

# Security Challenges for Peer-to-Peer SIP

Jan Seedorf, University of Hamburg

## Abstract

Recent research activities have proposed using a peer-to-peer network for user registration and user location in Session Initiation Protocol (SIP)-based voice-over-IP (VoIP) networks. The main motivation for peer-to-peer (P2P) SIP is higher robustness as well as easy configuration and maintenance (compared to client-server SIP). However, these advantages come at the price of security. In this article the security challenges of using a P2P network as a substrate for SIP communication are explored. After a short introduction to SIP and structured P2P networks, the different approaches that have been proposed for using P2P technology in conjunction with SIP are compared. We focus on design issues that affect security in order to identify the security implications of using a structured overlay network for SIP registration and location lookup. Finally, possible solutions for securing P2P SIP are examined and explicit suggestions on how to improve the security of P2P SIP are given.

Voice-over-IP (VoIP) has matured over the past couple of years. Much of the evolution of VoIP to a now widely used application is due to the Session Initiation Protocol (SIP). SIP is a signaling protocol for VoIP. Originally, SIP was specified as a client-server protocol to set up multimedia communications. However, recent proposals suggest to use SIP in a peer-to-peer (P2P) setting. We refer to this as P2P-SIP. More precisely, several researchers have proposed the use of a *distributed hash table* for SIP registration and user location as an alternative to SIP servers ([1, 2]). The initiators of P2P-SIP claim higher robustness (against failure) as well as easier configuration and maintenance as the main motivation for P2P-SIP. To analyze the advantages and business model of P2P-SIP is out of the scope of this article. We consider P2P-SIP as a future alternative to client-server SIP and solely look at the security issues. Clearly, a peer-to-peer setting imposes new security threats to SIP communications; for instance, the lack of a central authority makes authentication of peers a hard problem. Our goal is to look at known results on security of P2P networks and identify if and to what extent these results hold for P2P-SIP.

In this article when referring to *P2P-SIP* we mean the use of a peer-to-peer network as a substrate for SIP user registration and SIP user location as an open standard for VoIP. Somewhat related to P2P-SIP are *SIPShare* and *Skype*. SIPShare is a P2P file-sharing system that uses SIP for signaling. However, SIPShare is not used for VoIP and thus does not meet our definition of P2P-SIP. Skype is a peer-to-peer network for VoIP. However, Skype uses a secret and proprietary protocol instead of SIP for VoIP signaling. Therefore, Skype security is out of the scope of this article.

### Related Work

Much work has been done on the security of VoIP networks. Some of it is close to our work, as it focuses on the signaling part of VoIP communications and especially on the Session Initiation Protocol (SIP). However, SIP is specified as a client-server protocol [3]. Thus, prior research identifies the security risks of using SIP in a client-server setting. Because

P2P networks differ greatly from client-server networks, this article looks at SIP-security from a different perspective. Aside from SIP security, research has been done to identify the fundamental security problems in (structured) peer-to-peer networks. We build on this work and identify the security implications of using a distributed hash table in the application-domain of VoIP.

The rest of this article is organized as follows. We give a short introduction to SIP, present an overview on the evolution of peer-to-peer networks, and then explain distributed hash tables. We compare the different proposals that have been published for P2P-SIP. The challenges for securing these approaches are investigated. Possible solutions for securing P2P-SIP are discussed. Further, we make some explicit suggestions on how to improve the security of P2P-SIP. We conclude with a summary and remarks on future work.

### The Session Initiation Protocol

The Session Initiation Protocol (SIP) is an IETF standard for signaling in multimedia communications over IP [3]. SIP is a client-server protocol similar to HTTP. Signaling in SIP is based on (ASCII compatible) text messages; a message is composed of a message header and an optional message body. Messages are either *requests* or *responses*. The original RFC specifies six types of requests: INVITE (initiating a call signaling sequence), BYE (terminating a session), ACK (acknowledging), OPTIONS (querying of capabilities), CANCEL (canceling a request in progress), and REGISTER (used to register location information at a registrar). When a SIP entity receives a request, it performs the corresponding action and sends back a response to the originator of the request. Responses are three-digit status codes (as in HTTP/1.1), categorized into six classes (e.g., 3xx for redirect messages). Examples for response codes are “180-ringing,” “200-ok,” or “302-moved temporarily.”

SIP defines five types of (logical) entities: *user agent*, *proxy*, *registrar*, *redirect server*, and *location server*. A user agent is any

terminal (hardware or software) participating in SIP communications. A proxy forwards messages to another server or user agent. A redirect server tells the sender of a message where to send it, rather than forwarding it. To facilitate mobility, users can register their current location with the registrar of their domain: a location server is used by a registrar to store the location of users (the binding of a SIP-URI with a current IP-address). Other SIP entities (proxies or redirect servers) can use the location server to look up the current location of SIP users. Addressing is based on a uniform resource identifier (URI). A SIP-URI is similar to an e-mail address and generally of the type "sip:user@domain."

Figure 1 shows a basic example [4] for session establishment with SIP. User agents A and B are in different domains and have different proxies. First, the callee (user agent B) needs to register with its local registrar (1) so as to be able to receive calls. The registrar stores the location information at a location server (2). When user agent A wants to call user agent B, it sends an INVITE-request to its local SIP-proxy (3), which passes on the request (possibly after a DNS lookup) to the proxy of user B's domain (4). The proxy in domain B needs to look up the IP address of user agent B at the location server (5, 6) before it can send the request to user agent B (7). Often the registrar, location server, and proxy of a domain are implemented in one host (denoted by the dotted line in Fig. 1). In this example, the response message for user agent A takes the same route back (8, 9, 10), possibly for billing purposes.

