

Notes on the CRTDH Group Key Agreement Protocol*

Spyros Magliveras and Wandi Wei
Center for Cryptology and Information Security
Department of Mathematical Sciences
Florida Atlantic University
Boca Raton, FL 33431, USA
Email: spyros@fau.edu and wei@brain.math.fau.edu

Xukai Zou
Computer and Information Science Department
Indiana University Purdue University Indianapolis
Indianapolis IN 46202, USA
Email: xkzou@cs.iupui.edu

Abstract

Group key management (GKM) is one of the primary issues for secure group communication (SGC). Contributory group key agreement is crucial for SGC over wireless and mobile ad hoc networks (MANETs) when there is lack of a fixed infrastructure and it is difficult to have a central trusted entity. Recently, Balachandran et al. proposed a contributory key agreement protocol for MANETs, called CRTDH [5]. This protocol is based upon both the Chinese Remainder Theorem and the Diffie-Hellman key exchange scheme. CRTDH exhibits a very nice idea, but contains some problems that make it not very practical in terms of efficiency and security. We point out these problems and propose a solution to them. Analysis and experiments are also presented which show our improved protocol outperforms the original CRTDH protocol in terms of both efficiency and security.

Keywords: Secure Group Communication, Group Key Management (GKM), Contributory Group Key Agreement, Chinese Remainder Theorem, Diffie-Hellman Key Exchange.

1 Introduction

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 sce-

nario, a group key is established among all the participating members and this key is used to encrypt all the messages destined to the group. As a result, only the group members can decrypt the messages. The group key management protocols are typically classified in four categories: centralized group key distribution (CGKD), de-centralized group key management (DGKM), distributed/contributory group key agreement (CGKA), and distributed group key distribution (DGKD). (1) In CGKD, there exists a central entity (i.e. a group controller GC) which is responsible for generating, distributing, and updating the group key. The most famous CGKD scheme is the key tree scheme (also called Logical Key Hierarchy (LKH) in some papers). This scheme was independently proposed by several research groups nearly at the same time [33, 34, 12, 24], followed by many researchers proposing improvements and enhancements [11, 19, 21, 23, 25, 29, 30, 37]. (2) The DGKM approach involves splitting a large group into small subgroups. Each subgroup has a subgroup controller which is responsible for the key management of its subgroup. Subgroup controllers are also in charge of relaying encrypted data messages. The first DGKM scheme to appear was IOLUS [22]. There followed some improvements and hierarchical group key management schemes [6, 9, 13, 27, 29]. (3) The CGKA schemes involve the participation by all members of a group towards key management. Such schemes are characterized by the absence of the GC. The group key in such schemes is a function of the secret shares contributed by the members. Being contributory in nature, the distributed schemes help in the uniform distribution of the work-load for key management and eliminate the requirement for a central

*The authors are listed in the alphabetic order of their last names.

trusted entity. Typical CGKA schemes include binary tree based ones [14, 18] and n -party Diffie-Hellman key agreement [2, 10, 17, 8, 31, 18]. (4) The DGKD scheme, proposed in [1], eliminates the need for a trusted central authority and introduces the concepts of sponsors and co-distributors. All group members have the same capability and are equally trusted. Also, they have equal responsibility, i.e. any group member could be a potential sponsor of other members or a co-distributor. Whenever a member joins or leaves the group, the member's sponsor initiates the rekeying process. The sponsor generates the necessary keys and securely distributes the keys to co-distributors respectively. The co-distributors then distribute in parallel, corresponding keys to corresponding members. In addition to the above four typical classes of key management schemes, there are some other forms of key management schemes such as hierarchy and cluster based ones [6, 16]. The paper in [28] presents a good survey of earlier group key management schemes and the book [38] presents a comprehensive coverage of SGC as well as secure dynamic conferencing (SDC) and hierarchical access control (HAC).

