# An Information-Theoretic Framework for Analyzing Leak of Privacy in Distributed Hash Tables

Souvik Ray
Electrical and Computer Engineering
Iowa State University

Zhao Zhang
Electrical and Computer Engineering
Iowa State University

## Abstract

*An important security issue in DHT-based structured overlay networks is to provide anonymity to the storage nodes. Compromised routing tables in those DHTs leak information about other nodes in the system and therefore compromise privacy. In this paper, we use information theory to build a model to quantify the information leak from compromised routing tables for a given DHT with certain routing geometry and route table size. Based on this model, we have analyzed and compared how existing DHTs perform in face of anonymity attacks. We found that ring-based routing geometry (Chord) performs the best among the studied DHTs with the same routing complexity when no routing optimizations are used. The analysis of the interaction between routing geometries and recipient anonymity will help improve the design of future DHTs which can achieve a balance between routing efficiency and robustness against information leak.*

## 1 Introduction

An important security issue in a DHT (distributed hash table) based P2P storage system for file sharing is to provide recipient anonymity to the storage nodes in the system. In a P2P storage system using DHT, the DHT is used to return the network address of the storage node for a given document key. For example, in DHTs like Chord [13] and CAN [8] the query for a key returns the IP address of the node storing the documents that match that key. AChord [4] added anonymity feature to Chord by modifying "lookup-by-address" to "lookup-by-value". However, AChord is still vulnerable to attacks to the index nodes[1]: If a hacker breaks into an index node, information stored in the node's routing table will be leaked.

The risk of leaking information from DHT routing tables has not been well analyzed in current research. DHTs were primarily designed for routing efficiency and scalability. For example, Chord uses a structured routing geometry such that the search takes up to $O(\log N)$ steps to complete a query. The side effect is that the routing tables contain a substantial amount of information about other nodes in the system. This information can be used by adversaries, if the system is breached, to compromise the anonymity of storage nodes, i.e. the recipient anonymity. A recently proposed design called Neblo [2] uses imprecise routing to enhance recipient anonymity. Agyaat [11] proposes the use of unstructured clouds on top of a structured overlay to hide recipient anonymity (Table 1 shows the previous attempts at enhancing anonymity for structured networks). However, no research has been done to quantify and compare the information leak from routing tables in different DHT designs.

In this study, we build an analytical model to analyze, quantify and compare the leak of privacy from the routing tables in existing DHT designs. There are two primary factors that influence the amount of information contained in a routing table: the type of routing geometry and the size of routing table. Our model gives valuable insight into how different routing geometries will affect the recipient anonymity. It uses entropy to calculate the amount of information leak. Based on this model, we compare information leak in different DHTs when index nodes are breached. We have compared the use of Chord [13], CAN [8], Kademlia [6], Pastry [9] in building a P2P storage system with the same routing complexity, i.e. the number of hops in the routing. Our analytical results show that for the same routing complexity, ring-based DHT (Chord) has the minimum information leak. The general trend is that the state information stored in routing tables increases with routing optimizations, thereby resulting in significant information leak. We observe that Kademlia (XOR routing) has a significant amount of information leak for small overlay sizes. Pastrys (hybrid routing) performance is quite close to that of the ring structure. The hypercube-based routing in CAN has an important side-effect, i.e. of localizing the information loss. We also analyse the effect of routing table size on leak of

[1] It is possible that a node plays the role of an index node and a storage node simultaneously.

information. We believe that our preliminary findings can help us better understand the effect of routing geometry on the state information stored in routing tables which can lead to the development of DHT designs with an optimal balance between routing efficiency and information leak.

The paper is organized as follows. In section 2 we first give some background on recipient anonymity in Distributed Hash Tables and then propose an information-theoretic framework for calculating the information content of routing tables. In section 3 we use this framework to quantify the information leak for different DHT designs. We follow this with a detailed analysis of the interaction between different routing geometries and recipient anonymity, the effect of routing optimizations and the size of routing tables in section 4. We present related work in section 5 and finally conclude this study in section 6.

