2-2 Design and Phase-1 Development of Secure Overlay Networks

KADOBAYASHI Youki, NAKAO Koji, and TAKIZAWA Osamu

Recently, overlay networks are gaining attention as a means to eliminate single point of failure in the application layer. This paper focuses on secure routing in overlay networks. A threat model is presented, along with our countermeasure proposal. Our proposal consists of blind forwarding, distributed trust anchor, and probabilistic testing. Unlike previous contributions on this topic, our proposal is based on trust anchor. Furthermore, we attempt to reconstruct the threat model, based on the insights gained from previous contributions. We also briefly describe our phase-1 development efforts of a software framework that implements essential functions of secure overlay. Our software framework attempts to address application-specific threat models.

Keywords
Overlay networks, Secure overlay, Blind forwarding

1 Introduction

Recently, various forms of denial of service attacks against web servers and other critical infrastructures are gaining attention. Such attacks target single point of failure, such as routers and servers. As a countermeasure to these attacks, it is important to eliminate single point of failure in every layer of the OSI 7-layer reference model.

This research focuses on overlay networks as a means to bypass single point of failure in lower layers. It also provides fundamental solution to single point of failure in the application layer. “Overlay network” is an overarching terminology for application-layer networking among end systems.

While overlay networks can eliminate single point of failure, there are other security issues in overlay networks. For example, prior contributions uncovered that overlay networks are vulnerable to insertion of malicious nodes. In response to these problems, secure overlay networks are being explored, that address known vulnerabilities.

While prior contributions [1]-[4]successfully defined threat models that are perhaps most difficult to defend, they did not come up with effective countermeasures to these threats. We believe that countermeasures, along with threat models, should be addressed in general.

In this paper, we attempt to construct secure overlay networks in a stepwise manner, based on the fundamental understanding of simple threats. First, we consider securing basic functions of overlay networks, through which we attempt to relate respective problem with existing problem domain. We mainly focus on routing security in overlay networks and present threat models and countermeasures, since we consider routing security to be a new problem domain.

Next, we point out that different threat models should be defined for different applications. We are constructing software frameworks for secure overlay, that can be adapted to different threat models. Finally, we describe the result of our phase-1 development.

2 Overlay networks

In this section, we briefly describe overlay networks. Overlay networks are built on top of network layer, and it is typically implemented in the application layer. There are variety of algorithms for overlay networks, with different rendezvous, location, and routing characteristics. Hereafter, we collectively call rendezvous, location, and routing as the three aspects of overlay networks.

In Chord[5], for example, nodes participating in the overlay and resources stored in individual nodes are mapped onto a single ring, using a hash function, e.g., SHA-1. By exploiting order relation on the ring, routing can be implemented in a efficient manner. In other words, by mathematically defining order relation of both nodes and resources on the single ring, traditional directory functions – mapping resources to nodes – can be replaced by the hash function, thereby facilitating resource lookup. Also, such an order relation enables efficient routing; the implicit rule makes it possible to skip intervening nodes during the lookup process.

In Chord, a new node can join the ring by sending join request to arbitrary nodes on the ring. Also, it can access nodes and/or resources by specifying an identifier on the ring. So far, variety of routing algorithms are invented with variety of topologies, since there are large degree of freedom in the design of ID space: binary tree, mesh, and other structures have been employed so far[6][7].

Next, we describe route selection model in the overlay networks. While IP networks employ single route selection model (i.e., longest prefix matching algorithm), overlay networks have a variety of route selection models, since proximity in the ID space has nothing to do with proximity in the real network. If we employ single algorithm that only consider topological proximity in the ID space, suboptimal route will be selected from the viewpoint of network layer.

In order to reduce such inefficiency, three proximity-aware route selection models are proposed in the past[8]:

- PNS (Proximity Neighbor Selection): neighbor nodes in the routing table are selected according to node proximity, during the routing table generation process.
- PRS (Proximity Route Selection): multiple neighbor nodes are selectively used in the route selection process, based on node proximity.
- PIS (Proximity Identifier Selection): node identifier is assigned according to node proximity.

The Chord algorithm employs most simple route selection model which does not consider node proximity in lower layers; neighbor nodes in the routing table are computed by the order relation in the ID space. PNS, PRS and PIS are enhancements which can be applied to most of overlay routing algorithms.

