gPHI: Lightweight Anonymity Protocol for Anonymity at Host and AS Levels

Yutaro Yoshinaka*, Junji Takemasa*, Yuki Koizumi* and Toru Hasegawa*
*Graduate School of Information Science and Technology, Osaka University

***Abstract*—Lightweight anonymity protocols are expected to provide anonymity service at the networking layer. Among them, PHI and its successor dPHI are promising due to the fact that anonymous path selection according to IP routing enables easy deployment at the Internet infrastructure. However, the path selection based on IP routing incurs vulnerabilities against attacks which leverage underlying Internet topology information. This paper discovers such new attacks against PHI/dPHI and designs gPHI to mitigate such attacks by extending dPHI.**

***Index Terms*—Privacy, Anonymity Protocol, Relationship Anonymity, Internet**

I. Introduction

Global-scale pervasive surveillance can be performed anywhere on the Internet [1] by adversaries, such as state-level intelligence agencies that throttle traffic to and from designated sites and giant corporations that track, monitor, and analyze users' online activities [2]. Anonymity protocols are key countermeasures that unlink source node identities such as source IP addresses and content identities such as destination IP addresses. Currently, Tor [3], an application layer anonymity protocol, is being used to achieve sender anonymity. However, it suffers from two problems: low throughput and high latency. These problems arise because global adversaries can eavesdrop from multiple points on the Internet.

Lightweight anonymity protocols in the network layer provide a promising solution to these problems [4]–[8]. A key difference between lightweight anonymity protocols and Tor is that they assume weaker but more realistic adversaries. Some autonomous systems (ASes) are assumed to be local adversaries. This assumption enables the implementation of lightweight anonymity protocols on routers rather than on servers, like Tor's implementation. The merits of such protocols are summarized as follows: Anonymized end-to-end paths need not traverse many ASes like Tor, since the number of malicious ASes is limited. In addition, packet payloads need not be repeatedly encrypted like Tor, because local adversaries are not able to intercept packets at multiple ASes. Hereafter, we call lightweight anonymity protocols just *anonymity protocols*.

Among the existing anonymity protocols, PHI [7] and dPHI [8] are promising because they use IP routing to forward *path-setup packets* for setting up anonymized end-to-end paths in the *path-setup phase*. This enables an incremental deployment of the anonymity protocol over the Internet by avoiding the design of a complicated secure routing. However, a special node called a *helper* is introduced to guarantee relationship anonymity. The helper divides an end-to-end path into two sub-paths. The first sub-path is between the source node and the router called a *midway router* and the second sub-path is between the midway router and the destination node. In the *data-transmission phase*, *data packets* are forwarded according to their encrypted forwarding information. This prevents routers on the first sub-path from knowing the IP addresses of the source or destination nodes and prevents those on the second sub-path from knowing the IP addresses of the source nodes. Hence, PHI/dPHI provide relationship anonymity by ensuring that no router on the two sub-paths knows the IP addresses of both the source node and the destination node.

However, the use of IP routing gives rise to the two types of vulnerabilities in relationship anonymity. The first vulnerability is that the helpers know destination nodes' IP addresses. Since the helpers' IP addresses are open, they can be compromised. Moreover, existing studies have assumed that helpers do not collude with malicious ASes [7], [8]. If the helper and the *source AS* which accommodates the source node are compromised by the adversary, it can link the IP addresses of the source and destination nodes by observing the encrypted forwarding information of received packet headers. This obviously breaks the relationship anonymity. The second vulnerability is that the midway router, which knows the destination node's IP address, can infer the source AS by performing a *topological attack* that leverages the underlying Internet topology. The malicious midway router can infer the source AS by calculating a candidate set of ASes that it can reach according to the IP routing. In addition, it can narrow down the candidate set by measuring the round-trip times (RTTs) between itself and the source node [9]. In the worst-case scenario, the adversary can correctly infer the source AS and break the relationship anonymity at the AS level.

To patch these vulnerabilities, this study designs guard-PHI (gPHI) to guarantee relationship anonymity at both the host and the AS levels. The path-setup procedure of PHI/dPHI is extended because these vulnerabilities arise from a helper and midway router knowing critical information about path-setup packets.

The contributions of this study are summarized below: First, we precisely define relationship anonymity at both the

This work has been supported by JSPS KAKENHI Grant Number 21H03442.

ISBN 978-3-903176-48-5

host and the AS levels, whereas the existing studies do not explicitly do so. Second, we introduce another server called a *guard* to prevent helpers from linking a destination node's IP address and forwarding packet information. This guarantees relationship anonymity at the host level. Third, we clarify the conditions under which relationship anonymity at the AS level is broken and empirically evaluate the resilience of gPHI against topological attacks that leverage the underlying Internet topology information.

The remainder of this paper is structured as follows: Section II describes the vulnerabilities of PHI/dPHI. In Section III, gPHI is designed to guarantee relationship anonymity at the host and the AS levels. Section IV evaluates the resilience of gPHI against attacks that break the relationship anonymity at the AS level. Finally, Section V summarizes the related work and Section VI concludes the paper.

II. Vulnerabilities of Anonymity Protocol

A. Concept of PHI/dPHI

PHI/dPHI adopts the following principles to achieve a faster forwarding and shorter latency than Tor.

a) In-network Implementation: PHI/dPHI is deployed in the network layer, that is, it is implemented on routers rather than on servers, as seen in Tor. The implementation of the protocol in the network layer shortens end-to-end latency and decreases the probability of routers being compromised. This is because routers are not operated by volunteers but by network operators.

b) Source-based Routing and Forwarding Information Encryption: PHI/dPHI adopts source-based routing. In the *path-setup phase*, a source node uses *path-setup packets* to set up an end-to-end path to a destination node. A list of *forwarding states* is created by the routers on the path. Each forwarding state specifies the next hop and is encrypted with the router's secret key. At the end of the path-setup phase, the list is sent back to the source node. In the *data-transmission phase*, the source node records the list in the packet headers. Packet headers do not have an IP address in plaintext in this phase, and each router forwards a packet according to its forwarding state. An advantage of this encryption scheme is that a key exchange protocol need not be used.

c) Threat Model: PHI/dPHI assumes that a maximum of one AS on each end-to-end path is malicious. Since such a local adversary cannot intercept the same packet at more than two ASes, only the packet headers are encrypted and not the entire packets, as in the case of Tor. Note that all the routers in the malicious AS decrypt the encrypted packet headers.