# 2 Quantifying Information Leak from Routing Tables

Structured overlays like Chord [13], CAN [8], Pastry [9], Kademlia [6], and Viceroy [5] use distributed hashing to store keys at nodes. A node in such a DHT is identified by a tuple <IP address, Identifier>. The information contained in the tuple can be used by an adversary to compromise recipient (storage) anonymity. In the context of a distributed hash table, recipient anonymity is broken when the adversary can generate the mapping between a node's IP address and its identifier range. The objective of the adversary is to generate a map of the system. Such a mapping between a node's IP address and identifier range can be generated by compromising a sufficient number of routing tables [2]. The amount of information stored in routing tables is influenced by the type of routing geometry used in the DHT and also the size of the routing tables. Thus, our objective is to compare different DHT designs with respect to recipient anonymity through a common analytical framework and suggest improved design considerations. The important assumption here is that the lookup for a key is done through "lookup of data" and not "lookup of address" (See AChord [4]).

## 2.1 Information Stored in Routing Tables

A routing table of a node in a DHT either stores IP addresses of neighbors or mapping between IP addresses and identifier range. The information content of routing tables is directly related to routing efficiency. Consider the case of flooding (Gnutella which is an unstructured overlay). In Gnutella, each node only maintains information about its overlay neighbors and there is no mapping between a node and the keys that it stores. Thus, compromising a node only reveals information about keys stored at that node and nothing about the neighbors. However, in a DHT mapping information about neighbors is also stored for increased routing efficiency. While this leads to improved routing efficiency ($O(\log N)$ as compared to $O(N)$ in gnutella), it also makes the DHTs vulnerable to leak of recipient anonymity (Figure 1 compares the information leak from routing tables for different DHT designs). We consider the effect of the following factors on the amount of information stored in routing tables: (a) routing geometry and (b) size of routing tables.

[2]In this context, an adversary may not necessarily compromise a node to get access to its routing table. An adversary can simply occupy a certain position on the identifier space and use information contained in its own routing tables to compromise recipient anonymity

## 2.2 Information-Theoretic Framework for Analyzing Leak of Recipient Anonymity

We use the information-theoretic metric of *entropy* [10] to evaluate different DHT designs by calculating the leak of information in each design. Entropy is a measure of "randomness" in available information. Let $X$ be the random variable which represents the identifer range of a node as observed by an adversary when a routing table is compromised. Let this observation correspond to the event $\omega$. Figure 2 shows the identifier space of a DHT and the identifier range of node S corresponding to observation $\omega$. We assume that $x$ can take any value in $R$ with equal probability. We next highlight the different elements of the information-theoretic framework.

The entropy of random variable $X$ is given as,

$$H(X) = -\sum_{x \in R} Pr(X = x) \log Pr(X = x)$$

- Apriori Entropy: It is the entropy before any routing table has been compromised or in other words, before the adversary has made any observation about the mapping between a node and its identifier range. From the adversary's perspective, any node is equally likely to take any of the $N$ positions on the overlay (where $N$ is the size of the overlay). Therefore the apriori entropy is calculated as

$$H(X)^{\text{system}}_{\text{apriori}} = N \log N$$

- Aposteriori Entropy: When a routing table is compromised (corresponding to observation $\omega$), the information stored in the routing table can be used by the adversary (coalition of adversaries) to generate a mapping between node addresses and their identifier ranges. Thus, the aposteriori entropy corresponds to the entropy of the system after a routing table has been

| Privacy-enhanced DHT system | Type of information leak addressed | Method used |
|---|---|---|
| AChord | Query reply | Lookup by value |
| Neblo | Routing table | Imprecision in routing tables |
| Agyaat | Query reply | Unstructured cloud over a structured overlay |

Table 1: Proposed privacy-preserving DHT designs.

