Group-Based Key Management for Multicast of Ad Hoc Sensor Network*

Shaobin Cai, Wenbin Yao, NianminYao, Yong Li, and Guochang Gu

Harbin Engineering University, Harbin, China, 150001
Phone: 86-451-82518030
caishaobin@hrbeu.edu.cn

Abstract. In order to protect communication security among sensor nodes, a cryptographic method are needed by ad hoc sensor networks. According to that most ad hoc sensor networks are deployed in groups, a group-based multicast key management scheme is proposed to extend RPS scheme in this paper. Compared with RPS, our scheme improves the probability that a shared key exists between two sensor nodes, and reduce the probability that the shared key is decrypted.

Keywords: Ad hoc sensor networks, Security, Group-based, Key Management.

1 Introduction

The open architectures of ASNs (ad hoc sensor networks) make potential intruder easy to intercept, eavesdrop and fake messages. Therefore, they need strong security services. Most security methods can be realized by message encryption. Therefore, some kinds of cryptographic keys that need to be shared between the communicating parties are needed.

The pre-loaded key management of ad hoc sensor network, proposed by Blom [1], efficiently solves the key management for broadcast and multicast [2-5] of ad hoc sensor network. N-security r-conference key management scheme [6~10] is a typical pre-loaded key management scheme.

Since ad hoc sensor networks are mostly deployed in groups, the communications among nodes are mostly happened among nodes of same group. Therefore, a group-based multicast key management scheme is proposed in this paper to improve the probability that two nodes share at least one key.

The rest of the paper is organized as follow. First, an overview of n-secure r-conference key management scheme is given in section 2. Secondly, a group-based

* This paper is supported by the following foundation:
(1) National Science foundation of China “60603059” “60403026” “60503055”;
(2) Postdoctoral science foundation of china “2005038193”;
(3) Postdoctoral foundation of Heilongjiang province, china;
(4) Tackling key technical problem of Heilongjiang province “GC06C106”
(5) Science Foundation of Harbin Engineering University “HEUFFT05010”;
(6) Young backbone teacher project of Harbin Engineering University.

Y. Shi et al. (Eds.): ICCS 2007, Part III, LNCS 4489, pp. 50–57, 2007.

key management for multicast of ad hoc sensor network is proposed in section 3. Thirdly, the performance of group-based scheme is analyzed by mathematical method and simulations in section 4. Finally, we draw a conclusion in section 5.

2 RPS Key Management Scheme

RPS [10] (Random Preloaded Subset key distribution) scheme is proposed as an *n*-security *r*-conference key management. RPS determines the public key of each node by a public one way function $F_1()$, which is defined as follow:

$$\left|I_1 \cdots I_k\right| = F_1(A) \tag{1}$$

In the above formula, $1 \le I_1 \cdots I_k \le P$ is a random permutation of numbers between 1 and P. For instance, it could be obtained by choosing the first k elements of a random permutation of numbers between 1 and P. $I_1 \cdots I_k$ is the index of the keys preloaded in node A. By exchanging their IDs, two nodes can immediately determine their shared indices, and use their shared keys to derive their pair key. For an *r*-user conference, the *r* nodes can independently calculate their conference key based on the keys shared by all *r* nodes.

3 Group-Based *n*-Security *r*-Conference Key Management Scheme

In this section, GBKM (Group-Based Key Management) is proposed to extend RPS. In GBKM, keys preloaded in a node are not only tied to its public ID but also to its group ID. Therefore, GBKM not only has a one way public function $F_1()$ but also has a public one way group function $Fm(i)$, which is defined as follow:

$$\left|I_1 \cdots I_z\right| = Fm(i) \tag{2}$$

In the above formula, $1 \le I_1 \cdots I_Z \le P$ is a random permutation of numbers between 1 and P, and it is a subset A_i of A. When a node acquires its pre-loaded keys, it first calculates out the subset for its group by its group function $F_m()$ with its group ID, and then it calculates out its key from the key subset of group by function $F_1()$.

Therefore, in GBKM, any r nodes of same group can determine whether there exists at least a shared key among them after they exchange their group ID and their public ID. Although, the probability that two nodes of same group share at least one key is improved by the method above, the probability that two nodes of different groups share at least one key is still low. In order to improve the probability that two nodes of different groups share at least one key, they broadcast its received group ID and node ID in its group when two nodes of different groups want to setup a session and do not have any shared key. If one node of its group has shared keys, then it forwards the shared key to the node. Therefore, the probability that two nodes of different groups share at least one key is improved and is equal to that two groups share keys.

