Hierarchical Self-healing Key Distribution for Heterogeneous Wireless Sensor Networks

Yanjiang Yang[1], Jianying Zhou[1], Robert H. Deng[2], and Feng Bao[1]

[1] Institute for Infocomm Research, Singapore
{yyang,jyzhou,baofeng}@i2r.a-star.edu.sg
[2] School of Information Systems, Singapore Management University
robertdeng@smu.edu.sg

Abstract. Self-healing group key distribution aims to achieve robust key distribution over lossy channels in wireless sensor networks (WSNs). However, all existing self-healing group key distribution schemes in the literature consider homogenous WSNs which are known to be unscalable. Heterogeneous WSNs have better scalability and performance than homogenous ones. We are thus motivated to study *hierarchial* self-healing group key distribution, tailored to the heterogeneous WSN architecture. In particular, we revisit and adapt Dutta *et al.*'s model to the setting of hierarchical self-healing group key distribution, and propose a concrete scheme that achieves computational security and high efficiency.

Keywords: Wireless sensor network, self-healing group key distribution, wireless sensor network security.

1 Introduction

A wireless sensor network (WSN) consists of a large number of sensor nodes collecting and reporting the environmental data to a base station. A sensor node is a small sensing device capable of wireless communications through radio signals. Due to the low cost requirement, sensor nodes are extremely constrained in hardware, having limited computation capability, storage capacity, and radio transmission range. Worse yet, sensor nodes are usually powered by batteries, hence restricted power supply is yet another major limitation of WSNs.

WSNs are easily susceptible to adversaries who can intercept or interrupt the wireless communications. It is thus crucial to ensure secure communication when WSNs are deployed for mission-critical applications. A fundamental service to achieve secure communication is key distribution, whereby sensor nodes establish (secret) keys. Unfortunately, it is commonly acknowledged that key distribution in WSNs is not trivial, considering the resource-constrained nature of sensor nodes. Hence lots of efforts have been dedicated to the study of key management and distribution in WSNs [3,4,6,5,7,8,9,10,11,12,13,15]. These methods are categorized into group key distribution [3,6,8,10,11] and pairwise key distribution [4,5,7,9,12,15].

Y. Chen et al. (Eds.): SecureComm 2009, LNICST 19, pp. 285–295, 2009.

Among the existing group key distribution schemes, self-healing group key distribution [6,11,13] particularly suits WSNs. A prominent property of this type of group key distribution is *self-healing*, which allows group members to recover lost group keys of previous sessions based solely on the key update message of the current session. This makes group key distribution resilient to lossy wireless communication of WSNs.

All the self-healing group key distribution schemes in the literature considered homogeneous WSNs where all sensor nodes are assumed to be the same. However, homogeneous WSNs are not scalable. We are thus motivated to study self-healing group key distribution in *heterogenous* WSNs. A heterogenous WSN is composed of not only resource constrained sensor nodes, but also a number of more powerful high-end devices. Specifically, a WSN is partitioned into a number of *groups*, and a high-end device is placed into each group, acting as the *group manager*. A group manager is more powerful, and thus does not suffer from the resource scarceness problem as much as a sensor node does.

Our Contributions. Tailored to the heterogeneous WSN architecture, we propose the concept of *hierarchical* self-healing group key distribution. In particular, we formulate a security model for hierarchical self-healing group key distribution by revisiting and adapting Dutta *et al.*'s model [6]. We then propose a concrete scheme, proven secure under the model. Our scheme is "authenticated", compared to Dutta *et al.*'s schemes, in the sense that every non-revoked sensor node can ascertain the validity of the group keys it generated from the key update messages, without involving any extra communication overhead. As communication is more energy consuming than computation in WSNs, this property is important to prevent sensor nodes communicating using invalid group keys.

2 Related Work

Public key cryptosystems are in general too expensive for WSNs, so symmetric key primitives such as secret key encryption or cryptographic hash function are often preferred. As such, key management and distribution in WSNs boils down to sharing of secret keys among sensor nodes. To achieve this objective, a commonly used approach is to pre-load a set of secrets inside sensor nodes before their deployment. These pre-loaded secrets are then used either directly as pair-wise keys between a pair of neighboring sensor nodes, i.e., pair-wise key distribution [4,5,7,9,10,12,15], or as a basis to establish new common keys shared by a group of sensor nodes, i.e., group key distribution [3,6,8,10,11].

