INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

Secured Path For Message Transmission In Group Network Using Elliptic Curve Cryptography

Lokesh A
Asst.Professor, M S Engineering College, Bengaluru,Karnataka, India
Prapulla C
M.Tech II SEM CS&E, M S Engineering College, Bengaluru, India
Srinivas Goud
Asst.Prof, Dept. of CS&E Vignana Bharathi Institute of Technology,Aushapur,Hyderabad, India
Nivedita G Y
M.Tech Dept. of CS&E, M S Engineering College, Bengaluru,Karnataka, India

Abstract:

The main constraint in message transmission is security. The key must be shared among users so that we have a secure transmission in group based message transmission. In this paper we discuss an important security problem that happens in mobile ad hoc network for key agreement in dynamic group. For a communication to be secure, a group key must be shared by all the members of the group. The key must be updated when any member of the group leaves existing group or become a new member of the existing group. In this paper, we establish a secured transmission. The main idea here is to spilt the existing network into smaller sub-network and each sub-network maintains its own keys. These keys should be updated when a existing node leave the network or when an new node joins the existing system. The nodes that have the high stability in the sub-network will be made as the Inner gateway member and among Inner gateway member the one that has high stability will be made as Outer gateway member.

Keywords*: adhoc network, secured group transmission, elliptic curve cryptography, KASP.*

1.Introduction

This paper addresses an interesting security problem in mobile ad hoc networks that is dynamic group key agreement for key establishment. For secure communication, a group key must be shared by all group members. This group key should be updated when the existing group members are leaving the network or new members are entering into the existing network. In this paper, we propose an efficient group key agreement protocol called Key Agreement protocol based on Stability and Power (KASP). Here the idea is to split a large group into several subgroups, each maintaining its subgroup keys to manage the subgroup and managing many subgroups using Elliptic Curve Diffie-Hellman (ECDH) key agreement algorithm. In KASP, we develop two protocols namely, Subgroup Key Generation(SKG) and Group KeyGeneration(GKG) based on ECDH for subgroups and outer groups respectively. These subgroup keys and group keys should be changed when there are membership changes (such as when the current member leaves or the new member joins). By introducing group-based approach, messages and key updates will be limited within subgroup and outer group. Thus computation load is distributed to many mobile ad hoc nodes. Both theoretical and practical results show that this KASP, a new efficient group key agreement protocol performs well for the key establishment problem in ad hoc network in terms of efficiency and security

This paper addresses an interesting security problem in mobile ad hoc networks that is dynamic group key agreement for key establishment. For secure communication, a group key must be shared by all group members. This group key should be updated when the existing group members are leaving the network or new members are entering into the existing network.

In this paper, we propose an efficient group key agreement protocol called Key Agreement protocol based on Stability and Power (KASP). Here the idea is to split a large group into several subgroups, each maintaining its subgroup keys to manage the subgroup and managing many subgroups using Elliptic Curve Diffie-Hellman (ECDH) key agreement algorithm.

In KASP, we develop two protocols namely, Subgroup Key Generation (SKG) and Group Key Generation (GKG) based on ECDH for subgroups and outer groups respectively. These subgroup keys and group keys should be changed when there are membership changes (such as when the current member leaves or the new member joins). By introducing group-based approach, messages and key updates will be limited within subgroup and outer group. Thus computation load is distributed to many mobile

ad hoc nodes. Both theoretical and practical results show that this KASP, a new efficient group key agreement protocol performs well for the key establishment problem in ad hoc network in terms of efficiency and security.

2.Existing System

In mobile ad hoc networks, the security is the main constraint in message transmission.
Limitations of existing system:

- Ad-hoc networks are not generally having a trusted third party.
- Keys are generated for singal user which is not secured in case of group communication.

3.Proposed System

This paper addresses an interesting security problem in mobile ad hoc networks that is dynamic group key agreement for key establishment. For secure communication, a group key must be shared by all group members.
Advantages of proposed system:

- Keys are generated for the group user.
- Ad-hoc networks are having a trusted third party called gateway member. Through which communication of the message takes place.