Wireless networks, in particular mobile ad hoc networks (MANETs) and wireless sensor networks, 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, secure group communication over such networks becomes vital. In MANETs, since there is no pre-defined/fixed infrastructure, a central authority is not usually available, and there are generally equivalent levels of power and trust (or to say, untrust) among the participating members. A contributory group key agreement scheme is most appropriate for SGC in this kind of environment. Several group key management schemes have been proposed for SGC in wireless networks [3, 7, 15, 20, 26, 32, 35, 36]. Recently, Balachandran et al. proposed such a contributory key agreement protocol for MANETs, called CRTDH [5]. This protocol is based upon both the Chinese Remainder Theorem and the Diffie-Hellman key exchange scheme. CRTDH exhibits a very nice idea, but contains some problems that make it impractical in terms of efficiency and security. We point out three such problems: *low efficiency, possibly a small key, and possessing the same Least Common Multiple (LCM)*, and propose a solution to them. Analysis and experiments are also presented which show that our improved protocol outperforms the original CRTDH protocol in terms of both efficiency and security.

The rest of the paper is organized as follows. In Section 2 we briefly introduce the original CRTDH protocol. In Section 3 we point out problems with CRTDH and propose improvements. Section 4 includes a performance analysis and experimental comparison between the original and

improved CRTDH. Finally, Section 5 concludes the paper.

2 A brief description of the original CRTDH key agreement protocol

For simplicity, we use the following notations.

- U_i ($1 \leq i \leq n$): group members;
- p : a large prime;
- g : a primitive root or generator of Z_p^* ;
- h : a cryptographic hash function;
- CRT: Chinese Remainder Theorem;
- DH: Diffie-Hellman;
- GCD: Greatest Common Divisor;
- LCM: Least Common Multiple.

2.1 Group key establishment

Every group member U_i does as follows.

- Step 1.1. Selects a DH private share x_i , computes $y_i = g^{x_i} \bmod p$.
- Step 1.2. Broadcasts y_i .
- Step 1.3. Computes $m_{ij} = y_j^{x_i} \bmod p$, $j \neq i$.
- Step 1.4. Computes LCM $lcm_i = LCM_{j \neq i} \{m_{ij}\}$.
- Step 1.5. Selects random k_i , D_i , and P_i such that $k_i < \min_{j \neq i} \{m_{ij}\}$ and $D_i \neq k_i$, and $\gcd(P_i, lcm_i) = 1$.
- Step 1.6. Solves the congruences

$$\begin{aligned} crt_i &\equiv k_i \pmod{lcm_i} \\ crt_i &\equiv D_i \pmod{P_i} \end{aligned}$$

and broadcasts crt_i .

- Step 1.7. Computes k_j and the group key GK as $k_j = crt_j \bmod m_{ij}$, $j \neq i$, and $GK = k_1 \oplus k_2 \oplus \dots \oplus k_n$.

2.2 The CRTDH join operation

Suppose that U_{n+j} ($1 \leq j \leq l$) are going to join the group.

- Step 2.1. U_i ($1 \leq i \leq n$) compute the hash value $h(GK)$. One of them, say U_1 , transmits $h(GK)$ and y_i ($1 \leq i \leq n$) to U_{n+j} ($1 \leq j \leq l$).
- Step 2.2. U_{n+j} ($1 \leq j \leq l$) execute Steps 1.1-1.2, execute Step 1.3 only for $m_{n+j, n+t}$ ($1 \leq t \leq l$; $t \neq j$), execute Step 1.4 as $lcm_{n+j} = LCM_{1 \leq t \leq l; t \neq j} \{m_{n+j, n+t}\}$, execute Steps 1.5-1.6 with i, j ($n < i, j < n + l$), and broadcast crt_{n+j} , y_{n+j} .
- Step 2.3. U_i ($1 \leq i \leq n + l$) recover k_{n+j} ($1 \leq j \leq l$) and compute the new group key as $GK_{new} = h(GK) \oplus k_{n+1} \oplus \dots \oplus k_{n+l}$.

2.3 The CRTDH leave operation

Suppose that $n > s > 1$ and U_{s+j} ($1 \leq j \leq n - s$) are going to leave.

- Step 3.1. One of U_i ($1 \leq i \leq s$), say U_1 , re-does Steps 1.4-1.6 with a new k'_1 and a new $lcm_1 = LCM_{2 \leq i \leq s} \{m_i\}$, and computes and broadcasts the new crt_1 , and computes the new group key $GK_{new} = GK \oplus k_1$.
- Step 3.2. U_i ($2 \leq i \leq s$) recover k_1 from crt_1 and then compute the new group key $GK_{new} = GK \oplus k_1$.

