = \sum_{a \in D} p_a^A$. Hence, the deployment incentive of DP+CDP is $\sum_{a \in D} p_a^A (1 - p_a^I) + \sum_{i \in D} p_i^I (1 - p_v^A - p_i^A) - \sum_{i \in D} p_i^I (p_D^A - p_i^A)$, i.e.

$$inc(D,v) = \sum_{a \in D} p_a^A (1 - p_a^I) + \sum_{i \in D} p_i^I (1 - p_v^A - p_D^A)$$

The theoretical deployment incentives of SP, CSP and SP+CSP have exactly the same forms as DP, CDP and DP+CDP, respectively. Since the proof is also similar, we provide it in the supplementary material to save space here.

Monotonically Increasing Incentives: The deployment incentives of DISCS functions are monotonically increasing, i.e., $\forall D \subseteq D', \forall v \notin D'$, $inc(D, v) \leq inc(D', v)$. For lack of space, we move the detailed proof to the supplemental material. Intuitively, the more DASes there are, the more help v can get when it is under attack if it becomes a DAS. This is practical since all the functions are incrementally deployable. First, since the functions are end based or e2e based, they work correctly when partially deployed. Second, the structure of the system changes incrementally with the deployment, i.e., peering relationship with a new DAS can be established without changing old peering relationship.

2) Deployment Incentives in Simulation: We simulate the deployment process of DISCS, and illustrate how the deployment incentives of DISCS functions grow with the deployment ratio. Initially, we let $D = \phi$. Each time we randomly select an LAS and put it into D, until all the ASes are in D. For each D, we calculate the average deployment incentive, which is defined as the weighted average deployment incentives of the LASes, i.e., $inc(D) = \sum_{v \notin D} p_v^V inc(D, v) / \sum_{v \notin D} p_v^V$. We run the simulation for 50 times and plot the mean value in Figure 5.

To calculate incentives in the simulation, we use r_j to approximate p_j^A, p_j^I and p_j^V, where r_j is the ratio of the routable address space size of AS j to the global routable address space size. That is to say, we simply assume that every routable address has the same probability to be the agent, innocent and victim of a spoofing flow. This assumption is widely applied in this research area [23] [10] [12].

We obtained the real Internet prefix-to-AS mapping data from Caida [37] on Oct 11, 2012, and calculated the address space size of each AS by (longest prefix) matching the prefixes into the AS numbers. If a prefix is mapped to multiple ASes due to multiple origin ASes or AS sets, the space size of the prefix is evenly distributed to theses ASes. If the address space size of an AS is 0, it is manipulated to 1 to avoid division by zero when calculating the average incentives. We calculate the global routable address space size by summing up the sizes of all the 44036 ASes, and then calculate r_j for each AS j.

From Figure 5, we can see that the deployment incentives of the DISCS functions are monotonically increasing. The curve of DP/SP almost coincides with CDP/CSP, which indicates that DP/SP is more cost-effective than CDP/CSP since the later requires heavy cryptographic computation. The curve of DP+CDP/SP+CSP is higher. This suggests that the cost-effective invocation strategy is DP/SP or DP+CDP/SP+CSP, and CDP/CSP should not be invoked alone. Unless otherwise specified, we indicate the deployment incentive of DP+CDP or SP+CSP when we say the deployment incentive of DISCS. When 10% of ASes deploy DISCS, the incentive for LASes is 16.88%, i.e., an LAS can reduce 16.88% of the attacking traffic on average if it deploys DISCS. When the deployment ratio is 50%, the incentive becomes 68.65%.

