Dovetail: Stronger Anonymity in Next-Generation Internet Routing

Jody Sankey and Matthew Wright

University of Texas at Arlington
jody@jsankey.com, mwright@uta.edu

Abstract. Given current research initiatives advocating "clean slate" Internet designs, researchers have the opportunity to design an internetwork layer routing protocol that provides efficient anonymity by decoupling identity from network location. Prior work in anonymity for the next-generation Internet fully trusts the user's ISP. We propose Dovetail, which provides anonymity against an active attacker located at any single point within the network, including the user's ISP. A major design challenge is to provide this protection without including an application-layer proxy in data transmission. We address this in path construction by using a *matchmaker* node (an end host) to overlap two path segments at a *dovetail* node (a router). The dovetail then trims away part of the path so that data transmission bypasses the matchmaker. We develop a systematic mechanism to measure the topological anonymity of our designs, and we demonstrate their privacy and efficiency by Internet-scale simulations at the AS-level.

1 Introduction

When we use the Internet, a wide range of identifying information is commonly revealed, but one of the hardest forms of identity to remove is that defined by the network routing protocol (*layer 3*), since this identity is used to deliver data. In today's Internet, IP is the primary layer 3 protocol and IP addresses are in every data packet. Recording a user's IP address can allow an adversary to uniquely identify her, link that identity with her online activity, correlate connections to different services, and partially reveal her geographical and network locations. Previous work has proposed *low-latency anonymity systems* to conceal a user's identity [1, 2], including her IP address. Tor in particular has been adopted by hundreds of thousands of privacy-concious users worldwide [3]. Current anonymity systems, however, work by creating an overlay network on top of the layer 3 protocol, requiring a sequence of IP transmissions to disguise the original sender. This sequential forwarding and the queueing and processing required in intermediary nodes create substantial delay and overhead.

We prefer an alternative formulation for this problem: Rather than attempting to conceal a global layer 3 identifier by adding complexity in application protocols, we believe that the layer 3 protocol should not reveal a global identity. Instead, we leave identity management to higher layers in the protocol stack, in only those applications where it provides mutual benefit.

While privacy by itself is unlikely to motivate a change away from IP routing, a range of additional concerns have emerged within the networking field [4], including scalability, security, mobility, challenged environments, and network management, leading to major research initiatives investigating "clean slate" Internet designs [5–7] that could be used to build the *next-generation Internet (NGI)*. A wide range of different NGI routing concepts have already been proposed as a result of these activities [8–14]. Network virtualization research, showcased in testbeds such as GENI [15], offers hope for a progressive transition to a future routing protocol. These initiatives in NGI provide an opportunity to imagine anonymous communications that do not rely on an overlay network.

We thus propose *Dovetail*, an NGI routing protocol that prevents association of source and destination by an attacker located at any fixed point within the network. Recently, Hsiao et al. proposed LAP, a lightweight NGI anonymity protocol [16]. Unlike LAP, however, Dovetail provides protection against observation by local eavesdroppers and by an untrusted ISP, which is a critical requirement for many privacy-conscious users.

A major design challenge is to provide this protection without including a proxy in data transmission, which would be much slower than only traversing routers. We address this challenge in path construction by asking a *matchmaker* node (an end host) to put together two path segments so that they overlap at a *dovetail* node (a router), and enabling the dovetail to trim away the part of the path with the matchmaker. This technique is implemented using public-key operations only at the source and the matchmaker, while routers use only symmetric encryption and decryption of short header fields and a simple hash chain. The protocol enables the choice of many different paths through the network and does not require a trusted third party.

In brief, our key contributions are: (1) a novel privacy-preserving NGI routing protocol, (2) a systematic mechanism for measuring anonymity in terms of topological identity, and (3) evaluation of our protocol in terms of topological anonymity using an Internet-scale simulation.

2 Objectives

In this section, we describe the goals of the system we intend to deliver and the attacker we design against.

2.1 Anonymity Objectives

We refer to the party who initiates a connection as the *source* and the opposite party as the *destination*, although data is able to pass in both directions once the connection is established. Using the terminology of Pfitzmann and Hansen [17], we aim to provide *unlinkability* between the source and destination, such that no network location is able to sufficiently distinguish whether the source and destination are related, except for the source itself. This implies that network locations with good information on the source identity have little information

on the destination identity, and vice versa. Throughout our work, we constrain ourselves to the identifying properties defined at the network layer: network identity and network location, or *topographical anonymity* [16].

We do not protect the packet contents, which reside in higher network layers and are thus out of scope for this paper. Content should be protected end-to-end using a protocol such as IKEv2, which protects sender and receiver identities [18]. Such protection is effectively mandatory for strong anonymity protections, as many other forms of Internet identification exist, such as device fingerprinting [19] and persistent cookies [20]. Additionally, higher-level protocols like IKEv2 should be used with restricted options and implementations to limit the possibility of fingerprinting.

2.2 Performance & Practicality Objectives

Any anonymity system must route traffic fast enough to gain widespread adoption and thus provide a large set of potential message sources [21]. Performance problems with Tor have been widely discussed, and they are considered an important factor limiting its adoption [22,23]. We aim to provide a lightweight system where all communication for an established connection remains within the core networking infrastructure and occurs at layer 3. This avoids the frequently slow *last mile* connections [24] in overlay anonymity systems and also the queuing required to move between layers in the protocol stack. Finally, we require that our system provides mechanisms to trade anonymity for performance.

Another key to widespread adoption is recruiting service providers. Our work targets a future Internet, so Dovetail need only compete with other future routing protocols rather than motivate service providers to switch away from IP. Today's ISP business models may not apply, but it is unlikely that service providers are willing to spend substantial time and infrastructure for privacy. Our goal is to ensure that costs for service providers are limited, such that benefits for privacy-aware consumers are enough incentive to participate in the protocol. To this end, we recognize that Internet routers have high throughput and low computing resources per flow, so we limit cryptographic operations and avoid maintenance of any per-connection routing state. Additionally, our design does not require significant extra traffic and does not violate basic notions of consumer-provider relationships that exist in today's Internet.

