Performance Analysis of Sponsor Selection Algorithms in Group Key Agreement

Shital Supase[1] and Rajesh Ingle[2]
[1]Assistant Professor, [2]Professor
Department of Comp. Engg
PICT, Pune – 43

ABSTRACT
Many of the emerging group oriented, collaborative and distributed application are based on the secure group communication. The various group key agreement protocols are proposed for the same. The performance of all these group key agreement protocols depends on their performance metrics like communication cost and computation cost. In this work we present the performance analysis of STR protocol based on two different sponsor selection algorithms. These two algorithms are implemented by extending the Adaptive Middleware APIs for text conferencing application. Here we present the comparative analysis communication cost and computation cost for both algorithms.

Keywords: Security, Secure group communication, Group key agreement, sponsor selection

1. INTRODUCTION
Group oriented applications like video conferencing, text conferencing, and white boards are increasing in number day by day. These applications are collaborative and distributed in nature. Most of the applications are peer to peer network based. In these applications the secure group communication is very important issue to be considered. The numbers of group key agreement protocols are used for secure group communication. STR (Skinny Tree) protocol [3] [4] is one of the group key agreement protocol. This protocol extends the 2-party Diffie-Hellman key exchange [9] and forms the secure group for communication [2] [3] [4] [5] [8].

The STR protocol is proposed by Steer et al. in 1989 and is originally aimed at teleconferencing application. This STR protocol is extended here for dynamic group in communication. The STR protocol is implemented by extending the Adaptive Middleware APIs [1] for text conferencing application in peer to peer network. Here two sponsor selection algorithms for the STR protocol are implemented and analyzed in Local Area Network.

Main contribution to this work:

- A peer to peer group key management framework (APIs) allowing selecting from two different sponsorship algorithms for STR protocol.
- A performance analysis of STR protocol for both sponsor selection algorithms in respect of communication costs (communication latency) and computation costs (computation latency)

This paper presents a comparative analysis [5] [8] of two different sponsor selection algorithms for STR protocol. The performance of both of these algorithms is analyzed by actually implementing and experimenting the protocol in Local Area Network with 4 to 16 systems.

Rest of the paper is organized as follows: section 2 describes the need for group key management and security requirement in group key management. STR group key agreement protocol is described in section 3. In section 4 the two different sponsor selection algorithms are described. Implementation details of the STRproject are given in section 5. Comparative analysis based on the results obtained is given in section 6 and conclusion and future work is described in section 7. All the references are listed in section 8.

2. GROUP KEY MANAGEMENT IN PEER GROUP
The group key agreement protocol implemented in this work is STR contributory protocol. The contributory group key agreement protocol requires every group member to contribute an equal share to the common group secret, computed as a function of all members' contributions. This protocol is appropriate for dynamic peer groups (DPG) [5]. This approach avoids the problems with the single point(s) of trust and failure. This protocol offers strong key management security properties such as key independence and perfect forward secrecy (PFS). The group key agreement protocol here implements the two membership events as follows:

Member Join*:* This is the membership event in which the new member joins the existing group. The new member sends join request to the existing group along with its blinded random. Upon validating this request everyone in group adds the new member to the group and sponsor of the group sends the groups blinded key to new member. Now everyone in the group computes the same group key.

Member Leave: The leave membership event occurs when any of the existing group members notifies that it is leaving the group. On receiving the leave message from existing group member, all remaining group members delete that member from the group and sponsor sends the new groups blinded key to all remaining members. Upon receiving this blinded key, all group members calculate the new group key.

2.1 Requirements of Group Key Management Protocols
The group key management in peer to peer network should satisfy the cryptographic properties. And the performance metrics like communication cost and computation cost are to be considered while choosing the group key agreement protocol. The protocol chosen should satisfy these properties for any of the group size from group with single member to group with n members.

2.1.1 Cryptographic Properties

Key management in secure group communication should satisfy certain properties [7] [5] like.

