WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2008; 8:1297–1312
Published online 3 December 2007 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.575

An efficient and attack-resistant key agreement scheme for secure group communications in mobile ad-hoc networks

Ravi K. Balachandran[1], Xukai Zou[2*,†], Byrav Ramamurthy[3], Amandeep Thukral[2] and Vinodchandran N. Variyam[3]

[1]*Microsoft India Pvt Ltd, Hyderabad, India*

[2]*Department of Computer Science, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202, U.S.A.*

[3]*Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588, U.S.A.*

Summary

As a result of the growing popularity of wireless networks, in particular mobile ad hoc networks (MANET), security over such networks has become very important. Trust establishment, key management, authentication, and authorization are important areas that need to be thoroughly researched before security in MANETs becomes a reality. This work studies the problem of secure group communications (SGCs) and key management over MANETs. It identifies the key features of any SGC scheme over such networks. AUTH-CRTDH, an efficient key agreement scheme with authentication capability for SGC over MANETs, is proposed. Compared to the existing schemes, the proposed scheme has many desirable features such as contributory and efficient computation of group key, uniform work load for all members, few rounds of rekeying, efficient support for user dynamics, key agreement without member serialization and defense against the Man-in-the-Middle attack, and the Least Common Multiple (LCM) attack. These properties make the proposed scheme well suited for MANETs. The implementation results show that the proposed scheme is computationally efficient and scales well to a large number of mobile users. Copyright © 2007 John Wiley & Sons, Ltd.

KEY WORDS: wireless security; mobile ad-hoc networks (MANET); secure group communications; key management; Chinese Remainder Theorem; Diffie–Hellman key agreement

1. Introduction

Wireless networks, in particular mobile ad hoc networks (MANETs), have revolutionized the field of data networking with applications in numerous fields. MANETs can be used in all those situations where there is no time or resources available to setup a backbone network or infrastructure. With their increasing usage, information security over such networks becomes vital.

In wired networks, security services such as authentication, key management, and authorization are generally provided by a trusted central authority. In MANETs, since the services of such a central authority are not usually available, the members have to provide

*Correspondence to: Xukai Zou, Department of Computer Science, Indiana University-Purdue University, Indianapolis, IN 46202, U.S.A.

†E-mail: xkzou@cs.iupui.edu

such services themselves. Trust establishment, key management, and authorization are important areas that need to be thoroughly researched before security in MANETs becomes a reality. In this paper, we study the problem of authenticated key management and secure group communications over MANETs.

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 MANETs where the members are highly mobile and the network topology is dynamic. SGCs and key management in MANETs are closely related since the mobile nodes in the ad hoc environment automatically form a group. A number of protocols have been proposed to handle SGC over wired networks [1–6]. Considerable research has also been done on key management over infrastructure-based wireless sensor networks usually comprising of hundreds of small sensor nodes [7–14]. On the other hand, only a few schemes [15–17] have been proposed to handle SGC over MANETs and none of these protocols efficiently handle the unique problems posed by MANETs.

In this work, we consider the scenario of a 'true' MANET wherein, the participating members do not share any secret beforehand and wish to form a network for further communication. Earlier, we proposed an efficient group key management scheme for MANETs based on Chinese Remainder Theorem and Diffie–Hellman (DH) key exchange, called CRTDH [18]. In this paper, we extend the CRTDH scheme to include mutual authentication during the key agreement process, called AUTH-CRTDH. The mutual authentication among group members helps defend against the Man-in-the-Middle attack. In addition, AUTH-CRTDH solves the Least Common Multiple (LCM) problem and the small k_i problem existing in the CRTDH protocol. We also include the simulation and experiment results and comparisons of our scheme with the typical existing schemes. AUTH-CRTDH has many desirable features with regard to MANETs. AUTH-CRTDH does not require member serialization or structure, supports a high level of user dynamics (member(s) join/leave), and does not require the involvement of a central authority in key agreement. Moreover, computation is equally distributed among all the members and it is efficient in communication.

The rest of the paper is organized as follows. The desired SGC properties over MANETs are presented in Section 2. Section 3 describes the related work on SGC schemes for wireless networks. Section 4 describes the CRTDH protocol for SGC over MANETs. Section 5 introduces AUTH-CRTDH and discusses its defense capabilities. Section 6 presents our implementation results. The conclusions are given in Section 7.

2. Desired SGC Features Over MANETs

In this section, we discuss the properties of a secure group communication scheme, features of a MANET, and then the desired features of any SGC scheme operating over MANETs. In order to enable efficient SGC, any SGC scheme has to satisfy the following properties: key establishment to achieve a shared group key, member join with backward secrecy enforcement, member leave with forward secrecy enforcement, bursty operation allowing multiple users to join and/or leave simultaneously, and efficiency with the minimum amount of computation and communication.

Besides the above features, MANETs have some interesting features that need to be addressed. Here we identify the important MANET features with respect to SGC: (1) no Pre-shared secret—'Pre-shared secret' means that the group members have a common group-wide shared secret value in advance. This secret value is private and only known to the group members but not known to the public[‡]. In MANETs, since the participating members are not known beforehand, it is not possible to assume that the members share some secret information before forming the group; (2) no centralized Trusted Authority (TA) or Group Controller (GC)—in many wireless networks such as wireless LAN, the TA/GC is provided by a base station. In MANETs, the assumption of a central authority is invalid since the network is formed by all the participating nodes themselves without any infrastructure; (3) battery power—This is the primary concern for a mobile node. Any application or service should be power-optimized to run in an ad hoc environment since ad hoc nodes are energy starved

[‡]Note that many SGC group key management schemes, including our proposed CRTDH/AUTH-CRTDH scheme, assume there are some public parameters such as the prime p and the generator g and public information such as users' IDs and public keys. Such public information is pre-knowledge (i.e., is made known to all in advance) for schemes to work but not pre-shared secret information.

DOI: 10.1002/wcm

devices. Every time a wireless node transmits or receives information it has to expend some battery power. An adversary can launch a different type of attack which drains the nodes batteries [19]. Hence, an algorithm for SGC should be efficient in both computation and communication; (4) equal work load—all mobile nodes are in equal position and have equal capability; (5) mobility—The nodes in MANETs are highly mobile. A mobile node may fall in and out of range from any of the other nodes and thus from the MANET. With respect to an SGC protocol, a high level of mobility means that any SGC scheme should efficiently support user join and leave operations.

Due to the above features of MANETS, an ideal SGC scheme for MANETS should have its unique properties including the following: (1) avoidance of Member Serialization—'Member serialization' means that the group members are organized in certain order and numbered in a pre-defined (even cyclic) sequence. In order to create the group key, the messages are sent from one member to another in the pre-defined sequence. Or if multicast/broadcast is used, the keying multicast contains information about intended receiving members' IDs or sequence numbers so that the receiving members can identify/extract their corresponding portions of the key materials to perform correct execution of the protocol[§]. In MANETs 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. Also, in such protocols there is the additional overhead of serializing the members and operations; (2) 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 pre-determine the value of the key. The best example for key agreement is the DH key exchange protocol [20]. Since the existence of either a centralized TA, 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 MANET 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; (3) efficiency—as previously described, any scheme for MANETS should be efficient in both computation and communication since mobile nodes are typically computation and memory constrained devices with limited battery power; (4) 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 MANETs due to its highly dynamic topology and user mobility.

3. Related Work

In the last section, we concluded that an SGC scheme over MANETs should be a contributory key agreement scheme with efficient support for user join and leave. In this section, we first discuss relevant contributory key agreement schemes proposed in literature. Next, this section also deals with some earlier protocols proposed to support SGC over MANETs. The suitability of these schemes for MANETs is also discussed.

