Research Article

Location-Based Distributed Group Key Agreement Scheme for Vehicular Ad Hoc Network

Eric Ke Wang, 1 Yuming Ye, 2 and Xiaofei Xu 3

1 Shenzhen Graduate School, Harbin Institute of Technology, HIT Campus, Shenzhen University Town, Xili, Shenzhen 518055, China
2 Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen Graduate School, Harbin Institute of Technology, HIT Campus, Shenzhen University Town, Xili, Shenzhen 518055, China
3 Harbin Institute of Technology, 92 West Dazhi Street, Nan Gang District, Harbin 150001, China

Correspondence should be addressed to Eric Ke Wang; wk_hit@hitsz.edu.cn

Received 4 October 2013; Revised 21 December 2013; Accepted 15 January 2014; Published 23 February 2014

Academic Editor: Hai Vu

Copyright © 2014 Eric Ke Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of vehicular ad hoc networks (VANETs), security issues are concerned. Among various security measures, secure group key agreement for VANETs attracts more and more attention. Researchers have studied generic group key agreement schemes for a long while; however, most of them are not designed for VANETs. Considering energy-constrained and fast moving features in vehicular ad hoc network, we propose a new location-based distributed group key agreement LDGKA scheme for VANETs. We adopt a hybrid approach in which members in the vehicular ad hoc network form various logics groups in the same region. Within each group, virtual key tree model is employed so that the rekeying operation can be efficiently carried out when members leave or join. It only requires to rebuild $O(\log(n))$ keys when one node leaves or joins. Moreover, we adopt one-way key derivation to rebuild keys in order to reduce communication overhead of distributing new keys. We also design a protocol for establishing the secure temporary channel dynamically for the nodes that are located in different regions.

1. Introduction

A vehicular ad hoc network can be regarded as logic groups where the group members are the nodes in the network. Since such a network can be formed in an ad hoc manner, we need an efficient scheme to set up a group key to allow the group members to communicate secretly. For example, if a group of vehicles within a small region decide to share information or chat, they can form a small group network and establish a group key for secure group communication. Others outside the group cannot understand the contents of their discussion even if they can capture every single message of the discussion. Vehicular ad hoc network has the characteristics such that members can freely join in and leave the group; so group key management is a major issue and has been a fundamental block in the scheme. Moreover, the devices in a VANET are usually energy constrained and the bandwidth in the network is commonly limited, so a feasible scheme for VANETs must be of low computation with as fewer communication rounds as possible.

In order to solve these problems, we propose a new scheme of location-based distributed group key agreement for VANETs. In our scheme, no third parties or trusted central authority is required. All members in the network are equal. In the scheme we adopt a tree structure to reduce the computation when any rekeying operation is performed. In addition, this scheme can achieve the goal that given a group, any subset of members can form a private subgroup for secure communication. Moreover, members are able to form a new temporary session group if they are initially located in different regions. Besides, it does not require any central entity which could be a single point of failure or may cause the performance bottleneck. The scheme only requires two communication rounds for keys distribution and rekeying operation.

The rest of the paper is organized as follows. Section 2 introduces the related works. The new scheme is described in Section 3. And the performance issues and security characteristics are discussed in Section 4. Finally, we conclude the paper in Section 5.
2. Related Works

Some group key management techniques [1–4] had been proposed which are applied for various group communications in the early time. Lately, with the development of mobile ad hoc networks and wireless sensor networks, researchers found that traditional group key management schemes cannot be directly applied for the mobile ad hoc networks because of the energy-constrained feature in mobile ad hoc networks. Therefore, some group key management schemes customized for mobile ad hoc network have been proposed successively. Kim et al. [5] firstly propose a tree-based Diffie-Hellman group key agreement (TGDH) for mobile ad hoc networks, which starts to research on how to save the energy when members agree on keys. Later on, researchers proposed several group key agreement schemes that mainly focus on reducing the communication and computation of processing key agreement [6–16].