Next, we describe the service model of overlay. Three service models have been proposed so far: DHT (Distributed Hash Table), DOLR (Decentralized Object Location and Routing) and CAST[9]. DHT implements hash table on top of overlay, and it is suitable for applications which share (key, value) pairs among participating nodes. In DHT, any participating nodes in a overlay can perform put/get operation on any (key, value) pairs. DOLR is a kind of distributed directory service; it delivers message to the nearest object with the specific ID. In DOLR, many objects can have the same ID within single overlay. CAST is a service model for group communication. It enables nodes to join and leave from group, and multicast/anycast messages within group members. The prior contribution[9] points out that these service models can be implemented by the same generic lower layer called KBR (Key-Based Routing).

3 Related work

In this section, we outline security requirements for each of the three aspects. Meanwhile, we attempt to relate each requirement with existing problem domains and prior contributions.

First, we consider security requirements in the rendezvous. During rendezvous, validation of authenticity, countermeasure against abuse, and countermeasure against denial-of-service attacks are required. In order to secure authenticity, we can establish trust anchor outside of the overlay; for example, PKI (public key infrastructure), PGP (pretty good privacy), and IC cards can be used for this purpose. Known techniques to protect systems against abuse are Turing tests, creditcard billing, hardware token, and incentive techniques[10]. In order to protect systems against DoS attacks, several techniques are proposed to date: memory-bound function[11], single-signon authentication servers employing threshold cryptography [12], and redundant configuration of receiver nodes[13].

During resource location, confidentiality, integrity, and availability are essential security properties; these properties can be implemented by symmetric cryptography, hash function, and replication or erasure coding, respectively.

Essential security properties in routing algorithms are accountability, reliability, and availability. In prior contributions[1]-[4], various forms of threats are pointed out, including route insertion and Byzantine failure during message delivery. However, none of the prior contributions satisfy all of the requirements.

Other security properties have been investigated in prior literatures. For example, anonymity has been dealt with in Herbivore[14], and LOCKSS[15] addresses persistence and durability in distributed data storage.

In summary, it is necessary to combine security functions for all three aspects of overlay networks, in order to establish security in overlay networks. Prior contributions lack such a perspective, however. In this paper, we attempt to reconstruct threat model from this perspective, and propose countermeasures in the routing algorithm.

4 Threat model of overlay routing algorithms

We assume the following threat model to overlay routing algorithms.

- T-1: A few nodes may make adversary actions, but many nodes behave correctly.
- T-2: Many nodes can be DoS/DDoS targets.
- T-3: Adversary can eavesdrop traffic at one point, but not at many points.

As a result of T-1, a few nodes may forward messages to wrong destination, deny message forwarding, or inject incorrect routes. As a result of T-2, many nodes may become unavailable.

The threat model of overlay networks varies depending on the openness of overlay. For *private* overlays with limited membership, only T-2 and T-3 will be present if the overlay does not contain any adversary nodes. T-1 will be present only if a few adversary nodes exist within the private overlay. In *public* overlays, on the other hand, many nodes may join the overlay and act as adversary nodes.

Note that we limit the complexity of the problem by limiting the number of adversary nodes in T-1. We argue that we can deal with more broader class of problems by limiting the number of adversary nodes. Many techniques can be employed to limit the number of adversary nodes; Turing tests and credit-card billing are two examples of such techniques, as outlined in the Section 3. This is in sharp contrast to threat models in past literatures, e.g., Sybil attack, where the number of adversary nodes are not limited.

In addition, by employing the *blind forwarding* that we describe in the following section, we can reduce the incentives for adversary nodes to attack overlay routing algorithms.

5 Design of secure overlay routing

In this paper, we attempt to establish accountability, reliability, and availability in overlay routing algorithms. In order to implement these desirable properties, we make three proposals in this section. First, we propose *blind forwarding*, assuming that multiple competing communities are multiplexed onto single overlay. Next, we argue that subjects and objects of access alone cannot establish accountability and reliability; based on this argument, we propose a technique to form the distributed trust anchor within the ONOC (Overlay Network Operation Center). Next, we propose a technique to detect attacks to message forwarding by probabilistically sending a probe packet after the individual message.

5.1 Blind forwarding

Hereafter, we call the technique to forward messages without knowing the IP address, hostname or AS number of both source and destination node *blind forwarding*. We call the network layer attribute (IP address, hostname and AS number) collectively as *network attribute*. First, we describe the rationale of blind forwarding.