1. ***Forward Secrecy:*** It guarantees that, no any former group member should have access to current group communication.
2. ***Backward Secrecy:*** Any new member who joins the group should not have access to the previous group communication.
3. ***Group Key Secrecy:*** It ensures that for anyone who is not the member of existing group, it is computationally infeasible to calculate the group key.

2.1.2 Performance Metrics for Group Key Management Protocol

The group key agreement protocol performance depends upon the different performance metrics [3] [4] [5].

1. ***Communication Cost:*** This is number of messages transferred for any of the membership events like member join or member leave.
2. ***Computation Cost:*** This is number of fine grained integer operations (such as modular exponentiations) performed for group membership event of member join and member leave.

3. STR GROUP KEY AGREEMENT PROTOCOL

In peer to peer network the group key management protocols are used for secure group communication. STR (Skinny TRee) protocol is contributory group key management protocol [3] with an unbalanced tree structure whose height is always (n-1) where n is the group size. This protocol provides the following features:

- Each member contributes an equal share to group key computation, which is kept secret by every group member
- Group key is calculated as function of current group members share
- If any new member joins the group, its share is used by every existing member for key calculation
- If any member leaves the group the leaving members share is removed from the group and at least one of the group members (sponsor) refreshes its share

In Skinny Tree structure all of the leaf nodes are actual group members. And all intermediate nodes are dummy nodes. The root of the tree contains the group key always which is shared by all group members. Every member knows the complete structure of the tree and its position in the tree. Every member also knows its session random and the blinded session random of all other members. The protocol works on the fact that every member can compute the group key if it knows its own session random and all blinded keys in the tree structure.

3.1 Notations

The notations used to describe this STR tree are as follows:

p - Large prime number
g - Exponentiation base.
Mi - member of the group
INj- internal node at level j
ri - Mi's session random (that is secret key of node Mi)
bki - Mi's blinded key (that is g^{ri} mod p)
kj - secret key of internal node INj

Each peer in peer to peer network is represented as individual node of the tree. Every node of this tree has same structure. It has secret key and blinded key fields.
STR protocol always has a tree structure as shown below:

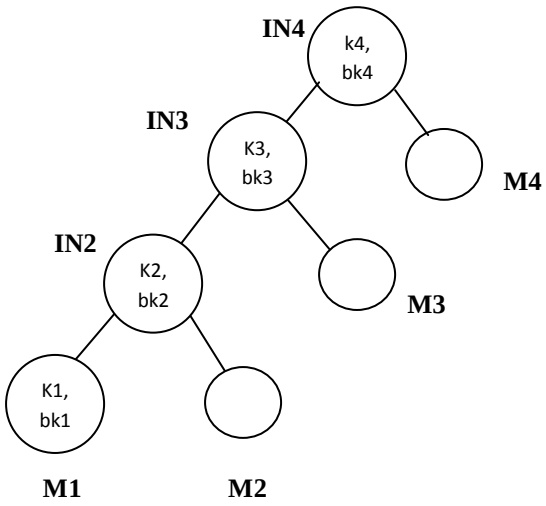


Figure 1: STR Tree Structure

Where, M1 to M4 are member nodes and IN1 to IN4 are internal nodes in tree structure. M1 is the initiator of the group which is responsible for creating a new group. M2 onwards are the members which have joined the group one after other. These individual member nodes represent the peer in peer to peer network. All internal nodes always have two children: one right leaf node and one left internal node.
Each leaf node chose same large prime number p and exponentiation base g. Each leaf node has a session random ri chosen and kept secret by Mi. The blinded version of this secret key is calculated as

$$bki = g^{ri} \bmod p$$

The group initiator or first member M1 chooses the random number r1 and calculates the blinded version of it bk1 as

$$bk1 = g^{r1} \bmod p$$

The session random of first member acts as its secret key. In single node tree structure this session random r1 acts as k1 that is group key of single node group. The basic key agreement protocol is as follows: Whenever M1 is the only member in the group it generates its own session random and calculates the blinded key. When new node M2 joins the group, both members M1 and M2 calculate the group key as

