[1]C. S. Nagamanjularani

R. Leela Velusamy[2]

Quaternary Tree Group Diffie- Hellman Based Group Key Management Protocol for Dynamic Peer Groups

Abstract: - Diffie-Hellman (DH) protocol for 2-party has made the key exchange to be simple yet powerful. It is widely used in many communication-oriented protocols for establishing the keys in public communication channels. With millions of internet users connected globally and using various group oriented applications, communication among them has to be in a secured manner. Due to its simplicity, DH has been further extended to group setting and number of protocols are developed. Establishing Group Key and regenerating it during membership changes is a challenging task in communication among large groups. This paper proposes a quaternary tree method that uses the key tree concept and Group Diffie Hellman protocol, GDH.3 to establish the common key. GDH.3, an extension of DH for n parties is best suited for large dynamic groups. Quaternary tree reduces the height almost by half as against binary trees, making the rekeying much easier. As the group size increases, GDH.3 outperforms other existing group protocols based on DH, since it requires much lesser number of exponentiations.

Keywords: Group Key Management, Group Diffie Hellman, Quaternary Tree.

I. INTRODUCTION

Wireless Sensor Networks(WSNs) is an important fundamental technology of the Internet of Things (IoT). The IoT technology has advanced in a wide variety of applications, including but not limited to homes, factories, hospitals or city streets. With the adoption of IoT technology in our day to day lives tremendously, security challenges have become a major concern. Improving security in IoT has become a major research topic in both academia and industry due to its increased benefits. With the proliferation of IoT devices in diverse applications and millions of users, the unicast communication amid users from one-to-one has been changed towards the communication in groups. Messages can be delivered efficiently in the form of broadcasting or multicasting instead of one-to-one communication in the grouporiented applications. Multicast or broadcast is used as an efficient communication mechanism as it enables direct communication within the group members, and is more efficient when compared to an equivalent unicast communication. Secured group communication consists of providing confidentiality, authenticity, and integrity of messages exchanged within the group. Group key management(GKM) is one of the fundamental building blocks for communication in large groups in a secured fashion. GKM is a cryptographic method to maintain a common group key among group of members via insecured communication channel. The group key essentially is a secret key shared by all the group members to offer security mechanisms for secured group communication. The group members alone can utilize the shared secret to encrypt, decrypt, and authenticate exchanged messages.

A GKM approach is useful for many collaborative group based applications like e-learning, video conferencing, and interactive online games, and so on. These applications generally have a large number of users connected globally, whose membership can be changed frequently. Designing an efficient GKM scheme for extensive set of users is important in such scenarios.

Most efficient GKM schemes such as LKH, TGDH, CCEGK, and ELK use the tree-based architecture. All these schemes make use of binary tree structure. They use the two-party Diffie-Hellman technique for key exchange in order to establish the secret key to be used among the participants. This paper proposes a method that uses a quaternary tree and GDH.3 to generate the shared secret key among the participants. It is advantageous over the

[1*]Department of Computer Science and Engineering,

[2]National Institute of Technology, Tiruchirappalli, Tamil Nadu, 620015 INDIA

***Corresponding author:** C.S. Nagamanjularani

*Department of Computer Science and Engineering,

other tree based schemes as the height of a perfect quaternary tree is approximately half the height of a perfect binary tree for the same set of nodes.

This paper is further considered as follows: Section II deals with preliminaries likekey establishment, Quaternary Trees, and DH 2-party key exchange. Section III describes GDH.3

in detail. Section IV further elaborates about GKM with its classification, Security Features required, and Operations that can be performed over groups. Section V discusses the related tree oriented group key protocols. Section VI describes the proposed method while its performance is estimated theoretically with the existing methods in Section VII. The paper concludes in Section VIII finally.

