ShadowWalker: Peer-to-peer Anonymous Communication Using Redundant Structured Topologies

Prateek Mittal
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
mittal2@illinois.edu

Nikita Borisov
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
nikita@illinois.edu

ABSTRACT

Peer-to-peer approaches to anonymous communication promise to eliminate the scalability concerns and central vulnerability points of current networks such as Tor. However, the P2P setting introduces many new opportunities for attack, and previous designs do not provide an adequate level of anonymity. We propose ShadowWalker: a new low-latency P2P anonymous communication system, based on a random walk over a redundant structured topology. We base our design on *shadows* that redundantly check and certify neighbor information; these certifications enable nodes to perform random walks over the structured topology while avoiding route capture and other attacks.

We analytically calculate the anonymity provided by ShadowWalker and show that it performs well for moderate levels of attackers, and is much better than the state of the art. We also design an extension that improves forwarding performance at a slight anonymity cost, while at the same time protecting against selective DoS attacks. We show that our system has manageable overhead and can handle moderate churn, making it an attractive new design for P2P anonymous communication.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Security

Keywords

Anonymity, peer-to-peer, random walks

1. INTRODUCTION

Anonymous communication is a key privacy enhancing technology, and is gaining widespread popularity in an era of pervasive surveillance [49]. Anonymous communication hides the identity of communication partners from third parties, or hides user identity from the remote party. The Tor network [16], deployed in 2003, now serves hundreds of thousands of users [25] and carries terabytes of traffic a day [35]. Originally an experimental network used by privacy enthusiasts, it is now entering mainstream use; a recent attack showed a number of foreign consulates were using Tor to avoid surveillance by their host countries [19].

The capacity of Tor is already strained and to support a growing population a peer-to-peer approach will likely be necessary, as P2P networks allow the network capacity to scale with the number of users. Indeed, several proposals for peer-to-peer anonymous communication have been put forward [18,28,33,38]. However, several recent results [5,29,47] have shown that even the best proposed systems are vulnerable to attacks on anonymity, motivating a new approach for P2P anonymous communication.

We propose a low-latency peer-to-peer anonymous communication system that is based on a random walk over redundant structured topologies. Our main idea is the creation of *shadow nodes*, which redundantly verify the correctness of a given nodes' routing table and certify it as correct. Such certificates can then be used to check the steps of a random walk; by using certificates rather than online checks, we can avoid information leak attacks [29]. We show that our design is effectively able to prevent route capture attacks by employing a small number of shadows per node. We also analytically model the effects of a restricted topology on the anonymity of the system and show that, with an appropriate choice of an underlying topology, we can mitigate this effect and achieve strong anonymity. In particular, the anonymity levels achieved by our system are much higher than those of Salsa [33] when 20% of all nodes are compromised.

We present an extension to our system that improves anonymous communication performance at the cost of slightly weakening the anonymity protection. This extension should result in latency and bandwidth constraints similar to those achieved by Tor [16]. It also provides an effective defense against the selective denial-of-service attack on anonymous systems [5]. We also verified our analytic model with the help of simulations. We show that our system has manageable communication and computation overheads, and is able to handle a moderate amount of churn in the network. As such, it presents a promising new direction for peer-to-peer anonymous communication.

The paper is organized as follows. In Section 2, we give an overview of anonymous communication and motivate the

CCS'09, November 9–13, 2009, Chicago, Illinois, USA.

need for a new peer-to-peer approach. We propose the ShadowWalker scheme based on a redundant structured topology in Section 3 and analytically evaluate the anonymity provided by our scheme in Section 4. We describe our experimental results in Section 5 and the related work in Section 6. Finally, we present our concluding remarks in Section 7.

2. BACKGROUND

In this section, we present a brief overview of anonymous communication. We discuss the state of art in peer-to-peer anonymous communication systems and motivate the need for a new design. We also describe our threat model.

2.1 Low-Latency Anonymous Communication Systems

Anonymous communication systems can be classified into low-latency and high-latency systems. High-latency systems like Mixminion [10] and Mixmaster [30] are designed to be secure even against a powerful global passive adversary; however, the message transmission times for such systems are typically of the order of several hours. This makes them unsuitable for use in applications involving interactive traffic, such as web browsing and instant messaging. Our goal is to design a low-latency P2P anonymous system, so we will focus our discussion on low-latency designs.

Tor [16] is a popular low-latency anonymous communication system. Users (clients) download a list of servers from central directory authorities and build anonymous paths using onion routing [46]. There are several problems with Tor's architecture. First, the reliance on central directory authorities makes them an attractive target for the attackers. Second, Tor serves hundreds of thousands of users and the use of a relatively small number of servers to build anonymous paths becomes a performance bottleneck. Finally, Tor requires all users to maintain a global view of all the servers. As the number of servers increases, maintaining a global view of the system becomes costly, since churn will cause frequent updates and large bandwidth overhead.

In order to address these problems, a peer-to-peer architecture will likely be necessary. However, peer-to-peer networks present new challenges to anonymity, one of which is the ability to locate relays for anonymous traffic.

2.2 Peer-to-Peer Anonymous Communication

Several designs for peer-to-peer low latency anonymous communication have been proposed. They can be broadly classified into two categories.

2.2.1 Random Paths Using Lookup

The design of Salsa [33] is similar to Tor, in that a circuit is built by selecting three random nodes in the network and constructing a circuit through them. For scalability reasons, Salsa does not maintain a global view; instead, it uses a specially designed secure lookup operation over a custom distributed hash table (DHT) to locate forwarder nodes. The secure lookups use redundant checks to mitigate potential attacks; these checks are able to limit the bias an adversary can introduce in the lookup, but make Salsa susceptible to information leak attacks: attackers can detect a large fraction of lookups and thus infer the path structure [29]. This results in a tradeoff between robustness to active and passive attacks, and even at the optimal point in this tradeoff, Salsa does not provide adequate level of anonymity. Salsa is also vulnerable to a selective denial-of-service (DoS) attack, where nodes try to deny service for circuits that they cannot compromise [5]. Selective DoS presents a significant problem for Salsa, violating anonymity guarantees when a moderate number of nodes are compromised.

AP3 [28] has a similar structure where paths are built by selecting random relays using a secure lookup mechanism [6]. The design of AP3 is more similar to Crowds [37] than to Tor, with paths being formed by performing a stochastic expected-length random walk. The stochastic nature of AP3 makes it difficult for a rogue node to decide whether its preceding hop is the initiator or simply a relay in the path; however, for low-latency communication, timing attacks may make this decision simpler. Similar to Salsa, the secure lookup used in AP3 reveals a lot of information about the lookup initiator, and makes the user vulnerable to passive information leak attacks [29].

2.2.2 Random Walks on Restricted Topologies

An alternate approach is to connect relays into a restricted (non-clique) topology and construct circuits along paths in this topology. For example, in Tarzan [18], each node has a small set of *mimics*, and all circuits must be created on links between mimics. The use of a restricted topology has the advantage that the local view at each hop is sufficient to extend the circuit. They also provide an opportunity for cover traffic to be sent along all the links in the restricted topology, something that would be infeasible for the full clique topology even of the current size of Tor, let alone much larger P2P networks of the future.

Though communication in Tarzan is carried out over links between mimics, to be able to verify that paths are constructed correctly, each node needs to maintain a global view of the system, updated using a gossip protocol. This limits Tarzan to networks of about 10 000 or fewer nodes. MorphMix [38] was designed to eliminate such scaling constraints by creating a randomized, unstructured overlay between relays, with circuits built on paths along the overlay. MorphMix faced a similar challenge in needing to trust a node to correctly specify its neighbors when extending a circuit. Instead of maintaining a global view, MorphMix designed a mechanism involving witness nodes and a collusion detection mechanism to verify neighbor information. However, the collusion detection mechanism can be circumvented by a set of colluding adversaries who model the internal state of each node, thus violating anonymity guarantees [47].

