IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 12, 2018, accepted October 30, 2018, date of publication November 9, 2018, date of current version November 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2879489

Network-Layer Accountability Protocols: A Survey

LIN HE, YING LIU, AND GANG REN
Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China
Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China

Corresponding author: Gang Ren (rengang@cernet.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grants 61772307 and 61402257 and in part by the Tsinghua University Self-Determined Project under Grant 2014z21051.

ABSTRACT Today's Internet is vulnerable to numerous attacks, including source spoofing, distributed denial of service, prefix hijacking, and route forgery. Network-layer accountability is considered as an effective deterrence tool which can be used to address these attacks. Much research has been devoted to improving network-layer accountability of today's Internet. In this paper, we first investigate the state-of-the-art network-layer accountability research and summarize a general definition of network-layer accountability. Next, we propose a network-layer accountability framework and present a taxonomy of network-layer accountability protocols according to accountability granularity. Furthermore, we compare these protocols and discuss their pros and cons mainly from accountability function, deployability, and security. Finally, some open research questions are emphasized for directing future designs.

INDEX TERMS Internet security, network-layer accountability protocols, survey.

I. INTRODUCTION

It is well-known that the Internet has evolved from being an early American military computer network (ARPANET) to the current huge commercial commodity. Although the Internet has achieved a splendid success, it fails to prevent a variety of attacks, which result in huge damages and losses. Distributed denial of service (DDoS) attacks are still very prevalent [1]–[3], cost enormous business losses [4], and can even disable a country's Internet access [5], [6]. Prefix and route hijacking also knocked a country offline [7] or made famous web services globally unreachable [8].

It is believed that these vulnerabilities stem from the fact that the initial Internet design did not take security into consideration. To solve these security problems and defend the Internet, there are generally two options: protection and deterrence. The former aims to reduce vulnerabilities by hardening possible targets against attacks, minimizing the damages, and preventing attackers from using attacking techniques successfully. The latter aims to reduce the incentive for attackers to engage in security attacks through credible threats of retaliation. Accountability is considered as a deterrence tool that can be utilized to address such attacks we face today.

Basically, information security is the confidentiality, integrity, and availability of information [9]–[11]. Although accountability is not one of the general security attributes, it also gets involved with security [10], [12]. Generally, accountability is defined as a process of being called to account to some authority for one's actions [13]. Although many different definitions of accountability in different areas have been proposed, including network [14]–[19], software-defined networking [20], [21], distributed systems [22]–[24], cloud [25], [26], and web services [27], we consider that the core ideas of accountability are to confirm one's actions and assign responsibility.

As accountability has specific meanings in different areas, the research issues and solutions differ. In this paper, we focus on network-layer accountability, i.e., how to associate senders with their packets and provide a firmer foundation of network-layer security to the Internet. The essence of the Internet [28]–[30] is the architecture which contains protocols, algorithms, mechanisms, and frameworks. Although we have several options in the layered architecture to improve the accountability of the Internet, we believe that the network layer is the best choice. The reasons include:

- Firstly, the network layer is considered as the core of the Internet architecture. The current network layer, however, provides a weak security foundation for building upper layer applications. Many upper layer protocols need significant additional mechanisms and external support to fix their security problems. A firmer

foundation for the network-layer security will help decrease the security fixes for upper layer applications.

- Secondly, lower-layer protocols do not provide global unified Internet-level connectivity, which makes the accountability-enhancing mechanisms have limited performance and efficiency to account for misbehaving entities from different networks. Furthermore, various lower-layer protocols will increase the workload for designing accountability-enhancing mechanisms.
- Thirdly, designing accountability-enhancing mechanisms for various upper-layer protocols requires a huge amount of work, while the support of network-layer accountability will simplify the design of upper-layer accountability mechanisms. Moreover, we often need to consider the security issues brought by lower layers (e.g., the network layer) when designing upper-layer accountability-enhancing mechanisms.

So, what is accountability at the network layer of the Internet? In the Internet, the actions of an entity are sending packets to other entities. These actions are what the Internet aims to account for, especially in case of misbehaving entities. Therefore, network-layer accountability of the Internet is to determine who sent a specific packet.

However, the current Internet is lacking accountability. On the one hand, although accountability was a design goal of the Internet architecture, the Internet architecture has few tools for accounting for packet flows [31]. On the other hand, although IP addresses can be used to identify the hosts to be accounted for, it is not sufficient or efficient to use them and current related mechanisms to account for the hosts' misbehaviors. The reasons are as follows:

- **Widespread IP spoofing**: Currently, more than 30% of prefixes and about half of autonomous systems (ASes) are spoofable [32]. DDoS attacks are still very prevalent and result in huge damages and losses [1]–[3]. IP spoofing is so widespread that it is unreliable and difficult to associate packets with their sources and account for malicious entities.
- **Dynamic address assignment and translation**: Even if the source addresses of the packets are authentic, the prevalence of address assignment protocols such as dynamic host configuration protocol (DHCP) [33], [34] and stateless address autoconfiguration (SLAAC) [35], and address translation techniques such as network address translation (NAT) [36] makes it difficult and complex to use IP addresses to trace back traffic sources.
- **Omnipresent mobile hosts**: Today, mobile hosts are becoming omnipresent. A mobile host can have more than one address in an access network and have several different addresses in another access network. However, the mobility of the hosts makes it difficult and complicated to trace back hosts and account for their behaviors.

Because network-layer accountability can not only help solve burning security problems such as source address spoofing, DDoS attacks, prefix hijacking, and route forgery but also provide convenience to analyze hosts' behaviors and build flexible routing polices, we focus on network-layer accountability protocols and their functionality, deployability and security in this paper. To date, the main related survey is [37], which discussed accountability in different areas, such as theory and metrics, logging, the Internet and network, distributed systems, cloud, and smart grid. Compared with [37], this paper focuses on the network layer of the Internet and proposes a general framework of network-layer accountability and comparative analysis among existing protocols from accountability function, deployability, and security. The contributions of this survey mainly include the three following points:

- A general definition of network-layer accountability is summarized.
- A framework, taxonomy, and comparative analysis of network-layer accountability protocols are proposed.
- Open research questions and future research directions are proposed for designing new network-layer accountability mechanisms.

The rest of the paper is organized as follows. Section II discusses the definition of accountability at the network layer of the Internet. Section III aims to provide a general network-layer accountability framework and classify the network-layer accountability into three categories according to accountability granularity: AS-targeted, host-targeted, and user-targeted. Section IV, V, and VI describe AS-, host-, and user-targeted accountability protocols, respectively. The analysis and discussion of the accountability protocols are provided in Section VII. Finally, we conclude the paper and provide open research questions in Section VIII.

II. NETWORK-LAYER ACCOUNTABILITY

Generally, accountability is defined as a process of being called to account to some authority for one's actions [13]. Described in an account-giving manner, $\mathcal{A}$ is accountable to $\mathcal{B}$ when $\mathcal{A}$ is obliged to inform $\mathcal{B}$ about $\mathcal{A}$'s (past or future) actions and decisions, to justify them, and to suffer punishment in the case of eventual misconduct [38]. In computers, accountability refers to holding a user responsible for any behaviors on the computer, including sending packets, installing a software, and modifying a firmware. Even in the computer science, accountability in different branches has different connotations. PeerReview [23] attempts to provide accountability by detecting general Byzantine faults [39], [40], linking to and exposing faulty nodes in distributed systems. CATS [24] is a network storage service with strong accountability properties by building on a history of work on authenticated data structures and incorporating the state of the art in that area into a prototype. AudIt [19] is an explicit accountability interface through which Internet Service Providers (ISPs) can send back information related to loss and delay to help traffic source find out which unit to blame. AVMs [25] are accountable virtual machines that can provide users with the capability to detect faults, to identify the faulty node, and to produce evidence that connects the faults to the corresponding machine.

TABLE 1. Definition of network-layer accountability in different arts.

Scheme	Definition
AIP [15]	The Internet architecture has fundamental ability to associate an action with the responsible entity.
IPA [18]	Accountability, the ability to identify misbehaving entities and deter them from misbehaving further.
P-Accountability [41]	Accountability implies that an entity should be held responsible for its own specific actions or behaviors so that it can be part of a larger chain of accountability.
Persona [42]	Accountability can be defined as the state of a subject being held responsible for a certain action taken by that subject.
APIP [43]	Hosts cannot send traffic with impunity: malicious behavior can be stopped and perpetrators can be punished.
APNA [44]	Source accountability refers to an unforgeable link between the identity of a sender host and the sent packet.
AaaS [14]	Accountability in a network is a means to identify the sources of traffic for two purposes: to selectively filter repeated unwanted, abusive, or non-compliant traffic from malicious sources on a per-destination basis, while permitting traffic from others to proceed, and to report and disconnect abusive machines before they attack others.
BAFI [16]	Accountability is defined as "an obligation to accept responsibility for one's actions", and it assumes that actions can accurately be linked to their sources and that sources can be punished for bad, and awarded for good behavior.

Ujcich *et al.* [20] propose an accountable SDN architecture which incorporates various notions of accountability for achieving system-wide cyber resiliency goals. SDNsec [21] provides an SDN security extension that provides accountability for the SDN data plane.

Before starting further discussions, we introduce two fundamental terms: **accountee** and **accountor**. The accountee is the entity that is accountable to the accountor. Based on the basic definitions and applications in different areas of accountability, we consider that the general definition of accountability contains four significant properties:

- **Entities**: All the entities that can take actions in a system can be accountees in the accounting process.
- **Actions**: Actions are taken by an entity. The entity will be an accountee in case that observed actions cause damages.
- **Proofs**: The accountor can provide proofs of authenticity of specific actions taken by the accountee.
- **Countermeasures**: Countermeasures should be provided when an accountee misbehaves. In another word, an accountee should suffer a punishment in case of misbehaviors.

So, what is accountability at the network layer of the Internet? Although different arts have different definitions of network-layer accountability as shown in Table 1, all these definitions are applicable to the scope of the general definition we provide above. More specifically, the actions of sending packets taken by specific entities are what the Internet aims to account for. From different points of view, the responsible entities can be different. The responsible entities at the network layer can be ASes, hosts, or users, etc. If an entity is judged to have taken specific actions, the countermeasures can be detecting, filtering, and stopping the (following) misbehaviors of the entity. Therefore, network-layer accountability of the Internet is to determine whether an entity has sent a specific packet and provide countermeasures in case of misbehaviors.

Once we can achieve network-layer accountability in the Internet, we can regulate the behaviors of hosts and improve the user experience and the performance of the Internet.