II. PRELIMINARIES

A. Key Establishment

Establishing a key is a precursor to key management. Key Establishment is a protocol where a common secret is computed and available to two or more participating entities, for further cryptographic use. Depending upon the entities involvement in the key generation, Key Establishment

Protocols (KEP) can be classified to two approaches viz.,

1. Key Agreement (Exchange) Protocols (KAP)
2. Key Transport (Distribution) Protocols (KTP)

A KAP is a KEP, in which the parties agree on a key in such a way that each participating party influences the outcome. A KTP is a KEP in which one party generates the secret key and transfers it securely to all the other parties. After the completion of protocols of both types, only current set of group members will have the shared group secret.

After any addition of a new group member or the withdrawal of the existing member from the group, the secret value has to be changed to prevent unauthorized access by persons outside the group. Among the class of Key Establishment Protocols, KAP are more accepted than KTP. Key Agreement is trustworthy than Key Transport as it results in higher quality random keys. Moreover, Key Exchange based on the Diffie-Hellman Key Exchange, perfect forward secrecy can be achieved.

B. Quaternary Trees

A binary tree is a connected acyclic graph in which each node has at most 2 children. It can be extended to N-ary tree. An N-ary tree is a tree in which each node has at most N children. A Quaternary tree is a tree in which each node has no more than 4 children.

The total nodes in a perfect N-ary tree of height h is

$$n = \sum_{k=0}^{h} N^k = \frac{N^{h+1} - 1}{N - 1} \quad (1)$$

A complete N-ary tree has height

$$h = \mathrm{b} log_N((N-1)n)\mathrm{c} \quad (2)$$

Similar to binary trees, even N-ary trees can also be stored appropriately using an array. Assuming the root node to be at index 0, the parent node of a node at index k is at the location $\lfloor \frac{k-1}{N} \rfloor$. The children nodes of a node at index k are at locations kN+j for j=1,2,...,N.

C. Diffie-Hellman Key Exchange(DHKE)

Whitfield Diffie and Martin Hellman proposed DHKE, the first asymmetric scheme. Simple yet powerful, it renders an emperical solution to the key exchange problem, i.e., it allows two parties to obtain a common session key for the further cryptographic operations, by communicating over an insecure channel.

Charlie	David
choose $x = K_{pr,X} \in [2, q-2]$	choose $y = K_{pr,Y} \in [2, q-2]$
compute $X = K_{pub,X} = \alpha^x mod q$	compute $Y = K_{pub,Y} = \alpha^y mod q$
	$K_{pub,X} = A \xrightarrow{\quad}$
	$K_{pub,X} = A \xleftarrow{\quad}$
$SK_{XY} = K_{pub,Y}^{K_{pr,X}} = Y^x mod q$	$SK_{XY} = K_{pub,X}^{K_{pr,Y}} = X^y mod q$

Diffie-Hellman Protocol

The primary idea behind the DHKE is that exponentiation in , q prime, is a one-way function and it is commutative, i.e., $sk = (\alpha^x)^y = (\alpha^y)^x \bmod q$. The value $sk = (\alpha^x)^y = (\alpha^y)^x \bmod q$ is the combined secret that can be utilized as the common secret key between the two parties.

The two party DiffieHellman protocol for key exchange works as follows: Two parties, Charlie and David would like to establish a shared secret key. Charlie and David generate the public parameters q and α. Initially, q, a large prime is chosen. An integer $\alpha \in [2, q-2]$ is chosen. Further, they can establish a common secret key sk with the steps specified in DiffieHellman Protocol:

