Indian Journal of Science and Technology, *Vol 7(6), 871–876, June 2014*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Distributed Group Key Management Protocol over Non-commutative Division Semirings

G. S. G. N. Anjaneyulu[1] and A. Sanyasirao[2*]

[1]Division of Applied Algebra, SAS, VIT University, Vellore, Tamilanadu, India; anjaneyulu.gsgn@vit.ac.in
[2]Dept. of ECE, Balaji Institute of Technology and Scince, Warangal, India; asr_bits2004@yahoo.co.in

Abstract

In this paper, we propose new method for group key distribution among 'four' persons using polynomials over non-commutative division semirings. We generate the common key or group key using polynomial symmetric decomposition problem. Security of the proposed protocol is based on PSDP over non-commutative division semirings. This can be extended to a group of 'n' persons also in similar fashion.

1. Introduction

Digital signature has become one of the most primal important techniques in modern information security system for its functionality of providing data integrity and authentication. Many emerging network applications and functions (such as teleconference and information dissemination services) are based upon a group communications model. As a result, protecting the security of group communications becomes a complicated and critical networking research issue. One primitive problem area in securing group communication is the group key management problem, which is associated with the secure distribution and refreshment of user keying material.

As a consequence and upshot of the increased popularity in group oriented applications and protocols, group communication becomes essential in many situations, from network layer multicasting to application layer teleconferencing and videoconferencing. Regardless of the environment and situation, security protocols and their services are necessary to provide communication privacy and integrity, authentication.

Let us presume that a little group of people at a conference or seminar has come together in a room for an ad hoc meeting. They would like to setup a structure of wireless network session with their laptop computers for the whole interval of the meeting. They would like to share and distribute information securely so that no one outside the room can eavesdrop and cram about the information or main contents of the meeting. The people physically there in the room be familiar with the contents and belief one another. However, they do not contain any a priori means of digitally identifying and authenticating each other, such as shared secrets or public key certificate authority or access to trusted third party key distribution centers. An invader can supervise and alter all traffic on the wireless communication channel and may also put an effort to impersonate as a legitimate member of the group. There is no system for safe and secure communication channel to connect the computers. The problem is: how can the group setup a system for safe and secure session among their computers under these circumstances? The network in the scenario depicted above is a paradigm of an Ad-hoc network in which entities create a communication network with little or no infrastructural support.

In recent years, mobile ad-hoc networks have established a massive deal of concentration in both academia and industry because they proffer anytime-anywhere networking services. Ad-hoc networks have devastating influence on military warfare where troops can be deployed anywhere in the world and in any hostile environment. Moreover, they need to set up a secure communication channel among themselves swiftly and also they have to preserve the security of that channel in

*Author for correspondence

case of group detachment and re-attachment. As wireless networks are being rapidly deployed, secure wireless environment will be essential and unavoidable. To ensure security, encryption can be utilized to protect messages exchanged among group members. A primal and vital element of any encryption technique is the cryptographic key (also called group key in ad-hoc networks). In ad-hoc networks, secure distribution of the group key to all valid members is a very huge and tedious deal.

1.1 Related Works

Key agreement in ad-hoc networks is categorized into three main modules:

1) Centralized group key management protocols: A particular entity called the Key Distribution Center (KDC) is performed for controlling the whole group members.
2) Decentralized group key management protocols: The management of a huge group is separated among subgroup managers, trying to minimize the problem of concentrating the work in a distinct place.
3) Distributed group key management protocols: There is no explicit KDC, and all the members contribute in the production of the group key and each member appends to a portion of the key.

1.1.1 Centralized Group Key Management Protocols

With a unique managing entity, the central server is a single point of breakdown. The group privacy is reliant on the successful execution of the single group controller; when the controller is not functioning, the group becomes vulnerable because the keys, which are the pedestal for the group privacy, are not being generate/regenerated and distributed. Furthermore, the group may become too hefty to be managed by a unique party, thus elevating the issue of scalability. The group key management security protocol can be implemented in a centralized system seeks to minimize the necessities of both group members and KDC in order to supplement the scalability of the group management. The competence of the protocol can be calculated by: Storage requirements, Size of messages, Backwards and forward secrecy and Collusion. Some well-liked centralized protocols are: Group Key Management Protocol (GKMP)[3], Logical Key Hierarchy (LKH)[4], One-way Function Tree (OFT)[5], Efficient Large-Group Key (ELK) Protocol[6] etc.