- **Enforce the network-layer security**. On the one hand, the accountability of the Internet makes it impossible for hosts to send traffic with impunity. Administrative or legal actions can be used to punish the perpetrators. On the other hand, effective accountable mechanisms of the Internet help solve many attacks, including source spoofing, DDoS, prefix hijacking, and route forgery, and improve the performance of the Internet.
- **Analyze hosts' behaviors**. The ISPs can analyze the traffic of the hosts and provide customized services to the hosts.
- **Build routing policies**. The ISP can build finer-grained and more efficient routing policies to forward traffic from different types of users.

III. ACCOUNTABILITY FRAMEWORK AND CLASSIFICATION

In this section, we propose a useful framework which can be used as a guide to accountability protocol designs. Also, we present a taxonomy of existing network-layer accountability protocols.

A. ACCOUNTABILITY FRAMEWORK

Through the arts we investigated, we find that network-layer accountability can be achieved by providing the following four properties at the network layer corresponding to the four tuples mentioned in Section II:

- **Identify each sender**. In the network, the responsible entities are packet senders. To account for a sender, it should be assigned a unique identifier in a network. If multiple users or hosts have a same identifier, it will lead confusions in determining the true sender identity. Sender identifiers can be used to build unforgeable and undeniable connections with packets through direct or indirect methods. More importantly, the identifiers should be distributed and managed by trustworthy entities rather than users [44], [45].
- **Ensure the authenticity of packets**. In the Internet, the packets are the evidence that a sender has taken such actions. Therefore, the authenticity of packets is a basic requirement when using packets as unforgeable

TABLE 2. Granularity classification of network-layer accountability.

Types	Accountees	Explanation	Examples
AS-targeted accountability	Source AS	Determine whether the source AS has sent a specific packet.	Passport [50], FAIR [56]
Host-targeted accountability	Hosts	Determine which host has sent a specific packet.	SAVI [46], AIP [15]
User-targeted accountability	Users	Determine which user has sent a specific packet (using a host).	NIDTGA [54], TrueID [57]

and undeniable evidence in case of misbehaviors. Many mechanisms are proposed to ensure the authenticity of packets, including anti-spoofing [46]–[48] and integrity protection [44], [49]. As hosts may be malicious, decisions on whether packets are real or not cannot be made only by hosts. At least another one trustworthy entity should have the ability to check the authenticity of the packets sent by hosts.

- **Associate packets with senders**. The protocol must associate sender identities with their sent packets. If the packets of an accountable protocol cannot be used to trace back the sender identities, the protocol lacks accountability. Note that the associations between packets and user identifiers made by association methods must be unforgeable. If the associations are easy to forge, attackers can easily forge packets which may appear to be sent by a benign source. Moreover, a source can deny having sent a packet. Generally, there are two types of association methods: out-of-packet association and in-packet association. The former refers that store the binding between a sender identifier and some fields of a packet. This method requires an AS to create and maintain an additional mapping table. The latter refers to embedding a user identifier into some field(s) of a packet with plaintext or ciphertext. This method allows an AS to get a user identity from some field(s) of a packet in a stateless manner.
- **Provide Accountability countermeasures or services**. Countermeasures or services should be provided to detect and filter misbehaviors, and even deter misbehaved senders from misbehaving further. Accountability countermeasures show how to use accountability to stop ongoing attacks or prevent future ones. Currently, typical countermeasures include source validation, shutoff, traceback, and reputation systems. Source validation checks if the packets have originated from the claimed source [46], [50]–[53]. Shutoff is used to allow destination hosts to selectively block traffic from particular source hosts [15], [43], [44]. Traceback is a type of service that uses methods to reliably determine the origin (user or host identity) of a packet in the Internet [54]. Reputation systems can be generally divided into two types: provider reputation systems [55] and client reputation systems [16]. The latter can be used to help service providers select trustworthy clients and provide better and high-priority services. This will require that service providers know a client's reputation score. People can also define other accountability countermeasures.

B. CLASSIFICATION BASED ON ACCOUNTABILITY GRANULARITY

The accountees of network-layer accountability protocols from coarse-grained to fine-grained granularity can be the source AS itself, hosts, and users. Therefore, we divide these protocols into three categories based on accountability granularity, and the results are shown in Table 2:

- **AS-targeted accountability**: Determines whether the source AS has sent a specific packet.
- **Host-targeted accountability**: Determines which host has sent a specific packet.
- **User-targeted accountability**: Determines which user has sent a specific packet.

In fact, user-targeted accountability and host-targeted accountability are the finer-grained version of AS-targeted data-plane accountability. When user-targeted accountability and host-targeted accountability are established, AS-targeted data-plane accountability is also established. Below we will discuss and analyze the design methods of network-layer accountability protocols in each category.

IV. AS-TARGETED ACCOUNTABILITY

In fact, many works related to AS-targeted accountability have been proposed. We divide AS-targeted accountability protocols into two categories: AS-targeted data-plane accountability and AS-targeted control-plane accountability. The former refers that holds ASes responsible for their data-plane traffic. The latter refers that holds ASes responsible for the traffic which they send to build routing information.

For AS-targeted data-plane accountability, there are generally two options for determining whether the source AS has sent specific data packets: self-verified and other-verified. The former refers that the source AS inserts proofs into packets, which can be used by participating ASes on the forwarding path to verify whether specific packets are from their claimed ASes, e.g., Passport [50]. The main idea is to validate source addresses from the inter-domain level. The latter is that transit ASes help determine whether the source AS has sent specific packets by inserting the proofs into packets they have forwarded, e.g., FAIR [56]. Currently, several surveys or reviews [58], [59] are dedicated to the discussion and comparison among inter-domain address validation technologies. Therefore, we will not pay more attention to these inter-domain address validation technologies but introduce two schemes that are used in host-targeted accountability protocols, i.e., ingress filtering [47] and Passport [50].

As for AS-targeted control-plane accountability, secure inter-domain routing techniques are mainly used to

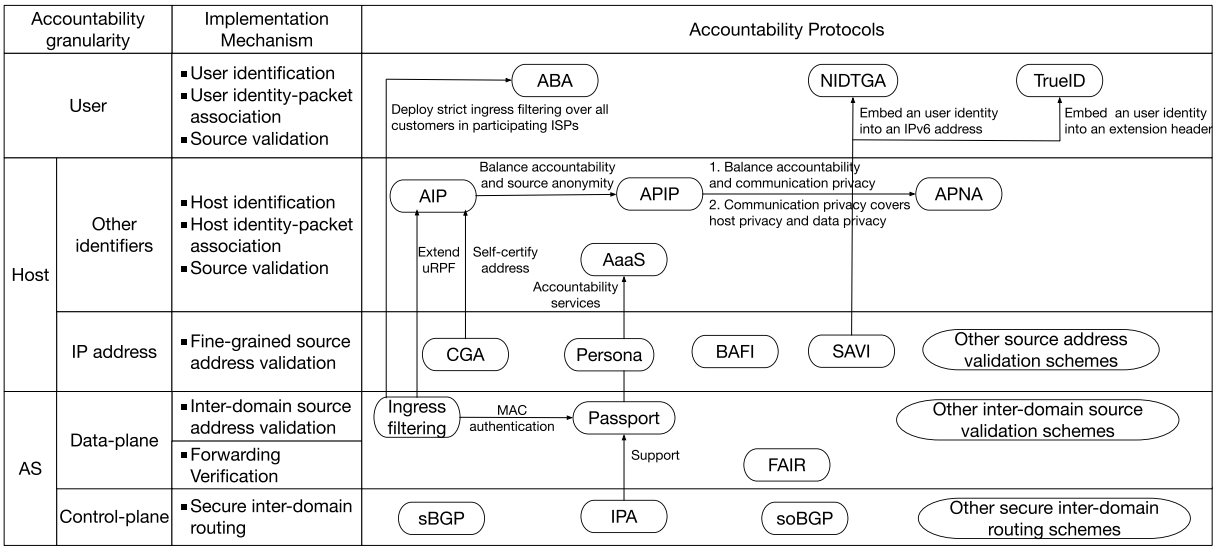

FIGURE 1. Evolution of network-layer accountability techniques.

achieve AS-targeted control-plane accountability, such as S-BGP [60], soBGP [61], psBGP [62], SPV [63], origin authentication [64], and RPKI [65]. As many great studies [66]–[68] have focused on this area, we will not conduct a detailed discussion of these protocols in this paper. But we will discuss one special protocol named IPA [17], [18] which enables securing routing protocols such as S-BGP [60], soBGP [61] to announce prefixes and achieve origin authentication and AS path authentication.

A. INGRESS FILTERING

Network ingress filtering [47] is a simple, effective, and straightforward method that ISPs can use to filter packets with forged IP addresses and prevent the acceptance of spoofed packets. When an edge device in an ISP installs ingress filtering, it will examine every inbound packet and filter the packets with source addresses belonging to the prefix announcements of the network that the edge device resides in. Although this method ensures that an outside sender cannot spoof the addresses belonging to the announced prefixes of a network, it cannot prevent an inside sender from spoofing other addresses in the same prefix.

Unicast Reverse Path Forwarding (uRPF) extends ingress filtering by requiring routers or firewalls to check whether the packet arrives on the interface which would be used to forward the traffic to the source address of the packet. However, uRPF will be ineffective due to route asymmetry. Several possible relaxations to uRPF are also provided to allow it to be more effective even in the case of route asymmetry [69].

At least five ways can be used to implement ingress filtering. These ways include: ingress access lists, strict reverse path forwarding, feasible path reverse path forwarding, loose reverse path forwarding, and loose reverse path forwarding ignoring default routes [69].

To a certain degree, ingress filtering provides accountability. An administrative network can be held accountable for its packets.

B. PASSPORT

Passport is a network-layer source authentication system that allows source addresses to be validated to the granularity of the origin AS within the network. As shown in Figure 2, when a packet leaves its own source AS, the border router will stamp a message authentication code (MAC) along the network path into the Passport header of the packet. Each MAC covers the source address, the destination address, the IP identifier, the packet length field of the packet, and the first 8 bytes of the payload, and is computed using the shared secret key between the source AS and each AS along the network path.

The shared secret key is distributed by piggybacking a Diffie-Hellman key exchange [70] on BGP routing advertisements. Other benefits that Passport gains from distributing shared secret key within the inter-domain routing system include bootstrapping the key distribution and efficiency.

When an AS along the network path receives an incoming Passport packet, the border router first checks whether the corresponding MAC value is valid using the secret key shared with the source AS. Because the correct MAC value can be only computed by the source AS, it is obvious that a verified packet must come from the source AS indicated by the source address. Otherwise, the Passport packet with invalid MAC value is source spoofed.

Passport also allows inter-operation of two upgraded ASes even if there are legacy ASes between them. An upgraded AS will discard a legacy packet with the source AS and destination AS both deploying Passport. If the source AS of a packet deploys Passport but the destination AS not, the upgraded AS will demote the packet.

