Considering Security in Distributed Hash Tables

Heiko Niedermayer, Klaus Wehrle, Thomas Schreiner, Georg Carle

{*niedermayer*|*wehrle*|*carle*}@informatik.uni-tuebingen.de
Wilhelm-Schickard-Institute for Computer Science
University of Tübingen

***Index Terms*— Peer-to-Peer, Distributed Hash Tables, Security**

A major purpose of Peer-to-Peer (P2P) systems is the management of large amounts of data distributed across many systems. Distributed Hash Tables (DHT) are designed for a highly scalable, self-organizing and efficient distribution and lookup of data. Regarding security, these so-called *structured Peer-to-Peer-systems* still show crucial security flaws that have to be solved before using DHTs within critical applications. In this contribution we want to address these security problems and discuss possible solutions.

The basic principle of Distributed Hash Tables is to build a highly scalable and self-organizing method to manage data across distributed systems. To achieve scalability and avoid single points of failure, the main design goal is an equal distribution of the efforts for storing and managing data among peers. The common DHT approaches currently rely on the assumption of a trusted web of peers, where statistical failures are the major cause for instability.

Distributed Hash Tables offer a generic API for managing distributed data, on top of which various applications may offer their respective services. Thus, security requirements may vary tremendously for different applications. First, we identify security goals for some applications using DHTs. Second, we address security goals special for Distributed Hash Tables. Finally, we present possible security problems.

Application-driven Security Goals: In Classical Filesharing, integrity and availability are important security goals. In Distributed File Systems with permission management we also need confidentiality, accountability, integrity, and controlled access. For Instant Messaging it should be availability and controlled access.

Confidentiality of the data can be achieved via encryption and secret-sharing approaches. Usually, this might be considered to be a problem above DHT level. Confidentiality of the DHT messages can be achieved with encryption.

There are two notions of integrity. First, message integrity should be proven somehow, e.g. with message authentication codes. The second notion is to ensure that the correct document was delivered.

DHT-centered Security Goals: Distributed Hash Tables provide routing to key-value pairs. Thus, security on the level of DHTs is mainly related to the security of DHT routing, i.e. correctness and availability of the routing. The correct structure of the network should be ensured. Controlled access to the DHT is also a security goal. This includes nodes joining the DHT as well as ensuring that only messages by DHT member nodes are accepted.

Attacks on DHTs: An important aspect of peer-to-peer overlay networks is that the topology may not be related to the topology of the underlay. Depending on the protocol a node may position itself anywhere on the net. This is different from IP networks where the topology of the infrastructure determines the position of a node.

Passive attacks on DHT queries can only be done when the attacking node is on the path from the node asking for an ID to the node responsible for the ID. Since the responding node can answer directly to the initiating node an attacker can usually only listen the query.

Active attacks are attacks that modify or delete messages or initiate fake messages. Sending fake queries can be done by any node. Modifying or deleting queries from other nodes can only be done by nodes on the path of the message. A node responsible for certain IDs can be attacked when the attacker places nodes before the node in ID space, so that messages are likely to pass the attacker. For example, in Chord all messages pass the predecessor of the node responsible for an ID. Attacking a particular node that is requesting data can be done by attacking the entries of its routing table. To be more precise, by positioning nodes on the DHT in such a way that these nodes are the correct routing table entries of the victim's routing table. To find a node with a particular IP is either easy (when node ID is determined by IP only or when the node tells its ID on request to anyone asking for it) or hard (asking for ID not supported by the protocol or ID not related to information the attacker knows about the node). In the latter case, scanning the DHT and creating a mapping of node ID and IP can be done in $O(N \log(N))$ time with N being the number of nodes in the DHT. For small and mid-size DHTs this is still feasible. The average lifetime of nodes in a DHT puts an upper time limit to this approach.

To conclude, we will discuss the security problems of DHTs and present ideas to solve or avoid them.