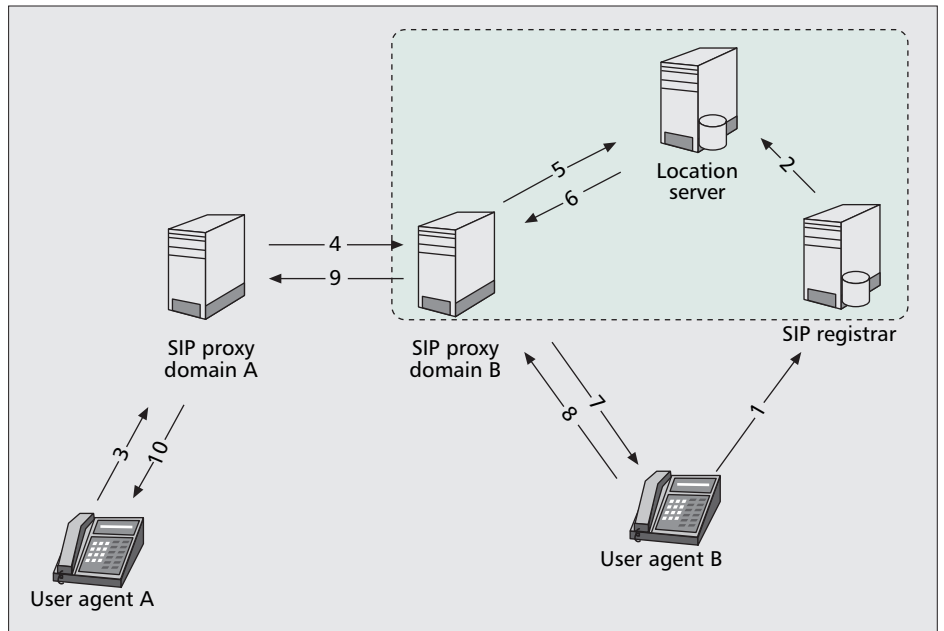


Figure 1. Session establishment of a voice connection with SIP [4].

## Structured Overlay Networks

Peer-to-peer networks follow a different paradigm than client-server based systems. A key underlying attribute is that each node participates in the network by offering and using services at the same time. There is no central control and the network organizes itself in a dynamic way. In [5], a peer-to-peer (P2P) system is defined as follows: "Peer-to-peer (P2P) systems are distributed systems without any centralized control or hierarchical organization, where the software running at each node is equivalent in functionality"

This definition defines pure P2P architectures only. Yet many networks are also considered P2P, even though they employ a central authority (hybrid P2P) or use super nodes (nodes that offer more functionality than others). Because P2P networks are built at the application layer and use the underlying network for the exchange of messages, P2P systems are also called *overlay networks*.

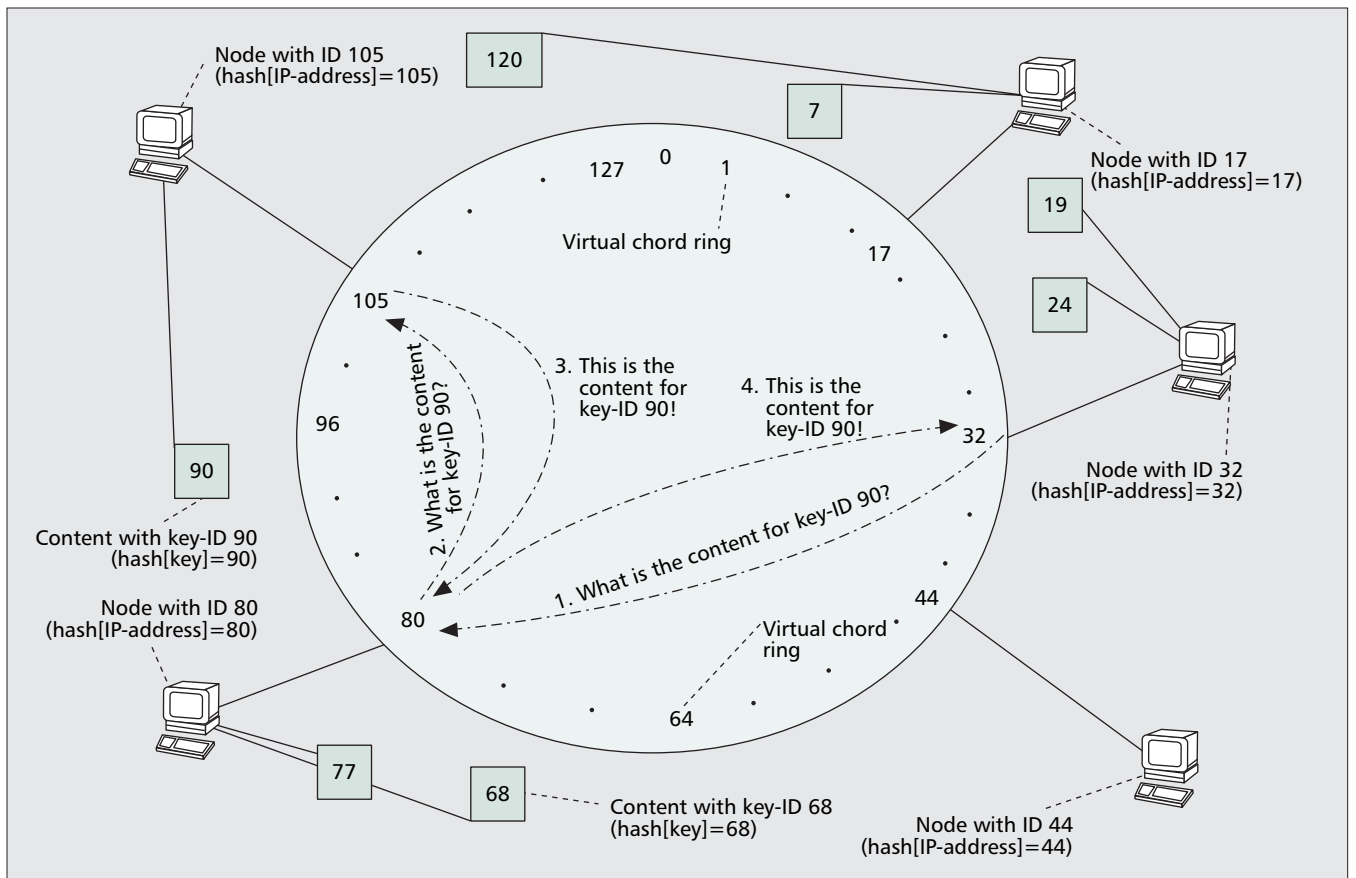
P2P networks have evolved in recent years. Early systems (e.g., Gnutella) used flooding for message routing in the network. Any node receiving a search request would broadcast this message to all its neighbors. The message has a time-to-live value which is reduced at every hop to prevent messages from being routed in the network forever. These systems cannot give any formal guarantees that a message in the network will reach its destination. Furthermore, broadcast messages impose an unnecessary traffic burden on the network.