B. Anonymity Definition

Relationship anonymity refers to the unlinkability between source and destination nodes [10]. This study explicitly defines two types of relationship anonymity: *host-level relationship anonymity* and *AS-level relationship anonymity*. In contrast, the existing studies do not distinguish them.

Definition II.1. *A protocol achieves host-level relationship anonymity if and only if it guarantees the unlinkability between source and destination addresses.*

Definition II.2. *A protocol achieves AS-level relationship anonymity if and only if it guarantees the unlinkability between source and destination ASes.*

The aim of this study is to guarantee both the types of relationship anonymities because only host-level anonymity is not sufficient to protect user privacy. In some cases, a compromise in the AS-level relationship anonymity may result in a compromise in the host-level relationship anonymity. If adversaries successfully identify a source AS, they can narrow down the candidates of source IP addresses connected to it. In addition, the leakage of a user's AS identity, which is strongly related to nationality, region, affiliation, and so on, can seriously affect privacy. Existing studies [7], [8] show that if some information pertaining to an end-to-end path, such as a hop count, is leaked to a compromised AS, it can infer the source AS. However, they do not precisely define AS-level relationship anonymity.

C. Path Setup Procedure of PHI/dPHI

This subsection describes the path-setup procedure to guarantee host-level relationship anonymity.

PHI/dPHI uses IP routing to forward path-setup packets, that is, the packets' destination IP addresses are used by routers to determine the subsequent hops. This enables an incremental deployment of the protocol over the Internet. Since a source node cannot send a path-setup packet directly to a destination node, PHI/dPHI introduces a special server called a *helper*, as illustrated in Fig. 1. The path-setup procedure consists of two rounds. First, the source node sends the path-setup request packet to the helper to set up the first sub-path from the source node to the helper. V^1 represents the forwarding state list of this sub-path. When the path-setup packet is sent back to the source node, the *midway router*, which works as an exchange point for the first and second sub-paths, is chosen. In the second round, the source node sends the path-setup request packet to the midway router, which sets up the second sub-path to the destination node. V^2 is the forwarding state list of the second sub-path. In the data-transmission phase, V^1 and V^2 are recorded in the packet headers, and each router decrypts the corresponding forwarding states in V^1 and V^2.

Host-level relationship anonymity can be broken if a router or helper linked forwarding state lists the V^1 and V^2 of data packets in the data-transmission phase and source and destination nodes' IP addresses IP_S and IP_D in the path-setup request/reply packets. However, neither the router nor helper links IP_S and IP_D. Routers on the first sub-path do not know the destination node's IP address IP_D because the source node encrypts IP_D using the helper's public key. Routers on the second sub-path do not know the source node's IP address IP_S because the path-setup request packets in the second round are forwarded to the midway router according to V^1. The helper

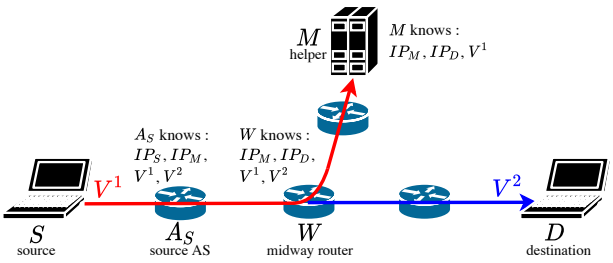


Fig. 1. dPHI path consisting of two forwarding state lists

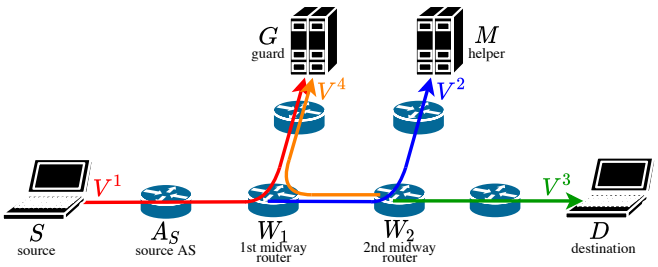


Fig. 2. gPHI path consisting of four forwarding state lists

knows IP_D, but does not observe data packets in the data-transmission phase.

D. Vulnerabilities Caused by IP Routing

PHI/dPHI has the two vulnerabilities because of the adoption of IP routing. The first vulnerability is that helpers and midway routers know the destination node's IP address. This vulnerability can be a threat to host-level relationship anonymity if a malicious source AS compromises the helper or if the midway router belongs to this malicious AS. The second vulnerability is that adversaries can perform a *topological attack* to infer candidates for source and destination ASes. Such an attack can be a serious threat to AS-level relationship anonymity.

1) Attacks for Breaking Host-level Relationship Anonymity: Two types of attacks break host-level relationship anonymity by leveraging the first vulnerability.

a) Source AS and Helper Colluding Attack: PHI/dPHI assumes that a malicious AS does not compromise a helper. However, this assumption is unrealistic because the helper is just a server whose IP address is open. If a malicious source AS compromises the helper, it can link IP_S and IP_D in the following manner. This malicious AS knows IP_S when it receives a path-setup request from the source node. The helper lets this AS know the pair of IP_D and forwarding list V^1. Consequently, the malicious AS correlates (links) IP_S with IP_D using V^1.

b) Midway Attack: Midway attack is performed by a midway router that knows IP_D. If any router of the source AS accidentally becomes a midway router, it can link IP_S and IP_D. Existing studies discuss this attack [7], [8] but do not present a specific solution to prevent it.

2) Attack for Breaking AS-level Relationship Anonymity: Topological attack with RTT measurement is performed by a malicious AS that knows the underlying Internet topology and the routing policies of ASes. This attack involves the two steps: a topological attack and an attack with RTT measurement.