Among the existing group key distribution schemes, self-healing group key distribution is particularly suitable for WSNs, because of its self-healing and membership revocation properties. Staddon *et al.* [13] first proposed the concept and a concrete construction of self-healing group key distribution based on secret sharing of two dimensional polynomials. Their construction, however, is not efficient, suffering from high communication and storage overhead. Liu *et al.* [11] then generalized the security notions in [13], and presented a new scheme with better efficiency by combining personal secret distribution with the self-healing

technique of [13]. Blundo *et al.* [1] analyzed the security definitions in [11,13] and concluded that it is impossible for any scheme to achieve all of the security requirements formulated in [11,13]. They then formulated a new definition for self-healing group key distribution and came up with a new scheme [2].

All the above self-healing group key distribution schemes are intended to achieve information theoretic security. In [6], Dutta *et al.* proposed a novel computationally secure scheme, based on a combination of a reverse one-way hash chain and a forward one-way hash chain. Their idea in achieving self-healing is that along the reverse hash chain, the hash value of $h^j(.)$ (associated with an earlier session) can be computed from any pre-image $h^i(.)$ (associated with a later session), where $i < j$ and $h^i(.) = \underbrace{h(h(\cdots h(.)))}_{i \ times}$. While Dutta *et al.*'s model is weaker, their schemes tremendously improve the efficiency of the information theoretically secure schemes. Our proposed scheme is based on Dutta *et al.*'s idea of a combination of reverse and forward one-way hash chains, but we rectify the vulnerability of their construction (it can be shown that their schemes cannot achieve t-Revocation). The main differences between our scheme and Dutta *et al.*'s schemes are twofold. First, our scheme is hierarchical, tailored to the heterogeneous WSNs. Second, our scheme achieves authenticated group key distribution, allowing every non-revoked sensor node to verify whether or not its generated group keys are valid, without requiring any extra communications.

3 Heterogeneous WSN Architecture

We partition a WSN into a number of *groups*. A high-end device is placed into each group, acting as the *group manager*. In contrast to sensor nodes, the high-end group managers have relatively higher computation capability, larger storage size, and longer radio range. They also have longer power supply, and can even be line-powered in some circumstances, e.g., when a WSN is deployed to monitor a building, the group managers can easily tap on the electricity lines to get power supply. Therefore unlike sensor nodes, group managers do not suffer too much from the resource scarceness problem. The introduction of high-end group managers into a WSN makes the once homogeneous network *heterogeneous*, as depicted in Figure 1.

In this architecture, downlink messages broadcast by the base station directly reach sensor nodes, whereas uplink messages sent by a sensor node to the base stattion is forwarded via its group manager, which acts as an intermediary between the base station and the sensor nodes within its jurisdiction. A sensor node may reach the group manager directly, or by traversing a *short* multi-hop path. Since group managers are not severely constrained by resources, communication at the level of group managers (including the base station) does not suffer from the limits upon sensor nodes.

Intuitively, the inclusion of powerful group managers provides shortcuts for data delivered from the sensor nodes to the base station, so the overall system performance and in turn the lifetime of the network are expected to be

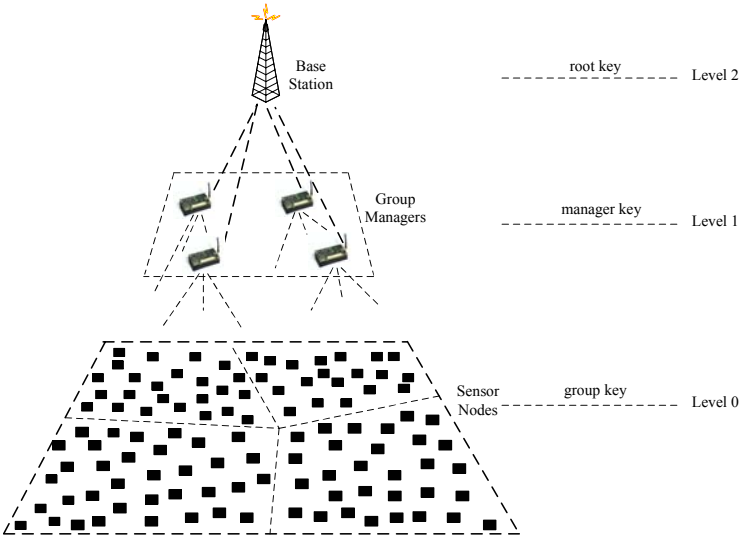

Fig. 1. Heterogeneous Wireless Sensor Network

greatly improved. Indeed, the effect of adding powerful nodes to WSNs has been analyzed in [14]: only a modest number of reliable, long-range backhaul links and line-powered nodes are required to have a significant effect, and if properly deployed, heterogeneity can triple the average delivery rate and yield a 5-fold increase in the lifetime of a large battery-powered sensor networks.