III.GENERALIZED GROUP DH KEY AGREEMENT

Diffie-Hellman 2-party key exchange has been extended to the group setting due to its simplicity and elegance. The primary inspiring feature is the increasing demand of several group oriented applications and the requirement to use them in secure way. As key establishment is the basis for group communication in a secured fashion, lot of attention has received naturally. Steiner et.al. proposed 3 group key distribution protocols, named GDH.1, GDH.2, and GDH.3. The distinction between these depend on how the group key is computed and the messages are communicated. As this paper adopts GDH.3, it alone is discussed here. The computational complexity increases as the group size grows, and it is essential to reduce the computation amount executed by each group member. GDH.3 has been intended to reduce the average calculation required for each principal. Unlike GDH.1/2, GDH.3 consists of four phases.In the first phase named as Upflow, contributions from all the principals is collected. After the upflow message is processed, P_{m-1} obtains $r^{Q\{Ek|k\in[1,m-1]\}}$.This value is broadcasted all other members in the second phase named as Broadcast phase. Every $P_i (i \neq m)$ eliminates its corresponding exponent and the result is forwarded to P_m. In the last phase, P_m gathers all inputs from the earlier phase, raises each of the input to the power of E_m and the resulting m-1 values are broadcasted to all the group members. Each P_i now has a value of the form $rQ\{Ek|k\in[1,m]\wedge k\neq i\}$ and the specific group key Sm can be easily generated.

IV.GROUP KEY MANAGEMENT

GKM is a process of managing the cryptographic keys in a cryptosystem. It is the set of processes which support establishing the key and maintain the ongoing keying relationships between members, including replacing previous keys with new keys as and when necessary. The public key cryptographic algorithms are mainly used in key management. The most widely used public key algorithms for key exchange are Diffie Hellman, RSA, Elliptic curve DH of Elliptic Curve Cryptography. Over the last few years, GKM has become an active research area as it plays a major role in several group oriented

applications, providing group communication in secure fashion using common session key. For the effective communication

P_i P_{i+1}

$\xrightarrow{r\prod\{E_k|k\in[1,i]\}}$

Phase 1(Upflow): Round i; $i \in [1, m-2]$

P_i P_{m-1}

$\xleftarrow{r\prod\{E_k|k\in[1,m-1]\}}$

Phase 2(Broadcast): Round m-1

P_i P_m

$\xrightarrow{r\prod\{E_k|k\in[1,m-1]\wedge k\neq i\}}$

Phase 3(Response): Round n;

P_i P_{m-1}

$\xrightarrow{r\prod\{E_k|k\in[1,m]\wedge k\neq i|i\in[1,m-1]\}}$

P_i calculates $S = (S^{k_i{}^{-1}})^{r_i}$

Phase 4(Broadcast): Round m+1

GDH.3 Protocol

TABLE I: GDH.3 Notations

Notations	Meaning
P	The set of principals
m	number of Principals in the protocol
i,j,k	indices of group members (ranging in [1,n])
P_i	The i^{th} principal in P, where $1 \leq i \leq m$
q	order of the algebraic group
r	generator in the algebraic group delimited by q
E_i	random exponent generated by P_i
T,U	Subsets of $\{N_1,...,N_m\}$
IIT (T)	Product of all elements in subset T
Sm	Group key shared among m members

in groups, GKM has to initially generate a common session key and regenerate it according to the dynamic membership changes.

A. Classification

GKM is categorized depending on how the keys are established and distributed in the group. According to the literature, the group key management techniques can be divided into three categories:

- Centralized GKM
- Decentralized GKM
- Contributory GKM

In centralized approach, a trusted third party (TTP) is used as a group controller that establishes and distributes the group key to all other participants. The TTP is responsible to regenerate the group key during join or leave in order to control the whole group. It is well suited in one-tomany multicast scenarios. In decentralized approach, the entire group is splitted into several subgroups, and each subgroup is controlled by a group controller. In this approach, each group controller acts like a central server as in centralized method which controls and manages its own subgroup. In contributory approach, all the participating entities contribute equally to establish the group key. This avoid problems with single point of failure and the centralized trust present in the above two GKM methods.

B. Security Features

The desirable properties for a group key management protocol are as mentioned below:

- **Forward Secrecy:** means that a participant which leaves the group must not disclose future exchanged information.
- **Backward Secrecy:** means that a participant which joins the group must not disclose previous exchanged information.
- **Key Independence:** Group key must be independent i.e., a newly joined participant having the new group key must not infer the previous keys (this achieves backward secrecy) and a participant leaving the group and having the previous group key must not infer the new keys (this achieves forward secrecy).
- **Key Freshness:** A key is considered as fresh, if it is guaranteed to be a newly generated key.