Generally speaking, infrastructure for specific community will be susceptible to attacks from other communities. In contrast, single overlay shared by multiple competing communities might not attract attacks. In other words, it is important to creat situation such that attacking the overlay results in attacking part of attacker's communities.

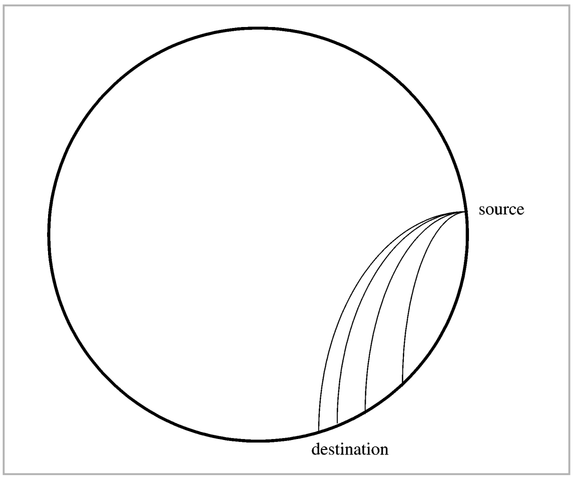


Fig.1 Iterative message forwarding

For example, in the case of Internet denial-of-service attacks, one can determine if particular host can be attack target by network attribute. As a result, tension between communities turns into such attacks, e.g., Japanese servers being attacked from other countries. In overlay networks, one might be able to solve this problem; for example, one can construct overlay networks such that network attributes are hidden from participating nodes, even if the node knows the overlay topology.

Denial-of-service attacks against message forwarding makes sense if the network attributes of both message sender and receiver are known. In other words, such attacks do not make sense if network attributes are hidden from intervening nodes during message forwarding. In order to thwart selective denial-of-service attacks with blind forwarding, we must implement message forwarding in a recursive manner.

In overlay networks, messages can be forwarded either iteratively or recursively. In iterative message forwarding, a node initiates a message to next-hop node, which returns a node closer to destination. The initiating node repeats this process until the message reaches the destination node which has the solicited resource (Fig.1). In this case, messages are sent from initiating node in a star-shaped topology. As a result, the network attribute of initiating node is revealed to all of the responding nodes.

In contrast, recursive message forwarding avoids disclosure of network attribute; the network attribute of initiating node is revealed only to the next-hop node. The initiating node sends a message to the next-hop node, which in turn forwards the message to its next-hop node, based on its own routing table. This process is repeated recursively until the message reaches the destination node. As a result, messages are sent from the initiating node to the destination node like a stepping stone (Fig.2).

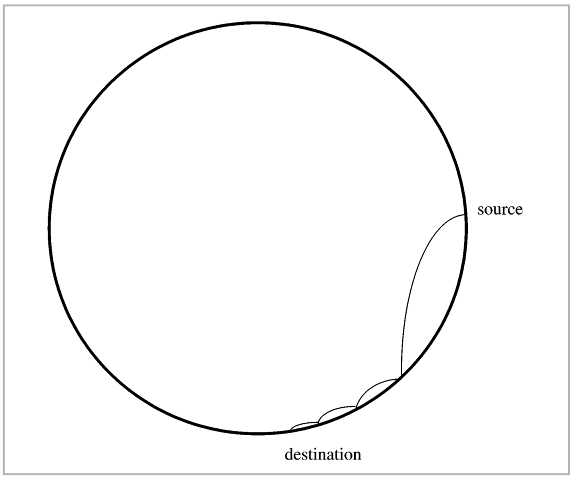


Fig.2 Recursive message forwarding

5.2 Distributed trust anchor

We argue that subjects and objects of access alone cannot accomplish accountability, reliability and availability in overlay networks. We assume the presence of ONOC (Overlay Network Operation Center) in this paper. In order to detect adversary node, ONOC aggregates reports from individual nodes and compares its integrity with reports from both preceding and following nodes. Also, accountability can be secured by logging reports from multiple nodes at ONOC. Furthermore, by computing overall availability at ONOC, ONOC can instruct participating nodes to perform various countermeasures: 1) unreliable nodes can be excluded from overlay, and 2) storage redundancy can be increased by replication or erasure coding.

Next, we discuss the configuration of ONOC. If ONOC is a single node, it can be single point of failure and it will hamper the availability of overlay. Therefore, it is natural to construct ONOC as a private overlay (Fig.3). Since ONOC is not a public overlay, it is difficult to mount DoS attacks against ONOC from outside. We will describe DoS countermeasure at ONOC in the next section.