3 The CRTDH problems and improvements

3.1 Problems in the original CRTDH

There are some problems with the CRTDH scheme which we discuss below as follows.

Problem 1. In Step 1.4, the value of lcm_i can be as large as

$$O((p-1)^{n-1}),$$

which can be prohibitively large since p is large and n is not very small, and then the storage needed for *and* operations (e.g., in Step 1.6) on lcm_i might not be practical.

Problem 2. Since x_i are chosen independently by U_i , it is possible that some of the m_{ij} in step 1.3 are small. If so, the randomness of k_i in Step 1.5 is affected and the security strength may be reduced.

Problem 3. Because of the fact that the LCM for a given set of numbers is not unique to the given set, the “*same LCM*” problem can occur. In other words, there can be more numbers that are added to a set and still result in the same LCM value. For example, $LCM\{4,6,8\}=LCM\{4,6,8,12\}=24$. Adding 12 into the previous set or removing 12 from the later set will not change the LCM value of 24. This could cause problems in the member join and member leave operations. Let us consider the *member join* operation first. The problem arises if the shared DH key between the existing user and the newly joined member does not affect the LCM of the new set. This could lead to breaching of backward secrecy: the new member would be able to obtain the secret share of the existing members. As for the *member leave* operation, the problem arises once again if the new LCM value without the departing member, is the same as the LCM value when the departing member was still in the group. In such a case, the departing member would still be able to decrypt new messages (so violating forward secrecy).

3.2 A modified key agreement protocol

The above problems can be resolved by modifying the CRTDH protocol as follows.

Steps 1.1'-1.2' are the same as Steps 1.1-1.2, respectively.

Steps 1.3'. Let¹

$$m_{ij} = \begin{cases} y_j^{x_i} \bmod p & \text{if } y_j^{x_i} \bmod p > \frac{p}{2} \\ p - y_j^{x_i} \bmod p & \text{otherwise} \end{cases}$$

Step 1.4 will be removed.

Step 1.5'. For a given $j \neq i$, U_i chooses P_{ij} such that $\gcd(P_{ij}, m_{ij}) = 1$.

Step 1.6'. For $j \neq i$, U_i uses the following new congruences to replace the original ones:

$$\begin{aligned} crt_{ij} &\equiv k_i \pmod{m_{ij}} \\ crt_{ij} &\equiv D_i \pmod{P_{ij}} \end{aligned}$$

where D_i is the same as in Step 1.6. U_i also broadcasts the set of pairs²

$$crt'_i = \{(U_j, crt_{ij}) : j \neq i\},$$

Each group member U_j finds his own matched crt_{ij} ($i \neq j$) from crt'_i broadcast by U_i . Note that usually, we have $crt_{ij} \neq crt_{ji}$.

Step 1.7'. U_i Computes $crt_{ji} \bmod m_{ij}$, $j \neq i$, which must be k_j since $m_{ij} = m_{ji}$, and then computes $GK = k_1 \oplus k_2 \oplus \dots \oplus k_n$, which is the group key.

3.3 A modified join operation

Suppose that U_{n+j} ($1 \leq j \leq l$) are going to join the group.

Step 2.1' is the same as Step 2.1, i.e., U_i ($1 \leq i \leq n$) compute the hash value $h(GK)$. One of them, say U_1 , transmits $h(GK)$ and y_i ($1 \leq i \leq n$) to U_{n+j} ($1 \leq j \leq l$).

Step 2.2'. U_{n+j} ($1 \leq j \leq l$) execute Steps 1.1'-1.2', execute Step 1.3' only for $m_{n+j,n+t}$ ($1 \leq t \leq l$; $t \neq j$), and execute Steps 1.5'-1.6' only for $crt_{n+j,n+t}$ ($1 \leq t \leq l$; $t \neq j$), i.e., compute and broadcast y_{n+j} and $crt'_{n+j} = \{crt_{n+j,n+t} : 1 \leq t \leq l; t \neq j\}$.

Step 2.3'. U_i ($1 \leq i \leq n+l$) recover k_{n+j} ($1 \leq j \leq l$) by using the method in Step 1.7', and compute the new group key as $GK_{new} = h(GK) \oplus k_{n+1} \oplus \dots \oplus k_{n+l}$.

¹This will solve Problem 2 since m_{ij} will not be small.