2.3 Attack Model

Selecting an attack model for anonymity systems is a challenging task in its own right, as the adversary may be different for different users and its capabilities are not known in advance. A few key points guide our choices. First, protecting a low-latency connection from an adversary who can observe traffic at multiple points of the network is very difficult. Tor uses layered encryption and fixed packet sizes to prevent trivial linkability, but this comes with significant expense and does not hide traffic patterns, which are linkable with a small chance of error [25]. Adding sufficient delays and cover traffic to mask traffic patterns is

expensive and can be undermined by manipulating the patterns [26, 27]. Second, users may be suspicious of any service provider that can link them with their Internet activities. This applies to anonymity service providers, such as Anonymizer.com, and also to Internet service providers. ISPs have proved to not be fully trustworthy with private browsing data [28,29]. We therefore aim to prevent any element of our system from being able to deanonymize users. Third, a user's local communication may be subject to eavesdropping, e.g. at a wireless hotspot or by an employer. Unlike LAP, we aim to protect against such adversaries. Fourth, many of the adversaries that we aim to protect against would be capable of various active attacks, such as replay or packet header manipulation, so we also aim to limit the exposure that such attacks might cause.

We thus consider an adversary who is *active* but *local*. Active means the adversary is able to initiate connections and to violate the rules of the protocol for the connections in which she is involved, in addition to passively monitoring these connections. We define local as confined to a single *Autonomous System* (AS) within the Internet. ASes are the level at which routing information and policies are commonly shared, so a compromise in security at one router may affect multiple routers controlled by the same AS. In contrast, in order to span multiple ASes, an attack must either compromise multiple organizations or involve collusion between these organizations. We note that if a particular set of ASes were suspected of collusion, our client logic could easily be modified to include no more than one member of the set in each connection. Our adversary is assumed to have local knowledge of traffic, but global knowledge of the network topology and routing data.

More concretely, the possible attackers we aim to protect against include: a local eavesdropper, the source ISP, the destination ISP, any single AS in between, any node facilitating our protocol operations, and the destination itself. Thus, we aim for significantly greater protection than LAP or a centralized proxy server like Anonymizer.com.

Given that we only protect against a single observation point, we offer no protection against attacks that require multiple observation points, even though such attacks may be practical for state-level adversaries [30] or Internet exchange points [31]. In common with LAP, but not Tor, we do not try to prevent trivial linkability based on packet contents and sizes. This means that linking attacks with multiple observation points need lower computational and storage resources and succeed with fewer observations than against Tor. Additionally, if both the source and destination are customers of the same ISP, it is simple for the ISP to correlate traffic. Again, Tor provides basic protection that makes this attack slightly harder, while both LAP and Dovetail provide no protection.

3 Background

In this section, we cover two research areas of direct relevance to our problem: source-controlled routing protocols and low-latency anonymity systems. Within each area, we describe a proposal that our design builds upon.

3.1 Source-Controlled Routing

One theme spanning a number of next-generation Internet routing proposals is that of source-controlled routing, in which the originator of a data packet has some control over the route it takes, usually using routing control information carried in the data packet. In some protocols, the source has influence over the route but not complete control [12, 14]; in others, the source explicitly declares the route that should be taken [10, 13]. As we explain in Sect. 4.1, this ability to express a route at the source has benefits for anonymity in addition to the robustness and flexibility considerations that initially motivated the research.

Pathlet Routing Pathlet routing [10] is one example of a source-controlled routing system. Each entity within a network defines a number of virtual nodes (or *vnodes*) and advertises path segments (or *pathlets*) that pass between these vnodes. Vnodes are a virtual construct, so a single physical router may process packets for multiple vnodes, or a single vnode may be distributed across multiple physical routers. Each vnode is defined by a forwarding table containing the set of allowed outgoing pathlets. All packets arriving from a particular communication peer are processed by one vnode whose forwarding table defines the set of allowed routes for that peer. The pathlet protocol provides an expressive system that is able to represent many different types of routing policy.

To send a packet, the source assembles a list of adjacent pathlets defining the intended route and includes this list in the packet header. Each pathlet is represented by a variable length Forwarding ID (*FID*), an index into the forwarding table of the vnode that defined the pathlet. When a vnode receives a packet, it removes the first FID and uses this as an index into its forwarding table to determine which link the packet should be sent over. Only legal routes are defined in the forwarding tables. Therefore, it is impossible to violate the routing policy by invoking unannounced routes, since no such routes exist. Pathlet routing moves the responsibility for network route creation from the network infrastructure to the end hosts originating traffic. This provides two features that are helpful for the design of Dovetail: First, the large routing information base embodying network topology need only be consulted each time a new route is constructed, and not each time a packet is forwarded. Second, it provides flexibility for an end host to control how its packets will traverse the network.

3.2 Low-Latency Anonymity Systems

A number of low-latency anonymity systems have been proposed with response times that are sufficient for general-purpose interactive use, such as Web browsing. Some of these have been fielded [1, 2, 32]. Current low-latency anonymity systems may be categorized as either centralized or distributed. Centralized systems pass all traffic though an anonymizing proxy, which must be trusted. Distributed systems overlay an additional network on top of the current layer 3 protocol and therefore require multiple IP transmissions to deliver each packet

from source to destination. These multiple transmissions, together with processing inside the intermediate hosts, contribute to latencies that are substantially higher than Internet usage without anonymization [33].

Lightweight Anonymity and Privacy In Lightweight Anonymity and Privacy (LAP) [16], Hsiao et al. propose the anonymity scheme that inspires our work. Their protocol relies upon *packet-carried forwarding state*, where the information required to deliver a packet is stored within the packet itself. To establish a connection, the source constructs a packet containing a sequence of *autonomous domains* (ADs) describing the route. As each AD receives the packet, it encrypts its own routing instruction using a private symmetric key and forwards the packet to the next AD. Once a connection has been constructed in this manner, data may be exchanged between the endpoints using the resulting encrypted header. Each path construction request contains a nonce that influences the encryption process, allowing a source to construct multiple unlinkable connections over the same route by using different nonces. Header padding may be included to partially obfuscate the path length. During construction, each AD on the path learns the identity of all ADs that follow it but not the identity of the ADs before it. Some information on predecessor identity may be inferred based on knowledge of the preceding AD, network topology, routing policy, observed header length, and observed response time, but these are not quantified. LAP assumes the user's own ISP is trustworthy, and it provides no protection of source-destination unlinkability against a local eavesdropper or an observer at the source ISP. Given previous well-publicized ISP indiscretions [28,29] and the possibility of a hacker infiltrating this single point of failure, it seems unlikely that privacy-conscious users will share this assumption.

