Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

*IJCSMC, Vol. 2, Issue. 3, March 2013, pg.120 – 126*

RESEARCH ARTICLE

# Efficient Key Generation for Multimedia and Web Applications

**S.Swamy Reddy[1], V.Ramana reddy[2], K.Praveen kumar[3]**

[1]Computer Science and Engineering & Narsimha Reddy Engineering College, India

[2]Computer Science and Engineering & Narsimha Reddy Engineering College, India

[3]Computer Science and Engineering & Narsimha Reddy Engineering College, India

*[1]swamyreddysingi@gmail.com; [2]ramanavundla@gmail.com; [3]kpraveen0123@gmail.com*

***Abstract— One of the main challenges for a secure multicast is access control, for making sure those only legitimate members of multicast group have access to the group communication. In the past two or three decades, cryptography has become the well-established means to solve the security problems in networking. However, there are still a lot of difficulties for directly deploying cryptography algorithms into multicasting environment as what has been done for unicasting environment. The commonly used technique to secure multicast communication is to maintain a group key that is known to all users in the multicast group, but should remain unknown to any person outside the group. This Paper discusses an enhanced technique which can be used for an efficient management of the group key.***

***Key Terms: - Multicast, Key Generation Centers, Group Controller.***

## I. INTRODUCTION

The Existing system have the drawbacks such as the Group Controller takes all responsibilities of key generation, re keys generation, message transmission to its sub group members and also to any other group controllers. So it may cause a lot of bottlenecks to the group controller in the sub group.

The sub group's members are not able to send information to any other subgroup at the time of re-keying process. So performance of the sub group degrades at that time. The re-keying process is done every time once a communication is completed between the users in the same group or to any other group members. The Efficient Key agreement for a Large and Dynamic Multicast Groups provides an efficient way of Group key Agreement in terms of Scalability and Authenticity between the Sub group members and to other group members in the network.

## II. SYSTEM ANALYSIS

### *A. Existing System*

In the existing system we use Lotus approach which proposes the notion of hierarchy subgroup for scalable and secure multicast. In this method, a large communication group is divided into smaller subgroups. Each subgroup is treated almost like a separate multicast group and is managed by a trusted group security intermediary (GSI).GSI connect between the subgroups and share the subgroup key with each of their subgroup members. GSIs act as message relays and key translators between the subgroups by receiving the multicast messages from one subgroup, decrypting them and then re multicasting to the next subgroup after encrypting

them by the subgroup key of the next subgroup. The GSIs are also grouped in a top-level group that is managed by a group security controller (GSC).

When a group member joins or leaves only the corresponding subgroup gets affected while the other subgroup does not get affected. It has the drawback of affecting data path. This occurs in the sense that there is a need for translating the data that goes from one subgroup and thereby one key to another. This becomes even more problematic when it takes into account that the GSI has to manage the subgroup and perform the translation needed. The GSI may thus become the bottleneck.

### *B. Limitations of Existing System*

- The Group controller takes all responsibilities for the group such as key generation, re-keying process and message transfer to any other groups
- The group members are not able to communicate with any other groups during the re-keying process.
- The Group controller maintains logical key tree where each nodes represents a key encryption key.

## III. PROPOSED SYSTEM

In the proposed system we use an identity tree instead of key tree in our scheme. Each node in the identity tree is associated with an identity. The leaf node's identity is corresponding to the user's identity and the intermediate node's identity is generated by its children's identity. Hence, in an identity tree, an intermediate node represents set users in the sub tree rooted at this node.

The keys used in each subgroup can be generated by a group of key generation centers (KGCs) in parallel. All the members in the same subgroup can compute the same subgroup key though the keys for them are generated by different KGCs. This is a desirable feature especially for the large-scale network systems, because it minimizes the problem of concentrating the workload on a single entity.

**Advantages of Proposed System:**

- The Group controller responsibilities are shared by the Group control intermediate such as Re keying process and scalability of the group process
- The group members are not affected by the key generation process when they are willing to communicate with any other group members.
- The Centralized key server used for key generation process and the KGC is also act as a Router for group to group communication.
- The Re-keying process is done only to the particular group members not to the entire group members**.**

### *A. System Design*

System design is the first design stage in which the basic approach to solving the problem is selected. During system design, the overall structure and style are decided. The system architecture is the overall organization of the system into components called subsystems. The architecture provides the context in which more detailed decisions are made in later design stages. By making high level decisions that apply to the entire system, the system designer partitions the problem into subsystems so that further work can be done by several designers working independently on different subsystems.

The system designer must make the following decisions

- Organize the system into subsystems.
- Identify the concurrency inherent in the problem.
- Allocate subsystems to processors and tasks.
- Choose an approach for management of data stores.
- Handle access to global resources.
- Choose the implementation of control in software.
- Handle boundary conditions.
- Set trade-off priorities.