Our design seeks to combine the best properties of both of these designs: our system is designed to scale to millions of nodes, as in MorphMix, but we use a structured topology to verify neighbor links and are able to resist collusion attacks.

2.3 Threat Model

Low-latency anonymous communication systems are not designed to to be secure against a global passive adversary. In particular, an adversary who can observe the whole network can use end-to-end timing analysis [24,42,45,55] to link two ends of a circuit. We consider a partial adversary who controls a fraction f of all the nodes in the network. This set of malicious nodes colludes and can launch both passive and active attacks. In terms of the standard terminology introduced by Raymond [36], our adversary is internal, active, and static.

P2P networks are vulnerable to Sybil attacks [17], which

would allow an adversary to attain an f arbitrarily close to 1. In context of secure structured P2P networks, there are two major schools of thought regarding Sybil defense. Castro et al. [6] proposed the use of a trusted authority which issues certificates to nodes, that binds the node identifier with a public key. The authority limits the number of certificates and prevents Sybil attacks. The second school of thought uses some scarce resource to bound the number of Sybil identities. For example, if the adversary has access to a limited number of IP addresses, then allowing one identity per IP address would limit the Sybil attack. The node identifier in this case is considered to be the hash of the IP address. Recently, there has been a new line of research that uses social networks for defense against Sybil attacks [11,12,53,54]. Here, node identifiers could be assigned based on their position in the social graph. Lesueur et al. [23] propose a Sybil proof distributed approach using social networks in which nodes cannot choose their identifiers.

We recognize that all the above defenses have some limitations, but coming up with effective distributed defenses to the problem of Sybil attacks is an open research problem, and not a subject of this paper. Our assumptions are that the fraction of colluding identities $f < 0.2$, and that the node identifiers belonging to the adversary are distributed uniformly at random in the ID space (adversary cannot choose its node identifiers). We do not consider $f \geq 0.2$ because it becomes very challenging to perform secure routing in such a network. We emphasize that the above assumptions are standard assumptions used in secure lookup literature [6, 22, 33], which we will review below.

2.4 Structured Peer-to-Peer Networks

We use structured peer-to-peer topologies, such as Chord [44] or Pastry [39] (also known as distributed hash tables, or DHTs), as a foundation for anonymous peer-to-peer communication. Each node in a structured peer-to-peer topology is assigned a collection of neighbors, also known as fingers. Finger relationships are assigned using a mathematical formula based on node identifiers. This allows the relationships to be verified externally, presenting fewer opportunities for attack. A node maintains a routing table, which consists of the IP addresses and the public keys of its fingers. By default, DHTs are extremely vulnerable to attacks on the lookup mechanism [43, 48]. Attackers can intercept lookup requests and return incorrect results by listing a colluding malicious node as the closest node to a key. Next, we discuss several mechanisms for secure lookup.

2.4.1 Castro et al.'s Secure Lookup [6]

The key ideas in this scheme are a routing failure test and redundant routing. The failure test makes use of the observation that the density of honest nodes is greater than the density of malicious nodes. The idea behind redundant routing is to ensure that multiple copies of messages are sent to the key root via diverse routes. Castro et al. [6] also proposed the use of a constrained routing table, in which each slot can have only a single possible node as a neighbor.

2.4.2 Halo [22] and Salsa [33]

In both the schemes, secure lookups are based on redundant routing. Note that naive approaches to redundant routing do not work well because of *convergence of lookups*. Due to convergence, a few nodes may be able to intercept all the redundant lookups, and subvert the result. Salsa proposes the use of a custom DHT topology such that redundant lookups take diverse paths with high probability. Halo makes use of the observation that to perform a lookup for A, it suffices to lookup the nodes which have A as its finger, and then query them.

The above mechanisms are quite effective at ensuring that lookup returns the actual closest node to the chosen identifier. However, anonymous communication systems that use secure lookups to locate forwarder nodes are susceptible to information leak attacks [29].

3. SHADOWWALKER

To motivate our design, we first briefly describe a simple random walk–based anonymity protocol and discuss the attacks on it. In a random walk–based protocol, an initiator first sets up a circuit with a random finger A. To further extend the circuit, initiator sends A a random index i, and A returns the public key of the finger B corresponding to the index i (i'th entry in the routing table). The initiator can then extend the circuit via A to B. By iterating these steps, a circuit of arbitrary length can be established. The above protocol is susceptible to the following attacks:

Route Capture: An intermediate node A in a circuit may lie when asked about its finger B and return an incorrect public key. Since traffic for B will be forwarded through A, A can give its own public key and then pretend to be B. Further, it can perform the same attack in the subsequent steps, emulating the rest of the hops.

Restricted Topology: The terminus of the random walk in restricted topologies reveals some information about the initiator of the random walk [4,8]. This is because only a subset of the nodes in the network can reach the terminus in a given number of hops. For instance, suppose that the first hop in a two-hop random walk is not compromised, but the second hop is compromised. In this scenario, although the initiator cannot be directly identified, the attacker can be certain that the initiator lies in the set of nodes which have the first hop as fingers. Because of route capture attacks, the random walk can be thought to terminate after encountering the first malicious node (say A). If the walk has traversed i hops so far, then the initiator of the random walk must be within the set of nodes that can reach the previous hop of node A in $i-1$ hops. For fixed-length random walks, the number i can be determined by emulating the rest of the random walk; for randomized-length walks, timing analysis would need to be used to guess i.

3.1 Overview

We now describe our ShadowWalker protocol for peer-to-peer anonymous communication. Our main idea is the creation of *shadow nodes* that redundantly verify the correctness of a given node's neighbor table and certify it as correct. Such certificates can then be used to check the steps of a random walk; by using certificates rather than online checks, we can avoid information leak attacks [29]. We first describe the concept of introducing redundancy into the topology itself, which lies at the heart of our solution. Next, we describe two circuit construction protocols for anonymous communication that perform random walks on redundant structure topologies in a secure manner. Finally we present a secure lookup protocol for routing table maintenance and algorithms to handle node churn.

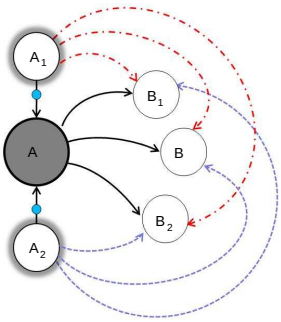

Figure 1: Redundant Structured Topology

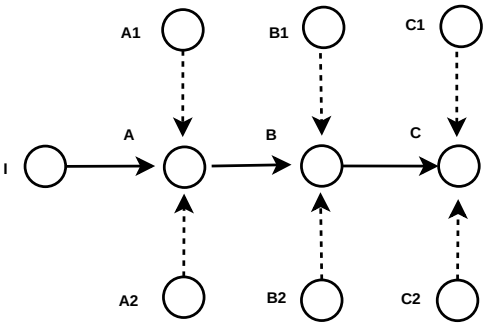

Figure 2: Circuit Construction

3.2 Redundant Structured Topology

We first define the concept of a *shadow*. Each node A has several shadows, and each shadow is required to independently maintain the neighbor information for A. The shadows will provide this information as a way to verify that A is not attempting to perform a route capture attack. For a redundancy parameter r, the shadow nodes of A are denoted as $A_1, \ldots, A_r$. The shadow relationship is a deterministic, verifiable relationship that is calculated by applying a mathematical formula to the node identifier. As an example, for $r = 2$, the shadows for a node A can be considered to be its successor and its predecessor. For a generic r, the shadows for a node A can be considered to be its $\lfloor \frac{r}{2} \rfloor$ predecessors and $\lceil \frac{r}{2} \rceil$ successors in the DHT.