5.3 Probabilistic probing

Ideally, in order to secure accountability, availability and reliability, ONOC should monitor all packets. This is not realistic from the standpoint of communication overhead and storage complexity. In this section, we describe probabilistic probing.

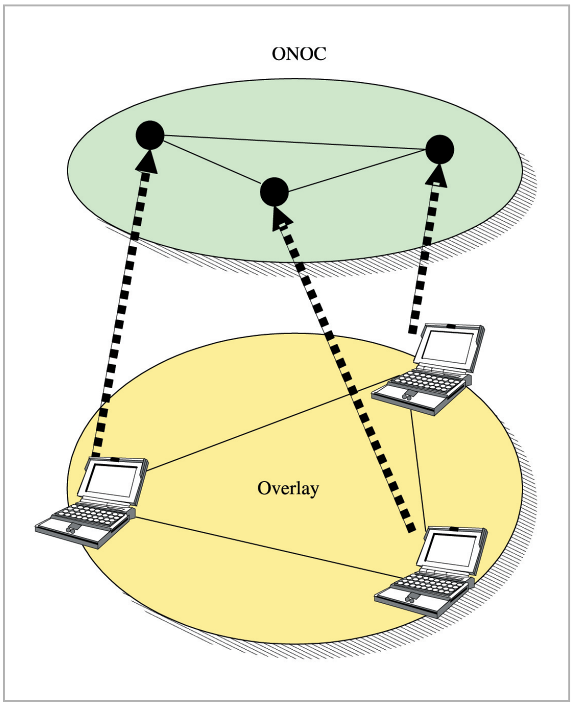
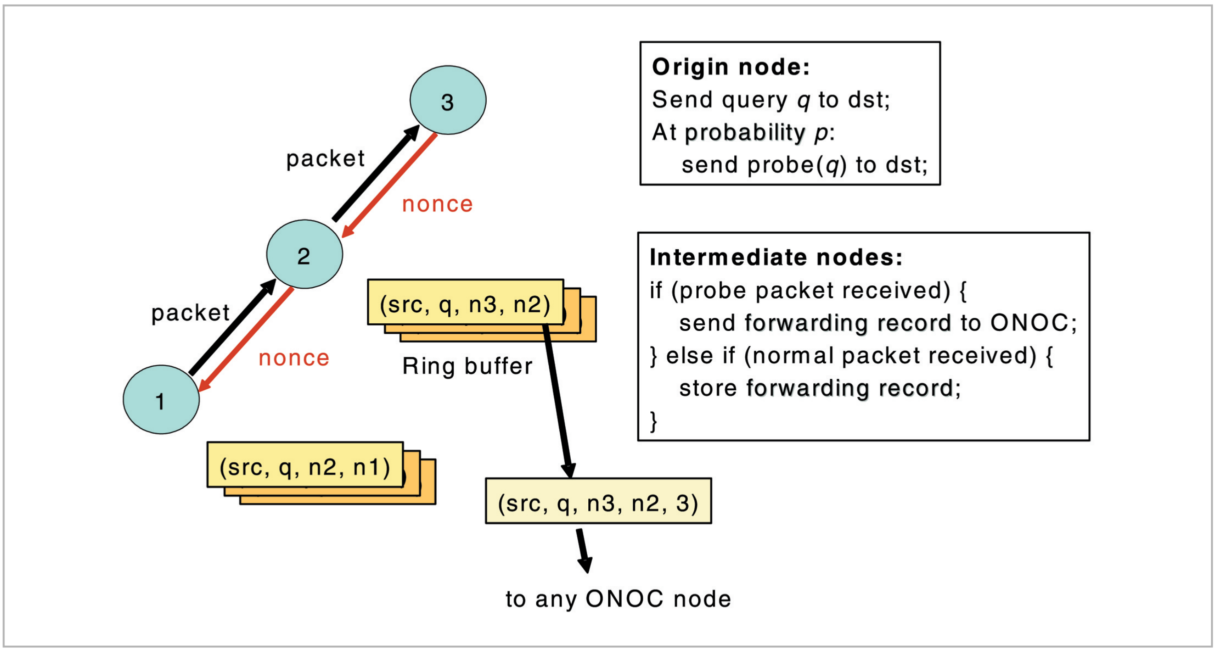


Fig.3 Relationship of overlay networks and ONOC

Nodes cannot misbehave by either discarding messages or forwarding messages to wrong destination, since each node cannot know the probability of receiving probe packet in advance, and whenever the node receives probe packet, it has to prove that it has forwarded the preceding message. While we focus on query messages in the following descriptions, various messages for routing algorithms, e.g., node addition or deletion, can be secured in a similar manner.