C. Operations

In the group communication system, a group key management protocol must maintain the group secrets to be available only to the group members according to all the membership change operations. The several operations needed in GKM are:

- Join
- Leave
- Rekeying

-Individual rekeying

-Batch rekeying

Whenever a new participant joins the group, GKM technique must be able to recompute the new group key in order to provide the backward secrecy such that newly joined member is unable to trace the previous group keys. Whenever an existing member is leaving the group, GKM technique must be able to recompute the new group key in order to provide the forward secrecy such that leaving participant is unable to trace the succeeding group keys. Both Join and Leave operations can either be done individually or in groups, hence further categorized as Single Join or Leave and Mass Join or Leave, also known as merging and partitioning. Rekeying is the process of achieving a new group key during the membership changes i.e., either join or leave. It can be done in two ways, namely Individual rekeying and Batch rekeying. Individual rekeying (IR) is performed instantly whenever any participant wants to join or leave the group, while Batch rekeying (BR) is performed either after a certain period of time or after a fixed number of members want to join or leave the group. IR is relatively inefficient and there is no synchronization between keys and data. BR improves the efficiency and alleviates the non synchronization problem.

TABLE II: Comparison of Group Protocols

Parameter	GDH.1	GDH.2	GDH.3
number of rounds	2(m-1)	m	m+1
number of messages	2(m-1)	m	2m-1
combined message size	(m-1)m	(m-1)(m/2+2)-1	3(m-1)

exponentiations per P_i	(i+1) for m $i <$ m, for P_m	(i+1) for $i <$ m, for P_m	for $i < 1$),($m -$ 2 for $Pm-1$, m-1 for P_m
total exponentiations	$\frac{(m+3)m}{2} - 1$	$\frac{(m+3)m}{2} - 1$	5n-6
Advantages	Simple and straightforward	low number of protocol rounds	Lesser number of exponentiations per principal
Disadvantages	large number of rounds	total exponentiations is more compared to GDH.3	number of messages is more

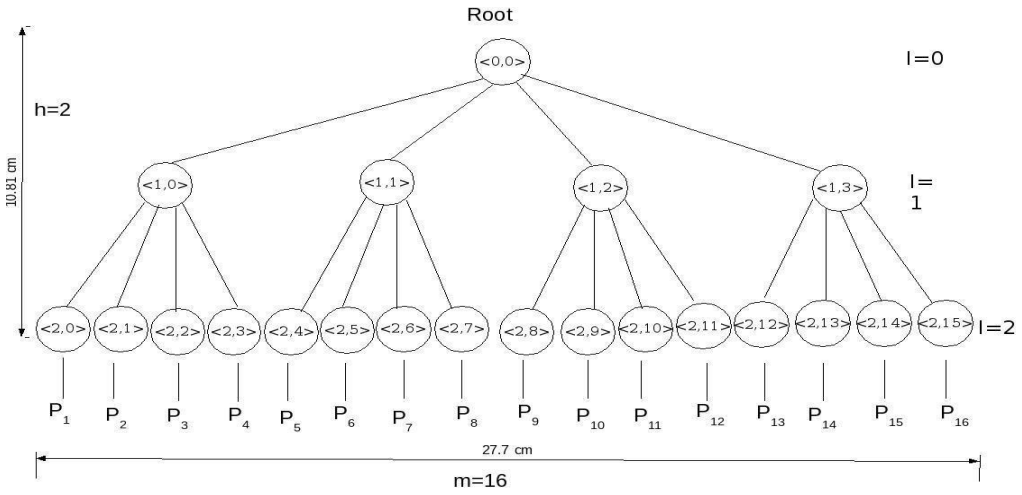

Fig. 1: Notation of a Quaternary Trees Key Tree

V. RELATED WORK

The tree based group key agreement protocols have increased the performance of join and leave operations from O(n) to

O(log n), where n is the size of group. This section narrates the tree based group key agreement protocols that use DH and GDH.2. A group key is contributed by all the group members and both forward and backward secrecy property is achieved. Hence all the tree based protocols are contributory in nature.