Using the shadow relationship, we can define a transformation to make any P2P topology into a redundant one:
Property 1: In addition to fingers, a node A maintains secure information about the shadow nodes of the fingers. This means that if $A \rightarrow B$ is an edge in the structured topology, $A \rightarrow B_j$ is also an edge in the redundant structured topology, for $j = 1, \ldots, r$ (r shadows of B).
Property 2: If a node A_j is the shadow of node A, it maintains a copy of the fingers (as well as the shadows of the fingers) of A. In other words, if $A \rightarrow B$ is an edge in the structured topology, then $A_j \rightarrow B$ and $A_j \rightarrow B_k$ are also edges in the redundant structured topology, for $j = 1, \ldots, r$ and $k = 1, \ldots, r$.

Figure 1 depicts the transformation of an edge $A \rightarrow B$ into a redundant structured topology with redundancy parameter $r = 2$. Danezis [8] analyzed the use of random paths along a restricted topology for mix networks and proposed the use of topologies with high expansion so that the route length necessary to provide maximal anonymity grows only logarithmically in the number of nodes in the network. Borisov [4] analyzed random walks on structured P2P topologies and proposed the use of the de Bruijn [14] topology to provide anonymity with small path lengths. We use the de Bruijn topology in our design. Note that nodes must be able to maintain the links in a redundant structured topology securely, as described later in Section 3.6.

3.3 Circuit Construction

We use the shadows of a node A to verify the information reported by A during circuit construction. Note that an initiator I cannot contact the shadows directly, since the shadows would learn that it was building a circuit through A. I could use the circuit it has established with A to communicate with A_j, similar to how MorphMix contacts its witness nodes. But this still lets the node A_j know that a circuit is being built through node A.

We can completely avoid this information leak by having each shadow A_j digitally sign its view of the routing table of A and transmit the signature to A. Since the initiator knows the public key of all the shadows (by Property 1), it can verify the signatures without having to contact the shadows at all. Thus, we are able to redundantly check the information provided by A without contacting any other node. We now describe our secure random walk protocol based on redundant structured topologies. Figure 3 shows the pseudocode for our protocol. The initiator I first establishes a circuit to a random finger A. Next, it queries node A for a finger B with random index i (i'th entry in the routing table). A returns the following information to I.
1. IP address and public key of B, and B_k for $k = 1..r$
2. Signatures about the above information from A_j, $j = 1..r$
The initiator I then verifies that signatures of all A_j are correct. Note that since A is a finger of I, A_j are also maintained by I (Property 1). Thus I knows about the public keys of all A_j and can verify the signatures. If the signatures are correct, I can extend the circuit to node B. Now, I can query B for finger C with a random index i', verify it using signatures from B_k and repeat the process. The above example is illustrated in Figure 2. If the signatures do not match, the circuit construction is aborted.

3.4 Using Shorter Circuits

Relaying an interactive stream over 5 or 6 nodes may be expensive; we propose a modification to our protocol where the initiator uses only the last two hops in the random walk to relay traffic. In essence, we use the random walk as an anonymous peer discovery protocol.

Let us consider our modification to the protocol: a node performs a secure l-hop random walk, and then uses the last two hops for anonymous communication, by building a circuit directly to the second to last hop and then extending it to the last hop.[1] Using only the last two hops will improve the system performance as compared to using all l hops for anonymous communication, at the cost of a slight loss of anonymity. Viewed from another perspective, our extension improves anonymity as compared to an 2-hop random walk. In general, if the initiator is interested in building a circuit of length k, it can increase anonymity by performing a l-hop

[1]If the initiator is unable to connect to the second to last hop because of non-transitive connectivity, the circuit construction is aborted.

$I.circuit\_setup(l)$

Let A be a random finger of I
Let A_j be the shadows of A, $\forall j = 1..r$
Let $Pub(A_j)$ be the public key of A_j, $\forall j = 1..r$
Create circuit between I and A
for $count = 1$ to $l - 1$ **do**
  Let B be a random finger of A with index i
  Let $Pub(B)$ be the public key of B
  /* The random finger is chosen by I*/
  Let B_k be the shadows of B, $\forall k = 1..r$
  Let $Pub(B_k)$ be the public key of B_k, $\forall k = 1..r$
  Let $Signature_j$ be the signature given by A_j for A's routing state.
  I obtains $B, Pub(B)$, all $B_k, Pub(B_k)$, and all $Signature_j$ from A via the established circuit.
  if $B, Pub(B)$, and all $B_k, Pub(B_k)$ are verified by all $Signature_j$ **then**
    extend circuit to B
    $A = B$
    $A_j = B_j$, $\forall j = 1..r$
    $Pub(A_j) = Pub(B_j)$, $\forall j = 1..r$
  else
    abort
  end if
end for

Figure 3: The pseudocode for circuit establishment of length l.

random walk for $l > k$, and then use only the last k hops for anonymous communication. (As long as $l < \log_d N$, since beyond that point, longer random walks provide a limited improvement of anonymity [4]. Here d denotes the average node degree in the topology.)

3.5 Using Merkle Hash Trees

Our circuit construction protocol requires that a node obtains signatures for its routing state from its shadow nodes. We can do this efficiently by creating a Merkle hash tree [27] over the set of fingers and have A_j sign the root of the tree. Then when queried about a finger B, A can send the signature on the root along with $\log_2 d$ hashes to I, proving that B was part of the Merkle hash tree signed by A_j.

3.6 Secure Lookup

In Section 2, we described techniques for secure lookups like Halo [22] and Castro et al. [6], which are effective at ensuring that a lookup returns the actual closest node to a chosen identifier. However, in the context of redundant structured topologies, these mechanisms are not very efficient. For instance, in a redundant structured topology, a node needs to maintain shadows of its fingers. To achieve this, the above lookup protocols need to be invoked multiple times for each shadow node, the overhead for which is significant. We propose a secure lookup protocol that is specifically tailored for redundant structured topologies.

Say a node I wishes to securely lookup an identifier ID. Let A be the closest preceding node for ID in the finger table of I. Following the iterative routing strategy, I will query A for its finger, B, which is the closest preceding node for ID. Since I also knows all of the shadows of A, I can verify this information with them. In this way, I can learn the correct identity of B, as well all of its shadows. It can now proceed iteratively, asking B and its shadows for the closest preceding finger for ID. Note that as long as one node among A and its shadows is honest, I will learn the true identity of B; in case of conflicting answers, I should pick the closest one to ID.[2] Thus, a lookup is successful if there is at least a single honest node in each step of the lookup.

An important consequence of our secure lookup protocol is that along with the node corresponding to the chosen ID, its shadows are returned as well! This significantly reduces the communication overhead of our protocol because it obviates the need for performing multiple secure lookups for the shadows of fingers.

3.7 Handling Churn

Handling node churn is a major issue in peer-to-peer systems. Existing DHT designs like Chord have developed algorithms that provide robustness guarantees in presence of churn. A *stabilization protocol* is used periodically to ensure that the information about new nodes is propagated to the other nodes in its neighborhood. Periodically, nodes perform a lookup for chosen identifiers to keep their finger tables up to date. A successor list is also maintained to handle the case of node failures. We refer the reader to [44] for a detailed description of how Chord handles churn.

Now, to accommodate a redundant structured topology, the following changes are to be made:
1. A node periodically performs secure lookups to determine the identity of nodes (say the set S) for which it is the shadow.
2. A node periodically performs secure lookups for the fingers of the nodes in the set S.

This above steps suffice to maintain a redundant structured topology because secure lookups return the shadows of the fingers as well. Moreover, for the purpose of anonymous communication, a node also periodically sends signatures to nodes in the set S over their respective routing states.