3.1. Contributory Key Agreement Schemes

Steiner *et al.* have proposed an elegant extension to DH exchange for dynamic peer groups in References [5,21,22] called Group DH (GDH) (consisting of GDH.1, GDH.2, and GDH.3). GDH.2 is a contributory scheme which has very good support for user join/leave. GDH.2 is probably a good scheme for wired networks but it has some properties that are not suited for MANETs. Most importantly, GDH.2 requires that the members be serialized or structured in a particular order and information be sent from one node to another in a serial fashion. This is not a desired feature on MANETs as discussed earlier. Also, as information is sent from one node to another in the sequence, the message size and the computation done by the nodes keep increasing. Additionally, the last node that receives the information has to act as the GC. There

[§] A number of SGC key management schemes require group member serialization or sequencing. For example, in GDH.2, strict member serialization is required. m_1 first sends rekeying message to m_2, then m_2 to m_3, $\cdots$, until m_{n-1} to m_n in sequence, and finally m_n sends rekeying materials back to m_1, m_2, $\cdots$, m_{n-1}. This last step can be done by unicasting each rekeying material rm_i to m_i or broadcasting $< rm_1, rm_2, \cdots, rm_{n-1} >$ to all. The broadcast message indicates every portion of rekeying materials clearly so that each member m_i can get its rekeying material rm_i from its corresponding ith portion. In contrast, in the newly proposed CRTDH/AUTH-CRTDH scheme, all k_is are mixed/combined into one CRT value (an integer). There is no need for ordering members and the CRT integer contains no information about which part of it is a specific k_i. A receiving member of the CRT integer can directly compute the needed k_i by a simple integer division.

DOI: 10.1002/wcm

does not exist a sense of 'fairness' among the nodes since some nodes do much more work than the others, particularly those higher up in the sequence do more work than others lower in the sequence. Thus, in using GDH.2 for MANETs deciding the structure or sequence order of the nodes and deciding the node which performs the operation of a GC is an important problem since the position of a node in the sequence corresponds to the amount of work it must perform.

Another extension of DH to groups was proposed by Steer *et al.* in Reference [23]. This protocol has several favorable properties: it has no GC, has a constant message size, and the member join operation is efficient. The drawbacks of this scheme are that the scheme involves serialization of the member nodes similar to the previous GDH.2 scheme. Also, the workload is not shared equally among all the members. Most importantly for mobile networks, the leave operation is highly inefficient with the complexity depending upon the member position in the serial order.

Kim *et al.* in Reference [24] modified the Steer protocol by incorporating a broadcast round, known as STR. The number of rounds was reduced from $n-1$ to 2 and member serialization was not required but this meant that one of the members in the group had to perform the role of a GC. Consequently, the GC did more work than the others and the message size was not constant anymore.

The initial attempt to extend DH to group communications was done by Ingemarsson *et al.* Reference in [25]. The protocol executes in $n-1$ rounds and requires that all the members be arranged in a logical ring. The positive properties of it are that there is no GC, every member does equal work, and the message size is constant. However, the protocol suffers from communication overhead, inefficient join/leave operations, and the requirement for a 'group structure'.

Another elegant protocol for key agreement was proposed in Reference [26] by Burmester and Desmedt. The protocol involves two broadcast rounds before the members agree on a group key. This scheme has several advantages such as the absence of a GC, equal work load for key establishment (different for join/leave), and a small constant message size. Some of the drawbacks of this scheme are that it requires the members to be serialized and the join/leave operation is not very efficient.

Tree based schemes such as LKH [27] and LKH+, LKH++ [28] have been used to generate group keys for secure communication. The LKH schemes are scalable and efficient in computation but they use a centralized approach for key establishment. Instead of these schemes, distributed tree based approaches such as TGDH [2] and DISEC [29] are preferred for MANETs. TGDH and DISEC use the DH principles and hash functions to achieve the group key respectively. They are scalable and require a few rounds (usually $O(lg(n))$) for key computation. Their major drawback is that they require a group structure and member serialization for group formation. The effect of node mobility, member join/leave on the tree structure, and key agreement complexity need to be clearly studied before these schemes can be efficiently used in MANETs. In Section 6, the comparison of the above schemes (except tree based schemes) with CRTDH is given in Table I.

3.2. Related SGC Protocols for MANETs

Apart from the contributory key agreement protocols discussed above, there have been a few schemes proposed to solve the problem of secure group communications over MANETs. At this point, it is important to understand the difference between SGC schemes and other protocols for providing security and pairwise keys between mobile nodes in a MANET. Active research has been done in the area of providing security services in a MANET [12,28,30-36]. Though all these protocols provide solutions for secure communication

Table I. Comparison of contributory key agreement schemes.

Protocol	ING [25]	BD [26]	GDH [22]	Steer's [23]	STR [24]	CRTDH
Rounds	$n-1$	2	n	$n-1$	2	2
Total messages	$n(n-1)$	$2n$	n	$2(n-1)$	$2(n-1)$	$2n$
Messages sent per m_i	$n-1$	2	1	2, 1 for m_0, m_{n-1}	1, n for m_0	2
Messages received per m_i	$n-1$	$n+1$	2, 1 for m_0, m_{n-1}	$n-i+1$	$n-i+1$	$2n-2$
Exponentiat-ions per m_i	n	$n+1$	$i+2$	$n-i+1$	$n-i+1, m_0$: $2(n-1)$	$5n-1$ for AUTH-CRTDH n for CRTDH
No GC	Y	Y	N	Y	N	Y
No member serialization	N	N	N	N	Y	Y
User dynamics	N	N	Y	N	Y	Y
Uniform work load	Y	N	N	N	N	Y

DOI: 10.1002/wcm

between pairs of nodes, they do not address the problem of SGC, which is the focus of this work.

Some of the schemes [37–39] proposed for SGC over MANETs assume the existence of a pre-shared secret among the group members. Basagni *et al.* described the concept of secure pebblenets in Reference [37]. Here, all the nodes in the network share a secret group identity key beforehand. Further, this key is stored in tamper-resistant devices that protect the key from attackers who may capture the mobile nodes. The paper also discussed various protocols for cluster formation. The proposed scheme is suitable in the case, where all the mobile nodes in the MANET are known beforehand. Furthermore, the overhead of such cluster formation protocols should be considered in highly dynamic MANETs.

A password-based multi-party key agreement scheme was proposed in Reference [38]. Here, all the participating members are assumed to share a weak password. The paper further discussed a scheme which derives a strong shared key starting from the already shared weak password. Pietro *et al.* in Reference [39] discussed a group key management scheme based on the logical key hierarchy (LKH). This scheme also assumes the existence of a shared secret between each member and the GC.

Li *et al.* proposed a hybrid key agreement protocol in Reference [16] which is based on GDH.2. The paper also discussed protocols for forming subgroups among ad hoc nodes using the dominating set concept. Though this scheme efficiently supports group formation and user join/leave, it suffers from the same drawbacks as GDH.2 such as the need for member serialization and a GC.

A tree based extension of the DH scheme for key agreement over MANETs was proposed in Reference [40]. In this scheme a spanning tree is constructed among the participating nodes before the key agreement phase. The key agreement method described is similar to the Tree based Group Diffie–Hellman (TGDH) scheme [2] though with some differences. Compared to the TGDH scheme, this approach is more centralized and the intermediate nodes are not virtual nodes but the members of the group. The study in Reference [40] does not discuss the aspects of node mobility or user join/leave and its effect on the key generation and tree formation. Its drawback, other than requiring member serialization, is that the root node (initiator of the group) of the tree also performs the role of a GC in order to set up the group.