DH and GDH.2 protocols: DH based protocols use a two party DH key exchange, in which a group key is computed by two entities. GDH.2, a generalization of DH for group settings, based protocols allow multiple entities to establish the shared group key.

Kim et al. proposed the first hierarchy based Group Key Agreement protocol called TGDH, where the binary key tree and DH protocol concepts were considered. In this, join and leave operation is controlled by one of the group member called sponsor, preferably the rightmost deepest member in the key tree, which updates the group key during membership change. The authors compared TGDH with Skinny Tree (STR) protocol. The mass leave operation is the complex operation and the group key is regenerated as per Individual Rekeying instantly whenever a participant joins or leaves. The TGDH is relatively efficient both in join and leave events. The mass join operation needs more number of rounds. It is more efficient with respect to both computation and communication overhead.

The STR protocol adopts the key tree notion used in TGDH. The computation complexity of leave, mass join, and mass leave operations are more when compared to TGDH. It is efficient in communication and is more secured.

Zheng et al. developed a protocol, named Communication Computation Efficient Group Key (CCEGK), that adopts two existing EGK and TGDH protocols for large and dynamic groups. It performs better than TGDH,

EGK, and STR protocols. Tripathi et al. designed a Ternary Tree Based Method (TTBM) to decrease the complexity of overall rekeying. In TTBM, a subgroup is formed by three members and GDH.2 is used to establish a subgroup key. The complexity is reduced to $O(log_3 n)$ from $O(log_2 n)$. It adopts the mass join operation used in CCEGK. The size of group and number of members to join and leave is restricted.

BR based protocols provide a trade-off between performance and forward/backward secrecy. The group key is changed after a certain time period or whenever the join tree is full. Join and Exit Tree, a new logical key tree topology concept has been used in Join-Exit-Tree (JET), Dynamic Sub Tree (DST), and Weighted-Join-Exit-Tree (WJT). To initiate and deactivate the Join and Exit Tree, some constraints are required. Whenever members would like to join the group, initially they are added to the join tree. Rekeying is performed only when a fixed number of users have joined in the join tree. The newly joined members are transferred into the main tree, either when the join tree is filled or there is a leave operation in the main tree. The average time complexity for group operations in these protocols is O(log(log n)) which is better than O(log n). So these methods have less computation and communication overhead. When the group is large, these methods give greater performance in successive user join events. But the drawback is that the participants are assumed to know their leaving time at the time of joining the group itself, .

JDH protocol improves the efficiency in rekeying operations as the average join time is decreased to O(1) from O(log(log n)). It adopts the second relocation method employed in DST protocol. In JDH, whenever a new member joins, the new user is inserted at the root node of the main tree. It leads to a skewed tree when the join tree is empty. The cost of leave operation is O(log n).

VI. PROPOSED APPROACH

This paper proposes establishing a group key by using Group Diffie-Hellman GDH.3 for 4-parties with the key tree concept of TGDH. The proposed scheme makes use of quaternary tree as shown in fig 1. The root node is at level 0. Since we are using quaternary tree, any node can have at most 4 TABLE III: Quaternary Tree Notations

Notations	Meaning
P	The set of principals
m	number of Principals in the protocol
P_i	The i^{th} principal in P, where $1 \le i \le m$
n_i	The random number or secret key of P_i , where $1 \le i \le m$
h	height of the tree
q	prime integer
g	generator in the algebraic group
$< l,v >$	v^{th} node at level l
$K<l,v>$	Key of v^{th} node at level l
$BK<l,v>$	Blinded Key of v^{th} node at level l

children. Every node $< l,v >$ is related with the key $K_{<l,v>}$ and the blinded key (bkey) $BK_{<l,v>} = f(K_{<l,v>})$ where $f(k) = g^n \bmod q$, analogous to the DH protocol. A leaf node $< l,v >$ corresponds to the participant P_i, the node $< l,v >$ has P_i's random number n_i as a secret key. So, in general terms, each leaf node $< l,v >$ corresponds to the key

