Key Agreement with Modified Batch Rekeying for Distributed Group in Cognitive Radio Networks

N. Renugadevi, C. Mala

Key Agreement with Modified Batch Rekeying for Distributed Group in Cognitive Radio Networks

N. Renugadevi, C. Mala

Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India, nrenu79@gmail.com, mala@nitt.edu

Abstract. Cognitive radio networks have received more research interest in recent years as they can provide a favourable solution to spectrum scarcity problem prevailing in the wireless systems. This paper presents a new key agreement protocol called 'TKTOFT' with modified batch rekeying algorithm for distributed group oriented applications in cognitive radio networks by integrating a ternary key tree and an one way function. It is inferred from the experimental results that TKTOFT outperforms the existing one way function based protocol both in terms of computation and communication overhead. Hence, TKTOFT is suited for establishing secure and quick group communication in dynamic groups in cognitive radio networks.

Keywords: Distributed group, Ternary key tree, One way function, Batch rekeying, Cognitive radio networks

1 Introduction

The group oriented applications are in the rise due to rapid developments in internet technology and mobile computing technology. The distributed collaborative applications such as video conferencing, online games and pay-per-view have received special interest in recent years [1]. The unlicensed frequency spectrums are heavily congested due to rapid proliferation of wireless mobile devices working in these spectrum bands.

Cognitive Radio (CR) [2] can resolve the spectrum scarcity problem present in the existing wireless networks through dynamic operations such as spectrum sensing, spectrum mobility, etc. The concepts such as *Dynamic Spectrum Access* [3] and *Secondary Spectrum Access* [4] used in *CR Networks (CRNs)* allow the unlicensed or *CR Users (CRUs)* to access the free portions of licensed spectrum bands without disturbing the operations of licensed or *Primary Users (PUs)*.

To ensure privacy [5] and data confidentiality in distributed and collaborative groups in CRNs, a secure group communication should be provided by establishing the common group key for all the CRUs or members. *Group Key Management (GKM)* is a building block for providing security in group oriented applications. The distributed GKM or key agreement protocol is suitable for providing security in group communication of distributed and dynamic networks [6, 7] such as CRNs rather than centralized and decentralized GKM techniques [8].

Batch Rekeying (BR) approach reduces the total rekeying cost than Individual Rekeying (IR) as it performs rekeying operations for a batch of join and leave requests at a time to compute the new group key [9]. The tree based GKM protocols also help in minimizing both computation and communication cost during rekeying [10].

An alternative method of developing key agreement protocols is to employ an *One Way Function (OWF)* [11] rather than a standard Diffie-Hellman primitive to get the group key. OWF helps to achieve computational savings by eliminating the expensive modular exponentiations [12] and therefore OWF is a best candidate for smaller and portable mobile devices in CRNs. Hence, this paper proposes a *Ternary Key Tree based OFT (TKTOFT)* protocol which integrates OWF, tree based distributed GKM and BR approach to improve the efficiency of both computation and communication involved in tree based GKM protocols.

The rest of the paper is organized as follows. Section 2 explains briefly about the existing OWF based research work. The proposed TKTOFT protocol is discussed in Section 3. Section 4 concludes this paper.

2 Literature review

This section discusses about available OWF based research work and GKM protocols which can be adopted for CRNs.

An efficient authentication algorithm [13] based on OWF and symmetric key cryptography was proposed for authenticating local sensing reports in cooperative spectrum sensing in CRNs. In sensing assignment phase, each user generates two one-way chains both for empty decision and occupied decision for each channel.

Sherman and McGrew presented a novel centralized algorithm based on OWF tree namely OFT [14] for dynamic large groups. The bottom-up construction of key tree halves the number of bits to be broadcast during rekeying.

A key distribution protocol which uses parametric OWF and Euler's totient function [15] for achieving high level of security with reduced computation time was developed for secure multicast communication. This paper uses an N-ary tree to minimize the number of multiplications performed during leave operation in the group which in trun reduces the computation complexity.

Zhou et al. proposed a multicast key management technique called Threshold based OFT (TOFT) [16]. In this paper, threshold-key mechanism and quad tree were employed to improve the security of algorithm and to reduce the storage as well as rekeying cost.

An efficient centralized GKM was proposed which integrates key trees with one-way key derivation in order to reduce the communication complexity during rekeying operations [17]. The member itself can derive the key by itself and hence, the total number of keys to be transmitted by the server, i.e., bandwidth of rekeying message was reduced.