M1 calculates:k2 = (blinded key of M2) ^{r1} mod p
M2 calculates:k2 = (blinded key of M1) ^{r2} mod p

Both of these calculate the same group key. And set the k2 as their root secret key. This group key is used for the further group communication. Both members calculate the blinded group key and store in their root's blinded key field. In this tree structure any member can calculate the group key if it knows:

1) Its own session random
2) Blinded key of the sibling sub tree
3) Blinded session random of the member higher in the tree.

The group key can be calculated recursively as:

$$ki = (bki\text{-}1)^{ri} \bmod p$$

Where, bki-1 is the blinded group key. All blinded keys are assumed to be public.

3.2 Group Membership Events

The STR protocol has different membership events such as member join and member leave. In this group key agreement protocol implementation, the performance analysis of both of these events is performed.

3.2.1 Member Join Event

New member Mn+1 broadcast a join request that containing its own blinded session random [3] [5].

All existing group member creates new root node with two children: the root node of the prior tree on left and new leaf node on right. Sponsor of the prior group sends groups blinded key to new member.

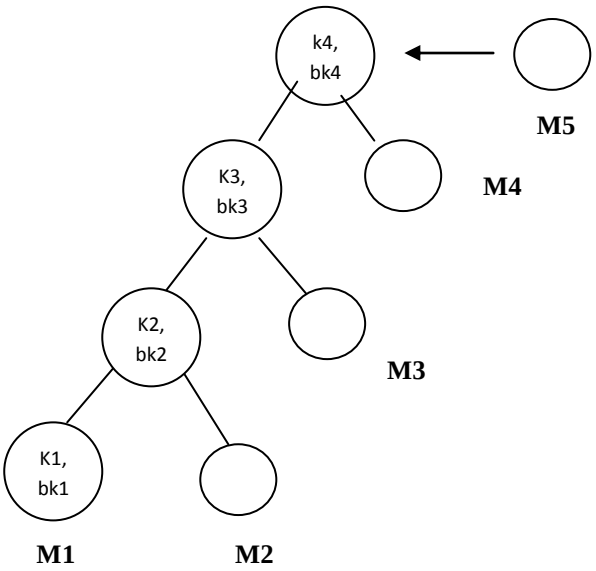


Figure 2: Before M5 join group

Now all group members can calculate the group key as:

- All existing members only need the new member's blinded key
- The new member needs the blinded group key of the prior group.

3.2.2 Member Leave Event

Upon hearing leave event each member updates its key tree by deleting that node and its parent node. New sponsor is chosen. Now sponsor node refreshes its session random and calculates the blinded key and sends this blinded key to all members of the group. This information allows all members to calculate the new group key.

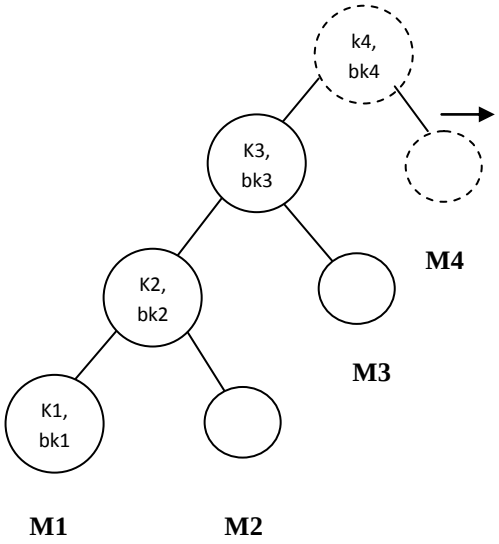


Figure 3: Before M4 leaves group

All member calculate the group key using own session random and new blinded key.