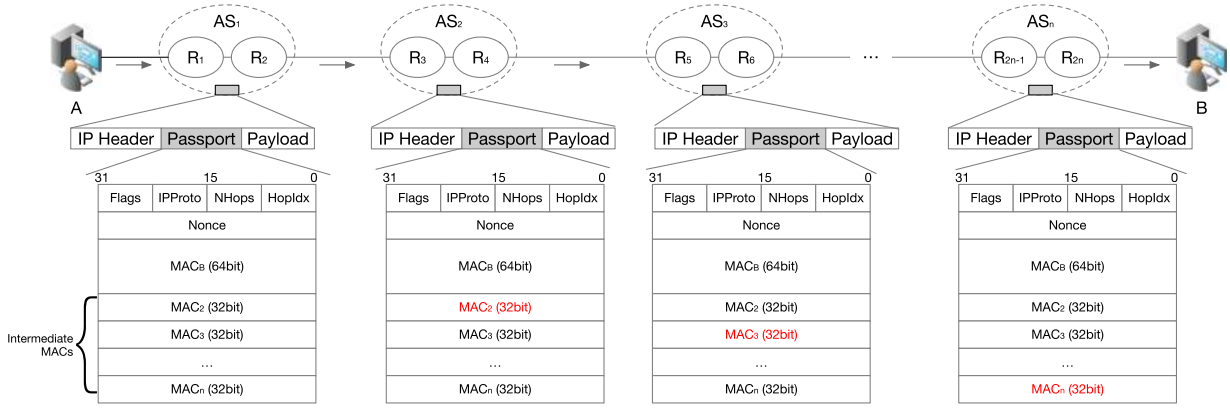

FIGURE 2. A high-level overview of passport.

C. IPA

IPA [17], [18] aims to provide the same accountability guarantees as Accountable Internet Protocol (AIP) [15] but keep compatible with current IP. IPA tries to secure the routing infrastructure of the current Internet and leverages existing mechanisms to provide accountability. Bootstrapping accountability in the Internet with lightweight and adoptable enhancements is the central design goal of IPA.

IPA retains the hierarchical IP address structure but adopts another addressing convention of AIP–Autonomous system numbers (ASNs) are generated from public keys. Because ASNs are currently flat, replacing them with ASes' self-certifying identifiers does not have an influence on the existing inter-domain routing protocols. IPA uses top-level reverse DNSSEC hierarchy other than a separate PKI to certify an ASN's ownership. An IP prefix is bound to the public key of an AS. The hash of the public key is used as the ASN. These bindings are stored as signed reverse DNSSEC records. Internet Assigned Number Authority (IANA), the root Internet registry, delegates prefix allocations and the corresponding reverse DNS zones to RIRs. RIRs can also sign records that certify sub-delegations to ASes. Therefore, prefix owners can use the DNSSEC records to authorize their prefix ownership.

After creating these secure bindings, IPA enables secure routing protocols such as S-BGP [60], soBGP [61] to announce prefixes and to achieve origin authentication and AS path authentication. It is assumed that IANA's root public key is globally known. One can query the corresponding DNSSEC records to validate the announcements. IPA also enables other implementations such as Passport [50] to provide accountability at an AS level, NetFence [71] to defend DDoS attacks. Compared with AIP, IPA does not require host renumbering or trusted host hardware.

D. FAIR

Forwarding Accountability for Internet Reputability (FAIR) [56] is an architectural mechanism that leverages forwarding accountability to incentive Internet service providers (ISPs) to apply stricter security policies to their customers. Forwarding accountability refers to holding transit ASes accountable for the traffic they forward. Transit ASes will embed short cryptographic proofs within the packets that a destination AS will show to the transit ASes to prove the fact that they have indeed forwarded the malicious traffic. In FAIR, packets collect proof that will remind the transit ASes of having forwarded these packets rather than carry capabilities in previous arts.

Communications under FAIR include three phases, namely 1) setup, 2) transmission, and 3) protest. For the first phase, source and destination ASes should set up a channel with sending a traffic policy formally expressed by the Token Bucket (TB) parameters [72] from the source AS to the destination AS. The policy communicated in the channel can be a specification of the average sending rate, the maximum burst size, or even the forbidden abnormal packet headers. Other future Internet proposals can also replace this setup phase.

The second phase is the data transmission process, which are mainly operated by the source AS, cooperating ASes, and the destination AS. In the source AS, after a host sends data packets over the known path, the source AS's border routers enforce the sending policy by applying the parameters to the TB and embed extra information within the packet to be used to construct proofs of violations thereafter. In the cooperating transit ASes, every egress border router verifies whether the source's timestamp in the packet deviates from the local time beyond a threshold and then marks the packet with a cryptographic MAC to prove that it indeed forwarded the packet. The destination AS monitors the communication channel to detect sending policy violations and stores packet headers containing the cryptographic markings for the proof of misbehaviors.

Finally, Phase 3 is used to make the destination ASes protest to other ASes provably. The sending policy and the data packet headers should be provided by the destination AS to all cooperating transit ASes along the routing path. All the cooperating transit ASes verify the proof and acknowledge or reject the complaints. The destination AS

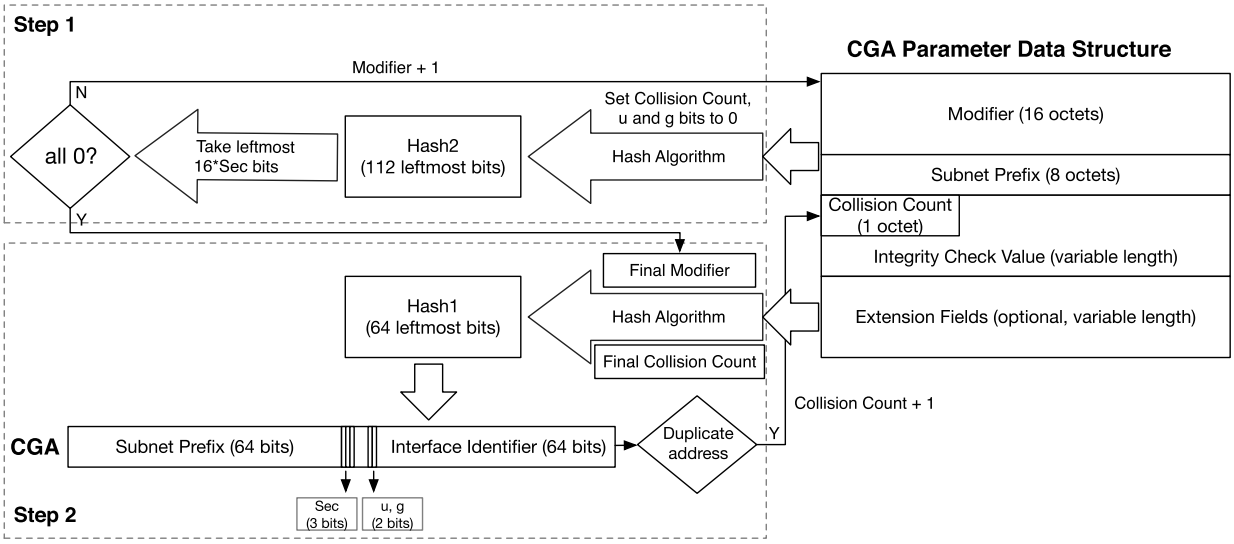

FIGURE 3. Generation process of CGAs.

will collect the complaint results. Furthermore, the destination AS sends the collected results back to all the cooperating transit ASes. Consequently, all the transit ASes along the path know whether the source AS is compromised.

V. HOST-TARGETED ACCOUNTABILITY

In this section, we describe the state of the art of host-targeted accountability. Host-based accountability refers to the ability that holds hosts responsible for the packets they have sent. To establish host-targeted accountability at the network layer, hosts must be identified uniquely first. Then, each packet originated from a host must be authentic. According to the host identifiers, we classify the host-targeted accountability protocols into two categories: IP address-based and other identifier-based. The former category uses IP addresses as the targets to be accounted for. A common idea to achieve such kind of accountability is to validate source addresses. Many related surveys have been proposed to address this issue. Therefore, we only discuss several mechanisms used by the latter category of protocols. Other identifier-based host-targeted accountability protocols often identify hosts by public keys or other self-defined identifiers.

A. SOURCE ADDRESS VALIDATION IMPROVEMENT

Source Address Validation Improvement (SAVI) methods are developed to prevent nodes attached to the same IP link from spoofing each other's IP addresses, so as to complement ingress filtering with finer-grained, standardized IP source address validation [46]. SAVI methods should be designed to be purely network-based so as to enable network operators to deploy fine-grained IP source address validation without any dependency on hosts. A three-step model is used to instruct a SAVI instance to enforce the host's use of legitimate IP source addresses. Firstly, through monitoring traffic originated from a host, a network can identify which IP source addresses are legitimate for the host. Secondly, a legitimate IP address must be bound to a link-layer property of the network the host attached to (binding anchor). Binding anchor must be more difficult to spoof than the host's IP source address. Thirdly, every time a host sends a packet, the source address of the packet should be checked whether it matches the binding anchor. Note that the closer a SAVI instance is located to the host, the more effective the SAVI method is. The binding anchor can be the IEEE extended unique identifier, the port on an Ethernet switch to which a host attaches, the security association between a host and the base station on wireless links, the combination of a host interface's link-layer address and a customer relationship in cable modem networks, an ATM virtual channel, a PPP session identifier, a Layer 2 Tunneling Protocol (L2TP) session identifier in a DSL network, and a tunnel that connects to a single host [46].

Currently, several SAVI documents have been standardized based on the different address assignment techniques (e.g., SLAAC [35], DHCP [33], [34], and secure neighbor discovery (SEND) [73]), including SAVI-FCFS [74], SAVI-DHCP [75], SAVI-SEND [76], and SAVI-MIX [77].

B. CRYPTOGRAPHICALLY GENERATED ADDRESSES

Cryptographically Generated Addresses (CGAs) [51] are IPv6 addresses generated by computing a cryptographic one-way hash method from a public key and some auxiliary parameters. To protect the security of the SEND protocol [73] in IPv6, the Internet Engineering Task Force proposes and standardizes CGAs in 2005. The main goal of CGAs is to prevent stealing and spoofing of existing IPv6 addresses. CGAs can also give the same level of pseudonymity of temporary addresses defined in [78].

Figure 3 presents the process of generating CGAs. A security parameter (Sec) is bound with a CGA to determine the strength against brute-force attacks. The interface

identifier is generated from modifier (any random 128-bit unsigned integer), subnet prefix, collision count, public key, 3-bit Sec, and optional extension fields using hash algorithms [79]. Once an address collision is detected after duplicate address detection [35], the host increments the collision count by one and regenerates a CGA.