Probabilistic probing can be implemented by the initiator node sending probe packet after the query message at certain probability. The probe packet uses the same source and destination address as the query message, and it is sent after certain period. All intervening nodes sends forwarding records to ONOC upon receiving the probe packet. Probe packet specifies target message by message ID. Each node stores forwarding record of individual messages in a ring buffer.

Next, we discuss the interval of original packet and the probe packet. The transmission



```
Origin node:
Send query $q$ to dst;
At probability $p$:
    send probe($q$) to dst;
```

```
Intermediate nodes:
if (probe packet received) {
    send forwarding record to ONOC;
} else if (normal packet received) {
    store forwarding record;
}
```

Fig.4 Forwarding report and probe packet

delay of probe packet should be adequately configured so that the forwarding record remains in the intervening nodes. The delay should not be zero, since zero delay permits adversary nodes to selectively transmit messages with succeeding probe packet, while discarding others.

Next, we describe the contents of forwarding record. In order to construct forwarding record, nodes are required to exchange the following information between itself and next-hop node. Whenever a node sends a message to next-hop node, it receives a random value. Likewise, whenever a node receives a message from the preceding node, it sends a random value. In this way, all nodes records the four tuple (src, query-id, next-hop-nonce, my-nonce) in the ring buffer, which is actually the forwarding record. Whenever a node receives probe packet, it can prove that it did not block message forwarding by sending forwarding report to the ONOC. The contents of forwarding report is (src, query-id, next-hop-nonce, my-nonce, next-hop) (Fig.4). Initiating node sends the destination address along with the source address to the ONOC. Destination node sends reports with next-hop set to blank.

ONOC can verify that messages are not blocked by comparing the integrity of two values: next-hop-nonce of preceding node, and the my-nonce of next-hop node (Fig.5(a)). Whenever successive forwarding report does not come from certain node and beyond, ONOC can detect that either the message is discarded or the corresponding probe packet is discarded at the preceding node. Also, one can detect message forwarding to wrong next-hop, by checking order relation of node ID and destination node ID; such order relation is defined in the overlay routing algorithms.

It is possible that multiple nodes may collude and forge a forwarding report. In this case, colluding node's next-hop node can be incorrectly determined as a misbehaving node. As a solution to this problem, nonce can be generated by the time-released key chain. Since ONOC can determine the integrity of nonce from the same node by simply applying hash function to previous nonce and comparing it with the current nonce, ONOC can detect forgery of next-hop-nonce (Fig.5(b)).

Next, we describe DoS countermeasure in the ONOC. Each node sends H(my-nonce, my-hop) along with the forwarding report which contains (next-hop-nonce, next-hop), where H is a hash function. In this way,

