ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

New Multiparty Authentication Services and Key Agreement Protocols

D. Jancy Rani[1], P. Sabarinathan[2]
M.E, Department of CSE, Pavendar Bharathidasan College of Engineering and Technology, Trichy, TamilNadu, India[1]
Assistant Professor, Department of CSE, Pavendar Bharathidasan College of Engineering and Technology, Trichy, TamilNadu, India[2]

ABSTRACT - Key management is equally important as compared to any other security measure such as encryption and authentication. With the growing usage of mobile devices and the advent of multicast communication, there has been a significant amount of work carried out in developing an optimum group key management protocol for mobile multicast systems. The paper presents a comprehensive survey of group key management protocols in wireless mobile environments that employ multicast communication. The existing system, Slot based Multiple Group Key Management Protocol supports the multiple group services; it can be reducing rekeying transmission overheads. The Domain Key Distributor and Area Key Distributor to providing intense security in terms of communication bandwidth, storage overhead. The proposed system, network dependent and independent protocols and further categorized into cluster-based key management protocols. Identity based encryption and decryption is used for secured transmission of data. At the destination end decryption can be done to view the transmitted information.

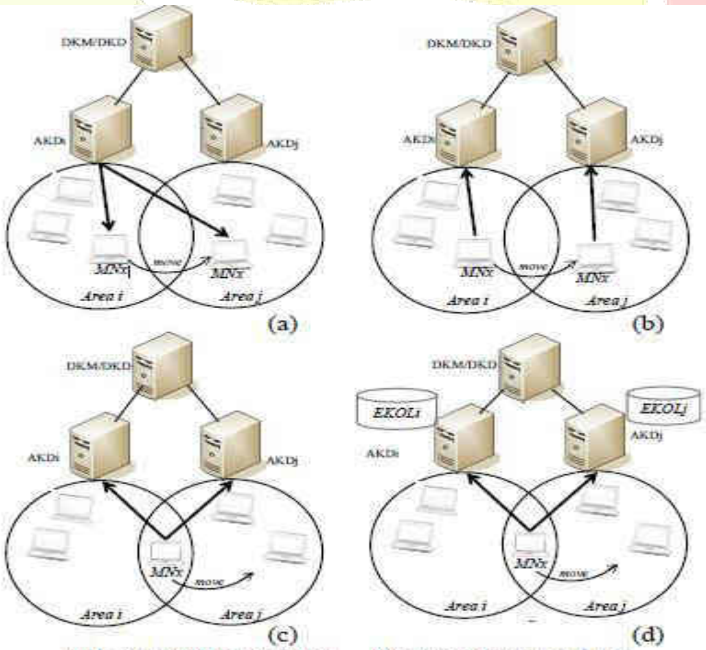
KEYWORDS - Group key management, security, multicast service, wireless network

I. INTRODUTION

The performance of SMGKM scheme is analyzed through numerical analysis and simulations in terms of rekeying transmission overhead corresponding to the additional signaling load caused by rekeying, storage overhead corresponding to the storage capacity of the key management keys stored by the entities (Mi, AKDi and DKD). The communication overheads for both rekeying approaches (pair wise and LKH) as a result of unicast or multicast transmissions of rekeying messages at the cluster level are also considered. Finally the security analyses section considers all types impossible attacks in SMGKM. To solve the rekeying complexity as multicast services cumulate in a single network, proposed system, network dependent and independent protocols and further categorized into cluster-based key management protocols. Identity based encryption and decryption is used for secured transmission of data. At the destination end decryption can be done to view the transmitted information. SMGKM integrate our concept of session key distribution list (SKDL) introduced in for fast and secure authenticated handover along with initial key establishment. SMGKM employ a lighter symmetric encryption suitable for resource constraint mobile devices than heavier asymmetric effort. Compared to the existing schemes, SMKGM save enormous communication bandwidth utilization in the presence of multi-handoffs in multi-services.