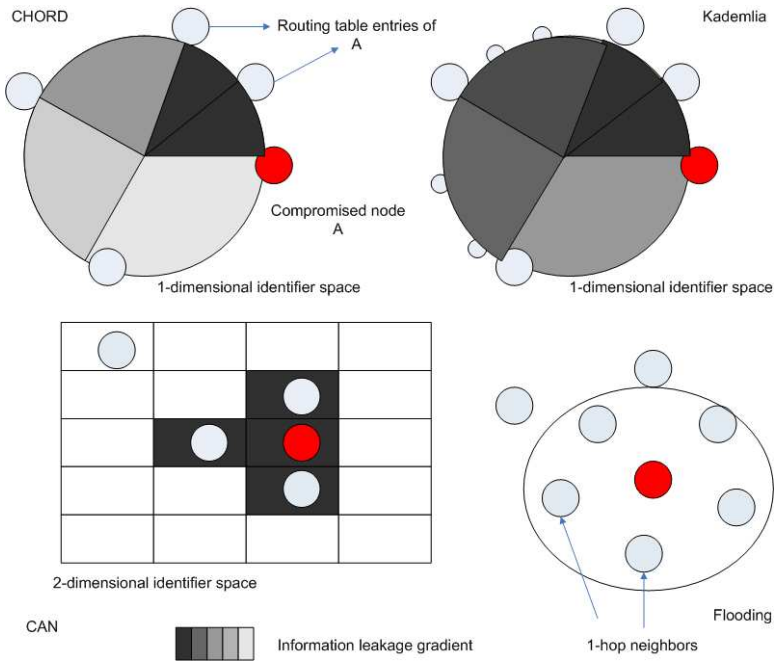


Figure 1: Information leak from compromised routing tables. The gradient shows the degree of information loss, darker shades representing high information loss. (a) Chord: The amount of information loss decreases with the distance of fingers; (b) CAN: Complete information loss about the identifier ranges of neighbors; (c) Kademlia: It exhibits similar properties as Chord; however the replication of data through the use of a replication factor $k$ requires maintenance of larger routing state information and thereby more leak of information; (d) Flooding (Gnutella): since no mapping information is maintained, negligible information about the keys stored at the neighbors is released; Pastry, although not shown here exhibits similar properties as Chord and Kademlia.

compromised. Higher the number of compromised routing tables, lower will be the system entropy (reduction in the randomness of the system).

- Information Loss

  Loss = Apriori Entropy (before any routing table has been compromised) - Aposteriori Entropy (after one or more routing tables have been compromised)

- Degree of Privacy [3]: To calculate the degree of privacy, we use the definition proposed in [12].

$$d(A) = \frac{H(X)^{\text{system}}_{\text{aposteriori}}}{H(X)^{\text{system}}_{\text{apriori}}}$$

[3] We use the terms anonymity and privacy interchangeably

## 3 Comparison of Existing DHTs

In this section, we compare how DHTs with different routing geometry perform in the face of anonymity attacks. We analyze how the routing geometry influences leak of information from compromised routing tables thereby affecting the privacy of the storage nodes. The routing geometry influences the amount of state information that is maintained in routing tables. Consider Chord which achieves logarithmic routing efficiency by maintaining $\log N$ entities in its routing table. However, two compromised routing tables might have some intersection in their routing tables. In comparison, 1 and 2-dimensional CAN maintain constant state information and the information loss is also localized (we discuss this in subsequent sections). However, to maintain the same routing efficiency as Chord, the state

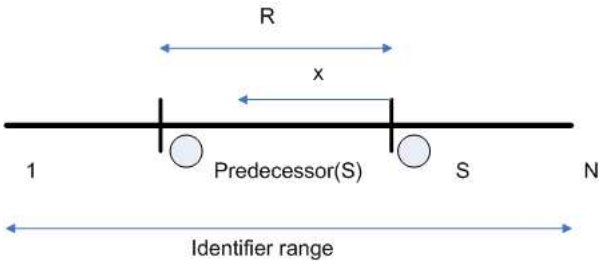


Figure 2: Identifier range of a node.

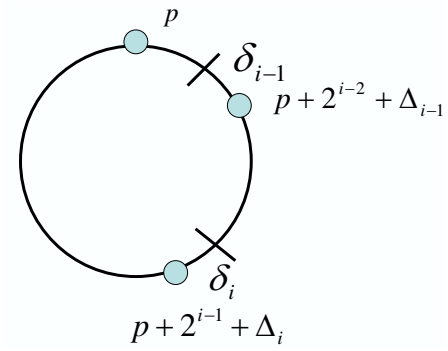


Figure 3: Identifier range of ith finger on Chordal Ring.

information in each routing table ($d$) is $\log N/2$. In DHTs like Kademlia the flexibility of routing improves routing efficiency (e.g. latency) but leads to an increased leak of information. In all these cases the state information maintained in routing tables increases with size. However, in systems like Viceroy, a constant number of entries are maintained in the routing tables and therefore information loss is constant. In contrast, flooding-based approaches are more secure because the mapping between neighbors and their overlay addresses is not stored. We now evaluate different DHT designs and quantify the information loss from compromised routing tables. Subsequently, we derive expressions for the degree of privacy. We consider the following routing geometries: a) Ring, b) Hypercube, c) XOR, d) Hybrid.