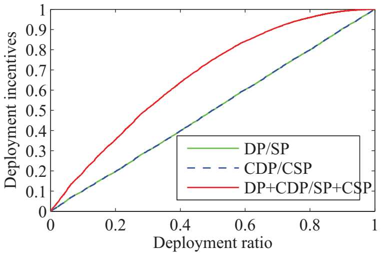
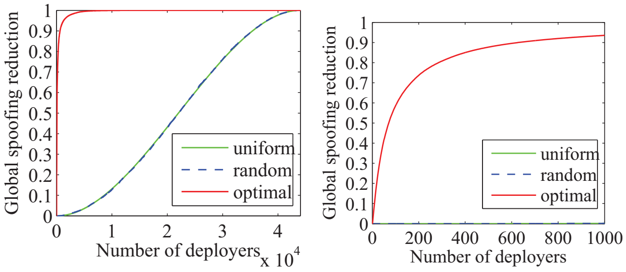


Fig. 5: Deployment incentives of DISCS functions

3) Optimal Deployment Strategy: DISCS provides continuously increasing incentive, but the initial incentive is low. In the real world, such collaborative security mechanisms often require the governments and industry groups to convince a small set of early deployers to accomplish the initial deployment, and then depend on the incentive of initial deployment to motivate the followers [38]. So the question here is that if we are asked to choose m early deployers, which ASes should be chosen to maximize the incentive for the followers. We prove that the optimal strategy is to choose the m largest ASes, where the size of AS j is the size of its address space r_j. The proof is provided in the supplementary material to save space here.

Because of the high variance of the address space sizes of ASes on the Internet, the optimal strategy can be very effective. Figure 6a shows the cumulated routable address space ratio of chosen ASes. If we always choose the largest ASes, the cumulated ratio grows very fast in the beginning. Instead, if we choose the ASes at random, the cumulated ratio grows linearly. If the address space sizes were uniformly distributed, the cumulated ratio would also grow linearly. Figure 6b shows that the incentive of DISCS almost grows perpendicularly at the beginning under the optimal deployment strategy. If we zoom in to the early deployment stage (Figure 6c), 50 largest ASes generate the incentive of 0.68, i.e., an LAS can reduce 68% of the spoofing traffic attacking it by deploying DISCS. This value increases to 0.88 when 200 largest ASes deploy DISCS. That is to say, DISCS has very high initial deployment incentive if a small set of ASes is strategically selected as early deployers.



(a) Whole deployment process

(b) Early stage

Fig. 7: Effectiveness of DISCS

B. Effectiveness

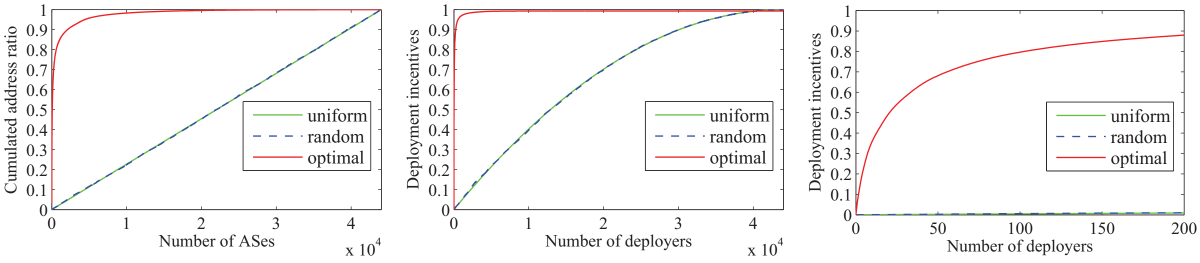
We use the similar setting to simulate DISCS's effectiveness, the reduction of the global spoofing traffic, when all functions are enabled for all traffic all the time. The results are shown in Figure 7. Under random deployment, effectiveness grows almost linearly. Under the optimal deployment strategy, 41% of the global spoofing traffic can be reduced when the 50 largest ASes deploy DISCS, and the reduction becomes 90% when the 629 largest ASes deploy it.

C. Cost

There are around 43k ASes and 442k routable IPv4 prefixes on the current Internet. We use these numbers to estimate the resource consumption on DISCS controllers and routers.