*B. SEQUENCE DIAGRAM*

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Chart. Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams.

Sequence Diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

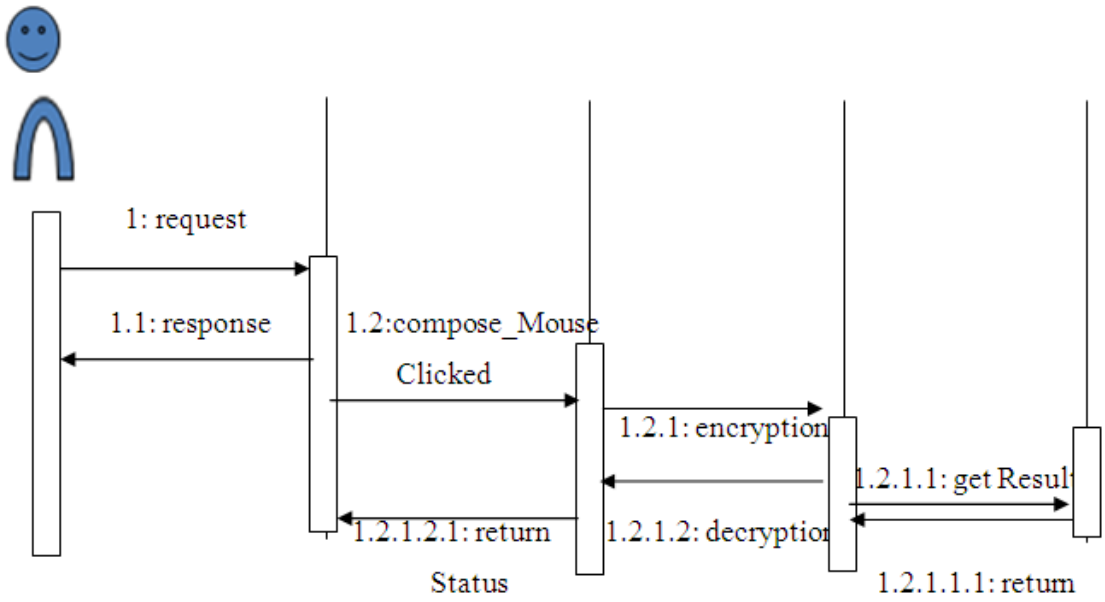



Fig.1 Sequence Diagram

*C. IMPLEMENTATION*

Implementation is the stage in the project where the theoretical design is turned into a working system. It involves careful planning, investigation of the current System and its constraints on implementation, design of methods to achieve the changeover, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation.

Implementation is the final and important phase, the most critical stage in achieving a successful new system and in giving the users confidence that the new system will work be effective .The system can be implemented only after through testing is done and if it found to working according to the specification.

*D. TESTING*

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

1) UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

2) FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:
Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

3) SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

4) PERFORMANCE TEST

The Performance test ensures that the output is produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

5) INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

6) ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

*E. Test Cases*

**TEST CASE 1**

**Test Precondition :** select the Batch files
**Type of Testing :** Valid function
**Objectives :** Running of server
**Valid Condition :** True
**Expected Result :** check the proxies that should be processed

**TEST CASE 2**
**Test Precondition :** valid login
**Type of Testing :** valid login
**Objectives :** Opening of login page
**Valid Condition :** True
**Expected Result :** check login status.

**TEST CASE 3**
**Test Precondition :** Mail must be sent
**Type of Testing :** Login information testing
**Objectives :** Check the log file for mails.
**Valid Condition :** True
**Expected Result :** It should contain information about the size of mail and what type of mail is sent.

**TEST CASE 4**
**Test Precondition :** Mail must be sent to group members
**Type of Testing :** group controller testing
**Objectives :** Add one or more group members and update keys to group members.
**Valid Condition :** True
**Expected Result :** Application should interact with the specified group members

**TEST CASE 5**
**Test Precondition :** Key generation
**Type of Testing :** Update keys for group members testing.

**Objectives** : Update keys for join members and release keys for leave members.
**Valid Condition** : True
**Expected Result** : Application should generate keys for group members.

