Bifrost: A Novel Anonymous Communication System with DHT

Masaki Kondo*, Shoichi Saito*, Kiyohisa Ishiguro*, Hiroyuki Tanaka*, and Hiroshi Matsuo*

*Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, Aichi, Japan
Email: m_kondo@matlab.nitech.ac.jp, shoichi@nitech.ac.jp

Abstract

An immense amount of information is processed on the Internet due to its spread, increasing the severity of such problems as the disclosure of personal information; privacy protection is required. Research to protect anonymity has become crucial. Anonymous communication systems must consider a sudden breakaway of nodes. However existing systems isn't considering this enough. This paper proposes separating a node management layer from an anonymous communication layer. A novel anonymous communication system is realized by a node management layer that uses Chord, which is a distributed hash table, and the anonymous communication layer uses multiplex encryptions.

Index Terms

Anonymous routing, DHT, multiplex encryption, Chord

1. Introduction

As the Internet continues to spread, various services are being provided on it. Some services need high confidentiality, for example, medical consultation services and e-voting. The content of messages remains secret, but encryption can't conceal the identities of sender and receiver. Network administrators and communication recording systems can observe much information that includes sender and receiver IP addresses, traffic, frequency, and so on. At worst, the possibility exists that the sender and receiver information are revealed due to administrator carelessness or malware infection. Cases like e-voting need strong concealment so that voters can anonymously cast ballots. Anonymous communication protocols are needed to satisfy those requirements. In this paper, we propose Bifrost, a novel anonymous protocol with a distributed hash table (DHT). Anonymous communications must meet the following three requirements [1]:

- Sender anonymity: unspecified sender
- Receiver anonymity: unspecified receiver
- Untraceability: flow of data untraceable

These properties are hereinafter called **anonymity**, and communications with *anonymity* are called **anonymous communication**. Furthermore untraceability can be met if sender anonymity and/or receiver anonymity are satisfied. To realize such *anonymity*, such information as the receiver's IP address and routing information must be encrypted and included in messages.

In Section 2, we describe Tor, which is an existing technique. Section 3 introduces the details of Bifrost, our proposed anonymous communication system. Section 4 provides evaluations of Bifrost, anonymity, security, and performance analysis. Section 5 concludes this paper and points out future directions.

2. Related work

2.1. Tor

Tor [2], which is an anonymous public communication system, has been developed as the next generation version of Onion Routing [3]. In Tor, each relay node can only learn the two IP addresses located before and after it on the communication route. Tor realizes sender and receiver anonymity and untraceability using multiple encryption.

A multiple encryption message (Ms) is obtained by $Ms = IP_1||K_1(IP_2||K_2(IP_3||K_3(IP_4||K_4(V))))$. Common keys are k_i (i is number of relay nodes, from 1 to 4, and 4 shows the final receiver in this expression), and a message body is V. In addition, A || B shows that A concatenates B, and IP_i shows the IP address of node i.

An Ms encoded by the above expression is sent to the final receiver. All relay nodes decode it and get

978-0-7695-3914-0/09 $26.00 © 2009 IEEE
DOI 10.1109/PDCAT.2009.35

the $nextIP_i$. The final receiver can obtain V, because it receives $K(V)$ as the result of multiply decrypting in all relay nodes. A reply message is built and encrypted in the same operation in reverse order. This operation is developed in Onion Routing. Bifrost uses the same technique for realizing anonymity.

Finally, when a node suddenly secedes, each node tries to rebuild the anonymous route. But Tor doesn't have a way to detect a seceding node. Therefore Tor reconnects all routes once a minute. Consequently the maximum time for reconnect after seceding a node is one minute. And Tor requires $O(R^2)$ messages (R is total of relay nodes).

Bifrost can reconnect in shorter time than Tor, because Bifrost can detect a seceding node and connect a predecessor to a successor of the seceding node. Therefore, Bifrost requires fewer messages than Tor for reconnecting the anonymous route.