A group key generation protocol was proposed in Reference [17] by Yasinsac *et al.* This protocol is contributory since it uses the concept of a combining function to combine the information sent by all the members. This is not a key agreement protocol since the key is sent encrypted from the GC to all the other members. Its major advantage is that there is no serialization. The scheme on the other hand suffers from drawbacks such as poor support for user join/leave and GC bottleneck.

4. The Proposed CRTDH Scheme

In this section, we will discuss the details of our Chinese Remainder Theorem based DH contributory key agreement protocol (CRTDH).

4.1. CRTDH Key Establishment

In order to establish the group key, each member U_i[‖] should execute the following steps:

- **Step 1.** Select the DH private share x_i and compute the public share $y_i = g^{x_i} \bmod p$. (g and p are the generator and the prime modulus and are made public.)
- **Step 2.** Broadcast the DH public share y_i to all the members in the group.
- **Step 3.** Receive the DH public shares and compute the DH key shared with each of them

$$m_{ij} = y_j^{x_i} \bmod p \text{ where}$$
$$j = 1, \ldots, i-1, i+1, \ldots, n \text{ and } j \neq i \quad (1)$$

  Note: since there is no verification on the authenticity of y_i, the Man-in-the-Middle attack may exist in the above steps. This is pointed out in Section 5 and the enhanced scheme AUTH-CRTDH will solve this problem.
- **Step 4.** Find the LCM of all the DH keys calculated above 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 $\gcd(D_p, \text{lcm}_i) = 1$.
- **Step 6.** Solve the CRT and broadcast crt_i to the group.

$$\text{crt}_i \equiv k_i \bmod \text{lcm}_i$$
$$\text{crt}_i \equiv D \bmod D_p \quad (2)$$

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

DOI: 10.1002/wcm

- **Step 7.** Receive the crt values from all the other members in the group and calculate

$$k_j = \text{crt}_j \bmod m_{ij} \tag{3}$$

for all $j \neq i$ (note: k_j has been selected to be less than m_{ij}) and compute the group key

$$\text{GK} = k_1 \oplus k_2 \oplus \cdots \oplus k_n \tag{4}$$

As can be seen from the above steps, each member contributes a share in the formation of the group key. The Chinese Remainder Theorem is used to send each member's key share (disguised) to all the other members in the group. The DH key exchange is performed to derive the modulus value in the CRT calculation.

To understand the details of the scheme, let us consider a member U_1 in a group of four 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. Note: $k_1 < \text{lcm}_1$ and $k_1 < m_{1j}$. In the next step, U_1 generates an arbitrary number D and D_p such that D_p and lcm_i are co-primes.

After solving the CRT in Step 6, the solution is broadcast to the group in Step 7. U_1 solves the CRT to obtain crt_1 and broadcasts it to 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.

$$\begin{aligned} k_2 &= \text{crt}_2 \,(\text{mod}\, m_{12}) \\ k_3 &= \text{crt}_3 \,(\text{mod}\, m_{13}) \\ k_4 &= \text{crt}_4 \,(\text{mod}\, m_{14}) \end{aligned} \tag{5}$$

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

Similarly all the members in the group calculate the same group key. 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. 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.

4.2. 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 $\{U_1, U_2, U_3, U_4\}$.

- **Step 1.** All the current members (U_1, U_2, U_3, U_4) should compute the hash of the current key GK, that is, $h(\text{GK})$. One of the existing (closest) member should transmit this hash value $h(\text{GK})$ 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 subsection 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. Note: h is a public cryptographic hash function.

$$\text{GK}_{\text{new}} = h(\text{GK}) \oplus k_5 \tag{6}$$

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 MANETs 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

DOI: 10.1002/wcm

operations with all the contributed key shares. Also, the join operation (single/multiple) involves only two rounds of communication: one unicast message and one broadcast message.

4.3. 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:

- Any one of the remaining members, say U_1, should redo the key agreement steps in Subsection 4.1 from Step 4. In Step 4, the LCM value is computed from the DH keys that a member shares with the other members in the group. For the leave operation, the DH key that U_1 shares with U_2 should be left out of this LCM computation. Then, U_1 selects a new key share k_1 and computes the CRT, which is broadcast to the group.
- 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

$$\text{GK}_{\text{new}} = \text{GK} \oplus k_1 \tag{7}$$

It should be noted that, when a member leaves the group, one of the existing members does the major portion of the work, that is, the LCM and CRT computation[1]. 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 LCM 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.

4.4. Correctness of the Scheme

It is not very difficult to show that all the members arrive at the same group key. The group key is computed from the key shares k_i of each member U_i. The key share k_i is in turn calculated using the crt_i values broadcasted by each member. Every member calculates the crt value using the LCM of all the DH keys it shares with the other members. Since the following holds:

$$k_i \equiv \text{crt}_i \bmod \text{LCM}(m_{ij}) \equiv \text{crt}_i \bmod m_{ji}$$
$$(k_i < m_{ij} = m_{ji}) \text{for all } j = 1, \ldots, i-1, i+1, \ldots, n. \tag{8}$$

Each member can successfully compute key shares and thus arrive at the group key.

4.5. Security of the Scheme

In this section we analyze CRTDH's security. As it will be pointed out in the next section, CRTDH suffers from the Man-in-the-Middle attack and the LCM attacks. However, CRTDH is secure against the attack trying to crack the group key under the common believe (assumption) that the underlying DLP (or DH Problem) problem and Chinese Remainder Theorem are intractable.

Theorem 4.1. *The CRTDH scheme is secure in protecting the group key from being uncovered.*

Proof. Since the group key $\text{GK} = k_1 \oplus k_2 \oplus \cdots \oplus k_n$, if an attacker wants to uncover GK, the attacker must first find out individual key share k_is contributed by every member m_is. k_i is contained in crt_i which m_i computes *via* Equation (2) and broadcasts in the second round. First, except the conditions of $D \neq k_i$ and $\gcd(\text{lcm}_i, D_p) = 1$ (of course, D should be less than D_p), there is no other constraint on choosing D and D_p. In particular, which kinds of values can be selected as D and D_p or how many bits D and D_p should have has no impact on the security of Equation (2). Due to the security of Chinese Remainder Theorem [41], k_i cannot be obtained by an attacker from crt_i unless the attacker obtained at least one of the following values: m_{ij}s and lcm_i. Note: obtaining D_p or D offers no help to an attacker. The following three sets of congruences justify the above statements.

$$\begin{array}{c|c|c} 13 = 3 \bmod 5 & 13 = 6 \bmod 7 & 13 = 1 \bmod 3 \\ 13 = 1 \bmod 3 & 13 = 3 \bmod 5 & 13 = 6 \bmod 7 \end{array} \tag{9}$$

Suppose $k_i = 3$ and $\text{lcm}_i = 5$. Which of (3, 1) and (7, 6) is selected as (D_p, D) does not matter. An attacker cannot figure out which one is correct even though both generate the same $\text{crt}_i = 13$ (if they generate different crt_i, it will be more difficult for an attacker to figure it out). On the other hand, even if we assume that an attacker figures out $D_p = 3$ and $D = 1$ somehow,

[1] In Section 6, implementation results have shown that this computation is efficient.

DOI: 10.1002/wcm

for example, by guessing, the attacker still cannot get k_i since k_i can be 3 (with $\text{lcm}_i = 5$) or equally be 6 (with $\text{lcm}_i = 7$).