4. ANONYMITY EVALUATION

4.1 Anonymity Metric

Low-latency anonymity systems are often studied from the point of view of *path compromise* attacks by counting the fraction of compromised circuits. This metric shows whether attackers are able to identify the initiator of a circuit or not. However, in P2P systems, there may be observations that reveal some information about the initiator even when complete identification is impossible. Therefore, rather than using the binary concept of path compromise, we use the entropy-based anonymity metric [15, 41]. This metric considers the *distribution* of potential initiators of a circuit, as computed by attackers, and computes its entropy:

$$H(I) = -\sum_i p_i \log_2 p_i \quad (1)$$

where p_i is the probability that the node i was the initiator of the circuit. Note that a colluding set of attackers can launch

[2]Note that for an anonymous lookup, all nodes must agree for the lookup to proceed. In the non-anonymous case, however, I can verify the existence of B directly, preventing attackers from responding with fake nodes.

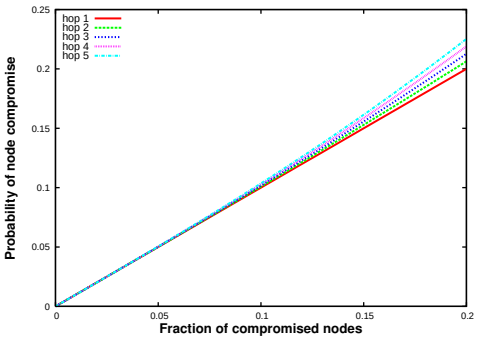

Figure 4: $P(k$'th hop is compromised$)$

a variety of attacks in order to infer the initiator of the circuit. Under some observation o, we can compute the probability distribution given o and compute the corresponding entropy $H(I|o)$. To model the entropy of the system as a whole, we compute a weighted average of the entropy for each observation (including the null observation):

$$H(I|O) = \sum_o P(o)H(I|o) \tag{2}$$

where $P(o)$ is the probability of the observation o occurring, and O is the set of all observations. This is also known as the conditional entropy of I based on observing O.

4.2 Circuit Construction

Our protocol is subject to the following attacks:

Route Capture Attacks: A single malicious intermediate node cannot launch route capture attacks, because its information is verified by its shadows. However, if an intermediate node and all of its shadows are compromised, they can launch a route capture attack by returning colluding malicious nodes as next hops, or by modifying the public keys of the remaining hops to emulate them. This means that if an intermediate node in the circuit and all of its shadows are are malicious, then the remaining nodes in the circuit are also malicious. Thus the initiator anonymity is compromised if the first node in the circuit and all its shadows are malicious.

End-to-End Timing Analysis: Like other low-latency schemes, ShadowWalker is also vulnerable to end-to-end timing analysis, where malicious nodes on both ends of the circuit can use timing correlations of the packets to infer that they are on the same circuit and compromise the initiator anonymity. If the first and the last nodes are compromised, the circuit anonymity is broken.

Restricted Topology Attack: In a simple random walk design, the first malicious node is also the terminus of a random walk, due to the route capture attack. However, in our protocol, the random walk may continue past the first malicious node in case one of its shadows is honest. In particular, if the last node in the circuit is honest, the malicious nodes will not learn the destination of the circuit, and as such will gain nothing by learning (or guessing at) the identity of the initiator. However, if the last node is compromised, then the first malicious node in the circuit can perform timing analysis to establish that the two nodes are on the same circuit. It can then assign probabilities to the initiator as before, by considering all nodes that can reach its previous hop within $i-1$ hops. Thus if the last hop is compromised, and the first malicious node is at the i'th position, then it can infer that the initiator lies in the set of nodes who can reach its previous hop in $i-1$ hops. (i will have to be found out by timing analysis between the first malicious node and the last.)

We first study the effect of route capture attacks by modeling the *sampling bias*. We can think of an k-hop random walk as sampling a node that is k hops away from the initiator. If the walk proceeded undisturbed, then the probability that this sampled node would be malicious would be f. However, the route capture attack introduces a bias into this sampling, such that the longer the random walk, the larger the possibility of the route being captured and thus the last node being compromised. We now compute the bias we can expect when sampling nodes using a k-hop random walk. The k'th hop will be definitely malicious if any of the first $k-1$ stages are able to launch a route capture attack. The probability of launching route capture is given by $1-\left(1-f^{r+1}\right)^{k-1}$. If the attacker is unable to launch the route capture attack in the first $k-1$ hops, then the k'th hop is malicious with probability f. We can now compute the probability that the k'th hop is compromised as follows:

$$\begin{aligned} P(k\text{'th hop is compromised}) &= \left(1-\left(1-f^{r+1}\right)^{k-1}\right)\cdot 1 \\ &\quad + \left(1-f^{r+1}\right)^{k-1}\cdot f \end{aligned} \tag{3}$$

Figure 4 shows the probability that the k'th hop is compromised, for $r=2$. We can see the cascading effect due to route capture attacks: as the random walk length is extended, the probability that the next hop is compromised becomes higher. Note that there is hardly any sampling bias for $f<0.1$, and even when $f=0.2$, the sampling bias is less than 3% for 5 hops. Thus, even at a small redundancy level, our protocol mitigates the route capture attack.

We will now quantify the anonymity that our design provides. Let M_i be the event that the first malicious node on the circuit (say A) is at the i'th position, and the last node is also compromised. Under the event M_i, let the entropy in the choice of the initiator be $H(I|M_i)$. Then, the conditional entropy for the simple random walk protocol can be computed as:

$$H = \sum_{i=1}^{l} P(M_i)H(I|M_i) + \left(1-\sum_{i=1}^{l} P(M_i)\right)\log_2 N \tag{4}$$

Let us compute $P(M_i)$. The first malicious node is at the i'th position with probability $(1-f)^{i-1}f$. Given this, we now need to compute the probability of the last node being malicious. The last node will be malicious with probability 1 if the attackers are able to launch the circuit capture attack between stages i to $l-1$ (capture). Otherwise (no capture), the last node is malicious with probability with f. $P(\text{no capture})$ is given by $(1-f^r)\cdot\left(1-f^{r+1}\right)^{l-(i+1)}$. Also, $P(\text{capture}) = 1-P(\text{no capture})$. Thus, we can express $P(M_i)$ as:

$$\begin{aligned} P(M_i) &= f(1-f)^{i-1}\left(P(\text{capture}) + P(\text{no capture})f\right) \\ &= f(1-f)^{i-1}\left(\left(1-(1-f^r)\left(1-f^{r+1}\right)^{l-(i+1)}\right)\right. \\ &\quad \left. + (1-f^r)\left(1-f^{r+1}\right)^{l-(i+1)}f\right) \end{aligned} \tag{5}$$

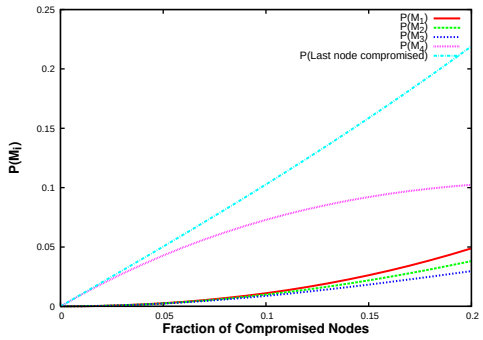

Figure 5: $P(M_i)$: Note that the probability of end to end timing analysis $P(M_1)$ is less than 5% for $f = 0.2$