2.2. Crowds

Crowds[4] is a technique to obtain anonymity by way of many nodes. The sender sends messages to other Crowds member nodes at the probability of $(1 - p)$. The member who receives the messages similarly transmits it to other members at probability $(1 - p)$ or to the final receiver at probability p. By repeating this processing, the sender becomes anonymous. Crowds does not use multiple encryption of messages. Therefore each node does not decrypt messages. However, Crowds has a problem on receiver anonymity that all relay nodes can know the final receiver.

3. Bifrost Design

Bifrost realizes availability of anonymous route. Features of Bifrost are 1) Dividing anonymous communication system into Node Management Layer by DHT and Anonymous Routing Layer. And 2) Bifrost can keep a backup node of a relay node. This section is described those features.

3.1. Bifrost structure

Bifrost is an overlay network comprised of many nodes and a public key server (PKS) that manages a public key of each node. Each node connects to the overlay network by a participation procedure and registers its public key in the PKS, which relates the node to the public key. A node can request any other node's public key from the PKS. And the PKS offers the keys to the request node.

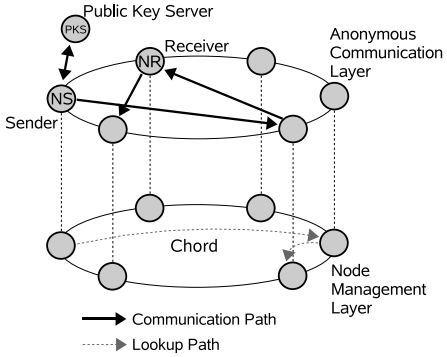

Figure 1. Overview of Bifrost

Figure 1 shows an overview of Bifrost, which is composed of double layers: a node management layer (NML) and an anonymous routing layer (ARL). NML manages all Node IDs and is realized by Chord[5], a DHT. NML controls node participation and secession and maintains a routing table on each node . ARL realizes actual communications including route decisions, route construction, and message encryption.

3.2. Node Management Layer

The NML uses Chord to manage all Node IDs , participation and /seceding procedures, and the routing table of each node. All procedures, which are the participation and secession of nodes, are the same as the Chord algorithm. How to search for the next-hop node in ARL uses the Finger Table of Chord.

The advantage of the separation of NML from ARL is that node management remains independent from anonymity. Since anonymity decreases if a node has much information about the others, a node should only keep minimum information on the other nodes. The minimum information is only an IP address of nodes that are before and after the node itself on the route. Then Bifrost separates NML from ARL to control the nodes without being affected by anonymous route.

3.3. Anonymous Routing Layer

ARL communication includes constructing a route and encoding messages. The ARL does not need to assign a NodeID or to verify node status (participation and secession). On the other hand, all processing concerning anonymous communication is done in ARL. Details of all anonymous routing process are described next.

3.3.1. Overview of ARL. Bifrost messages are multiply encrypted similar to onion routing. A difference between onion routing is how to route and arrange the final receiver's location on the route to improve anonymity. The final receiver, which receives the message contents , is located halfway on the route. This is effective for an analytical attack that analyzes the communication to decrease anonymity because communications after received by the final receiver become dummies. Relay nodes are selected by the sender, who selects them considering their number and round trip time.

Three kinds of messages are used in ARL. The first is a construction message that is only transmitted once when an anonymous communication begins. The second is a data message to carry the contents data. The third is a control message. The construction messages contain keys shared with the relay nodes and the final receiver, and the construction message for the reply (the construction message for the reply is made by the sender). The construction message is multiply encrypted using the public keys of a public key cryptosystem (e.g. RSA), and relays some nodes specified by the sender. When the relay nodes receive, they decode it using the private RSA key and obtain the common key of the sender. Afterwards, the relay nodes and the final receiver memorize the connection information on each anonymous route.

The connection information is composed of a Path ID, an IP address, and a NodeID of the next-hop node in the Chord, and a common key included in the construction message. This information is used to judge which routes are used. Moreover, the common key is periodically renewed because it might be leaked.