In the first step, the malicious AS performs a topological attack to infer the *AS-level relationship anonymity set*. This set is a direct product of the *AS-level source anonymity set* and the *AS-level destination anonymity set*. AS-level source anonymity set is calculated as a set of ASes to which the hosts in the host-level source anonymity set (defined as sender anonymity set in [10]) belong. AS-level destination anonymity set is similarly calculated from a host-level destination anonymity set (recipient anonymity set in [10]). In the case of PHI/dPHI, the midway router is the most powerful entity because it knows a destination node's IP address IP_D and is closest to the source node, while other routers on the sub-path between the midway router and destination node also know IP_D. Thus, the rest of this subsection discusses how a malicious midway router calculates its AS-level source anonymity set. Here, let A_{NW} be the AS from which the AS of the midway router A_W receives the path-setup request. Further, let A_i be an AS in the AS-level source anonymity set of A_W. AS A_i in the AS-level source anonymity set of A_W satisfies the following condition: there exists an IP routing path from A_i to A_W, and it traverses A_{NW}. Note that A_W is the next-hop AS to A_{NW}.

In the second step, the midway router performs an attack with RTT measurement [9], which is briefly mentioned by Chen et al. [7]. In advance, the midway router measures the RTT values of the sub-paths between itself and all ASes in the AS-level source anonymity set. When the victim source node communicates with the destination node, the midway router intercepts all data packets transmitted on the sub-path between itself and the victim and records their RTT values. It then chooses a sub-path whose RTT distribution is the most similar to that of the recorded RTT values. This AS can be inferred as the source AS.

III. gPHI: Anonymity Protocol for Relationship Anonymity at Host and AS Levels

This section presents the design of gPHI using the notations summarized in Table I.

A. Design Rationale

The goals of gPHI are to guarantee host-level relationship anonymity against the source AS and helper colluding attacks and midway attacks, and increase the resilience of AS-level relationship anonymity against topological attacks. The primary idea behind guaranteeing host-level relationship anonymity is preventing the helper from knowing any sub-path's forwarding state list that is used in the data-transmission phase. To realize this, gPHI introduces another server called a *guard* G, as illustrated in Fig. 2. Table II summarizes the *items of interest (IOIs)* that each node knows. We can infer that the source AS and the helper do not know the common sub-path's forwarding list. Hence, even if the malicious source AS A_S compromises

the helper M, it cannot link IP_D and IP_S because there is no common forwarding list in the IOIs of A_S and M.

The introduction of a guard divides an end-to-end path into three sub-paths whose respective forwarding lists are V^1, V^4, and V^3. The sub-path V^2 is used for the communication between the guard G and helper M. Note that two midway routers W_1 and W_2 are used. The setup of all sub-paths and end-to-end paths is described in Section III-D. This path division prevents midway attacks. Because W_1 does not know IP_D, only W_2 is a candidate for performing the midway attack. Here, the two midway routers belong to different ASes, and thus, midway router W_2 does not belong to the source AS, whereas midway router W_1 belongs to the source AS with some probability. In other words, this prevents a midway attack.

The introduction of a guard has two merits in terms of increasing the resilience of AS-level relationship anonymity against topological attacks with RTT measurement. The first is that longer end-to-end paths implicitly increase the size of both AS-level source and destination anonymity sets. The tradeoff between the resilience of AS-level relationship anonymity and an increase in the path stretch is experimentally evaluated in Section IV.

The second merit is that the division of the end-to-end path into three sub-paths makes it difficult for the midway routers W_1 and W_2 to perform an attack with RTT measurements. The reason the source AS is difficult to infer is described as follows. Let us consider that W_2 tries to infer the source AS from the recorded RTT values on the sub-path between W_2 and the source node. Since the candidates for the source ASes in the AS-level source anonymity set use the same guard G and helper M, the paths from W_2 to the candidate source ASes traverse (share) the same sequence of routers, that is, they have a common sub-path. In the worst-case scenario, the common sub-path becomes the sub-path between W_1 and W_2. Hence, we conclude that such a common sub-path makes it difficult to correctly infer the source AS with RTT measurements, according to Hopper et al.'s study [9]. They demonstrated that it is difficult to distinguish paths that have a common sub-path by measuring RTT values [9]. Similarly, the destination AS is difficult to infer by W_1.

B. Threat Model and Goals

1) Threat Model: gPHI assumes that a maximum of one AS on each end-to-end path is malicious. The nodes compromised by the malicious AS are summarized as follows:

- Either a helper or a guard is not compromised.
- Either a helper or a guard is compromised.
- Any number of end hosts which are not on the end-to-end path are compromised.

The malicious AS decrypts all encrypted information using its secret keys and knows everything about the underlying network topology, such as the routing policies of all ASes.

2) Goals: The goals of gPHI are to guarantee host-level and AS-level relationship anonymities to the maximum extent. The cases in which AS-level relationship anonymity can be compromised are summarized below: In the first case, an AS between the first midway router and guard G is malicious and it compromises M. Here, the AS knows sid' and the compromised helper M knows sid' and IP_D. Because this AS knows IP_D, it can perform a topological attack to infer the source AS. If the AS-level source anonymity set of this AS is one, the source AS can be correctly identified. The probability of this case depends on the probability of each AS being malicious, and thus, this study does not evaluate it. In the second case, the first and third sub-paths share the same AS. The details of this case are discussed in Section III-E.