²This will solve both Problem 1 and Problem 3 since there is no involvement of LCM, also see the performance analysis in the next section.

3.4 A modified leave operation

Suppose that $n > s > 1$ and U_{s+j} ($1 \leq j \leq n - s$) are going to leave.

Step 3.1'. One of U_i ($1 \leq i \leq s$), say U_1 , redoes Steps 1.5'-1.6' with a new k'_1 , broadcasts the new $crt'_1 = \{crt_{1,t} : 2 \leq t \leq s\}$,

and computes the new group key $GK_{new} = GK \oplus k_1$.

Step 3.2'. U_i ($2 \leq i \leq s$) recover k_1 from crt'_1 and then compute the new group key $GK_{new} = GK \oplus k_1$.

It is worthy to mention that CRTDH, including the above improvement, does not perform user authentication in the key agreement process, thus, suffering from the Man-in-the-Middle attack. The detail discussion of such a problem and one possible solution can be found in [39].

4 Performance analysis

In this section, we analyze the complexities of the improved protocol and the CRTDH protocol. We also performed experiments on both protocols and the experimental results are presented here.

Theorem 4.1 *Suppose p is the system prime and n is the group size, then the time complexity executing the CRTDH protocol is $O(n^2 \lg p)$ while the time complexity executing the improved protocol is $O(n \lg p)$. Thus, the improved protocol is faster than the original CRTDH protocol by a factor of n .*

Proof: For proving the above complexity, we can use some results from [4] which are restated here:

Corollary 4.2.4. Let u, v be (positive) integers. we can compute the gcd of u and v using $O((\lg u)(\lg v))$ bit operations.

Corollary 5.5.6. Let m_1, m_2, \dots, m_k be integers, each ≥ 2 , and define $m = m_1 m_2 \dots m_k$, and $m' = \text{lcm}(m_1, m_2, \dots, m_k)$. Given the system S of congruences $x \equiv x_i \pmod{m_i}$, $1 \leq i \leq k$, we can determine if S has a solution, using $O((\lg m)^2)$ bit operations, and if so, we can find the unique solution $(\text{mod } m')$, using $O((\lg m)^2)$ bit operations.

From the above Corollary 4.2.4, it can be obtained that the time complexity for computing $\text{lcm}(u, v)$ is also $O((\lg u)(\lg v))$ bit operations since $\text{lcm}(u, v) = (u \times v) / \text{gcd}(u, v)$ (or $(u / \text{gcd}(u, v)) \times v$).

Let us analyze CRTDH first. The time complexity of CRTDH mainly comes from two operations: computing lcm_i and then solving crt_i . There are at least two ways to compute $\text{lcm}_i (= \text{lcm}\{m_{i1}, \dots, m_{i(i-1)}, m_{i(i+1)}, \dots, m_{in}\})$. The first one is by the linearly recursive formation: $\text{lcm}_{i2} = \text{lcm}\{m_{i1}, m_{i2}\}$, \dots , $\text{lcm}_{ij} = \text{lcm}\{\text{lcm}_{i(j-1)}, m_{ij}\}$ ($j \neq i, j \geq 3$), \dots , $\text{lcm}_i = \text{lcm}_{in} = \text{lcm}\{\text{lcm}_{i(n-1)}, m_{in}\}$. A second method is by the binary recursive formation:

$\text{lcm}_i = \text{lcm}_{i(1,n)} = \text{lcm}\{\text{lcm}_{i(1,n/2)}, \text{lcm}_{i(n/2+1,n)}\}, \dots$, $\text{lcm}_{i(j,k)} = \text{lcm}\{\text{lcm}_{i(j,(j+k)/2)}, \text{lcm}_{i((j+k)/2+1,k)}\}$ and $\text{lcm}_{i(j,j+1)} = \text{lcm}\{m_{ij}, m_{i(j+1)}\}$ ($j \neq i$). No matter which method is used, their complexity is $O(n^2(\lg p)^2)$. As for computing crt_i , since lcm_i can reach the magnitude of $(p-1)^{n-1}$ and $\text{lcm}_i \times P_i$ is in the magnitude of $(p-1)^n$, by Corollary 5.5.6, the time complexity for computing crt_i is of $O(n^2(\lg p)^2)$. Thus, the total time complexity in the CRTDH protocol is $O(n^2(\lg p)^2)$.