Secondly, as for any m_{ij}, it has been computed using the DH key exchange protocol which is further based on the DLP problem. Since DLP is intractable [41], the attacker cannot figure out any m_{ij}. Thirdly, as for lcm_i, it is computed by finding the LCM of DH keys that member U_i shares with the other members in the group. This value is private and is known only to the member U_i. For anybody else to calculate this lcm_i value, knowledge of the DH keys that the member U_i shares with every other member in the group is required. Thus, computing lcm_i again depends on breaking DH key exchange, which is intractable. As a result, CRTDH is able to protect the group key from unauthorized access. □

5. AUTH-CRTDH: Authenticated Key Agreement

In this section, we describe AUTH-CRTDH by extending CRTDH with mutual authentication among users [42]. The users utilize the services of a central key generation center (KGC) for obtaining a secret corresponding to the ID. This central entity is different than a GC (present in many schemes) as the services of the KGC are required only at system setup and it does not participate in the key agreement procedure. The operations performed at the KGC can be thought of as *offline* operations that need to be performed prior to the formation of any ad hoc environment. However, these operations ensure that each node in the MANET can authenticate itself to any other node in the network. Thus, the Man-in-the Middle attack will be avoided.

There exists another kind of attack, even very low probability, associated with CRTDH. We call it the *LCM attack*. The attack is possible due to the fact that the LCM for any given set of numbers is not unique to the given set. In other words, there could be many more numbers that could possibly be added to the set and still result in the same LCM value. This could cause problems in the member join and member leave operations as discussed below.

Problem with member join: Assume that there exists a group of four members, $\{U_1, U_2, U_3, U_4\}$ who share the group key GK and a user U_5 wishes to join the group. The problem arises in the step where an existing user computes the DH key that it shares with the newly joined member and then proceeds to compute the LCM for the set of values. There could be a case where the addition of the shared DH key does not affect the LCM and hence the LCM value remains the same as before. For example, assume that user U_4 computes the following after receiving the public share of U_5.

$$\{U_4\} \rightarrow m_{41} = 6, m_{42} = 4, m_{43} = 8, m_{45} = 12 \tag{10}$$

As can be observed, $\text{lcm}_4(= 24)$ remains unchanged upon the addition of m_{45}. Hence user U_5 could obtain the shared secret k_4, if it could capture previous messages sent by user U_4. Similarly, it is possible that the values for all other lcms remain unchanged after U_5 joins, thus making it possible for U_5 to obtain all the previous key shares. This way U_5 can compute the previous group key.

Problem with member leave: When an existing member of the group decides to leave, say U_i, the rekeying operation is performed in order to maintain forward secrecy. The problem arises once again due to the fact that there may be cases where the new LCM value for a user may still cover the DH key value that it shared with the departing member. In such a case, the departing member would still be able to decrypt the message that the user broadcasts in order to distribute its new key share.

One more issue associated with CRTDH is that k_i must be less than every m_{ij} for $j \neq i$. m_{ij} can be small so k_i can be small too. This can be a serious security hole. We solve the problem as follows: taking m_{ij} as it was if it is larger than the half of the modulus and the modulus $-m_{ij}$, otherwise. In summary, AUTH-CRTDH is described as follows.

5.1. Offline System Setup

The users in the system are identified with a unique identity (ID). The scheme is analogous to the RSA public key cryptosystem, with a value of $e = 3$. The system setup procedure carried out at the KGC is described below.

- **Step 1.** Generate two large prime numbers p_1 and p_2, and let $M = p_1 \cdot p_2$.
- **Step 2.** Select the center's secret key d from the computation.

$$3 \cdot d(\text{mod}\,(p_1 - 1) \cdot (p_2 - 1)) = 1 \tag{11}$$

- **Step 3.** Select an integer g that is a primitive element in Z_M^*.
- **Step 4.** Select a one-way hash function h which would be used to compute the extended identity

DOI: 10.1002/wcm

(EID_i) of user U_i as follows. h is made public.

$$\text{EID}_i = h(\text{ID}_i) \tag{12}$$

- **Step 5.** Generate the user secret key S_i as

$$S_i = \text{EID}_i^d (\bmod M) \tag{13}$$

As a result of the above relation, the following equation holds.

$$\text{EID}_i = S_i^3 (\bmod M) \tag{14}$$

When a user U_i registers with the system, he sends his ID_i to the KGC, which performs Steps 4 and 5 mentioned above. The KGC sends (M, g, h, S_i) to U_i. U_i keeps S_i secret and stores the public information (M, g, h). In addition, the ID of each user is publicly known.

5.2. Key Agreement

In order to establish the group key for a group with m members, each member U_i should execute the following steps, where $i = 1, 2, \ldots, n$.

- **Step 1.** Select the DH private share x_i, and compute the following values for the first broadcast.

$$\begin{aligned} A_i &= S_i \cdot g^{2x_i} \pmod M \\ B_i &= g^{3x_i} \pmod M \end{aligned} \tag{15}$$

- **Step 2.** Broadcast A_i and B_i to all members of the group.
- **Step 3.** Receive the public shares A_j and B_j from other members in the group and authenticate the users. Each member U_i calculates $\text{EID}_j = h(\text{ID}_j)$ and checks the validity of the member U_j's broadcast message through the following equation.

$$\text{EID}_j = \frac{A_j^3}{B_j^2} \tag{16}$$

If the equation holds true, then the user computes the DH shared secret with each of the members as follows:

$$m_{ij} = \begin{cases} B_j^{x_i} \bmod M & \text{if } B_j^{x_i} \bmod M > \frac{M}{2} \\ M - B_j^{x_i} \bmod M & \text{otherwise} \end{cases} \tag{17}$$

Otherwise, U_j's authentication fails and action would be initiated to remove U_j from the group.

- **Step 4.** Find the LCM of all the DH keys calculated above as lcm_i.
- **Step 5.** Select a random share for the group key k_i, such that $k_i < \min\{m_{ij}, \forall j\}$. Also select an arbitrary number D such that $D \neq k_i$ and another number D_p such that $\gcd(D_p, lcm_i) = 1$ (similar to the original CRTDH scheme).
- **Step 6.** Solve the CRT: (as in the original CRTDH scheme)

$$\begin{aligned} \text{crt}_i &= k_i \bmod \text{lcm}_i \\ \text{crt}_i &= D \bmod D_p \end{aligned} \tag{18}$$

For authentication purposes, the user also computes the following:

$$\begin{aligned} X_i &= h(k_i) \cdot g^{2D} \cdot S_i \pmod n \\ Y_i &= g^{3D} \pmod n \\ Z_i &= \{A_i \| X_i\}_{k_i} \end{aligned} \tag{19}$$

and broadcasts $\{X_i, Y_i, Z_i, crt_i\}$ to the group.

- **Step 7.** Receive the CRT values from all the other members in the group and calculate the following: (similar to the CRTDH scheme)

$$k_j = \text{crt}_j \bmod m_{ij} \tag{20}$$

for all $j \neq i$. To validate the authenticity of the key k_j, the user also computes EID_j and verifies the following equation

$$(h(k_j))^3 = \frac{X_j^3}{(Y_j^2 \cdot \text{EID}_j)} \tag{21}$$

In addition, the user computes $\{A_j \| X_j\}_{k_j}$ and then verifies:

$$Z_j = \{A_j \| X_j\}_{k_j} \tag{22}$$

The purpose of this verification is authenticating A_i and X_i to defeat the following attack: an attacker selects a random r and replaces A_i with A_i' and B_i with B_i' as follows: $A_i' = A_i r^2$ and $B_i' = B_i r^3$, thus, $(A_i')^3/(B_i')^2 = (A_i^3/B_i^2)$. Similarly, this is true for X_i and Y_i.

