RUSH - An Approach To Improve the Scalability and Security of Hop By Hop Multicast Routing Protocol

K. Raghunathan
Sr. Assistant Professor
Dept. of Information Technology
GVP College of Engg.for Women.
Kommadi, Visakhapatnam, India

V. Lakshmana Rao
Assistant Professor
Dept. of Information Technology
GVP College of Engg.for Women.
Kommadi, Visakhapatnam, India

D. Usha Reddy
Student- B.Tech
Dept. of Information Technology
GVP College of Engg.for Women.
Kommadi, Visakhapatnam, India

P. Roopavathi
Student- B.Tech
Dept. of Information Technology
GVP College of Engg.for Women.
Kommadi, Visakhapatnam, India

***Abstract* - We present RUSH (Rehashing Using Secure Host) technique in this paper, that gracefully integrates the recursive unicast with hash algorithm and key based host identity to achieve scalability and security in HBH (hop by hop multicast routing protocol).In this model, data packets have unicast destination addresses. The key idea is to implement multicast distribution and to simplify address allocation using secure recursive unicast hash trees. HBH adopts the source specific channel abstraction to tackle the address allocation and the sender access control problems. RUSH technique changes the multicast routing model so that only trusted members are able to join the multicast tree. This protects the multicast routing against the branch formed to unauthorized receivers, prevents replay attacks and limits the effects of flooding attacks. It also provides a simple mechanism for distributing encrypted data along the transmission path to receivers.**

Keywords –RUSH, Multicast, Hash Function, AES, Multicast Forward Table (MFT), Multicast Control Table (MCT).

I. INTRODUCTION

IP MULTICAST is composed of a service model that defines a group as an open conversation from M sources to N receivers, an addressing scheme based on class-D IP addresses and routing protocols. In IP Multicast, any host can join and receive the data and any host can send the data to multicast group.

The ability to transparently support unicast routers is the main motivation of the hop-by-hop multicast routing protocol (HBH) [1]. HBH uses the unicast infrastructure to do packet forwarding with smaller routing tables, and can identify the group by channel abstraction. Thus, HBH preserves compatibility with IP Multicast as it uses class-D IP addresses for group identification. HBH constructs shortest-path trees (SPTs) instead of reverse SPTs. Consequently, HBH has the capability to provide better routes in asymmetric networks [10]. Additionally, the tree management algorithm of HBH provides enhanced tree stability in the presence of group dynamics and reduces tree bandwidth consumption in asymmetric networks, compared to most of the alternative solutions like: shared trees, application-layer trees, and REUNITE [3].

In order to counter the common threats to multicast communications, we can apply several of the fundamental security services, including authentication, integrity, confidentiality and authorization. A secure multicast session[5] may use all or a combination of these services to achieve the desired security level.

Authentication services provide assurance of a participating host identity. Therefore, the availability and distribution of keys should be restricted to only authorize group members according to the policy of trust established for the session. Authentication mechanisms can identify the source of the key material and provide a means to counter various masquerades and replay attacks that may be launched against a secure multicast session.

Integrity requires the data and control packets originated at an authorized source not to be intercepted or altered while traversing through the multicast tree. The possibility of preventing a denial-of-service attack through the transmission of such packets can be minimized or eliminated.

Confidentiality services are essential in creating a private multicast session. It should also beapplied to key management transactions during the exchange of key material and can be applied to session announcements allowing them to advertise publicly through standard methods while keeping the details of the session private.

Authorization can be implied to only those entities with specific permission that may use or alter the multicast routing tree of a given group after they have been suitably authenticated.

This paper is organized as follows : Section II presents Building a secure HBH Multicast Tree, Clustering in HBH[6] , AES Cryptographic technique in HBH, AES (Advanced Encryption Standard) .Section III describes the RUSH Technique in HBH, HBH Multicast Protocol using RUSH, Routing Tables, Data Forwarding in HBH with security. Section IV Implementation, Avoiding Packet Duplication in HBH, Avoiding Packet Duplication in HBH using RUSH, Comparison of protocols.Finally, Section V concludes the paper.

II. BUILDING A SECURE HBH MULTICAST TREE

Secure HBH has a tree construction algorithm [5] that is able to better treat with the uncertain cases due to asymmetric unicast routes. Secure HBH uses two tables, one MCT and one MFT. MFT stores the address of a next branching nodeinstead of the address of a receiver. The MFT has no *DST*entry. Data received by a branching router, H_B, has unicast destination address set to H_B. A multicast channel in HBH is identified by <*S*, *G*>, where *S* is the unicast address of the source and *G* is a class-D IP address allocated by the source. This can solve the address allocation problem. HBH's tree structure has the advantage of an enhanced stability of the table entries and minimizes the impact of member departures. This is possible because the MFT receiver entry is located at the branching node nearest the receiver. The advantage in HBH is that each data packet received by a branching node produces *n*+1 modified packet copies.