4.system design

4.1.Design Constraints

The project aims to develop a system which allowsgroup communication in network, by splitting the group into subgroups and generating keys for group and subgroup using ECDH. These keys will be applied for the message which needs to be transmitted over the network. The algorithms being implemented have a range of efficiency and complexity. The code is intended to be written in Java language, supporting JCreator and JDK environment.

4.2.Interfaces

Interface refers to a point of interaction between components and is applicable at the level of hardware and software.

4.2.1.User Interface

A user interface is the system by which usersinteract with a machine. The user interface includes hardware (physical) and software (logical) components. User interfaces exist for various systems, and provide a means of input that allow the users to manipulate a system, and output that allow the system to indicate the effects of the users manipulation.

The two proposed algorithms i.e. registration protocol and signature generation protocol to achieve high performance and scalability with respect to the security being provided. Application can be accessed from the cloud storage service over all nodes, and then exchanges the data via a network.

4.3.High Level Design

Software Development is generally a stepwise process. Before the process of implementing the software at hand it involves the process of software design. A software design is a description of the structure of the software to be implemented, the data which is part of the system, the interfaces between the components, sometimes, the algorithms used. Designers do not arrive at a finished design immediately but develop the design iteratively through a number of different versions. The design process involves adding formality and detail as the design is developed, with constant backtracking to correct earlier designs.

In many software development projects, software design is an ad hoc process. Starting from the set of requirements, usually in natural language, an informal design is prepared. Coding commences and the design stage is modified as the system is implemented. When the implementation stage is complete, the design has usually changed so much from the initial specification that the original design document becomes an incorrect and incomplete description of the system. There are several advantages of the design phase. Some of them are listed below:

- The design phase helps to understand the user requirements and helps to map the user requirements into implementation phase.

- The iterations in the design phase helps in incorporating as many user requirements as possible in the final software being developed.
- The design phase reduces the cost involved in the development of the software as many changes would be made to the software in the implementation if the design is not clear.

The design process is iterative and requires consideration of various design alternatives at every stage. The objective of the design stage is to produce the overall design of the software.

The design stage involves two sub-stages namely:

- High-Level Design
- Detailed-Level Design

In the High-Level Design, the Technical Architect of the project will study the proposed applications functional and non-functional (qualitative) requirements and design overall solution architecture of the application, which can handle those needs. High Level Design means precisely that. A high level design discusses an overall view of how something should work and the top-level components that will comprise the proposed solution.

It should have very little detail on implementation, i.e. no explicit class definitions, and in some cases not even details such as database type (relational or object) and programming language and platform. In this chapter we give an overview of the design of the system and how it is organized and the flow of data through the system. By reading this document the user should have an overall understanding of the problem and its solution. We have also discussed about the problems encountered during the design of the system and justified the use of the design. The Data Flow Diagrams (DFD), given in the later sections of the chapter, shows the flow of data through the system.

4.4.Design Considerations

The design process is iterative and requires consideration of various design alternatives at every stage. The design process is constrained by the assumptions made prior to the development of the system. It involves deciding on the type of approach used for the development of each portion of the system with the rationale for the selection of the same. Thus this section describes many of the issues which need to be addressed or resolved before attempting to devise a complete design solution.

4.4.1.Assumptions And Dependencies

Several assumptions regarding the hardware required and the working environment of the system influence design decisions. The assumptions have been made after considerable consultation with the end user and are more or less reasonable.

- The system will be implemented on the Windows operating system, using Java. The system should be modified so as to enable its use on the Linux operating system as well.
- The system runs on the Windows Operating System which is Windows XP or above.