4 Model and Definition

System Model. Three types of entities are involved in our hierarchical group key distribution system: the base station, group managers, and a large number of sensor nodes. The sensor nodes are partitioned into a number of n_G groups, and each group has a group manager. A group has a unique group ID, and we use G_ν to denote the ID of group $\nu \in \{1, \cdots, n_G\}$. Each sensor node in a group is uniquely identified by an ID number i, where $i \in I \subseteq \{1, \cdots, n\}$, and I is the set of all node ID numbers of that group and n is the largest possible ID# in the system.

In correspondence to the heterogenous architecture, the keys held by the entities form a hierarchy, as shown in Figure 1: the base station holds a *root key* at level 2, each group manager has a distinct *manager key* at level 1, and sensor nodes in every group hold a common *group key* during each session at level 0. Traffic generated at lower level can be decrypted or authenticated using the keys at higher levels, but not the other way around. This key hierarchy helps to implement "separation of duty" within the system, e.g., it is not necessary for the sensor nodes to process the control messages broadcast by the base station to the group managers.

A group manager takes charge of distribution of group keys within its group. A group key is associated with each session. To distribute a group key for a new session, the group manager broadcasts a *key update message* to all its sensor nodes. The group key is then computed by a sensor node based on the received key update message and its preloaded *personal secret*. Denote the personal secret of sensor node i as S_i, which is a vector of m elements where m is the maximum number of sessions supported by the WSN. Each element in S_i corresponds to a session and we use $S_i[j]$ to denote the element corresponding to the jth session, $j \in \{1, \cdots, m\}$. $S_i[j]$ becomes *obsolete* once the group key for the jth session is established; otherwise $S_i[j]$ is *fresh*. A sensor node can be revoked or non-revoked. Only non-revoked sensor nodes are able to compute the group keys. To be resilient to the lossy channel of WSNs, the generation of group keys is self-healing in the sense that a non-revoked sensor node can recover group keys of all previous sessions as long as it successfully receives the key update message of the current session.

Adversary Model. We assume that the base station and the group managers are trusted, as we mainly concern with the distribution of group keys among sensor nodes. An adversary is able to passively eavesdrop on, or actively intercept, modify, insert, or drop key update messages from a group manager to all its sensor nodes. We also allow the adversary to compromise up to t sensor nodes in a group, where t is a system parameter.

Definition. We formally define the concept and security requirements of hierarchical self-healing group key distribution, by revisiting and extending the definition in [6].

Definition 1. *(Hierarchical Self-healing Group Key Distribution with t-Revocation) Let n, m, t be system parameters. $\mathcal{D}$ is hierarchical self-healing group key distribution with t-revocation, if the following holds:*

a. (Key Hierarchy) The manager keys held by the group managers are derived from the root key of the base station, but it is computationally infeasible to compute the root key from the manager keys. The same relationship should hold between group keys and the corresponding manager key.

b. (Secrecy of Personal Secret) For any $U \subset \{1, \cdots, n\}, |U| \leq t$, it is computationally infeasible for the nodes in U to collectively determine the fresh elements of S_i for any $i \notin U$.

c. (Authenticated Generation of Group Key) Let gK_j be the group key for session j, and B_j be the broadcast key update message from the group manager, where $j \in \{1, \cdots, m\}$. For any non-revoked sensor node in the group, gK_j is efficiently computed from B_j and $S_i[j]$ in an authenticated manner. On the contrary, it is computationally infeasible to compute the group session key from the key update message or a personal secret alone.

d. (t-Revocation) For any session j, let R_j be the set of revoked nodes at the start of session j, where $|R| \leq t$, it is computationally infeasible to compute gK_j from the broadcast message B_j and $\{S_i\}_{i \in R_j}$.