Cluster-based Mobile **Key Management** Scheme (CMKMS): the cluster-based mobile multicast group key management protocols for wireless networks which are as follows: Micro-Grouped Iolus Scheme: The authors of proposed an improved version of the scheme proposed in which supports member mobility known as micro-grouped Iolus (M-Iolus) which further divides subgroups into micro-groups. M-Iolus adopts a decentralized approach similar to Iolus with independent TEK per subgroup. The network entities involved include the central Group Security Controller (GSC) which manages all the trusted Group Security Intermediaries (GSI) linked to it and the GSI which manages the key management of members within its subgroup. The protocol reduces overhead by introducing the concept of time-stamp association in each sub-group and micro-group key update maintained by each GSI for its group. When MNx moves form micro-group 1 to micro group 2 under the same GSI, GSI detects the area that the MNx has moved in and then the moving member notifies the GSI by sending a move request.

DeCleene et al: The authors of and proposed a hierarchy framework and key distribution algorithms for dynamic environment. The authors focus on how keys and trust relationships are transferred when members move across areas in the hierarchy. Furthermore, the make comparable study of rekeying algorithms involved every time a member moves from area to area.



AKDi – Area Key Distributor of area i AKDj – Area Key Distributor of area j
DKM/DKD – Domain Key Manager or Domain Key Distributor MNx – Mobile Node

It relies on the central server known as Domain Key Distributor (DKD) at the domain level for Key generation, key updating and key distribution. Each area is managed by their controllers known as the Area Key Distributors (AKD) which operate under the jurisdiction of the DKD. AKDs distribute the group key to members under its area securely by encrypting it with the arealocal key held by each AKD. The main security keys involved are the group key (data key) held by the DKD which encrypts the actual multicast traffic before is it sent and the Area local key held by the AKDs to securely send the encrypted traffic to its members. Figure illustrates the several rekeying algorithms proposed to minimize the need of rekeying in decentralized framework.

II. RELATED WORK

The most common transport layer protocol to use multicast addressing is User Datagram Protocol (UDP). By its nature, UDP is not reliable messages may be lost or delivered out of order. By adding loss detection and retransmission mechanisms, reliable multicast has been implemented on top of UDP or IP by various middle ware products, e.g. those that implement the Real-Time Publish-Subscribe (RTPS) Protocol of the Object Management Group(OMG) Data Distribution Service (DDS) standard, as well as by special transport protocols such as Pragmatic General Multicast (PGM). IP multicast is widely deployed in enterprises, commercial stock exchanges, and multimedia content delivery networks. A common enterprise use of IP multicast is for IPTV applications such as distance learning and televised company meetings.

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a**cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters). Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It will often be necessary to modify data preprocessing and model parameters until the result achieves the desired properties. Besides the term clustering, there are a number of terms with similar meanings, including automatic classification, numerical taxonomy, botryology (from Greek βότρυς"grape") and typological analysis. The subtle differences are often in the usage of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification the resulting discriminative power is of interest. This often leads to misunderstandings between researchers coming from the fields of data mining and machine learning, since they use the same terms and often the same algorithms, but have different goals.

The notion of a "cluster" cannot be precisely defined, which is one of the reasons why there are so many clustering algorithms. There is a common denominator: a group of data objects. However, different researchers employ different cluster models, and for each of these cluster models again different algorithms can be given. The notion of a cluster, as found by different algorithms, varies significantly in its properties. Understanding these "cluster models" is key to understanding the differences between the various algorithms. Typical cluster models include:

Connectivity models: for example, hierarchical clustering builds models based on distance connectivity.

Centroid models: for example, the k-means algorithm represents each cluster by a single mean vector.