4.4.2.General Constraints

There might be some global limitations or constraints that have a significant impact on the design of the system's software or an associated impact. Such constraints may be imposed on the following issues related to our project which are as follows:

- There exists a maximum limit on the number of nodes that can be deployed in the network so as to prevent degradation in performance of the proposed algorithm while it is being executed on the system.
- As the number of nodes increases, the complexity of showing different cases for the algorithm becomes difficult. Therefore, we limit ourselves to a fewer number of nodes for demonstration purpose.

5.System Architecture

The system is developed using the Top-Down approach. In this method, the system divides the files based on the number of nodes. Each node downloads the files and then requests the missing files from the neighboring nodes.

- The 'JAVA' programming language has been used for development of the application.
- The WINDOWS XP or later versions of windows XP operating system has been used as the platform for development.
- The processes communicate through Sockets.

5.1.System Architecture

This section provides a high-level overview of how the functionality and the responsibilities of the system were partitioned and then assigned to subsystems or the components or the modules appropriately. The main purpose here is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together to provide the desired functionality. The system architecture is as show in figure 1.

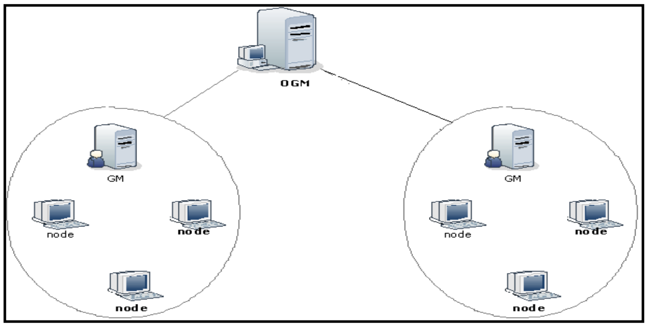


Figure 1: System Architecture

5.2.Diffie Hellman Algorithm

Diffie–Hellman key exchange (D–H)[nb 1] is a specific method of exchanging cryptographic keys. It is one of the earliest practical examples of key exchange implemented within the field of cryptography. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

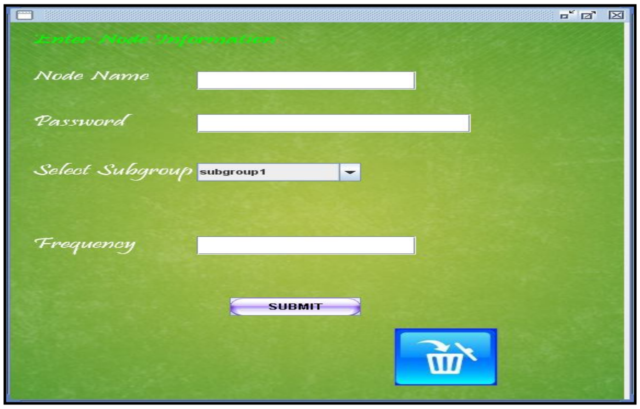
5.3.Elliptic Curve Cryptography using Diffie Hellman Algorithm

Elliptic curve Diffie–Hellman (ECDH) is an anonymous key agreement protocol that allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel.[1][2][3] This shared secret may be directly used as a key, or better yet, to derive another key which can then be used to encrypt subsequent communications using a symmetric key cipher. It is a variant of the Diffie–Hellman protocol using elliptic curve cryptography.