$K_{<l,v>}P_i = n_i$ where $1 \leq i \leq 4^l$. The bkey of each leaf node is computed based on the secret key of the corresponding node i.e., $BK_{<l,v>}P_i = f(K_{<l,v>})P_i = g^{ni} \ mod \ q$ where $1 \leq i \leq 4^l$. Once the bkeys of leaf nodes are computed, its corresponding parent node's bkey is computed by GDH.3. GDH.3 has 4 phases unlike GDH.1 and GDH.2 and works as follows for the 4 principals:

- Phase 1: P_1 generates a random number n_1 and sends $gn1$ to $P2$. P_2 generates a random number n_2 and sends g^{n1n2} to P_3.
- Phase 2: P_3 generates its own secret n_3 and broadcasts g^{n1n2n3} to all the principals.
- Phase 3: P_3 computes inverse to factor out its secret and sends g^{n1n2} to P_4.

P_2 computes inverse to factor out its secret and sends $gn1n3$ to $P4$.

P_1 computes inverse to factor out its secret and sends $gn2n3$ to $P4$.

- Phase 4: $P4$ broadcasts $gn1n2n4$,$gn2n3n4$,$gn1n3n4$ to all the principals.

The intended group key can be generated easily by the respective principals by adding its own secret. Furthermore, all the parent node's group keys can be calculated in the same manner as discussed above. The grand parents group key can be calculated recursively. The group secret computed by all members is at the root node $K_{<0,0>}$.

This protocol is well suited for dynamic groups. Once the group key has been generated, explicit protocols can be provided for joining and leaving group members. One participant must generate new random value while most of the keying material from the GDH.3 protocol is reused. The newly generated random exponent is in such a way that the fresh and previous keys are independent so that any joining member cannot determine the previous key (backward secrecy) and leaving member cannot determine the new key (forward secrecy). A protocol can also be provided for many members to join the group together.

While a member is joined, the principal that has generated a new random exponent must redo the phase just before the final broadcast phase. Then the joined member can broadcast a new message and all the other participants establish the new group key as in the final phase of GDH.3. Whenever a participant wants to leave the group, P_m can broadcast a new message for the last phase after eliminating the exponent component of the removed participant.

VII. PERFORMANCE

According to the literature, in many GKM approaches only GDH.2 has been used so far for the binary trees and ternary trees as well. As per the authors knowledge, GDH.3 and Quaternary trees in the context of establishing a group key for group communications is being used for the first time. This approach groups 4 members unlike 2 or 3 incase of binary or ternary trees. So the number of principals considered per round is more when compared to the earlier approaches. Quaternary tree height is half of the height of binary tree and less than the ternary tree, which in turn reduces the number of rounds required to compute a group key.

With GDH.3, there is a considerable difference in the total number of exponentiations, and the message size when compared to the existing DH based protocols. When compared to binary or ternary trees, in Quaternary trees, there will be less number of rekeying operations involved whenever a membership change occurs.

Ingemarsson et al. proposed ING protocol, requires 2 rounds lesser than GDH.3, i.e., (n-1) to generate the group key. The message size and total exponentiations being n(n-1) and n^2, it is much complex compared to GDH.3. Apart from this, ING requires all the participants to start synchronously.

A much more efficient protocol named BD proposed by Burmster and Desmedt requires only 2 rounds to compute group key. Though message size is better than GDH.3 as the size of group is increased, the total number of exponentiations is much higher than GDH.3. Hence, GDH.3 can be considered as well suited especially for the large groups as the total number of exponentiations is lesser than among all the existing protocols.

