Performance Analysis of Group Key Establishment Protocols in Ad Hoc Networks

Eric Ricardo Anton and Otto Carlos Muniz Bandeira Duarte
Grupo de Teleinformática e Automação
COPPE/EE – Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
http://www.gta.ufrj.br/

Abstract— **Wireless ad hoc networks possess two main characteristics: absence of infrastructure and the possibility to connect mobile devices. These networks are however very fragile regarding security. This work focuses on the establishment of a secret cryptographic key among a group of devices that are part of an ad hoc network. Amongst the analyzed protocols for group key establishment, two present superior performance. The results indicate that mobility favors the cases of successful group key establishment and that scenarios with group movement demand fewer messages, although they demand higher density in order to guarantee device connectivity.**

I. Introduction

Wireless ad hoc networks are wireless networks with no infrastructure were the devices have to provide themselves the necessary means of communication. Deployment of this kind of network is becoming more common mainly because of its ease of installation and flexibility. However, wireless ad hoc networks are still very insecure since existing security mechanisms are not appropriate for use in this kind of network. Most of the research on security on wireless ad hoc networks focuses on secure routing [1], [2], [3]. However, these studies assume that a secret cryptographic key is shared among the devices that constitute the ad hoc network.

Some work has been done on developing group key establishment protocols [4], [5], [6], [7], [8], [9], [10], [11]. However, these protocols are generic, not being specified for a specific kind of network. Little work has been done on adapting or evaluating the use of these protocols over wireless ad hoc networks. In [12] a fault tolerant version of the Hypercube protocol [6] with authentication support is proposed.

This paper analyzes the performance of two group key establishment protocols over wireless ad hoc networks. The authentication issue is not addressed in this paper since it can be independently deployed.

The remainder of this paper is organized as follows. Section II describes and compares some group key establishment protocols and presents the specifications of the two key establishment protocols evaluated in this paper. Section III evaluates the performance of these protocols for the cases of wireless ad hoc networks in which the devices move independently and move gathered in groups. Section III-A evaluates these protocols for the case of networks in which the devices move in groups. The paper is finalized in Section IV with some final conclusions.

II. Group key Establishment in wireless ad hoc networks

Several protocol have been proposed in order to extend the 2-party Diffie-Hellman [13] key exchange to a group setting. In [14] some of these protocols are presented an analyzed. In [4] a protocol, referred to in this work as ING, that relies on the devices being arranged in a logical ring is proposed. The Hypercube [6] protocol relies on the devices being arranged in a logical hypercube. In [5] a protocol, referred to in this work as BD, in which every device broadcasts two sets of information to all the other devices in such a way that every device is able to compute a common group key is proposed. Two other protocols, IKA.1 and IKA.2 [7], [8], [9], [10], [11], initially collect contributions for the group key from all the group members, one by one, and then broadcasts these contributions for all the group members. These protocols are presented in details in Sections II-A and II-B.

These protocols specify the need for data broadcast. However they were developed with focus in security, not being specified for a specific kind of network. Since the transmission of data by broadcast is a topic still under study, not having been completely solved, even in traditional computer networks. In wireless ad hoc networks there are aggravating circumstances that include higher loss rates and limited network connectivity. Broadcast transmissions are by nature unreliable. The use of broadcast transmissions is more critic in wireless ad hoc networks, due to the presented characteristics, and in security applications, where delivery confirmation is necessary. Although some studies [15], [16] have been developed aiming at relieving the problems associated with broadcast transmission in wireless ad hoc networks, no definite solution has been achieved. In security applications over wireless ad hoc networks several unicast transmissions, one for each receiver, must therefore be carried out, instead of only one broadcast transmission.

Table I presents a comparison between the mentioned group key establishment protocols. For each protocol, the number of transmitted messages, the total amount of transmitted data in relation to the key size, the number of performed exponentiation operations and the need for synchronism among the participants when sending messages to other participants is presented. Fig. 1 graphically displays this information.

TABLE I

COMPARISON BETWEEN THE PROTOCOLS PRESENTED FOR GROUP KEY ESTABLISHMENT.