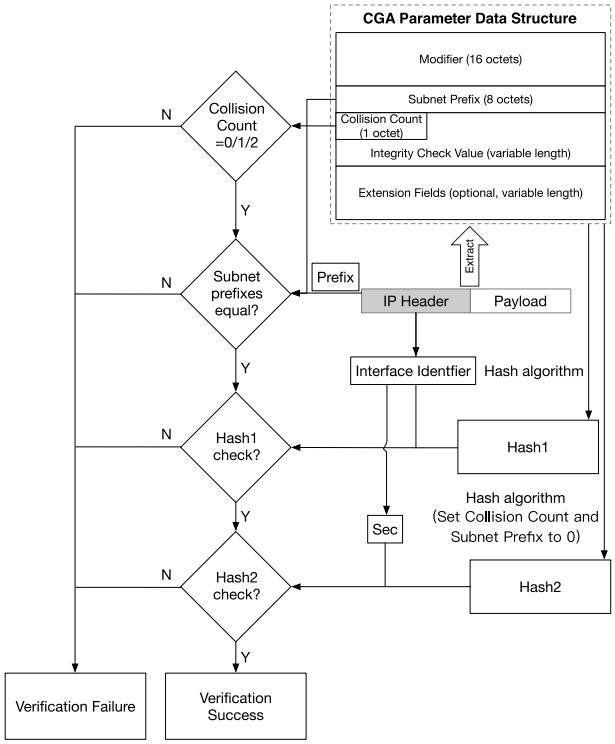

FIGURE 4. Verification process of CGAs.

Figure 4 shows the verification process of CGAs. The verification process mainly contains four parts, including collision count check, subnet prefix check, Hash1 check, and Hash2 check. Firstly, check the collision count is 0, 1, or 2. Secondly, check the subnet prefix in the CGA parameters data structure is equal to the subnet prefix of the address. Thirdly, generate Hash1 from the CGA Parameters data structure and compare Hash1 with the interface identifier of the address (ignore the 1, 2, 3, 6, 7 bits). Finally, check the 16*Sec leftmost bits of Hash2 are equal to zero. If any step of the four checks fails, the verification fails. Although the verification process of CGAs does not require the support of extra security frameworks, it cannot prove that an address is not a CGA.

As CGAs are not certified, an attacker can create a new address from any arbitrary subnet prefix and its own or someone else's public key. However, it is difficult for an attacker to find a collision of the cryptographic hash value Hash1, so the attacker cannot impersonate someone else's address. Another limitation for CGAs is that the cost of address generation is very high, especially in the case of a high Sec value.

C. PERSONA

Persona [42] is a protocol that attempts to balance anonymity and accountability at the network layer of the Next Generation Internet (NGI). The goal of Persona is to combine accountability and anonymity, which appear mutually exclusive properties, in a stateless manner within routers. Therefore, Persona can help discover malicious nodes, while it keeps user identities anonymous and even allows users to choose different levels of anonymity.

In Persona, it is assumed that routers are installed with a Trusted Platform Module (TPM) [80], which is a microcontroller that stores keys, passwords and digital certificates. Symmetric secret keys shared between routers and the ISP are embedded within the TPM. When a user first connects to the network through a router of the ISP, the router uses the keys in the TPMs to encrypt the information (including the addresses of the routers that the user can contact as "first hop" and the shared keys between these routers and the user) exchanged between the user and the ISP. To achieve per packet anonymity, the sender binds an incremental sequence number (SN) to each packet it sends. Hence, Persona identifies each packet by two unique fields, the sender's IP and the SN.

Figure 5 shows the message exchanges between the sender, the routers and the receiver in Persona. The user encrypts the destination using the secret key shared with the router and sends a Persona packet to the router. The router maps the source IP address of the user to hide the source identity of the packet. Once the IP address has been changed, the router forwards the packet to the Internet. In this way, the receiver will not know how to respond to the sender, because it cannot know the true source address of the sender and there is no tunnel established.

As for the response to the sender, the receiver can create a packet with the destination of tuple $(IP_d||SN_d)$ and forward it to R_n. Note that the router needs to know that packet is being sent "forward" or "backwards" so that it can know whether to encrypt or decrypt the tuple $(IP||SN)$.

D. BUILDING ACCOUNTABILITY INTO THE FUTURE INTERNET

Building Accountability into the Future Internet (BAFI) [16] is a future Internet architecture aiming to prevent IP spoofing, DDoS attacks, distributed scanning and intrusions, and widespread worm infections. The architecture mainly includes three parts: source signatures, packet tickets, and reputation system.

Source signatures attached to the packets are the core of the architecture. Senders attach cryptographic signatures to each packet at the departure time. The signature depends on the source identity, the packet's header and contents. It is assumed that naming and location services will be separated in the future Internet. Identity and location identifiers carried in the packets should be globally unique. When an AS border router receives an outgoing packet, it first verifies the host-level signature. If succeed, the border router will replace the host-level signature with an AS-level signature. The packet will be verified by all the routers on the routing path until it reaches the destination host.

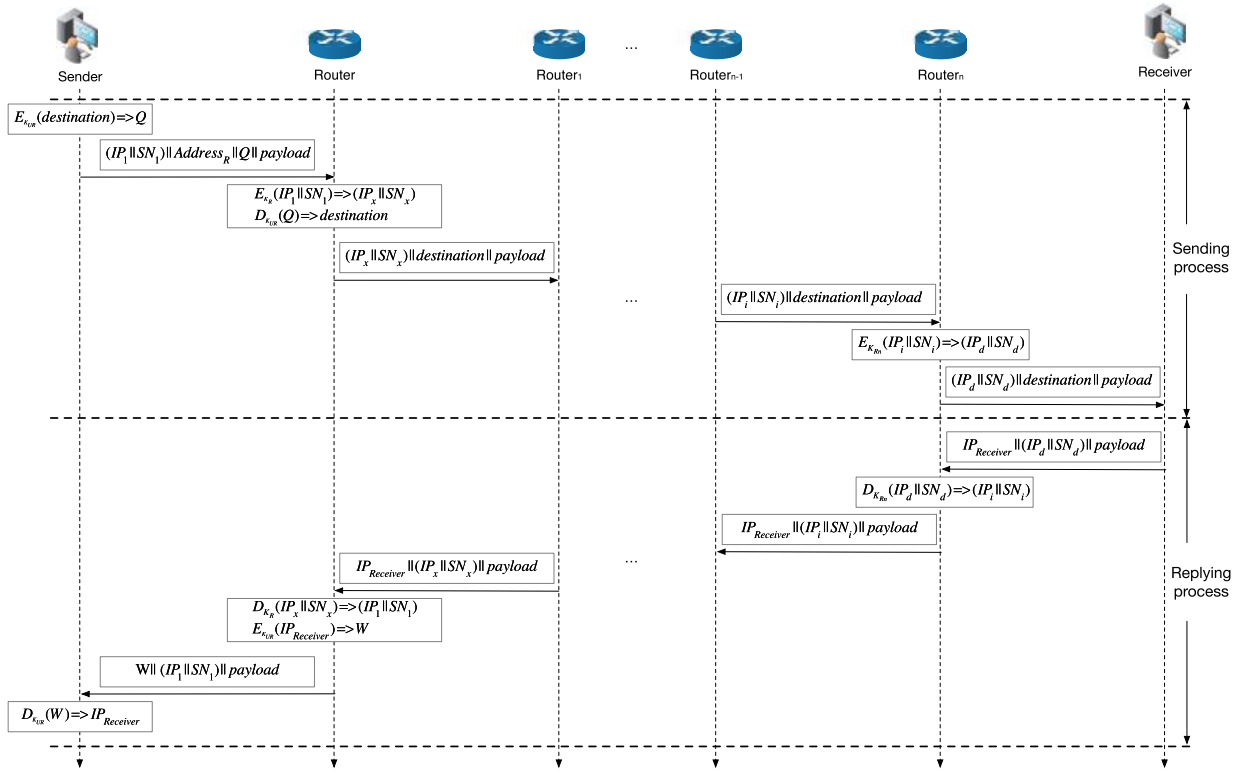

FIGURE 5. Message exchanges among the sender, the routers and the receiver in Persona.

The packet tickets acting as capabilities to reduce unwanted traffic lie in the second layer of the architecture. A client sends a ticket request to a server. The server determines whether it will grant access to the client. Routers on the routing path also verify tickets in all packets they forward. Note that a server cannot know the previous behaviors to decide whether to grant access to a new coming client. Two choices can be made to solve the problem. The first one is that the server makes the decisions based on the client's reputation from the reputation system. The second one is that the server grants a ticket to each new client.

The top layer of the architecture is a reputation system. It collects server reports about malicious client behaviors and provides information to servers that can be used to decide whether a client can be granted a ticket. It is assumed that servers have means of identifying misbehavior. Each report contains the identity of the misbehaved client and the content of the misbehavior. It is required that each report should be coupled with a traffic sample to prove the occurrence of the alleged activities.

E. ACCOUNTABILITY AS A SERVICE

Accountability as a Service (AaaS) [14] considers accountability as a first-class network service, independent of addressing and routing. An accountability service provides its authenticated clients with identifiers that can be used to mark packets accountable. The accountability service vouches for the packets of its authenticated clients and could not necessarily be the ISP. Internet users determine what

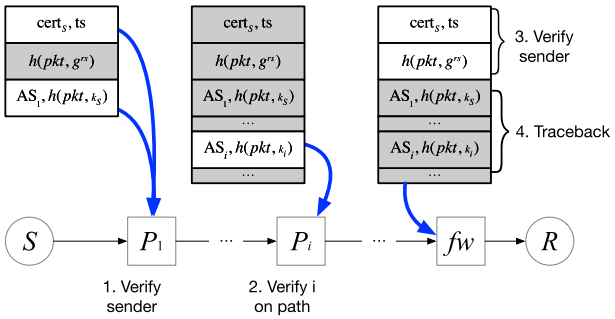

FIGURE 6. A high-level overview of AaaS [14].

accountability services they prefer and what level of accountability they require and ask the network to block traffic from any specific source for any reason. The sender and all the ISPs along the path should sign the signatures and add them in packets as shown in Figure 6. The receiver validates the sender certificate to learn which accountability service vouches for the sender. If a victim still receives unwanted traffic, it can trace back all the ISP signatures to find out the responsible ISP that did not check the packet and present the evidence to its accountability service.

F. ACCOUNTABLE INTERNET PROTOCOL

Accountable Internet Protocol (AIP) [15] is a network architecture that provides accountability as a first-order property. AIP eschews the use of CIDR-style addresses and employs an

innovative approach, a hierarchy of self-certifying addresses derived from public keys of the corresponding entities, to provide accountability at the network layer. Although this prohibits compatibility with the current Internet protocols (i.e., IPv4 and IPv6), it allows hosts and domains to prove they have the address they claim to have without relying on any global trusted authority. AIP also uses secure hardware (i.e., smart-NIC) in each well-intentioned host to allow destinations to block unwanted traffic, i.e., shut-off protocol.