A Hash-chain based Authentication Protocol (HAP) [18] was designed for vehicular communication in which vehicle can be verified by combining its public

key and its hash code. A new GKM for dynamic access control in a large leaf class hierarchy was proposed [19] which improves previous related research works by using symmetric key cryptography and OWF with less computational and storage overheads.

An image based group key agreement protocol [20] was designed which employs OWF as an image morphing operation to hide the secret information of each member in the morphed image. An Extended Chaotic Map and password based three Party Authenticated Key Exchange (ECM-3PAKE) [21] was developed which provides both implicit and explicit key confirmation.

Li et al. proposed a secure BR scheme which employs two algorithms namely, Distributed BR Marking (DBRM) and Secure Distributed BR (SDBR) [22] for marking the key tree and re-computing the group key respectively.

3 Proposed TKTOFT Protocol

Subsection 3.1 explains briefly about OWF and Subsection 3.2 discusses about BR scheme used in the existing SDBR algorithm [22]. The proposed protocol which uses an improved BR scheme is described in Subsection 3.3.

3.1 One Way Function (OWF)

An n-bit hash (h) is a map from a binary string of any arbitrary length to n-bit binary string and the properties of OWF are as follows [23].
1. Preimage resistance: It is easy to compute y for the given x, such that y=h(x). But it is not possible to find x given h(x).
2. Second Preimage resistance: Given an input x, it is not possible to find different y, such that h(y)=h(x).
3. Collision resistance: It is not possible to find any two different x and y, such that h(x)=h(y). The security of proposed TKTOFT protocol is based on one way property of hash function.

3.2 Existing SDBR and DBRM Algorithms

The SDBR algorithm [22] uses a combination of binary key tree and OWF for generating the group key in a distributed dynamic collaborative group. It avoids a renewed node to be rekeyed more than once. In DBRM marking algorithm, four cases of join and leave possibilities are discussed.

The distributed OWF based key tree (OFT) used in SDBR is shown in Fig.1. The leaf nodes in the key tree store individual members' keys as they represent group members. An unique secret key 'k_i' of the member 'i' is generated using the pseudo random number generator and OWF is used to compute its corresponding Blinded Key, BK_i, such that $BK_i = f(k_i)$, where f is an OWF. The secret key of parent node is computed from the blinded keys of its children, i.e., parent's secret key = $F(BK_{2i+1}, BK_{2i+2})$, where F is a mixing function.

Each member in the key tree maintains the secret keys of nodes in its keypath

and also blinded keys of sibling of nodes in its keypath in order to compute the group key. The four cases of DBRM algorithm are discussed as follows.

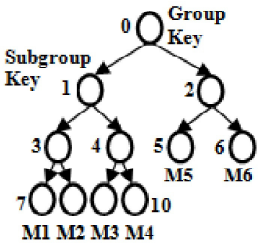

Fig. 1: Distributed OFT

Case 1. $J = L$ **:** Join members replace all the leave members. The algorithm marks nodes in the path from sibling nodes of all the leave members to root node as UPDATE.

Case 2. $J < L$ **:** The locations of J number of leave members with minimum height are selected to replace them with join members. The nodes associated with the remaining leave members are removed from the key tree. The nodes in the path from sibling nodes of replaced and remaining nodes to the root are marked as UPDATE.

Case 3. $J > L$ & $L = 0$ **:** Create a key tree 'STB' for new joining members and a new root node. Connect the existing key tree 'STA' as left child of newly created root node and STB as its right child. The root of STA and nodes in the path from sponsor of STB to its root are marked as UPDATE.

Case 4. $J > L$ & $L > 0$ **:** All the leave members are replaced by join members. Then, Case 3 is applied for remaining J-L joining members.

The DBRM algorithm has some limitations. In Case 3, insertion of STB as a right child of new root node increases the height of key tree which is shown in Fig.2. Whenever the system has $J > L$ & $L = 0$, this algorithm repeats the same operation which causes the increase in height of key tree. This step may significantly degrade the performance of BR algorithm when the system has only less number of joining members.