The data message contains a general communication date that is encrypted multiply by common keys. Each relay node receives and decodes the data message by referring to the connection information and then relays it to the next node. The control message is used for various anonymous route controls. For instance, route annulment and common key update commands are also encrypted and referred to the connection information and can't control the other routes.

3.3.2. Receiver Area. Bifrost has the following problems: 1) Because routing depends on NML, the construction message communication time is long. 2) A message multiple encryption and decoding time is long. Accordingly Bifrost introduces Receiver Area (RA) to solve these problems. RA is a node group that is a partial continuity space of the total ID space in Chord. RA is composed of nodes from a node of a NodeID specified in the construction message to a node that can decode a construction message header. No node can know a start node and an RA terminal node. Each node in an RA only sends a message to the Successor. All nodes in an RA always receive messages and try to decode message contents to receive messages, but the final receiver can only decode and receive them. The terminal node in an RA can decode the message header and relay a decoded message to the start node of the next RA. The terminal node can obtain the start node of the next RA by decoding the message header. Routing to the next RA is done by NML.

An RA has the following advantages.1) The searching cost for the next node isn't needed in an RA.2) Encryption cost can be reduced because decryption only needs the terminal node. Those advantages are solutions for problems 1 and 2 that are shown in the above.

3.4. Secession of Relay Node and Backup Node

When a relay node secedes in Bifrost, a backup node (BN) is automatically assigned by NML. A BN is a successor of a seceding node. But the BN doesn't have keys owned by the seceding node. A private key and common keys owned by it are needed for reconstructing the route. In Bifrost, these keys are divided into parts [1] and entrusted to a successor and a next-successor before a secession.

When a node suddenly secedes, a predecessor of it can detect within 30 seconds, which is a default parameter of Chord. The predecessor connects and sends a message to the BN, which is a next-successor of the detecting predecessor. And then the BN obtains a pair key from a successor, which is a next-successor of a seceding node. Consequently, the BN can take over the keys and the relay node role from the seceding node. In addition, the new relay node, which was the BN, begins to entrust own keys to a successor and a next-successor immediately. Therefore an anonymous route can be restored without any relation to ARL.

3.5. Outline of Communication

Table 1 shows components of an anonymous route. Figure 2 shows communications for two RAs. Only the necessary relay nodes and the Intermediate Nodes of Chord are shown in Figure 2. First, sender NS sends message MS to start node ID(A_s1) in the first RA based on the Chord algorithm (Lookup Path in

1. We think to select a method of divide from the existing researches in a future work.

Table 1. Component of anonymous route

NS	The sender node
NR	The final receiver node
$A_i (i = 1, 2, ...)$	RA(Message is sent in order of 1,2,...)
$R_{i,j}$ $(j = 1, 2, ...)$	A node relaying to RA(A_i) based on algorithm of Chord
$ID_{min}(A_i)$	Start NodeID of RA A_i
$ID_{max}(A_i)$	Terminal NodeID of RA A_i
As_i	A node managing $ID_{min}(A_i)$
At_i	A node managing $ID_{max}(A_i)$

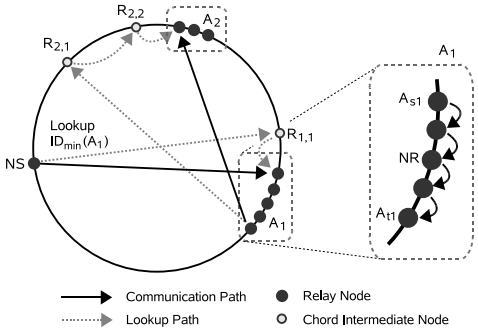

Figure 2. Outline of communication

Figure 2). The message is relayed from A_s1 to the Successor (Communication Path in A_1 of Figure 2). The node in RA tries decoding the header with its own private key. If decoding is impossible, it relays the message to the Successor. If decoding is possible, the node relays the message based on NML because the node can learn $ID_{min}(A_2)$, which is the start node of the subsequent RA. At relaying, the terminal node decodes the message to avoid leaks between sending and receiving messages. This process is repeated, and the message is sent until the next ID in the header becomes null. Moreover, after a message is relayed, a node in an RA tries decoding a body with its private key for the data. If decoding is possible, it becomes the receiver (NR).