Distribution models: clusters are modeled using statistical distributions, such as multivariate normal distributions used by the Expectation-maximization algorithm.
Subspace models: in Biclustering (also known as Co-clustering or two-mode-clustering), clusters are modeled with both cluster members and relevant attributes.
Group models: some algorithms do not provide a refined model for their results and just provide the grouping information.
Graph-based models: a clique, that is, a subset of nodes in a graph such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster. Relaxations of the complete connectivity requirement (a fraction of the edges can be missing) are known as quasi-cliques, as in the HCS clustering algorithm.
A "clustering" is essentially a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example, a hierarchy of clusters embedded in each other. Clustering does can be roughly distinguished as:
Hard clustering: each object belongs to a cluster or not
Soft clustering (also: fuzzy clustering): each object belongs to each cluster to a certain degree (for example, a likelihood of belonging to the cluster)
There are also finer distinctions possible, for example:
Strict partitioning clustering: here each object belongs to exactly one cluster
Strict partitioning clustering with outliers: objects can also belong to no cluster, and are considered outliers.
Overlapping clustering (also: alternative clustering, multi-view clustering): while usually a hard clustering, objects may belong to more than one cluster.
Hierarchical clustering: objects that belong to a child cluster also belong to the parent cluster Subspace clustering: while an overlapping clustering, expected to overlap, within a uniquely defined subspace, clusters are not.

III. PROPOSED ALGORITHM

Identity Based Encryption: An Identity Base Encryption (IBE) scheme is a public-key cryptosystem where any string is a valid public key. In particular, email addresses and dates can be public keys. The IBE email system is based on the first practical Identity-Based Encryption scheme (IBE).
The cryptosystem has chosen cipher text security in the random oracle model assuming an elliptic curve variant of the computational Diffie-Hellman problem.
The IBE email system has some nice properties such as:

- Senders can send mail to recipients, who have not yet setup a public key, When sending email there is no need for an online lookup to obtain the recipient's certificate,
- Senders can send email that can only be read at some specified time in the future, and
- The system proactively refreshes the recipient's private key every short time period.

Applications for Identity-Based Encryption: The original motivation for identity-based encryption is to help the deployment of a public key infrastructure. More generally, IBE can simplify systems that manage a large number of public keys. Rather than storing a big database of public keys the system can either derive these public keys from usernames, or simply use the integers {1,…n} as distinct public keys. We discuss several specific applications below.

Revocation of Public Keys: Public key certificates contain a preset expiration date. In an IBE system key expiration can be done by having Alice encrypt e-mail sent to Bob using the public key: bob@hotmail.com current-year. In doing so Bob can use his private key during the current year only. Once a year Bob needs to obtain a new private key from the PKG. Hence, we get the effect of annual private key expiration. Note that unlike the existing PKI, Alice does not need to obtain a new certificate from Bob every time Bob refreshes his certificate.

This forces Bob to obtain a new private key every day. This might be feasible in a corporate PKI where the PKG is maintained by the corporation. With this approach key revocation is quite simple: when Bob leaves the company and his key needs to be revoked, the corporate PKG is instructed to stop issuing private keys for Bob's e-mail address. The interesting property is that Alice does not need to communicate with any third party to obtain Bob's daily public key. This approach enables Alice to send messages into the future: Bob will only be able to decrypt the e-mail on the date specified by Alice.

Delegation of Decryption Keys: Another application for IBE systems is delegation of decryption capabilities. We give two example applications. In both applications the user Bob plays the role of the PKG. Bob runs the setup algorithm to generate his own IBE system parameters params and his own master-key. Here we view params as Bob's public key. Bob obtains a certificate from a CA for his public key params. When Alice wishes to send mail to Bob she first obtains Bob's public key params from Bob's public key certificate. Note that Bob is the only one who knows his master-key and hence there is no key-escrow with this setup. Delegation to a laptop. Suppose Alice encrypts mail to Bob using the current date as the IBE encryption key (she uses Bob's params as the IBE system parameters). Since Bob has the master-key he can extract the private key corresponding to this IBE encryption key and then decrypt the message. Now, suppose Bob goes on a trip for seven days. Normally, Bob would put his private key on his laptop. If the laptop is stolen the private key is compromised. When using the IBE system Bob could simply install on his laptop the seven private keys corresponding to the seven days of the trip. If the laptop is stolen, only the private keys for those seven days are compromised. The master-key is unharmed. Delegation to a duty. Suppose Alice encrypts mail to Bob using the subject line as the IBE encryption key. Bob can decrypt mail using his master-key. Now, suppose Bob has several assistants each responsible for a different task (e.g. one is `purchasing', another is `human-resources', etc.). Bob gives one private key to each of his assistants corresponding to the assistant's responsibility. Each assistant can then decrypt messages whose subject line falls within its responsibilities, but it cannot decrypt messages intended for other assistants. Note that Alice only obtains a single public key from Bob (params) and she uses that public key to send mail with any subject line of her choice. The mail can only be read by the assistant responsible for that subject.