A. *Clustering in HBH*

In Clustering of HBH [6], a receiver's address is maintained by only one node in the group's delivery tree. To multicast a packet, the root sends a copy of the packet to each hash address [9] in its list, which leads to the related sub-trees. Similarly, when a branching node forwards such a packet, it sends a copy of the packet to each receiver in its own list. This procedure continues recursively until packets reach all leaf nodes of the tree, i.e., all the receivers.

The receiver r_i sends *join* (S, r_i) upstream toward the source S and the route is: r_i> H_i> ----- >S .S uses hash algorithm to build sub-trees from the IP addresses of r_i rooted at S (Source Specific Trees) for multicast distribution[2] (Fig 1). It is one of the characteristics that differentiate HBH from other routing protocols.

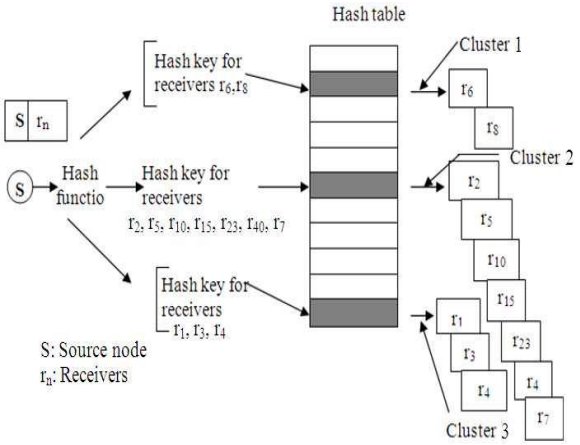

Fig 1: Clustering of routers at Source

B. *AES Cryptographic technique in HBH*

Cryptography is the art and science of keeping information secure from unintended audiences, of encrypting it. Today, cryptography is used to provide secrecy and integrity to our data, and both authentication and anonymity to our communications. Modern cryptographers emphasize that security should not depend on the secrecy of the encryption method (or algorithm), only the secrecy of the keys. The secret keys must not be revealed when plaintext and cipher text are compared, and no person should have knowledge of the key.

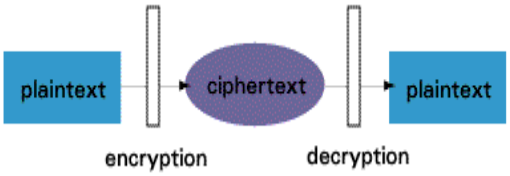

Fig 2: Basic Encryption and Decryption

There are two types of key-based encryption:
1. Symmetric (or secret-key)
2. Asymmetric (or public-key) algorithms.

Symmetric algorithms (Fig: 3) use the same key for encryption and decryption (or the decryption key is easily derived from the encryption key). Symmetric algorithms [10] can be divided into stream ciphers and block ciphers. Stream ciphers can encrypt a single bit of plaintext at a time, whereas block ciphers take a number of bits (typically 64 bits in modern ciphers), and encrypt them as a single unit.

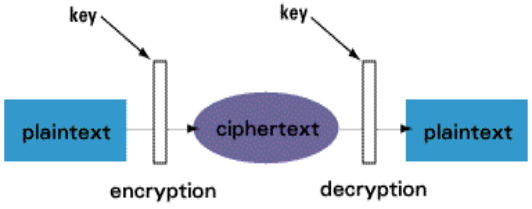

Fig 3: Symmetric Algorithm

Asymmetric algorithms [10] (Fig 4) use a different key for encryption and decryption, and the decryption key cannot

be derived from the encryption key. Asymmetric ciphers make a public key universally available, while only one individual possesses the private key. When data is encrypted with the public key, it can only be decrypted with the private key, and vice versa.

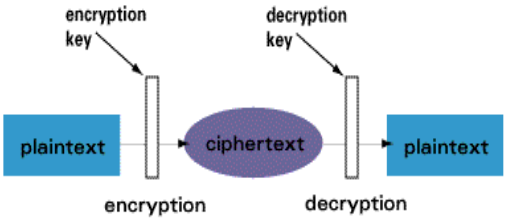

Fig 4: Asymmetric Algorithm

C. AES (Advanced Encryption Standard)

AES [7] algorithm is the advanced encryption standard form of algorithm which had been used as a symmetric form of encryption. This form of algorithm is being used by the government of United States in various applications. The AES algorithm has three different types of block ciphers, AES-128, AES-192 and AES-256. AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware. The design and strength of all key lengths of the AES algorithm (i.e., 128, 192 and 256) are sufficient to protect classified information up to the secret level. So, we are using AES to encrypt the data in HBH multicast routing protocol.