### 3.1 Ring-based DHT

In a ring-based DHT design, nodes lie on a one-dimensional identifier space on which the distance between two identifiers is the clockwise distance between them. The Chord DHT represents a ring-based design. In Chord, each node stores information about $\log(N)$ fingers, such that if a node has identifier $p$, its $i^{th}$ finger is the node closest to $p + 2^i$ on the identifier space. Moreover, the range information stored about a finger decreases as its distance from $p$ increases.

Consider Chord with $O(\log N)$ finger table entries. When a finger table is compromised, information about $O(\log N)$ $IP$ addresses are leaked to the adversary. Let the compromised node have an identifier $p$. Consider its $i^{th}$ finger. We need to calculate the lower and upper bound of its address interval. We know that the $i^{th}$ finger succeeds $p$ by at least $2^{i-1}$ on the identifier circle. Therefore, the address interval of the $i^{th}$ finger ($R_i$) is ($p$, id of $i^{th}$ finger]. This address interval can be reduced using the $(i-1)^{th}$ finger to (id of $(i-1)^{th}$ finger, id of $i^{th}$ finger]. In terms of $p$,

$$\begin{aligned} R_i &= |address - interval| \\ &= (p + 2^{i-1} + \Delta_i) - (p + 2^{i-2} + \Delta_{i-1}) \\ &= 2^{i-2} + (\Delta_i - \Delta_{i-1}) \end{aligned}$$

The aposteriori entropy corresponding to a single compromised routing table is then given as:

$$\begin{aligned} H(X)_{\text{aposteriori}} &= \sum_i \log R_i \\ &\leq \sum_i \log(2^{i-2} + \Delta_i - \Delta_{i-1}) \end{aligned}$$

Since the $i^{th}$ finger is the first finger which exceeds $p$ by atleast $2^{i-1}$, there is no other node in $\Delta_i$. Therefore, $R_i = (2^{i-2} + \Delta_i - \Delta_{i-1}) - \Delta_i = (id_i - id_{i-1})$

$$\begin{aligned} H(X)_{aposteriori} &\leq \sum_i log(2^{i-2} - \Delta_{i-1}) \\ &= \sum_i \log 2^{i-2} \\ &= \sum_{3 \leq i \leq \log N} \log 2^{i-2} \\ &= \frac{1}{2}(\log N - 1)(\log N - 2) \end{aligned}$$

If $c$ routing tables are compromised,

$$H(X)^{system}_{aposteriori} = (N - c\log N)\log N + \frac{c}{2}(\log N - 1)(\log N - 2)$$

Degree of anonymity can then be calculated as:

$$d(A) = \frac{(N - c\log N)\log N + \frac{c}{2}(\log N - 1)(\log N - 2)}{N \log N} \quad (1)$$

We observe that the information leak from a single compromised routing table is of $O(log^2 N)$ and is influenced by the number of entries in the routing table.

**Observation 1.** *A coalition of $O(\frac{N}{\log N})$ adversaries can map the entire overlay. This can be derived by setting $d(A) = 0$.*

| | CAN | Chord | Kademlia | Pastry | Gnutella |
|---|---|---|---|---|---|
| Size of adversarial coalition | $O(\frac{N}{\log^2 N})$ | $O(\frac{N}{\log N})$ | $O(\frac{N}{k \log N})$ | $O(\frac{N}{\log_{2^b} N})$ | $O(N)$ |

Table 2: Size of adversary set which can map the overlay of size $N$. The replication factor $k$ affects the leak of information in Kademlia.