1.1.2 Decentralized Group Key Management Protocols

In the decentralized subgroup approach, the bulky members of group are split into small subgroups. Different controllers recycled to manage each subgroup, minimizing the problem of directing the work on a single place. In this approach, more entities are permitted to be unsuccessful before the whole group is affected. We use the following characteristics to estimate the efficiency of decentralized frameworks: Key independence, Decentralized controller, Local rekey, Keys vs. data and Rekeyper membership. Scalable Multicast Key Distribution[7], Kronos[8], Intra-Domain Group Key Management (IGKMP)[9], Hydra[10] are some of the popular security protocols that go after the decentralized architecture.

1.1.3 Distributed Group Key Management Protocols

The distributed key administration approach is characterized by having no group controller. The group key can be either produced in a contributory fashion, where all members contribute their own share to computation of the group key, or computed by one member. In the latter case, even though it is fault-tolerant, it may not be secure to leave any member to generate new keys since key generation involves secure mechanisms, such as random number generators, that may not be available to all members. Moreover, in the majority contributory protocols (apart from tree-based approaches), processing time and communication requirements amplify linearly in term of the number of members. Additionally, contributory protocols require every user to be aware of the group membership list to make sure that the protocols are robust. We use the following attributes to compute the competence of distributed key management protocols:

- Number of rounds: The protocol should try to minimize the number of iterations among the entities to trim down processing and communication requirements.
- Number of messages: The overhead introduced by each message exchanged between members produces unbearable delays as the group increases. Therefore, the protocol should have need of a minimum number of messages.
- DH key: Identify whether the protocol employs Diffie-Hellman (DH) to generate the keys. The use of DH to generate the group key implies that the group key is produced in a contributory fashion.

- Number of Exponentiations: Since exponentiations inflict more overhead than additions/multiplications, the number of exponentiations performed by a node should be kept to as minimum as possible. Some fashionable protocols in this category are Burmester and Desmedt (BD) Protocol[11], Group Diffie-Hellman Key Exchange (G-DH)[12], Octopus Protocol[13], Conference Key Agreement (CKA)[14], Diffie-Hellman Logical Key Hierarchy (DH-LKH)[15], Password Authenticated Multi-Party Diffie-Hellman Key Exchange (PAMPDHKE) Protocol[16].

Several emerging network applications (such as teleconference and information dissemination services) are supported upon a group communications model. As a consequence, securing group communications becomes a decisive networking research issue. Recently, Internet Research Task Force (IRTF) has framed Secure Multicast Research Group (SMuG)[17] to inspect the problem of securing group communications. One foremost problem area in securing group communication is the group key supervision problem, which is concerned with the secure distribution and refreshment of user keying material.

The intention of a key management system is to augment access control on top of efficient multicast communication such as over IP multicast. A standard technique to this end is to maintain a common group key that is identified to all multicast group members, but is unknown to non-group members. All group communication will be encrypted by means of this shared key. The significant problem for this approach is that in a dynamic membership atmosphere, clients will join and leave the group, therefore, efficiently altering the group key becomes a performance issue.

It is evident that a user join requests do not pose a question because all users in the group distribute a common group key prior to the new user joins, and therefore can vary to a new group key using the current group key. It is user depart requests that pose the scalability issue. Since the departing user shares the group key with other users, in order to distribute a new group key to the remaining members, other keys may have to be used. In the simplest case, the key server may have to send the new group key encrypted by a remaining user's individual key, which is only distributed between a user and the key server. If the number of users in the group is ' n' then complexity of this trouble-free scheme has a complexity of O(n). In the past few years, several schemes have been projected to progress rekeying performance, and these schemes can pick up the rekeying complexity from O(n) to O(log(n)) [19].