6.Result

6.1.Setup Mechanism

Figure 2: Open Jcreator and run java programs


Enter Node Information

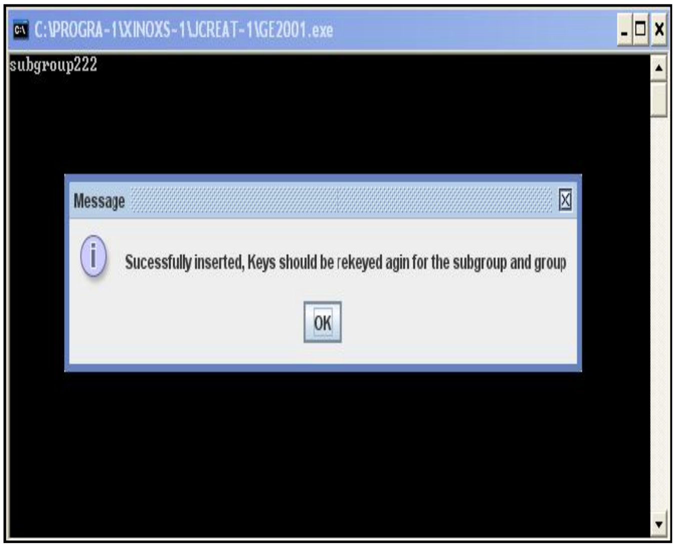
Node Name

Password

Select Subgroup subgroup1

Frequency

Figure 3: Enter the node details to add or delete a node



```
subgroup222
```

Message

Sucessfully inserted, Keys should be rekeyed agin for the subgroup and group

Figure 4: After successfully inserting the node a message appears

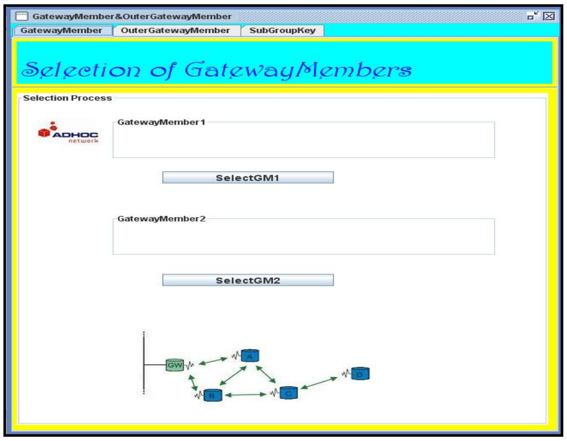
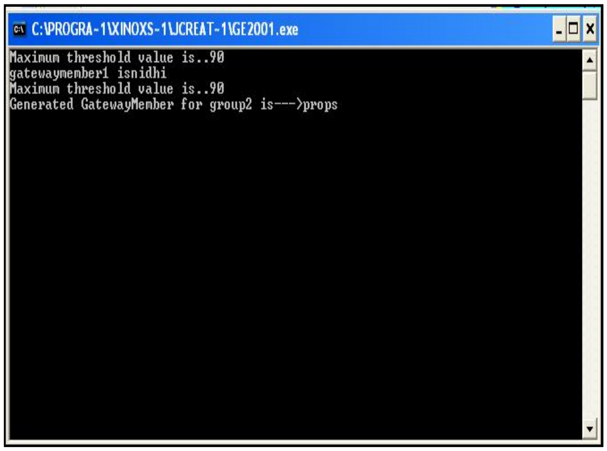
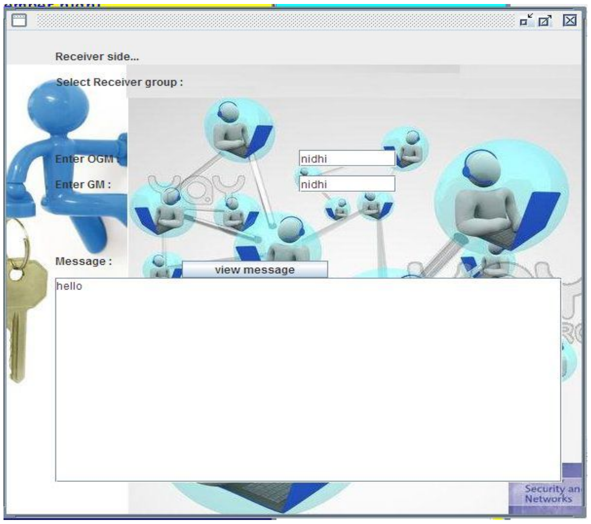


Figure 4: Select the GatewayMembers



```
Maximum threshold value is..90
gatewaymember1 isnidhi
Maximum threshold value is..90
Generated GatewayMember for group2 is--->props
```

Figure 5: Gateway Members for two groups are selected


Receiver side...