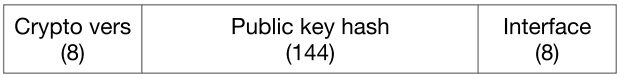

FIGURE 7. The AIP address structure [15].

In the AIP architecture, an accountability domain (AD) with a unique identifier is decomposed from the current administered networks and plays the role of today's AS. Each host is assigned a globally unique endpoint identifier (EID). The AIP address of a host currently homed in some AD would have an address of the form $AD : EID$. Both AD and EID of an address are hashed from public keys as shown in Figure 7. Each AIP address contains a cryptography version number that suggests what signature scheme incarnation was used to generate the address in case that the strength of cryptographic primitives degrades. Obviously, it is impossible for AIP addresses to be aggregated or routed based on current practice. For inter-AD routing, routers use only the destination *AD* to forward the packet before the packet reaches the destination *AD*. For intra-AD routing, routers forward the packet using only its *EID*. The AIP header packet is shown in Figure 8.

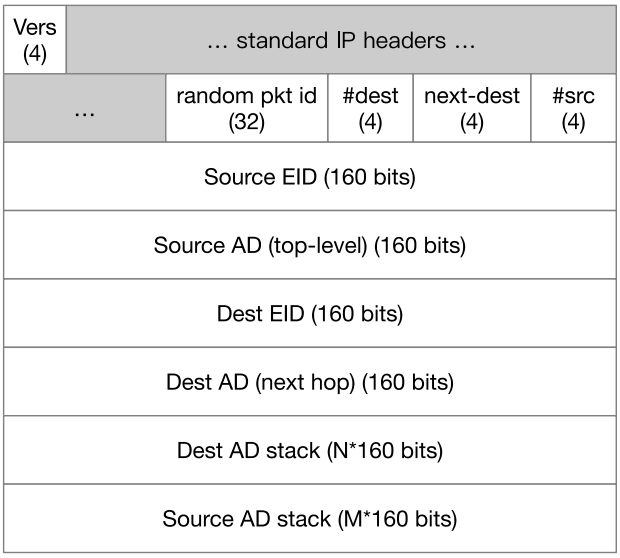

FIGURE 8. The AIP packet header [15].

The goal of using self-certifying address is to prevent source spoofing. Source accountability mechanism of AIP is an extension of "unicast reverse path forwarding" (uRPF) [47]. The goal of uRPF is to automatically filter packets when the route to the source address of the packets points to the same interface which the packet arrived. AIP combines uRPF with a second mechanism to automatically verify the authenticity of packets even though packets arrived on one interface and the reverse route points to another.

Because public keys are used to generate AIP addresses, it is possible to use public keys to validate source addresses. In AIP, source addresses are verified in two places in the network: at the first-hop routers and when crossing AD boundaries. If the source host S sends a packet but has not been verified by the first-hop router or switch R recently, R will drop the packet and send a *verification packet* V to S. S signs the packet V with the private key corresponding to its *EID* to prove it has identity *EID* and forwards the signature to R. R checks whether the signature is correct. If the signature is correct, it stores the information and forwards subsequent packets for S. Note that the host should re-send the packet that triggered the V.

When a packet crosses the boundary from AD_1 to AD_2, AD_2 must decide whether the source address of the packet is valid. Three possible cases exist. If AD_2 trusts AD_1 to have verified the source address of the packet, then AD_2 forwards the packet. Otherwise, AD_2 conducts uRPF checks to determine if the packet arrived on the same interface that the route to the source address of the packet points to. If the check succeeds, AD_2 forwards the packet. However, if both tests fail, AD_2 uses the same verification procedure as a first-hop router or switch and sends a verification packet to the source address $AD : EID$ of the packet. Similarly, if the sender can reply a correct signature, AD_2 forwards the subsequent packets and the router in AD_2 creates a new entry in its *accept cache* suggesting that it should forwards the packet from $AD : EID$.

A victim that receives unwanted traffic can throttle the traffic from the source by sending a *shut-off packet* (SOP) to the source. It is assumed that well-intentioned hosts are equipped with a trusted network interface card named smart-NIC. When receiving an SOP request, the smart-NIC first checks whether the SOP request is valid. If so, the smart-NIC installs a filter that throttles further packets to the victim for a period of time (TTL) specified in the SOP. It is required that SOPs include the hash of a packet recently sent by the NIC to the victim, which prevents replay attacks and spoofing-capable attackers from suppressing traffic between innocent hosts.

G. ACCOUNTABLE AND PRIVATE INTERNET PROTOCOL

Accountable and Private Internet Protocol (APIP) [43] is a new protocol that tries to strike a balance between accountability and privacy at the network layer. APIP considers the center of the tussle between accountability and privacy is the source address. Source addresses should be undeniable links between packets and senders in an accountable Internet, while source addresses should be hidden as much as possible in a privacy-preserving Internet.

APIP considers that source addresses try to play at least five distinct roles, including return address, sender identity, error reporting, flow ID, and accountability. APIP separates accountability and return addresses. Each APIP packet has at least two addresses, namely, a destination address and an accountability address. Each address contains three parts: 1) a network ID (NID), 2) a host ID (HID), and 3) a socket ID (SID). An NID is used to forward packets to the destination domain. An HID is used to forward packets to the host within the destination domain. An SID is used at the destination host to demultiplex packets to sockets.

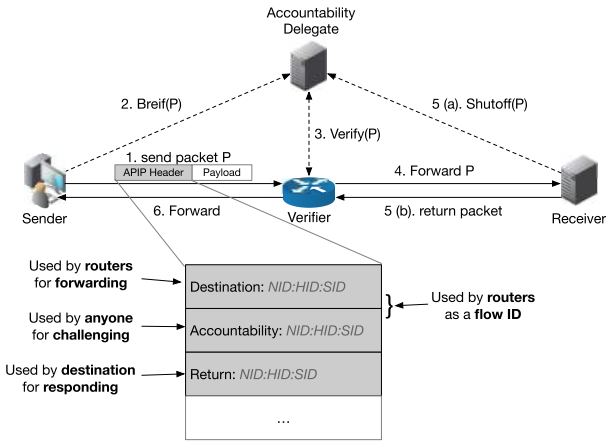

FIGURE 9. A high-level overview of APIP [43].

Figure 9 shows a high-level overview of APIP. Firstly, the sender sends a packet which contains an accountability address to identify its accountability delegate. Next, the sender sends the fingerprint of the packet to its accountability delegate. The fingerprint of the packet is used by its accountability delegate to vouch for the sender. Then, when the verifier receives the packet the sender sends, it can confirm with the sender's accountability delegate that the packet is really generated by the sender and vouched by the accountability delegate. If the packet is invalid, it will be dropped by the verifier. The verifier can be any on-path router or the receiver. After that, the receiver can determine whether the packets are part of a malicious flow. If so, it can send a shutoff request to ask the sender's accountability delegate to stop vouching for the packets of the sender or even pursue a longer term administrative or legal solution. Otherwise, the receiver can respond to the sender with the return address in the packet as the destination address.

APIP mainly focuses on the sender-flow unlinkability. In APIP, when a packet is required to contain a return address, the return address can be encrypted or otherwise masked.

H. ACCOUNTABLE AND PRIVATE NETWORK ARCHITECTURE

Accountable and Private Network Architecture (APNA) [44] is a new architecture that guarantees source accountability and privacy-preserving communication by enlisting ISPs as accountability agents and privacy brokers. In APNA, network communication is based on Ephemeral IDentifiers (EphIDs) rather than long-lived network addresses. EphIDs are cryptographically linked to the identity of a host. APNA also establishes shared secret keys based on EphIDs and encrypt all payload data by default. In short, EphIDs serve as both accountability units and privacy units. Because EphIDs are only identifiers that identify the hosts, they are not sufficient for routing packets to a destination. Therefore, APNA introduces the AS Identifier (AID) to indicate the location information. Consequently, a host can be reached by AID:EphID where the AID identifies the AS the host resides in, and the EphID identifies the host of the corresponding AS.

Figure 10 shows a high-level overview of APNA. Firstly, a host authenticates to the Registry Service (RS) of its AS and obtains bootstrapping information. Next, the host communicates with the Management Service (MS) of its AS to get the EphID. Then, the two communicating parties should know each other's AID:EphID identifiers, which can be achieved by the extension of DNS. The two hosts establish a shared symmetric secret key derived from public keys associated with the hosts' EphIDs to be used for network-layer data encryption. Finally, the two communicating parties use the shared secret symmetric key to encrypt every packet to achieve privacy-preserving communication.

The APNA packet header contains the source and destination AID:EphID tuples and a MAC of the packet's contents. A Border Router (BR) in the source AS can only allow packets originated from authenticated hosts and authorized EphIDs to leave the AS. The transit ASes just simply forward the packets to the next AS along the routing path.

VI. USER-TARGETED ACCOUNTABILITY

In this section, we present the protocols that aim to account for users. User-targeted accountability refers to the ability that holds users responsible for the packets they have sent using a device. Similarly, users must be identified uniquely. In fact, when a user subscribes to an ISP, the user will be assigned authentication credentials which generally contain a unique identifier used for user identification. Then, effective mechanisms should be used to ensure the authenticity of the packets that users send through devices, e.g., SAVI [46]. After that, association between users and their sent packets should be created. Currently, several methods can be used to achieve such a goal:

- The first method is to embed user identifiers into IP addresses. Currently, the length of IPv6 addresses allows for embedding extra information to meet specific requirements. Therefore, user identifiers and extra information can be used to generate IPv6 addresses, e.g., NIDTGA [54].
- The second one is to carry user identifiers in IP options or extension headers. IP protocols (i.e., IPv4 and IPv6) allow people to define extra options or extension headers to increase new functionalities and meet specific requirements. For example, TrueID [57] insert user identifiers into an extension header to associate users with their packets.