With regard to the improved protocol, by the above Corollary 5.5.6, computing every crt_{ij} requires $O((\lg p)^2)$. There are $n-1$ crt_{ij} , so the total time complexity in the improved protocol is $O(n(\lg p)^2)$. As a result, the improved protocol is more efficient than the CRTDH protocol by a factor of n . \square

As for the communication complexity, crt_i will be of length $(n-1)(\lg p)$ bits and the combination of all crt_{ij} will also have length of $(n-1)(\lg p)$ bits. Therefore, the two protocols have the same communication cost.

Table 1. Running times of computing lcm_i , crt_i and crt_{ij} for 32-bit prime

#users	CRTDH			Improved
	lcm_i	crt_i	$\text{lcm}_i + crt_i$	Total crt_{ij}
64	551768	1176071	1727839	1582363
128	1813671	3515796	5329467	3191863
256	6631611	13530292	20161903	6415793
512	23518885	45259282	68778167	12985918
1024	90008450	169990247	259998697	25822287

Table 2. Running times of computing lcm_i , crt_i and crt_{ij} for 64-bit prime

#users	CRTDH			Improved
	lcm_i	crt_i	$\text{lcm}_i + crt_i$	Total crt_{ij}
64	1816920	5139217	6956137	3332764
128	6335419	18082233	24417652	6678804
256	23666491	70995992	94662483	14167649
512	93344922	271524897	364869819	27078570
1024	348512973	1003013296	1351526269	54375547

We performed the experiments for computing the CRT values on a DELL laptop with a 1.8GHz Intel(R) Pentium (R) M processor and 1GB RAM under Windows XP using JAVA. The running time for each category is averaged over 100 runs. The experimental results validate the theoretical analysis. For example, when p is a 32-bit, 64-bit prime respectively, and the number of members n is 64, 128, 256, 512, and 1024, the running times in terms of nanoseconds for computing lcm_i , crt_i (in the CRTDH protocol) and crt_{ij} (in the improved protocol) are shown in

Tables 1 and 2, also shown in Figures 1 and 2.

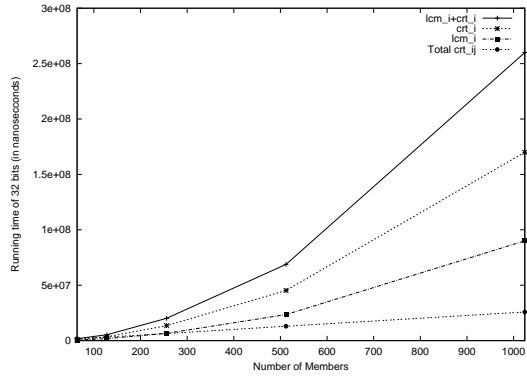


Figure 1. Running times of computing lcm_i , crt_i and crt_{ij} for 32 bits

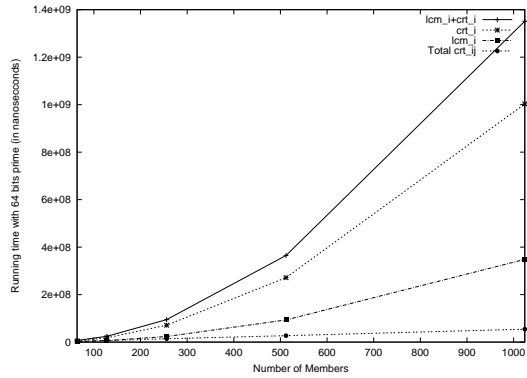


Figure 2. Running times of computing lcm_i , crt_i and crt_{ij} for 64 bits

It is worth pointing out that one other modification to the CRTDH protocol is to form the congruences for all m_{ij} and compute one overall CRT value:

$$\begin{aligned}
 crt_i &\equiv k_i \pmod{m_{i1}} \\
 &\vdots \\
 crt_i &\equiv k_i \pmod{m_{i,i-1}} \\
 crt_i &\equiv k_i \pmod{m_{i,i+1}} \\
 &\vdots \\
 crt_i &\equiv k_i \pmod{m_{in}} \\
 crt_i &\equiv D_i \pmod{P_i}
 \end{aligned}$$

There are some problems with this solution. At first, the time complexity for computing the above crt_i is $O(n^2lgp)$, rather than $O(nlgp)$. Secondly, it is required that all m_{ij} be pairwise co-prime, which is difficult to achieve. Whenever

there are some m_{ij} which are not co-prime, there is a need to ask the related members to reselect a new x_i , recompute y_i , and recompute m_{ij} . This process may require several repetitions, and consequently is inefficient.