IV. PSEUDO CODE

The private key generator (PKG) chooses:

The public groups G_1 (with generator P) and G_2 as stated above, with the size of q depending on security parameter k,

The corresponding pairing e,

A random private master-key $K_m = s \in \mathbb{Z}_q^*$,
A public key $K_{pub} = sP$,
A public hash function $H_1 : \{0,1\}^* \to G_1^*$,
A public hash function $H_2 : G_2 \to \{0,1\}^n$ for some fixed n and The message space and the cipher space
$\mathcal{M} = \{0,1\}^n$, $\mathcal{C} = G_1^* \times \{0,1\}^n$

Extraction:

To create the public key for $ID \in \{0,1\}^*$**, the PKG computes**
$Q_{ID} = H_1(ID)$ and
the private key $d_{ID} = sQ_{ID}$ which is given to the user.
Encryption: Compute $g_{ID} = e(Q_{ID}, K_{pub}) \in G_2$ and
Given $m \in \mathcal{M}$**, the cipher text** c **is obtained as follows:**
$Q_{ID} = H_1(ID) \in G_1^*$, Choose random $r \in \mathbb{Z}_q^*$,
Compute $g_{ID} = e(Q_{ID}, K_{pub}) \in G_2$ and
Set $c = (rP, m \oplus H_2(g_{ID}^r))$.
Note that K_{pub} is the PKG's public key and thus independent of the recipient's ID.

Decryption:

Given $c = (u, v) \in \mathcal{C}$, the plaintext can be retrieved using the private key:
$m = v \oplus H_2(e(d_{ID}, u))$

V. SYSTEM IMPLEMENTAION

SMGKM scheme are analyzed the numerical analysis and simulation in terms of rekeying transmission overhead, rekeying communication overhead, storage overhead, bandwidth consumption in SMGKM, security analysis. The Rekey signal messages are delivered into the DKD and AKD in 'w' unit, and also delivered to the MN and the AKD$_s$ be α unit respectively. SMGKM using pair wise and LKH rekeying approaches for induced the communication overhead; it also compared the cluster level into conventional approaches. The protocol also lacks trust relationship due to data transformations which can expose the data to eavesdropping. Multiple members moving will be a performance hurdle to the previous GSI which has to deal with multiple authentication requests. If the previous GSI fails, members moving will face service disruptions. If multiple members who had moved between several GSIs leave the group, this will trigger rekeying in all the affected areas hence adding more control overheads which wastes bandwidth.

Set-Up:

Input: desired security level.
Output: PP and msk for the PKG.

Key Generation:

Input: identity ID, PP and msk.

Output:
dID, the secret key for ID.
Encryption:
Input: identity ID, msg
M, PP.
Output: ciphertext
C.
Decryption:
Input: ID, C, dID.
Output:
M or bad.

The service provider is to all the files are selected and are uploaded into the server**.** Maintaining an efficient key management system is challenging due to group membership. In multicast services, members not only dynamically join or leave the services as addressed in single service scenario. Multiple multicast groups will co-exist within the same network due to the emergence of various group based applications and computationally fast mobile devices along with increased data rates for next generation wireless networks. During group generation cluster formation of subscribers are developed and for each subscriber within the cluster a separate IP address is generated. The received file will be in encrypted format. To decrypt the file corresponding area key and domain key must be given.