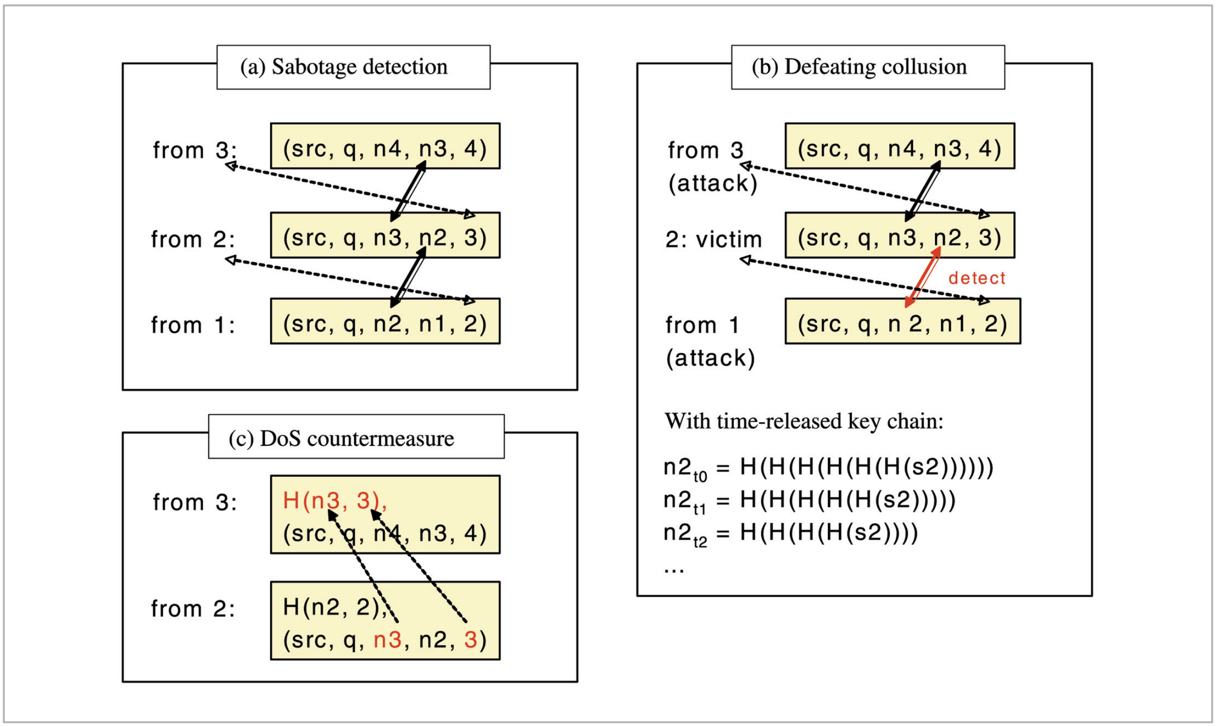


Fig.5 Sabotage detection with forwarding report

ONOC can compute the hash value with (next-hop-nonce, next-hop) of the previous hop and compare it with the hash value from current hop, easily discarding DoS packets (Fig.5(c)).

Likewise, by reusing the nonce contained in the forwarding report at the ONOC, it can secure authenticity of notification from ONOC to participating nodes. Using such a notification, ONOC can instruct overlay nodes to control the degree of replication or redundancy of erasure coding, making it possible to control availability.

6 Toward implementation of secure overlay

In Section **4**, we discussed threat model in the overlay routing algorithms, and in Section **5**, we made a proposal to secure accountability, reliability and availability in the overlay routing algorithms. In addition, security in rendezvous and location are equally important, as we discussed in Section **3**.

Techniques that can be employed in the rendezvous and location are restricted by both application and platform. For example, in electronic appliances, suitable anti-abuse methods for joining overlay would be hardware token; Turing tests might not be suitable. In contrast, for instant-messaging purposes, Turing tests and incentive techniques would be required as a countermeasure to SPIM (SPAM in IM). For application-layer multicast with unlimited membership, integrity of content will be required, while confidentiality of content might not be required. On the other hand, application-layer multicast for groupware purposes will require both confidentiality and integrity.

As we have seen above, we need to deal with different threat models in different applications; it is important for applications to be able to choose cryptographic mechanisms and authentication mechanisms from variety of choices. Currently, NICT security advancement group is constructing a software framework that provides elementary functions of secure overlay. In the following section, we describe the result of our phase-1 development effort.

6.1 Phase-1 development

This framework implements a software for conducting research on secure overlay networks; it can be considered a digital facility for research and development. Therefore, the techniques described in the previous section can be implemented on top of this framework. Also, this framework can be a software bus, making it possible to circulate proposed mechanisms as a working code among research communities. It is worth noting that prior contributions[1]-[4]did not have proof-of-concept implementations, making it difficult for third parties to verify the results.

Our software framework is written in the C++ language, and it runs on major operating systems, including FreeBSD, Linux, MacOS X, and Windows. Also, it is light-weight compared to other overlay toolkits written in Java, making it possible to run 60 nodes on a personal computer. Therefore, it is possible to construct massive overlay networks within a relatively small PC cluster. In order to maintain portability in C++, we used ACE as the communication library, Crypto++ as the cryptographic and hash algorithm library, Qt as the GUI library. Using these existing, portable software frameworks as a substrate, we achieved good portability without using expensive run-time harness like Java VM.

Also, this framework implements Chord and Pastry as the overlay routing algorithms, lowering the barrier to conduct secure overlay research. Also, it implements hybrid routing technique[3] and diffusive communication among multiple routes[14].