5 Conclusion

In this paper, we point out three problems that are present in the CRTDH protocol and propose corresponding solutions. We also analyze the performance of our improved protocol and compare it with the original CRTDH protocol, both theoretically and by experiments. The results show that the improved protocol has higher efficiency than the original CRTDH. The improved protocol also has higher security strength than the original CRTDH when some of the m_{ij} are small.

References

- [1] P. Adusumilli, X. Zou, and B. Ramamurthy. DGKD: Distributed group key distribution with authentication capability. *Proceedings of 2005 IEEE Workshop on Information Assurance and Security, West Point, NY, USA*, pages 476–481, June 2005.
- [2] Y. Amir, Y. Kim, C. Nita-Rotaru, J. L. Schultz, J. Stan, and G. Tsudik. Secure group communication using robust contributory key agreement. *IEEE Trans. Parallel and Distributed Systems*, 15(5):468–480, 2004.
- [3] N. Asokan and P. Ginzboorg. Key agreement in ad-hoc networks. In *Computer Communications*, volume 23, pages 1627–1637, 2000.
- [4] E. Bach and J. Shallit. Algorithmic number theory, volume I: Efficient algorithms. *The MIT Press*, 1996.
- [5] R. Balachandran, B. Ramamurthy, X. Zou, and N. Vinodchandran. CRTDH: An efficient key agreement scheme for secure group communications in wireless ad hoc networks. *Proceedings of IEEE International Conference on Communications (ICC)*, pages 1123–1127, 2005.
- [6] S. Banerjee and B. Bhattacharjee. Scalable secure group communication over IP multicast. *IEEE Journal on Selected Areas in Communications*, 20(8):1151–1527, 2002.
- [7] S. Basagni, K. Herrin, E. Rosti, and D. Bruschi. Secure pebblenets. In *In Proc. of ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)*, 2001.
- [8] C. Becker and U. Wille. Communication complexity of group key distribution. In *In Proc. of 5th ACM Conference on Computer and Communication Security*, 1998.
- [9] B. Briscoe. MARKS: Multicast key management using arbitrarily revealed key sequences. *Proceedings of 1st International Workshop on Networked Group Communication*, 1999.
- [10] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. In *In Advances in Cryptology - EUROCRYPT*, 1994.
- [11] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast security: a taxonomy and some efficient constructions. *Proceedings of INFOCOM'99: Conference on Computer Communications*, 2:708–716, Mar. 1999.