	Messages	Transmitted data	Exp. operations	Sync.
ING	$n^2 - n$	$n^2 - n$	n^2	Yes
BD	$2n^2 - 2n$	$2n^2 - 2n$	$n^2 - n$	Yes
Hyp.	$n \cdot \log_2 n$	$n \cdot \log_2 n$	$2n \cdot \log_2 n$	Yes
IKA.1	$2n - 2$	$\frac{1}{2}n^2 + \frac{3}{2}n - 3$	$\frac{1}{2}n^2 + \frac{3}{2}n - 1$	No
IKA.2	$4n - 5$	$4n - 5$	$5n - 6$	No

The ING and BD protocols are the most costly regarding the three metrics, which make them very unappealing to use when compared to the others. The IKA.1 and IKA.2 protocols present some advantages with respect to the Hypercube protocol that consist on the possibility of establishing the contribution sequence during the protocol execution and the possibility of member addition and exclusion without the need to completely execute the key establishment protocol. Moreover, the Hypercube protocol demands synchronism among the devices. This way, a protocol round can only be initiated after the previous round has been completed. This leads to the need of a synchronizer mechanism that guarantees all the participating devices at every round the information of that all the other participating devices have received all the messages intended to them. The use of this kind of mechanism increases the execution time and the number of transmitted messages that varies with respect to the complexity of the synchronizing mechanism.

Therefore, the IKA.2 protocol has the best overall performance regarding the amount of transmitted data and the number of exponential operations. Although the IKA.1 protocol is not as efficient regarding these metrics, its use as an alternative to the IKA.2 protocol may be interesting since it is more efficient regarding the number of messages and both protocols belong to the same protocol suite, sharing the cryptographic key form. This way, eventual group key changes can take place by either protocol.

A. IKA.1

The IKA.1 protocol is executed in two stages:

1) $M_i \Rightarrow M_{i+1}$:
   - $\{g^{\frac{S_1 \cdot S_2 \cdot \ldots \cdot S_i}{S_k}} \mid k \in [1, i]\}, \; g^{S_1 \cdot S_2 \cdot \ldots \cdot S_i}$
2) $M_n \Rightarrow M_i$:
   - $\{g^{\frac{S_1 \cdot S_2 \cdot \ldots \cdot S_n}{S_i}} \mid i \in [1, n-1]\}$,

for $i \in [1, n-1]$, where n is the number of participants (group members), g is the exponentiation base, M_i is the i-th group member, S_i is the random secret exponent generated by M_i and K_n is the group key shared by the n members. All opperations are executed modulo n, which leads to constant number sizes.

In the first stage contributions from all the group members are collected though $n-1$ rounds. At each round each group member (except the first one) receives a data set composed of several intermediate values. Each intermediate value possesses the contribution of every participant that has taken place in the protocol execution, except one. The member then adds its contribution and sends a new data set to the next participant. The last group member, called the group controller, receives a data set containing the contributions of every participant and then calculates the group key $K_n = g^{S_1 \cdot S_2 \cdot \ldots \cdot S_n}$.

In the second stage, the group controller adds its contribution to each of the intermediate values, and distributes them to all the other group members. Each member M_i in order to calculate the group key locates its intermediate value (the one that does not contain its contribution) and adds its contribution S_i, obtaining K_n.

B. IKA.2

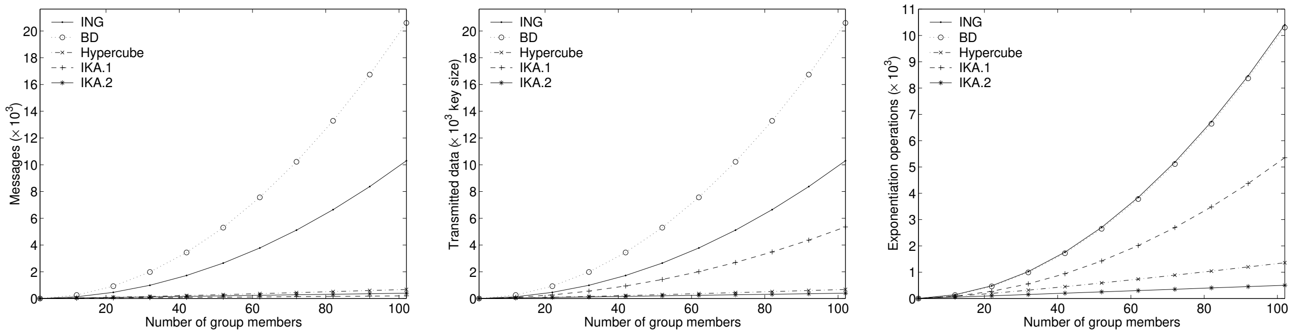
The IKA.1 protocol requires $i+1$ exponential operations to be executed by the i-th group member. In some environments it is desirable to minimize the computational effort required of each group member for the group key establishment, as in groups with high number of participants or formed by devices with limited computing capabilities. In order to minimize this effort, the IKA.2 protocol was proposed. It is similar to the IKA.1 protocol, but is executed in four stages:

1) $M_i \Rightarrow M_{i+1} \; , \; i \in [1, n-2]$:
   - $g^{S_1 \cdot S_2 \cdot \ldots \cdot S_i}$
2) $M_{n-1} \Rightarrow M_i \; , \; i \in [1, n-2]$:
   - $g^{S_1 \cdot S_2 \cdot \ldots \cdot S_{n-1}}$
3) $M_i \Rightarrow M_n \; , \; i \in [1, n-1]$:
   - $g^{\frac{S_1 \cdot S_2 \cdot \ldots \cdot S_{n-1}}{S_i}}$
4) $M_n \Rightarrow M_i \; , \; i \in [1, n-1]$:
   - $\{g^{\frac{S_1 \cdot S_2 \cdot \ldots \cdot S_n}{S_i}} \mid i \in [1, n-1]\}$.

In the first stage contributions from the $n-2$ first group members are collected by means of constant sized messages sent from one member to the next that accumulates all contributions to the group key. In the second stage M_{n-1} adds its contribution to the received value and broadcasts it to all the $n-2$ first members. In the third stage each member factors out its contribution and forwards the result to M_n. In the last stage, M_n receives messages from all the other group members, raises each of them to its contribution S_n and forwards it back to the sender. This way, each group member is able to calculate the group key K_n.

III. PERFORMANCE ANALYSIS

The analysis presented in Section II overlooks some important factors that influence the protocols performance and even feasibility over wireless ad hoc networks. Some of these factors are packet forwarding, network connectivity, and the medium access and routing protocols. In practical situations, when for example a message is transmitted through a path consisted of four hops, the total amount of transmitted data



(a) Transmitted messages.

(b) Transmitted data.

(c) Exponential operations.

Fig. 1. Group key establishment protocols performance with respect to the number of participants.

is four times the initial amount. If the devices are geographically distant from each other network connectivity becomes a problem and may even make key establishment impossible. Network congestion can take place at the link layer as there can be collisions, contention and random backoffs. Routing protocols may also influence performance as it is necessary to discover routes to the destination before sending packets.

In order to consider these factors, the performance of the IKA.1 and IKA.2 protocols was analyzed by simulation using the ns-2 [17] computer network simulator. The simulated wireless devices use the IEEE 802.11 standard and have a 250 meters transmission range. The DSR (Dynamic Source Routing) protocol is used to provide communication routes among the devices. Several problems were experienced by using TCP (Transmission Control Protocol) over wireless networks as mentioned in other studies [18], [19], [20]. Even though some studies have tried to improve TCP performance over wireless networks [21], [22], no standard has been established. A simple reliable transport protocol based on a stop-and-wait mechanism was therefore developed in order to provide communication among the wireless devices. Error margins were calculated for 95% confidence intervals relative to the measures mean value. These error margins are graphically represented by vertical bars. The maximum simulation time was 1000 seconds. As can be observed from the plotted graphics, this is a reasonable amount of time for the analyzed scenarios, as it is a lot greater than the average key establishment time. If the group key is not established within this time interval, the key establishment is considered unsuccessful. Device movement scenarios were generated based on the Random Waypoint mobility model provided by the "setdest" tool.