Select Receiver group :

Enter OGM : nidhi

Enter GM : nidhi

Message :

hello

Figure 6: Decrypted message is received by the reciever

7.Conclusion

Mechanisms that enable individual peers of unstructured P2P content sharing networks to register longstanding queries and receive notification when new matching items appear can significantly improve their utility and effectiveness. While the pub-sub paradigm can provide this capability, implementing pub-sub systems on unstructured overlays is often a very complex endeavor. The continuous query paradigm studied in this paper is similar to pub-sub, but it provides best effort notification service. We presented the design and evaluation of a lightweight system, called CoQUOS, which supports continuous queries in unstructured P2P networks.

The CoQUOS system incorporates severalnovel features such as cluster resilient random walk for query propagation and dynamic probability scheme for query registration, and a lazy replication technique for countering network churn.

8.Future Enhancement

8.1.*3GPP*

- The 3rd Generation Partnership Project(3GPP) standard is developing System Architecture Evolution(SAE)/Long Term Evolution(LTE) architecture for the next generation mobile communication system.
- To provide secure 3G-WLAN interworking in the SAE/LTE architecture, Extensible Authentication Protocol-Authentication and Key Agreement (EAP-AKA) is used.
- EAP-AKA protocol has several vulnerabilities such as disclosure of user identity, man-in-the-middle attack, Sequence Number (SQN) synchronization, and additional bandwidth consumption.
- The analyzes threats and attacks in 3G and proposes a new authentication and key agreement protocol based on EAP-AKA.
- The proposed protocol combines Elliptic Curve Diffie-Hellman (ECDH) with symmetric key cryptosystem to overcome the vulnerabilities present in the EAP-AKA protocol.

8.2.FGPA

- FGPA involves the hardware implementation of elliptic curve cryptography (ECC).
- Experimental results demonstrate that the FPGA implementation can speedup the point multiplication by 31.6 times compared to a software based implementation.
- The main contribution of FPGA based design is the resources sharing and parallel processing optimization.

9.Reference

1. Gnutella P2P Network. www.gnutella.com.
2. Kazaa P2P Network. www.kazaa.com.
3. TIB/Rendezvous. White paper, 1999.
4. S. Androutsellis-Theotokis and D. Spinellis. A Survey of Peerto-Peer Content Distribution Technologies. ACM Comput. Surv., 2004.
5. B. Arai, G. Das, D. Gunopulos, and V. Kalogeraki. Approximating Aggregation Queries in Peer-to-Peer Networks. In Proceedings of the 22nd International Conference on Data Engineering (ICDE), 2006.
6. R.Baldoni, C.Marchetti, A.Virgillito, and R.Vitenberg. Contentbased Publish-Subscribe over Structured Overlay Networks. In Proceedings of ICDCS, 2005.
7. G.Banavar, T.Chandra, B.Mukherjee, J.Nagarajarao, R.E.Strom, and D.C.Sturman. An Efficient Multicast Protocol for Content- Based Publish-Subscribe Systems. In Proceedings of ICDCS 1999, 1999.
8. T. Bu and D. F. Towsley. On Distinguishing between Internet Power Law Topology Generators. In INFOCOM, 2002.
9. A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event noti_cation service. ACM Transactions on Computer Systems, 19(3):332.383, 2001.
10. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making Gnutella-like P2P Systems Scalable. In Proceedings of ACM SIGCOMM 2003, 2003.
11. J. Chen, L. Ramaswamy, and A. Meka. Message Diffusion in Unstructured Overlay Networks. In Proceedings of NCA, 2007.
12. P. Chirita, S. Idreos, M. Koubarakis, and W. Nejdl. Publish/ Subscribe for RDF-based P2P Networks. In Proceedings of the 1st European Semantic Web Symposium, May 2004.