After both of the above verifications succeed, the user then computes the group key (as in the CRTDH

DOI: 10.1002/wcm

scheme)

$$GK = k_1 \oplus k_2 \oplus \cdots \oplus k_n \tag{23}$$

Thus the Chinese Remainder Theorem is used to send the secret key share of each user to all other members in the group. The mutual authentication is provided by using an ID based scheme (See Reference [43] for the principle of ID based cryptography).

With AUTH-CRTDH, the member join operation needs to be modified to make sure that the new lcm is different from the old one. If they are same, the new joining member is asked to reselect its new secret share and broadcast its public share again. Similarly, for the member leaving operation, it is needed to check whether the new lcm covers the DH value formed with the leaving user. If covered, a new secret share is selected and new lcm is computed.

5.3. Security of AUTH-CRTDH

Similar to the security analysis of CRTDH, we prove AUTH-CRTDH's security below.

Theorem 5.1. *AUTH-CRTDH is secure in protecting the group key from cracking and in defending against the Man-in-the-Middle and LCM attacks.*

Proof. (1) AUTH-CRTDH uses the same mechanism to agree upon the group key as CRTDH, thus, by Theorem 4.1, AUTH-CRTDH is also secure in protecting the group key from cracking. (2) Based on the principle of ID based cryptography [43] and by adding some additional steps and messages, the Man-in-the-Middle attack can be prevented by AUTH-CRTDH. Equation (16) verifies the identity of the message sender and Equation (21) performs verification again. These double verifications will defeat the Man-in-the-Middle attack. In addition, A_i is included in Z_i (See Equation (19)) and verified in Equation (22), thus, discovering any potential attack which modified A_i to $A_i r^2$ and B_i to $B_i r^3$ and passed the verification in Equation (16). (3). The modified joining and leaving operations nullify all LCM attacks since whenever the new *lcm* is the same as the old one, a new DH secret share will be selected and the new DH public share is broadcast for generating a new lcm. □

6. Results and Discussion

The CRTDH protocol 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 TA or a group controller. The DH key exchange and the Chinese Remainder Theorem are not very computationally intensive. The use of elliptic curves for the DH key exchange will make the scheme more efficient. Communication-wise the scheme involves only two rounds for the initial key agreement and join operation and only one round for the leave operation.

More importantly for MANETs, serialization or ordering of group members and communication is not required for the proper execution of the protocol. Every node in the MANET 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 MANETs.

The enhanced AUTH-CTRDH protocol has all the above features except that there is need for a user to go to an offline central entity to get its personal secret corresponding to its public ID initially. This offline central entity does not participate in any key agreement process. Most importantly, AUTH-CRTDH is able to verify users and to prevent illegal users from joining the communication group, in particular, defending against the Man-in-the-Middle attack and the LCM attack.

6.1. Comparison with Other Schemes

A comparison with other contributory key agreement schemes is presented in Table I.

As can be seen from the Table I, the CRTDH protocol involves two broadcast rounds. This is similar to the BD protocol, but it does not require member serialization whereas the BD protocol does. It also supports user dynamics better than the BD protocol. In BD, the key establishment phase ensures uniform work load among all the members but during join/leave this is not true. Hence, the BD protocol is indicated as not having uniform work load in the table. GDH.2 (referred to as GDH in the table) performs well in many regards but its major drawback is that it requires member (as well as message) serialization, the presence of a GC and a non-uniform work load.

The Steer protocol performs well during key establishment and member join but member leave is inefficient. The STR protocol, which is an extension of

DOI: 10.1002/wcm

the Steer protocol, supports efficient user dynamics and key establishment but it uses the services of a GC and thus does not ensure uniform work load among all the nodes. The ING protocol has no GC and has uniform work load for all members but here again, join/leave support is not good.

A comparison with other proposed key management schemes for MANETs is given in Table II. The Mobile CA approaches in References [32,33] 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 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: one for broadcasting each users public key and the second round for the GC to distribute 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 GC 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 Li's 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. These two schemes use encryption algorithms in order to send the group key from the GC to the members in the group. Hence information need not be passed in any particular order, only the encrypted group key is sent from the GC to all the 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.

6.2. Implementation Results

Different SGC protocols were implemented in order to find out the computation time for the protocols. The implementation was done in C++ using the Crypto++ library [44] for cryptographic functions. All tests were carried out on an AMD Athlon 900 MHz machine with 256 MB RAM running Linux Redhat 9. It should be noted that only the computation times were calculated and compared in the following graphs. The communication times for the different protocols were not taken into account in the following graphs.

Graphs comparing the computation time of a single member for key establishment, member join, and leave are presented in this section. Other graphs discussing the overhead on existing members when a member joins or leaves the group are also detailed. The following SGC protocols were implemented: CRTDH, RSA, GDH.2, ING, BD, and STR. RSA here means the Mobile CA approach discussed in the previous section, where a GC sends n encryptions of the group key to n members. For GDH.2 and STR, the computation times for the GC are calculated.

First, the computation times of a single member for key establishment using the different protocols are given in Figure 1 (left). Group sizes from 10 to 100 were considered for the tests, the time is given in milliseconds. A group key size of 128 bits was chosen since this provides a good balance between security and performance. As can be seen from Figure 1 (left), the CRTDH computation time increases linearly and depends on group sizes. The ING and STR computation times are almost equal and the computation time for RSA does not vary much since the encryptions of a 128 bit number is not very computationally intensive. The CRTDH scheme performs better than the ING, STR, and RSA schemes and is only slightly more expensive than the GDH.2 scheme. But the BD scheme is the most computationally efficient among all the schemes for key establishment.

The computation time for the join operation is shown in Figure 1 (right). The join operation times are similar to the key establishment times in Figure 1 (left), since the newly joining member essentially has to perform the same steps as in key establishment. The CRTDH computation time for the newly joining member is slightly less than the key establishment time since the newly joining member need not perform Step 7 (modular operation) of the key establishment process mentioned in Subsection 4.2. It is important to note that though the BD protocol performs well computationally during the join operation, it involves two rounds of communication whereas the CRTDH protocol involves only one round. Also, in the BD protocol in addition to the joining member, other existing members need to perform considerable amount of computation whereas in the CRTDH protocol, the computation performed by the existing members is minimal. The computation

DOI: 10.1002/wcm

Table II. Comparison of SGC schemes over MANETs.

Protocol	Li [5]	Mobile CA [32, 33]	Yasinsac [17]	(AUTH-)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	×	×	×	√

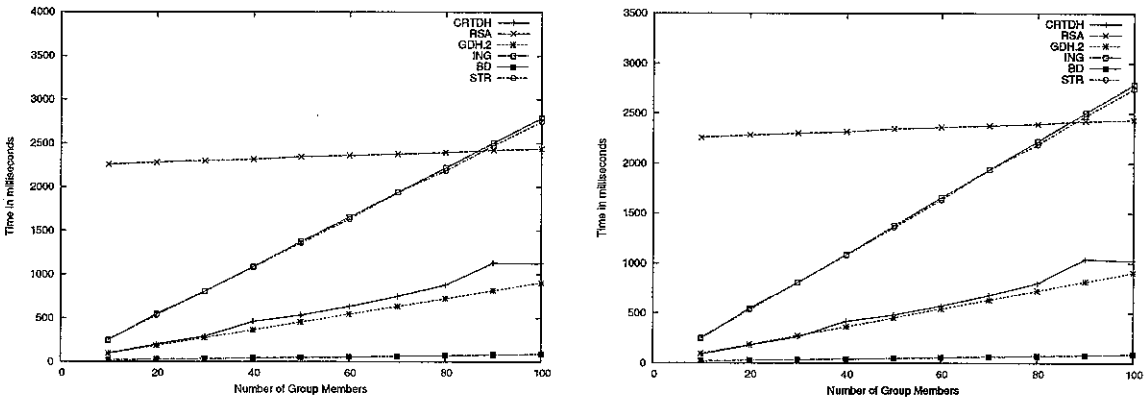

Fig. 1. Computation time: a single member for group key establishment (left) and a joining member for join operation (right).

load on the other members during member join/leave will be considered later in this section.