Since the use of wireless ad hoc networks is only feasible as long as it is possible for the devices to communicate to each other, network connectivity plays a mayor roll on contributive group key establishment. Network connectivity can be influenced by many factors such as the size of the area where the devices are located, the number of devices that constitute the network, and the devices speed and transmission range. The device density ρ is given by $\rho = d/S$, where d is the number of devices distributed over the simulation area S. There are three approaches for changing the device density: changing the size of the simulation area S for a fixed number d of devices, changing the number d of devices for a fixed area S or changing both parameters simultaneously. Since the last two approaches would make it unlikely to differentiate results due to variations on the number of devices from those due to variations on the device density, the first approach is used in this study. The square simulation area was therefore varied in respect to the number of devices. The densities typically used were $(0.2\ R)^{-2}$, $(0.4\ R)^{-2}$, $(0.6\ R)^{-2}$, $(0.8\ R)^{-2}$ and R^{-2}, where $R = 250$ meters is the devices transmission range.

[As the device density increases network connectivity tends to increase, as the devices get closer to each other.]

In order to analyze the influence of device connectivity over the IKA.1 and IKA.2 protocols, these protocols were initially simulated for scenarios with no mobility. Fig. 2 presents the obtained success rates in respect to the number of group members, for specific device densities. The key exchange success rate is calculated by dividing the number of simulations that successfully established a group key by the total number of simulations executed.

For higher device densities such as $(0.2\ R)^{-2}$ and $(0.4\ R)^{-2}$ the network connectivity is high and the group key establishment is always successful. As the device density decreases, so does the group key establishment success rate, until it becomes impracticable (densities $(0.8\ R)^{-2}$ and R^{-2}). Also, for a given device density, as number of devices increases, the group key establishment success rate diminishes. This is because as the area increases with the increasing number of devices, there is a higher probability that in a larger area there will be at least one device unreachable to the others, hindering the protocol execution.

Fig. 3 shows the total amount of data transmitted at link layer by the group key establishment protocols in relation to the number of group members, for specific device densities. Messages related to routing protocols and ARP (Address Resolution Protocol) requests, for example, are not included. For

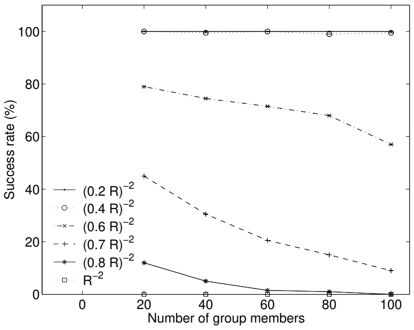
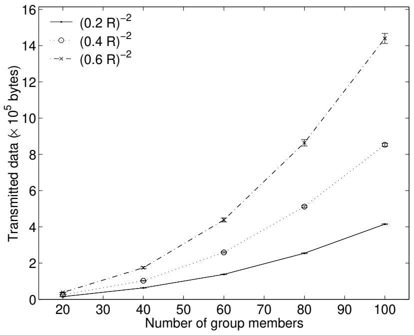
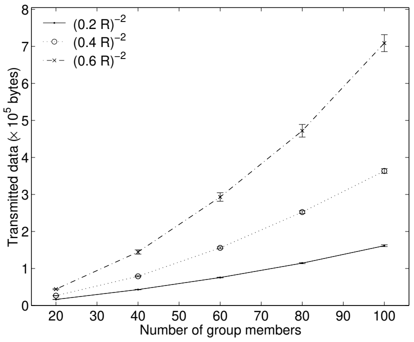


Fig. 2. Success rate for devices with no mobility (IKA.1 and IKA.2).



(a) IKA.1.



(b) IKA.2.

Fig. 3. Transmitted data for devices with no mobility.

a given density value, as the number of group members grows, the amount of transmitted data increases, as expected. For the IKA.2 protocol this growth is more accentuated then the predicted linear growth represented by Table I and Fig. 1(b). This is due to packet forwarding through multiple hops since at each hop new messages are generated and forwarded to the receiver increasing the total amount of data transmitted in the network. Since the obtained success rates for densities greater than $(0.6\ R)^{-2}$ are too low, or even zero, it is not possible to plot other performance graphs for these densities.

As the device density decreases, the total transmitted data increases due to the greater distance between the devices, which tends to increase the number of hops necessary to forward transmitted messages. Comparison between the two protocols shows that the IKA.2 protocol transmits about half the amount of data as the IKA.1 protocol does, unlike the expected value of about one-thirteenth (Table I and Fig. 1(b)). This is due to packet header overhead. For the IKA.2 protocol this overhead is constant since all the messages have constant size. For the IKA.1 protocol as the number of group members increases, so does the messages sizes and lower is the header overhead.