4 Analysis

4.1 The Probability That Two Nodes Share at Least One Keys

Let p be the probability that two nodes of same group share at least one key. The probability that two key rings share at least one keys is equal to 1 – Pr [two nodes do not share any key]. Therefore, we first compute the number of the possible key rings. Since each key of a key ring is drawn out of a pool of P keys without replacement, the number of possible key rings is $\frac{P!}{k!(P-k)!}$.

After picking out the first key ring, the total number of possible key rings that do not share a key with the first key ring is the number of key rings that can be drawn out of the remaining P - k unused key in the pool, namely $\frac{(P-k)!}{k!(P-2k)!}$.

Therefore, the probability that no key is shared between the two rings is the ratio of the number of rings without a match by the total number of rings. Thus, the probability that there is at least a shared key between two key rings is

$$p = 1 - \frac{k!(P-k)!}{P!} \times \frac{(P-k)!}{k!(P-2k)!} = 1 - \frac{((P-k)!)^2}{(P-2k)!P!} \tag{3}$$

Since P is very large, Stirling's approximation $n! \approx \sqrt{2\pi}\, n^{n+\frac{1}{2}} e^{-n}$ is used to simplify the expression of p, and obtain: $p = 1 - \frac{(1-\frac{k}{P})^{2(P-k+\frac{1}{2})}}{(1-\frac{2k}{P})^{(P-2k+\frac{1}{2})}}$. Since the size of sub-pool is much smaller than that of pool, the probability that a shared key exists between two sensor nodes of the same group is improved by our scheme.

Figure 1 describes the relationship between the size of ring and the probability that two nodes share at least one key. When pool has 10,000 keys and key ring has 75 keys, the probability of Laurent's scheme that two node share at least one key is only about 0.5. However, the probability of our scheme that two nodes share at least one key is almost about 1 when each sub-pool of our scheme has 1,000 keys.

p' is used to present the probability that a shared key exists between two nodes of different groups, whose sub-pools have P' keys. Each group can draw $\frac{P!}{P'!(P-P')!}$ sub-pools from its main pool. Therefore, probability that a shared key exists between two nodes of different groups is

$$p = 1 - \frac{P'!(P-P')!}{P!} \times \frac{(P-P')!}{P'!(P-2P')!} = 1 - \frac{((P-P')!)^2}{(P-2P')!P!} \tag{4}$$

From the results showed in fig. 1, we can know that the probability, which two nodes of different groups have at least one key, is almost 100% when sub-pool has more than 200 keys. Therefore, GBKM can be improved further by reduce the scope of exchanging group ID and node ID when two nodes belong to different groups. In

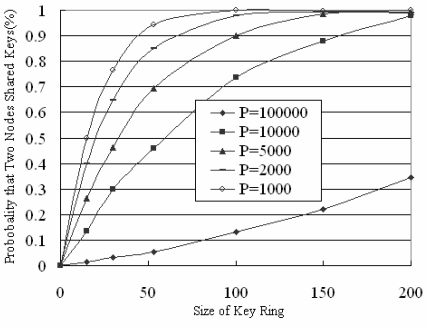

Fig. 1. The relationship between the size of ring and the probability that two nodes share a key

the improved GBMK, the node only exchange group ID and node ID with its neighbors of the same group.

Now, the probability that *r* nodes share at least one key is calculated. If there is at least a shared key among r nodes, then the intersection $A_k^1 \cap \cdots \cap A_k^r$ is not empty. Therefore, $PS_m^{\mathrm{r}} = \Pr\{\left|A_{\mathrm{k}}^1 \cap \cdots \cap A_{\mathrm{k}}^{\mathrm{r}}\right| = m\}$ is first calculated for the probability that $A_k^1 \cap \cdots \cap A_k^r \neq \varnothing$. Therefore, $PS^{\mathrm{r}} \leq (p)^{r-1}$ and *r* nodes hardly have shared keys when p is small. Therefore, it is necessary to improve the probability that two nodes share at least one key.

4.2 Security

In this section, the security of GBKM is analyzed. The communications among *r* nodes are safe only when $\exists a \in A_k^1 \cap \cdots \cap A_k^r$ and $a \notin A_k^{r+1} \cup \cdots \cup A_k^{R+n}$. Therefore, $PE_R(P,k,n,r)$ is used to present the probability that *r*-node communications can be eavesdropped by *n* nodes, and PS_m^r is calculated first to calculate $PE_R(P,k,n,r)$.