- [12] G. Caronni, K. Waldvogel, D. Sun, and B. Plattner. Efficient security for large and dynamic multicast groups. *Proceedings of the Seventh IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '98) (Cat. No.98TB100253)*, pages 376–383, June 1998.
- [13] L. R. Dondeti, S. Mukherjee, and A. Samal. A dual encryption protocol for scalable secure multicasting. *In Fourth IEEE Symposium on Computers and Communications*, pages 2–8, July 1999.
- [14] L. R. Dondeti, S. Mukherjee, and A. Samal. DISEC: a distributed framework for scalable secure many-to-many communication. *In Proceedings of Fifth IEEE Symposium on Computers and Communications (ISCC 2000)*, pages 693–698, July 2000.
- [15] M. Hietalahti. Efficient key agreement for ad-hoc networks. Master's thesis, Helsinki University of Technology, 2001.
- [16] J.-H. Huang and S. Mishra. Mykil: a highly scalable key distribution protocol for large group multicast. *IEEE Global Telecommunications Conference, (GLOBECOM)*, 3:1476–1480, 2003.
- [17] I. Ingemarsson, D. Tang, and C. Wong. A conference key distribution system. *IEEE Transactions on Information Theory*, 28(5):714–720, Sept. 1982.
- [18] Y. Kim, A. Perrig, and G. Tsudik. Tree-based group key agreement. *ACM Transactions on Information Systems Security*, 7(1):60–96, Feb. 2004.
- [19] P. Lee, J. Lui, and D. Yau. Distributed collaborative key agreement protocols for dynamic peer groups. *Proc. IEEE Int'l Conf. Network Protocols (ICNP)*, pages 322–333, Nov. 2002.
- [20] X. Li, Y. Wang, and O. Frieder. Efficient hybrid key agreement protocol for wireless ad-hoc networks. *In IEEE 11th International Conference on Computer, Communication and Networks*, pages 404–409, 2002.
- [21] X. Li, Y. Yang, M. Gouda, and S. Lam. Batch rekeying for secure group communications. *Proc. 10th Int'l WWW Conf.*, pages 525–534, May 2001.
- [22] S. Mittra. Iolus: A framework for scalable secure multicasting. *Journal of Computer Communication Reviews*, 27(4):277–288, 1997.
- [23] W. H. D. Ng, M. Howarth, Z. Sun, and H. Cruickshank. Dynamic balanced key tree management for secure multicast communications. *IEEE Transactions on Computers*, 56(5):577–589, May 2007.
- [24] G. Noubir. Multicast security. *European Space Agency, Project: Performance Optimization of Internet Protocol Via Satellite*, Apr. 1998.
- [25] J. Pegueroles and F. Rico-Novella. Balanced batch lkh: New proposal, implementation and performance evaluation. *Proc. IEEE Symp. Computers and Comm. (ISCC)*, pages 815–820, June 2003.
- [26] R. Pietro, L. Mancini, and S. Jajodia. Efficient and secure keys management for wireless mobile communications. *In Proceedings of the second ACM international workshop on Principles of mobile computing*, pages 66–73, 2002.
- [27] S. Rafaeli and D. Hutchison. Hydra: A decentralized group key management. *Proceedings of 11th IEEE International WETICE: Enterprise Security Workshop*, 2002.
- [28] S. Rafaeli and D. Hutchison. A survey of key management for secure group communication. *ACM Computing Surveys*, 35(3):309–329, 2003.
- [29] S. Setia, S. Koussih, and S. Jajodia. Kronos: A scalable group re-keying approach for secure multicast. *Proceedings of IEEE Symposium on Security and Privacy*, 2000.
- [30] A. T. Sherman and D. A. McGrew. Key establishment in large dynamic groups using one-way function trees. *IEEE transactions on Software Engineering*, 29(5):444–458, May 2003.
- [31] M. Steiner, G. Tsudik, and M. Waidner. Key agreement in dynamic peer groups. *IEEE Transactions on Parallel and Distributed Systems*, 11(8):769–780, Aug. 2000.
- [32] A. Wadaa, S. Olariu, and L. Wilson. Scalable cryptographic key management in wireless sensor networks. *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops*, pages 796–802, 2004.
- [33] D. Wallner, E. Harder, and R. Agee. Key management for multicast: Issues and architectures. *Internet Draft (work in progress), draft-wallner-key-arch-01.txt, Internet Eng. Task Force*, Sept. 1998.
- [34] C. K. Wong, M. Gouda, and S. S. Lam. Secure group communications using key graphs. *SIGCOMM '98, Also University of Texas at Austin, Computer Science Technical report TR 97-23*, pages 68–79, Dec. 1998.
- [35] B. Wu, J. Wu, E. B. Fernandez, M. Ilyas, and S. Magliveras. Secure and efficient key management in mobile ad hoc networks. *Journal of Network and Computer Applications*, 30(3):937–954, 2007.
- [36] Z. Yu and Y. Guan. A key pre-distribution scheme using deployment knowledge for wireless sensor networks. *Proceedings of the 4th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 261–268, 2005.
- [37] X. B. Zhang, S. S. Lam, D.-Y. Lee, and Y. R. Yang. Protocol design for scalable and reliable group rekeying. *Proceedings SPIE Conference on Scalability and Traffic Control in IP Networks*, pages 87–108, Aug. 2001.
- [38] X. Zou, B. Ramamurthy, and S. S. Magliveras, editors. *Secure Group Communications over Data Networks*. Springer, Norwell, MA, 2004.
- [39] X. Zou, A. Thukral, and B. Ramamurthy. An authenticated key agreement protocol for mobile ad hoc networks. *Lecture Notes in Computer Science (LNCS)*, 4325:509–520, Dec. 2006.