With the research development on VANETs, the problem of the group key management for VANETs started to be concerning. Yeh et al. [17] proposed a group key management framework based on certificate authority (CA) in VANETs. Since CA and asymmetric cryptography were employed in the scheme, it requires a central control point and the cost is still high for vehicular ad hoc networks. Hao et al. [18] proposed a group key management by corporation messages between nodes; Don applied multivariate symmetric polynomial algorithm to group key agreement for VANETs, while most proposed group key management schemes for VANETs are from the aspects of security algorithms. Considering the special features of VANETs such as fast moving characteristic, that old members leave or new nodes join, we need to rethink the group key management protocols for VANETs.

3. Location-Based Distributed Group Key Agreement (LDGKA)

3.1. Assumptions. In the scheme, we assume that vehicular nodes can be classified in different logical groups according to their location distribution, and virtual binary key trees are built among the members in each group initially. Members belonging to the same logical group are close to each other in physical location. Each member has an unforgeable pair of public and private keys and a certificate of the public key, and subsets of nodes share the internal keys. Leaf keys are members’ public keys; each member contains the key tree data. The nodes whose positions are most left are selected as the supernodes in the groups; if any supernode leaves, the second leftmost sensor becomes the supernode automatically.

3.2. Group Key Initialization and Distribution. According to our observation, we found that the vehicular nodes in the same physical region can probably be neighbors. Hence, we propose a location-based distributed group key agreement scheme, which enables the members in the same physical region to establish group keys. The notations in the rest of the paper can be referenced in Table 1.

Suppose that members \{d_1, d_2, \ldots, d_n\} are located in the same region and they want to generate a logic group, then a virtual key tree needs to be built firstly. To build a tree, they select a supernode which creates an order for nodes to add in the tree one by one. The supernode can be the leftmost node in the group. The key tree data in each member is renewed by adding new members one by one according to the tree order. We define that the one who is leftmost in a subtree is responsible for generating a key. With more nodes joining, the number of supernodes becomes larger and the key tree grows bigger.

3.3. Rekeying Operations Using One-Way Key Derivation. We employ light-weight one-way key derivation inside the key trees. One-way key derivation means that each node would generate the succession keys by itself while the supernode has no need to distribute fresh keys to those members; thus, it can reduce communication rounds in rekeying process. Since communication to transfer keys inside the group is unnecessary, the communication overhead of rekeying distribution can be decreased.

When groups are initialized, supernodes are responsible for distributing a nonzero salt value SK to the groups’ members. When a new node \(j\) wants to join a group, it would send a request \{\text{R}_j, \text{Signature}_{\text{R}_j}, \text{Cert}_{\text{pk}_j}, \text{pk}_j\} to the supernode. When the supernode receives the request, it checks the request by verifying certificate firstly and secondly verifying digital signature with the public key and identifies the new member and points the position where the new user joins.

Suppose an outside node joins the network, then it is inserted into the key tree. Moreover, all secondary keys along the path from it to the root should be refreshed. The process of generating these new keys is shown in Figure 1. New secondary key \(k'\) is computed by (1), where \(k\) is the old key, as follows:

\[
k' = f(k \oplus SK).
\]

When members of a group request to leave, they send request messages with digital signature \{\text{R}_j, \text{Signature}_{\text{R}_j}, \text{Cert}_{\text{pk}_j}, \text{pk}_j\} to supernode. The supernode verifies the message by signature verification. If it is true, it notifies all the other members to delete the leaving nodes in their own tree structure. And the supernode also needs to send a new