### 3.2 Hypercube-based DHT

The routing used in CAN resembles a hypercube geometry. A $d$-torus is partitioned among the nodes, such that each node owns a zone. In a $d$-dimensional coordinate space, two nodes are neighbors if their coordinate spans overlap along $d-1$ dimensions and abut along one dimension. Each node maintains a maximum of $2d$ neighbors. This information includes the $IP$ address of the neighbor and its virtual coordinates. The virtual coordinates reveal the exact keyspace for which the neighbor is responsible. Thus, if a node is compromised, the exact identifier range of $(2d+1)$ nodes is revealed. Note that in CAN since each node maintains information about its neighbors which are close to it in the identifier space, the information loss from a compromised routing table is localized. Contrast this with Chord, where a compromised routing table can give information about distant nodes. We discuss this issue and its implications later.

Apriori Entropy: Using a similar analysis as Chord, a node is responsible for $\frac{1}{N}$ of the unit identifier volume. Therefore,

$$H(X)_{\text{apriori}} = \log N, \quad H(X)^{\text{system}}_{\text{apriori}} = N \log N$$

Aposteriori Entropy: We next derive the aposteriori entropy after $c$ routing tables have been compromised. Note that in CAN, when a node is compromised, the exact information about the identifier space of the node and all its neighbors can be deciphered. This corresponds to zero entropy. Therefore,

$$H(X)^{\text{system}}_{\text{aposteriori}} = (N - c(2d+1)) \log N$$

Degree of privacy is then given by

$$\begin{aligned} d(A) &\geq \frac{H(X)^{\text{system}}_{\text{aposteriori}}}{H(X)^{\text{system}}_{\text{apriori}}} \\ &= \frac{(N - c(2d+1)) \log N}{N \log N} \end{aligned} \tag{2}$$

**Lemma 1.** *For the same scaling properties, CAN is less robust to privacy attacks than Chord.*

*Proof.* To achieve the same scaling properties, $d = \frac{\log N}{2}$ in CAN. Therefore a coalition of $\frac{N}{\log^2 N}$ adversaries is sufficient to map the overlay. □

Table 2 shows the size of the adversarial coalition required for mapping the overlay for different DHT designs. The values can be easily obtained by setting $d(A) = 0$ in the respective equations and calculating for $c$.

### 3.3 DHT with XOR Routing

The routing in Kademlia is based on the concept of XOR distance: the distance between two nodes is the numeric value of the exclusive OR (XOR) of their identifiers. If the identifier space is represented by $m$ bits, for each $0 \leq i < m$, every node stores a list of $< IPaddress, UDPport, NodeID >$ triples for nodes of distance between $2^i$ and $2^{i+1}$ from itself. These lists are called $k$-buckets. While on one hand this gives routing flexibility (for example in comparison to Chord), a compromised routing table gives more information about other nodes in the system. The leak of information increases with $k$.

Consider the $i^{th}$ k-bucket. We assume that the identifier range of a node in any bucket is equally distributed among the nodes in that bucket. Using a Chord-like analysis,

$$\begin{aligned} H(X)_{\text{aposteriori}} &= \sum_i \log R_i \\ &\leq k \sum_i \log(2^{i-2}/k) \\ &= \frac{k}{2}(\log N - 1)(\log N - 2) \\ &\quad -k \log k (\log N - 3) \end{aligned}$$

Degree of privacy is then given by

$$\begin{aligned} d(A) &= \frac{1}{N \log N}((N - ck \log N) \log N \\ &\quad + \frac{ck}{2}(\log N - 1)(\log N - 2) \\ &\quad -ck \log k (\log N - 3)) \end{aligned} \tag{3}$$

### 3.4 DHT with Hybrid Routing

We use Pastry as an example of hybrid routing. Pastry uses both tree and ring based routing to search for keys. Node identifiers are regarded as both the leaves of a binary tree and as points on a 1-dimensional circle. Each node maintains a leaf-set, neighbor-set and a routing table. We evaluate the range of an entry in the routing table as perceived by an adversary. Each row in a routing table has a

maximum of $2^b - 1$ entries and there are a total of $\log_{2^b} N$ rows. Consider the $i^{th}$ row. The range covered by the $i^{th}$ row is $2^{b^{m-i}} - 1$. The range covered by the $(i-1)^{th}$ row is $2^{b^{m-(i-1)}}$. Therefore the effective identifier range covered by the $i^{th}$ row is $2^{b^{m-(i-1)}}(2^b - 1)$. Since each row contains $2^b - 1$ entries, the effective identifier range corresponding to a single node in the routing table is $2^{b^{m-(i-1)}}$. We plug this range into entropy equation and evaluate information loss as outlined below.