Other than LAP, ANDāNA is the only other next-generation Internet anonymity protocol that we know of [34]. It is only designed for named-data networks and it is built using onion routing, both of which are very different from Dovetail.

4 Design

In this section, we first provide context for our design point and then describe the protocol from four different perspectives in increasing detail.

4.1 Layer 3 Anonymity Design Space

To provide a broadly applicable anonymity system, we assert that any layer 3 solution should provide two features:

Deviation from shortest path. An eavesdropper can measure information on the length of the network path before and after her vantage point. If a routing protocol always selects the shortest possible route, then when the shortest route between participants is significantly shorter or longer than the Internet average, the protocol will reveal this information and limit their anonymity.

Partitioned routing information. When the routing information is stored as a single field, such as an IP address, any entity with access to the field may calculate the destination identity. When routing information is divided across multiple fields, then an entity must access multiple fields to learn the destination identity. Fields may be protected independently to prevent this access.

Source-controlled routing is useful since it accommodates both of these features: when the source of a message can dictate a path, she is free to pick one that is not the shortest, and she may express the path as a separate instruction for each entity along the route. Dovetail builds upon the pathlet source routing protocol presented by Godfrey et al. [10]. Pathlet routing works well for our system, but we are not reliant on any unique feature of this protocol. The principles we describe could be applied to any protocol that provides complete control over the selected route and a wide range of allowable routes.

4.2 Network Model

We propose a clear distinction in routing at the AS boundary; each AS should expose the minimum number of vnodes and pathlets necessary to satisfy its routing policies. This distinction provides two practical benefits: First, minimizing the number of externally visible vnodes reduces the size of the routing information base that must be held in end hosts. Second, distinguishing between internal and external connectivity allows an AS to retain a flexible and dynamic internal routing policy. Adjacent ASes share routing information to establish the network topology. This communication should be secured against MITM attacks that could selectively filter the topology. We assume that hosts know the numeric identity of the vnodes they wish to contact. An equivalent to DNS would be required to translate human-readable identities into vnode identities. The translation service itself could be accessible using Dovetail to protect privacy, but is outside the scope of our current work.

The most common form of routing policy used in the Internet today is *valley-free routing* [35], which reflects the contractual relationships between ASes. A *customer* AS is one who pays a *provider* AS to forward its traffic, while two ASes with a *peer* relationship will each forward each other's traffic without payment. In valley-free routing, each AS will only forward traffic when there is a financial incentive to do so, i.e. when the traffic originates from or is destined for a paying customer. As illustrated in Figure 1a, two vnodes are required per AS to enforce this strict definition of a valley-free routing policy: one to receive traffic from customer ASes and one to receive traffic from peer and provider ASes. Although valley-free routing is common, Internet routing allows for arbitrarily complex policies, and valley-free routing is not ubiquitous [36]. In particular, there are a growing number of Internet exchange points (IXPs), which offer ASes the ability to peer with each other and thereby save money [37]. Most transit and access provider ASes will peer with any non-customer AS [38]. This suggests that peering is compatible with ASes' incentives and is likely to continue to be common.

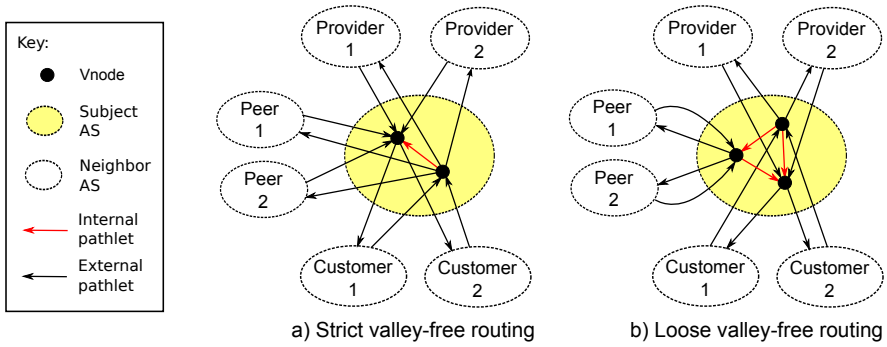

Fig. 1. AS vnode and pathlet structure by routing policy

We thus consider a slightly relaxed routing policy, which we refer to as *loose valley-free*. In this scheme, an AS will allow traffic to pass between its peers. The AS would not receive payment from a customer for performing this service, but also is not required to make a payment and could avoid payments at other times if peers provide a reciprocal service. As shown in Figure 1b, three vnodes are required per AS to enforce a loose valley-free routing policy: one to receive traffic from customer ASes, one to receive traffic from provider ASes, and the third to send and receive peer traffic.

For good anonymity properties as described in Section 4.4, Dovetail relies on a modest fraction of ASes using the loose valley-free policy or other policies that are less strict than valley-free routing. If all ASes use strict valley-free routing, Dovetail still provides anonymity, but with smaller anonymity sets.

4.3 Path Construction

Figure 2 illustrates the Dovetail path creation process. A Dovetail path comprises multiple *path segments*. As with LAP, an AS that is present on a path segment may learn the identity of subsequent ASes and its direct predecessor, but not earlier ASes.

The path cannot be constructed directly from the source to the destination, since the source's ISP would be able to link source and destination. Instead, we make use of a randomly selected, untrusted third-party vnode called the *matchmaker*. This matchmaker may either be an end host or functionality exposed by a service provider. Providing matchmaker services should cost little relative to enabling our protocol in routers. The identities of vnodes willing to act as matchmakers could be distributed as a part of routing information maintenance.