| Notation | Description |
|----------|-------------|
| \(G_x\) | Group \(x\) |
| \(S_x\) | The supernode of group \(x\) |
| \(k_{i,j}\) | The group key of subgroup \{node \(i\) to node \(j\)\} |
| \(pk_i\) | The public key of node \(i\) |
| \(\text{Cert}_{pk_i}\) | The certificate of \(pk_i\) |
| \(\{a_i, a_j\}_{\text{CA}}\) | Nodes \(a_i, a_j\) which belong to the Group A |
| \(\text{M}, \text{FD}, \text{G}_x\) | \(M\): message; \(\text{FD}\): the node ID of final destination; \(\text{G}_x\): the group of final destination |
| \(k_j\) | The session key of node \(i\), node \(j\) |
| \(N\) | Group size |
| \(M\) | Number of groups |
nonzero salt value $SK'$ to all the other members. Then the secondary keys should be updated using (2). Figure 2 shows the operation when a single node leaves the group:

$$k' = f(k \oplus SK').$$

(2)

3.4. Path Pairwise Key Establishment Protocol. We propose a path pairwise key establishment protocol (Algorithm 1) to establish the session keys between vehicular nodes located in different logic groups. Suppose that $S_x$ is the supernode of Group $G_x$ and $S_y$ is the supernode of Group $G_y$, as shown in Figure 3. If $x_i$ wants to communicate with $y_j$, they would go as the following steps.

(1) $x_i$ requests the public key of $y_j$ from $S_x$. Then $S_x$ forwards this request to $S_y$; $S_y$ resends the public key $pk_{yj}$ to $S_x$; $S_x$ forwards this public key to $x_i$. 

Figure 1: Rekeying operations when node joins.

Figure 2: Rekeying operations when single user leaves.
Algorithm (Path Pairwise Key Establishment Protocol):
If \((y_j, x_i)\) are not in the same group
  \(x_i\) requests the public key of \(y_j\) from the \(G_y\);
  If (Get public key \(pk_{y_j}\))
  \(x_i\) creates a secret key \(k_{ij}\) and sends \(pk_{y_j}(k_{ij})\) to \(S_y\);
  \(S_x\) forwards it to \(S_y\);
  \(S_y\) forwards it to \(y_j\);
  \(y_j\) decrypts it and get the session key.

**Figure 3:** (a) physical graph and (b) key tree model.

Step 2. \(a_i\) requests the public keys of \({\{b_i, b_j\}_{G_y}, \{c_i\}_{G_y}, \{d_i\}_{G_y}}\) and receives the public keys \(pk_{b_i}, pk_{b_j}, pk_{c_i}, \text{and } pk_{d_i}\).

Step 3. \(a_i\) generates a temporary session group key \(k_{temp}\), and \(a_i\) encrypts \(k_{temp}\) with the above individual public keys. For \(a_i\) and \(a_j\) only needs to use the existing group key \(k_{ij}\) (which is shared by \(a_i\) and \(a_j\)) to encrypt it: \(pk_{a_i}(k_{temp}), pk_{a_j}(k_{temp}), pk_{c_i}(k_{temp}), pk_{d_i}(k_{temp}), k_{ij}(k_{temp})\).

Step 4. \(a_i\) multicasts the set \(\{pk_{b_i}(k_{temp}), pk_{b_j}(k_{temp}), pk_{c_i}(k_{temp}), pk_{d_i}(k_{temp}), k_{ij}(k_{temp})\}\) with destination ID: \(\{pk_{b_i}(k_{temp}), b_i, G_{N_i}, b_j, G_{N_j}, c_i, G_{C_i}, d_i, G_{D_i}\}\).

Step 5. \(a_j\) receives it and decrypts one part of the set and gets \(k_{temp}\) with \(k_{ij}\); the rest of nodes receive it with the forwarding process of their individual supernodes. And they decrypt the corresponding parts and get \(k_{temp}\). The temporary group session key is distributed completely.

4. Security Analysis

4.1. Group Confidentiality. In secure group communication scheme, any node outside of the group is not expected to be able to decrypt the messages transmitted in group. In our scheme, all the messages in the group are encrypted by the session keys established in the group key agreement process. Because the group keys are real-time created according to instant group communication requirement, outsiders cannot retrieve any group session keys without any insider knowledge. Without the group keys, the adversary cannot decrypt the messages in the group communication. While since the group members are mobile and the group members may change with the time elapsing, the group confidentiality also depends on the secrecy of dynamic group keys that should guarantee the forward secrecy and backward secrecy, which is described as follows.