## Structured Overlay Networks/Distributed Hash Tables

To improve lookup time for a search request, *structured overlay networks* have been developed that provide load balancing and efficient routing of messages. Structured overlay networks offer an infrastructure on which different applications can be built rather than a separate P2P network for every application (like traditional P2P file-sharing systems). The overlay provides content distribution to the application using it. Specifically, given a key (as a search request inserted by a participating node into the network), the network returns the node responsible for storing data belonging to that key. To provide efficient lookup of the node responsible for a given key, structured overlay networks use distributed hash tables (DHTs). A DHT essentially is a hash table that is distributed among the nodes participating in the network. Each node stores only a small part of the whole hash table for which it is responsible. The DHT protocol is designed in such a way that a message querying a particular key can be routed efficiently to the node responsible for that key and will always succeed in reaching the responsible node. Further, the overlay is capable of handling nodes entering and leaving the network at a high frequency. Examples of structured overlay networks are CAN, Chord, Pastry, and Tapestry. These networks can give formal guarantees (upper or lower bounds) on the number of hops needed for a search request to succeed.

## Chord

We use Chord [6] to exemplify structured overlay networks because it is the protocol used as a prototype in most proposals for P2P-SIP. In a Chord overlay, every node's IP-address is mapped to an  $m$ -bit Chord identifier (node-ID) with a predefined hash function  $h$ . The same hash function  $h$  is also used to map any key (of data) to be stored by the Chord network onto a key-ID; this forms the distributed hash table. Chord uses a virtual ring as the structure for routing of messages and key lookup. In this ring the nodes are ordered clockwise from 0 to  $2^m-1$  according to their ID. Each node in the ring is responsible for storing the content of all key-IDs that are equal to or less than its own identifier but larger than the identifier of the node's predecessor in the Chord ring. In a routing table, each node  $n$  stores the IP addresses of  $m$  suc-



■ Figure 2. Message routing in a Chord network with  $m = 7$ .

cessor nodes plus its predecessor in the ring. The  $m$  successor entries in the routing table point to nodes at increasing distance from  $n$ . In particular, the  $i$ th entry in the routing table contains the IP address of the first node in the ring that follows  $n$  by at least  $2^{i-1}$  in the Chord ring. Note that all routing is done mod  $2^m$  (e.g., nodes can route past  $2^m-1$  in the ring). Thus, the first entry in the routing table is the node directly succeeding  $n$  in the ring, while the last entry in the routing table points to a node about  $2^m/2$  away from  $n$  in the ring. Routing is done by forwarding messages to the largest node-ID in the routing table that precedes the key-ID until the direct successor of a node has a larger ID than the key-ID. This successor node is responsible for the key. When a node enters the system, it hashes its IP address to calculate its node-ID. It then contacts one or more bootstrap nodes and inserts a join message into the network that is routed to the node with the lowest ID higher than the joining node's ID in the ring. The joining node and its successor and predecessor exchange messages to update their routing tables. All nodes must frequently update their routing tables to keep up with node joins and leaves. Chord also offers a primitive for storing content in the network which is routed to the node responsible for the content's key. In a Chord network with  $N$  nodes, each node has a routing table size of at most  $O(\log N)$ , and all lookups are resolved via  $O(\log N)$  messages in the network [6].

Figure 2 shows a Chord ring where  $m = 7$ , five nodes are present in the network.<sup>1</sup> In this example, if the node with ID 32 wants to lookup the key that hashes to 90, it contacts the

node in its routing table with the node-ID closest to (but preceding) 90 (the node with ID 80 in our example). The node with ID 80 discovers that the first node in its routing table has a higher ID (105) than the key (90). Thus, the node with ID 105 must be responsible for the key and the message is routed to it. Two ways for message routing are possible in Chord: *iterative* or *recursive*. First, a querying node contacts the node in its routing table that is closest to the key. With *iterative routing* the contacted node returns the closest node to the key from its routing table to the querying node. The querying node goes on contacting this node to iteratively get closer to the key. When *recursive routing* is used (as in our example), the query is proxied hop-by-hop through the network until it reaches the node responsible for the key.

### Proposals for P2P-SIP

Recently, proposals for using a structured overlay network in conjunction with SIP-based VoIP have been published. The idea is as follows. Instead of servers, a distributed hash table can be used for registering and locating a user-id (SIP-URI). The proposals use Chord for explaining the design but emphasize that it may be replaced by some other overlay protocol in the future. The key to be looked up is the user-id; the network stores and returns the current user location record. We first describe the proposed solutions below and then discuss the differences. Our focus is primarily on design decisions that affect security, which include node-ID computation, overlay routing, authentication of nodes, SIP message semantics, and representation of identity.

Singh and Schulzrinne [1] envision a hierarchical architecture in which multiple P2P networks are represented by a DNS domain. A global DHT is used for interdomain routing of messages. To avoid the introduction of new SIP message-

<sup>1</sup> For simplicity, throughout this article we exemplify Chord operations and attacks on Chord with  $m = 7$ . In reality, a Chord ring will be much larger (for instance,  $m = 160$  if SHA-1 is used as the hash function  $h$ ).

headers, node-IDs and key-IDs are represented as SIP-URIs. The node-ID is the hash of the node's IP address followed by the IP address or domain name, for example, [hash(IP address)]@IP address or [hash(IP address)]@domain. Though this URI structure contains redundant information, it is desired because all SIP implementations support SIP-URIs in headers. The user-id (the key to be looked up in the overlay) is a valid e-mail address within the domain. It can be used for e-mail-based authentication to verify users' identities. SIP is used as the protocol for message exchange within the Chord network. The SIP-register method is used for storing key/data pairs in the network as well as for node queries. User-id queries are done by INVITE messages. Not all user agents participate in the P2P network; only nodes with sufficient memory, bandwidth, and uptime become super nodes and form the DHT. Regular nodes communicate with super nodes as if these were SIP-proxies. Within each P2P network (domain), any super node is responsible for keys as specified by Chord. A P2P-adapter, called *SIPpeer* [7], has been implemented. It can be placed between any regular SIP user agent and the P2P network.