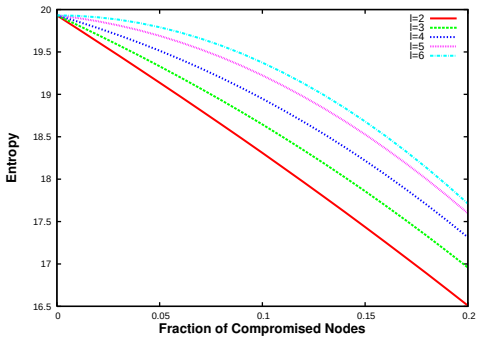

Figure 6: Effect of varying circuit length : Increasing circuit length increases entropy

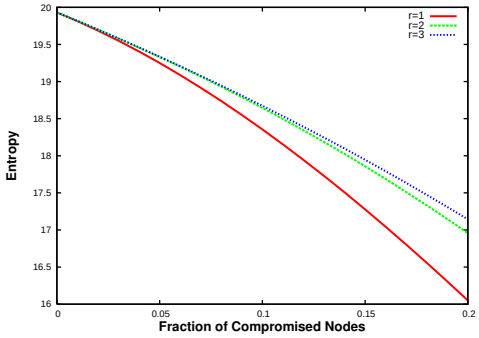

Figure 7: Effect of varying redundancy: There is little advantage in increasing redundancy beyond $r = 2$

Figure 5 shows the values of $P(M_i)$ as a function of f for $l = 4, r = 2$. $P(M_1)$ is the probability of end-to-end timing analysis, and is about 5% for $f = 0.2$. This is close to the current state in Tor, where the probability of end-to-end timing analysis is 4%.[3] Also note that M_l is the dominating event, because unlike other events, it only requires a single node (last node) to be compromised.

We now need to compute $H(I|M_i)$. Note that $H(I|M_i)$ depends on the particular network topology. Though any topology can be used for the random walk, we have considered the de Bruijn [14] topology in this paper because it has optimal mixing properties. In this topology, the expected number of nodes who can reach a particular node in i hops is given by d^i, where d is the average node degree in the topology.[4] We compute $H(I|M_i)$ as follows.

$$H(I|M_i) = \min(\log_2 d^{i-1}, \log_2 N) \quad (6)$$

We can now compute the conditional entropy using equation 4. Figure 6 shows the plot of entropy with varying circuit length for $r = 2$, $N = 1\,000\,000$ and $d = 20$. We can see that increasing circuit length results in a significant increase in entropy. In our secure random walk design, the sampling bias due to route capture is small, and the restricted topology attack dominates. Increasing circuit length mitigates the restricted topology attack and thus increases anonymity. (Note that increasing the circuit length past $l = 6$ will offer no benefit, unless $\log_d N > 6$.[5]) Finally, we study the effect of increasing redundancy. Figure 7 shows the plot of entropy with varying redundancy for $l = 3$. We can see that increasing redundancy beyond 2 does not have any significant benefit. We use $r = 2$ in the remainder of our analysis.

4.3 Using Shorter Circuits

First, consider a two-hop random walk. Let us denote the first hop as A and the second hop as B. If both A, B are malicious, then the initiator anonymity is compromised. When only B is compromised, the initiator can be narrowed to the set of nodes that have A as their fingers. The expected size of this set is quite small (d), resulting in poor anonymity. Also note that the latter event happens frequently, with probability about f, where as both A and B are malicious with probability about f^2.

Now, let us consider our modification to the protocol: a node performs a secure three-hop random walk (A, B, C), and then uses the last two hops (B, C) for anonymous communication, by building a circuit directly to B and then extending it to C. Again, the dominant event is when only C is compromised. Under this event, the attacker can narrow the choice of the initiator to the set of nodes who have B within two hops. The expected size of this set is now d^2. Thus our modification results in an increase in anonymity, while keeping the circuit length constant.

Note that in the anonymity analysis of the modified two hop random walk protocol, the entropy is 0 when the last two nodes are compromised. Thus let us redefine M_i for $(i \leq l-2)$ to be the event such that the first malicious node on the circuit is at the $i'th$ position, the last node is also compromised, but the second last node is honest. We define M_{l-1} as the event that the last two nodes are compromised, regardless of whether any previous nodes were compromised as well. $P(M_{l-1}) = f^2$, and $H(I|M_{l-1}) = 0$, since the initiator contacts the second last node directly. We keep the definition of M_l the same as before; i.e., only the last hop is compromised. For $i \leq l-2$, $P(M_i)$ can be expressed as:

$$P(M_i) = f(1-f)^{i-1}(1-f^r)\left(1-f^{r+1}\right)^{l-2-i}(1-f)f \quad (7)$$

Figure 8 shows the plot of entropy for our modified pro-

[3]This is a slight simplification, as the exact fraction of compromised tunnels will depend on the share of bandwidth and the guard/exit status of compromised nodes

[4]Fingers of fingers do not overlap in a regular de Bruijn topology.

[5]In real networks, the lack of perfect load-balancing will result in somewhat worse mixing, and thus values of $l > \log_d N$ may still make sense.

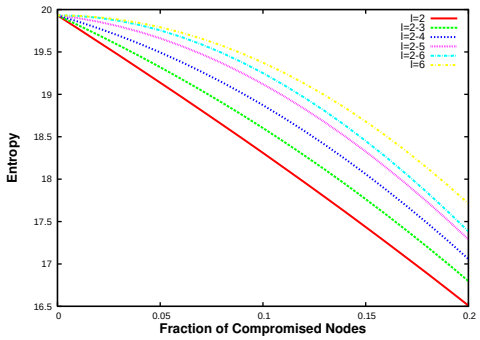

Figure 8: Using last two hops for anonymous communication: Mitigating restricted topology attacks while keeping circuit length constant

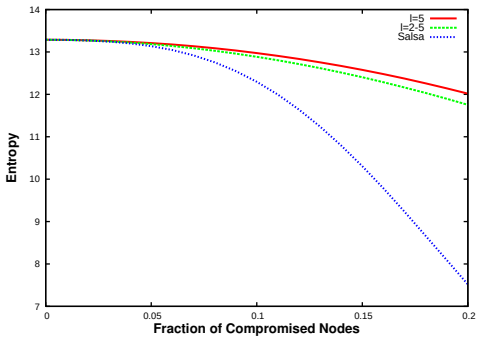

Figure 9: Comparison with Salsa: For $f = 0.2$, our protocol has 4.5 bits more entropy than Salsa

tocol, computed as:

$$H = \sum_{i=1}^{l-2} P(M_i)H(I|M_i) + P(M_l)H(I|M_l) + \left(1 - \sum_{i=1}^{l} P(M_i)\right) \log_2 N \quad (8)$$

Here, $l = 2$–6 refers to our modified protocol where a node performs a 6 hop random walk and then uses only the last two hops for anonymous communication. We can see that that our modification allows a user to derive higher anonymity using longer random walks, but keeping the circuit length constant. Viewed from another perspective, this extension creates a tradeoff between anonymity and performance. Using all l hops for anonymous communication is more secure, but introduces higher latency on the communication and uses more system resources. Using only the last two hops will improve the system performance, at the cost of revealing the identity of the initiator to the second-to-last hop. As can be see in Figure 8, the loss of anonymity is slight: using $l = 2$–6 results in anonymity that is only slightly lower than $l = 6$.

4.4 Comparison with Salsa

We will now compare our ShadowWalker protocol with Salsa [33]. Salsa uses secure lookup as a primitive to build a circuit for anonymous communication, which makes Salsa susceptible to information leak attacks [29]. To compute the effect of active attacks on lookups, we used a simulator developed by the authors of Salsa [32]. The simulator was configured to simulate 1000 topologies, and in each topology, results were averaged over 1000 random lookups. The Salsa architecture divides the identifier space into groups, where the number of groups is denoted by $|G|$. We used the parameters $N = 10,000$ and $|G| = 128$ for the simulation (it is difficult to scale the simulations beyond $10,000$ nodes). Next, we modeled the Salsa path building process as a stochastic activity network in the Möbius framework [7]. Figure 9 compares the anonymity provided by ShadowWalker and Salsa. In our system, we use the degree $d = 13$ and $r = 2$. In the next section, we will see that this translates into an effective degree of 39 ($(r + 1) \cdot d$). This is comparable to the effective degree of Salsa in this configuration, which is 85 ($10000/128+\log_2 128$). We can see that for $f = 0.2$, our protocol using $l = 5$ has an entropy of 12, while Salsa only has an entropy of 7.5. Even our modified protocol which uses only two hops for anonymous communication, gives much better anonymity than Salsa.