The calculation complexity of PS_m^r increases when *r* increases and it can not be calculated when *r* is much larger. Therefore, ϕ_{r} is used here to present expected *m*, namely

$$\phi_{\mathrm{r}} = E[m] = \sum_{m=1}^{k} mPS_m^r = k(\frac{k}{P})^{r-1} \tag{5}$$

Therefore, the larger the $\frac{k}{P}$ is, the higher probability that *r* nodes share at least one key. Fig.2 descries the relationship between $\phi_{\mathrm{r}}(r=5)$ and the size of ring. In fig.2, when $\frac{k}{P}$ is larger, smaller key ring can guarantee *r* nodes share at least one key; when $\frac{k}{P}$ is smaller, larger key ring is even can not guarantee *r* nodes share at least one key. Therefore, it is necessary to improve probability that *r* nodes share at least

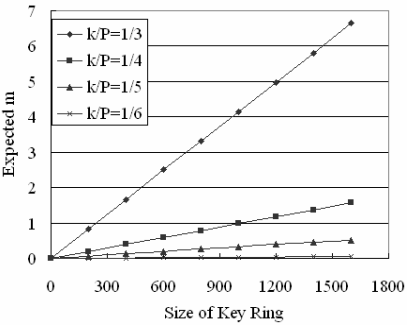

Fig. 2. The relationship between $\phi_r(r=5)$ and the size of ring

one key by the improvement of $\frac{k}{P}$, which is resulted from group-based key management.

Secondly, $PC_{n,k}^{q} = \Pr\{\left|A_k^1 \cup \cdots \cup A_k^n\right| = q\}$ is used here to present the probability that $A_k^1 \cup \cdots \cup A_k^n$ has q keys, where $k \le q \le q_{\max} = \min(nk, P)$.

The calculation complexity of PS_m^r also increases when r increases, and PS_m^r cannot be calculated when r is much larger. $\theta_n = E[q] = \sum_{q=k}^{q_{\max}} qPC_{n,k}^{q}$ is used here to present expected q. The value of θ_n can be calculated by recursion, and θ_n is defined as follow:

$$\theta_n = \theta_{n-1} + \frac{k}{P}(P - \theta_{n-1}), \quad \theta_0 = 0 \tag{6}$$

$\theta'_n = \frac{\theta_n}{P} = \theta'_{n-1} + \frac{k}{P}(1 - \theta'_{n-1})$ is used here to present expected $\frac{q}{P}$. Fig.3 descries the relationship between θ'_n and the number of nodes captured by the adversaries. The larger $\frac{k}{P}$ is, the lower the security of r nodes communications is, and the easier the adversity can acquire all keys by capturing few nodes. The smaller $\frac{k}{P}$ is, the higher the security of r nodes communications is, and the more difficult the adversities acquire all keys by capturing few nodes.

However, from the analysis results in fig3, we know that a higher $\frac{k}{P}$ is needed by a key management scheme to guarantee that there is at least one shared keys among r nodes. When adversity does not know the adscription of the nodes, $P = \left|U_g\right|$ (U_g presents the sub-pool)in equation (5) mostly, and $\frac{k}{P}$ is larger; $P = \left|U_p\right|$ in equation

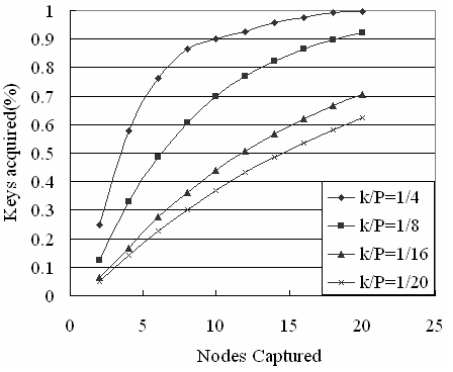

Fig. 3. The relationship between $\theta_{\mathrm{n}}^{'}$ and the number of nodes acquired by adversity

(7), and $\frac{k}{P}$ is smaller. Therefore, group-based key management not only improves the probability that *r* nodes share at least one key, but also reduces the probability that the shared keys are decrypted.

Thirdly, $PE_{\mathrm{m}}^{\mathrm{q}} = \Pr\{A_m \subset A_q\}$ is used here to present the probability that $A_m \subset A_q$, where $A_m \subset U_p$ ($|A_m| = m$) and $A_q \subset U_p$ ($|A_q| = q$). Therefore,

$$PE_m^q = \frac{\binom{P-m}{q-m}}{\binom{P}{q}} = \frac{(P-m)!q!}{(q-m)!P!} \tag{7}$$

$PE_R(P,k,n,r) = \Pr\{G \subset B\}$ is used here to present the probability that $G \subset B$, where $G = A_k^1 \cap \cdots \cap A_k^r$ and $B = A_k^{r+1} \cap \cdots \cap A_k^{n+r}$. Therefore, $PE_R(P,k,n,r)$ can be expressed further as follow

$$PE_R(P,k,n,r) = \sum_{q=k}^{q_{\max}} PC_{n,k}^q \sum_{m=0}^{k} PS_m^r PE_m^q \tag{8}$$