Bryan, Lowekamp, and Jennings [2] have introduced *SOSIMPLE* as a P2P enhancement to the *SIMPLE* protocol. *SIMPLE* is an extension to SIP for instant messaging where the content (instant messages) is transported within SIP messages. *SOSIMPLE* is based on an internet-draft for P2P SIP registration and user location by the same authors [8]. The approach from Bryan *et al.*<sup>2</sup> is similar to the proposal in [1]. SIP is used for messaging in the Chord overlay. However, it has some differences, which include:

- Not only the IP address, but also the port is hashed to compute the node-ID.
- New SIP headers are introduced (DHT-NodeID, DHT-Link); this enables the transfer of more than one routing table entry within a single SIP-message.
- Solely SIP-Register messages are used for registrations and queries of node-IDs/user-IDs (not SIP-Invite for user-ID queries)
- Bryan *et al.* consider a pure P2P setting without super nodes; any user agent can directly use the overlay (possibly with an adapter).
- Bryan *et al.* explicitly propose iterative overlay routing while Singh and Schulzrinne allow for both routing strategies and use recursive overlay routing in their prototype implementation [7].

The authors of *SOSIMPLE* recommend a public key infrastructure (PKI) to verify users' identities.

Johnston and Sinnreich [9] envision a general location service to be used mostly by SIP servers. This approach is different than the ones described previously, as it does not eliminate any of the SIP servers: the location service shall be a P2P network that can optionally be used by SIP servers to store and look up location information. They advise not to use SIP as the protocol for DHT message exchange. Instead, a general overlay network for a location service shall be used by other protocols/applications (besides SIP) as well. Shim, Narayanan, and Daley have described a general architecture for P2P-SIP [10] that also separates DHT and SIP functionality in two layers. Because these approaches [9, 10] are not very precise on the design issues affecting security, we will mostly discuss the proposals by Singh and Schulzrinne [1] and Bryan *et al.* [2, 8].

More recently, Singh and Schulzrinne have described an architecture for using an existing DHT implementation (called

OpenDHT) as a SIP location service [11]. In this approach, SIP entities do not necessarily become part of an overlay. Instead, they use an externally managed overlay as clients by sending put/get messages to nodes in that overlay with remote procedure calls. Besides location bindings, certificates or public keys belonging to SIP-URIs can also be stored within the overlay. A public key infrastructure or a web-of-trust model is suggested for verification of certificates. Much of the security is based on the assumption that nodes are not malicious because the DHT is a managed network, consisting of trusted nodes. For our discussion of P2P-SIP security, we consider the more general case where arbitrary nodes can join the DHT and nodes are initially nontrusted.<sup>3</sup>

## Security Challenges for P2P-SIP

Because P2P systems are inherently different from client-server systems, new challenges for security arise. For instance, the lack of a central authority makes authentication in a pure P2P network difficult. Without authentication, adversary nodes can spoof identity and falsify messages in the overlay. This enables malicious nodes to launch man-in-the-middle or denial-of-service attacks. Douceur showed in [12] that without a trusted agency which certifies identities, adversary nodes can control a large fraction of an overlay network. Castro *et al.* identify three requirements for secure structured overlay networks: secure node-ID assignment, secure routing table maintenance, and secure message forwarding [13]. We look at attacks on these requirements in the next two paragraphs, followed by other security challenges.

### Attacks on the ID Mapping Scheme

Much of the security of structured overlay networks is based on the assumption that joining nodes are assigned IDs at random. To analyze node-ID generation in P2P-SIP,<sup>4</sup> we take the model from [14]:

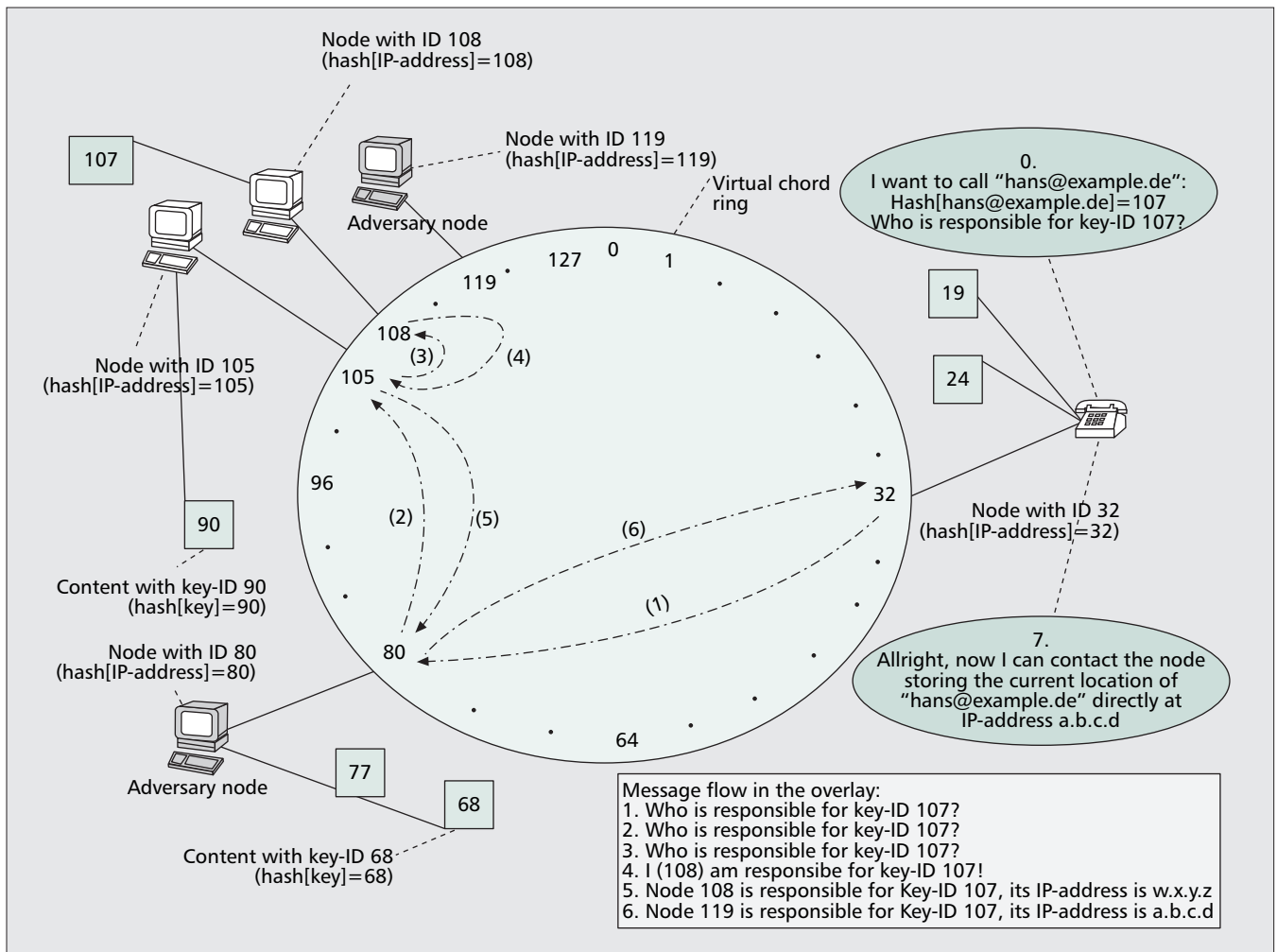
*At any time t we have n honest nodes and e \* n adversary nodes in the Chord ring (e < 1). A malicious entity owning multiple nodes can control the Chord ring by launching join-leave attacks: the attacker joins and leaves the network with its nodes until a sequence of O(log n) adversary nodes is reached. The goal is to find a joining strategy that prevents an adversary from gaining such a sequence. In that case, it can be guaranteed that any message reaches its destination (though at a high performance cost) by simply routing each message to O(log n) nodes directly following in the ring [14].*

This model focuses on the availability of the network; the goal is to guarantee that a message will reach the responsible node. It also gives a lower bound for a routing strategy that will provide availability (if the joining strategy of the overlay prevents an attacker from gaining sequences) by using multiple lookup paths. With IPv6, vast numbers of IP addresses can be available to an attacker to form such *join-leave attacks* on a P2P-SIP Chord network. But even with IPv4, *join-leave attacks* on node-ID generation are possible when an attacker gets assigned IP addresses dynamically (e.g., by its ISP). Further, in the approach in Bryan *et al.* attacks on node-ID generation are possible: an attacker tries ports at will until the

<sup>3</sup> The approach in [11] can be extended to this broader case as well. However, for our discussion, when we refer to "Singh and Schulzrinne" we mean the approach presented in [1, 7].

<sup>4</sup> We exemplify our analysis with Chord. Nonetheless, the revealed attacks are also possible on other distributed hash tables and our general results apply to other distributed hash tables as well.

<sup>2</sup> We refer to "Bryan *et al.*" to subsume the *SOSIMPLE* framework and the Internet draft by Bryan, Lowekamp, and Jennings.



■ Figure 3. Man-in-the-middle attack on P2P-SIP.

hash of  $[IP:Port]$  matches the desired value. Note that an attacker does not have to actually join and leave the network: because node-ID assignment in Chord is deterministic by hashing the IP address, attackers can compute node-IDs in advance. Scheideler suggests using a  $k$ -rotation joining strategy to perturb nodes frequently (see [14] for details), resulting in random node-ID assignment and making join-leave attacks difficult for an attacker. Another way to prevent join-leave attacks would be to make the join operation in some way costly for the joining node. For example, computational puzzles or micropayment systems can make it costly for an attacker to repeatedly join the network (or to precompute node-ID values). However, these strategies cannot be used for P2P-SIP unless nodes can be forced to change their IP address, or node-ID generation will be based on something different than the IP address (e.g., by adding a timed randomness service [15]).

### Attacks on Overlay Routing

Any malicious node within the overlay can drop, alter, or wrongly forward a message it receives instead of routing it according to the overlay protocol. This can result in severe degradation of the overlay's availability. Furthermore, confidentiality and integrity can not be provided for P2P-SIP registration and location lookup messages by the overlay. In P2P-SIP this means that an attacker cannot only prevent access to location information, but also forge responses to SIP-messages. Srivatsa and Liu observe in [16] that Chord nodes can check the results of routing queries as follows. The node

looking up a key in the network exploits the fact that with any routing hop it should get closer to the node-ID responsible for the key. Thus, detecting invalid lookups is possible in P2P-SIP networks if iterative routing is used: a requesting node simply hashes the IP address it receives within a SIP-redirect message. It then checks if that hash is closer to the key-ID than the node-ID it received on the previous hop. However, an attacker can still perform an undetected man-in-the-middle attack if it can place itself close enough to the desired target key (using attacks on node-ID generation described previously). In this case it is likely that the attacker is the last node before the node responsible for the key in the Chord ring.

Figure 3 shows an example of a man-in-the-middle attack on P2P-SIP. The nodes with ID 80 and ID 119 conspire to form an attack on the content of key-ID 107. The (honest) node responsible for the key is node 108.<sup>5</sup> Because node 119 could also be responsible for the key, the initiator of the key lookup (node 32) cannot detect the attack. Note that messages (2) to (5) are not necessary to form the attack. However, obtaining the real location for key-ID 107 can be useful information for the adversary nodes 80 and 119 (perhaps to perform further man-in-the-middle attacks on the media stream). Recursive routing is used in the example. It can be observed that with iterative routing node 80 would not be able to perform the attack (node 32 would get suspicious if node

<sup>5</sup> For simplicity, in this example we denote node  $x$  as the node with Chord-ID  $x$ .