4.2. Authentication of Joining. All new nodes that request to join one group have to be verified by authentication process. The authentication is executed by verification of digital signature with the public keys of new nodes which is done by supernode. Thus supernode can assure whether the new one is its claiming one.

4.3. Forward Secrecy. When an old node leaves, it cannot be able to get the group key after it leaves, since new nonzero salt value \(SK'\) is generated and sent to the rest of the nodes of the group after it leaves. Without the knowledge of \(SK'\), the
Algorithm: (Cross-Group Key Agreement Protocol, \(a_i\), initialize the protocol)

If not \((a_j, b_i, b_j, c_i, d_i)\) are in the same group

\(a_i\) requests the public key of other nodes \([b_i, b_j, c_i, d_i]\) outside of its group;

If (Get public keys \(\{pk_{b_i}, pk_{b_j}, pk_{c_i}, pk_{d_i}\}\))

\(a_i\) creates a temporary session key \(k_{\text{temp}}\) and multicasts \(pk_{b_i}(k_{\text{temp}})\), \(pk_{b_j}(k_{\text{temp}})\), \(pk_{c_i}(k_{\text{temp}})\), \(pk_{d_i}(k_{\text{temp}})\), \(k_{ij}(k_{\text{temp}})\);

\(a_i\) decrypts it and get the session key \(k_{\text{temp}}\);

\(S_A\) forwards it to \(S_B, S_C, S_D\);

\(b_i, b_j, c_i, d_i\) decrypt it and retrieve the key \(k_{\text{temp}}\).

### Algorithm 2

![Algorithm 2](image)

**Figure 4:** (a) physical graph and (b) key tree model.

leaving old node is not able to retrieve the next keys of the group since the \(SK^t\) is random and completely different from \(SK\).

#### 4.4. Backward Secrecy

A new node is not able to retrieve the current keys used in the group before it joins. That means the new node cannot decrypt the messages transmitted in the group before it joins. Because after any new node joins, the internal keys of the key tree would be refreshed and recomputed by one-way key derivation algorithm. Therefore, after new node joins, it hardly retrieves the former keys used in the group based on current key information. The backward secrecy is ensured.

#### 4.5. Secure keys in the Transmission

Since keys are the crucial data which should be kept safe at all time, we employ self-key derivation and public cryptography approaches. For inside group, the session keys do not need to be transmitted in the group. Thus the eavesdropping on the transmission has no use. For cross-groups, the path keys and cross-group keys are transmitted by encryption of public cryptography. Without private keys, the outsider can hardly get the path keys and cross-group keys.

#### 4.6. Voting Approach to Solve Compromised Nodes

A sensor node inside certain group may be compromised or out of function. When an adversary has compromised certain sensor node(s), he may not launch direct attacks against the group communication immediately since once the misbehavior is detected, the supernode may abandon these compromised nodes. Instead, the adversary may let those compromised nodes behave normally but report false data to others in the group. The purpose of the attacker is to mislead others with falsified data. This kind of attacks may lead to many serious consequences; for instance, a false report for the traffic data may lead to traffic jam or even car accident. Thus, it is an important issue to detect these compromised nodes in spite of such problems.

In our scheme, we propose a voting approach to detect the compromised nodes. Since the compromised nodes probably report false or strange data to others, the sensors inside group can give the proper reputations of their neighbors based on
common data patterns by checking validity of the neighbors’ talking data. All the voting messages are within other normal group communication messages; thus, it would incur only little extra overhead without increasing extra communications.