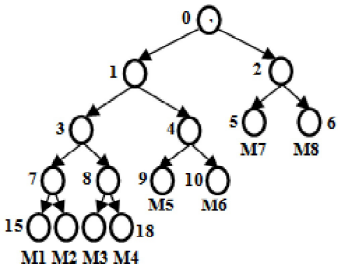

Fig. 2: OFT after inserting joining members

There may be a situation in which the system may have only join members without any leave members. If this happens repeatedly, then after a few rekeying operations, the key tree will become either skewed or unbalanced. The high rekeying cost will be the consequence, irrespective of the relationship between join and leave members in future rekey operation.

Though the system may have one or more leave members in Case 4, after replacing the leave memebrs with L join members, again it will have only join members, i.e., $J > L$ & $L = 0$. This will make the system to follow Case 3 which will result in further performance degradation.

An Efficient Distributed Key Agreement Scheme (EDKAS) [24] also uses a binary key tree and SDBR outperforms well than EDKAS by modifying all blinded keys which are known to leaving members. The efficiency of BR algorithm depends on the structure of key tree being used. An unbalanced key tree leads to increased number of operations which in turn increases the total rekey-

ing cost. The limitations of DBRM algorithm explained above are overcome in the proposed algorithm and is explained in the next subsection.

3.3 Proposed TKTOFT Protocol

The efficiency of BR method in this type of tree based GKA depends on the structure of key tree [10]. An efficient BR algorithm should maintain the balanced key tree in order to reduce the rekeying cost. Li et al. proved that the optimal key tree to provide a minimal rekeying cost in the group with unrestricted size during batch update is a *ternary key tree* [25]. Therefore, the proposed TKTOFT uses the ternary key tree to organize the members in the group which is depicted in Fig.3.

The leaf nodes indicate CRUs in distributed dynamic group in CRNs, i.e., members in the group. As ternary key tree is being used in the proposed protocol, each set of maximum of three CRUs can form a subgroup which corresponds to a subtree in the key tree.

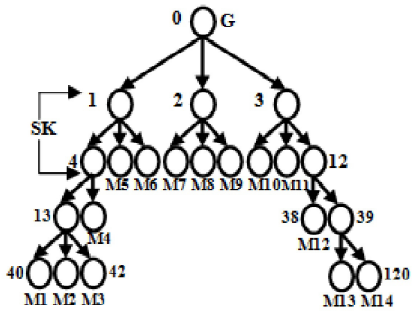

Fig. 3: Ternary OFT

The Subgroup Key (SK) is stored in the intermediate nodes whereas the root node has key for the entire group of CRUs. The rightmost nodes both in subtrees and entire key tree act as sponsors. The subgroup sponsor generates the sub group key and the sponsor of entire group establishes the Group key (G). This paper uses the terms 'members' and 'CRUs' alternatively to mean the group members.

An algorithm for *Batch Process (BP)* operation of the proposed protocol called *TKTOFT_BP* is given in Alg.1. It considers each specific case with the appropriate number of join and leave requests. This BP algorithm improves the Cases 3 and 4 and the remaining Cases are same as in DBRM. The abbreviations used in TKTOFT_BP are Key Tree (KT), Sub Key Tree (SKT), Root Node (RN), Link (LK), Internal Node (IN), Closest IN (CIN), Insertion Location (IL), Height (Ht), Minimum Height (MinHt), Node ID (NID), Minimum ID (MinID), New KT (NKT), Leave Members (LMs) and First ID (FID).

SKT is created for new joining members. Case 3 searches for an appropriate IL and inserts SKT, where the height of KT is not increased. First, it checks the links of root node. If the root has null links, then SKT is inserted as a subtree to the root. If the root is full, i.e., if it has three children, then TKTOFT_BP() searches the IN which is closest to root node (CIN).

If CINs in key tree have null links, then CIN with minimum ID is selected as an IL. Else, CIN with minimum height is selected at which SKT can be inserted into KT. After selecting IL, unlike exisiting methods, TKTOFT_BP() checks whether the insertion of SKT into KT at this IL will increase the height of KT or not. It will be merged with KT only when there is no increase in height. Otherwise, the root node will be selected as IL.