4.5 Selective DoS Attack

Recently, Borisov et al. [5] proposed a selective denial-of-service attack on anonymous communication. In this attack, malicious nodes can selectively drop packets in order to shut down any circuits that they are a part of, but which they cannot compromise. Borisov et al. found that selective DoS attack is most effective against peer-to-peer anonymous communication systems, because the circuit construction in P2P systems is complex and may provide many nodes with the opportunity to selectively deny service. Our design is vulnerable to the selective DoS attack in two ways:

Selective DoS by shadows: As a shadow node, a malicious node M may refuse to give signatures to honest nodes, or may give incorrect signatures to honest nodes. This attack will ensure that the honest nodes who have a malicious node as a shadow will never get selected in the random walk as an intermediate node, since the initiator will not be able to verify the neighbor relationships.

Selective DoS during circuit construction: Malicious nodes can also selectively break any circuits that they cannot compromise. Whenever malicious nodes find that they are part of a circuit in which they are unable to infer any information about the initiator, they stop forwarding packets on the circuit, causing a new circuit to be constructed. This attack is similar to the selective-DoS attack on Tor described by Borisov et al.

We can mitigate the first attack by using a symmetric shadow relationship. This means that if node A is a shadow of node B, then node B is a shadow of node A. If a node stops receiving signatures from its shadow, it can reciprocate by no longer certifying the shadow's routing information. As a result, malicious nodes that do not follow the protocol and refuse to provide signatures will themselves be excluded from the circuit construction process. An adversary may decide to sacrifice its nodes, and in this process DoS (atmost) r honest nodes. However, since small redundancy levels of $r = 2, 3$ suffice for the security of our protocol, this strategy does not benefit the adversary.

For the second attack, the best strategy for malicious nodes is to shut down any circuit in which the last node is honest, since there is no hope of compromising it. Thus the only circuits that will be built are those where the last node is compromised, or where all the nodes are honest. The

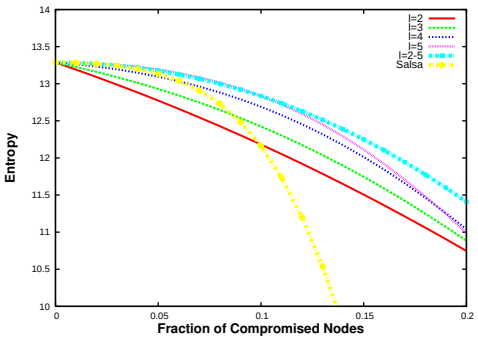

Figure 10: Selective-DoS Attack: Using l =2–6 resists selective DoS attack

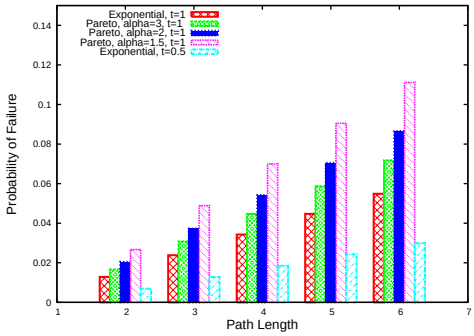

Figure 11: Impact of Churn on Reliability

following equation quantifies the effect of the selective DoS attack on our protocol.

$$H = \sum_{i=1}^{l} \frac{P(M_i)}{\sum_{j=1}^{l} P(M_j) + (1-f)^l} H(I|M_i) + \frac{(1-f)^l}{\sum_{i=1}^{l} P(M_i) + (1-f)^l} \log_2 N \quad (9)$$

Figure 10 plots the entropy for our protocol under the selective DoS attack. There is an interesting tradeoff here. On one hand, increasing circuit length mitigates the restricted topology attack and increases anonymity. On the other hand, increasing circuit length gives more opportunities to the attackers to launch a selective DoS attack. We can see that for small values of f, the former effect dominates, and increasing circuit length increases anonymity. There is a crossover point at about 18% of compromised nodes, when increasing circuit length beyond $l = 4$ becomes counterproductive, because of the selective-DoS attack. We note that our modified protocol, in which the initiator only chooses the last two hops for anonymous communication, provides a good defense against the selective-DoS attack. This is because the intermediate hops do not decide to abort until the circuit construction has reached the last hop. However, at that point, only the second-to-last hop can perform denial-of-service on the circuit. We can see from the figure, that $l = 2$–5 is most resilient to selective-DoS attack. Also note that selective-DoS presents a significant problem for Salsa. Salsa is able to provide only 4 bits of entropy at $f = 0.2$, as compared to about 11.5 bits of entropy for l =2–5.

5. EXPERIMENTAL RESULTS

We implemented our protocol using an event-based simulator in C++ with 1.2KLOC. We consider a WAN setting, where latencies between each pair of nodes are estimated using the King data set [20]. This data set contains measured latencies between Internet domain name servers (DNS) and is highly heterogeneous. The average round trip time (RTT) in the data set is around 182ms and the maximum RTT is around 800ms. To handle churn, we run the stabilization protocol every second. The time period for refreshing fingers and signing certificates is also set to 1 second. We simulate our protocol for $N = 1000$ nodes with a redundancy parameter $r = 2$ and $d = 10$.

Studies have shown that in many popular peer-to-peer networks, the mean value of node uptime is about 60 minutes [3,40]. We considered two widely used synthetic models for node uptime 1) PDF $f(x) = \lambda e^{-\lambda x}$. We set $\lambda = 1/60$. This is an exponential distribution with mean 60 minutes. 2)PDF $f(x) = \frac{ab^a}{(x+b)^{a+1}}$. We set $a = 1.5, 2, 3$ and b fixed so that the distribution had mean 60 minutes. This is a standard Pareto distribution, shifted b units (without the shift, a node would be guaranteed to be up for at least b minutes).

5.1 Communication Overhead

Topology maintenance: As compared to a structured network, the overhead for topology maintenance in our protocol is higher due to the inherent redundancy in topology. The transformation from a structured topology to a redundant structured topology increases the effective node degree from d to $(r+1)^2 d$. (Each finger has r shadows, and each node is a shadow for (around) $r+1$ nodes.) An important consequence of our secure lookup protocol is that along with the node corresponding to the chosen ID, its shadows are returned as well. This significantly reduces the communication overhead of our protocol because it obviates the need for performing multiple secure lookups for the shadows of fingers. The use of our secure lookup protocol reduces the effective node degree to $(r+1) \cdot d$. In the previous section, we had seen that our system provides better anonymity than Salsa with similar effective node degree. For $N = 1000$ nodes and $r = 2$, the mean communication overhead per node was measured to be 5980 bytes/sec.

Circuit Setup: To establish a circuit of length l, the initiator performs l key establishments and rl signature verifications. The corresponding figure for Salsa is $r(l-1)+1$ key establishments and $r^2(l-1)+r$ lookups. The table below shows the mean circuit setup latency. We can see that even for $l = 6$, the circuit setup time is less than 4 seconds. Since we avoid the use of lookups, the circuit setup latency for our protocol is smaller than Salsa.