VIII. CONCLUSION

The paper proposes a new quaternary tree protocol for establishing the group key that makes use of GDH.3. It deals with establishing a group key for large dynamic groups. Instead of GDH.2 and binary trees, GDH.3, an

extension of DH for n parties and quaternary tree using key tree concept are used for generating the group key. The group size is considered to be in terms of 4^l for simplicity. The message size and total number of exponentiations of the proposed method are compared with the existing group DH protocols. It is observed that the proposed method outperforms other existing techniques especially when number of exponentiations are considered. As this paper deals with only the initial key generation, as a further work, merge and partition operations will be addressed. The established group key is secured as the proposed method uses GDH.3 for key exchange and GDH.3 security is already proved.

REFERENCES

[1] W.Diffie and M.Hellman. New directions in cryptography.IEEE Transactions on Information Theory, 22(6): 644652, 1976.

[2] I.Ingemarsson, D.Tang, and C.Wong. A conference key distribution system. IEEE Transactions on Information Theory, 1982.

[3] Michael Steiner,Gene Tsudik, and Michael Waidner. Diffie hellman key distribution extended to groups. In Third ACM Conference on Computer and Communications Security, 31-37.ACM Press, 1996.

[4] M.Steiner, G.Tsudik, and M.Waidner. Key agreement in dynamic peer groups. Technical report, Information Sciences Institute, 1999.

[5] C.Wong, M.Gouda, and S.Lam. Secure group communication using key graphs. IEEE/ACM Transactions on Networking, pages 16-30, 2000.

[6] Y.Kim, A. Perrig, G. Tsudik, Simple and fault-tolerance keyagreement for dynamic collaborative groups, In Proc. of 7th ACM Conference on Computer and Communications Security, pp. 235-244, 2000.

[7] Y. Kim, A. Perrig, G. Tsudik, Tree-based group key agreement, ACM Transactions on Information and System Security, 7(1): 60-96, 2004.

[8] Y. Kim, A. Perrig, G. Tsudik, Group key agreement efficient in communication, IEEE Transactions on Computers 53 (7): 905-921, 2004.

[9] Shanyu Zheng, David Manz, JIM Alves-Doss: A communication computation efficient group key algorithm for large and dynamic groups. Computer Networks: The International Journal of Computer and Telecommunications Networking. 51(1), pp:69-93, 2007.

[10] Omar Zakaria, Aisha-Hassan A.Hashim, Wan H.Hassan. An Efficient Scalable Batch Rekeying Scheme For Secure Multicast Communication Using Multiple Logical Key Trees. International Journal of Computer Science and Network Security, 14(11), 2014.

[11] Halford T.R., Courtade T.A., Chugg K.M., and Thatte G. Energy efficient group key agreement for wireless networks. IEEE Transactions on Wireless Communications, 14(10), 5552-5564, 2015.

[12] Omar Cheikhrouhou. Secure Group Communication in Wireless Sensor Networks: A survey. Journal of Network and Computer Applications. Volume 61, Pages 115-132, 2016.

[13] Shaukat Ali, Azhar Rauf, Naveed Islam, Haleem Farman, Bilal Jan, Murad Khan, Awais Ahmad. SGKMP: A scalable group key management protocol. Sustainable Cities and Society 39, 37-42, 2018.

[14] Yi - Hsuan Kung and Hsu - Chun Hsiao. GroupIt: Lightweight Group Key Management for Dynamic IoT Environments. IEEE Internet of Things Journal. 5(6), 5155-5165, 2018.

[15] Anshul Anand, Mauro Conti, Pallavi Kaliyar, Chhagan Lal. TARE: Topology Adaptive RekEying scheme for secure group communication in IoT networks. Wireless Networks, 2019.

[16] Aparna S. Pande, Yashwant Joshi, Manisha Y. Joshi. Analysis on Logical Key Hierarchy and Variants for Secure Group Communication. Computing, Communication, and Signal Processing, Advances in Intelligent Systems and Computing, vol 810. Springer, pp.419-430, 2019.