Algorithm 1 TKTOFT_BP(KT, J, L)

This algorithm modifies only Cases 3 & 4 discussed in subsection 3.2
Begin
Step 1 % Modified Case 3
If ($J > L$ **&** $L = 0$) **then**
Begin Case
Create a SKT
5: **If** ($LK(RN) = \phi$) then
IL=LK(RN)
Else
If ($LK(CIN) = \phi$) then
$IL \leftarrow CIN_{MinID}$
Else
$IL \leftarrow CIN_{MinHt}$
Endif
If ($Ht(NKT) > Ht(KT)$) then
$IL \leftarrow RN$
Endif
Endif
End Case
Endif
Step 2 % Modified Case 4
If ($J > L$ & $L > 0$) **then**
Begin Case
Create a SKT
$NID \leftarrow$ Node IDs (LMs)
$IL \leftarrow$ FID in sorted NID
If ($Ht(NKT) > Ht(KT)$) then
IL= go to 5
Endif
End Case
Endif
End

In Case 4, the location of leave member which is closest to root node is selected first as an IL. If the insertion of SKT at this IL increases the height of KT, then an appropriate IL is chosen using Case 3. Else, SKT is inserted at this selected IL. In Fig.3, when the system has 3 leave members (M3,M8,M12) and 6 join members, it results in a key tree shown in Fig.4. The location of M8 is selected as an IL, where the SKT created for 6 join members is inserted without increase in height of KT.

Thus, the proposed BR algorithm always chooses the correct IL in order to maintain the balanced key tree. It will select the root of KT as IL only after checking all the possibilities for inserting SKT into KT without increasing the height of KT. Thus, the number of operations to generate the group key tree will be reduced.

If selected IL has null link, then SKT is inserted as its child node. Else, a new

IN is generated. The member stored at IL is connected as a left child and root of SKT is connected as a middle child of newly created IN. In all the cases of algorithm, the sponsor nodes are selected based on the group operations (i.e., join and leave) and positions of both join and leave members in the key tree. All the sponsors in the key tree recompute the group key by updating their secret key. The remaining members will compute the new group key after receiving the broadcast message from the sponsor.

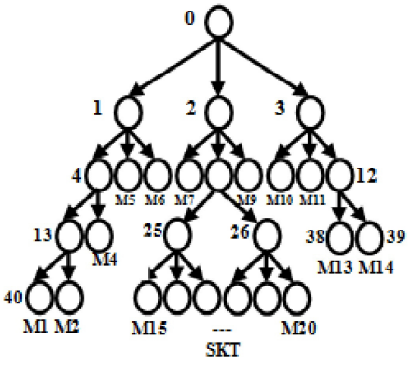

Fig. 4: Merging SKT with KT

4 Results and Discussion

The performance of proposed TKTOFT protocol is analysed and compared with the existing SDBR protocol. The computation complexity is decided based on a) time to generate initial group key and b) number of secret key computation of parent node. The communication complexity is determined based on the number of renewed nodes generated in the key tree [6,22] which are non-leaf nodes whose keys are modified during BR operation.

A group with size 3^5 (243 members) was considered for generating the initial group key. In Figs.5 and 6, x-axis represents the group size. Fig.5 compares the time to compute the initial group key which is represented in y-axis between SDBR and TKTOFT. From the figure it is inferred that TKTOFT takes less time to generate the group key than SDBR for the same group size.

Fig.6 depicts the performance analysis between the existing and proposed protocols based on the number of secret key computation of parent node in the key tree. As the number of internal nodes in the ternary key tree is less than binary key tree for the same group size, TKTOFT performs less number of key computations when compared to SDBR protocol.

It is clearly seen from Figs.5 and 6 that, initially there is no big difference between SDBR and proposed TKTOFT. But, when the group is increased, both key generation time and the number of secret key computation in SDBR are also increased. As SDBR uses a binary key tree, the height of key tree is increased even for a small change in group size. The increase in height leads to performance degradation in SDBR.

As proposed TKTOFT uses ternary key tree, its key tree height is minimum when compared to SDBR for the same group size. This reduced height of the key tree helps in minimizing the group key generation time and number of key computation of parent nodes in the key tree. The difference in values of y axis both in SDBR and TKTOFT is more prominent when the group has more number of members. From Figs.5 and 6, it is concluded that the proposed TKTOFT protocol has reduced computation complexity than SDBR protocol.