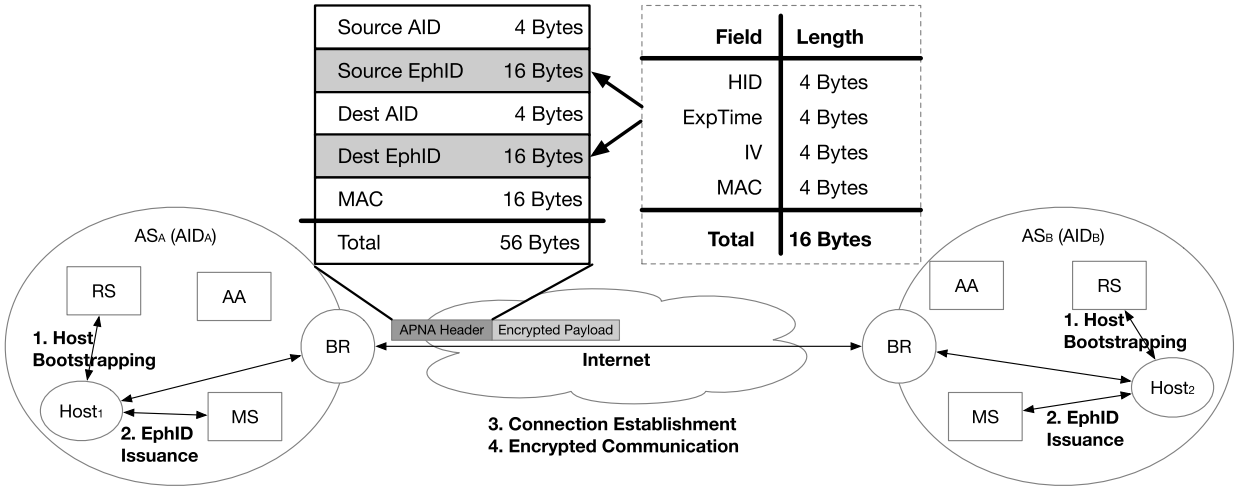

FIGURE 10. A high-level overview of APNA [44].

- The third options is to store binding logs of user identifiers and IP addresses [45]. Every time a user connects to the Internet, its ISP stores related information of this access, including user identifier, IP address, time, and MAC address.

A. AS-BASED ACCOUNTABILITY

Simon *et al.* [45] argue that accountability is the key to cost-effective handling of DDoS attacks on a network such as Internet. To solve network-layer DoS attacks, the solution should have a number of attractive features: 1) incremental deployability, 2) independence of infrastructure hardware, 3) backward compatibility. and 4) economic viability. Therefore, the authors propose an AS-based accountability (ABA) architecture as a cost-effective defense to filter unwanted traffic at the source, which uses a set of techniques that together satisfy the above criteria.

Simon *et al.* [45] define accountability in a network as containing two components: 1) identification, and 2) defensibility. The former means that the traffic sender can be identified by some persistent attribute. The latter means that receivers have the ability to prevent traffic from a source with a persistent attribute.

ABA uses five steps to create accountability among a group of ASes. The first step is to identify the customer and track the association between source IP address and customer by upgraded customer relationship management (CRM) systems or address assignment logs. The second step is to deploy strict ingress filtering over all customers in participating ISPs. The third step is to relay a target's request to filter traffic from a particular IP address using the filter request server (FRS) system. The fourth step is that a participating ISP marks the packets with the "evil bit" when the packets enter non-participating ISPs. The final step is to stop reflection attacks by keeping "evil bit state" in hosts.

B. NETWORK IDENTITY AND TIME GENERATED ADDRESS

Network Identity and Time Generated Address (NIDTGA) [54] is an IPv6 address generation algorithm embedded Network Identity and time information that can be used to design and implement an IPv6 address generation and traceback system. NIDTGA is based on Source Address Validation Architecture (SAVA) [81] which ensures the authenticity of the IPv6 source addresses. SAVA aims at constructing a secure environment for the compulsory source address validation at the access network, intra-domain and inter-domain levels.

NIDTGA mainly solves three key issues. Firstly, it designs a scalable structure of NID. To realize the principle of NID design, –namely hierarchy, scalability, confidentiality, flexibility, memorability, and usability, 40-bit NIDs can be divided into three parts, including Division Part (4 bits), Organization Part (m bits) and User Part ($36 - m$ bits). Totally, there exist 16 organization's sizes with the maximum of 2^{34} and the minimum of 2^4. User Part of NIDs can be generated from the current organization identities.

Secondly, it defines the IPv6 address generation algorithm using NID and time information. The interface identifier of an NIDTGA is generated from the encryption of a 40-bit NID and 24-bit time information using IDEA algorithm rather than embedding NID in the interface identifier of an IPv6 address directly, which protects the user's privacy.

Finally, it implements the whole system for the generation, allocation, management and traceback of IPv6 address with privacy protection. In the NIDTGA system, Address Generation Server uses IDEA algorithm to encrypt the concatenation of NID and time information and assigns the NIDTGA to the host. NID Management Server generates NIDs based on hosts' current organization identifiers and validates the login credentials. NID Traceback Server is used to trace back the user identity of an IPv6 address.

TABLE 3. Functionality comparison of all the network-layer accountability protocols.

Protocols	Category of Accountability			Identifier	Packet integrity protect mechanism	Packet-identity Association	Accountability services
	AS-targeted	Host-targeted	User-targeted				
Ingress filtering	√			Prefix	Route entry verification	IP prefix	Packet filtering
Passport	√			Prefix	AS-level signatures	IP prefix	Path authentication
IPA	√			ASN/prefix	Extended DNSSEC records	DNSSEC records	Prefix authentication
FAIR	√			Prefix	Transit ASes' forwarding signatures	IP prefix	Forwarding optimization
SAVI		√		IP address	Binding entry verification	IP address	Packet filtering
CGA		√		IPv6 address	Self-certifying	IPv6 address	Packet filtering
Persona		√		IP address/SN	TPM [80]	IP address	Packet filtering
BAFI		√		IP address	Source and AS-level signatures	IP address	Reputation system
AaaS		√		Self-defined ID	Source and AS-level signatures	Self-defined ID	Identity disclosure
AIP		√		Public key	Extended uRPF	Public key hash	Shutoff protocol
APIP		√		Public key	Packet signature	Public key hash	Shutoff protocol
APNA		√		HID	Packet signature	EphID	Shutoff protocol
ABA			√	Customer ID	Ingress filtering	IP address	Source filter request
NIDTGA			√	NID	Binding entry verification	NID embedded IPv6	Traceback
TrueID			√	UID	Binding entry verification	UID embedded extension header	Packet filtering

C. TrueID

TrueID [57] is a user accountability protocol which labels Internet actions with solid proofs through embedding packets with a sender's undeniable identity code.

SAVI devices are used to keep IP source address authentic within domains in TrueID. Once a user is self-authenticated, SuperFlow switch [82] will bind the user's private keys to other related information and embed the user's credible identity into the extension header of the packets. Also, SuperFlow switch can control flows with different granularities due to its support of SDN/Openflow architecture. At the same time, an SDN controller is used to distribute the flow control rule in the network. A public key exchange server (PKES) is used to store local users' keys and respond to public-key inquiries from local or allied domains.

In TrueID, when a sender sends packets to a receiver, SAVI devices will firstly filter spoofing packets. Then, SuperFlow switch embeds a TrueID header containing a provable signature into each packet. After the receiver receives the packets, it inquires the sender's public key from the local PKES and validates the signatures of the packets. Note that the local PKES may inquiry the source PKES for the public key of the sender for the first time.

VII. ANALYSIS AND DISCUSSION

In this section, we present a qualitative analysis of the accountability protocols that have been discussed throughout this paper. We mainly focus on the analysis of the accountability function, deployability, and security of all the accountability protocols.

A. ACCOUNTABILITY FUNCTION ANALYSIS

We summarize the accountability function analysis results in Table 3. The second column indicates the accountability type of each protocol. The third column indicates which identifier each protocol uses to identify entities. Note that ''IP address'' in the third column refers to both IPv4 address and IPv6 address. The finer-grained and more accurate identifier is, the more correct and effective the protocol can account for misbehaviors. The fourth column represents the packet validation mechanism that each protocol uses. To achieve accountability in the Internet, authentic packets must be ensured. If packets are bogus, the accountability results are incorrect and untrustworthy. The fifth column presents the packet-identity association method of each protocol. Some protocols use IP addresses to identify networks and hosts and associate their packets with them, e.g., Passport, CGA, SAVI, and Persona. Others use new identifiers such as public key, UID, and NID to identify hosts and users and embed such identifiers into addresses or an extension header to associate a host or user identity with their packets, e.g., AIP, APIP, APNA, NIDTGA, and TrueID. The final column shows the accountability services available in each protocol. Most of the protocols can provide packet filtering service. Some protocols can provide advanced services, such as shutoff and traceback.

B. DEPLOYABILITY ANALYSIS

Table 4 summarizes the analysis results of deployability. The second and third columns indicate whether each protocol supports deployment at the current IP version (IPv4 and IPv6). Some protocols such as CGA, NIDTGA, TrueID, and APNA only support either IPv6 or IPv4, while BAFI requires the modification of the current IP. The fourth column (Renumbering) indicates whether the accountability protocols require AS and host renumbering. AIP, APIP, and APNA require both AS and host renumbering. BAFI requires host renumbering, while IPA only requires AS renumbering. Renumbering will increase the difficulty to adopt the protocol in the current Internet, especially host renumbering. The fifth column (cost) presents the cost of deploying the protocol. Some protocols such as ingress filtering, SAVI and NIDTGA only require simple configuration, or upgrading and adding devices, which cost little. Some other protocols

TABLE 4. Deployability comparison of all the network-layer accountability protocols.

Protocols	IPv4	IPv6	Renumbering		Cost	Legacy network compatibility	Adoption incentive	Required infrastructure	Protocol modification
			AS	Host					
Ingress filtering	√	√			Low		√	None	None
Passport	√	√			High	√	√	Passport routers	Passport header
IPA	√	√	√		Low	√	√	Extended DNSSEC	IPA BGP update
FAIR	√	√			Normal	√	√	FAIR routers	FAIR header
SAVI	√	√			Low		√	SAVI devices	None
CGA		√			High	√		None	SEND [73]
Persona	√	√			Low		√	Persona routers	IP address replacement
BAFI				√	High	√		Reputation system	Identifier-location split
AaaS	√	√			High	√		Accountability services and Passport routers	Passport header
AIP	√	√	√	√	High			AIP routers and smart NICs	AIP header protocol, DNS
APIP	√	√	√	√	High			Accountability delegates and APIP verifiers	APIP header, DNS
APNA	√		√	√	High	√	√	APNA routers and accountability agents	APNA header, DNS
ABA	√	√			Low	√	√	FRSes	None
NIDTGA		√			Low		√	NIDTGA servers and SAVI devices	None
TrueID		√			High	√	√	Superflow switches and PKI system	TrueID header

TABLE 5. Security comparison of all the network-layer accountability protocols.