2. Cryptographic Assumptions on Non-Commutative Groups

2.1 Two Well-known Cryptographic Assumptions

In a non-commutative group G, two elements x, y are conjugate, denote $x \sim y$, if $y = z^{-1} x z$ for some $z \in G$. Here z or z^{-1} is known as a conjugator. Over a non commutative group G, we can define the following two cryptographic issues, which are related to conjugacy

Conjugator Search Problem (CSP): Given $(x,y) \in G \times G$, compute $z \in G$ such that $y = z^{-1} x z$

Decomposition Problem (DP): Given $(x,y) \in G \times G$ and $S \subseteq G$, compute $z_1, z_2 \in S$ such that $y = z_1 x z_2$

At present, we trust that for general non-commutative group G, both of the above problems CSP and DP are not tractable.

2.2 Symmetrical Decomposition and Computational Diffie–Hellman Assumptions over Non-commutative Groups

Enlightened by the above said problems, Zhenfu Cao et al.[2] defined the following Cryptographic problems over a non-commutative group G.

Symmetrical Decomposition Problem (SDP): Given $(x,y) \in G \times G$ and $m, n \in Z$, the set of integers, compute $z \in G$ such that $y = z^m x z^n$

Generalized Symmetrical Decomposition Problems (GSDP): Given $(x,y) \in G \times G$, $S \subseteq G$ and $m, n \in Z$, compute $z \in S$ such that $y = z^m x z^n$.

3. Building Blocks for Proposed Group Key Distribution

3.1 Integral Co-efficient Ring Polynomials

Let R be a ring with (R, +, 0) and (R,•, 1) as its additive Abelian group and multiplicative non-abelian semigroup, respectively. Now, we define positive integral co-efficient ring Polynomials. Let $f(x) = a_0 + a_1 x + \ldots + a_n x^n \in Z_{>0}[x]$ be

a given positive integral coefficient polynomial. We transform this polynomial as an element of R by selecting an element 'r' in R and in conclusion, we achieve

which is an element in R.

$$f(r) = \sum_{i=0}^{n} (a_i) r^i = (a_0) + (a_1) r + \ldots + (a_n) r^n$$

Further, if we regard r as an arbitrary element in R, then $f(r)$ can be looked as polynomial about r. The set of all this class of polynomials, taking over all $f(x) \in Z_{>0}[x]$, can be seemed the extension of $Z_{>0}$ with r, denoted by $Z_{>0}[r]$. We entitle it as the set of 1-ary positive integral coefficient R – Polynomials.

3.2 Semiring

A Semiring R is a non-void set, on which the compositions of addition & multiplication have been defined such that the following conditions are satisfied.

(i) (R, +) is a commutative monoid with identity element "0"
(ii) (R, •) is a monoid with identity element 1.
(iii) Multiplication distributes over addition from either side
(iv) 0 • r = r • 0 for all r in R

3.3 Division Semiring

An element r in a semiring R, is a "unit" if there exists an element r^1 of R satisfying $r \bullet r^1 = 1 = r^1 \bullet r$

Then that element r^1 is called the inverse of r in R. If such an inverse r^1 exists for a unit r, it must be unique. We will generally denote the inverse of r by r^{-1}. It is uncomplicated to see that, if r & r^1 units of R, then $r \bullet (r^1)^{-1} = (r^1)^{-1} \bullet r^{-1}$ and also, in particular $(r^{-1})^{-1} = r$.

We will designate the set of all units of R, by U(R). This set is always non-empty, since it contains "1" & is not all of R, since it does not contain '0'. We have just identified that U(R) is a submonoid of (R,•), which is in fact a group. If U(R) = R/{0}, Then such R is called as a division semiring.

3.4 Polynomials on Division Semiring

Suppose (R, +, •) is a non-commutative division semiring. Let us think about positive integral co-efficient polynomials with semiring assignment as follows.

In the begining, the concept of scale multiplication over R is already on hand. For $k \in Z_{>0}$ & $r \in R$. Then (k) r =r + r + r +… + r + r (k times)

For k = 0, it is very trivial to define (k) r = 0

Property 1. $(a)r^m \bullet (b)r^n = (ab) \bullet r^{m+n} = (b)r^n \bullet (a)r^m$, $\forall$ a,b,m,n $\in$ Z , $\forall$ r$\in$R

Remark: Note that in common

$(a)r \bullet (b)s \neq (b)s \bullet (a)r$ when $r \neq s$, since the multiplication in R is non-commutative.

