View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by UNL | Libraries

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

CSE Conference and Workshop Papers

Computer Science and Engineering, Department of

2005

CRTDH: An Efficient Key Agreement Scheme for Secure Group Communications in Wireless Ad Hoc Networks

Ravi K Balachandran
University of Nebraska-Lincoln

Byrav Ramamurthy
University of Nebraska-Lincoln, bramamurthy2@unl.edu

Xukai Zou
Purdue University School of Science

N.V. Vinodchandran
University of Nebraska-Lincoln, vvariyam2@unl.edu

Follow this and additional works at: https://digitalcommons.unl.edu/cseconfwork

 Part of the Computer Sciences Commons

Balachandran, Ravi K; Ramamurthy, Byrav; Zou, Xukai; and Vinodchandran, N.V., "CRTDH: An Efficient Key Agreement Scheme for Secure Group Communications in Wireless Ad Hoc Networks" (2005). *CSE Conference and Workshop Papers*. 66.
https://digitalcommons.unl.edu/cseconfwork/66

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

2005 IEEE International Conference on Communications, 2005. ICC 2005.
Volume: 2
doi: 0.1109/ICC.2005.1494522

CRTDH: An Efficient Key Agreement Scheme for Secure Group Communications in Wireless Ad Hoc Networks

Ravi K Balachandran†, Byrav Ramamurthy†, Xukai Zou‡ and N.V.Vinodchandran†
†Dept. of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68503
Email: {ravib, byrav, vinod}@cse.unl.edu
‡Dept. of Computer and Information Science
Purdue University School of Science
Indianapolis, IN 46202
Email: xkzou@cs.iupui.edu

***Abstract*-As a result of the growing popularity of wireless networks, in particular ad hoc networks, security over such networks has become very important. In this paper, we study the problem of secure group communications (SGC) and key management over ad hoc networks. We identify the key features of any SGC protocol for such networks. We also propose an efficient key agreement scheme for SGC. The scheme solves two important problems that exist in most current SGC schemes: requirement of member serialization and existence of a central entity. Besides this, the protocol also has many highly desirable properties such as contributory and efficient computation of group key, uniform work load for all the members, few rounds of rekeying (2 rounds for the initial key formation and join and 1 round for leave), and efficient support for high dynamics. These properties make the protocol well suited for wireless ad hoc networks.**

I. Introduction

Wireless networks, in particular IEEE 802.11 networks, have revolutionized the field of data networking with applications in numerous fields such as business, home and military. Security of information over such networks are of paramount importance. In wired networks, security services such as authentication, key management and authorization are generally provided by a trusted central authority. In an ad hoc environment since the services of such a central authority are not usually available, the members have to provide such services themselves.

Trust establishment, key management and authorization are important areas that need to be thoroughly researched before security in wireless ad hoc networks becomes a reality. In this paper, we study the problem of *key management and secure group communications* over ad hoc networks. We consider a scenario where a number of mobile nodes form an ad hoc network with no prior knowledge or information.

Secure group communication (SGC) is defined as the process by which members in a group can securely communicate with each other and the information being shared is inaccessible to anybody outside the group. In such a scenario, a group key is established among all the participating members and this key is used to encrypt all the messages destined to the group. A good SGC protocol should efficiently manage the group key when members join and leave; this is especially true in ad hoc networks where the members are highly mobile and the network topology is dynamic. Recently, a number of protocols have been proposed to handle SGC over wireless networks but none of these protocols efficiently handle the unique problems posed by ad hoc networks. In this paper, we analyze such ad hoc network features which are relevant for SGC applications and propose a new protocol that is suited for ad hoc networks.

The new protocol has many desirable features with regard to ad hoc networks. It does not require member serialization or structure, supports a high level of user dynamics (member(s) join/leave), assumes no pre-shared information or the presence of a trusted authority. Moreover, computation is equally distributed among all the members and it is efficient in communication. In any SGC scheme, recomputing the group key after member leave operations is a difficult problem. The proposed scheme handles this situation very efficiently since it requires only 1 round of broadcast to recompute the group key after a member leave operation.

The rest of this paper first discusses the desired SGC properties over ad hoc networks in Section 2. Section 3 describes the related work on SGC schemes for wireless networks. Section 4 describes the new protocols for SGC over ad hoc networks and comparison with other schemes is presented in Section 5. Section 6 presents our conclusions.

0-7803-8938-7/05/$20.00 (C) 2005 IEEE

II. Desired features of an SGC protocol over ad hoc networks

Before formulating a scheme for secure group communications over ad hoc networks, the desired properties of any SGC scheme over ad hoc networks need to be clearly identified.