The overall structure of AES [7] encryption/decryption is shown in Fig 5.

- The number of rounds shown in Figure 2, 10, is for the case when the encryption key is 128 bit long. (If the number of rounds is 12 then the key is 192 bits , and when rounds are 14 the key is 256.)
- Before any round-based processing, encryption can begin, and the input state array is XORed with the first four words of the key schedule. The same thing happens during decryption, except that now we XOR the cipher text state array with the last four words of the key schedule.
- For encryption, each round consists of the following four steps:
  1) Substitute bytes
  2) Shift rows
  3) Mix columns and
  4) Add round key.

  The last step consists of XORing the output of the previous three steps with four words from the key schedule.
- For decryption, each round consists of the following four steps: 1) Inverse shift rows, 2) Inverse substitute bytes, 3) Add round key, and 4) Inverse mix columns. The third step consists of XORing the output of the previous two steps with four words from the key schedule.
- The last round for encryption does not involve the "Mix columns" step. The last round for decryption does not involve the "Inverse mix columns" step.

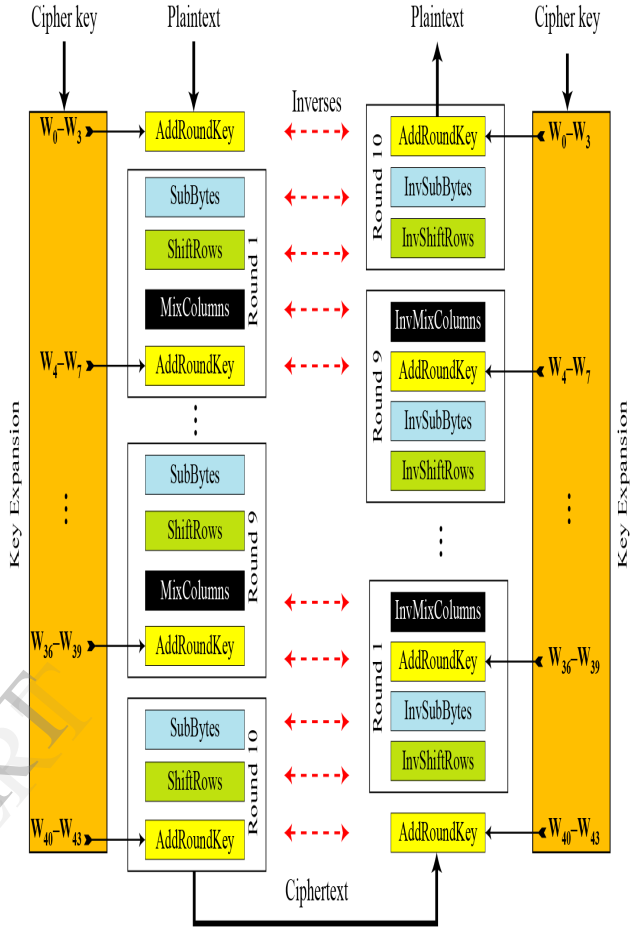

Fig 5: Overall structure of AES encryption/decryption

III. RUSH TECHNIQUE IN HBH

A. *HBH Multicast Protocol using RUSH*

The HBH multicast protocol has a tree construction algorithm that is able to better deal with the different cases occurred due to asymmetric unicast routes. HBH uses two tables, an MCT and an MFT, HBH stores the address of a next branchingnode instead of the address of a receiver, except the branching router nearest the receiver. By using RUSH technique the encrypted data received by a branching router H_B , has unicast destination address set to H_B .A multicast channel in HBH is identified by <S,G>, where S is the unicast address of the source and G is a class-D IP address allocated by the source. This definition solves the address allocation problem. The tree structure of HBH has the advantage of an enhanced tree stability of the table entries

B. *Tree Management in HBH*

HBH has three control messages: join, tree, and fusion. Join messages are periodically unicast by the receivers in the direction of the source and are used to refresh the forwarding state (MFT entry) at the router where the receiver was connected to the tree. A branching router itself "joins" the

multicast channel at the next upstream branching router. The join messages may be intercepted by the branching routers, which must sign themselves join messages, and filter the join messages received from downstream nodes. The source periodically multicasts a tree message that refreshes the tree structure. Fusion messages are sent by potential branching routers and construct the distribution tree together with the tree messages.