VI. CONCLUSION

In contrast to convectional schemes targeted for a single service, SMGKM used a new rekeying strategy based on lightweight KUS and SKDL for effectively performing key management and authentication phases respectively during handoff. SMGKM adopted independent TEK per cluster to localize rekeying and mitigate one-affect-n phenomenon. Numerical analysis and simulation results of the SMGKM performed much better using both rekeying approaches in comparison to previous work. Finally, the analytical study was explored by simulation for solving the bandwidth optimization problem in SMGKM which showed efficiency in bandwidth consumption in the presence of multi-services. However, SMGKM is expected to become a practical dynamic solution for securely and efficiently managing multi-services which can be received concurrently by huge mobile subscriber's in the future wireless networks such as emerging Software Defined Networks. They are classified into network dependent and independent protocols and further categorized into cluster-based key management protocols. Identity based encryption and decryption is used for secured transmission of data. At the destination end decryption can be done to view the transmitted information. To safeguard the contents inside the file encryption and decryption is used in this paper. While transmitting a file it must be encrypted for security. At the destination node only the appropriate receiver can decrypt the information. This encryption and decryption is based on Identity based encryption system.

REFERENCES

[1] G. S. V. R. K. Rao and G. Radhamani, WiMax: A Wireless Technology Revolution. Boca Raton, FL, USA: Auerbach Publishers, 2008.

[2] 3GPP, "Multimedia Broadcast/Multicast Service; Stage 1 (Release 8)," Technical Specification 3GPP TS 22.146, vol. 8.3.0, (2007-06), Jun. 2007.

[3] 3GPP, "Digital cellular telecommunications system (Phase 2þ); Universal Mobile Telecommunications System (UMTS);LTE; Multimedia Broadcast/Multicast Service (MBMS); Stage 1 (Release 9)," Technical Specification 3GPP TS 22.146, vol. 9.0.0, (2010-01), 2010.

[4] P. Judge and M. Ammar, "Security issues and solutions in multicast content distribution: A survey," IEEE Netw., vol. 17, no. 1, pp. 30–36, Jan./Feb. 2003.

[5] W. Chung Kei, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," IEEE/ACM Trans. Netw., vol. 8, no. 1, pp. 16–30, Feb. 2000.

[6] S. Rafaeli and D. Hutchison, " A survey of key management for secure group communication," ACM Comput. Surveys, vol. 35, pp. 309–329, Sept. 2003.

[7] T. T. Mapoka, "Group key management protocols for secure mobile multicast communication: A comprehensive survey," Int. J. Comput. Appl., vol. 84, pp. 28–38, Dec. 2013.

[8] Paul Judge and Ammar.M, (2003), "Security issues and solutions in multicast content distribution: A Survey", IEEE Netw., vol. 17, no. 1, pp. 30-36.

[9] Suganya Devi.D, and Padmavathi.G, (2014), "A Reliable secure multicast key distribution scheme for mobile adhoc networks", in Proc. 3rd Int. Conf. Future Generation Commun. Technol., pp. 66-71.

[10] Sandro Rafaeli and Hutchison.D, (2003), "A Survey of key management for secure group communication", ACM Comput. Surveys, vol. 35, pp. 309-329.

[11] SuvoMittra.S, (1997), "Iolus: A framework for scalable secure multicasting", SIGCOMM Comput. Commun. Rev., vol.27, pp.277-288.

[12] Tshepo Mapoka.T, (2013), "Group key management protocols for secure mobile multicast communication: A comprehensive survey," Int. J. Comput. Appl., vol. 84, pp. 28-38.

[13] Yacine Challal and Hamida Seba, (2005), "Group key management protocols: A novel taxonomy," Int. J. Inf. Technol., vol. 2, pp. 105-119.