e. *(t-wise Forward Secrecy) Let $U \subseteq \{1, \cdots, n\}$ denote the sensor nodes which joined the group after session j. Given that $|U| \leq t$, it is computationally infeasible for all members in U to collectively compute $gK_1, \cdots, gK_j$, even with the knowledge of $gK_{j+1}, \cdots, gK_m$.*
f. *(Self-healing) A non-revoked sensor node between sessions j_1 and j_2, $1 \leq j_1 < j_2 \leq m$, can efficiently compute any gK_j, $j_1 \leq j \leq j_2$, from B_{j_2} and its personal secret.*

5 Our Construction

5.1 Scheme Details

We suppose that the set of revoked users is monotonic, i.e., a revoked user never rejoins the network. Let F_q be a finite field, where q is a large prime number. All arithmetic operations are performed in F_q. Let $h, h_R, h_F : \{0,1\}^* \rightarrow F_q$ be cryptographic hash functions, and n_G, n, m, t be system parameters.

– ***System Initialization.*** The base station chooses a root key $rK = [rk_1, rk_2]$, where rk_1 and rk_2 are random numbers of appropriate length. For each group $G_\nu, \nu = 1, 2, \cdots, n_G$, the base station computes a manager key as $mK_{G_\nu} = [mk_1, mk_2]$, where $mk_1 = h(G_\nu, rk_1)$ and $mk_2 = h(G_\nu, rk_2)$. Clearly, it is computationally infeasible to compute one manager key from another without knowing the root key. Then the base station securely passes the manager keys to the corresponding group managers. We do not specify how this can be done, but it often suffices by using some out-of-band channel. Upon receipt of the manager keys, the group managers begin the preparation for setting up group keys. Without loss of generality, let's consider a particular group G_ν whose manager key is mK_{G_ν}. The group manager sets mk_1 to be the seed s_R for a one-way hash chain of length $m+1$, i.e.,

$$\begin{aligned} k_R^j &= h_R(k_R^{j-1}) \\ &= h_R(h_R(k_R^{j-2})) = \cdots = h_R^j(s_R), 1 \leq j \leq m+1 \end{aligned} \tag{1}$$

and sets mk_2 to be the seed s_F for another hash chain of length m, i.e.,

$$\begin{aligned} k_F^j &= h_F(k_F^{j-1}) \\ &= h_F(h_F(k_F^{j-2})) = \cdots = h_F^j(s_F), 1 \leq j \leq m \end{aligned} \tag{2}$$

The group key gK_j for session $j \in \{1, \cdots, m\}$ is defined to be:

$$\begin{aligned} gK_j &= k_R^{m-j+1} + k_F^j \\ &= h_R^{m-j+1}(s_R) + h_F^j(s_F) \end{aligned} \tag{3}$$

We can see that the hash chain associating with $h_R()$ is used in the *reverse* order thus called the reverse hash chain, and that associated with

$h_F()$ called the forward hash chain. The group manager then selects m random t-degree polynomials $f_1(x), \cdots, f_m(x) \in F_q[x]$, each corresponding to a session. The personal secret for the member sensor node i is defined to be $S_i = [f_1(i), \cdots, f_m(i)]$. The group manager sends S_i together with k_R^{m+1} and s_F to each node i in a secure manner. Note that k_R^{m+1} will be used as the initial *authenticator* (denoted as $Auth$) in the process of group key generation.

- ***Broadcast.*** At the start of each session, the group manager broadcasts a key update message to enable sensor nodes to generate a new group key. Let $R_j = \{i_1, ..., i_w\}$ be the set of revoked sensor nodes upon the start of session $j \in \{1, \cdots, m\}$ and $|R_j| = w \leq t$. The group manager chooses a random set $R'_j = \{i'_t, \cdots, i'_{w+1}\} \subset \{1, \cdots, n\} \setminus I$, where I is the set of all node IDs of that group. That is, the group manager chooses $t - w$ random IDs that are not in that group. Next, the group manager computes k_R^{m-j+1} from s_R by Equation (1), and then computes the following polynomials:

$$r_j(x) = (x - i_1) \cdots (x - i_w)(x - i'_{w+1}) \cdots (x - i'_t)$$
$$b_j(x) = k_R^{m-j+1}.r_j(x) + f_j(x)$$