The source encrypts the identity of the final destination using a public key for the matchmaker and builds a *head* path segment to the matchmaker, who then extends the path to the destination with a *tail* path segment. Here, the source ISP no longer learns the identity of the destination, only of the matchmaker. The matchmaker learns the identity of the destination, but cannot identify the source through the intervening ASes. The source may learn the matchmaker's public key

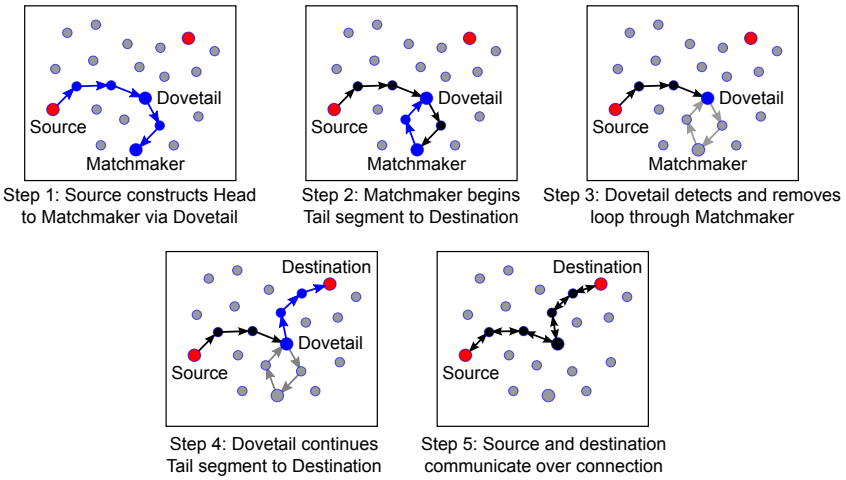

Fig. 2. Construction of a Dovetail connection

without compromising anonymity by requesting a signed certificate over the same path used to establish the connection. To improve performance and minimize the trust we must place in the matchmaker, we prefer that the matchmaker not be involved in the exchange of data. Therefore, we require that the head and tail segments cross at some vnode, referred to as the *dovetail*[1]. The source encrypts the identity of the dovetail and provides it to the matchmaker for inclusion on the tail segment. The dovetail detects the crossing condition and joins the two segments, removing the loop in the path along with the matchmaker.

The tail path segment would ideally be selected by the source, but the source does not have complete knowledge of distant Internet topology. The matchmaker has sufficient knowledge to construct a path to the destination, but the user's anonymity can be degraded if an AS appears on both the head and tail segments, and therefore we prefer that the tail segment avoids ASes already used on the head segment. Providing a list of head ASes to the matchmaker would reveal substantial information on the source identity, so instead we ask the matchmaker to return a set of potential tail routes that the source selects from. The source then sends its choice to the matchmaker to complete the route.

4.4 Segment Route Selection

A source-controlled routing system may attempt to obfuscate path length, but an attacker located on the path will be able to infer some information about her distance to the source and destination through round trip timing, packet length and structure analysis, and active probing. We prefer a system that is robust even when an attacker learns path length to one that relies on keeping it

[1] We use the term to reflect a dovetail joint in carpentry, where two elements are joined securely and compactly

hidden. For the remainder of the discussion, we assume the attacker has perfect knowledge of the number of ASes preceding and following her own, but limit the value of this knowledge through a non-deterministic path selection process.

Our mechanism for routing each path segment is based upon the principle of *path diversity*, where a large number of possible paths may be taken from any given source to any given destination. We note that this is beneficial for the robustness of the system in addition to its anonymity. To achieve path diversity, each host must have a comprehensive, but not necessarily complete, map of the network. We extend the pathlet routing protocol by exporting extra pathlets in addition to the shortest path tree (SPT). The optimal set of additional pathlets depends on network size and topology, but our experiments show that for the current Internet, is it appropriate to export 50% of the SPT size, selecting pathlets closest to the sender. An important consequence is that routing knowledge varies across the network, and so any assessment of available path options can only be made in the context of the vnode (in our case, the source or the matchmaker) selecting the path. Maintenance of routing information in response to network changes could be performed using path vector distribution methods similar to BGP [39], but this is not relevant to the anonymity properties of the system and so is not discussed further.

When a host constructs a path segment, it will normally have a wide range of options available with different *costs*, where we define cost as the number of times the route changes AS. Other cost metrics such as latency or bandwidth could also be integrated into the protocol. The distribution of options across cost reflects the network topology between the source and destination. Selecting a random path uniformly from among the complete set of options would reveal information about this distribution, such as picking the most common path cost most frequently, and thus leak information about the topology. Instead, we use a *cost window approach*: we select a path by first selecting a path cost and then randomly selecting one of the paths at this cost.

4.5 Data Packet Structure

Dovetail extends the basic packet format used in pathlet routing, providing a set of different packet types composed of variable-length segments. Each dovetail path is constructed using a path construction packet and a construction return packet. Data is then exchanged over the path using a sequence of encrypted data and encrypted response packets. The data formats and processing algorithms for these packets are provided in our technical report [40]. In summary, these algorithms provide the following security properties:

1. An AS does not learn the identity of ASes before its immediate predecessor.
2. AS routing information is protected by a key known only to the AS.
3. Different connections travelling over the same route do not produce the same ciphertext.
4. The final ciphertext for each AS depends on the entire path.
5. An AS may only create a removable loop in the path when given access to privileged information. This information is only given to the matchmaker.

5 Security Analysis

In this section, we assess the security of the Dovetail protocol. We consider a range of anonymity attacks that might be applied against the protocol and then analyze the information available to a passive attacker at each point in the network. We end with brief discussions of timing attacks and attacks on availability and integrity.

5.1 Attacks on Anonymity

As Dovetail is lightweight, it does not protect against attacks that succeed against an overlay system like Tor. In particular, an entity who can observe traffic at multiple points in the connection can link both of those points, which can link a source to her destinations. In Dovetail, this is trivial, as the packet contents are not encrypted differently at different points in the network. In Tor, however, timing analysis can enable this linking with high accuracy [26,27]. Other attacks that rely on multiple points of observation, such as selective denial of service [41] and predecessor [42] attacks will be just as effective in Dovetail. Additionally, Dovetail is vulnerable to the same types of side-channel attacks that impact Tor [43–47].