The basic idea of HBH is the first join issued by a receiver is never intercepted, reaching the source, and the tree messages are periodically multicasted by the source. These tree messages are combined with fusion messages, sent by potential branching nodes, to construct and refine the tree structure.

C. *Routing Tables*

Routers implementing HBH which are in the distribution tree of Shave<S,G> entries in their routing tables. Normally, non-branching routers have <S,G> entries in their MCT. Branching routers have <S,G> entries in their MFT. Nevertheless, a non-branching router may also have an MFT, in some configurations, to cope with asymmetric routes.

- The MCT<S,G> has a single entry, which can be fresh or stale.
- Two timers are associated to the MCT entry, t1and t2.
- At the expiration of t1 the MCT becomes stale and at the expiration of t2 the MCT is destroyed.
- MCT entries are not used to replicate data. Depending on its state, different actions are taken upon reception of control messages.
- The MFT entries are also soft-state.
- Two timers, t1 and t2, are associated with each entry in MFT<S, G>.
- When t1 times out, the MFT entry becomes stale and it is destroyed when t2 expires.
- MFT and MCT entries stored for channel <S,G>has three states:
  a) Default - The default state of an MFT entry is to be fresh and unmarked. In default state, the MFT entry is used for both data and control forwarding.
  b) Stale - If the MFT entry is stale, it is used to forward data but it is no longer used to forward control messages.
  c) Marked- The MFT entry may also be marked. As opposed to a stale entry, a marked entry is used to produce tree control messages but does not participate in data replication.
- A branching node is a router that has more than one outgoing multicast link, which is the common definition of a branching node. Therefore, an HBH branching router has MFTstate. Nevertheless, a non-branching router may also keep MFT state, in certain scenarios, with asymmetric routes.

D. *Data Forwarding In HBH with security*

HBH uses IP unicast destination addresses for data packets. Data packets are replicated at HBH branching routers, in such a way that all the receivers connected to the multicast channel receive the data.

Data forwarding in HBH works as follows. Suppose a source, S, and the source's first-hop routerH_1. The source produces IP packets which have the IP source address set toS's IP address and the IP destination address set toH_1's IP address. The router H_1 creates copies of the data packet, according to its MFT. The copies have the IP destination address set to the unicast address stored in the MFT (typically, that is the address of the next-hop HBH router). Packets are recursively replicated according to the distribution tree, until they get to the tree leaves. Fig 6 shows how the recursive unicast data distribution works for our Secure HBH protocol. In this, $H_j$is an HBH router. The source S sends encrypted data and cluster id to H_1. The router H_1 creates two packet copies and sends them to H_4 and H_5(the next downstream branching nodes). The router H_3 simply forwards the encrypted packets in unicast. H_5receives the data and sends the modified packet copy to cluster 2, $H_7$and r_8. Finally, H_7creates one packet copy to r_4, r_5 and r_6. The receiver receives the packets copy and decrypts the original data using AES algorithm.

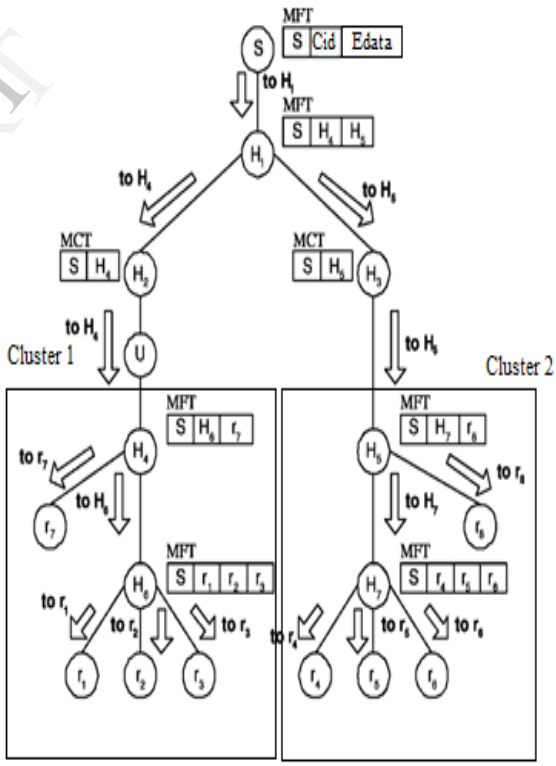

Fig 6: Data Forwarding in HBH using RUSH

IV. IMPLEMENTATION

A. *Avoiding Packet Duplication in HBH*

Asymmetric routing means that the unicast path from A to B may differ from the path from B to A. In HBH we can avoid packet duplication in asymmetric routing using fusion message. First join(S, r2)message will reach the source. After