The three anonymities are secured based on independent encrypting of a body and a header; the sender prepares a reply message and continues transmission after it is received by the receiver.

4. Verification of Anonymity

This section explains the verifications of Bifrost anonymity. We describe how much information a node can obtain about other nodes. Verification is described in cases of conspiring with two or more nodes, because conspiring nodes can obtain much information than a single attacker node. And we describe communications to PKS.

4.1. Conspiracy Attacks

The possibility is described that conspiring attackers guess the sender and the final receiver by sharing information by considering two or more nodes. First, for a terminal RA node that isn't conspiring, the message is identical in the same RA. However, this is different on different RAs, because it must be decoded on a terminal RA node. Therefore, when the terminal RA node that can decode a message isn't conspiring, an attacker cannot trace the relations between RAs. The attacker can't learn the entire anonymous route, the sender, or the final receiver.

Next, we consider the final terminal node of a conspiring RA. A terminal RA node can discover the connections between this RA and the next RA, because it can decode the message and detect that they are the same before and after decoding the message. If all terminal nodes of RAs conspire, an attacker can learn all RAs. However, even if it learns all the RAs, it cannot discover the final receiver, because the final receiver and the sender have disappeared somewhere on an anonymous route.

4.2. Analysis of Communications to PKS

The sender must acquire the public keys of the relay nodes and the final receiver from the PKS. Therefore, an attacker can learn an anonymous route by observing the communications between sender and PKS. Bifrost prevents attackers by the following methods. 1) Communications between the PKS are protected by using SSL, etc. 2) The sender simultaneously acquires beyond necessary keys.

5. Implementation and Performance

5.1. Implementation

Bifrost is implemented on Overlay Weaver[6], an overlay construction toolkit. Overlay Weaver offers functions for the construction of overlay networks using Chord, routing on overlay networks, sending and receiving messages, etc. Bifrost is developed by adding an anonymous route control process, a multiple encryption process, and the decoding process explained in Section 3.4 to Overlay Weaver. Messages operations, which are used by an Overlay Weaver mechanism, are the implemented functions explained in Section 3. Moreover, each node is connected based on the Chord algorithm offered by Overlay Weaver.

5.2. Performance assessment

Bifrost is assumed to generate delay because some nodes relay a message. Experiments examined the delay time and the transmission rates of Bifrost using an overlay network composed of 32 computers connected by Ethernet (100 Mbps) with a network switch. The computers had Sempron 2800+, memory of 1 GB, and a Linux OS. RSA is used in route generation, AES is used in data communication. Any computer could become a sender and a final receiver. The evaluation parameters of the experiments are message size (Exp. 1), number of RAs (Exp. 2), and the number of relay nodes (Exp. 3). The control message shown in Section 3 was not implemented. An outline of each experiment is shown as follows.

Exp.1 RTT and encryption/decryption times are measured. A route has 16 hops (32 in a round trip), 2 RAs (4 in a round trip), and L_m transmission data sizes ($L_m = 1, 64, 128, 256, 512, 1024, 2048, 4096[KB]$).

Exp.2 Route generation and communication times are measured when data size is 100 KB. A route has 16 hops and 1 to 5 RAs.

Exp.3 Route generation and communication times are measured when data size is 100 KB. A route has five RAs and in a round trip 5 to 30 nodes in steps of 5: 5,10,15,20,25,30.