$$\begin{aligned} H(X)_{\text{aposteriori}} &= \sum_i \log R_i \\ &\leq (2^b - 1) \sum_i \log(2^{b^{m-1}}) \\ &= (2^b - 1) b \log^2_{2^b} N/2 \end{aligned}$$

Degree of privacy is then given by

$$\begin{aligned} d(A) &= \frac{1}{N \log N}((N - c \log_{2^b} N(2^b - 1)) \log N \\ &\quad + \frac{c}{4}(2^b - 1) b log^2_{2^b} N) \end{aligned} \tag{4}$$

# 4 Discussion

Here we analyze the effect of routing geometry on the amount of information leak from compromised routing tables. We also compare DHTs based on the size of the routing table and how that affects leak of privacy. Finally we compare and contrast structured DHT designs with unstructured overlays which use flooding.

## 4.1 Routing Geometry

Figures 4 and 5 show the variation of degree of privacy with fraction of compromised nodes for different DHT designs. For the same scaling properties (in case of Distributed Hash Tables), Chord is the most robust against leak of information. In the case of 1- and 2-dimensional CAN, the decrease in privacy with fraction of compromised nodes is less than Chord. For 1- and 2-dimensional CAN, the information leak is a function of $c$ only and is not dependent on $N$, since the size of the routing tables is constant at $2d$. However, 1- and 2-dimensional CAN take a larger number of routing steps to converge. On the other hand, to achieve the same scaling properties as Chord, $d = (\log N)/2$) and information leak increases with $N$. Observe that the plot is much steeper in case of CAN with $d = \log N/2$ as compared to Chord. Routing optimization in Kademlia is done through the maintenance of $k$ buckets at each node. While this improves lookup latency, it requires the maintenance of a large amount of state information at each node. Therefore a compromised routing table leaks more information than Chord. For the same scaling properties, Pastry is the closest to Chord. However, for lookup optimization, Pastry maintains a leaf-set (besides the routing table) at each node. This leaf-set can leak information about an additional set of nodes in the system.

Table 3 shows the percent of information loss (in bits) when a routing table is compromised. We observe that Chord shows the maximum resilience to privacy leak for different overlay sizes. Observe that in the case of 1 and 2-dimensional CAN, the fraction of information loss $\propto \frac{1}{N}$ since the number of entries in the routing table is fixed (=2d). However, when $d = \log N/2$, the leak of privacy increases and is appreciably higher than Chord. In Pastry a routing table stores information about $(2^b - 1) \log_{2^b} N$ nodes. If $b = 1$, the state information maintained is same as that of Chord. Our analytical model shows that the privacy leak in that case is similar to that of Chord. However, the leak increases when the base is 4. In Kademlia the replication parameter $k$ is typically set as 20. We observe that for small overlay sizes, the number of replicas has an adverse effect on the leak of information from routing tables. However, the information leak decreases with an increase in overlay size. The general trend is that DHT designs with routing optimizations tend to exhibit higher leak of information from compromised routing tables.

We also observe that the routing geometry of CAN leads to "localized" information loss when a routing table is compromised. By "localized" we mean that when a node is compromised, the routing table gives information only about neighbors which are close on the identifier space. Contrast this with Chord, in which the finger table stores information about distant nodes. The implication is that in CAN (as compared to other designs), compromised nodes in a certain region of the identifier space localize the information leak without affecting substantial portions of the overlay.