The primary information available to a passive attacker in the network is the cost to the source and destination and the preceding and following ASes in the path, and we examine the affect of these on anonymity in Section 5.2. Beyond this, however, we need to examine additional attacks that could leverage the unique aspects of the Dovetail protocol. These attacks include:

Observe or correlate packet content. Dovetail is a layer 3 protocol and does not provide any protections for the data it carries. In cases where packet content would reveal identity, or where confidentiality is important, a higher layer protocol such as IKEv2 should be used to provide encryption [18].

Correlate connections from a source. Each connection includes a source-defined nonce. When the source changes this nonce, a different ciphertext will be produced, preventing an observer from associating multiple connections over the same path from their header content. When connections between a source-destination pair are distinctive, and may hence be correlated by some other property, the source could reuse the same matchmaker and path to prevent intersection and predecessor attacks.

Replay packets. A replayed packet will take the same path as its original transmission and therefore not provide an attacker with new information. An adversary might try to probe for the source by prepending an unencrypted path to a recorded packet, but each AS empties the unencrypted segment on receipt to prevent this attack.

Probe for a later AS. To determine the destination of an observed connection, an attacker on the head segment may try to construct many new connections through the same dovetail and search for matches in the header ciphertext.

Dovetail protects against this attack by including a hash of the entire path in the IV for encrypted transit segments. Any change in the selected path will therefore perturb the ciphertext for all segments.

Probe for an earlier AS. The joining of a Dovetail path provides confirmation that the joining AS appeared on the path twice, and an attacker may wish use this feature to probe for suspected predecessors. During connection construction, an attacker may attempt to extend the path to a suspect and then back to herself, where she could observe whether a join occurred. Our use of hash chaining prevents this attack, since the attacker cannot replicate the nonce initially presented to the suspect. The matchmaker is provided with an earlier nonce to create a legal join and may perform some probing, but this is heavily constrained by the dovetail-matchmaker cost limit.

Matchmaker intersection. The matchmaker provides the source with a set of possible tail segments from which the source picks one. Since the source will not select an AS already on the head segment, including it's own ISP, the matchmaker could try to offer tail segments that help it isolate possible source ASes. In particular, if there is a source AS of interest A, then the matchmaker could pick tail segments that include likely ASes between itself and A. If the source avoids these tail segments, it adds to the likelihood that the source is in A. However, fully unmasking the source AS with this type of intersection attack would require a large number of requests. As matchmakers are selected randomly from a large set, an attacker located at any particular matchmaker is unlikely to receive many connection requests from the same source.

Modify the requested path. An AS along the path could modify the unencrypted header segment to alter the route taken for the remainder of the path segment, but gains little from doing so. All vnodes along a path segment can identify the destination, and earlier vnodes have a better knowledge of the source. Thus, an attacker that places herself later in the same path segment does not learn any additional information regarding source or destination.

Modify the tail path. The matchmaker could use a different tail option than that selected by the source. However, the matchmaker does not learn whether unselected paths were acceptable and cannot identify the source and so cannot predict whether a particular path will be bad for that source. A matchmaker could speculatively route all connections through a particular ISP to allow identification of any sources within that ISP. This attack may be effective given a sufficient number of matchmakers, but widespread collusion falls outside our attack model.

5.2 Anonymity Analysis

A passive adversary who observes a dovetail path segment during construction learns the destination of the segment, the preceding AS, and may measure the cost to the source. In our technical report [40], we show how these properties may be used by an eavesdropper to calculate an anonymity set for the source of

a path segment. The size of this set increases as the attacker moves further from the source, but also depends upon the algorithm used to select the segment path. We consider two different algorithms, showing that our *cost window* approach is superior or equal to shortest path selection in all cases. In addition, we present an entropy based assessment of *effective anonymity set size*, utilizing differences between the routing tables of potential sources.

We now discuss the complete set of source and destination identity information available to a passive adversary at each location on a Dovetail path, using both the path construction packet and the construction return packet. Whenever a measurable cost is discussed, this infers that a set of possible identities can be constructed.

Source Identity. The source identity is known to the source ISP. An attacker at each subsequent AS towards the matchmaker (which includes the dovetail node) can use its knowledge of the preceding AS identity, cost from the source, and all subsequent pathlets up to the matchmaker to limit the possible source identities. At the matchmaker itself, for paths of more than three or four hops, the number of possible sources should be quite large. After the matchmaker, the amount of information about the source will be even less.

Destination Identity. The destination identity is known to every AS from the matchmaker to the destination ISP due to the construction request. Any AS on the head segment between the dovetail and the matchmaker, but that does not appear on the data path, has no knowledge of the destination. Between the source and the dovetail, an attacker can measure the cost from the destination to her own AS using the data return path. If the attacker is able to guess which AS on the head segment serves as the dovetail, she can infer cost from the destination to the dovetail.

As intended, locations where the source is easily identified have little information about the destination and vice versa. The dovetail is the closest AS to the source that learns destination identity; it is typically the strongest location for a passive attacker. To avoid elevating the capability of an attacker located at the dovetail AS, we require that this AS only appear on the head segment once. Any other AS that appears twice in a given segment gains no additional information from its second inclusion.

Each segment of the dovetail path serves a purpose in maintaining a particular anonymity property; this should be considered when setting the segment length. The head segment must be long enough to conceal source identity from the dovetail, and the tail segment must be long enough to conceal destination identity from the source ISP. Finally, we note that uniform random selection of the matchmaker, uncorrelated with either the source or destination, is effective in isolating the anonymity properties of our system. An AS on the head segment can identify the matchmaker, but this does not help to identify the destination; an AS on the tail segment may be able to identify the matchmaker, but this does not help to identify the source.

5.3 Response Timing Attacks

The path diversity used to select each segment should hinder an attacker's ability to identify participants from response timing data. Each potential source could have used one of many thousand possible routes to reach the destination, and each of these routes has its own latency distribution. The superposition of these distributions blurs the range of possible response times for a source significantly when compared to shortest path routing and thus makes distinguishing between different sources harder.

5.4 Availability and Integrity Attacks

Violate routing policy. As with pathlets, all forwarding tables entries are valid expressions of the routing policy, and hence it is not possible to construct a path that violates this policy.

Construct arbitrarily long paths. Our packet design constrains the maximum length of both encrypted and unencrypted packet header segments and thus limits the longest path an adversary intending to waste resources can construct.