4. SPONSOR SELECTION ALGORITHMS

In this paper, performance analysis of the two sponsor selection algorithms for STR protocol is performed and results are given in section 5.

4.1 Leftmost Leaf Node Sponsor Selection Algorithm (STR-1)

In this algorithm, the leftmost leaf node member is always chosen as sponsor for all membership events such as member join and members leave. As the peer to peer network is dynamic by its nature, the initiator of the peer group can also leave the group at any time. In this case, tree is reconstructed and second member takes the place of initiator in the group. And new leftmost leaf node member after tree reconstruction is chosen as sponsor of the group.

This member remains the group sponsor for all events like new member join or any member leave event except itself.

4.2 Rightmost Leaf Node Sponsor Selection Algorithm (STR-2)

In this sponsor selection algorithm, the rightmost leaf node member is always chosen as sponsor for all membership events such as member join and members leave. Even rightmost leaf node member can leave the peer group. In this case, tree is reconstructed by deleting the leaving rightmost node and its intermediate node from the tree. And new rightmost leaf node member after tree reconstruction is chosen as sponsor of the group.If new member joins the group, then newly joined member is chosen as sponsor of the group because that becomes the rightmost node member now.

5. IMPLEMENTATION DETAILS

In this section, implementation details of this work are presented. First part presents the preliminary requirements and second part presents some of the system components which are part of implementation.

5.1 System Preliminaries

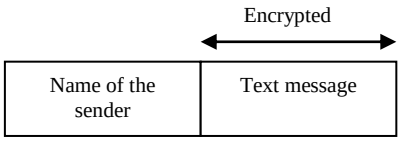
This project is implemented in JAVA under Windows XP Professional. Our implementation uses the Adaptive Middleware API functions. This implementation contains certain requirements. We require the complete list of IP addresses of the machines which may request to join the group. We require that each group member should first generate its own secret random (ri) number and blinded random (bri). The blinded random is generated by performing the exponential operation on random number (bri=g^ri mod p) [section 3.1]. This blinded random is sent to all members in group. Even if new member wants to join the existing group, it has to generate its own secret random and blinded random. And then it will send the join request to all existing group members along with its blinded random.

The sponsor member of the group which is chosen based on the algorithm used (STR-1 or STR-2) is always responsible for refreshing the group blinded key in membership event and sending the same to remaining member.

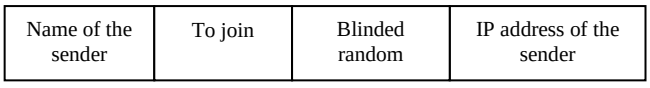
5.2 System Components

This project is composed of various components: (1) Classes, which holds variables and methods for various functionalities. (2) STR tree structure, which stores the various member nodes for a group. (3) Threads, the method which keeps running once started for receiving various messages. The messages received by these threads are join request message, leave message, rekey message, general text message, etc. (4) Messages, are the information packets which are exchanged among group members for various operations.

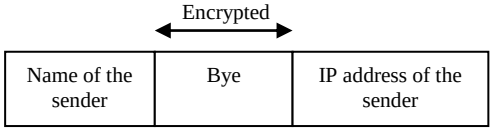
The messages in this work are classified into following categories: general message, membership event message and rekey message. General messages are text messages which are exchanged among group members for text conferencing application. The text in this type of messages is encrypted with the group key, which is decrypted on the other side with the same group key to retrieve the original message. The format of the general type of message is shown in figure 4.(a) below:



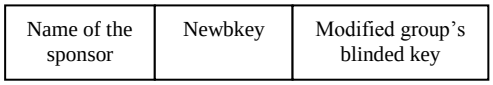
(a) General text message



(b) Join membership event message



(c) Leave membership event message



(d) Rekey message

Figure 4: General message format

Figure 4.(b) above shows the join request message. The leave message sent by the member leaving the group (as shown in figure 4.(c)) contains the encrypted message 'bye'. As shown in figure 4.(d) above, the rekey message is sent by the sponsor of the group and it contains the modified group's blinded key.