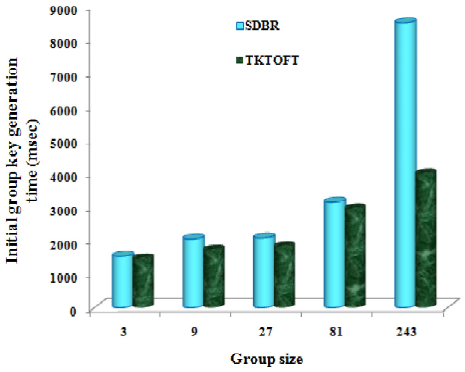

Fig. 5: Group size Vs Initial group key generation time

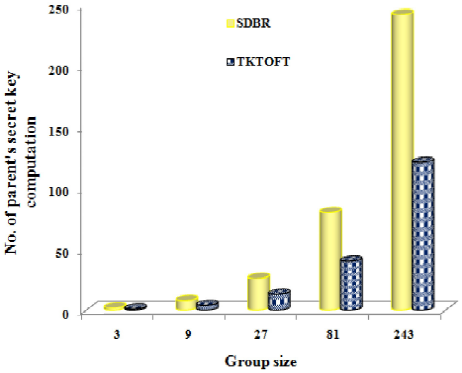

Fig. 6: Group size Vs Number of parent's secret key computation

As ternary key tree is being used in this paper, a group with 243 (3^5) members was considered during each iteration of batch rekeying. The values for number of join and leave members in the group were varied between 0 and 81 for measuring the total number of renewed nodes which indicates the communication complexity. In Fig.7, the total number of join members (J) and total number of leave members (L) are represented in x and y-axis respectively. The total number of renewed nodes created in the key tree are mentioned in z-axis of Fig.7.

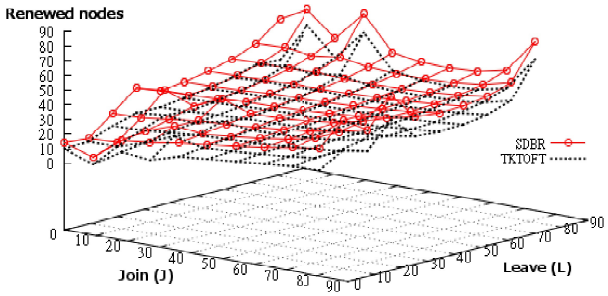

Fig. 7: No. of Join & Leave operations Vs Renewed nodes

From the graph, it is understood that TKTOFT has generated less number of renewed nodes than SDBR. The reasons are, a) the height of key tree is minimum, b) maintaining the same key tree height even when J is large with no leave members by choosing the correct insertion location for the sub key tree and c) pruning the leave members and choosing the appropriate location for merging. These prevent increase in key tree height which in turn minimizes the number of nodes to be renewed in the key tree. In this figure also, the significant difference in total renewed nodes can be clearly seen when J and L are large. Because, SDBR replaces the leave members with join members which result in the same key tree height instead of pruning them.

In Case 3 of SDBR, SKT is inserted at the root node without checking the status of KT. This leads to an unbalanced key tree with increased height. These limitations have been overcome in the proposed TKTOFT BP algorithm which gives an improved performance. TKTOFT protocol reduces the number of renewed nodes by choosing the appropriate insertion location to merge the key tree created for joining members and by pruning the leave members when $J > L$ & $L > 0$.

From the experimental analysis, it is concluded that the modified Cases 3 and 4 of the proposed protocol improves the performance in terms of both computation and communication complexities. Hence, this proposed TKTOFT can be adopted for a distributed group with highly dynamic scenarios in CRNs.

5 Conclusion

Cognitive radio network can help in providing a quick communication as its nodes solve spectrum scarcity problem prevalent in the present wireless systems. An improved distributed group key agreement called TKTOFT has been proposed in this paper which integrates ternary key tree, one way function and modified batch rekeying algorithm. The experimental results show that TKTOFT improves the efficiency of both computation and communication. Hence, this proposed protcol is suited for providing quick and secure group communication among cognitive radio devices in distributed collaborative applications in cognitive radio networks.