## 4.2 Routing Table Size

The size of the routing table affects the leak of information about the overlay. The routing geometry and routing optimizations influence the number of entries in the routing table (Table 4 shows the routing table size for different DHT designgs). We have observed that in all the aforementioned DHT designs the size of the routing table varies with the size of the overlay $N$. In contrast flooding-based approaches in unstructured overlays maintain constant state information (typically $3 - 8$ in gnutella). However, the important question to ask is can we have a DHT design which achieves logarithmic routing efficiency by maintaining constant state information. Viceroy [5], which emulates the butterfly network, achieves such efficiency. We did not include

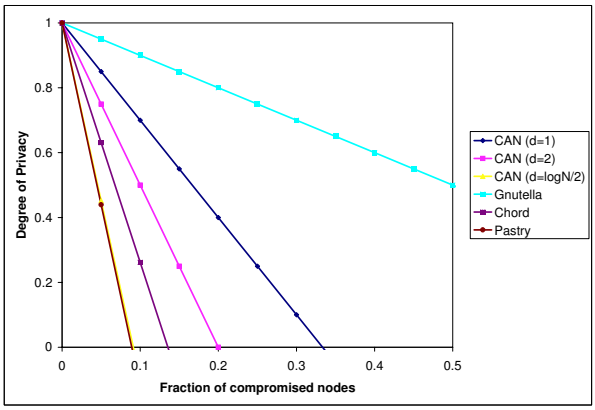


Figure 4: Leak of privacy for N=1000.

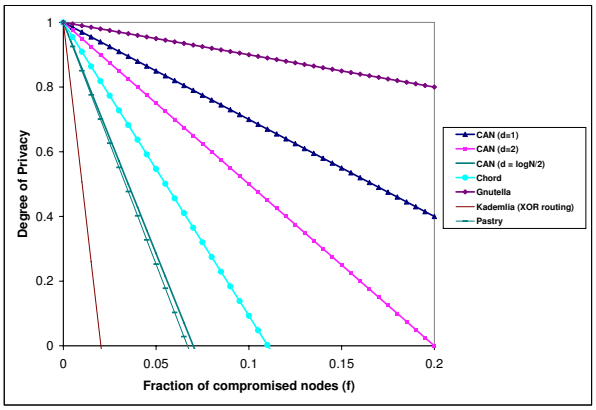


Figure 5: Leak of privacy for N=50000.

| Overlay Size | CAN (1-dim) | CAN (2-dimension) | CAN ($d = \log N/2$) | Chord | Kademlia | Pastry | Gnutella |
|---|---|---|---|---|---|---|---|
| 500 | 1 | 0.6 | 2 | 0.55 | 34.9 | 1.34 | 0.2 |
| 1000 | 0.5 | 0.3 | 1.1 | 0.28 | 18.8 | 0.75 | 0.1 |
| 10000 | 0.05 | 0.03 | 0.14 | 0.03 | 2.3 | 0.1 | 0.01 |
| 50000 | 0.01 | 0.006 | 0.03 | 0.005 | 0.5 | 0.02 | 0.002 |

Table 3: Percentage of information loss (measured in bits) when a routing table is compromised. A 100 % information loss corresponds to the case when the entire overlay can be mapped by the adversary (for kademlia, k=20 and for Pastry, b=2).

Viceroy in our analysis since we wanted to analyze DHTs which are based on a similar design principle. Each node maintains information about 7 other nodes in the overlay. As part of our future work, we plan to analyze the information leak from the Viceroy network.

### 4.3 Comparison with Unstructured Networks

In this section, we compare the information leak property of DHTs with that of flooding based search systems. We use the Gnutella [3] protocol as the basis for an unstructured routing geometry. In Gnutella and other flooding-based search systems, no state information corresponding to mapping between nodes and keys is maintained. Thus, each node only maintains information about its neighbors but is not aware of the keys that are stored at the neighbors. Any search query propagates through the unstructured network and eventually reaches the node responsible for the key. Therefore, if $c$ nodes are compromised, the adversary can know only about the keys mapped to those compromised nodes.

The degree of privacy is then given by

$$d(A) = \frac{H(X)^{\text{system}}_{\text{aposteriori}}}{H(X)^{\text{system}}_{\text{apriori}}} \quad (5)$$

$$= \frac{(N-c)\log N}{N \log N} \quad (6)$$

Observe the bits of information leaked in the case of Gnutella (Table 3). Since each node maintains information only about its 1-hop neighbors and is blind with respect to the information stored in the neighbors, an adversarial coalition of size $O(N)$ is required to map the overlay. Thus, an unstructured overlay has very good privacy properties; however, it lies at the end of the routing efficiency spectrum.