Protocols	Solutions				Privacy considerations			Security limitations
	Spoofing	DDoS	Prefix hijacking	Route forgery	Anonymity	Location untraceability	Data confidentiality	
Ingress filtering	√				N/A	N/A	N/A	* Installing these filters in the middle of the network is difficult.
Passport	√				N/A	N/A	N/A	* Diffie-Hellman key exchange is not secure.
IPA	√	√	√	√	N/A	N/A	N/A	* A trust root can become a target for attack.
FAIR		√			N/A	N/A	N/A	* Distributed malicious or compromised hosts in different ASes can still consume a target's resources within each AS's bandwidth limitation.
SAVI	√							* SAVI devices cannot resist DoS attacks.
CGA	√					√		* An attacker can create a new address from an arbitrary subnet prefix and anyone's public key.
Persona					√	√		* Persona does not enforce any anti-spoofing mechanism.
BAFI	√							* The owner of the root of the PKI could make it impossible for entire segments to provide accountability for their packets.
AaaS	√	√						* Diffie-Hellman key exchange is not secure.
AIP	√	√	√	√				* An attack can create many EIDs within limits.
APIP	√	√			√	√	√	* A malicious sender can omit briefing packets after being verified.
APNA	√	√			√	√	√	* A malicious application may use the EphID of a legal application to send packets.
ABA	√	√						* The evil bit can be modified by non-participating ISPs.
NIDTGA	√				√	√		* SAVI devices cannot resist DoS attacks.
TrueID	√							* SAVI devices cannot resist DoS attacks.

require the modification of hosts, network devices and basic infrastructure and cost much, such as AIP, BAFI, APIP and APNA. The sixth column indicates whether the protocol will work in the case of partial deployment. The seventh column shows the adoption incentive for each protocol. The results are obtained from comprehensively considering all the columns before. The last two columns present the infrastructures and protocol modification that are required for the current Internet. For example, SAVI requires the update of the network devices, while AIP requires the modification of IP header and adding smart NICs to hosts.

C. SECURITY ANALYSIS

The security analysis and comparison results of network-layer accountability protocols are summarized in Table 5. The second to fifth columns indicate what attacks each protocol solves. Most protocols solve source spoofing attack. Although most protocols can solve DDoS attacks based on source spoofing, we consider a protocol can solve DDoS attacks based on compromised hosts in the third "DDoS" column. Because Persona, APIP, and APNA consider a balance between accountability and privacy, they have better performance in privacy considerations. Some protocols may still have security limitations. For example, a malicious host can omit briefing packets after being verified in APIP. Under the circumstances, those packets cannot be accounted for due to the lack of packet fingerprints in the malicious host's accountability delegate.

VIII. CONCLUSION

This paper begins by providing an overview of accountability definitions. In fact, it is important to give a definition of accountability before studying its properties at the network layer of the Internet. Through the investigation, analysis, and comparison of related work, we consider that network-layer accountability is to determine whether an entity has sent a specific packet and provide countermeasures in the case of misbehaviors.

Then, we propose a framework and taxonomy of network-layer accountability. The accountability framework can be used as a guide to new network-layer accountability designs. Currently, we classify the state of the art according to accountability granularity. Actually, we can also explore other classifications of these protocols, e.g., according to the actions of entities in the packet forwarding process. The actions of an entity in the Internet can be classified into three categories:

1) to send packets to other entities,
2) to forward packets for other entities,
3) and to receive packets from other entities.

Therefore, we can also classify these protocols into three categories: source accountability, forwarding accountability, and destination accountability. Source accountability is to determine whether a source has sent a packet. The accountees of source accountability can be any entity that sends packets. Forwarding accountability is to determine whether an interconnected entity has correctly forwarded a packet. The accountee of forwarding accountability can be any entity that forwards packets, such as switches, routers, or even transit ASes. Destination accountability is to determine whether a destination has received a packet. Most of the network-layer accountability protocols belong to source accountability, while none of them are destination accountability protocols. We believe one reason for this situation is that among the above three types of actions, sending packets is the most important because there would be none of the other types of actions if initially there is no packet sent. Obviously, this kind of classification is not the best choice. But it does help us consider why there are no destination accountability protocols. In our context, destination accountability refers to account for whether a receiver has received specific packets. Then, destination accountability must at least ensure that a receiver cannot deny having received packets and cannot be accused of receiving packets that it did not receive. As the receiving actions are passive, it is difficult to account for these actions and define misbehaviors.

Finally, we conduct a comparative analysis of the accountability function, deployability, and security of these protocols. According to the analysis in Section VII-A, we find that all the protocols follow the framework we proposed in Section III-A. According to the analysis in Section VII-B, some protocols [15], [43], [44] are well-designed but require new communication identifiers and large-scale modifications to the deployed infrastructures (e.g., DNS), while some other protocols [14], [16], [51], [57] are costly.

The goal of this paper is to provide an overview of network-layer accountability research. Based on this survey of network-layer accountability protocols, the authors identified some issues that need to be addressed, for instance:

- Currently, many protocols are proposed to address the accountability issues of the Internet. Each protocol provides a method of building accountability into the Internet. All these methods differ with each other. A lack of a general framework for building accountability into the Internet makes it difficult to design new accountability protocols. In Section III-A, we summarize a simple but useful framework for achieving network-layer accountability in the Internet. This framework is not perfect enough and can still be improved through the formalization of four properties and the necessary verification.
- From the above analyses, we find that each protocol has different goals, such as fine-grained accountability (i.e., host- or user-targeted accountability), modest cost, and good deployability. However, how to design a protocol that simultaneously achieves fine-grained accountability, modest costs, and good deployability is still a big problem. To ensure fine-grained accountability, host or user identifiers should be used and managed. To achieve good deployability, new designed protocols should be incrementally deployable and attract early deployment. We believe that using currently deployed infrastructures or technologies with a few minor modifications will have lower deployment costs than rolling out new infrastructures. During the design process, we should also adopt lightweight operations to achieve modest costs.
- Although many protocols focus on the establishment of accountability in the Internet, little attention is paid to the quantification of accountability. As far as we know, Xiao *et al.* [41] proposed a hierarchy model to quantify the accountability of a system. Actually, we can consider the quantification of accountability from multiple levels and different requirements of accountors. We believe that network-layer accountability is achieved when each packet is accountable. Therefore, we can quantify network-layer accountability by defining and calculating the accountability of each packet according to the accountability framework mentioned in Section III-A.
- Accountability and privacy are both considered valuable properties, but they appear to conflict. Currently, APIP [43] and APNA [44] are the main proposals that try to balance accountability and privacy. But they both require new communication identifiers and large scale of modifications to fully deployed Internet infrastructures and protocols. How to design new protocols that avoid these modifications and use resources and protocols available remains unresolved.

All of these questions remain open and any answers would greatly improve network-layer accountability of the Internet.

ACKNOWLEDGMENT

The authors thank Shaomin Lu and anonymous reviewers for their valuable feedback.

REFERENCES

[1] (2017). *Akamai's [State of the Internet]/Security, Q1 2017 Report*. [Online]. Available: https://bit.ly/2sDn91x
[2] (2017). *Akamai's [State of the Internet]/Security, Q2 2017 Report*. [Online]. Available: https://bit.ly/2vnTtml
[3] (2017). *Akamai's [State of the Internet]/Security, Q3 2017 Report*. [Online]. Available: https://bit.ly/2LLQf9D

[4] (2014). *Incapsula Finds DDOS Attacks Cost Businesses an Average of $500, 000*. [Online]. Available: https://www.incapsula.com/about/press-releases/incapsula-finds-ddos-attacks-cost-businesses-an-average-of-500,000usd/

[5] (2016). *Mirai Attack on DYN Internet Infrastructure*. [Online]. Available: https://coar.risc.anl.gov/mirai-attack-dyn-internet-infrastructure/

[6] (2010). *Massive Denial of Service Attack Severs Myanmar from Internet*. [Online]. Available: https://threatpost.com/massive-denial-service-attack-severs-myanmar-internet-110310/74638/

[7] (2017). *Google Hijack Made Japan 'Land of no Internet' for More than 30 Minutes*. [Online]. Available: http://www.thedrum.com/news/2017/08/28/google-hijack-made-japan-land-no-internet-more-30-minutes

[8] D. G. Blogs. (2008). *Pakistan Hijacks Youtube*. [Online]. Available: http://dyn.com/blog/pakistan-hijacks-youtube-1/

[9] R. Schaeffer, "National information assurance (IA) glossary," Committee on National Security Systems, NSA, Ft. Meade, MD, USA, Tech. Rep. 4009, 2010.

[10] *Information Technology–Security Techniques–Information Security Management Systems–Overview and Vocabulary*, document JTC1/SC27, 2016.

[11] I. S. Audit and C. Association. (2017). *Glossary of Terms*. [Online]. Available: http://www.isaca.org/Pages/Glossary.aspx

[12] Y. Cherdantseva and J. Hilton, "Information security and information assurance. The discussion about the meaning, scope and goals," in *Organizational, Legal, and Technological Dimensions of Information System Administration*. Hershey, PA, USA: IGI Global, 2013, pp. 167–198.

[13] G. W. Jones, *The Search for Local Accountability, Strengthening Local Government 1990s*, S. Leach Ed. White Plains, NY, USA: Longman, 1992, pp. 49–78.

[14] A. Bender, N. Spring, D. Levin, and B. Bhattacharjee, "Accountability as a service," in *Proc. SRUTI*, vol. 7, 2007, pp. 1–6.

[15] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, "Accountable Internet protocol (AIP)," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 339–350, 2008.

[16] J. Mirkovic and P. Reiher, "Building accountability into the future Internet," in *Proc. 4th Workshop Secure Netw. Protocols*, Oct. 2008, pp. 45–51.

[17] X. Yang and X. Liu, "Internet protocol made accountable," in *Proc. HotNets*, 2009, pp. 1–6.

[18] A. Li, X. Liu, and X. Yang, "Bootstrapping accountability in the internet we have," in *Proc. 8th USENIX Conf. Netw. Syst. Design Implement. (NSDI)*, 2011, pp. 155–168.

[19] K. Argyraki, P. Maniatis, O. Irzak, S. Ashish, and S. Shenker, "Loss and delay accountability for the Internet," in *Proc. IEEE Int. Conf. Netw. Protocols*, Oct. 2007, pp. 194–205.

[20] B. E. Ujcich, A. Miller, A. Bates, and W. H. Sanders, "Towards an accountable software-defined networking architecture," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2017, pp. 1–5.

[21] T. Sasaki, C. Pappas, T. Lee, T. Hoefler, and A. Perrig, "SDNsec: Forwarding accountability for the SDN data plane," in *Proc. 25th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2016, pp. 1–10.

[22] P. Veríssimo and L. Rodrigues, "Fundamental security concepts," in *Distributed Systems for System Architects*. Boston, MA, USA: Springer, 2001, pp. 377–393.

[23] A. Haeberlen, P. Kouznetsov, and P. Druschel, "PeerReview: Practical accountability for distributed systems," in *Proc. ACM SIGOPS Symp. Operating Syst. Princ.*, 2007, pp. 175–188.

[24] A. R. Yumerefendi and J. S. Chase, "Strong accountability for network storage," *ACM Trans. Storage*, vol. 3, no. 3, 2007, Art. no. 11.

[25] A. Haeberlen, P. Aditya, R. Rodrigues, and P. Druschel, "Accountable virtual machines," in *Proc. 9th USENIX Conf. Operating Syst. Design Implement. (OSDI)*, Vancouver, BC, Canada, Oct. 2010, pp. 119–134.