Overload a matchmaker. A matchmaker could be overloaded by sending a large number of continuation requests, but matchmakers are distributed throughout the network and the effect on clients is minor if the first matchmaker they contact is unavailable.

Overload a routing vnode. Our forwarding operations are simple and intended to operate at the full data rate of a router. Connection construction requires more operations, but a maximum connection rate could be enforced to constrain this resource utilization.

Modify packet contents. Dovetail is a layer 3 protocol and does not provide any protections for the data it is used to carry. In cases where integrity is important, a higher layer protocol should be used to provide authentication.

Discard packet data. If the quality of service provided by a connection drops below some threshold, this would be observed as a failure, for which the recommended remedy is to reconnect over a different path. Paths are constructed by random selection from the available routes, and so this reconnection is likely to remove any intermediate AS discarding data.

6 Evaluation

Our proposal is evaluated primarily by simulation, using a model of the complete Internet at the AS level. In this section, we first introduce our simulation and input data, then discuss the anonymity and cost results for path segments and for complete paths, and conclude by estimating a variety of resource requirements for our system.

6.1 Simulation Scope

Our simulation models a network of ASes, each containing up to three routing vnodes plus host vnodes to represent its end users and matchmaking capability. ASes are connected by pathlets that codify their contractual arrangement; customer, provider, or peer. All pathlets within an AS have a cost of zero and all pathlets between different ASes have a cost of one. We simulate the exchange of routing information at initialization, leading to a unique routing perspective for each AS that contains all routing vnodes but not all pathlets. Separately, we simulate packets at a bit level during a connection, allowing us to test header design to ensure that routers and the matchmaker could correctly run the protocol.

Our Internet topology is derived from the CAIDA *inferred AS relationship dataset* [48]. The dataset contains *sibling* relations, which permitted infinitely long valley-free routes in some circumstances. To avoid optimistic bias, we replaced all sibling relationships with the more restrictive *peer* relationship. This reclassification causes 5.5% of the network to lose complete reachability, so we disallow traffic originating from or terminating at these ASes. We consider each AS without customer ASes to be a service provider for end users and add a host vnode to represent these users. Ideally, we would model the number of users, but accurate ISP customer size data are not available. Rather than risk skewing our conclusions, we restrict ourselves to measuring anonymity based on the number of possible source or destination ISPs, recognizing that some ISPs are far larger than others. We consider a mixture of ASes following the strict and loose valley-free routing policies defined in Section 4.2. Experimentation shows that when all ASes follow a strict valley-free routing policy, the number of routing options is limited, but introducing even a small proportion of loose valley-free ASes leads to far greater diversity. 10% loose valley-free ASes gives a median of 91,000 options for each path, and we use this topology for the remainder of our evaluation. Studies show that strict valley-free routing is not universal today [36], but we acknowledge that our selection of 10% is arbitrary.

6.2 Single Segment Performance

To select a path segment, the source compiles a set of available routes using a modified depth first search. Our implementation limits this set to a maximum cost of 13, based on the longest distance present in the network, and also a maximum of 20,000 routes at each path cost to limit computation. We first select a cost from the set of *available costs* (i.e. costs with at least one route) and then select a random route of this cost. In our technical report [40] we evaluate four selection algorithms that differ in their probability of selecting a given cost. Based on this evaluation we use the *Exponential4* algorithm, which selects longer paths less frequently but never selects a path with a cost under four. The Exponential4 algorithm results in an average cost approximately 25% greater than shortest path routing, and yet it achieves an anonymity set containing over half the network in 98% of the tests.

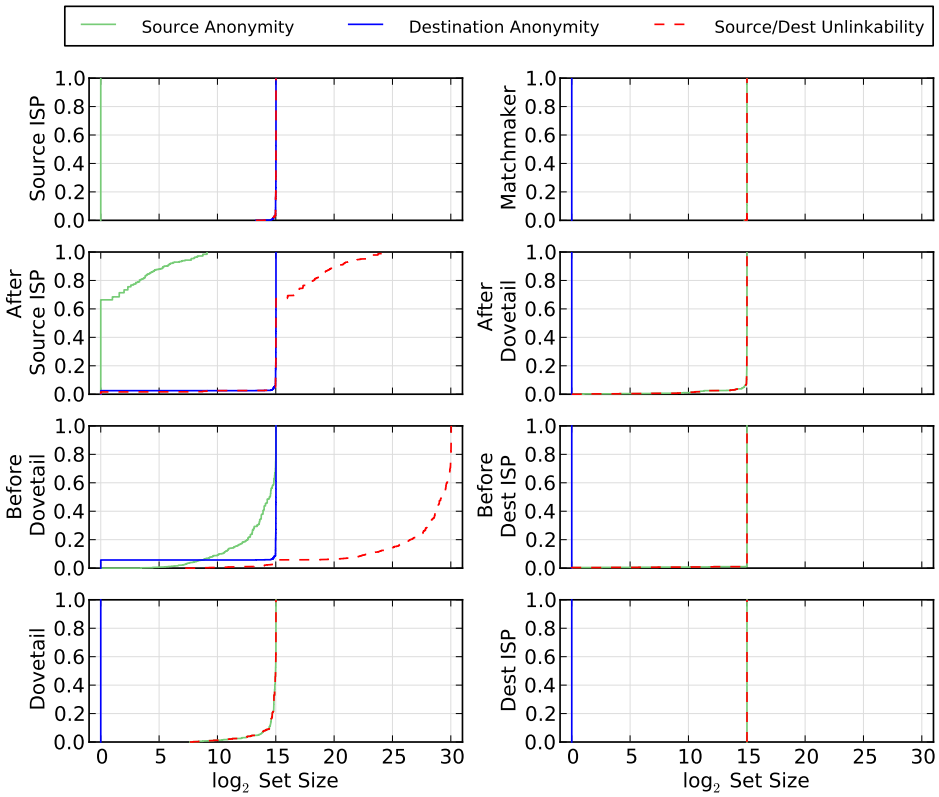

Fig. 3. Source and destination anonymity set size along the complete path

6.3 Complete Path Performance

We now evaluate the anonymity and cost properties of complete paths. Dovetail includes parameters that users can configure to trade performance against anonymity. Our objective here is to demonstrate the anonymity limit of this sliding scale, but many users will prefer a lower setting. The parameter settings we use are:

Dovetail to Matchmaker Cost = Two. Provides strong limits on matchmaker capability without requiring that dovetail and matchmaker are adjacent.