As the first step toward the solution to the problem, we model it as a weight-based network. Each node can vote on its neighbor nodes as the value 1 or 0. These votes can be delivered to the supernode. The supernode collects all data (including the votes for each node in the network) provided by sensor nodes and calculates an aggregation result using the votes for each node inside the group as follows (3):

$$R_j = \frac{\sum_{i=0}^{n} \text{vote}_{i,j}}{n}. \tag{3}$$

$\text{Vote}_{i,j}$ is the vote for the current node $j$ from neighbor node $i$. $R_j$ is the result aggregated by the supernode. $n$ is the number of nodes who vote for the node $j$. If the $R_j$ is lower than one threshold number (in our approach, we set the threshold number 0.5 based on common sense), the node $j$ would be considered malicious.

5. Performance Analysis

In this section, we analyze the performance of our protocol based on computation and communication overhead. Because we adopt virtual binary key tree as group model, when new nodes join or old nodes leave, it only needs to rebuild $\log_2(n)$ keys, where $n$ is the group size. When two nodes decide to establish a pairwise key, they only need three rounds of communication and one round encryption and one round decryption.

Let $N$ be the total number of nodes involved and let $m$ be the number of LDGKA groups. We compare our scheme LDGKA with the pure DH group key agreement scheme GDH and popular tree-based DH group key agreement scheme TGDH. We can see that when nodes join, LDGKA only requires one round of communication while TGDH requires two rounds and GDH requires $N$ rounds. When nodes leave, LDGKA does not need any additional communication by using one-way key derivation while GDH requires $N$ rounds of communication and TGDH requires one round of communication. Tables 2, 3, and 4 show the number of communication rounds and the computation for the operations of setup, join, and leave in the three schemes. Table 5 shows the number of keys, that is, the memory to be stored for each scheme. We plot the graph for the communication rounds and computations required by the three protocols for the setup operation with different number of nodes (see Figures 5 and 6).

From the analysis and comparison of above tables and graphs, we can conclude that LDGKA requires the fewest communication rounds and the least amount of memories to store keys. In the setup operation, LDGKA requires more exponentiations than TGDH; however, in the join or leave operations, LDGKA does not require any exponentiations at all, while the others still require exponentiations to complete the operations. LDGKA adopts one-way functions (hash) to do the key derivation, and computational overhead of hash operations is far less than exponentiations. So we can see for the join or leave operations, LDGKA costs less in computation.
Table 3: Comparison of the join operation.

| Communication rounds | Computations                  |
|----------------------|-------------------------------|
| Pure GDH             | $N$                           |
| TGDH                 | $2$                           |
| LDGKA                | $1$                           |

Table 4: Comparison of the leave operation.

| Communication rounds | Computations                  |
|----------------------|-------------------------------|
| Pure GDH             | $N$                           |
| TGDH                 | $1$                           |
| LDGKA                | $0$                           |

Table 5: Comparison of average memory.

| Keys | Public keys |
|------|-------------|
| Pure GDH | $N-1$         |
| TGDH     | $\log_2 N + 1$ |
| LDGKA    | $\log_2 N - \log_2 m + 1$ |

6. Conclusion

We propose a new group key agreement for vehicular ad hoc networks (LDGKA). The proposed protocol handles several critical operations in VANETs such as new nodes joining the group, old nodes leaving the group, and nodes in different regions forming a dynamic group with low computation and communication overhead.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by National Natural Science Foundation of China (no. 61100192) and Research Fund for the Doctoral Program of Higher Education of China (no. 20112302120074), and it was partially supported by Shenzhen Strategic Emerging Industry Development Foundation (no. JCYJ2012061315032592 and no. ZDSY2012061325016389), National Key Technology R&D Program of MOST China under Grant no. 2012BAK17B08, and National Commonweal Technology R&D Program of AQSIQ China under Grant no. 201310087. The authors thank the reviewers for their comments.

References

[1] G. Ateniese, M. Steiner, and G. Tsudik, “New multiparty authentication services and key agreement protocols,” IEEE Journal of Selected Areas in Communications, vol. 18, no. 4, pp. 1–13, 2000.