Now, we define positive integral coefficient semiring polynomials. Let us suppose that $f(x) = a_0 + a_1 x + a_2 x^2 + \ldots\ldots + a_n x^n \in Z_{>0}[x]$ be a given positive integral coefficient polynomial. We put this polynomial in R by selecting an element r in R & finally, we obtain $f(r) = a_0 + a_1 r + a_2 r^2 + \ldots\ldots + a_n r^n \in R$

Similarly $h(r) = b_0 + b_1 r + b_2 r^2 + \ldots\ldots + b_m r^m \in R$ for some polynomial h(x) $Z_{>0}[x]$ and for some n ³ m. Then we have the following upshots

Theorem1: f(r).h(r) = h(r).f(r) for f(r), h(r) € R

Remark: If r & s are two distinct variables in R, then f(r) •h(s) ¹ h(s) •f(r) in general.

3.5 Additional Cryptographic Assumptions on Non-commutative Division Semirings

Suppose that (R, +, •) is a non-commutative division semiring. For any a$\in$R, we identify the set $P_a \subseteq R$ by $P_a \triangleq \{f(a) / f(x) \in Z_{>0}[x]\}$. Then, let us think about the new versions of GSD and CDH problems over (R,•.) with respect to its subset P_a, and entitle them as Polynomial Symmetrical Decomposition (PSD) Problem and Polynomial Diffie – Hellman (PDH) problem – respectively:

Polynomial Symmetrical Decomposition (PSD) problem over Non- commutative division semiring R: Given arbitrary triple $(a, x, y) \in R^3$ and $m, n, \in Z$, find $z \in P_a$ such that $y = z^m x z^n$

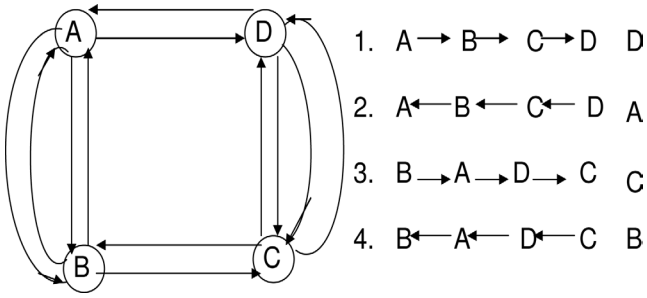
4. Outline of the Article

The presentation of this article is as follows. The Proposed Group Key Distribution has been presented in section 5 and followed by security analysis of the Proposed Group Key Distribution in section 6. The article will be completed by conclusions in section 5.

5. Proposed Group Key Distribution

Let (R, +, •) be the non-commutative division semiring. Let there are four members in the group be A, B, C, D.

One will Select two elements p≠ 0, q ≠0 in R, and m, n € I as public parameters. Four members A, B, C, D would choose f_i (x) for 1≤i≤4 , as positive integral co-efficient polynomials independently. Next, they would calculate f_i (p) ≠ 0 in R, as their individual private keys. Therefore, they can compute $y_i = \{f_i(p)\}^m\, q\{f_i(p)\}^n$ for 1≤i≤4 as their public keys. There are many routing algorithms available, by applying them we can able to compute contributed common group key in all possible different routes. So finally, they generate the contributed common group key by applying following algorithm



In route 1, A will send his public key to B, then B will calculate part of the common key and then he will send to C, at last C will do the same and Finally D can compute common group key. Similarly the same thing can be done in other routes. So that A, C, and B can able to calculate the same common group key. Then the common and contributed group key in all routes is

$$k = \{f_1(p).f_2(p).f_3(p).f_4(p)\}^m\, q\{f_1(p).f_2(p).f_3(p).f_4(p)\}^n$$

Remark: The above algorithm can be generalized among the group of n – people also. For this, we find n-different routes and in each route, one will get group common key.