## 5 Related Work

There have been some attempts at providing anonymity for structured overlays. As mentioned earlier, AChord [4] attempts to improve recipient anonymity in Chord through the use of data lookup instead of address lookup. Thus, the IP address of the storage node is not revealed in the query reply. However, information leak from routing table entries cannot be prevented. Several other studies [7, 2, 1] have focused on the issue of sender-anonymity in Chord. We aim at analyzing the effect of leak of information from routing tables on recipient anonymity. An analytical framework for calculating information leak in the Chord protocol (with respect to the identity of the sender) is presented in [7]. Neblo [2] proposes the use of imprecise routing tables for enhancing recipient anonymity. While it highlights the importance of information leak from routing tables, the focus is on the Chord routing protocol and the design objective is obfuscating the information content of routing tables. We try to analyze the effect of the DHT routing geome-

| | CAN | Chord | Kademlia | Pastry | Gnutella | Viceroy |
|---|---|---|---|---|---|---|
| Size of routing table | $2d$ | $\log N$ | $k \log N$ | $\log_{2^b} N$ | Constant | Constant |

Table 4: Size of routing table.

try on the amount of information leak from routing tables. Anonymity in structured P2P networks was also studied in [1]. An empirical entropy-based metric was developed to measure source-anonymity in Chord. A routing extension was proposed which allows a tradeoff between anonymity and performance. Agyaat [11] attempts to provide recipient anonymity through the use of a two-tier hybrid organization in which the Chord structured overlay works together with a gnutella-like overlay to route messages. Gnutella-like clouds are connected with one another by means of a Chord ring.

## 6 Conclusion

In this paper we have proposed an information-theoretic framework for evaluating the resilience of different DHT designs against leak of privacy. Our entropy-based analytical model helps us to quantify the leak of information from compromised routing tables. We analyze the effect of routing geometry, optimizations and route table size on the amount of information leak.

Our analytical results show that for the same routing complexity, ring-based DHT (Chord) has the minimum information leak. The general trend is that the state information stored in routing tables increases with routing optimizations, thereby resulting in significant information leak. We observe that Kademlia (XOR routing) has a significant amount of information leak for small overlay sizes. Pastry's (hybrid routing) performance is quite close to that of the ring structure. The hypercube-based routing in CAN has an important side-effect, i.e. of localizing the information loss. We also analyse the effect of routing table size on leak of information. We believe that our preliminary findings can help better understand the effect of routing geometry on the state information stored in routing tables which can lead to development of DHT designs with an optimal balance between routing efficiency and information leak.

## References


[1] N. Borisov and J. Waddle. Anonymity in structured peer-to-peer networks. Technical Report UCB/CSD-05-1390, EECS Department, University of California, Berkeley, 2005.

[2] G. Ciaccio. Improving sender anonymity in a structured overlay with imprecise routing. In *Proceedings of the Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, 2006.

[3] Gnutella. http://gnutella.wego.com.

[4] 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 the 1st International Workshop on Peer-to-Peer Systems(IPTPS)*, 2002.

[5] D. Malkhi, M. Naor, and D. Ratajczak. Viceroy: A scalable and dynamic emulation of the butterfly. In *Proceedings of the 21st Annual Symposium on Principles of Distributed Computing*, 2002.

[6] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems(IPTPS)*, 2002.

[7] C. W. O'Donnell and V. Vaikuntanathan. Information leak in the chord lookup protocol. In *4th International Conference on Peer-to-Peer Computing*, 2004.

[8] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *Proceedings of the 2001 ACM SIGCOMM*, San Diego, CA, 2001.

[9] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of the ACM/IFIP/USENIX Middleware*, 2001.

[10] C. Shannon. The mathematical theory of communication. *Bell Systems Technical Journal*, 30:50–64, 1948.

[11] A. Singh, B. Gedik, and L. Liu. Agyaat: Mutual anonymity over structured p2p networks. In *Emerald Internet Research Journal*, volume 16, 2006.

[12] S. Steinbrecher and S. Köpsell. Modelling unlinkability. In R. Dingledine, editor, *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*. 2003.

[13] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 ACM SIGCOMM*, San Diego, CA, 2001.