TABLE I
NOTATION IN PROTOCOL DESCRIPTION

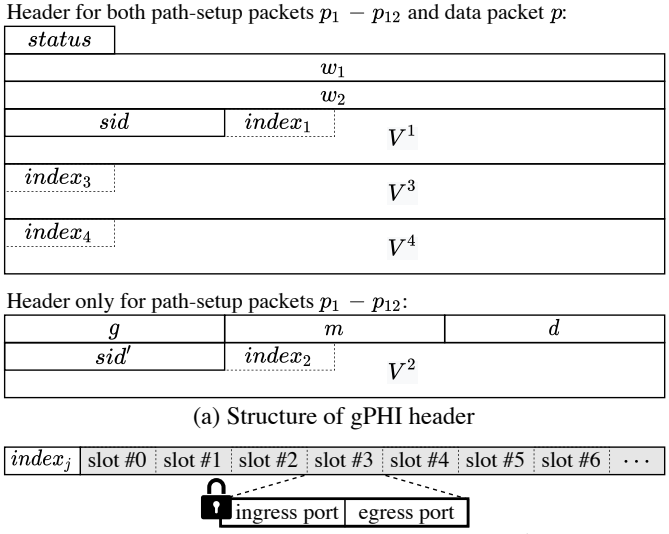
Notation	Definition
On-path node	
S, D, G, M	Source, destination, guard, and helper.
A_S, A_D, A_G, A_M	AS of source, destination, guard, and helper.
W_1, W_2	The first and second midway router (AS).
Value in packet fields	
ϕ	Random bits.
$\{X\}_N$	Ciphertext of plaintext X encrypted by node N's public key.
IP_S, IP_D, IP_G, IP_M	IP addresses of S, D, G, M.
$status$	Packet type (p_1, ..., p_{12}).
sid, sid'	Two session identifiers.
$V^1, ..., V^4$	Forwarding state lists.
w_1, w_2	State kept by midway router W_1 and W_2.
g, m, d	Identifiers of G, M, D. For example, d represents the value IP_D in plaintext, $\{IP_D\}_N$ in ciphertext, or ϕ, depending on the case.
$payload$	Payload encrypted in an end-to-end manner.
Packet	
$p_i(sid_1; V^j; X)$	i-th packet with session ID sid_1 which has filed X, and is forwarded according to V^j.
$p_i(sid_1; V^j, IP_d; X)$	i-th packet with session ID sid_1 which has field X, it is forwarded according to destination address IP_d, and creates V^j.
$p_i(sid_1, sid_2; V^j, V^k; X)$	i-th packet which has field X, is forwarded according to V^k with session id sid_2, creates V^j with session id sid_1.
Path	
P_{X-Y}	The path between node X and Y

TABLE II
ITEMS OF INTEREST (IOIS) KNOWN BY NODES ON GPHI PATHS

node	IP addess	other IOIs
A_S	IP_S	sid, IP_G, V^1, V^3, V^4
W_1	-	$sid, sid', IP_G, IP_M, V^1, V^2, V^3, V^4$
G	-	$sid, sid', IP_G, IP_M, V^1, V^2, V^3, V^4$
W_2	IP_D	$sid, sid', IP_M, V^1, V^2, V^3, V^4$
M	IP_D	sid', IP_M, V^2

C. Structure of Header and Forwarding State List

This subsection describes how a forwarding state list is created in the path-setup phase by routers on an end-to-end path. Figure 3 (a) shows two fragments of the header structure used by gPHI. A packet header contains both the fragments in the path-setup phase and only the upper fragment in the data-transmission phase. The three forwarding state lists V^1, V^3, and V^4 represent a sub-path on an end-to-end path. gPHI



(a) Structure of gPHI header

(b) Structure of forwarding state list V^j

Fig. 3. Header strutcure of gPHI

uses twelve types of packets, p_1 to p_{12}, as illustrated in Fig. 4. A *status* parameter is used to specify the packet type.

Figure 3 (b) shows the structure of a forwarding state list, which follows the design of dPHI [8]. The forwarding state list has a fixed number of slots (e.g., eight), where each on-path router stores its forwarding state. The forwarding state consists of the router's ingress and egress ports, and is encrypted with its secret key in the path-setup phase. The list is accessed as a circular buffer and contains an index pointing to the head of the list. The forwarding state list is created in a hop-by-hop manner in the path-setup phase. Before a path-setup request packet is sent, the index is set to a random value, and all slots are initialized as random bits. This index allows the next-hop router to know which slot the router accesses. When receiving the path-setup request packet, the router decides the egress port for its next-hop router according to IP routing, encrypts the ingress and egress ports to the slot specified by the index as the forwarding state, and sends the packet to the next hop after incrementing the index. Finally, the packet is forwarded to one of the destinations, that is guard G, helper M, or destination D.

Each forwarding state is encrypted by a router of interest using authenticated encryption such as AES-GCM. Thus, an adversary cannot decrypt any forwarding state. In addition, because the list uses a circular buffer and the index has a random initial value, its position in the circular buffer does not leak any information about the hop count from the source to the router of interest. In other words, the adversary cannot obtain any underlying network topology information from forwarding state lists [8].

D. Path Setup Procedure

This subsection describes the path-setup procedure illustrated in Figure 4. Hereafter, we refer to call packets in the path-setup phase as *packets* and those in the data-transmission phase as *data packets*.

Twelve types of packets, p_1 to p_{12}, are used in the path-setup phase. A sub-path is denoted by the symbol P_{X-Y}, using symbols of nodes, for example, X and Y. Further, two sessions are defined. One session, which is identified as sid, is used by S, G, D, W_1, and W_2 to identify and authenticate each other. These nodes create the forwarding lists V^1, V^3, and V^4. Another session, which is identified by sid', is used by G, M, and W_2 to identify and authenticate each other. These nodes create the forwarding list V^2. The objective of using two separate sessions is to prevent helper M from knowing any of the forwarding lists V^1, V^3, or V^4, as described in Section III-A.

The path-setup procedure consists of two rounds of packet exchanges, as illustrated in Figure 4.

1) First Round: The goal of the first round is to create the forwarding state lists V^1 and V^2, and assign midway routers W_1 and W_2. Source node S initializes the forwarding state list V^1 and sends the packet p_1 with V^1 to guard G. Packet p_1 has the helper's IP address encrypted with the public key of G, that is, $\{IP_M\}_G$, and the destination's IP address encrypted with the public keys of G and M, that is, $\{\{IP_D\}_M\}_G$. By decrypting them, G knows the helper's IP address IP_M but not the destination node's IP address IP_D. Helper M only knows the destination node's IP address encrypted with the public key of M. During the transmission of p_1, forwarding state list V^1 is updated at each hop by leveraging IP_G and sid. Subsequently, G initializes forwarding state list V^2 and sends packet p_2 with V^2 to M to create V^2 so that M sends reply packet p_3 to G. Unlike V^1, V^2 is identified and authenticated with a different identifier, sid', rather than sid. Packet p_2 has the destination node's IP address encrypted by the public key of helper M; that is, $\{IP_D\}_M$, so that M knows D.