References

1. Daghighi, B., Kiah, M. L. M., Shamshirband, S., Rehman, M. H. U.: Toward secure group communication in wireless mobile environments: Issues, solutions, and challenges. Journal of Network and Computer Applications. 50 (2015) 1-14
2. Mitola, J., and Maguire Jr, G. Q.: Cognitive radio: making software radios more personal. IEEE Personal Communications. **6**(4) (1999) 13-18
3. Grandblaise, D., Bourse, D., Moessner, K., Leaves, P.: Dynamic spectrum allocation (DSA) and reconfigurability. IEEE Communications Magazine (2004) 72-81
4. Wyglinski, A. M., Nekovee, M., Hou, T.: Cognitive radio communications and networks: principles and practice. Academic Press (2009)
5. Armando, A., Bocci, G., Chiarelli, G., Costa, G., De Maglie, G., Mammoliti, R., Merlo, A.: Mobile App Security Analysis with the MAVeriC Static Analysis Module. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), **5**(4)(2014) 103-119
6. Kiyomoto, S., Fukushima, K., Miyake, Y.: Design of categorization mechanism for disaster-information-gathering system. J Wirel Mob Netw Ubiquitous Comput Dependable Appl, **3**(4) (2012) 21-34
7. Pokri, B., Kro, S., Draji, D., Pokri, M., Rajs, V., Mihajlovi, , Kneevi, P., Jovanovi, D.: Augmented Reality Enabled IoT Services for Environmental Monitoring Utilising Serious Gaming Concept. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), **6**(1) (2015) 37-55

8. Zou, X., Ramamurthy, B., Magliveras, S. S.: Secure group communications over data networks. Springer Science & Business Media (2007)
9. Lee, P.C., Lui, C.S., Yau, K.Y.: Distributed collaborative key agreement and authentication protocols for dynamic peer groups. IEEE/ACM Transactions on Networking **14**(2) (2006) 263-276
10. Kim, Y., Perrig, A., Tsudik, G.: Simple and fault-tolerance key agreement for dynamic collaborative groups. In: 7th ACM Conference on Computer and Communications Security. (2000) 235-244
11. Diffie, W., Hellman, M. E.: New directions in cryptography. Information Theory, IEEE Transactions on, **22**(6) (1976) 644-654
12. Boyd, C., Mathuria, A.: Protocols for authentication and key establishment. Springer Science & Business Media (2003)
13. Rif-Pous, H., Garrigues, C.: Authenticating hard decision sensing reports in cognitive radio networks. Computer Networks, **56**(2) (2012) 566-576
14. Sherman, A. T., McGrew, D. A.: Key establishment in large dynamic groups using one-way function trees. Software Engineering, IEEE Transactions on, **29**(5) (2003) 444-458
15. Vijayakumar, P., Bose, S., Kannan, A., Subramanian, S. S.: An effective key distribution protocol for secure multicast communication. In Advanced Computing (ICoAC), IEEE 2010 Second International Conference on. (2010) 102-107
16. Zhou, F., Xu, J., Lin, L., Xu, H.: Multicast key management scheme based on TOFT. In 10th IEEE International Conference on HPCC'08. (2008) 1030-1035
17. Lin, J. C., Lai, F., Lee, H. C.: Efficient group key management protocol with one-way key derivation. In IEEE Conference on Local Computer Networks. (2005) 336-343
18. Sulaiman, A., Raja, S. K., Park, S. H.: Improving scalability in vehicular communication using one-way hash chain method. Ad Hoc Networks. **11**(8) (2013) 2526-2540
19. Odelu, V., Das, A. K., Goswami, A.: A secure effective key management scheme for dynamic access control in a large leaf class hierarchy. Information Sciences. 269 (2014) 270-285
20. Mao, Q., Chang, C. C., Harn, L., Chang, S. C.: An image-based key agreement protocol using the morphing technique. Multimedia Tools and Applications. 74 (2013) 32073229
21. Islam, S. H.: Design and analysis of a three party password-based authenticated key exchange protocol using extended chaotic maps. Information Sciences. 312 (2015) 104-130
22. Li, B., Yang, Y., Lu, Z., Yuan, B., Long, T.: Secure distributed batch rekeying algorithm for dynamic group. In Communication Technology (ICCT), 2012 IEEE 14th International Conference on. (2012) 664-667
23. Rogaway, P., Shrimpton, T.: Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Fast Software Encryption. Springer Berlin Heidelberg. (2004) 371-388
24. Zhang, J., Li, B., Chen, C.X., Tao, P., Yang, S.Q.: EDKAS: A Efficient Distributed Key Agreement Scheme Using One Way Function Trees for Dynamic Collaborative Groups. IEEE IMACS Multiconference on Computational Engineering in Systems Applications. 2 (2006) 1215-1222
25. Li, M., Feng, Z., Zang, N., Graham, R. L., Yao, F. F.: Approximately optimal trees for group key management with batch updates. Theoretical Computer Science. **410**(11) (2009) 1013-1021