Avoidance of Member Serialization - A number of SGC schemes require group member serialization or sequencing. In such protocols, information is sent from one node to another in a pre-defined sequence in order to create the group key. In ad hoc networks with high node mobility, such serialization is not efficient since the sequence may not correspond to the best geographic node placement and may lead to increased communication cost.

Contributory Key Agreement - This is defined as a key establishment protocol whose secret key is a function of information contributed by all the participants in the group, so that no member can predetermine the value of the key. It is a method for negotiating a key value without actually transferring the keys, even in encrypted form. The best example for key agreement is the Diffie-Hellman (DH) key exchange protocol [3]. Since the existence of either a centralized trusted authority (TA), group controller (GC) or a pre-shared secret among all the mobile nodes is not assumed, the SGC scheme should be a key agreement protocol. Also, using a contributory protocol ensures that all the group members in the ad hoc network play an equal role in the computation of the group key instead of a few nodes doing the bulk of the work. This results in uniform energy consumption at all nodes, which is significant in wireless ad hoc nodes with limited power budget.

Efficiency - Any scheme for ad hoc networks should be efficient in both computation and communication. Mobile nodes are typically computation and memory constrained devices with limited battery power.

Good user dynamics - This means that the SGC scheme should be able to support member join/leave operations efficiently. This is a very important feature in ad hoc networks due to its highly dynamic topology and user mobility.

III. Related work

Recently a number of protocols have been proposed to solve the problem of key management over wireless ad hoc networks. Key pre-distribution has been discussed in [11, 2, 1]. Zhu et al. in [11] discuss a probabilistic key sharing scheme in which an offline key server is used to initialize all the nodes. A password based multi-party key agreement scheme was proposed in [1] where all the nodes are assumed to share a password. Basagni et al. in [2] describe a secure ad hoc network in which all the nodes share a group identification key stored in tamper-resistant devices. Though all the above schemes perform efficiently, they require that all the nodes have some pre-determined knowledge. In ad hoc networks where mobile nodes do not have the privilege of knowing other group members beforehand, assumption of such a pre-shared secret is invalid.

The concept of mobile certificate authorities has been discussed in [10, 6]. In such schemes, the responsibilities of a CA is distributed among a set of wireless nodes. A subset (threshold) of such CAs must be contacted to obtain a valid certificate. Such schemes have several advantages such as providing data integrity, authentication and non-repudiation. The drawbacks of such schemes are: (a) identifying nodes that perform the role of a CA, consequently these nodes must spend more power; (b) constant availability of a threshold of CAs in a mobile network; and (c) the use of the computationally expensive Public Key (PKI) encryption systems.

Public key certificates are also used in [5], where all the nodes are assumed to maintain a local certificate repository and a probabilistic method is used to achieve a certificate chain between two nodes. This scheme requires that all the nodes are preloaded with a set of certificates and it is possible that two nodes in the ad hoc network do not achieve a certificate chain. Also, the authors in [10, 6, 5] do not address the distinct features of SGC such as group key formation and member join/leave.

Key establishment using contributory key agreement protocols are discussed in [4]. Anton et al. in [4] discuss a number of such protocols previously used on wired networks and conclude that the CLIQUES [8] protocol suite is best suited for ad hoc networks. Li et al. in [7] also use the GDH.2 (Group Diffie Hellman) protocol, part of the CLIQUES protocol suite, for key agreement over ad hoc networks. GDH is an efficient protocol with good support for member join and leave operations but it has some unfavorable features with regard to ad hoc networks. Most importantly, the GDH scheme requires that the members be serialized or structured in order to compute the group. Also, the last member in the group acts as a Group Controller (GC). Consequently the GC does more computation than the other members in the group. Thus, in using GDH for ad hoc networks deciding which member is going to perform the operation of a GC is an important problem.

Finally, Yasinsac at al. in [9] present a key agreement protocol using the Diffie-Hellman key exchange concept. The main advantage of this protocol is that it does not involve member serialization. On the downside, the protocol does not efficiently support member join/leave operations and the protocol also involves the services of a GC.

IV. CRTDH: a key agreement scheme for ad hoc networks

In this section, we will discuss the details of our proposed contributory key agreement protocol, Chinese Remainder Theorem and Diffie-Hellman (CRTDH) based scheme for secure group communications. The different steps of the key establishment process and the join/leave operations are discussed in detail.