6. PERFORMANCE ANALYSIS OF THE PROTOCOL AND EXPERIMENTAL RESULTS

In this section first part illustrates the performance analysis of STR protocol algorithms in terms of different performance metrics and second part presents the experimental set up, experimental results and the different graphs which are plot as a result of the benchmarking performed in the implementation.

6.1 Complexity analysis

Table 1 below shows the complexity of protocol in membership event of join and leave.

6.1.1 Join: The join membership event requires 2 messages to be exchanged for joining the new member in group. It takes 3 exponentiation operations to be performed for computing new group key in join event.

Table1. Complexity analysis

Membership event	Communication cost (no. of messages)	Computation cost(no. of exponentiations)
Join	2	3
Leave	1	3n/2 +1

6.1.2 Leave: In leave membership event only one message is sent by the member who is leaving to the all group members. After this, no any message is to be sent back to the leaving member. The computation cost is always 3n/2 +1, where n is the group size.

6.2 Experimental Results

The actual performance of both the sponsor selection algorithms is compared by implementing the STR protocol. The benchmarking is done for measuring different performance metrics like computation delay (latency) right from the join of leave membership event happens to the time when group key agreement is done.

As all the members do not complete membership event and group key computation at the same time, the average latency is taken into consideration.

6.2.1 Test Environment

The systems with the configuration given here are used for getting the experimental results. The Processor is Intel (R) Core TM 2 Duo CPU with a speed of 2.93 GHz, having 4GB of RAM and Operating system: Microsoft Win XP Professional versions 2002, Service pack 3. Total 16 systems connected through a CISCO switch (CISCO 1900 series 24 port, 100.0 Mbps) are used for getting the benchmarking results.

The benchmarking results are taken for group size 4, 8, 12 and 16.

6.2.2 Experimental set up

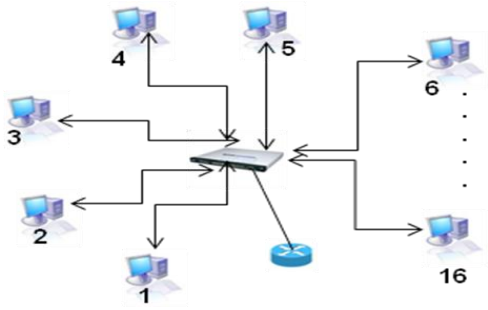


Figure 5: Experimental set up

Average Latency Results
The graph in Figure 6 below shows the average latency measurement and comparison for both the algorithms

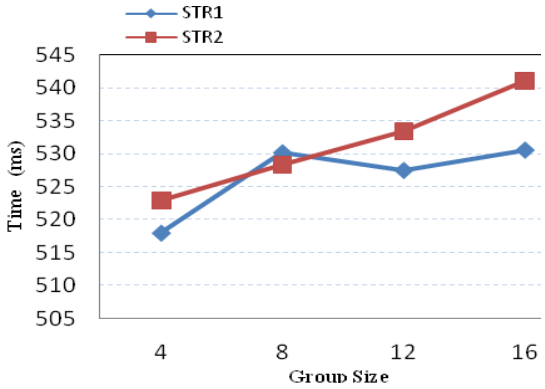


Figure 6: Average Latency Graph

It shows the average time in ms taken by systems to be reachable for further communication.

Average Join Latency Results
The graph in Figure 7 below shows the average latency for membership event join for both sponsor selection algorithms (STR-1 and STR-2) of STR protocol.