The computation time for a member leave operation is given in Figure 2 (left). The CRTDH and BD protocols have the least computation times compared to the other protocols. The other protocols have similar computation times as the key establishment and join operations since all these three operations are quite similar. Again, the CRTDH leave operation involves only one round of communication and less overhead on existing members whereas the BD protocol involves two rounds of communication and high overhead on the remaining members in the group.

The overhead on an existing member when another member joins/leaves the group is an important factor. In MANETs with high mobility there may be frequent

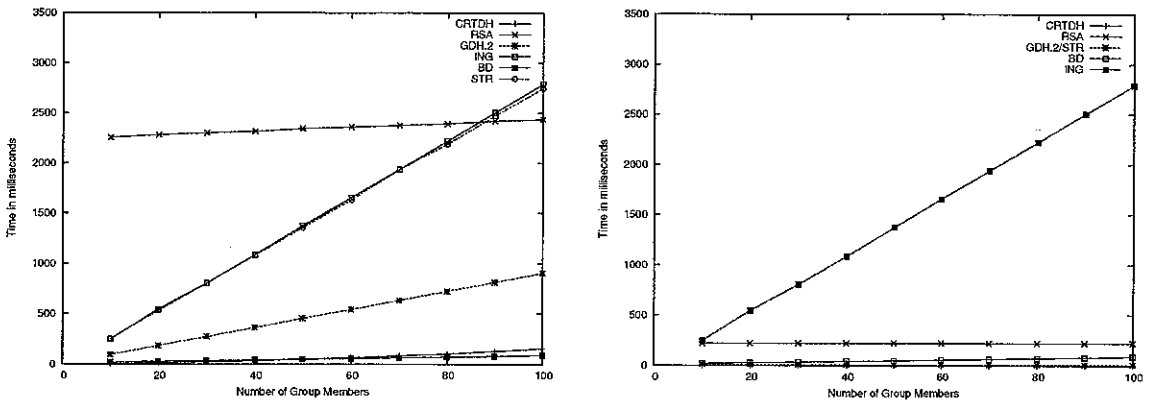

Fig. 2. Computation time for leave operation (left) and overhead on an existing member when another member joins/leaves (right).

DOI: 10.1002/wcm

group membership changes, hence it is desirable that the overhead on an existing member during such events is minimal in order to save battery power. The graph in Figure 2 (right) shows the overhead on one of the existing members when another member joins/leaves the group. As can be seen from Figure 2 (right), the overhead on an existing member is highest when the ING protocol is used. In order to make the graph more clearer, another graph without the ING computation times is shown in Figure 3 (left). It can be seen that the overhead on an existing member is constant as the group size increases for the RSA and GDH.2/STR protocols since they involve one RSA decryption and one DH public exponentiation, respectively. The computational overhead for the BD protocol increases as the group size increases. The CRTDH protocol has the least overhead for an existing member compared to all the other protocols. This is a desired property in MANETs as previously discussed.

It is also interesting to find out the overhead on the system when a member joins/leaves the group. The system here refers to all the existing members in the group, hence finding the sum of the computational overhead for all the members in the group gives a good indication about the overhead of the protocol. The graph in Figure 3 (right) gives the sum of the computational overhead of all the existing members when another member joins/leaves the group. As can be seen from Figure 3 (right), the ING protocol has the maximum overhead followed by the RSA based scheme. In order to make the graph more clearer, both the ING and RSA computation overheads are removed from the graph in Figure 4 (left). It can be seen that the CRTDH protocol has the least overhead on the system. With the BD protocol the overhead on the system grows with the increase in group sizes. In the case of highly dynamic environments with frequent membership changes, such a high overhead is not preferred.

From the above graphs, it can be seen that the CRTDH key establishment process is efficient compared to the other schemes except for the BD protocol. The join operation is similar to the key establishment process but can be performed in one round of communication. The leave operation is most efficiently supported by the CRTDH protocol. With regard to the overhead on the existing members in the group, the CRTDH scheme performs better than all the other schemes compared above. Considering the other CRTDH features such as no member serialization, no GC, no pre-shared secret, and uniform work load, the CRTDH protocol has the best overall performance.

6.3. Comparisons of AUTH-CRTDH with CRTDH

The performance of AUTH-CRTDH is simulated and compared with CRTDH. The three figures give examples for this close relation between AUTH-CRTDH and CRTDH. Figure 4 (right) shows the initial group key establishment complexity. Figure 5 (left) shows the computation time for join operation and Figure 5 (right) shows the computation time for leave operation, all in terms of CRTDH and AUTH-CRTDH. As can be seen, the results show that AUTH-CRTDH is quite efficient and its performance is within a constant factor (i.e., 3) of CRTDH's performance. The reason behind this is that AUTH-CRTDH follows the same steps as CRTDH, the only difference is that for authentication purpose, AUTH-CRTDH computes five times more exponentiations of CRTDH.

6.4. Some Implementation Related Issues

Due to the space limitation, the paper has focused on proposing the new CRTDH/AUTH-CRTDH scheme by presenting its main principles and security and performance features. There are some other related issues for a complete and runnable SGC key management protocol which the paper has not covered. Some typical issues include, for example, (1) how is the group key establishment process started? (2) who defines the size of a group? (3) during a joining/leaving operation, when does the new key come into use? (4) what is the renewal process and when does the renewal occurs? (5) what needs to be done in case a rekeying message is lost? Many of these issues are either application-dependent or related to implementation details. As for (1), all initial group members can start the key agreement process simultaneously if they knew their existence well at the beginning. An alternative way is to start the key agreement process incrementally: two members begin first, then other members are added *via* joining operations. Some other mechanisms may also be used. Moreover, the group key is computed by XORing all key shares which are random numbers. Thus the key shares can be generated using a pseudo-random number generator. As for (2), the group size can be determined either by a third party, one of the group members, majority of the group members, or all members of the group. It is worth mentioning that the CRTDH/AUTH-CRTDH scheme does not limit the size of a group. As for (3), this is the very issue of how to keep keys synchronized. Some mechanism such as

DOI: 10.1002/wcm

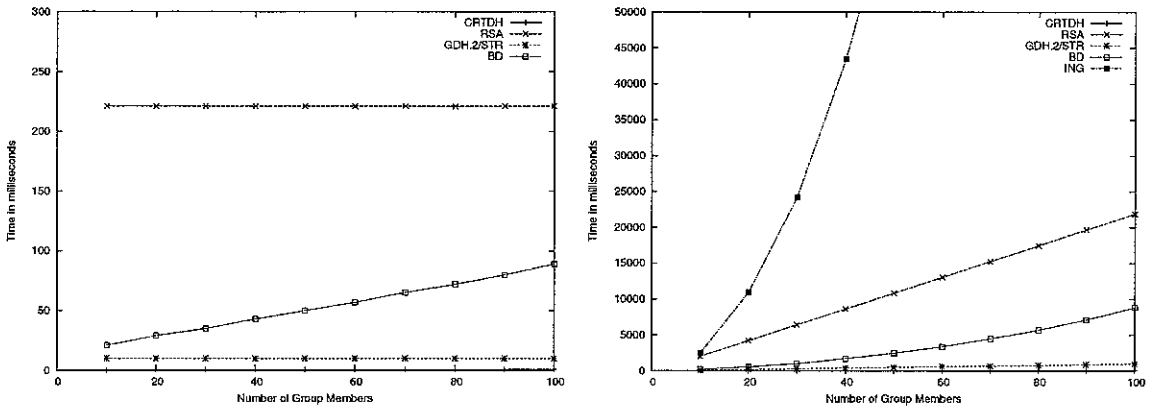

Fig. 3. Overhead: on an existing member when another member joins/leaves without ING computation times (left) and on the system when a member joins/leaves (right).