Helper M, then sends packet p_3 back to G, which is then forwarded according to V^2. Packet p_3 has the destination node's IP address IP_D in plaintext to let routers on sub-path P_{M-G} know D. One of the routers on this sub-path decides to become the second midway router W_2. Subsequently, W_2 forwards packet p_4 to G according to V^2. Here, IP_D is removed from packet p_4 and stored in w_2 on path P_{W_2-G}. When packet p_4 reaches G, G knows that W_2 has been determined and sends packet p_5 back to S. Note that G does not know who the second midway router is. Meanwhile, one of the routers on sub-path P_{S-G} decides to become the first midway router W_1 according to IP_M, which is then removed from the packet. Now, S receives packet p_6, which allows it to view forwarding list V^2 and understand that W_1 and W_2 have been determined. Packet p_6 also has the second session identifier sid' and V^2 encrypted by the secret key of G, which is used in the second round.

2) Second Round: The goal of the second round is to create forwarding state lists V^3 and V^4. S sends packet p_7 with V^1 and $\{sid', V^2\}_G$. V^1 is used to forward p_7 to G by routers on sub-path P_{S-G}, and $\{sid', V^2\}_G$ is used by G to forward the next packet, p_8, to W_2. Then G sends packet p_8

to W_2. Packet p_8 is forwarded according to V^2 decrypted by G. When W_2 receives packet p_8, it initializes V^3, reads the destination node's IP address IP_D in plaintext from w_2, and records IP_D in the packet p_9 header. Packet p_9 is forwarded to destination node D according to IP routing, and routers on sub-path P_{W_2-D} create V^3, which corresponds to session identifier sid.

After receiving packet p_9, D sends back packet p_{10} to let W_2 know that V^3 has been created. Then, W_2 initializes V^4 and sends packet p_{11} to G to create V^4. Packet p_{11} is forwarded, and V^4 is constructed according to V^2. However, V^4 corresponds to session identifier sid rather than sid'. When packet p_{11}reaches G, V^4 is created. Finally, G removes V^2 from the packet to prevent the source AS from intercepting it and sends packet p_{12} back to S to let it know lists V^3 and V^4.

E. Circular Path Attack and Countermeasure

1) Circular Path Attack: As explained in Section III-A, the introduction of a guard prevents source AS and helper colluding attacks and midway attacks. However, it is open to a new attack: *circular path attack*. ASes that are on both sub-paths P_{S-G} and P_{W_2-D} can perform a circular path attack. Such a malicious AS knows the destination node's IP address because it is on sub-path P_{W_2-D}. The malicious AS can also become the source AS with some probability because it is on sub-path P_{S-G}. If it becomes the source AS, it knows both destination's and source's IP addresses. Consequently, host-level relationship anonymity is broken. Moreover, even if the common AS of the two sub-paths is not the source AS, the routers of the malicious AS can perform a topological attack better than midway routers W_1 and W_2. This is because this AS is more or equally near both the source and destination nodes than to W_1 and W_2, respectively.

2) Countermeasure: Selection Method of Guard and Helper: To prevent a circular path attack, we design a heuristic method for selecting the guard and helper to satisfy the following condition: two sub-paths P_{S-G} and P_{W_2-D} must not traverse the same AS. Here, let A_S, A_D, A_G, and A_M be the source AS, destination AS, guard AS, and helper AS, respectively. Further, let $anc(A)$ and $des(A)$ be the sets of ancestor and descendent ASes of the AS of interest, that is, A, respectively. An ancestor (descendent) AS has a recursive provider (customer) relationship with the AS of interest. For example, if AS A is a customer of AS B, then A is a descendent AS of B.

We ensure a sufficient condition by limiting the IP routing paths used by the above ASes. The limit is that A_S is a descendent AS of A_G and A_D is a descendent AS of A_M. Under this limit, the sufficient condition is defined as follows: $A_G \in anc(A_S), A_G \notin anc(A_D)$ and $A_M \in anc(A_D), A_M \notin anc(A_S)$. This sufficient condition is interpreted as follows. First, A_G is selected from the ancestor ASes of source AS A_S, whereas A_M is selected from the ancestor ASes of destination AS A_D. Second, A_G and A_M must not be ancestors of A_D and A_S, respectively. Accordingly, A_G and A_M are selected from AS sets $anc(A_S) - anc(A_D)$ and $anc(A_D) - anc(A_S)$, respectively.