Fig. 4 shows the group key establishment protocols execution times in relation to the number of group members, for specific device densities. As happens to the total amount of transmitted data, the protocols execution time increases as the number of group members increases and the device density decreases. The execution times are similar for both protocols.

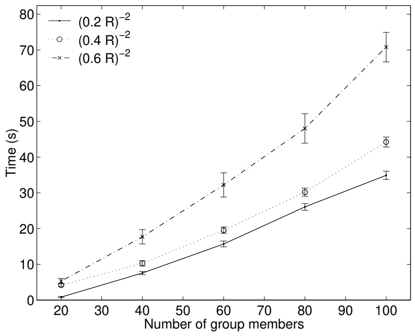
In order to evaluate the influence of mobility over the IKA.1 and IKA.2 protocols they were simulated in scenarios with continuous device movement with speeds around 1 m/s. For a density of $(0.6\ R)^{-2}$ the success rate for both protocols increased to around 100% and for lower densities ($(0.8\ R)^{-2}$ and R^{-2}) better success rates were also obtained. Better performance was therefore obtained regarding the success rates with mobility than with standstill scenarios. Device movement therefore favors network connectivity by changing network topology and increasing the probability that a certain device is capable of eventually communicatin to the others.

The total amount of transmitted data was equivalent for both kinds of scenarios.

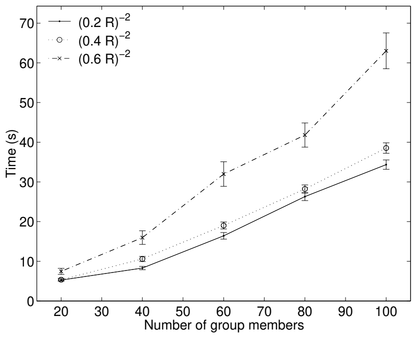
For device densities of $(0.4\ R)^{-2}$ and $(0.6\ R)^{-2}$ the protocol execution time increases with mobility, due to factors like topology changes which leads to execution of route discovery procedures by the routing protocol. For a higher device density ($(0.2\ R)^{-2}$) topology changes are less frequent and the execution times are equivalent to those obtained for scenarios with no mobility.

A. *Performance analysis for devices gathered in subgroups*

The Random Waypoint movement model is considered very generic and distant from real movement situations. This is in part due to the way devices move, completely independently from each other. The "scengen" [23] tool was therefore used to generate scenarios where the devices move in groups. This way, instead of each device moving randomly, groups of devices move randomly. This movement model is more appropriate to represent certain applications as for example a military operation where the groups would be the platoons. In the generated scenarios the groups are composed by 10 devices and the area defined for each group is 150 m × 150 m, so that all of the devices from a same group are close together and can communicate directly. Fig. 5 ilustrates the differences between the two kinds of scenarios for the same number of devices and the same area. The term "subgroup" is used in this paper to refer to each group of devices thar move together in order to avoid confusion with the group of devices that wish

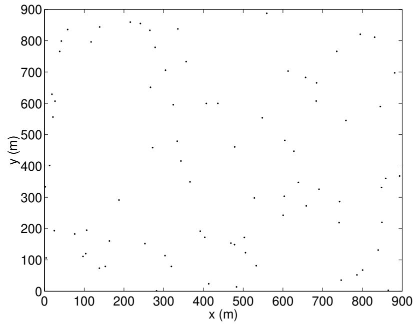


(a) IKA.1.

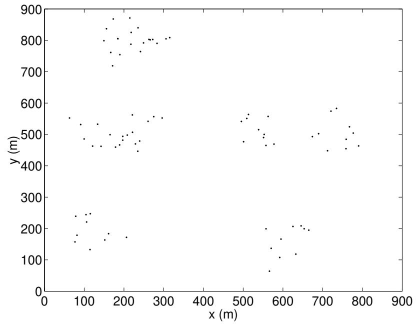


(b) IKA.2.

Fig. 4. Execution time for standstill devices.



(a) Without subgroups.



(b) With subgroups.

Fig. 5. Examples of two scenarios, with and without subgroups (80 devices, density of $(0.4\ R^{-2})$.