Source to Matchmaker Algorithm = Exponential6. Effectively delivers Exponential4 at the dovetail.

Dovetail to Destination Algorithm = Exponential4. Shown to provide near maximum anonymity [40].

In our experiment, we select source and destination hosts at random and construct a dovetail path between them. The matchmaker generates eight tail

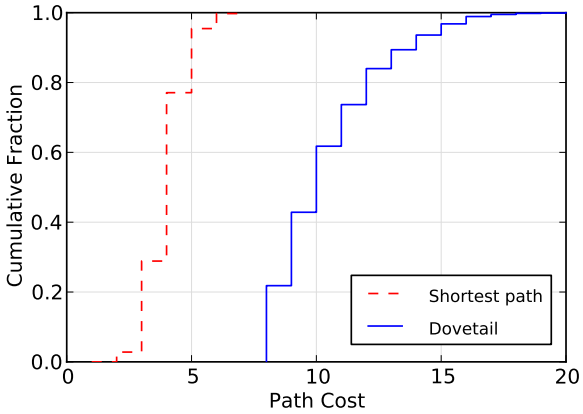

Fig. 4. Cost distribution for complete path

path options and the source selects one from this set. Where possible, the source selects an option that does not reuse a head AS, but in 23% of paths constructed all options required such reuse.[2] We measure the source and destination anonymity set size observable by an attacker at each location in the path. Random selection of a matchmaker decouples the source and destination anonymity sets, and therefore we can also consider the *source-destination unlinkability*, i.e. the number of potential source-destination pairs associated with an observed connection, to be the product of the source and destination anonymity set sizes. Figure 3 presents the distribution of these three properties at a series of key locations along the path, and Figure 4 presents the cost distribution, with the cost of shortest path routing included for comparison with IP and LAP.

Each successive step adds ambiguity to the source identity. At the dovetail AS, source anonymity is approximately equal to network size in 80% of cases. Destination identity is known at the dovetail and all subsequent locations, but locations prior to the dovetail are unable to calculate a meaningful destination identity. No location except the source is able to clearly link source and destination. The AS immediately preceding the dovetail is most likely to be duplicated in head and tail segments, being adjacent to an AS that is always present in both. As illustrated by the destination anonymity for "Before Dovetail", this occurred in 5% of our experiments. The dovetail may partially calculate source identity in around 20% of cases, but this is limited to around one thousand possible source ISPs, each containing many users.

Figure 4 shows that a Dovetail path passes through approximately 2.5 times more ASes than the shortest path routing used in the current Internet. This is a modest penalty when compared to the prevailing option for anonymity today; an anonymous circuit in Tor typically passes through three relays for a total of four IP paths, including six more last-mile connections than a direct path, and incurs additional processing and queuing delays at each relay.

[2] We plan in future work to develop a heuristic to select dovetail vnodes with a lower probability of reuse.

6.4 Resource Utilization

Rather than proposing a near-term solution, we aim to show that privacy is a feasible feature to include in future routing protocol designs. Nevertheless, we now briefly consider a variety of resource requirements to demonstrate that implementation would be feasible.

Host memory utilization. Each Dovetail host must maintain a model of the Internet to generate routes. In the 2012 dataset we use there are 252,666 visible pathlets, of which an average of 22% are known, requiring 680kB.

Router memory utilization. A Dovetail forwarding table scales with the number of local peers and not the total number of Internet prefixes as with BGP. All forwarding information is carried by the packet itself, and so a router need not store any information per connection.

Router latency. The only cryptographic operation required to forward a data packet is a symmetric decryption of one word. This is the same task performed by LAP; Hsiao et al. measure an additional latency of under one microsecond in a software-based implementation of their system [16].

Transmission efficiency. A Dovetail packet must specify a complete path rather than only an endpoint, potentially leading to large headers and low efficiency. The average header length in our experiments is 92 bytes. Given an MTU of 1500 bytes, this represents a 3.5% reduction in payload compared to IPv6. LAP would require a 60 byte header.

7 Conclusion

In this paper we presented Dovetail, a next-generation Internet routing protocol, and have demonstrated that it provides a workable solution for anonymity at the network layer. The overhead is approximately 2.5 times that of shortest path routing when configured to provide near complete anonymity against our chosen attacker, and we include mechanisms to exchange anonymity for performance. We have demonstrated key aspects of the feasibility and effectiveness of this direction and hope this this motivates serious consideration of privacy as a requirement in the development of other next-generation routing protocols.

Acknowledgements. We thank our shepherd, Amir Houmansadr, and numerous anonymous reviewers for their help in improving the paper. This material is based upon work supported by the National Science Foundation under CAREER Grant No. CNS-0954133.