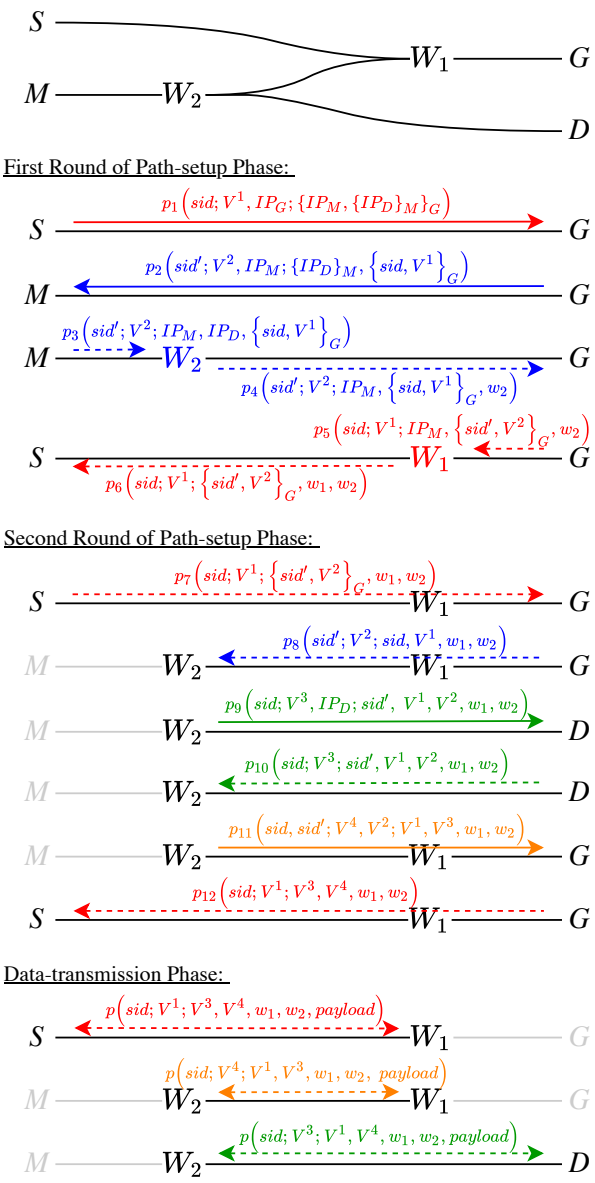


Fig. 4. Detailed protocol of gPHI

We prove that the selected A_G and M satisfy the above condition using a proof by contradiction. We assume that a common AS A_C exists on two sub-paths P_{S-G} and P_{W_2-D}, where A_G and A_M are selected. A_C is an ancestor AS of both A_S and A_D, and thus, $A_G \in anc(A_S)$ and $A_G \in anc(A_D)$. However, this contradicts the selection condition $A_G \notin anc(A_D)$. Thus, we have proved that the two sub-paths do not traverse the common (same) AS.

Consequently, the AS-level destination anonymity set of W_1 becomes $des(A_M) - des(A_G)$ and the AS-level source anonymity set of W_2 becomes $des(A_G) - des(A_M)$. Note that such guards and helpers do not exist in some cases. In one case, the source AS is a stub AS directly connected to

the destination AS. In another case, both the source AS and customer are stub ASes of the same provider AS.

F. Security Analysis

This subsection analyzes the resilience of gPHI and compares it with the following anonymity protocols: LAP [4], Dovetail [5], PHI [7], and dPHI [8], as summarized in Table III.

Existing studies assume three types of adversaries in their threat models: $T1$, $T2$, and $T3$. In $T1$, one of the AS on the end-to-end path is malicious. In $T2$, the source AS is malicious. In $T3$, end hosts not on the path are malicious and collude with a malicious AS. In this study, we have assumed an adversary $T4$, where a midway router undermines AS-level relationship anonymity, and an adversary $T5$, where a helper is malicious and colludes with the malicious source AS.

Attacks $A1$ to $A5$ are performed under threat models $T1$ to $T3$, and attacks $A6$ and $A7$ are performed under threat models $T4$ and $T5$. Only gPHI considers all types of attacks.

a) Direct de-anonymization attack (A1): breaks the unlikability of source and destination nodes' IP addresses if they are written into packets in plaintext or if they are encrypted with a weak ciphertext. In the context of gPHI, each forwarding state is encrypted by the corresponding router's secret key to defend against such an attack.

b) Topological attack by accessing forwarding state lists once (A2): reveals the topological information about an end-to-end path, such as path lengths, from the forwarding state itself. This attack narrows down the anonymity set using prior knowledge of the network topology [7]. Fixed-length design of forwarding state lists of PHI and its successors, including gPHI, hide such information from adversaries.

c) Topological attack by accessing forwarding state lists twice (A3): narrows down the anonymity set by accessing a forwarding state list, which is discussed by Bajic and Becker [8]. In a passive manner, a malicious AS observes a forwarding state list twice, that is, before and after the creation of a forwarding state list is completed. It calculates the number of slots modified by comparing the two observations, which indicates the hop count between the malicious AS and destination. In an active manner, a malicious AS rewrites slots to check if they are empty. If the rewritten entries are not empty, the packets will be lost. The adversary eventually knows the number of active entries, which is the hop count to either the source or destination nodes. dPHI prevents a passive attack by dividing the forwarding state list into V^1 and V^2. To detect a active attack, the source node checks the number of slots that have been rewritten. If the number is greater than the hop count estimated in advance, the source considers that the forwarding state list to have been altered by the adversary. gPHI inherits the above features except for the check of V^4. For the first packet of the data-transmission phase, the midway router that initializes V^4 should also verify it in the same way as dPHI.

d) Slot collision probability attack (A4): leverages the collision of slots, which occurs only in PHI. dPHI solves the problem by incorporating a circular design of forwarding state lists.

TABLE III
THREAT MODELS OF AND DE-ANONYMIZING ATTACKS ON ANONYMITY PROTOCOLS

		LAP	Dovetail	PHI	dPHI	gPHI
Threat model (nodes to be compromised)						
(T1) One on-path AS		✓	✓	✓	✓	✓
(T2) Source AS		✗	✓	✓	✓	✓
(T3) Off-path hosts		✗	✗	✗	✓	✓
(T4) Midway against AS-level anonymity		✗	✗	✗	✗	✓
(T5) Helper colluding with source AS		-	✗	✗	✗	✓
Attack scenarios						
(A1) Direct de-anonymization attack		✓	✓	✓	✓	✓
(A2) Topological attacks by accessing forwarding state lists once		✗	✗	✓	✓	✓
(A3) Topological attacks by accessing forwarding state lists twice		✗	✗	✗	✓	✓
Other attacks	(A4) Slot collision probability attack	✓	✓	✗	✓	✓
	(A5) Chosen-ciphertext attack	✗	✗	✗	✓	✓
	(A6) Topological attack with RTT measurement	✗	✗	✗	✗	✓
	(A7) Collusion of helper and source AS	-	✗	✗	✗	✓

✓: The property is considered in that protocol.
✗: The property is not considered in that protocol.
- : That protocol does not meet the prerequisite of the property.

e) Chosen-ciphertext attack (A5): is also solved by dPHI because of adoption of a cryptosystem with freshness.

f) Topological attack with RTT measurement (A6): is a new type of attack. gPHI achieves better AS-level relationship anonymity than the existing protocols against topological attacks and mitigates an attack with RTT measurement.

g) Source AS and Helper Colluding Attack (A7): is newly defined. Only gPHI achieves host-level relationship anonymity against this attack.

IV. Experimental Analysis on Topological Attack

This section experimentally evaluates how gPHI and dPHI provide AS-level relationship anonymity against topological attacks using RTT measurements.