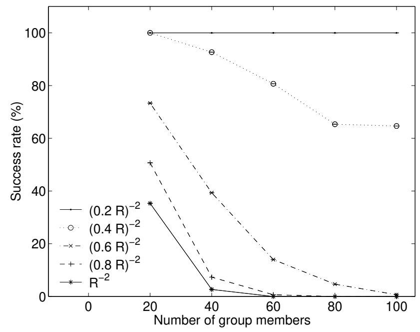


Fig. 6. Success rate for standstill devices in subgroups (IKA.1 e IKA.2).

to establish a secret group key.

The IKA.1 and IKA.2 protocols were initially simulated for scenarios with no mobility. The obtained success rates, presented in Fig. 6, were lower than those obtained for scenarios with independent device movement. This worse performance is due to device concentration, which leads to low connectivity among the subgroups, as can be observed in Fig. 5.

The total amounts of transmitted data for this kind of scenario were lower for both protocols than those obtained for scenarios with no subgroups. This reduction is due to the proximity among the members of a same subgroup. Since all devices that belong to a same subgroup are adjacent regarding the sequence in which the group key contributions take place, messages transmitted during the protocols execution are initially restricted to a subgroup, and are therefore transmitted with no connectivity problems and always by one hop, before of they are forwarded to the next subgroup.

Device gathering in subgroups leads to a reduction on the data transmitted by the IKA.1 protocol of about 35% for a density of $(0.2\ R)^{-2}$ and about 50% for a density of $(0.4\ R)^{-2}$. Therefore this kind of scenario has greater influence on scenarios with lower device densities by diminishing the average number of hops and consequently in the amount of transmitted data.

For the IKA.2 protocol this reduction is of about 25% for a density of $(0.2\ R)^{-2}$ and of about 15% for a density of $(0.4\ R)^{-2}$. Subgroup scenarios therefore play a minor roll reducing transmitted data as device density diminishes. This is due to the fact that subgroups diminish the average number of hops necessary to transmit messages on the first round of both the protocols, as explained before. For the following rounds (round 2 of the IKA.1 protocol and rounds 2, 3 and 4 of the IKA.2 protocol), this effect of economy is minor because transmissions occur between one (IKA.1) or two (IKA.2) specific devices and all the others, i.e., between a subgroup and all the others. Therefore these rounds do not suffer from such a significant decrease on the average number of hops.

As the IKA.2 protocol transmits about three higher times the number of messages that the IKA.1 protocol does in these rounds, the decrease of the transmitted data is less significant.

Comparison between the execution time for this kind of scenario and scenarios without subgroups shows that for a density of $(0.2\ R)^{-2}$ the execution time decreases in about 40% for both the protocols. This is again due to one hop communication within each subgroup. For a density of $(0.4\ R)^{-2}$ the execution time is about 20% to 25% higher for scenarios with subgroups, since the time necessary for transmitting messages inside a subgroup is practically constant. This is due to the extra time necessary for communications between subgroups, due to lower connectivity among subgroups,

The IKA.1 and IKA.2 protocols were also simulated for scenarios with subgroups were the devices are in constant movement at speeds around 1 m/s. The obtained success rates show improvement when compared to scenarios with no mobility. For densities of $(0.2\ R)^{-2}$ and $(0.4\ R)^{-2}$ the success rates are around 100% and for a density of $(0.6\ R)^{-2}$ the execution of the protocols became feasible.

Regarding the amount of transmitted data, the performances are equivalent to those obtained with standstill devices gathered in subgroups. This is consistent with the results obtained comparing scenarios with no subgroups.

The observation of the execution time shows that for a density of $(0.2\ R)^{-2}$ the performances are similar to those obtained with the standstill devices and that for lower densities there is an increase on the execution time due to topologies changes, as verified for scenarios without subgroups.

IV. Conclusion

This paper analyzed some protocols for establishment of a group key among the members of a group of devices that form an ad hoc network. The IKA.1 and IKA.2 protocols were selected as the most convenient for execution in this environment because they present good performances regarding the number of transmitted messages, the total amount of transmitted data and the number of exponentiation operations carried out. Also, these protocols have the advantage of not demanding synchronism among the group members and allow for group key refresh without the need to execute the entire protocol from the beginning.