References

1. Reiter, M., Rubin, A.: Crowds: Anonymity for web transactions. ACM TISSEC (1998)

2. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: USENIX Security. (2004)
3. The Tor Project, Inc: Tor metrics portal: Users. https://metrics.torproject.org/users.html Accessed: 2014-02-11.
4. Paul, S., Pan, J., Jain, R.: Architectures for the future networks and the next generation internet: A survey. Computer Communications (2011)
5. The National Science Foundation: NSF NeTS FIND initiative. http://www.nets-find.net/index.php Accessed: 2014-02-11.
6. CORDIS: FIRE home page. http://cordis.europa.eu/fp7/ict/fire/home_en.html Accessed: 2014-02-11.
7. National Institute of Information and Communications Technology: "AKARI" architecture design project for new generation network. http://www.nict.go.jp/en/photonic_nw/archi/akari/akari-top_e.html Accessed: 2014-02-11.
8. Papadopoulos, F., Krioukov, D., Bogua, M., Vahdat, A.: Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces. In: IEEE INFOCOM. (2010)
9. Bhattacharjee, B., Calvert, K., Griffioen, J., Spring, N., Sterbenz, J.P.: Postmodern internetwork architecture. NSF Nets FIND Initiative (2006)
10. Godfrey, P.B., Ganichev, I., Shenker, S., Stoica, I.: Pathlet routing. In: ACM SIGCOMM. (2009)
11. Farinacci, D., Lewis, D., Meyer, D., Fuller, V.: The locator/ID separation protocol (LISP). RFC 6830 (2013)
12. Yang, X., Wetherall, D.: Source selectable path diversity via routing deflections. ACM SIGCOMM Computer Communication Review (2006)
13. Yang, X.: NIRA: A new internet routing architecture. In: ACM SIGCOMM FDNA. (2003)
14. Zhang, X., Hsiao, H.C., Hasker, G., Chan, H., Perrig, A., Andersen, D.G.: SCION: Scalability, control, and isolation on next-generation networks. In: IEEE S&P. (2011)
15. Falk, A.: GENI at a glance. http://www.geni.net/wp-content/uploads/2011/06/GENI-at-a-Glance-1Jun2011.pdf (2011)
16. Hsiao, H.C., Kim, T.J., Perrig, A., Yamada, A., Nelson, S.C., Gruteser, M., Meng, W.: LAP: Lightweight anonymity and privacy. In: IEEE S&P. (2012)
17. Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf (2010) v0.34.
18. Kaufman, C., Hoffman, P., Nir, Y., Eronen, P.: Internet Key Exchange Protocol Version 2 (IKEv2). RFC 5996 (Proposed Standard) (September 2010) Updated by RFCs 5998, 6989.
19. Eckersley, P.: How unique is your web browser? In: PETS. (2010)
20. Soltani, A., Canty, S., Mayo, Q., Thomas, L., Hoofnagle, C.J.: Flash cookies and privacy. In: SSRN eLibrary. (2009)
21. Acquisti, A., Dingledine, R., Syverson, P.: On the economics of anonymity. In: FC. (2003)
22. Dingledine, R., Murdoch, S.J.: Performance improvements on Tor or, why Tor is slow and what we're going to do about it. http://www.torproject.org/press/presskit/2009-03-11-performance.pdf (2009)
23. Jansen, R., Johnson, A., Syverson, P.: LIRA: Lightweight Incentivized Routing for Anonymity. In: NDSS. (2013)
24. Dischinger, M., Haeberlen, A., Gummadi, K.P., Saroiu, S.: Characterizing residential broadband networks. In: ACM SIGCOMM IMC. (2007)

25. Levine, B.N., Reiter, M.K., Wang, C., Wright, M.: Timing attacks in low-latency mix systems. In: Proc. Financial Cryptography. (2004) 251–265
26. Houmansadr, A., Kiyavash, N., Borisov, N.: RAINBOW: A robust and invisible non-blind watermark for network flows. In: NDSS. (2009)
27. Chen, S., Wang, X., Jajodia, S.: On the anonymity and traceability of peer-to-peer voip calls. Network, IEEE **20**(5) (2006) 32–37
28. Reimer, J.: Your ISP may be selling your web clicks. http://arstechnica.com/tech-policy/2007/03/your-isp-may-be-selling-your-web-clicks/ (2007)
29. Dampier, P.: 'Cable ONE spied on customers' alleges federal class action lawsuit. http://stopthecap.com/2010/02/08/cable-one-spied-on-customers-alleges-federal-class-action-lawsuit (2012)
30. Syverson, P.: Why I'm not an entropist. In: Seventeenth International Workshop on Security Protocols., Springer (2009)
31. Murdoch, S.J., Zieliński, P.: Sampled traffic analysis by Internet-exchange-level adversaries. In: PETS. (2007)
32. Boyan, J.: The anonymizer. Computer-Mediated Communication Magazine (1997)
33. Panchenko, A., Pimenidis, L., Renner, J.: Performance analysis of anonymous communication channels provided by Tor. In: ARES. (2008)
34. DiBenedetto, S., Gasti, P., Tsudik, G., Uzun, E.: ANDaNA: Anonymous named data networking application. In: NDSS. (2013)
35. Gao, L.: On inferring autonomous system relationships in the internet. IEEE/ACM ToN (2001)
36. Giotsas, V., Zhou, S.: Valley-free violation in internet routing-analysis based on BGP community data. In: IEEE ICC. (2012)
37. Ryan, P.S., Gerson, J.: A primer on Internet exchange points for policymakers and non-engineers. http://ssrn.com/abstract=2128103 (Aug. 2012)
38. Lodhi, A., Dhamdhere, A., Dovrolis, C.: Open peering by Internet transit providers: Peer preference or peer pressure? In: Proc. IEEE INFOCOM. (2014)
39. Rekhter, Y., Li, T., Hares, S.: A border gateway protocol 4 (BGP-4). RFC 4271 (2006)
40. Sankey, J., Wright, M.: Dovetail: Stronger anonymity in next-generation internet routing. http://arxiv.org/abs/1405.0351 (April. 2014)
41. Borisov, N., Danezis, G., Mittal, P., Tabriz, P.: Denial of service or denial of security? In: CCS. (2007)
42. Wright, M.K., Adler, M., Levine, B.N., Shields, C.: Passive-logging attacks against anonymous communications systems. ACM Transactions on Information and System Security (TISSEC) **11**(2) (2008)
43. Chen, S., Wang, R., Wang, X., Zhang, K.: Side-channel leaks in web applications: A reality today, a challenge tomorrow. In: IEEE S&P. (2010)
44. Mittal, P., Khurshid, A., Juen, J., Caesar, M., Borisov, N.: Stealthy traffic analysis of low-latency anonymous communication using throughput fingerprinting. In: ACM CCS. (2011)
45. Hopper, N., Vasserman, E.Y., Chan-Tin, E.: How much anonymity does network latency leak? In: ACM CCS. (2007)
46. Murdoch, S.J., Danezis, G.: Low-cost traffic analysis of Tor. In: IEEE S&P. (2005)
47. Evans, N., Dingledine, R., Grothoff, C.: A practical congestion attack on Tor using long paths. In: USENIX Security. (2009)
48. CAIDA: The CAIDA UCSD inferred AS relationships - 20120601. http://www.caida.org/data/active/as-relationships/index.xml (2012)