Mean Circuit Setup Latency (ms)				
l=2	l=3	l=4	l=5	l=6
546	1092	1820	2730	3822

5.2 Reliability of Circuit Construction under Churn

Due to churn, the routing states at different nodes may be inconsistent at times, resulting in different views of the network. This will mean that corresponding signatures by shadow nodes for the routing state of a node A may not be consistent, and our circuit construction protocol may fail.

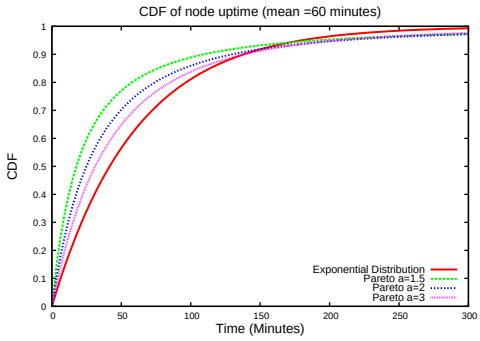

Figure 12: Churn Distributions

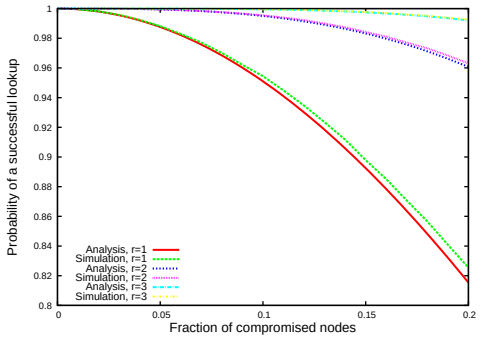

Figure 13: Lookup Security

Figure 11 shows the effect of churn on the reliability of our circuit construction protocol. Let us first consider the exponential distribution for node uptimes. We can see that increasing path lengths increases the probability of failure. This is because there is an higher chance of a node and its shadows having an inconsistent view of the network. For a path length $l = 6$, the probability of failure is about 0.05. Next, observe that the probability of failure increases if we model node churn as a Pareto distribution. Moreover, smaller values of the exponent a lead to higher probabilities of failure. To get some intuition for this, observe that Pareto distributions with smaller exponents a have a longer 'tail' in the CDF, as depicted by Figure 12. This results in a larger number of node arrivals and node departures (even though the mean node uptimes are the same), leading to an decrease in reliability of circuit construction.

We note that reliability of circuit construction can be increased by being more aggressive in topology maintenance (i.e., reducing the time period t for refreshing fingers and signing state). Figure 11 depicts this tradeoff between bandwidth use and reliability of circuit construction. We can see that for exponential distribution of node uptimes, by reducing the time period from $t = 1$ seconds to $t = 0.5$ seconds, the probability of failure has been approximately halved.

5.3 Secure Lookup

A lookup is successful if there is at least a single honest node in each step of the lookup. For a lookup of path length l, the probability that a lookup succeeds can be modeled as:

$$P(\text{secure lookup}) = \left(1 - f^{r+1}\right)^l$$

Figure 13 plots of probability of a successful lookup for varying values of r. For $f = 0.1, r = 2$, the probability of a successful lookup is 0.99. Even when we increase the value of f to $f = 0.2$, the lookup is still successful with probability 0.95. The lookup security improves exponentially with increasing r, because the chance that a node and all its shadows are malicious falls exponentially in r. Thus for $f = 0.2$ and $r = 3$, the lookup succeeds with probability 0.99. Note that for small values of r, the lookup security can also be improved by performing redundant versions of the above lookup.

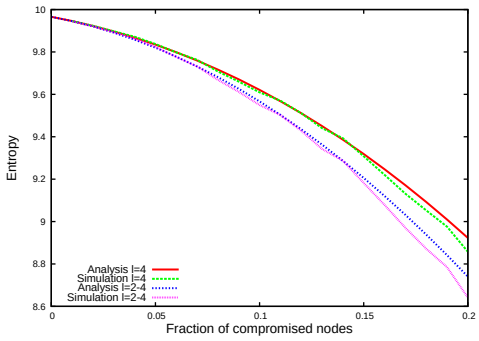

Figure 14: Anonymity

5.4 Anonymity

Finally, we present simulation results for the anonymity provided by ShadowWalker. Using simulations, we have performed a whole system evaluation of ShadowWalker to check for any hidden correlations not captured by our analytic model. Our simulator also captures real world behavior like the effect of irregular topologies, which is not considered in our model. Figure 14 depicts the anonymity provided by ShadowWalker for $l = 4$ and $l = 2 - 4$. We can see that our simulation and analytic results closely match.

6. RELATED WORK

Danezis and Clayton [9] studied attacks on peer discovery and route setup in anonymous peer-to-peer networks. They show that if the attacker learns the subset of nodes known to the initiator, its routes can be fingerprinted unless the initiator knows about a substantial fraction of the network. Danezis and Syverson [13] extend this work to observe that an attacker who learns that certain nodes are *unknown* to the initiator can carry out attacks as well and separate traffic going through a relay node. Both these attacks assume a global passive adversary, but are similar in spirit to the restricted topology attack.

Another attack relevant to P2P systems is the circuit-clogging attack [1, 31], as McLahlan and Hopper [26] observed that, in P2P systems, this attack can reveal the true initiator. They proposed a stochastic fair queueing mechanism to mitigate the attack. We note that the circuit clogging attack is particularly effective against a restricted route topology, since during traceback of the random walk, there are only d possibilities at each step. Our extension of using only the last two hops of an l hop random walk for anonymous communication makes the traceback significantly harder for the adversary, since it is now necessary to measure d^{l-2} hosts.

Several important attacks consider the degradation of ano-

nymity with time. The predecessor attack, originally proposed by Reiter and Rubin [37], has been analyzed in detail by Wright et al. [50, 51]. As applied to our work, the attack notes that eventually a low-anonymity circuit will be constructed. Guard nodes [51] are a defense against predecessor attacks that is used in the current version of Tor [34]. However, the use of guard nodes in P2P systems needs more study; a straightforward implementation would allow attackers to quickly arrive at an effective anonymity set size of d.

Intersection attacks [2,36] work by noting which nodes are active at the time a message is received. These attacks are a particular concern for P2P systems due to the highly dynamic participation of most nodes [52]. The best approaches for combating these attacks are to reduce the perspective on the network that is given to the attackers [21]. However, even with the best defenses, a large fraction of nodes will be able to achieve a near-global view. Our redundant topology exacerbates the problem by increasing the effective node degree. Whether a network that is resilient to intersection attacks can achieve similar levels of anonymity to our design remains an open question.

A variant of intersection attack is applicable on our protocol, where instead of noting the set of active nodes, the adversary can use the probabilistic information about the initiator using the restricted topology attack. This attack would work much faster as compared to the traditional intersection attack. Due to lack of space, we have omitted a complete analysis of this attack. Our results indicate that the de Bruijn topology is able to effectively resist this attack.

7. CONCLUDING REMARKS

We proposed ShadowWalker: a new design for low-latency P2P anonymous communication. ShadowWalker is able to effectively defend against common attacks on peer-to-peer systems and achieve levels of anonymity superior to the state of the art in P2P anonymous communication. In particular, when 20% of all nodes are compromised, ShadowWalker provides 4.5 bits more entropy than Salsa. Moreover, the probability of end to end timing analysis in this case is less than 5%, which is close to the ideal scenario as in Tor, where the probability of end to end timing analysis is 4%.

Our system presents several tradeoffs between anonymity and performance overhead. We have demonstrated points along these tradeoffs that have manageable computation and communication overheads while providing strong anonymity guarantees. ShadowWalker is also able to handle moderate churn in the network. As such, it presents a promising new direction for P2P anonymous communication.

We note that our redundant structured topology design has benefits that may extend beyond anonymity systems, which we shall study in future work. We shall also extend our design to incorporate the issues of heterogeneous node bandwidth and exit policies.