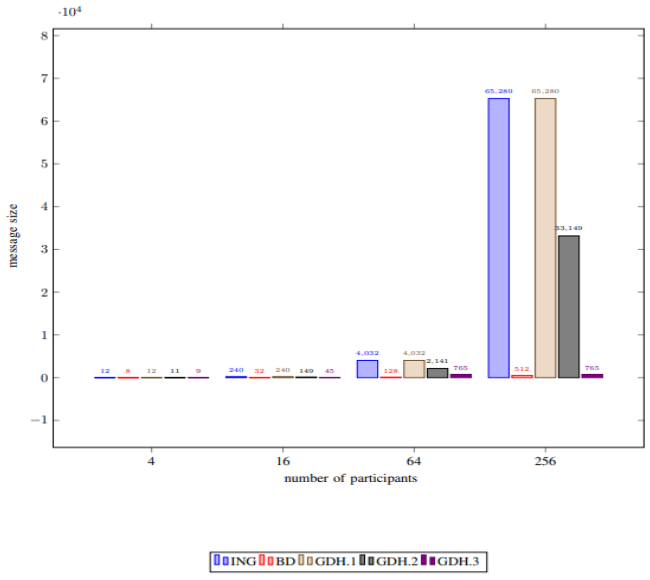

Fig. 2: Performance based on communication cost

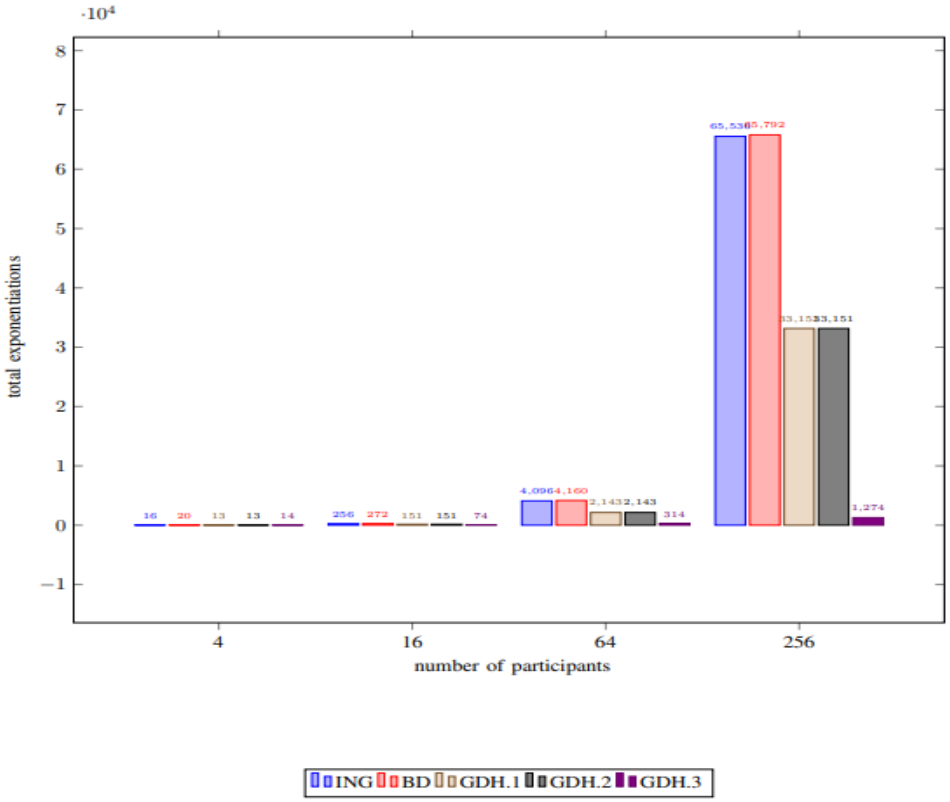

Fig. 3: Performance based on computation cost