receiving different tree messages for $r_1$and r_2, router H_1 sends a fusion(S, {r_1, r2}) to S. Subsequent join(S, r1) and join(S, r2) messages will be intercepted by H_1. At its turn, routerH_6 receives two different trees and sends a fusion(S, {r_1, r2}) upstream. In this case, however, it will neverreceive join messages issued by receiver'sr$_1$ and r_2. The consequence is that H_6's entry in H_1 will be kept stale and r_1 and r_2 entries will be fresh, but marked. Thus, data will be produced to H_6 as control will be addressed to r_1 and r_2. The design choice of HBH imposes some control overhead but minimizes data duplication. By this design we can avoid packet duplication in HBH to provide scalability service.

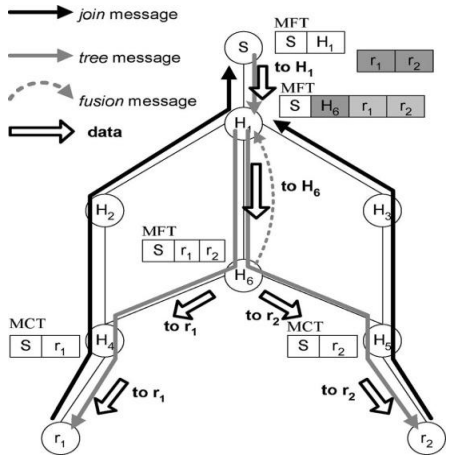

Fig 7: Avoiding Packet Duplication in HBH

RESULTS:

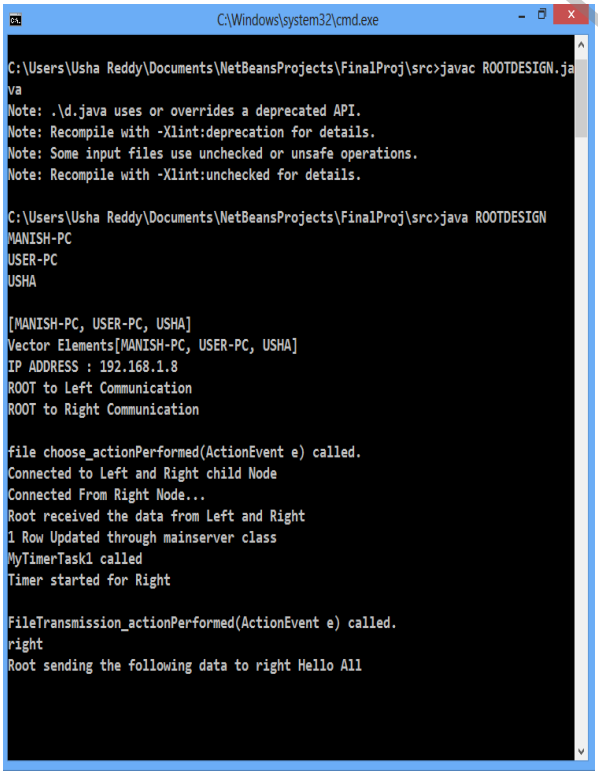

Fig 8: Data forwarding from the source

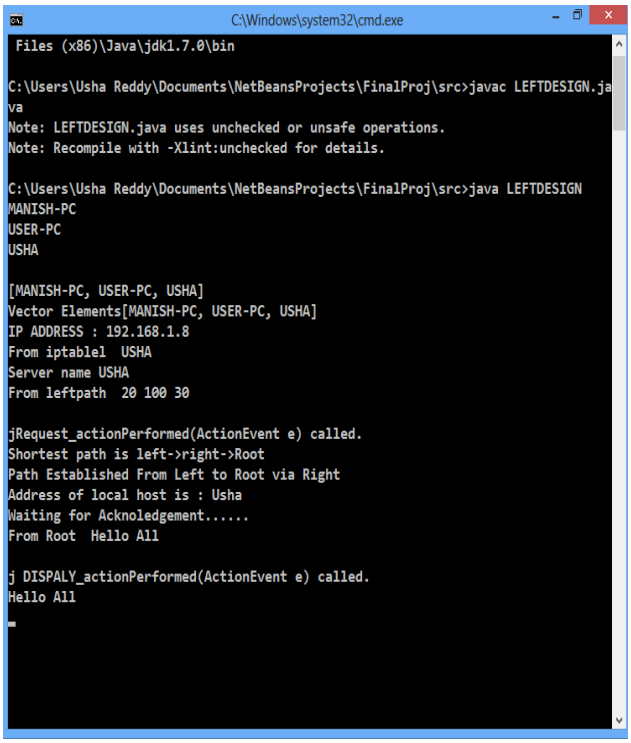

Fig 9: Data received from source to left router

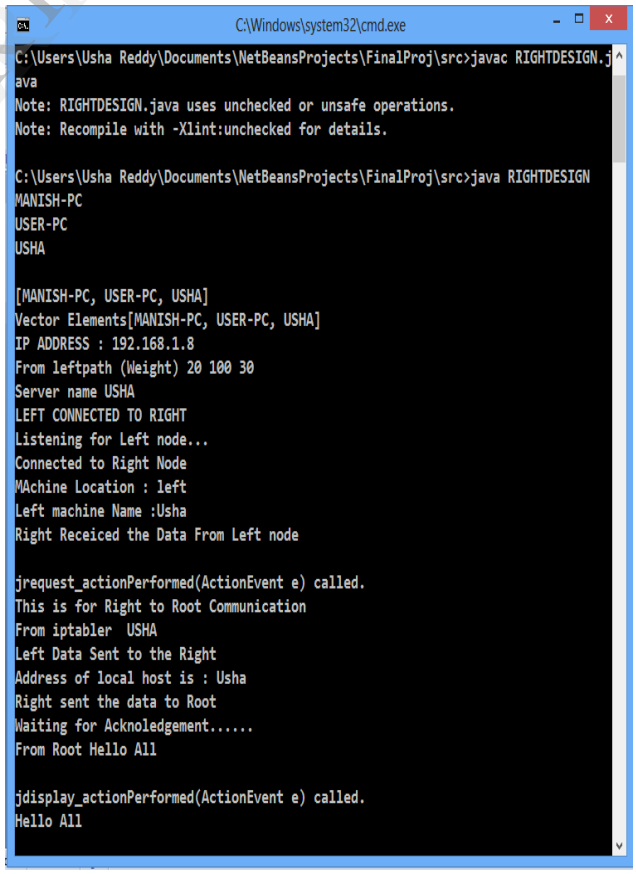

Fig 10: Data received from source to right router

B. *Avoiding Packet Duplication in HBH using RUSH:*

In RUSH Technique we can avoid packet duplication in asymmetric routing using fusion message. First *join(S, Cid, Edata) message* will reach the source. After receiving different *tree* messages for r_1 and r_2, router H_1 sends a *fusion(S,{r_1,r_2})* to *S*. Subsequent *join(S,r_1)* and *join(S,r_2)* messages will be intercepted by H_1. At its turn, routerH_6 receives two different *tree*s and sends a *fusion(S,{r_1,r_2})* upstream. In this case, however, it will neverreceive *join* messages issued by receiversr_1 and r_2. The consequence is that H_6's entry in H_1 will be kept stale and r_1 and r_2 entries will be fresh, but *marked*. Data will be produced to H_6 as control will be addressed to r_1 and r_2. Thus, Encrypted data is transferred to the respective receiver's in a secure format.

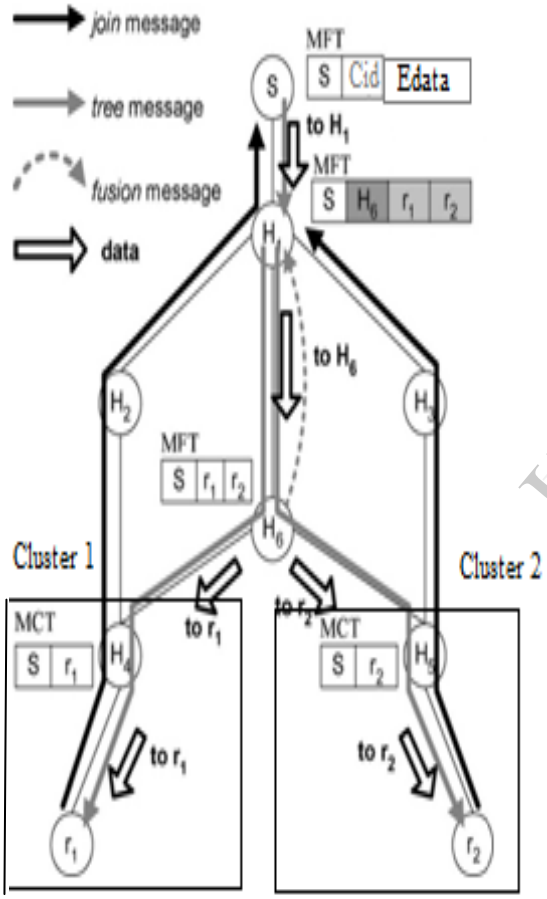

Fig 11: Avoiding packet duplication in HBH using RUSH

RESULTS:

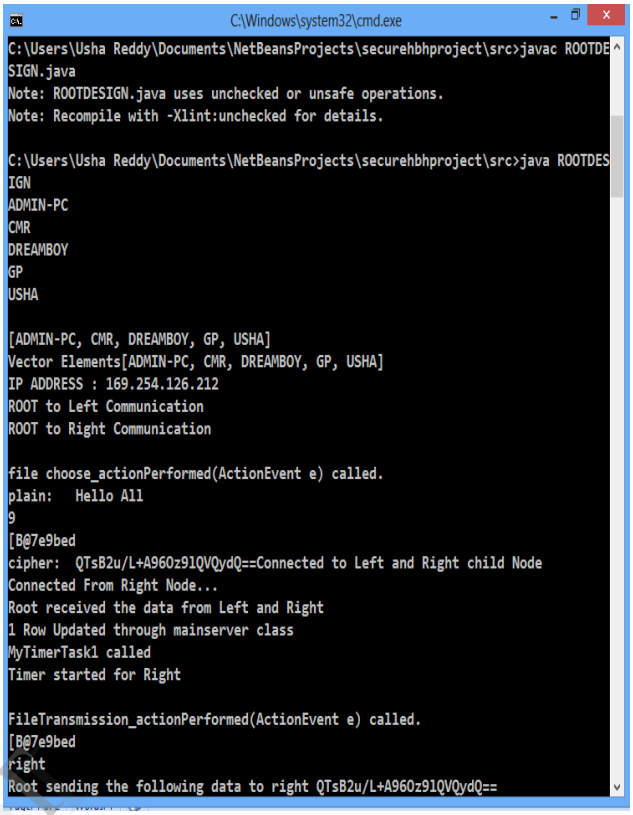

Fig 12: Encrypted data forwarded from source to branching routers

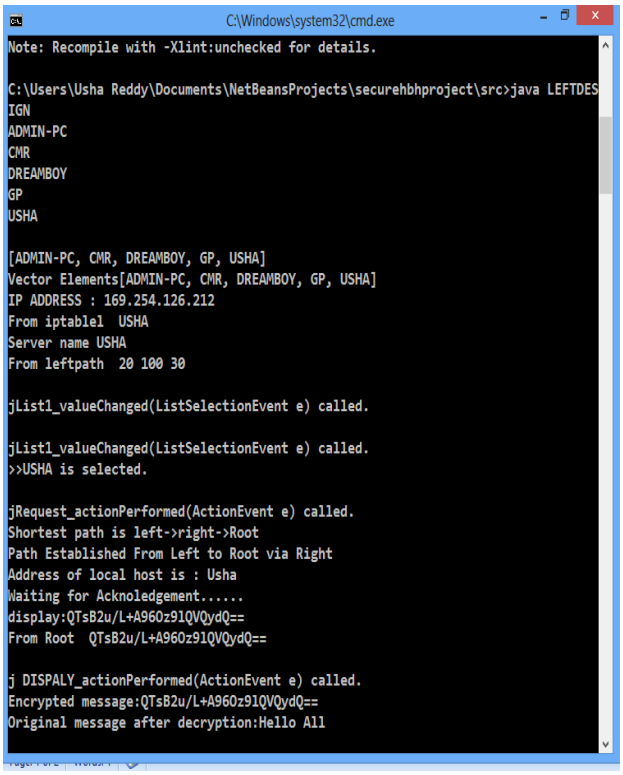

Fig 13: Decrypted data received at the left router

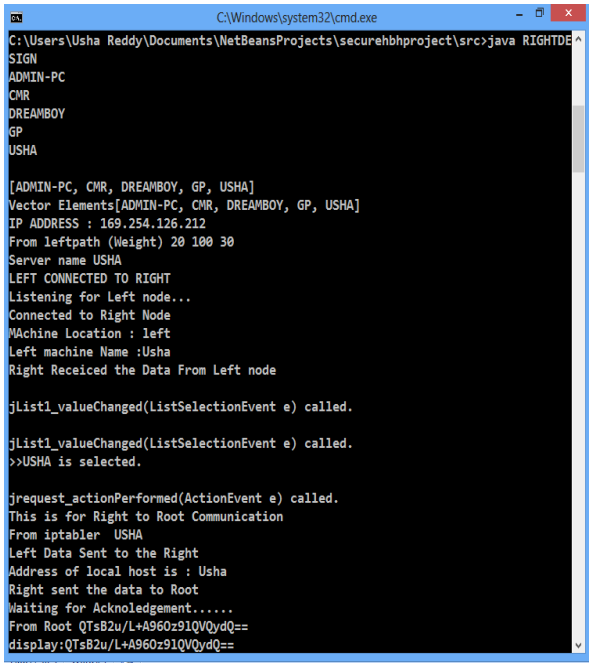

Fig 14: Decrypted data received at the right router

C. *Comparison of Protocols:*

TABLE 1. Comparison of Protocols

Protocols /Parameters	***REUNITE***	***HBH***	***REHASH***	***RUSH***
Supports Recursive unicast trees	*Yes*	*Yes*	*Yes*	*Yes*
Asymmetric Routing to determine shortest path	*No*	*Yes.*	*Yes*	*Yes*
Symmetric Routing	*Yes*	*Yes*	*Yes*	*Yes*
Packet Duplication	*Yes*	*No*	*No*	*No*
Clustering the receivers	*No*	*No*	*Yes*	*Yes*
delay experienced	*High*	*Low*	*Medium*	*Low*
Tree Cost Analysis	*High*	*Low*	-	*Low*
Message Cost Analysis	*More overhead*	*Less over head*	-	*Less over head*
Incremental Deployment	*Yes*	*Yes*	-	*Yes*
Scalability	*No*	*No*	*Yes*	*Yes*
Security	*No*	*No*	*No*	*Yes*

V. CONCLUSION

We have presented RUSH Technique in the existing HBH [1] Multicast Protocol that implements secure data distribution through recursive unicast hash tree[9]. The tree management algorithm of HBH [8] uses three control messages to construct an SPT. Join messages are periodically sent to the source by the receivers. The source periodically produces tree messages that are multicasted to the receivers. As the tree messages travels in the tree, the intermediate nodes may generate fusion messages that are responsible of refining the tree structure.

So by using these three control messages a tree structure is formed .After the formation of the tree structure clustering [6] is done based on the hash function.Then to multicast a packet, the root sends a copy of the packet to each hash address in its list, which leads to the related sub-trees. Similarly, when a branching node forwards such a packet, it sends a copy of the packet to each receiver in its own list. This procedure continues recursively until packets reach all leaf nodes of the tree, i.e., all the receivers. This enhances the stability in HBH [1] by using RUSH technique.