A. Key Agreement

In order to establish the group key, each member U_i[1], where $i = 1, \ldots, n$ should execute the following steps

- Step 1: Select the Diffie-Hellman (DH) private share x_i and compute the public share $y_i = g^{x_i} \, mod \, p$. ($g$ and p are the generator and the prime modulo used in the Diffie-Hellman computation. This information is public and if the nodes do not share this, then an initial broadcast round is needed.)
- Step 2: Broadcast the DH public share y_i to all the members in the group.
- Step 3: Receive the DH public share of all the other members in the group and compute the DH key shared with each of them

$$m_{ij} \quad = \quad y_j^{x_i} \, mod \, p$$

  where $j = 1, \ldots, i-1, i+1, \ldots, n$ and $j \neq i$
- Step 4: Find the Least Common Multiple (LCM) of all the DH keys calculated in Step 3 as lcm_i.
- Step 5: Select a random k_i, such that $k_i < \min(m_{ij}, \forall j)$, which will be its share of the group key. Also select an arbitrary number D such that $D \neq k_i$ and another number D_p such that the $gcd(D_p, lcm_i) = 1$.
- Step 6: Solve the CRT

$$\begin{aligned} crt_i &\equiv k_i \bmod lcm_i \\ crt_i &\equiv D \bmod D_p \end{aligned}$$

  and broadcast it to the group.
- Step 7: Receive the crt values from all the other members in the group and calculate

$$k_j \quad = \quad crt_j \bmod m_{ij}$$

  for all $j \neq i$ and compute the group key

$$GK \quad = \quad k_1 \oplus k_2 \oplus \ldots \oplus k_n$$

As can be seen from the above steps, the Chinese Remainder Theorem is used to send each member's key share (disguised) to all the other members in the group. The Diffie-Hellman key exchange is performed to derive the modulo value in the CRT calculation.

To understand the details of the scheme, let us consider a member U_1 in a group of 4 members. The first two steps of the protocol involve the generation and distribution of the DH public share by each member in the group. U_1 selects a DH private share x_1 and computes its DH public share $y_1 = g^{x_1} \bmod p$. U_1 then broadcasts the DH public share y_1 to all the other members in the group.

In Step 3 of the protocol, all the m_{ij} values are generated, which are nothing but the DH keys shared between U_1 and the other members. U_1 calculates three m values m_{12}, m_{13}, m_{14} which are equal to $y_2^{x_1}, y_3^{x_1}, y_4^{x_1}$ respectively. y_2, y_3, y_4 are the DH public shares of members U_2, U_3, U_4 broadcasted in Step 2. The three DH keys (m_{12}, m_{13}, m_{14}) generated by U_1 are equal to m_{21}, m_{31}, m_{41} generated by U_2, U_3, U_4 respectively. U_1 then calculates the LCM of the DH keys m_{12}, m_{13} and m_{14}. This LCM value will be later used for the CRT calculation in Step 6.

Step 5 of the protocol involves the generation of a random key share k_1 by U_1. This k_1 share has to be less than all DH keys m_1, m_2 and m_3 and the lcm_1 value since we want the other members to obtain k_1 and not $k_1 \, (\bmod m_{ij})$ or $k_1 \, (\bmod lcm_1)$ respectively. In the next step, U_1 generates an arbitrary number D and D_p which will be used in solving the CRT. The D_p value should be selected such that D_p and lcm_i are co-primes, in order to solve the CRT. Also, the number D should not be equal to k_1, since if they are, then the solution to the CRT will be equal to the group key, k_1.

After solving the CRT in Step 6, the solution is broadcasted to the group in Step 7. U_1 solves the CRT to obtain crt_1 and broadcasts it to the the group. U_1 also receives the CRT values crt_2, crt_3, crt_4 from the other members in the group. U_1 can obtain k_2, k_3, k_4 by performing the following operations.

$$\begin{aligned} k_2 &= crt_2 \, (\bmod m_{12}) \\ k_3 &= crt_3 \, (\bmod m_{13}) \\ k_4 &= crt_4 \, (\bmod m_{14}) \end{aligned}$$

The individual k_i shares are then XOR-ed to obtain the group key GK.