Figure 3 shows the results of Exp. 1. The route generation times of Experiments 2 and 3 are indicated in (a) and (b) of Figure 4, and the data communication times of 100 KB with a common key are indicated in (a) and (b) of Figure 5

5.3. Consideration of Results

Figure 3 shows that RTT(Round Trip Time) is proportionate to message size. RTT is 2.7 seconds for routes of 2 RAs (4 in round trip) and 16 hops (32 in round trip) and transmitting 1 MB. On the other hand, RTT was about 2 seconds for a 2-hop route and transmitting 1 MB by Chida's anonymous communication method [7]. Bifrost is about 0.7 seconds slower. But Bifrost hop count is eight times larger by Chida's method. Bifrost performance is equal of more than it and offers sufficient performance for Web service.

Figure 3 indicates that about half of the processing time is encryption and decoding. A detailed analysis of the communication time was done in Experiments 2 and 3. Anonymous communication processing is faster than route generation at the overall communication time in the results that compared Figs. 4 and 5, caused by the difference between using the public key at

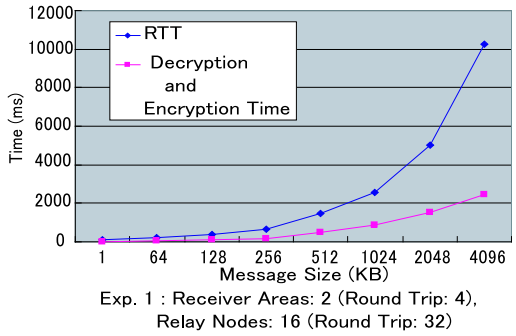

Figure 3. Exp. 1: RTT and Encryption and Decryption Time by Message Size

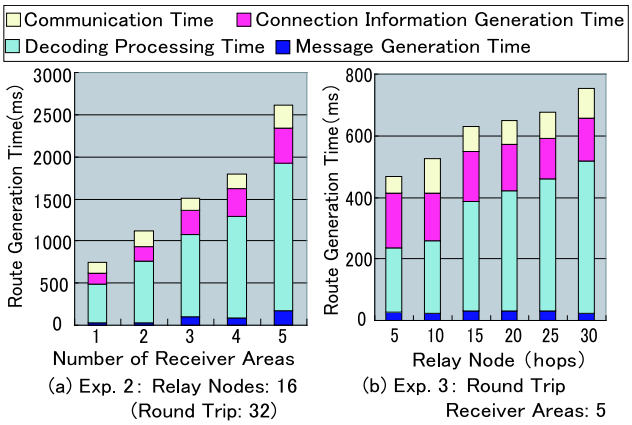

Figure 4. Route Generation Time

route generate and using the common key at data communicate. Comparing (a) and (b) in Figs. 4 and 5, an increase was seen for (a) that changed the number of RAs at the message generation time. On the other hand, no increase was seen for (b) that changed the number of relay nodes because the encryption processing time of the outbound message increased as RAs increased. The number of RAs means the number of multiply encryptions. As for (a) of Figure 5, the decoding time greatly increased compared with (b). Increasing the RAs means the number of nodes (= a terminal RA node) that can decode increases (decoding cost increases). The processing cost of a terminal node of RA is larger than the cost of a general relay node, because the amount of increase at the decoding processing time of (a) in Figure 4 is larger than (b). Therefore, when the number of RAs is increased, the communication delay increases more than increasing the number of relay nodes.

Based on the above, we examined the number of RAs and relay nodes. If an attacker tries to detect the