80 redirected it directly to node 119 because it assumes there exists some node with an ID lower than 107).<sup>6</sup> To perform a man-in-the-middle attack with iterative routing, the adversary has to place itself close to the targeted key-ID (e.g., between nodes 105 and 108 in our example). Danezis *et al.* suggest iterative routing where a node returns its whole routing table instead of the node closest to the key [17]. Hence, the requesting node is in full control at any routing hop and can decide on the next routing hop itself. This routing decision can be based on other measurements than closeness in the ID-space (as in normal Chord routing). Danezis *et al.* introduce routing based on trust diversity: nodes keep a history of nodes they previously queried and try to balance nodes they use in queries. The intention is not to put too much “trust” in single nodes, but rather spread the nodes used for routing over time.

### Bootstrapping

Any node wanting to join the overlay needs to know at least one node that already participates in the overlay. Possible options for this *bootstrapping* are static (persistent) nodes, cached nodes, or broadcast mechanisms (e.g., SIP-multicast). In any case, if the initial bootstrap node is malicious, the joining node can easily be attacked. Without a preestablished trust relationship with the bootstrap node, secure bootstrapping is an open question.

### Identity Enforcement

Without a central authority, one of the open problems for P2P-SIP is identity enforcement. The overlay network has to prevent duplicate IDs. This applies to node-IDs (i.e., if two IP addresses hash to the same value) as well as to key-IDs (i.e., two SIP-URIs hashing to the same value). The Singh and Schulzrinne approach reduces this problem somewhat by using a P2P network for each domain. Further, Singh and Schulzrinne suggest e-mail-based authentication in which a node joining the network receives a password via e-mail. It uses this password to authenticate itself to the network (e.g., to the node previously responsible for the ID space).

### Free Riding

In any P2P system there is a risk of free riding: nodes use services but fail to provide services to the network. Certainly, in a P2P-SIP setting there is a risk of free riding: nodes use the overlay for registration and location service but drop other messages. A huge amount of nodes that “ride free” would eventually result in reduction of the overlay’s availability.

### Anonymity

With P2P-SIP, any node in the overlay can keep track of SIP messages that it routes. Thus, any node in the overlay can keep a profile of registrations and lookups for SIP-URIs the node is responsible for.<sup>7</sup> Furthermore, using join-leave attacks, an adversary can place itself intentionally at any location in the Chord ring to log access to a particular SIP-URI. In general, anonymity for SIP messages can be achieved by using anony-

<sup>6</sup> Remember that in Chord a node can only route to a node with higher ID than the key-ID if this node is the first in its routing table; otherwise it must route to the node with highest ID below the key-ID.

<sup>7</sup> Note that such attacks are also possible in Skype: in a Skype network any node with a public IP address that is not behind a middlebox (e.g., Nat/Firewall) eventually becomes a super node. Super nodes in Skype are used for routing of content for Skype users who are located behind a middlebox. Though Skype traffic is encrypted, logging of traffic data (who called whom for how long) is possible.

mous values in SIP headers. However, certain headers must contain information about the sender/path of a message to successfully route response-messages back to the sender (e.g., Via-header<sup>8</sup>). If recursive routing is used, an option to ensure anonymity within the overlay would be to replace SIP headers at every hop in the overlay: this disguises the originator of the request. A similar approach would be to use a *friend-to-friend* model where nodes only exchange messages with other nodes they trust. A message can be delivered anonymously between two nodes that do not trust each other: if a transitive trust path exists, the message can be sent indirectly using hops between “friends” that trust each other. These “friends” hide each others’ origins before passing on messages in the network. As with client-server SIP, privacy can also be achieved by using a pseudonymity-service, which acts as a back-to-back user agent (B2BUA) [4]. Such a B2BUA would be placed as a node in the overlay and provide services to user agents within or outside of the overlay. It replaces headers in both directions in order to disguise message origin.

### Other Challenges

Some problems for client-server SIP are presumably even harder to solve for P2P-SIP:

- *Reliable and secure emergency services*: Besides the problem of prioritizing signaling for emergency calls in an overlay network, the fundamental task of ascertaining the physical location of users in real-time may be difficult.
- *Lawful interception*: VoIP signaling and content use different paths. Further, there is no predefined route for content. With P2P-SIP, there is not even a predefined route for signaling traffic due to the highly dynamic nature of P2P-networks. Therefore, it appears almost impossible to implement a surveillance system for law enforcement agencies with P2P-SIP.
- *Spam-prevention*: Though spam over Internet telephony (SPIT) is not a problem in VoIP networks yet, it is estimated to be highly attractive for marketers to use VoIP for unsolicited calls in the near future. With P2P-SIP, spam prevention should be implemented at the receiving user agent. Due to the highly dynamic nature of an overlay, the node responsible for a key and the routing path vary over time and may therefore not be relied on for spam filtering.

## Securing P2P-Based SIP Networks

After having investigated the challenges for securing P2P-SIP, possible solutions are discussed in this section. To solve the main problem of authentication in a P2P network, there are generally two options: adding a central authority to the network or employing distributed algorithms for trust and reward. Additionally, we discuss self-certifying approaches and present some concrete improvements for securing P2P-SIP.

### Nonscalable Add-Ons

To secure the assignment of node-IDs, Castro *et al.* suggest certifying node-IDs by certificate authorities (CAs) [13]. The CAs assign IDs to joining nodes at random and prevent nodes from forging node-IDs. SOSIMPLE [2] also proposes the use of a public key infrastructure (PKI) for identity enforcement. Condie *et al.* present an approach for unpredictable node-IDs based on a trusted randomness service [15]. In principle, a trusted authority can provide secure node-ID assignment. Nonetheless, there are some limitations to this approach. First, it is well known that — in practice — PKIs do not scale well. Second, the certificate

<sup>8</sup> In the approach from Bryan *et al.* there is another example, the newly introduced header “DHT-NodeID.”

authorities present central points of attack. The overlay would depend on the PKI for operation (at least for new nodes joining the network, which will happen frequently in any P2P-SIP scenario). Thus, most of the advantages of P2P networking are alleviated if a PKI (or any other central authority) is added.