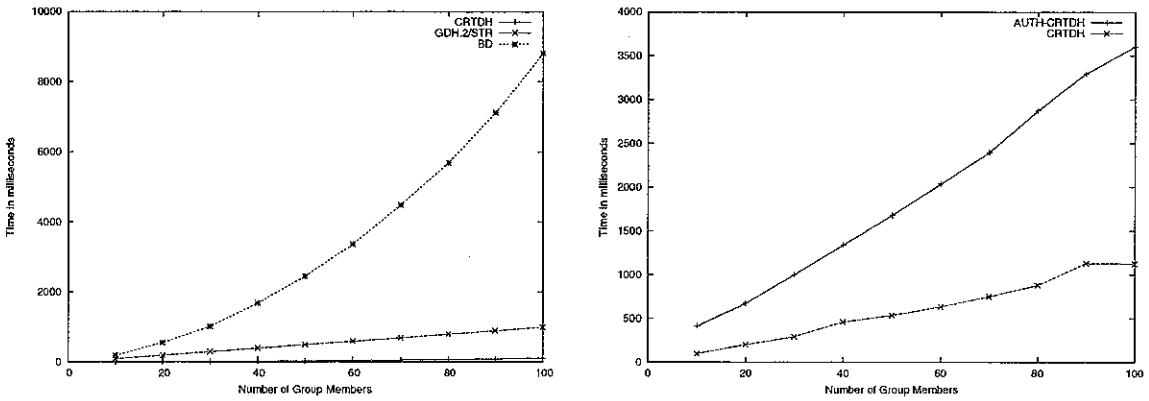

Fig. 4. Overhead on the system when a member joins/leaves without ING and RSA (left) and establishment computation times of CRTDH and AUTH-CRTDH (right).

versioning keys can be used: whenever a new group key is computed, the key version increases by one and the new key can begin to be used for encryption. When a member sends a data message, it also includes the version of the key used to encrypt the message in the transmission packet. The receiving member of the message will use the corresponding version key to decrypt the message. Key renewal, that is, (4), is an important issue. Periodically renewing keys will prevent attackers from getting enough time to crack a group key. In

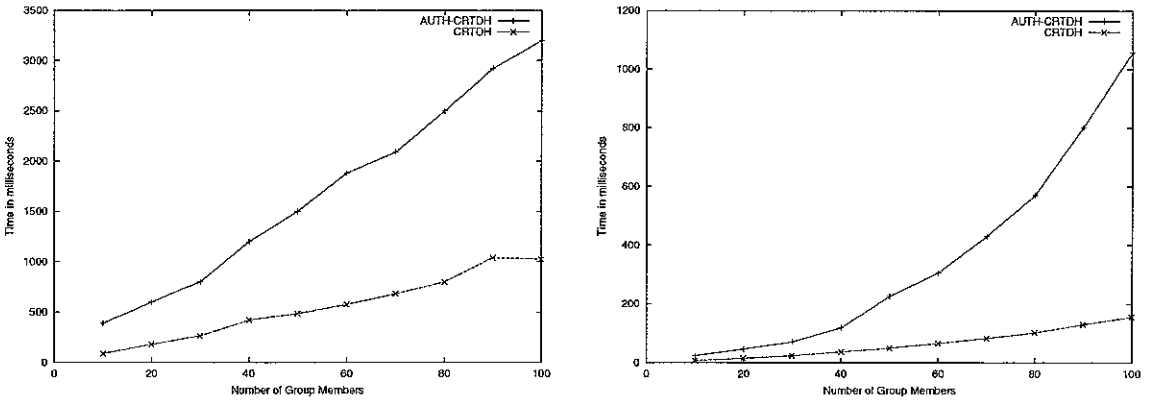

Fig. 5. CRTDH and AUTH-CRTDH computation times: Join (left) and Leave (right).

DOI: 10.1002/wcm

CRTDH/AUTH-CRTDH, the renewal process can be easily initiated by any of the group members and the group key will get renewed when the initiating member selects and broadcasts a new key share k_i. As for the renewal frequency, it is dependent on applications features and it is also a tradeoff between efficiency and security. As for the message loss, that is, (5), the self-healing mechanism can be incorporated to mitigate it.

These issues, along with more others, can be investigated further in the future implementation of AUTH-CRTDH.

7. Conclusions

With its numerous advantages, wireless networks are set to dominate the field of computer networking. Security of information over such networks is of utmost importance to ensure reliable services. In this work, we studied the problem of 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 (and its enhanced version AUTH-CRTDH), which does not require member serialization. When compared to other key management schemes for ad hoc networks, the proposed scheme satisfies all the desirable properties that we have identified. In addition, AUTH-CRTDH is able to defend against the attacks of Man-in-the-Middle and LCM.

Acknowledgment

This work was supported in part by the U. S. NSF grants (CCR-0311577 and EPS-0091900).