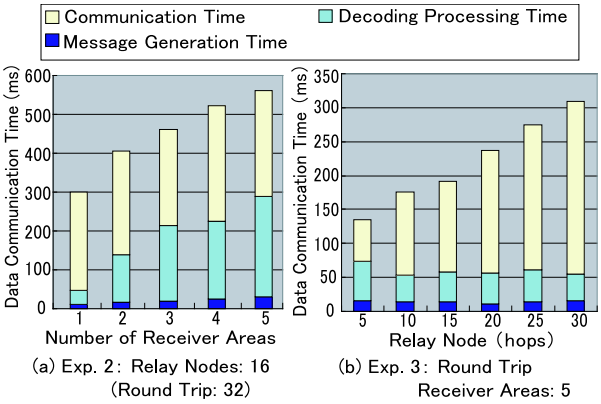

Figure 5. Data Communication Time

final receiver of Bifrost, first the attacker need assume an RA in which the final receiver is, and identify the final receiver in the RA. As described in 3.3, when RAs increase, the presumption of the RA location of the final receiver becomes difficult due to many multiple encryptions. The possibility of the presumption of the final receiver position increases with few RAs.On the other hand, when relay nodes increase, assuming the final receiver becomes difficult because the number of possible nodes that might be the final receiver also increases.

Finally, we consider the possibility of practical use. In its evaluation of Tor, [8] explains: "When the route of 4 hops (number of encryption times is 4) is generated, Tor takes about 7 seconds, and when data communication of 4 hops (number of encryption times is 4) is done, Tor takes about 2 seconds." In our evaluations, Bifrost gives about 2 seconds during route generation and 0.5 seconds during data communication when the route includes 16 hops and four RAs (four encryption times). Because [8] evaluated on the Internet, and our experiment used a LAN, the situations are not equal. However, the possibility of practical use enough.

6. Conclusion

In this paper, we proposed Bifrost, a novel anonymous communication method with two layers, node management with Chord and anonymous communication with multiple encryption. We also described its evaluations. We measured the route generation time, the anonymous communication time, and RTT when the data size is changed. Its performance was very practicable. Measurement results of 3.4 seconds were obtained under conditions of data size 100 KB and 16 hops, and 1 seconds or less was obtained by 1 MB and 16 hops. Senders must choose many RAs for highly anonymous communication and few RAs and many relay nodes in the RAs for high speed and anonymity. Present implementation cannot evaluate on the Internet. We will implement a control message of key exchanges and evaluate its overhead and performance on the Internet.

Acknowledgments

This research is partly supported by the Grant-in-Aid for Scientific Research (#20500064) from Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan, and Grant-in Aid from The Hori Information Science Promotion Foundation, Japan.

References

[1] Pfitzmann A. and Waidner M.: Networks without user observability, Eurocrypt'85, LNCS 219, pp. 245-253 (1986).

[2] Dingledine, R. and Mathewson, N.: Tor: The Second-Generation Onion Router, Pro ceedings of 13th USENIX Security Symposium, pp. 303-320 (2004).

[3] Reed, M.G., Syverson, P.F., and Goldschlag, D.M: Anonymous connections and Onion routing, IEEE Journal on Specific Areas in Communications, Vol. 16, No. 4, pp. 482-494 (1998).

[4] Reiter, M.K. and Rubin, A.D.: Crowds:Anonymity for web transactions, ACM Trans.Information and System Security, pp.66-92(1998).

[5] Stoica, I., Morris, R., Karger, D., Kaashoek, F., and Balakrishnan, H.: Chord : A Scalable Peer-To-Peer Lookup Service for Internet Applications, Proc. 2001 ACM SIGCOMM Conference, pp. 149-160 (2001).

[6] Kazuyuki Shudo, Yoshio Tanaka, and Satoshi Sekiguchi: Overlay Weaver: An Overlay Construction Toolkit, Computer Communications, Vol. 31, Issue2, pp. 402-412 (2007).

[7] Koji Chida, Teruyuki Komiya, Osamu Shionoiri, and Atsushi Kanai: Implementation and Evaluation of Anonymous Networks with a Robust Anonymity Revocation Scheme, IPSJ Technical Report, 2005-CSEC-29, pp. 25-30 (2005). (in Japanese)

[8] Andriy Panchenko, Lexi Pimenidis, and Johannes Renner.: Performance Analysis of Anonymous Communication Channels Provided by Tor, The Third International Conference on Availability, Reliability and Security, pp. 221-228 (2008).