Similarly all the members in the group arrive at the same group key, since the following holds

$$k_j \equiv crt_j \bmod LCM_j \equiv crt_j \bmod m_{ij}$$

Any member, such as U_i, receives the (broadcast) values crt_1 from $U_1, \ldots$, crt_{i-1} from U_{i-1}, crt_{i+1} from U_{i+1}, $\ldots$, and crt_n from U_n. U_i can then compute $k_1, \ldots, k_{i-1}$, $k_{i+1}, \ldots$ and k_n using $m_{(i,1)}, \ldots, m_{(i,i-1)}$, $m_{(i,i+1)}$, $\ldots$ and $m_{(i,n)}$ respectively. Along with its own k_i, U_i has all the elements for computing the group key. As a result, all the members will compute the same key.

B. Join Operation

The operations to be performed when a new member joins a group are explained below. Let us assume the member U_5 wishes to join an existing group of four members $\{U_1, U_2, U_3, U_4\}$.

[1]The notation is only for naming purposes and does not represent any order/serialization of members.

- Step 1: All the current members (U_1, U_2, U_3, U_4) should compute the hash of the current key GK i.e. $h(GK)$. One of the existing (closest) member should transmit this hash value $h(GK)$ and all the DH public shares y_1, y_2, y_3, y_4 to the new member U_5.

- Step 2: U_5 will execute the steps given in the previous section and broadcast the CRT value crt_5 along with its public DH share y_5.

- Step 3: Existing members can compute the DH key they share with U_5 and thereby calculate the k_5 key share selected by U_5. The new group key GK_{new} is computed by XORing the hash of the current key and the key share of the newly joining member U_5

$$GK_{new} \quad = \quad h(GK) \oplus k_5$$

It is obvious from the above steps that only the newly joining member does the bulk of the work. The existing members only do minimal work in receiving the new key share and XORing with the hash of the old group key. This is a desired feature in ad hoc networks since there are frequent group membership changes due to node mobility. The hash of the old key is sent to the joining member since it should receive the shares of the existing members but also not be able to read the messages sent to the group previously.

In case of multiple joins, all the joining members should execute the above steps to contribute their share towards the group key. The existing members then XOR all key shares from the newly joining members to get the new group key. This makes multiple joins very efficient since existing members only perform XOR operations with all the contributed key shares. Also, the join operation (single/multiple) involves only two rounds of communication.

C. Leave Operation

The leave operation is similar to the join operation but consists of only one round. Let us assume U_2 is going to leave the group. Then the following operations need to be performed to recompute the group key.

- Step 1: Any one of the remaining members, say U_1, should redo the key agreement steps in Section 4.1 from Step 4, but this time U_2 should be left out of the computation. Member U_1 should select a new key share k_1 and not include the DH key it shares with U_2 in the LCM and CRT computations.

- Step 2: The other members receive the crt_1 value from U_1 and calculate the new k_1 value. The new group key GK_{new} is computed as follows

$$GK_{new} \quad = \quad GK \oplus k_1$$

It should be noted that, when a member leaves the group, one of the existing members does the major portion of the work. During implementation, suitable methods should be used that distribute this responsibility to other existing members when there are frequent leave operations.

In case of multiple leaves, all the leaving members should be left out of the computation as shown above. No extra computation is needed since the protocol need not be repeated for each leaving member. Thus the CRTDH protocol efficiently supports leave operations and more importantly multiple leave operations in a single round of computation.

V. Discussion

The protocol described in the previous section meets the requirements specified in Section 2. The protocol does not assume any pre-shared secret between the members and does not require the services of a trusted authority or a group controller. The Diffie-Hellman key exchange and the Chinese Remainder Theorem are not very computationally intensive. The use of Elliptic curves for the Diffie-Hellman key exchange will make the scheme more efficient. Communication wise the scheme involves only two rounds for initial key agreement and join operations and only one round for leave operations.

More importantly for ad hoc networks, serialization or ordering of group members and communication is not required for the proper execution of the protocol. Every node in the ad hoc network is treated equally and has to perform the same amount of work to compute the group key. It also efficiently supports single/multiple user join/leave operations, which is an important factor in highly dynamic environments such as ad hoc networks.

Regarding the security of the scheme, we do not give a formal proof security in this paper but analyze the means by which the scheme can be broken. In order for an attacker to obtain the group key, knowledge of all individual key shares, k_i selected by each member is necessary. The key share k_i can be obtained by using either: (i) any of the DH shared key m_{ij} computed by a member U_i or (ii) the LCM_i, which in turn depends upon the DH shared keys. Both these methods depend upon breaking the Diffie-Hellman key exchange method. The Diffie-Hellman key exchange is a well studied problem in literature and depends upon the Discrete Logarithmic Problem (DLP). So far, no efficient algorithms have been designed to break the Diffie-Hellman key exchange.