To provide data privacy, our goal is to prevent an attacker from gaining any information about sensitive data. By using hop-by-hop encryption between the sender and receiver, the technique can easily be implemented as all data between them will be encrypted. Only the sender and receiver need to share encryption keys. This approach also allows each router to inspect the contents of incoming and outgoing data to ensure local information security by passing through them. The data packet can be decrypted before being displayed by the receiver.

So, we can increase the key features like scalability and security in HBH multicast routing protocol [2] to make it more efficient to use.

REFERENCES

[1] "Incremental Service Deployment Using the "Hop-By-Hop Multicast Routing Protocol"Luis Henrique, M. K. Costa, *Member, IEEE*, Serge Fdida, *Senior Member, IEEE*, and Otto Carlos M. B. Duarte.IEEE/ACMTRANSACTIONS ON NETWORKING, VOL. 14, NO. 3, JUNE 2006.

[2] "An Approach to Improve the State Scalability of Source Specific Multicast,"S.A. Al-Talib, B.M. Ali and S. Khatun.

[3] I. Stoica, T. S. E. Ng, and H. Zhang, "REUNITE: A recursive unicast approach to multicast," in *Proc. IEEE INFOCOM*, Mar. 2000, vol. 1644–1653.

[4] C. Diot, B. N. Levine, B. Liles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture" *IEEENetwork*, vol. 14, no. 1, Jan 2000.

[5] KHIP | "A Scalable Protocol for Secure Multicast Routing", Clay Shields J.J. Garcia-Luna-Aceves.

[6] REHASH: Integrating Recursive Unicast with Hash Algorithm to provide Multicast Service to ReceiversB. M. Ali, S. A. Al-Talib, S. Khatun, S. Subramaniam.

[7] "Announcing the ADVANCED ENCRYPTION STANDARD (AES)".Federal Information Processing Standards Publication 197.United States National Institute of Standards and Technology (NIST).Nov-2001.

[8] L. H. M. K. Costa, S. Fdida, and O. C. M. B. Duarte, "Hop by hop multicast routing protocol," in *Proc. ACM SIGCOMM*, Aug. 2001.

[9] "Multicasting through Hop-by-Hop Routing ProtocolUsing Modified Recursive Unicast",U. Raghunath Reddy, K. Sekhar, P.Prabhavathi .

[10] "Analyzing the Effects of Asymmetric Unicast Routes on Multicast Routing Protocol", *Luis Henrique M. K. Costa, Serge Fdida, and Otto Carlos M. B. Duarte*.