### *Distributed Trust and Reward*

To keep up with the spirit of P2P computing, a solution for authentication of nodes should rather be scalable and thus distributed itself. Reputation management systems offer a somewhat weaker form of distributed authentication by assigning trust values to nodes. Several reputation management systems for P2P networks have been proposed (e.g., *Eigentrust*). These systems can be used to counter the problem of free riding: only nodes that have a high reputation of delivering services to the overlay may use services from the overlay. However, most research on reputation management in P2P networks focuses on traditional file-sharing systems. Further, most existing systems rely on a central authority and an underlying structured overlay network that facilitates secure routing. Hence — at present — it remains an open problem how to securely integrate a (distributed) reputation management system with P2P-SIP.

### *Self-Certifying Approaches*

In the absence of a central authority, self-certifying identities can be used for authentication of content in an overlay. A self-certifying identity is an identity that can be verified without consulting a (trusted) third party. For instance, the association of an identity to a public key can be verified mathematically if the identity is represented as the hash of the public key. In P2P-SIP, a user could hash his public key to generate a SIP-URI (e.g.,  $[hash(public\_key)]@domain$ ) [18]. The user can then digitally sign its location data (the content stored in the overlay) with his private key. Any node looking up the location can verify its authenticity: it hashes the public key (which is sent along) and compares the result with the first component of the SIP-URI before it uses the public key to verify the signature. However, such an approach has some drawbacks. Using a 160-bit hash function results in SIP-URIs that are not easy to memorize by users. Further, there can be collision attacks on the hash function. Finally, how can a user be sure that the URI he is calling (which consists of a hash value, e.g., an arbitrary sequence) belongs to the desired callee? This assurance has to be solved outside of the overlay, for example, via e-mail or a trusted webpage. Once a user knows that a particular (cryptographic) SIP-URI belongs to a callee, the integrity of the callee's location binding can be verified automatically.

### *Suggestions for Securing P2P-SIP*

Each of the general options for securing P2P-SIP presented above has its advantages and limitations. Depending on the scenario in which P2P-SIP will be used, one or the other approach can be appropriate to secure the overlay. We summarize our discussion by making some suggestions on how to generally improve the security of P2P-SIP:

- Solely the IP address (without port) shall be used for node-ID generation. Although this does not provide full protection against attacks on node-ID generation, we have shown that it is a much better choice than using a combination of IP address and port. Future developments should consider security add-ons as in [15] for node-ID generation.
- Iterative routing using SIP redirect messages shall be used. First, this reduces the risk of denial-of-service attacks in which attackers put heavy load on the overlay by inserting fake requests. Second, it enables nodes to check that they get closer to the desired key at every routing hop.

- In addition to iterative routing, we suggest using advanced routing strategies, as proposed in [17], which return the whole routing table at each iteration. This enables nodes not only to be in control at any routing hop, but also to decide on the next routing hop.
- For user agents with a need for anonymity, we advise the use of a pseudonymity service instead of recursive routing (we prefer iterative routing because it is a better choice to detect man-in-the-middle attacks).
- For the protection of content stored in the overlay, we suggest self-certifying SIP-URIs [18]. Such a solution protects the integrity of URI/location bindings without using a central authority. Further, it is compatible with any existing (client-server) SIP entity or any future DHT used for P2P-SIP because the cryptographic add-on is encoded within the SIP-URI. An integration into the current approaches for P2P-SIP is therefore possible.

When comparing the approaches for P2P-SIP from a security perspective, one can see that each approach can be improved. Singh and Schulzrinne use only the IP address for node-ID generation, but also use recursive routing (at least in their prototype implementation). On the other hand, Bryan *et al.* use iterative routing, but their node-ID generation can be attacked with little effort.

### *Conclusion*

We have examined approaches for using a structured P2P network for SIP user-registration and location lookup. Using prior work on the security of structured overlay networks, we have identified security challenges for P2P-SIP. Further, we have shown some specific attacks and discussed possible countermeasures. Finally, we made explicit suggestions how to improve the security of P2P-SIP by use of the following: solely the IP address for node-ID generation, iterative routing, advanced routing strategies that utilize complete routing tables of other nodes, a pseudonymity service for privacy needs, and self-certifying SIP-URIs for protecting the content stored in the network. Future work on the security of P2P-SIP will include the development of advanced routing strategies and the investigation of other schemes/DHTs for improved node-ID generation. Furthermore, measurement on how advanced strategies for overlay routing will delay SIP user location is needed. It remains an open question how to integrate a fully distributed, secure reputation management system in P2P-SIP. In addition, secure bootstrapping without using a central authority for certifying bootstrap nodes is still an open challenge.

### *References*

- [1] K. Singh and H. Schulzrinne, "Peer-to-Peer Internet Telephony using SIP," *Proc. Int'l. Wksp. Network and Op. Sys. Support for Digital Audio and Video*, Stevenson, WA, June 2005, pp. 63–68.
- [2] D. A. Bryan, B. B. Lowekamp, and C. Jennings, "SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System," *Proc. Int'l. Wksp. Advanced Architectures and Algorithms for Internet Delivery and Apps.*, Orlando, FL, June 2005.
- [3] J. Rosenberg *et al.*, "SIP: Session Initiation Protocol," RFC 3261, 2002.
- [4] J. Posegga and J. Seedorf, "Voice over IP: Unsafe at Any Bandwidth?," *Proc. Eurescom Summit 2005 — Ubiquitous Svcs. and Apps.*, Heidelberg, Germany, Apr. 27–29, 2005, VDE Verlag, pp. 305–14.
- [5] D. Liben-Novell, H. Balakrishnan, and D. Karger, "Analysis of the Evolution of Peer-to-Peer Systems," *Proc. 21st Annual Symp. Principles of Distrib. Comp.*, Monterey, CA, July 21–24, 2002.
- [6] I. Stoica *et al.*, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *IEEE/ACM Trans. Net.*, vol. 11, no. 1, Feb. 2003.
- [7] K. Singh and H. Schulzrinne, "SIPpeer: a Session Initiation Protocol (SIP)-based Peer-to-Peer Internet Telephony Client Adaptor," White paper, Comp. Sci. Dept., Columbia Univ., Jan. 2005, <http://www1.cs.columbia.edu/~kns10/publication/sip-p2p-design.pdf>
- [8] D. A. Bryan, B. Lowekamp, and C. Jennings, "A P2P Approach to SIP Registration and Resource Location," Mar. 2006, work in progress, draft-bryan-

sipping-p2p-02, <http://www.ietf.org/internet-drafts/draft-bryan-sipping-p2p-02.txt>