[26] V. Sekar and P. Maniatis, "Verifiable resource accounting for cloud computing services," in *Proc. 3rd ACM Workshop Cloud Comput. Secur. Workshop*, 2011, pp. 21–26.

[27] T. Baker, A. Taleb-Bendiab, and M. Randles, "Auditable intention-oriented Web applications using PAA auditing/accounting paradigm," in *Proc. Conf. Techn. Appl. Mobile Commerce (TAMoCo)*, 2009, pp. 61–70.

[28] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, "Tussle in cyberspace: Defining tomorrow's Internet," *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 462–475, Jun. 2005.

[29] A. S. Tanenbaum and D. Wetherall, "Computer networks," *Home*, vol. 54, no. 7, pp. 1169–1182, 1996.

[30] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, vol. 36, no. 4, 3rd ed. Amsterdam, The Netherlands: Elsevier, 2003, pp. 169–305.

[31] D. Clark, "The design philosophy of the DARPA Internet protocols," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 18, no. 4, pp. 106–114, 1988.

[32] CAIDA. (2018). *State of IP Spoofing*. [Online]. Available: https://spoofer.caida.org/summary.php

[33] R. Droms, *Dynamic Host Configuration Protocol*, document RFC 2131, Mar. 1997.

[34] J. Bound, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, document RFC 3315, Jul. 2003. [Online]. Available: https://rfc-editor.org/rfc/rfc3315.txt

[35] D. T. Narten, T. Jinmei, and D. S. Thomson, *IPv6 Stateless Address Autoconfiguration*, document RFC 4862, Sep. 2007.

[36] K. B. Egevang and P. Srisuresh, *Traditional IP Network Address Translator (Traditional NAT)*, document RFC 3022, Jan. 2001.

[37] Z. Xiao, N. Kathiresshan, and Y. Xiao, "A survey of accountability in computer networks and distributed systems," *Secur. Commun. Netw.*, vol. 9, no. 4, pp. 290–315, 2016.

[38] A. Schedler *et al.*, "Conceptualizing accountability," in *The Self-Restraining State: Power Accountability New Democracies*, vol. 13. 1999, p. 17.

[39] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *J. ACM*, vol. 27, no. 2, pp. 228–234, 1980.

[40] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[41] Z. Xiao, Y. Xiao, and J. Wu, "A quantitative study of accountability in wireless multi-hop networks," in *Proc. Int. Conf. Parallel Process. (ICPP)*, Sep. 2010, pp. 198–207.

[42] Y. Mallios, S. Modi, A. Agarwala, and C. Johns, "Persona: Network layer anonymity and accountability for next generation Internet," in *Proc. IFIP Int. Inf. Security Conf.* Berlin, Germany: Springer, 2009, pp. 410–420.

[43] D. Naylor, M. K. Mukerjee, and P. Steenkiste, "Balancing accountability and privacy in the network," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 75–86, 2015.

[44] T. Lee, C. Pappas, D. Barrera, P. Szalachowski, and A. Perrig, "Source accountability with domain-brokered privacy," in *Proc. 12th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Irvine, CA, USA, Dec. 2016, pp. 345–358.

[45] D. R. Simon, S. Agarwal, and D. A. Maltz, "AS-based accountability as a cost-effective DDoS defense," in *Proc. HotBots*, vol. 7, 2007, p. 9.

[46] J. Wu, J. Bi, M. Bagnulo, F. Baker, and C. Vogt, *Source Address Validation Improvement (SAVI) Framework*, document RFC 7039, Oct. 2013.

[47] P. Ferguson and D. Senie, *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*, document RFC 2827, May 2000. [Online]. Available: https://rfc-editor.org/rfc/rfc2827.txt

[48] Z. Al-Qudah, E. Johnson, M. Rabinovich, and O. Spatscheck, "Internet with transient destination-controlled addressing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 731–744, Apr. 2016.

[49] K. Seo and S. Kent, *Security Architecture for the Internet Protocol*, document RFC 4301, Dec. 2005. [Online]. Available: https://rfc-editor.org/rfc/rfc4301.txt

[50] X. Liu, A. Li, X. Yang, and D. Wetherall, "Passport: Secure and adoptable source authentication," in *Proc. NSDI*, vol. 8, 2008, pp. 365–378.

[51] T. Aura, *Cryptographically Generated Addresses (CGA)*, document RFC 3972, Mar. 2005. [Online]. Available: https://rfc-editor.org/rfc/rfc3972.txt

[52] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang, "SAVE: Source address validity enforcement protocol," in *Proc. 21st Annu. Joint Conf., IEEE Comput. Commun. Soc. INFOCOM*, vol. 3, Jun. 2002, pp. 1557–1566.

[53] C. A. Shue, M. Gupta, and M. P. Davy, "Packet forwarding with source verification," *Comput. Netw.*, vol. 52, no. 8, pp. 1567–1582, 2008.

[54] Y. Liu, G. Ren, J. Wu, S. Zhang, L. He, and Y. Jia, "Building an IPV6 address generation and traceback system with NIDTGA in address driven network," *Sci. China Inf. Sci.*, vol. 58, no. 12, pp. 1–14, 2015.

[55] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in *Proc. 10th Int. Conf. Inf. Knowl. Manage.*, 2001, pp. 310–317.

[56] C. Pappas, R. M. Reischuk, and A. Perrig, "Fair: Forwarding accountability for Internet reputability," in *Proc. IEEE 23rd Int. Conf. Netw. Protocols (ICNP)*, Nov. 2015, pp. 189–200.

[57] G. Hu, W. Chen, Q. Li, Y. Jiang, and K. Xu, "TrueID: A practical solution to enhance Internet accountability by assigning packets with creditable user identity code," *Future Gener. Comput. Syst.*, vol. 72, pp. 219–226, Jul. 2017.

[58] J. Mirkovic and E. Kissel, "Comparative evaluation of spoofing defenses," *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 2, pp. 218–232, Mar. 2011.

[59] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, Apr. 2004.
[60] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (S-BGP)," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 582–592, Apr. 2000.
[61] R. White, "Securing BGP through secure origin BGP," *Int. Protocol J.*, vol. 6, no. 3, pp. 15–22, 2003.
[62] T. Wan, E. Kranakis, and P. C. van Oorschot, "Pretty secure BGP, PSBGP," in *Proc. NDSS*, 2005, pp. 1–23.
[63] Y.-C. Hu, A. Perrig, and M. Sirbu, "SPV: Secure path vector routing for securing BGP," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 179–192, 2004.
[64] W. Aiello, J. Ioannidis, and P. McDaniel, "Origin authentication in interdomain routing," in *Proc. 10th ACM Conf. Comput. Commun. Security*, 2003, pp. 165–178.
[65] R. Bush and R. Austein, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*, document RFC 6810, 2013.
[66] G. Huston, M. Rossi, and G. Armitage, "Securing BGP—A literature survey," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 199–222, 2nd Quart., 2011.
[67] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford, "A Survey of BGP Security Issues and Solutions," *Proc. IEEE*, vol. 98, no. 1, pp. 100–122, Jan. 2010.
[68] M. O. Nicholes and B. Mukherjee, "A survey of security techniques for the border gateway protocol (BGP)," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 52–65, 1st Quart., 2009.
[69] F. Baker and P. Savola, *Ingress Filtering for Multihomed Networks*, document RFC 3704, Mar. 2004. [Online]. Available: https://rfc-editor.org/rfc/rfc3704.txt
[70] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.
[71] X. Liu, X. Yang, and Y. Xia, "NetFence: Preventing Internet denial of service from inside out," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 255–266, 2010.
[72] Cisco, IOS, "Quality of service solutions configuration guide, policing and shaping overview," Cisco, San Jose, CA, USA, Tech. Rep. QC-203-QC-218, May 2003.
[73] J. Kempf, J. Arkko, B. Zill, and P. Nikander, *SEcure Neighbor Discovery (SEND)*, document RFC 3971, Mar. 2005. [Online]. Available: https://rfc-editor.org/rfc/rfc3971.txt
[74] E. Nordmark, M. Bagnulo, and E. Levy-Abegnoli, *FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses*, document RFC 6620, May 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6620.txt
[75] J. Bi, J. Wu, G. Yao, and F. Baker, *Source Address Validation Improvement (SAVI) Solution for DHCP*, document RFC 7513, May 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7513.txt
[76] M. Bagnulo and A. Garcia-Martinez, *SEcure Neighbor Discovery (SEND) Source Address Validation Improvement (SAVI)*, document RFC 7219, May 2014. [Online]. Available: https://rfc-editor.org/rfc/rfc7219.txt
[77] J. Bi, G. Yao, J. M. Halpern, and E. Levy-Abegnoli, *Source Address Validation Improvement (SAVI) for Mixed Address Assignment Methods Scenario*, document RFC 8074, Feb. 2017. [Online]. Available: https://rfc-editor.org/rfc/rfc8074.txt
[78] D. T. Narten, R. P. Draves, and S. Krishnan, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, document RFC 4941, Sep. 2007. [Online]. Available: https://rfc-editor.org/rfc/rfc4941.txt
[79] J. Arkko and M. Bagnulo, *Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs)*, document RFC 4982, Jul. 2007. [Online]. Available: https://rfc-editor.org/rfc/rfc4982.txt
[80] J. M. McCune, B. Parno, A. Perrig, M. K. Reiter, and A. Seshadri, "Minimal TCB code execution," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 267–272.
[81] J. Wu, J. Bi, X. Li, G. Ren, M. Williams, and K. Xu, *A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience*, document RFC 5210, Jun. 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5210.txt
[82] G. Hu, K. Xu, and J. Wu, "SuperFlow: A reliable, controllable and scalable architecture for large-scale enterprise networks," in *Proc. IEEE 10th Int. Conf. High Perform. Comput. Commun., IEEE Int. Conf. Embedded Ubiquitous Comput. (HPCC_EUC)*, Nov. 2013, pp. 1195–1202.

LIN HE received the bachelor's degree from the Beijing University of Posts and Telecommunications. He is currently pursuing the Ph.D. degree with the Institute for Network Sciences and Cyberspace, Tsinghua University. His major research interests include network architecture and protocol design.

YING LIU received the M.S. degree in computer science and the Ph.D. degree in applied mathematics from Xidian University, China, in 1998 and 2001, respectively. She is currently an Associate Professor with Tsinghua University, China. Her major research interests include network architecture design, next-generation Internet architecture, routing algorithm, and protocol.

GANG REN received the Ph.D. degree in computer system architecture from Tsinghua University, China, in 2009. He is currently an Assistant Professor with Tsinghua University. His major research interests include network architecture design, next-generation Internet architecture, and network security.