6. Soundness and Security Analysis

The key inspiration is that pickingup a polynomial f(x) randomly, with semiring assignment and for any p € S, such that f(p) (≠ 0) €(R, +, •). Any hacker P* has no way to tract the polynomial f(x) € $Z_{>0}$[x] such that f(p) (≠0) € (S, +, •), even if he has unlimited computational power. Let n be the number of elements of S, P* best strategy is to estimate the value of p, and there are n choices for p. Hence, even with infinite computing power, the cheating prover P* with a insignificant probability to trace the exact private key f(p) € S, so as to present a valid response for an invalid common key. Hence this key agreement protocol is sound.

The strength and security of the proposed key agreement protocol is based on the intractability of Polynomial Symmetrical Decomposition Problem over the given non-commutative division semirings. This PSDP is another hard mathematical problem on all non-commutative algebraic structures. This problem is generalization of conjugacy problem on non-commutative algebraic structures and polynomials as elements.

7. Conclusions

In this research article, we designed a distributed group key agreement protocol among four persons, depending on general non-commutative division semiring. The key notion behind our scheme lies that we derive polynomials over the given non-commutative algebraic system as the underlying work structure for constructing key agreement scheme. The security of the proposed scheme is depending on the intractability of Polynomial Symmetrical Decomposition Problem over the given non-commutative division semirings. The base of PSDP is Conjugacy problem, which is intractable on non-commutative algebraic structures.

8. References

1. Deering SE, Cheriton DR. Multicast routing in datagram internet works and extended LANs. ACM Transactions on Computer Systems. 1990 May; 8(2):85–110.
2. Cao Z, Dong X, Wang L. New public key cryptosystems using polynomials over Non-commutative rings; 2007. Available from: http://eprint.iacr.org/2007/009.pdf
3. Harney H, Muckenhirn C. Group Key Management Protoco(GKMP) specification. RFC 2093; 1997.
4. Wallner D, Harder E, Agee R. Key management for multicast: issues and architectures. RFC 2627; 1999.
5. McGrew DA, Sherman AT. Key establishment in large dynamic groups using one-way function trees. Techical Report No. 0755 (May). Glenwood, MD: TIS Labs at Network Associates, Inc.; 1998.
6. Perrig A, Song D, Tygar JD. ELK, a new protocol for efficient large-group key distribution. IEEE Security and Privacy Symposium; 2001 May.
7. Ballardie A. Scalable multicast key distribution. RFC 1949; 1996.
8. Setia S, Koussih S, Jajodia S, Harder E. Kronos: a scalable group re-keying approach for secure multicast. IEEE Symposium on Security and Privacy; 2000 May.

9. DeCleene B, Dondeti L, Griffin S, Hardjono T, Kiwior D, Kurose J, Towsley D, Vasudevan S, Zhang C. Secure group communications for wireless networks. MILCOM; 2001 Jun.
10. Rafaeli S, Hutchison D. Hydra: a decentralized group key management. 11th IEEE International WETICE: Enterprise Security Workshop; 2002 Jun.
11. Burmester M, Desmedt Y. A secure and efficient conference key distribution system. EUROCRYP'94, LNCS(950). 1994. p. 275–86.
12. Steiner M, Tsudik G, Waidner M. Diffie-hellman key distribution extended to group communication. 3rd ACM Conference on Computer and Communications Security; 1996 Mar. p 31–37.
13. Becker C, Wille U. Communication complexity of group key distribution. 5th ACM Conference on Computer and Communications Security. 1998 Nov.
14. Boyd C. On key agreement and conference key agreement. Information Security and Privacy: Australasian Conference, LNCS(1270). 1997; 294–302.
15. Kim Y, Perrig A, Tsudik G. Simple and fault-tolerant key agreement for dynamic collaborative groups. 7th ACM Conference on Computer and Communications Security. 2000 Nov.
16. Asokan N, Ginzboorg P. Key agreement in ad hoc networks. Journal of Computer Communications. 2000; 23:1627–37.
17. Internet Research Task Force (IRTF). The secure multicast research group (SMuG). Available from: http://www.ipmulticast.com/community/smug/