- [9] A. Johnston and H. Sinnreich, "SIP, P2P, and Internet Communications," Mar. 2006, work in progress, draft-johnston-sipping-p2p-ipcom-01, <http://www.ietf.org/internet-drafts/draft-johnston-sipping-p2p-ipcom-02.txt>
- [10] E. Shim, S. Narayanan, and G. Daley, "An Architecture for Peer-to-Peer Session Initiation Protocol (P2P SIP)," Feb. 2006, work in progress, draft-shim-sipping-p2p-arch-00, <http://www.ietf.org/internet-drafts/draft-shim-sipping-p2p-arch-00.txt>
- [11] K. Singh and H. Schulzrinne, "Using an External DHT as a SIP Location Service," Columbia Univ. tech. rep. CUCS-007-06, Feb. 2006.
- [12] J. R. Douceur, "The Sybil Attack," Revised paper, 1st Int'l. Wksp. Peer-to-Peer Sys., LCNS, vol. 2429, Cambridge, MA, Mar. 2002.
- [13] M. Castro *et al.*, "Secure Routing for Structured Peer-to-Peer Overlay Networks," Proc. 5th Symp. Op. Sys. Design and Implementation, Boston, MA, Dec. 2002.
- [14] C. Scheideler, "How to Spread Adversarial Nodes?: Rotate!," Proc. 37th Annual ACM Symp. Theory of Comp., Baltimore, MD, pp. 704-13.
- [15] T. Condie *et al.*, "Maelstrom: Churn as Shelter," UC Berkeley tech. rep. UCB/ECS-2005-11, Nov. 2005.

## EDITOR'S NOTE / *continued from page 2*

helpful in gaining an appreciation of the challenges in this important paradigm-shifting area in telecommunications.

If you would like to contribute to our magazine on this or other networking-related topics, please visit <http://www.comsoc.org/pubs/net/ntwrk/authors.html> for submission guidelines. If you would like to guest edit a Special Issue for our magazine, please contact me or visit [http://www.comsoc.org/pubs/net/ntwrk/special/special\\_prop.html](http://www.comsoc.org/pubs/net/ntwrk/special/special_prop.html) for information on how to prepare and submit a proposal for a Special Issue. As always, you may provide your feedback on any aspect of our magazine by contacting me at [bisdik@us.ibm.com](mailto:bisdik@us.ibm.com). I would very much

- [16] M. Srivatsa and L. Liu, "Vulnerabilities and Security Threats in Structured Overlay Networks: A Quantitative Analysis," Proc. 20th Annual Comp. Sec. Apps. Conf., Tucson, AZ, Dec. 6-10, 2004, pp. 251-61.
- [17] G. Danezis *et al.*, "Sybil Resistant DHT Routing," 10th Euro Symp. Res. Comp. Sec., Milan, Italy, Sept. 12-14, 2005, vol. 3679, pp. 305-18.
- [18] J. Seedorf, "Using Cryptographically Generated SIP-URLs to Protect the Integrity of Content in P2P-SIP," 3rd Annual VoIP Security Wksp., Berlin, Germany, June 2006.

## Biography

JAN SEEDORF ([seedorf@informatik.uni-hamburg.de](mailto:seedorf@informatik.uni-hamburg.de)) received a B.Sc. degree in 2001 and a Diploma degree in 2002 in computer science from the University of Hamburg, Germany, with special focus on IT security. Since 2002 he has been a research assistant with the Computer Science Department at the University of Hamburg, where he is conducting teaching and research within the Security in Distributed Systems group, working toward a Ph.D. degree. His current research focuses on the security of voice over IP systems, and the security of peer-to-peer networks. In particular, he is interested in SIP, and the security of structured overlay networks and distributed hash tables.

like to hear your thoughts on any part of this editorial, on how we can further publicize the magazine's distinctive features and objectives, and ultimately on how we can make it more appealing to you, our valued readers.

## References

- [1] "Telephia Reports 4.1 Percent of Online U.S. Households subscribe to a VoIP Telephone Service, Up from 3.1 Percent in Q1 2006," Telephia press release, July 21, 2006.
- [2] "Carrier VoIP Market Hits New High in 2005 at \$2.5B," Infonetics Research press release, July 31, 2006.
- [3] "Yankee Group Expects VoIP Market to Reach \$3.3 Billion by 2010," Yankee Group press release, Jan. 31, 2006.
- [4] "Yankee Group Research Finds VoIP Adoption in Contact Centers on Accelerated Rise," Yankee Group press release, Aug. 14, 2006.

### VOICE RECORDING SYSTEM



Call Management Utility

**Monitor & Log Calls over T1 / E1 / VoIP Trunks**

- ▶ Search Specific Calls
- ▶ Log Call Records/Voice
- ▶ Filter Incoming Calls Real-time
- ▶ Monitor/Playback Calls Real-time
- ▶ Statistics of Completed/Filtered Calls
- ▶ Playback/Filter/Analyze Completed Calls

 **GL Communications Inc.**  
Comprehensive Telecom Test Solutions  
☎ 301-670-4784 ✉ [info@gl.com](mailto:info@gl.com) 🌐 [www.gl.com](http://www.gl.com)