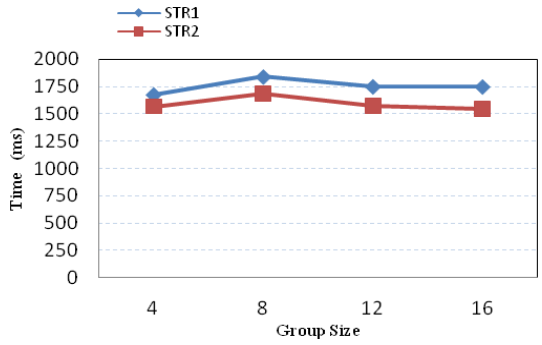


Figure: 7 Average Join Latency Graph

Here the latency time (in ms) for join event is measured as time from the new member sends the join request, it's validated by the existing group members, creating new STR protocol tree and calculating the new group key for new group.

Here it is observed that second sponsor selection algorithm which is depicted by STR-2 in graph takes less average time for join event.

Average Leave Latency Results
The graph shown in Figure 8 below shows the average latency for membership event leave for both the sponsor selection algorithms of (STR-1 and STR-2) of STR protocol.

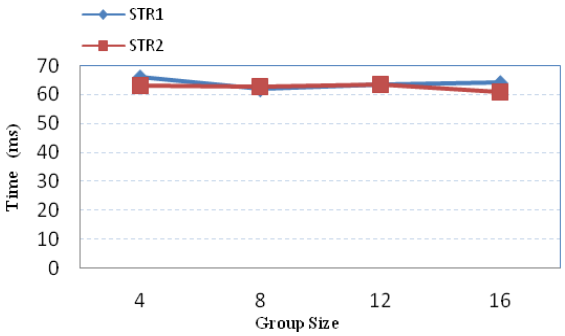


Figure 8: Average Leave Latency

The leave latency time (in ms) is calculated as from the time existing member sends leave message ('bye'), all exiting members deletes the leaving member from the tree, update tree, decide new sponsor based on the algorithm chosen (STR-1 or STR-2) and calculating new group key using new blinded group key sent by the new group's sponsor.

Again here it is observed that the second sponsor selection algorithm i.e. STR-2 takes less average time for leave event in group.

7. CONCLUSION AND FUTURE WORK

In this work, the STR group key agreement protocol is implemented successfully by extending adaptive middleware APIs. The performance of STR protocol for two different sponsor selection algorithms is analyzed.

Here the conclusion is drawn that the second sponsor selection algorithm that is STR-2 has always less communication and computation latency in both the membership events (join and leave).

This work can be extended further for analyzing the performance of other group key agreement algorithms. Other algorithm which can be implemented with this project is Tree based group key agreement protocol (TGDH).

8. REFERENCES

[1] Rajesh Ingle, G. Sivakumar, "TGKAM: Adaptive Middleware Architecture for Secure Group Communication", 2009 Sixth International Conference on Information Technology: New Generations

[2] Yongdae Kim, Adrian Perrig, and Gene Tsudik, "Tree-based Group Key Agreement".

[3] Yacine Challal, Hamida Seba, "Group Key Management Protocols: A Novel Taxonomy", International Journal Of Information Technology Volume 2 Number 1 2005 ISSN: 1305-2403

[4] Patrick P. C. Lee, John C. S. Lui, David K. Y. Yau , "SEAL:ASecure Communication Library for Building Dynamic Group Key Agreement Applications"

[5] Yongdae Kim, Adrian Perrig, Gene Tsudik," Group Key Agreement Efficient in Communication", in IEEE Computer Society 2004.

[6] Yongdae Kim, Adrian Perrig, and Gene Tsudik, "Communication-Efficient Group Key Agreement"

[7] Yair Amir, Y. Kim, Cristina Nita-Rotaru, Gene Tsudik, "On the Performance of Group Key Agreement Protocols"

[8] Hemlata Narvekar, Madhumita Chatterjee, "Key Management Protocols for Wired and Wireless Ad-hoc Networks: A Comparative Analysis", in ICSCI 2007, Hyderabad, India.

[9] William Stallings, "Cryptography and Network Security Principles and Practices", Fourth Edition.