1) Controller: **Storage overhead:** A controller needs to maintain information for peers and IP prefixes. Each AS number (4B) is mapped to whether it is blacklisted (1B), whether it is a peer (1B), and the stamping key and verification key (2*16 = 32B). So the memory overhead for ASes is 43k*(4+1+1+32) = 1.6MB. Each IP prefix (5B) is mapped to an AS number (4B) and the start and end time of the four functions (4*2*8 = 64B). So the memory overhead for prefixes is 442k * (5+4+64) = 31.5MB. Since the per connection memory consumption of SSL is less than 10kB [39], if a controller concurrently communicates with all the other controllers, the required memory is 43k*10k = 430MB. Thus, the total memory overhead is 1.6+31.5+430 = 463.1MB.

Computation and network overhead: The control plane includes DAS discovery, peer relationship setup, key negotiation and function invocation. The overhead of DAS discovery and peer relationship setup is negligible since they are both one-shot effort through the entire life cycle. If keys are renegotiated every 10 days, a controller only needs to handle 6.1



(a) Address space ratio (b) Deployment incentives (c) Deployment incentives (early stage)

Fig. 6: Optimal deployment strategy on the asymmetric Internet

negotiations per minute on average. If there are 1611 attacks per day[6] and every attack triggers a function invocation, a controller will receive only 1.1 invocation requests per minute. The most intensive task is function invocation when a DAS is under attack. If the controller must react to the attack in 5 minutes, 147 SSL connections must be processed per second. Fortunately, recent implementations of SSL make it possible for a low-end dual-core CPU (Intel Atom @1.66GHz) to process more than 2000 SSL connections per second [41]. Thus, the CPU utilization is around 7.3%. Assuming that each connection consumes 1.5kB data with SSL session cache, the bandwidth cost is 1.76Mbps.

2) Router: **Storage overhead:** The Pfx2AS table and function tables map a prefix to an AS number (4B) and the functions (1B) to be executed on the prefix, respectively. We use 1 byte to store the functions since only 6 bits are needed (1 bit for In-Src, 1 bit for In-Dst, 2 bits for Out-Src and 2 bits for Out-Dst). These tables can be combined with the existing routing tables on routers to avoid consuming ternary content-addressable memory (TCAM). Key tables map an AS number (32b) to a stamping key and a verification key (2*16 = 32B). In total, DISCS consumes 442k* (4+1) + 43k*32 = 3.5MB static random-access memory (SRAM) and 43k*32b content-addressable memory (CAM) of a router.

Computation overhead: Actually, to support line-speed processing, functions must be performed in hardware instead of general CPUs. DP and SP can be supported by existing hardware since address based packet filtering is a common function in modern routers. AES-CMAC must be implemented in hardware to support CDP and CSP. Existing implementations on ASIC and FPGA can achieve the maximum message throughput of around 2Gbps per IP core [42] [43], i.e., 8Mpps and 5.33Mpps, or 26.25Gbps and 18.33Gbps assuming 400B payload size, for IPv4 and IPv6 packet throughput, respectively. Given that DISCS functions are invoked on demand and only executed on the traffic related with the victim, we believe existing hardware can deal with most attacks. Besides, following the suggestion proposed in Section VI-A2 (DP or SP should be invoked whenever CDP or CSP is invoked) can further reduce the load on CDP or CSP since many packets are previously dropped by DP or SP.

Network overhead: A MAC enlarges an IPv6 packet by at most 8 bytes. Assuming the average payload size of 400 bytes, the goodput rate decreases by only 1.6%, and only for the traffic related with the victim. There is no additional network overhead for IPv4 packets.

D. False Positives

FP consists of IFP and OFP. DISCS is IFP-free since all functions are end or e2e based. DISCS is designed to require minimal manual intervention to avoid OFP. However, since RPKI has not been universally deployed, we have to rely on manual input of the prefixes owned by the local DAS. Hence misconfigurations may cause genuine packets to be identified as spoofed. In this case, using the alarm mode for debugging manual configurations can alleviate or avoid OFP.

E. Security

This subsection analyzes the extent to which an attack can bypass DISCS to successfully send spoofing traffic.