References

1. Wong CK, Gouda M, Lam SS. Secure group communications using key groups. In *SIGCOMM'98*, pp. 68–79, December 1998.
2. Kim Y, Perrig A, Tsudik G. Simple and fault-tolerant key agreement for dynamic collaborative groups. In *Proceedings of the 7th ACM Conference on Computer and Communication Security*, pp. 235–244, November 2000.
3. Mittra S. Iolus a framework for scalable secure multicasting. *Journal of Computer Communication Reviews* 1997; **27**: 277–288.
4. Dondeti LR. *Efficient Private Group Communication Over Public Networks*. Ph.D. thesis, Computer Science and Engineering, University of Nebraska-Lincoln, 1999.
5. Steiner M, Tsudik G, Waidner M. Cliques: a new approach to group key agreement. In *Proceedings of the 18th International Conference on Distributed Computing Systems*, pp. 380–387, May 1998.
6. Zou X, Ramamurthy B, Magliveras S. Chinese remainder theorem based hierarchical access control for secure group communications. In *International Conference on Information and Communication Security*, vol. 2229, pp. 381–385, 2001.
7. Eschenauer L, Gligor V. A key-management scheme for distributed sensor networks. In *Proceedings of 9th ACM Conference on Computer and Communication Security*, pp. 41–47, November 2002.
8. Du W, Han YS, Deng J, Varshney PK. A pairwise key pre-distribution scheme for wireless sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communication Security*, pp. 42–51, October 2003.
9. Liu D, Ning P. Establishing pairwise keys in distributed sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communication Security*, pp. 52–61, October 2003.
10. Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks. In *Proceedings of the IEEE Security and Privacy Symposium*, p. 197, May 2003.
11. Ren K, Zeng K, Lou W. A new approach for random key pre-distribution in large-scale wireless sensor networks. *Wireless Communications and Mobile Computing* 2006; **6**(3): 307–318.
12. Zhu S, Xu S, Setia S, Jajodia S. Establishing pair-wise keys for secure communication in ad-hoc networks: a probabilistic approach. In *IEEE International Conference on Network Protocols*, p. 326, March 2003.
13. Perrig A, Szewczyk R, Wen V, Culler D, Tygar J. Spins: security protocols for sensor networks. In *Proceedings of 7th ACM Mobicom*, pp. 189–199, July 2001.
14. DeCleen B, Dondeti L, Griffin S, *et al.* Secure group communications for wireless networks. In *IEEE Military Communications Conference (MILCOM)*, pp. 113–117, October 2001.
15. Li R, Li J, Liu P, Chen H-H. On-demand public-key management for mobile ad hoc networks. *Wireless Communications and Mobile Computing* 2006; **6**(3):295–307.
16. Li XY, Wang Y, Frieder O. Efficient hybrid key agreement protocol for wireless ad-hoc networks. In *IEEE 11th International Conference on Computer, Communication and Networks*, pp. 404–409, October 2002.
17. Yasinsac A, Thakur V, Carter S, Cubukcu I. A family of protocols for group key generation in ad hoc networks. In *IASTED Conference on Communication and Computer Networks*, pp. 183–187, November 2002.
18. Balachandran RK, Ramamurthy B, Zou X, Vinodchandran NV. CRTDH: an efficient key agreement scheme for secure group communications in wireless ad hoc networks. In *Proceedings of the IEEE International Conference on Communications*, pp. 41–47, May 2005.
19. Stajano F, Anderson R. The resurrecting duckling: security issues for ad-hoc wireless networks. In *Security Protocols, 7th International Workshop*, pp. 172–194, April 1999.
20. Diffie W, Hellman M. New directions in cryptography. *IEEE transactions on Information Theory* 1976; **IT-22**(6): 644–654.
21. Steiner M, Tsudik G, Waidner M. Diffie-hellman key distribution extended to group communication. In *Proceedings of 3rd ACM Conference on Computer and Communication Security*, pp. 31–37, May 1996.
22. Steiner M, Tsudik G, Waidner M. Key agreement in dynamic peer groups. *IEEE Transactions on Parallel and Distributed Systems* 2000; **11**(8): 769–779.
23. Steer D, Strawczynski L, Diffie W, Wiener M. A secure audio teleconference system. In *Advances in Cryptology—CRYPTO 88*, pp. 520–528, August 1990.
24. Kim Y, Perrig A, Tsudik G. Communication efficient group key agreement. In *Proceedings of the 17th International Information Security Conference*, pp. 229–244, June 2001.

DOI: 10.1002/wcm

25. Ingemarsson I, Tang D, Wong C. A conference key distribution system. *IEEE Transactions on Information Theory* 1982; **28**(5): 714–720.
26. Burmester M, Desmedt Y. A secure and efficient conference key distribution system. In *Advances in Cryptology—EUROCRYPT*, pp. 275–286, May 1994.
27. Harney H, Harder H. Logical key heirarchy protocol. In *Internet Draft, Internet Engineering Task Force*, April 1999.
28. Perrig A, Tygar JD. *Secure Broadcast Communication in Wired and Wireless Networks*. Kluwer Academic Publishers: Norwell, MA, USA, 2002.
29. Dondeti LR, Mukherjee S, Samal A. Disec: a distributed framework for scalable secure many-to-many communication. In *IEEE Symposium on Computers and Communication*, pp. 693–698, July 1999.
30. Hubaux J, Buttyan L, Capkun S. The quest for security in mobile ad hoc networks. In *the Proceeding of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, pp. 146–155, October 2001.
31. Zhou L, Haas ZJ. Securing ad hoc networks. *IEEE Networks* 1999; **13**: 24–30.
32. Yi S, Kravets R. Key management for heterogeneous ad hoc wireless networks. In *IEEE International Conference on Network Protocols*, pp. 202–205, November 2002.
33. Kong J, Zerfos P, Luo H, Lu S, Zhang L. Providing robust and ubiquitous security support for wireless mobile networks. In *Internation Conference on Network Protocols (ICNP'01)*, pp. 251–260, November 2001.
34. Huang Q, Cukier J, Kobayashi H, Liu B, Zhang J. Fast authenticated key establishment protocols for self-organizing sensor networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pp. 141–150, September 2003.
35. Khalili A, Katz J, Arbaugh WA. Towards secure key distribution in truly ad-hoc networks. In *IEEE Workshop on Security and Assurance in Ad-Hoc Networks*, pp. 342–346, January 2003.
36. Pietro RD, Mancini LV, Mei A. Random key-assignment for secure wireless sensor networks. In *Proceedings of the 1st ACM Workshop on Security of ad hoc and Sensor Networks*, pp. 62–71, October 2003.
37. Basagni S, Herrin K, Rosti E, Bruschi D. Secure pebblenets. In *Proceedings of ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)*, pp. 156–163, October 2001.
38. Asokan N, Ginzboorg P. Key agreement in ad-hoc networks. *Computer Communications* 2000; **23**: 1627–1637.
39. Pietro RD, Mancini LV, Jajodia S. Efficient and secure keys management for wireless mobile communications. In *Proceedings of the Second ACM International Workshop on Principles of Mobile Computing*, pp. 66–73, 2002.
40. Hietalahti M. Efficient key agreement for ad-hoc networks. Master's thesis, Helsinki University of Technology, 2001.
41. Stinson DR. *Cryptography: Theory and Practice* (3rd edn). CRC Press: Boca Raton, FL, 2005.
42. Yang W, Shieh S. Secure key agreement for group communications. *International journal of Network Management* 2001; **11**(6): 365–374.
43. Shamir A. Identity-based cryptosystems and signature schemes. *Advances in Cryptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science* 1984; **7**: 47–53.
44. Dai W. Crypto++ library. In http://www.eskimo.com/weidai/cryptlib.html

Authors' Biographies

Ravi K. Balachandran is a Software Development Engineer with Microsoft at their Hyderabad campus in India. He obtained his M.S. degree from the University of Nebraska-Lincoln, U.S.A. in 2004. His research interests include network security and wireless networks.

Xukai Zou is an Assistant Professor in the Department of Computer and Information Sciences at Indiana University-Purdue University Indianapolis, U.S.A. He completed his Ph.D in Computer Science from University of Nebraska-Lincoln in 2000. His research focus is in applied cryptography, network security, and communication networks.

Byrav Ramamurthy received his Ph.D. in Computer Science from University of California (UC), Davis in 1995. Since August 2003, he has been an associate professor in the Department of Computer Science and Engineering at the University of Nebraska-Lincoln (UNL). He is the founding Co-Director of the Advanced Networking and Distributed Experimental Systems (ANDES) Laboratory at UNL. His research areas include optical and wireless networks, distributed systems, computer security and telecommunications.

Mr Amandeep Thukral is working as a Software Consultant for Deloitte Consulting LLP since February 2006. He received his Masters Degree in Computer and Information Science from Purdue School of Science, Indianapolis in December 2005. His research area is cryptography and network security specific to Secure Group Communication (SGC).

Prof. Vinodchandran N. Variyam is an Associate Professor in the Department of Computer Science and Engineering at the University of Nebraska-Lincoln. He received his Ph.D. from the Institute of Mathematical Sciences, Chennai, India. His research is in the area of computational complexity theory and its applications to computational learning theory and network security. He has published widely in theoretical computer science journals and conferences.

DOI: 10.1002/wcm