A. Evaluation Method

1) Metric: Existing studies have adopted the size of AS-level relationship anonymity set as the metric for resilience against topological attacks [8]. However, the size does not consider the resilience against an attack with RTT measurement. In this study, we adopt the entropy of AS-level relationship anonymity set as the metric of leaked information from the set. We then define the entropies of AS-level relationship anonymity sets obtained by midway routers W of dPHI and W_1 and W_2 of gPHI when they perform the topological attack.

Here, let $\{s_i | i = 1, \ldots, n\}$ and $\{d_i | i = 1, \ldots, m\}$ be the AS-level source and destination anonymity sets, respectively, and n and m their sizes. Let $\Pr[A_S = s_i]$ and $\Pr[A_D = d_i]$ be the probabilities that candidate ASes s_i and d_j are the source and destination ASes, respectively.

The entropy of the AS-level relationship anonymity set for W and W_2 is equivalent to that of the AS-level source anonymity set because the IP address of the destination node is known. Thus, the entropy is defined as $-\sum_{i=1}^{n} \Pr[A_S = s_i] \times \log_2(\Pr[A_S = s_i])$. In contrast, the entropy of W_1 is defined as the AS-level destination anonymity set entropy. This is a conservative estimation of the entropy. In other words, this entropy would be the largest amount of information that the most powerful adversary can obtain. W_1 is between the source and guard G and is closer to the source AS than to the other ASes. We conservatively assume that W_1 knows the source AS and W_2 knows the AS to which the guard belongs. Thus, the entropy of the AS-level relationship set is equivalent to that of the AS-level destination set: $-\sum_{i=1}^{m} \Pr[A_D = d_i] \times \log_2(\Pr[A_D = d_i])$.

2) Simulation: We simulated the path-setup procedures of dPHI and gPHI on a graph modeling the Internet. We leverage the CAIDA AS Relationships dataset [11] to create a graph that expresses relationships and routing policies between ASes by *vnodes* and directed edges between vnodes, motivated by Pathlet Routing [12]. We assume that all ASes employ the valley-free routing policy, and use Dijkstra's shortest path algorithm [13] to determine the best AS paths between vnodes.

We create 1,000 end-to-end paths for dPHI and gPHI. In the case of dPHI, the source and destination ASes and the AS of a helper are independently and randomly selected from all ASes. In the case of gPHI, the same pairs of source and destination ASes as those of dPHI are selected. ASes that accommodate helpers and guards are selected according to the selection method proposed in Section III-E.

3) Topological Attack with RTT Measurement: The topological attack with RTT measurement is simulated for midway routers W, W_1, and W_2, as described below:

a) dPHI: We calculate an AS-level source anonymity set for each midway router, as described in Section II-D, and then simulate the attack with RTT measurement. in the following way: In this subsection, we define *AS links* as links between neighboring ASes. We assume that the propagation delays of an AS link follow a normal distribution, and that the means of the AS links' propagation delays also follow a normal distribution. For simplicity, we call propagation delays as delays.

The first step is to assign normal distributions of delays to all AS links in the graph. We leverage the King dataset [14], which records the actual delays between DNS servers. Since each delay in the dataset corresponds to an AS path, we calculate the mean of the delays divided by the hop count of the AS path as the mean delay of each AS link. Second, we fit the distribution of the AS links' mean delays to a normal distribution, and obtain the mean and the variance of the normal distribution of all the AS links' mean delays. Finally, we randomly choose the mean delay of each AS link in our graph from the above normal distribution. The variance of this AS link is calculated such that its mean and variance are on the regression line of the King dataset.

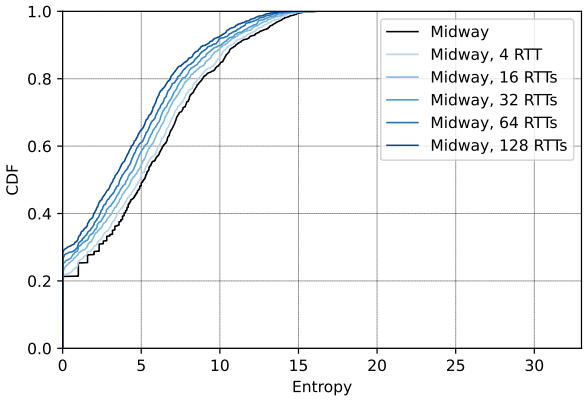


Fig. 5. Entropy of relationship anonymity in dPHI

The second step is to calculate the mean and variance of the AS path between each AS in the AS-level source anonymity set and midway router W by adding the random delay variables of all AS links on the AS path.

The third step is to simulate the attack. We choose k sample delays randomly from the distribution of delays of the AS path of interest. We classify the collection of k sample delays into the candidate path using the Naive Bayes classifier, leveraging the above distribution of AS paths between the midway router and the anonymity set.

b) gPHI: As described in Section III-A, it is difficult to perform an attack with RTT measurement in the case of gPHI. We assume that midway routers can only perform topological attacks to estimate AS-level source/destination anonymity sets. Midway router W_1/W_2 calculates the AS-level destination/source anonymity set, as described in Section III-E.

B. Evaluation Result

1) dPHI: Figure 5 presents the cumulative distribution function (CDF) of the entropies for the AS-level source anonymity sets of 1,000 midway routers when different numbers of delay values are sampled from $k = 0$ to 128. In the case of $k = 0$, the probability that each AS is the source AS is equal. Hence, the entropy of the anonymity set is equivalent to the binary logarithm of the anonymity set size. The midway router breaks the AS-level relationship anonymity for 21.4% of the 1,000 paths, even without using the RTT measurement attack. Furthermore, entropy decreases substantially as the number of measured RTT values increases. The average information gain of 128 measured RTT values is 1.5 bits.