Acknowledgments

We are very grateful to Matthew Wright, Nick Mathewson and the anonymous reviewers for their invaluable feedback on the draft manuscript. We would also like to thank George Danezis, Paul Syverson and Steven Murdoch for helpful discussions about the research. This research was supported in part by NSF grant CNS–0627671.

8. REFERENCES

[1] A. Back, U. Möller, and A. Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In *Proceedings of the IH*, 2001.

[2] O. Berthold, H. Federrath, and M. Köhntopp. Project anonymity and unobservability in the Internet. In *Proceedings of CFP*, 2000.

[3] R. Bhagwan, S. Savage, and G. Voelker. Understanding availability. *Proceedings of IPTPS*, pages 256–267, 2003.

[4] N. Borisov. *Anonymous routing in structured peer-to-peer overlays*. PhD thesis, University of California at Berkeley, Berkeley, CA, USA, 2005.

[5] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz. Denial of service or denial of security? How attacks on reliability can compromise anonymity. In *Proceedings of CCS*, October 2007.

[6] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *Proceedings of OSDI*, December 2002.

[7] D. Daly, D. D. Deavours, J. M. Doyle, P. G. Webster, and W. H. Sanders. Möbius: An extensible tool for performance and dependability modeling. In B. R. Haverkort, H. C. Bohnenkamp, and C. U. Smith, editors, *Computer Performance Evaluation: Modelling Techniques and Tools*, volume 1786, pages 332–336, Schaumburg, IL, Mar. 2000. Springer.

[8] G. Danezis. Mix-networks with restricted routes. In R. Dingledine, editor, *Proceedings of PET*, pages 1–17. Springer-Verlag, LNCS 2760, March 2003.

[9] G. Danezis and R. Clayton. Route Fingerprinting in Anonymous Communications. *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, pages 69–72, 2006.

[10] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proceedings of S & P*, pages 2–15, May 2003.

[11] G. Danezis, C. Lesniewski-Laas, M. F. Kaashoek, and R. J. Anderson. Sybil-resistant dht routing. In *Proceedings of ESORICS*, pages 305–318, 2005.

[12] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Proceedings of NDSS*, 2009.

[13] G. Danezis and P. Syverson. Bridging and fingerprinting: Epistemic attacks on route selection. In *Proceedings of PET*, Leuven, Belgium, July 2008.

[14] N. de Bruijn. A combinatorial problem. *Koninklijke Nederlandse Akademie van Wetenschappen*, 49, 1946.

[15] C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In R. Dingledine and P. Syverson, editors, *Proceedings of PET*, San Diego, CA, April 2002. Springer-Verlag, LNCS 2482.

[16] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of USENIX Security Symposium*, August 2004.

[17] J. Douceur. The Sybil Attack. In *Proceedings of IPTPS*, March 2002.

[18] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of CCS*, Washington, DC, November 2002.

[19] D. Goodin. Tor at heart of embassy passwords leak. *The Register*, September 10 2007.

[20] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: estimating latency between arbitrary internet end hosts. *SIGCOMM CCR.*, 32(3):11–11, 2002.

[21] S. Hazel and B. Wiley. Achord: A variant of the Chord lookup service for use in censorship resistant peer-to-peer publishing systems. In *Proceedings of IPTPS*, Cambridge, MA, Mar. 2002.

[22] A. Kapadia and N. Triandopoulos. Halo: High-assurance locate for distributed hash tables. In *Proceedings of NDSS*, pages 61–79, February 2008.

[23] F. Lesueur, L. Me, and V. V. T. Tong. A sybilproof distributed identity management for p2p networks. In *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pages 246–253, 2008.

[24] B. N. Levine, M. K. Reiter, C. Wang, and M. K. Wright. Timing attacks in low-latency mix-based systems. In A. Juels, editor, *Proceedings of FC*, pages 251–265. Springer-Verlag, LNCS 3110, February 2004.

[25] K. Loesing. Measuring the Tor network: Evaluation of client requests to directories. `https://git.torproject.org/checkout/metrics/master/report/dirreq/directory-requests-2009-06-25.pdf`, 2009.

[26] J. McLachlan and N. Hopper. Don't clog the queue: Circuit clogging and mitigation in P2P anonymity schemes. In *Proceedings of FC*, January 2008.

[27] R. Merkle. Protocols for public key cryptosystems. In *Proceedings of S & P*, pages 122–133, 1980.

[28] A. Mislove, G. Oberoi, A. Post, C. Reis, P. Druschel, and D. S. Wallach. AP3: Cooperative, decentralized anonymous communication. In *Proceedings of the ACM SIGOPS European Workshop*, 2004.

[29] P. Mittal and N. Borisov. Information leaks in structured peer-to-peer anonymous communication. In *Proceedings of CCS*, October 2008.

[30] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman. Mixmaster Protocol — Version 2. IETF Internet Draft, July 2003.

[31] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of Tor. In *Proceedings of S & P*, 2005.

[32] A. Nambiar and M. Wright. The Salsa simulator: http://ranger.uta.edu/ mwright/code/salsa-sims.zip.

[33] A. Nambiar and M. Wright. Salsa: a structured approach to large-scale anonymity. In *Proceedings of CCS*, pages 17–26, New York, NY, USA, 2006. ACM.

[34] L. Øverlier and P. Syverson. Locating hidden servers. In *Proceedings of S & P*. IEEE CS, May 2006.

[35] T. T. Project. Torstatus–tor network status. `http://torstatus.kgprog.com/`.

[36] J.-F. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 10–29. Springer-Verlag, LNCS 2009, July 2000.

[37] M. Reiter and A. Rubin. Crowds: Anonymity for web transactions. *ACM TISSEC*, 1(1), June 1998.

[38] M. Rennhard and B. Plattner. Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In *Proceedings of WPES*, Washington, DC, USA, November 2002.

[39] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Middleware*, pages 329–350, November 2001.

[40] S. Saroiu, P. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems. In *Multimedia Computing and Networking*, 2002.

[41] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of PET*, 2002.

[42] V. Shmatikov and M.-H. Wang. Timing analysis in low-latency mix networks: Attacks and defenses. In *Proceedings of ESORICS*, September 2006.

[43] E. Sit and R. Morris. Security considerations for peer-to-peer distributed hash tables. In *IPTPS*, pages 261–269, London, UK, 2002. Springer-Verlag.

[44] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, 2003.

[45] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr. Towards an analysis of onion routing security. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 96–114. Springer-Verlag, LNCS 2009, July 2000.

[46] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. *Security & Privacy, IEEE*, 4-7:44–54, 1997.

[47] P. Tabriz and N. Borisov. Breaking the collusion detection mechanism of MorphMix. In G. Danezis and P. Golle, editors, *Proceedings of PET*, pages 368–384, Cambridge, UK, June 2006. Springer.

[48] D. Wallach. A survey of peer-to-peer security issues. In *International Symposium on Software Security*, Tokyo, Japan, 2002.

[49] C. Williams. Bt admits misleading customers over Phorm experiments. *The Register*, March 17 2008.

[50] M. Wright, M. Adler, B. N. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. In *Proceedings of NDSS*. IEEE, February 2002.

[51] M. Wright, M. Adler, B. N. Levine, and C. Shields. Defending anonymous communication against passive logging attacks. In *Proceedings of S & P*, May 2003.

[52] M. Wright, M. Adler, B. N. Levine, and C. Shields. Passive logging attacks against anonymous communications. *ACM TISSEC*, 11(2), May 2008.

[53] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Proceedings of S & P*, 2008.

[54] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. *SIGCOMM CCR.*, 36(4):267–278, 2006.

[55] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao. On flow correlation attacks and countermeasures in mix networks. In *Proceedings of PET*, 2004.