$PC_{n,k}^q$ is hardly calculated when $n \geq 3$. Hence, an approximation of $PE_R(P,k,n,r)$ should be calculated, and equitation (8) can be defined further as follow:

$$PE_R(P,k,n) \approx \hat{P}E_R(P,k,n) = \sum_{m=0}^{k} PS_m^r PE_m^{\phi_n} \tag{9}$$

By the method that the approximation of $PE_R(P,k,n,r)$ is calculated when $n \geq 3$, the approximation of $PE_R(P,k,n,r)$ can be acquired when *r* is larger. Therefore, $PE_R(P,k,n,r)$ can be defined as follow:

$$
\begin{aligned}
& PE_R(P,k,n,r) \approx \hat{P}E_R(P,k,n,r) \\
&= \sum_{q=k}^{q_{\max}} PC_{m,k}^{q} PE_{\phi_r}^{q} \approx \ddot{P}E_R(P,k,n,r) \\
&= PE_{\phi_r}^{\theta_m} = \frac{(P-\phi_r)!\theta_m!}{(\theta_m-\phi_r)!P!}
\end{aligned} \tag{10}
$$

$\phi_r = i$, and $PE_R^i(P,k,n,r)$ is used to present the probability that the r node communications are eavesdropped. Therefore, $PE_R^1(P,k,n,r) = \theta'_m$, $PE_R^2(P,k,n,r) = \theta'_m \frac{(\theta_m - 1)}{(P-1)}$. Hence, the probability that r nodes communications are eavesdropped decreases exponentially when ϕ_r increases because of $1 > \frac{\theta_m}{P} > \frac{(\theta_m - 1)}{(P-1)}$.

From the analysis above, a larger ϕ_r and a smaller θ_m is needed to improve the session security among r nodes. When $\frac{k}{P}$ is larger, ϕ_r is larger and θ_m is smaller. When $\frac{k}{P}$ is smaller, ϕ_r is smaller and θ_m is bigger. Therefore, a key management scheme needs a reasonable $\frac{k}{P}$.

When adversity does not know the adscription of the nodes, $P = |U_g|$ in equation (5) mostly, and $\frac{k}{P}$ is larger; $P = |U_p|$ in equation (7), and $\frac{k}{P}$ is smaller. Therefore, group-based key management not only improves the probability that r nodes share at least one key, but also reduces the probability that the shared keys are decrypted.

5 Conclusions

According to that most ad hoc sensor networks are deployed for assigned missions, a group-based key management scheme is proposed in the paper. GBKM divides key pool into some sub-pool according to the relativities of missions, and the numbers of shared keys among missions are determined by the relativities among missions. In ad hoc sensor network, most communications are happened among nodes of same missions. Therefore, the group-based key management not only improves the probability that communicating nodes share at least one key but also reduce the probability that the shared key is decrypted.

References

1. C. Blundo, A. Santi, A. Herzberg, U. Vaccaro, M. Yung. Perfectly-secure Key Distribution for Dynamic Conferences. In Proc. of Crypto'92, Santa Barbara, California, USA, (1992) 471~486
2. A. Fiat, M. Naor. Broadcast Encryption. In Proc. of Crypto'93, Santa Barbara, California, USA, (1994) 480~491

3. D. Halevy, A. Shamir. The LSD Broadcast Encryption Scheme. In Proc. of Crypto'02, Santa Barbara, California, USA, (2002) 47~60
4. R. Kumar, S. Rajagopalan, A. Sahai. Coding Contractions for Blacklisting Problems without Computational Assumptions. In Proc. of Crypto'99, Santa Barbara, California, USA, (1999) 609~623
5. D. Naor, M. Naor, J. Lotspiech. Revocation and Tracing Schemes for Stateless Receivers. In Proc. of Crypto'01, Santa Barbara, California, USA, (2001) 41~62
6. R. Blom. An Optimal Class of Symmetric Key Generation Systems. In Proc. of Santa Barbara, California, USA, (1984) 335~338
7. T. Matsumoto, H. Imai. On the Key Predistribution System: A Practical Solution to the Key Distribution Problem. In Proc. of Crypto'87, Santa Barbara, California, USA, (1987) 185~193
8. T. Leighton, S. Micali. Secret-key Agreement without Public-Key Cryptography. In Proc. of Crypto'93, Santa Barbara, California, USA, (1993) 456~479
9. M. G. Zapata, N. Asokan. Securing Ad-Hoc Routing Protocols. In Proc. of WiSe'02, Singapore, (2002) 1~10
10. H. Chan, A. Perrig, D. Song. Random Key Predistribution Schemes for Sensor Networks. In Proc. of S&P'03, California, US, (2003) 197~215