A comparison with other proposed key management schemes for ad hoc networks is given in Table 1 [2] . The pre-shared schemes [11, 2, 1] are left out of the comparison since they use a totally different approach than the

[2] A number of schemes have been proposed in the literature that deal with security in ad hoc networks, but we only consider those schemes that deal with secure group communications.

Table 1: Comparison of Key Management schemes

Protocol	CLIQUES(GDH.2) [8]	Mobile CA [10, 6]	Yasinsac [9]	CRTDH
Rounds	n	2	2	2
Total messages	n	n-1	n+1	$2n$
No Pre-Shared secret	√	√	√	√
No GC/CA	×	×	×	√
Uniform Work load	×	×	×	√
No Serialization	×	√	√	√
Key Agreement	√	×	×	√
High Dynamics	×	×	×	√

key establishment process in the other schemes and are not applicable in 'truly' ad hoc networks. The Mobile CA approaches in [10, 6] do not deal with group key generation as such, but they can be easily extended to do so. Any member in the group can act as the Group Controller (GC) and generate the group key, which is later sent to all the other members by encrypting it with each member's public key. This requires a minimum of two rounds for selecting the GC and distributing the group key.

As can be seen from the table all the schemes do not assume the existence of a pre-shared secret among the members. On the other hand, except for the proposed CRTDH scheme all the other protocols need the services of a Group Controller to distribute the key. Since the major work is done by the GC, there is no uniform distribution of work load among the members and the selection of a GC is also an important issue.

With regard to the serialization of members, only the GDH.2 protocol requires this feature. In this scheme information is sent from one node to another in a serial fashion requiring $n-1$ rounds and one last broadcast round. The Mobile CA and the Yasinsac schemes do not require serialization since they are not key agreement schemes by definition. The two schemes use encryption algorithms in order to send the group key from the GC to the other members in the group. Hence, information need not be passed in any order, only the encrypted group key is sent from the GC to all the other members.

Also, due to the use of encryption, these schemes do not efficiently support user join/leave operations. When a member joins or leaves the group, $n-1$ encryptions of the new group key need to be performed. The GDH.2 protocol efficiently supports single join/leave operations but not a high level of dynamics. The proposed CRTDH scheme on the other hand is a key agreement protocol without member serialization and with efficient support for user join/leave operations.

VI. Conclusion

In this paper, we studied the problem of secure group communications (SGC) and key management over wireless ad hoc networks. After an analysis of the properties of SGC and ad hoc networks, we have identified the ideal features of an SGC scheme over ad hoc networks. Also, we have proposed an efficient contributory key agreement protocol, CRTDH, which does not require member serialization. When compared to other key management schemes for ad hoc networks, the proposed scheme has several favorable features that make it well suited for such networks. As future work, we intend to work on a formal proof of security and implement CRTDH and related schemes.

Acknowledgment

We thank Dr. Wandi Wei of Florida Atlantic University for several useful discussions regarding this work.

References

[1] N. Asokan and P. Ginzboorg. Key agreement in ad hoc networks. In *Computer Comm.*, volume 23, pages 1627–1637, 2000.

[2] S. Basagni, K. Herrin, E. Rosti, and D. Bruschi. Secure pebblenets. In *ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2001.

[3] W. Diffie and M. Hellman. New directions in cryptography. In *IEEE Transactions on Information Theory*, 1976.

[4] E.Anton and O.Duarte. Group key establishment in wireless ad hoc networks. In *Workshop on Quality of Service and Mobility*, 2002.

[5] J.P. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *ACM MobiHoc*, 2001.

[6] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *IEEE ICNP*, 2001.

[7] X.Y. Li, Y. Wang, and O. Frieder. Efficient hybrid key agreement protocol for wireless ad-hoc networks. In *IEEE International Conference on Computer Communications and Networks*, 2002.

[8] M. Steiner, G. Tsudik, and M. Waidner. Cliques: A new approach to group key agreement. In *International Conference on Distributed Computing Systems*, 1998.

[9] A. Yasinsac, V. Thakur, S. Carter, and I. Cubukcu. A family of protocols for group key generation in ad hoc networks. In *IASTED Conference on Communication and Computer Networks*, 2002.

[10] S. Yi and R. Kraverts. Key management in heterogeneous ad hoc wireless networks. In *IEEE ICNP*, 2002.

[11] S. Zhu, S. Xu, S. Setia, and S. Jajodia. Establishing pair-wise keys for secure communication in ad-hoc networks: A probabilistic approach. In *IEEE International Conference on Network Protocols*, 2003.