1) MAC Forgery: In a brute-force MAC forgery attack, an attacker is expected to send 2^{28} and 2^{31} packets to make one correct guess, for IPv4 and IPv6, respectively. In other words, DISCS mitigates brute-force attacks by a factor of 2^{28} for IPv4 and 2^{31} for IPv6. During re-keying periods, the factor becomes 2^{27} and 2^{30} since two keys are considered valid.

2) Replay Attacks: MACs are erased by the destination DAS to prevent the destination hosts from sniffing valid MACs. However, by sending a packet whose TTL expires right after the packet's crossing the border of the source DAS, the source host can learn the valid MAC from the returned ICMP or ICMPv6 TTL exceeded messages. This attack can be prevented by letting the routers of the source DAS inspect the inbound TTL exceeded messages and erase the encapsulated MACs. The overhead of the inspection should not become a new vulnerability since ICMP and ICMPv6 messages are rate limited in practice.

In case that an attacker obtains a valid MAC, she can replays it. However, since a MAC is bound to a unique msg, replayed packets must have identical msg. Thus a replayed attack is

[6] According to the report of Arbor Networks, there are 1128 DoS attacks per day [40]. The reported attacks take about 70% of the total attacks since about 70% of the ISPs are monitored. Thus, the approximate total number of attacks should be 1128/0.7=1611.

easy to detect by destination hosts. The hosts can defeat this attack by asking its provider to block the source address of this msg.

3) Key Leakage: The cryptographic property of the MAC generation algorithm protects keys from being inferred from MACs, not to mention that MACs are erased after verification. Even if an attacker obtains the keys of DAS j by compromising j's controller or routers, the damage is limited mainly to j. In fact, all j's peers become new potential innocents for both d-DDoS and s-DDoS against j, while j is the only new potential innocent for the attacks against j's peers. Once the key leakage is detected, j's controller actively renews all the stamping keys and requests its peers to renew the verification keys. Of course, j must fix the compromised devices to protect the new keys.

VII. Conclusion

Recent research has shown that combating distributed spoofing attacks require the collaboration among different ASes. Hence, a flexible, incrementally deployable and high-incentive inter-AS collaboration system is required. This paper proposes DISCS, a distributed collaboration system for inter-AS spoofing defense. We present the scalable and flexible control plane design and low-cost and incrementally deployable data plane design. With theoretical proof and extensive simulations using real Internet data, our evaluation results show that DISCS has strong deployment incentives, high effectiveness, minimal false positives, modest resource consumption and strong security.

References

[1] A. C. Darren Anstee and G. Sockrider, *Worldwide Infrastructure Security Report*, vol. IIX, 2014.

[2] D. B. Darren Anstee and G. Sockrider, *Worldwide Infrastructure Security Report*, vol. VIII, 2012.

[3] R. Dobbins and C. Morales, *Worldwide Infrastructure Security Report*, vol. V, 2009.

[4] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[5] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *17th USENIX Security Symposium*, 2008.

[6] D. Piscitello, "Anatomy of a dns ddos amplification attack," http://www.watchguard.com/infocenter/editorial/41649.asp.

[7] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of internet background radiation," in *the 4th ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2004.

[8] C. Labovitz, "Botnets, ddos and ground-truth," in *NANOG50*, 2010.

[9] B. Zigterman, "Largest ddos attack ever nails company that protects against ddos attacks," http://bgr.com/2014/02/11/largest-ddos-attack/, 2014.

[10] A. Bremler-Barr and H. Levy, "Spoofing prevention method," in *the 24th Annual IEEE International Conference on Computer Communications (INFOCOM)*, 2005, pp. 536–547.

[11] X. Liu, A. Li, X. Yang, and D. Wetherall, "Passport: Secure and adoptable source authentication." in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2008, pp. 365–378.

[12] B. Liu, J. Bi, and X. Yang, "Faas: Filtering ip spoofing traffic as a service," in *ACM SIGCOMM*. ACM, 2012, pp. 113–114.

[13] B. Liu, J. Bi, and A. V. Vasilakos, "Towards incentivizing anti-spoofing deployment," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 9, no. 3, pp. 436–450, 2014.