Furthermore, this framework is designed using KBR (Key-Based Routing) API, making it possible to implement various service models other than DHT, e.g., DOLR and CAST. Currently, DHT is implemented and put/get operations to (key, value) pair are provided. Apart from DHT, a simple API for send/receive is implemented, with which publish/subscribe model can be implemented easily. Our protocol design makes it possible to use multiple service models on top of single overlay.

Confidentiality and integrity can be added to applications without making major modifications to application code itself, since cryptographic functions and hash functions can be applied as a filter to data stream. In Crypto++ library, AES, IDEA and RC5 are available as cryptographic algorithms, and SHA-1, SHA-384 and MD5 are available as hash algorithms, among others. This framework can employ arbitrary bit length for node identifiers, detaching itself from potential weakness in a particular hash function, e.g., SHA-1.

7 Conclusion

We outlined the security functions of overlay networks from three aspects of rendezvous, location, and routing. We pointed out that security functions from these three aspects should be combined to solve security issues in overlay networks. We dealt with security in overlay routing by constructing threat models and elaborating techniques for countermeasure. Our proposal consists of blind forwarding, ONOC as a distributed trust anchor, and probabilistic probing. Since applications have different security requirements in rendezvous and location, we provided elementary functions of overlay networks as a C++ framework. By combining this framework with existing authentication, cryptographic or hash modules, we believe secure overlay can be brought to applications. We will continue our research efforts for secure overlay, both in algorithms and in implementations.

Reference

1 E.Sit and R.Morris : “Security considerations for peer-to-peer distributed hash tables”, IPTPS, 2002.

2 J.R.Douceur : “The sybil attack”, IPTPS, 2002.

3 M.Castro, P.Druschel, A.Ganesh, A.Rowstron, and D.S. Wallach : “Secure routing for structured peer-to-peer overlay networks”, OSDI, 2002.

4 A.Singh, M.Castro, P.Druschel, and A.Rowstron : “Defending against eclipse attacks on overlay networks”, ACM SIGOPS European Workshop, 2004.

5 I.Stoica, R.Morris, D.Karger, M.F.Kaashoek, and H.Balakrishnan : “Chord : A scalable peer-topeer lookup service for internet applications”, SIGCOMM, 2001.

6 A.Rowstron and P.Druschel : “Pastry : Scalable, decentralized object location and routing for large-scale peer-to-peer systems”, Middleware, 2001.

7 D.M.Petar Maymounkov : “Kademlia : A peer-to-peer information system based on the XOR metric”, IPTPS, 2002.

8 K.Gummadi, R.Gummadi, S.Gribble, S.Ratnasamy, S.Shenker, and I.Stoica : “The impact of dht routing geometry on resilience and proximity”, SIGCOMM, 2003.

9 F.Dabek, B.Zhao, P.Druschel, J.Kubiatowicz ,and I.Stoica : “Towards a common api for structured peer-to-peer overlays”, IPTPS, 2003.

10 M.Feldman, K.Lai, I.Stoica, and J.Chuang : “Robust incentive techniques for peer-to-peer networks”, ACM Electronic Commerce, 2004.

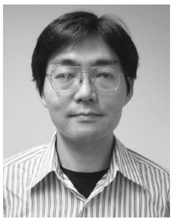
11 M.Abadi, M.Burrows, M.Manasse, and T.Wobber : “Moderately hard, memory-bound functions”, NDSS, 2003.

12 W.K.Josephson, E.G.Sirer and F.B.Schneider : “Peer-to-peer authentication with a distributed single sign-on service”, IPTPS, 2004.

13 A.D.Keromytis, V.Misra and D.Rubenstein : “SOS : Secure overlay services”, SIGCOMM, 2002.

14 S.Goel, M.Robson, M.Polte and E.G.Sirer : “Herbivore : A scalable and efficient protocol for anonymous communication”, Technical Report TR2003-1890, Cornell University Computing and Information Science, 2003.

15 P.Maniatis, M.Roussopoulos, T.Giuli, D.S.H.Rosenthal, M.Baker, and Y.Muliadi : “Preserving peer replicas by rate-limited sampled voting”, SOSP, 2003.



***KADOBAYASHI Youki**, Ph.D.*

Expert Researcher, Security Advancement Group, Information and Network Systems Department

Overlay Networks, Network Security

NAKAO Koji

Group Leader, Security Advancement Group, Information and Network Systems Department

Information Security

***TAKIZAWA Osamu**, Ph.D.*

Senior Researcher, Security Advancement Group, Information and Network Systems Department

Contents Security, Telecommunication Technology for Disaster Relief