[2] M. Steiner, G. Tsudik, and M. Waidner, “Key agreement in dynamic peer groups,” IEEE Transactions on Parallel and Distributed Systems, vol. 11, no. 8, pp. 769–780, 2000.

[3] H. Khurana, R. Bonilla, A. Slagell, R. Afandi, H.-S. Hahm, and J. Basney, “Scalable group key management with partially trusted controllers,” in Proceedings of International Conference on Networking (ICN ’05), pp. 662–672, April 2005.

[4] M. Li, Z. Feng, N. Zang, R. L. Graham, and F. F. Yao, “Approximately optimal trees for group key management with batch updates,” Theoretical Computer Science, vol. 410, no. 11, pp. 1013–1021, 2009.

[5] Y. Kim, A. Perrig, and G. Tsudik, “Tree-based group key agreement,” ACM Transactions on Information and System Security, vol. 7, no. 1, pp. 60–96, 2004.

[6] J. C. M. Teo and C. H. Tan, “Energy-efficient and scalable group key agreement for large ad hoc networks,” in Proceedings of the ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN ’05), pp. 114–121, October 2005.

[7] P. Adusumilli and X. Zou, “DGKD: distributed group key distribution with authentication capability,” in Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC ’05), 2005.

[8] L. Zhou, J. Ni, and C. V. Ravishankar, “Efficient key establishment for group-based wireless sensor deployments,” in Proceedings of the ACM Workshop on Wireless Security (WiSe ’05), pp. 1–10, September 2005.

[9] A. Abdel-Hafez, A. MRI, and L. Ortonzo-Barbosa, “Authenticated group key agreement protocols for Ad Hoc wireless networks,” International Journal of Network Security, vol. 4, no. 1, pp. 90–98, 2007.

[10] J. C. M. Teo and C. H. Tan, “Denial-of-service resilience password-based group key agreement for wireless networks,” in Proceedings of the 3rd ACM Workshop on QoS and Security for Wireless and Mobile Networks (Q2SWinet ’07), pp. 136–143, October 2007.

[11] P. Szczewiak, L. Oliveira, M. Scott, M. Collier, R. Dahab, and E. C. C. Nano, “Testing the limits of elliptic curve cryptography in sensor networks,” in Proceedings of the 5th European Conference on Wireless Sensor Networks (EWSN ’08), vol. 4913 of Lecture Notes in Computer Science, pp. 305–320, Springer, 2008.

[12] D. Galindo, R. Roman, and J. Lopez, “A killer application for pairings: authenticated key establishment in underwater wireless sensor networks,” in Proceedings of the 7th International Conference on Cryptology and Network Security (CANS ’08), vol. 5339 of Lecture Notes in Computer Science, pp. 120–132, Springer, 2008.

[13] L. Zhang, Y. Wang, and G.-L. Li, “A novel group key agreement protocol for wireless sensor networks,” in Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP ’09), November 2009.

[14] M. Li, Z. Feng, N. Zang, R. L. Graham, and F. F. Yao, “Approximately optimal trees for group key management with batch updates,” Theoretical Computer Science, vol. 410, no. 11, pp. 1013–1021, 2009.

[15] R. H. Rahman and N. Nowshee, “A new group key management protocol for wireless Ad-Hoc networks,” World Academy of Science, Engineering and Technology, vol. 75, 2011.

[16] B. Wu and Y. Dong, “A simple group key management approach for mobile Ad Hoc networks,” in Proceedings of the 5th IEEE International Conference on Networking, Architecture and Storage (NAS ’10), pp. 73–78, Macau, China, July 2010.
[17] C.-H. Yeh, M.-Y. Hsieh, and K.-C. Li, “A certificate enhanced group key framework for vehicular Ad Hoc networks,” *Ubiquitous Information Technologies and Applications*, vol. 214, pp. 215–222, 2013.

[18] Y. Hao, Y. Cheng, C. Zhou, and W. Song, “A distributed key management framework with cooperative message authentication in VANETs,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 616–629, 2011.