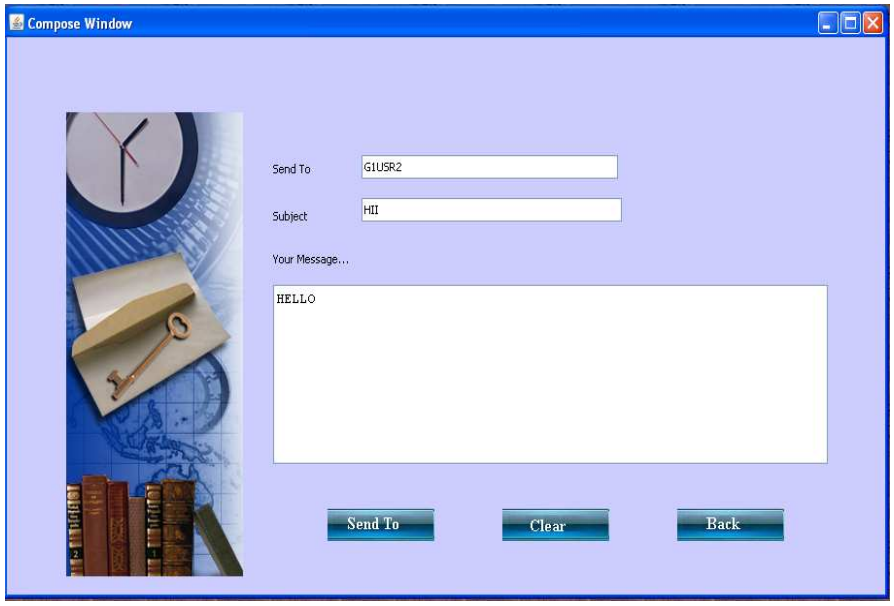
**TEST CASE 6**
**Test Precondition** : Deleting mails of group members.
**Type of Testing** : Deleting of mails testing.
**Objectives** : Deleting of mails by group members is done.
**Valid Condition** : True
**Expected Result** : Application should delete mails by group members.

**SCREEN SHOTS**

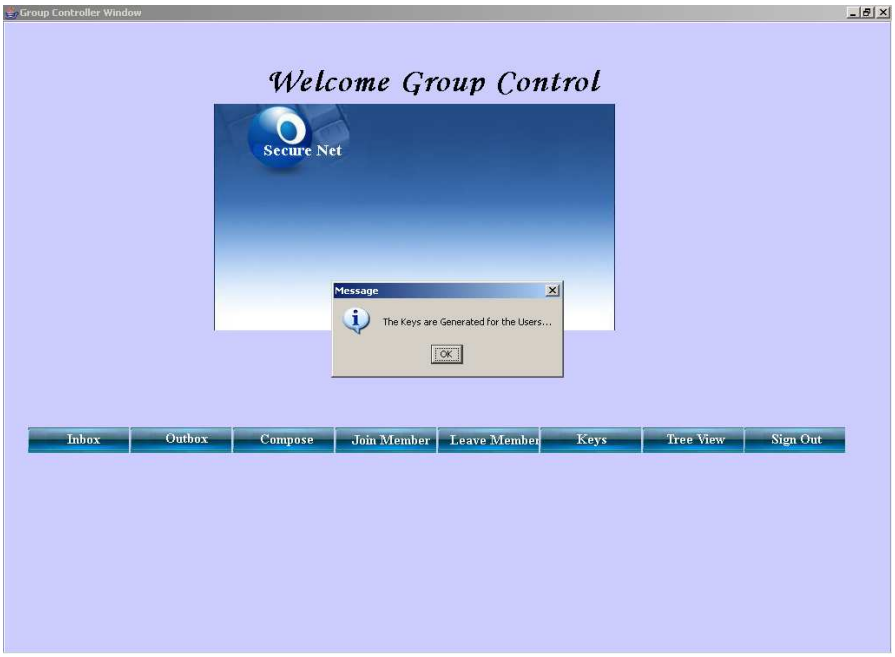

**User Login Page**


**User Logins Welcome Page**




**Compose Window**




**Key Generation Page**

## IV. CONCLUSION

The Proposed system is an efficient, authenticated, scalable key agreement for large and dynamic multicast systems, which is based on the bilinear map. New modules are in pipeline for to increase the compatibility of the project. Once these improvements have been done, the majority of the features that make an application an excellent one would be there and the usage would become wider.

## REFERENCES

[1] Y. Amir, Yakima, C. Nita-Rotary, J. L. Schultz, J. Stanton, and G.Tsudik, "Secure group communication using robust contributory key agreement," IEEE Trans, Parallel Distrib. Syst., vol: 15, no.5, pp, 468-480, May 2004.
[2] G. Ateniese, M. Steiner, and G. Tsudik, "Authenticated group key agreement protocols,"in Proc.5th Annu. Workshop on selected Areas in Cryptography Security (SAC'98), 1998, pp. 17-26.
[3] S.Blake-Wilson and A.Menezes, "Authenticated Diffie-Hellman Key agreement protocols," in Proc. 5th Annu. Workshop on selected Areas in Cyrptography (SAC'98), 1998, vol. LNCS 950, pp. 275-286.
[4] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Trans. Inf. Theory, vol. 22, no. 6, pp. 644–654, 1976.
[5] http://www java.sun.com
[6] http://www.java2s.com
[7] http://www.w3schools.com