[14] P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing," RFC 2827, 2000.

[15] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed dos attack prevention in power-law internets," in *ACM SIGCOMM*, 2001, pp. 15–26.

[16] F. Baker and P. Savola, "Ingress filtering for multihomed networks," RFC 3704, 2004.

[17] Z. Duan, X. Yuan, and J. Chandrashekar, "Controlling ip spoofing through interdomain packet filters," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 5, no. 1, pp. 22–36, 2008.

[18] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang, "Save: Source address validity enforcement protocol," in *the 21st Annual IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2002, pp. 1557–1566.

[19] H. Wang, C. Jin, and K. G. Shin, "Defense against spoofed ip traffic using hop-count filtering," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 1, pp. 40–53, 2007.

[20] A. Yaar, A. Perrig, and D. Song, "Stackpi: New packet marking and filtering mechanisms for ddos and ip spoofing defense," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 10, pp. 1853–1863, 2006.

[21] R. Beverly, A. Berger, Y. Hyun *et al.*, "Understanding the efficacy of deployed internet source address validation filtering," in *the 9th ACM SIGCOMM Conference on Internet Measurement Conference (IMC)*. ACM, 2009, pp. 356–369.

[22] "Spoofer project," http://spoofer.cmand.org/index.php.

[23] J. Mirkovic and E. Kissel, "Comparative evaluation of spoofing defenses," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 8, no. 2, pp. 218–232, 2011.

[24] T. Bates, R. Chandra, and E. Chen, "Bgp route reflection: An alternative to full mesh internal bgp (ibgp)," RFC 4456, 2006.

[25] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (bgp-4)," RFC 4271, 2006.

[26] M. Lepinski and S. Kent, "An infrastructure to support secure internet routing," RFC 6480, 2012.

[27] "Resource public key infrastructure (rpki)," https://www.arin.net/resources/rpki.html.

[28] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (s-bgp)," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 582–592, 2000.

[29] R. White, "Deployment considerations for secure origin bgp (sobgp)," http://tools.ietf.org/html/draft-white-sobgp-bgp-deployment-01, 2003.

[30] "The business value of ddos protection," *Arbor Networks Special Report*, 2011.

[31] B. Claise, G. Sadasivan, V. Valluri, and M. Djernaes, "Cisco systems netflow services export version 9," RFC 3954, 2004.

[32] P. Phaal, S. Panchen, and N. McKee, "Inmon corporation's sflow: A method for monitoring traffic in switched and routed networks," RFC 3176, 2001.

[33] J. Song, J. Lee, R. Poovendran, and T. Iwata, "The aes-cmac algorithm," RFC 4493, 2006.

[34] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, "Hash-based ip traceback," in *ACM SIGCOMM*, 2001, pp. 3–14.

[35] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed," in *the 7th ACM SIGCOMM Conference on Internet Measurement (IMC)*. ACM, 2007, pp. 111–116.

[36] S. Amante, B. Carpenter, S. Jiang, and J. Rajahalme, "Ipv6 flow label specification," RFC 6437, 2011.

[37] "Routeviews prefix to as mappings," http://data.caida.org/datasets/routing/routeviews-prefix2as/.

[38] P. Gill, M. Schapira, and S. Goldberg, "Let the market drive deployment: A strategy for transitioning to bgp security," in *ACM SIGCOMM*, 2011.

[39] V. Bernat, "Ssl termination: Stunnel, nginx & stud," http://vincent.bernat.im/en/blog/2011-ssl-benchmark.html, 2010.

[40] "Distributed denialof service attacks - global insights and mitigation techniques," Arbor Networks Special Report, 2008.

[41] B. Assmann, "Benchmarking ssl performance," http://blog.exceliance.fr/2011/09/16/benchmarking_ssl_performance/, 2011.

[42] "Aes cores," http://www.heliontech.com/aes.htm, helion Technology.

[43] "Cmac and xcbc aes core," http://www.ipcores.com/AES_CMAC_XCBC_IP_core.htm, iP Cores.