2) gPHI: Figure 6 shows the CDFs of the entropies of the AS-level relationship anonymity set at for W_1 and W_2. Because an attack with RTT measurement is not performed, each candidate AS is the true destination/source AS with an equal probability. W_1 and W_2 break the AS-level relationship anonymity of the end-to-end paths by 0.6% and 1.1%, respectively. This shows that the selection method prevents the

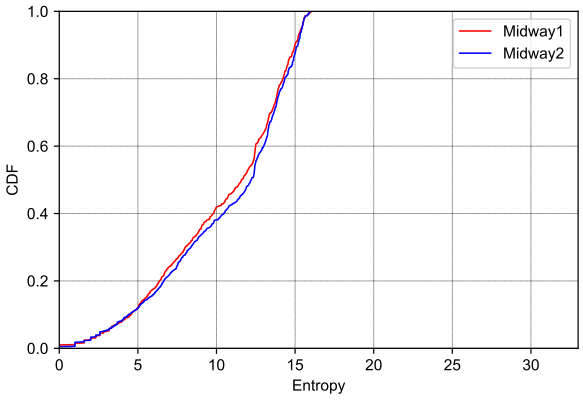


Fig. 6. Entropy of relationship anonymity in gPHI

circular path attack (the second case in Section III-B2), but the AS-level source/destination anonymity set size becomes one in the above cases. However, these probabilities are much lower than 21.4% in the case of dPHI.

The figure also shows that the leaked information in the case of gPHI is less than that of dPHI. The probability that the entropy of the AS-level relationship anonymity set of gPHI is narrower than a certain value is always lower than that of dPHI. This reduction partly comes from the fact that the average end-to-end path length of gPHI becomes longer than that of dPHI. Thus, we calculate the average hop counts of gPHI and dPHI, which are 6.9 and 5.0 hops, respectively. The average path stretch is approximately 1.9 hops. Based on the above experiments, we conclude that the entropy increases, the path stretch is well balanced, and gPHI improves AS-level relationship anonymity without incurring excessive path stretches.

V. Related Work

LAP [4] is considered a pioneer in lightweight anonymity protocols. Because a packet is directly routed by IP routing from the source to the destination, users must trust the source AS. Dovetail [5] has strengthened the threat model of LAP by tolerating a compromised source AS. To guarantee relationship anonymity against the malicious source AS, an indirect path is established via the *matchmaker* and *dovetail node*, which are similar to the helper and midway routers in PHI. Another notable feature of LAP is that an end-to-end path is created by leveraging Pathlet Routing [12] rather than IP routing.

PHI [7] changed the structure of the forwarding information from an array to a fixed-length list, which each router accessed randomly, to prevent forwarding state lists from leaking the hop count of an AS-level end-to-end path. dPHI [8] has presented several new attack scenarios, including those in which an adversary commits topological attacks by accessing the forwarding state lists twice. In addition, it assumes a new type of adversary that compromises both an on-path AS and an arbitrary number of off-path end-hosts. To address such attacks, dPHI employs a path-establishment method of two round trips and a new design of forwarding states such as a circular buffer, which gPHI also adopts.

HORNET [6] is another type of anonymity protocol implemented in the network layer. It is different from lightweight anonymity protocols in that it performs onion routing, and it does not use IP routing for path-setup but designs a source routing protocol, which makes the deployment difficult. Tor instead of IP [15] is an onion routing protocol implemented in the network layer. It is similar to gPHI in leveraging the hierarchy of ASes of the Internet for path construction.

VI. Conclusion

In this study, we designed gPHI, an anonymity protocol, by extending PHI/dPHI. The contributions of this study are two-fold: First, we assume more realistic scenarios than PHI/dPHI, where a server that plays an important role in path-setup is compromised, and extends it to be resilient against adversaries that compromise them. Second, we precisely define relationship anonymity at the host and AS levels and clarify how gPHI achieves better anonymity than PHI/dPHI.

References

[1] D. Xue, R. Ramesh, L. Evdokimov, A. Viktorov, A. Jain, E. Wustrow, S. Basso, and R. Ensafi, "Throttling twitter: an emerging censorship technique in russia," in *Proceedings of ACM Internet Measurement Conference*, 2021, pp. 435–443.

[2] S. Farrell and H. Tschofenig, "Pervasive monitoring is an attack," IETF RFC 7258, 2014.

[3] P. Syverson, R. Dingledine, and N. Mathewson, "Tor: The second generation onion router," in *Usenix Security*, 2004, pp. 303–320.

[4] H.-C. Hsiao, T. H.-J. Kim, A. Perrig, A. Yamada, S. C. Nelson, M. Gruteser, and W. Meng, "Lap: Lightweight anonymity and privacy," in *Proceedings of IEEE Symposium on Security and Privacy*, 2012, pp. 506–520.

[5] J. Sankey and M. Wright, "Dovetail: Stronger anonymity in next-generation internet routing," in *International Symposium on Privacy Enhancing Technologies Symposium*, 2014, pp. 283–303.

[6] C. Chen, D. E. Asoni, D. Barrera, G. Danezis, and A. Perrig, "Hornet: High-speed onion routing at the network layer," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1441–1454.

[7] C. Chen and A. Perrig, "Phi: Path-hidden lightweight anonymity protocol at network layer," *Proceedings on Privacy Enhancing Technologies*, pp. 100–117, 2017.

[8] A. Bajic and G. T. Becker, "dphi: An improved high-speed network-layer anonymity protocol." in *Proceedings on Privacy Enhancing Technologies*, 2020, pp. 304–326.

[9] N. Hopper, E. Y. Vasserman, and E. Chan-Tin, "How much anonymity does network latency leak?" *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 2, pp. 1–28, 2010.

[10] A. Pfitzmann and M. Köhntopp, "Anonymity, unobservability, and pseudonymity - a proposal for terminology," in *Designing privacy enhancing technologies*. Springer, 2001, pp. 1–9.

[11] "The caida as relationships dataset, 20210901," http://www.caida.org/data/active/as-relationships/.

[12] P. B. Godfrey, I. Ganichev, S. Shenker, and I. Stoica, "Pathlet routing," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 111–122, 2009.

[13] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[14] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating latency between arbitrary internet end hosts," in *Proceedings of ACM SIGCOMM Workshop on Internet measurment*, 2002, pp. 5–18.

[15] V. Liu, S. Han, A. Krishnamurthy, and T. Anderson, "Tor instead of ip," in *Proceedings of ACM Workshop on Hot Topics in Networks*, 2011, pp. 1–6.