Simulation evaluations were carried out based on the mobility model generally used, that consists of devices randomly distributed throughout the simulation area, and on the introduced model, that consists on devices gathered in subgroups. In both cases the lower the device density the more compromised are the protocols executions due to lower network connectivity. Mobility favors the group key establishment by increasing network connectivity at the cost of higher execution time.

For scenarios based on group movement, higher device densities are needed in order to guarantee good connectivity among the subgroups. In these conditions, the total amount of transmitted data and the execution time diminish due to high connectivity among devices from a same subgroup.

It was also noted that at all evaluated scenarios the IKA.2 protocol presented equivalent or better performance then the IKA.1 protocol.

References

[1] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Ariadne: A secure on-demand routing protocol for ad hoc networks,” in *The Eighth Annual International Conference on Mobile Computing and Networking (MobiCom 2002)*, Sept. 2002.

[2] P. Papadimitratos and Z. J. Haas, “Secure routing for mobile ad hoc networks,” in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, Jan. 2002.

[3] Y.-C. Hu, D. B. Johnson, and A. Perrig, “Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks,” in *Fourth IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[4] I. Ingemarsson, D. T. Tang, and C. K. Wong, “A conference key distribution system,” *IEEE Transactions on Information Theory*, vol. IT-28, no. 5, pp. 714–720, Sept. 1982.

[5] M. Burmester and Y. Desmedt, “A secure and efficient conference key distribution system,” in *Advances in Cryptology (EUROCRYPT ’94)*, May 1994, pp. 275–286.

[6] K. Becker and U. Wille, “Communication complexity of group key distribution,” in *5th ACM conference on Computer and Communication Security*, Nov. 1998.

[7] M. Steiner, G. Tsudik, and M. Waidner, “Diffie-Hellman key distribution extended to group communication,” in *3rd ACM Conference on Computer and Communications Security*, Mar. 1996.

[8] ——, “CLIQUES: A new approach to group key agreement,” in *18th International Conference on Distributed Computing Systems*, May 1998.

[9] G. Ateniese, M. Steiner, and G. Tsudik, “Authenticated group key agreement and friends,” in *5th ACM Conference on Computer and Communications Security*, Nov. 1998.

[10] ——, “New multiparty authentication services and key agreement protocols,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, Apr. 2000.

[11] 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, Aug. 2000.

[12] N. Asokan and P. Ginzboorg, “Key agreement in ad hoc networks,” *Computer Communications*, vol. 23, no. 17, pp. 1627–1637, Nov. 2000.

[13] W. Diffie and M. E. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[14] E. R. Anton and O. C. M. B. Duarte, “Group key establishment in wireless ad hoc networks,” in *Workshop on Quality of Service and Mobility (WQoSM 2002)*, Angra dos Reis, RJ, Brazil, Nov. 2002.

[15] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, Aug. 1999, pp. 151–162.

[16] M.-T. Sun, W. Feng, and T.-H. Lai, “Location aided broadcast in wireless ad hoc networks,” in *IEEE GLOBECOM 2001*, Nov. 2001, pp. 2842–2846.

[17] K. Fall and K. Varadhan, *The ns Manual*, Apr. 2002.

[18] M. Gerla, K. Tang, and R. Bagrodia, “TCP performance in wireless multi-hop networks,” in *2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA’99)*, Feb. 1999.

[19] S. Biaz and N. H. Vaidya, “Distinguishing congestion losses from wireless transmission losses: A negative result,” in *International Conference on Computer Communications and Networks*, Oct. 1998.

[20] G. Holland and N. Vaidya, “Analysis of TCP performance over mobile ad hoc networks,” in *5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, Aug. 1999, pp. 219–230.

[21] C. Parsa and J. J. Garcia-Luna-Aceves, “Improving TCP performance over wireless networks at the link layer,” *ACM Mobile Networks and Applications*, vol. 5, no. 1, pp. 57–71, Mar. 2000.

[22] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A comparison of mechanisms for improving TCP performance over wireless links,” *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756–769, Dec. 1997.

[23] L. Qiming, “The Scenario Generator,” 2001, `http://www.comp.nus.edu.sg/~liqm/scengen/`.