  We call $r_j(x)$ the revocation polynomial and $f_j(x)$ the masking polynomial. Finally, the group manager broadcasts the key update message B_j to the sensor nodes in its group, where

$$B_j = R_j \cup R'_j \cup \{b_j(x)\}$$

- ***Session Key Generation.*** Upon receipt of B_j, if node i is not revoked, it is able to compute $k_R^{m-j+1} = \frac{b_j(i) - f_j(i)}{r_j(i)}$. Then it can validate k_R^{m-j+1} using the authenticator $Auth$. For example, if $Auth = k_R^{m+1}$, then the validation is to test $Auth \stackrel{?}{=} h_R^j(k_R^{m-j+1})$. If the validation fails, the node aborts the key generation. Otherwise, it continues to compute $k_F^j = h_F^j(s_F)$ using s_F (Equation (2)), and in turn the group key $gK_j = k_R^{m-j+1} + k_F^j$. The node also updates $Auth$ by setting $Auth = k_R^{m-j+1}$. For efficiency reason, the node can also choose to keep k_F^j instead of s_F for future sessions.
- ***Addition of New Group Member.*** A newly added member in session j is not allowed to compute group keys of previous sessions. To add a new member with ID $\alpha \in \{1, \cdots, n\}$ starting from session j, the group manager computes and gives $S_\alpha = \{f_j(\alpha), f_{j+1}(\alpha), \cdots, f_m(\alpha)\}$ and $k_F^j = h_F^j(s_F)$ to the node.

5.2 Efficiency

Our scheme is highly efficient in terms of storage, communication, and computation overhead. For storage, the personal secret together with the authenticator accounts for $(m + 1) \log q$ bits storage in each sensor node (compared to Dutta *et al.*'s scheme, ours only needs $\log q$-bit more storage for the authenticator).

For communications, our scheme generates $t(\log q + \log n) \approx t \log q$ bits key update message (since $n \ll q$), which is almost the same as the bit length of the key update message in Dutta *et al.*'s scheme. For computation, no costly public key primitive is involved in our scheme, and the computation overhead inflicted upon sensor nodes includes only cryptographic hash function and polynomial operations.

5.3 Security Analysis

Theorem 1. *The above construction is a hierarchical self-healing group key distribution scheme with respect to Definition 1.*

Proof. It is not difficult to check that our scheme meets the properties of key hierarchy, authenticated computation of group key, and self-healing. We thus, in what follows, focus on showing that our scheme satisfies other security requirements.

$\diamondsuit$ Secrecy of Personal Secret. Personal secrets are computed from t-degree polynomials. For any t-degree polynomial corresponding to a session, a set U of sensor nodes, where $|U| = \tau \leq t$, contributes τ points over the polynomial. It is thus impossible (in an information theoretic sense) for τ nodes to determine the polynomial **solely** from the personal secrets they have, and in turn any other value of the polynomial. It remains to check whether the broadcast key update messages reveal information on personal secrets. Let us consider a particular non-revoked node i in session j. From the broadcast message B_j, node i calculates $k_R^{m-j+1} = \frac{b_j(i)-f_j(i)}{r_j(i)}$. Then with k_R^{m-j+1}, node i can actually compute any $f(i'), i' \neq i$, as $f(i') = b_j(i') - k_R^{m-j+1}.r_j(i')$. This suggests that once a group session key is established, the element of a sensor node's personal secret corresponding to that session is revealed to all other non-revoked nodes. This is exactly the reason why we distinguish between obsolete and fresh elements within a personal secret. We stress that the fresh elements of a personal secret remain secret, since they are computed from different polynomials.

$\diamondsuit$ t-Revocation. Without loss of generality, let's consider the last session m, and assume the set of t revoked nodes $R_m = \{1, 2, \cdots, t\}$, the maximum number of allowed revoked nodes, at the start of the session. Our goal is to show that these t revoked nodes cannot compute gK_m from the broadcast key update message B_m and their personal secrets. We model the coalition of the t revoked nodes as a polynomial-time algorithm $\mathcal{A}$, which takes View of the protocol as input and outputs a guessed group key gK'_m for session m. We say $\mathcal{A}$ breaks t-revocation if gK'_m is authenticated with respect to the group keys of previous sessions (or the authenticator). We prove, by contradiction, that if $\mathcal{A}$ breaks t-revocation, then we can construct a polynomial-time algorithm $\mathcal{B}$ for inverting one-way hash function $h_R(.)$, using $\mathcal{A}$. In particular, given $y = h_R(x)$, $\mathcal{B}$ computes x by invoking $\mathcal{A}$ as follows.

$\mathcal{B}$ sets $k_R^2 = y$ while leaves k_R^1 undefined (k_R^1 should be x by definition), and then computes $k_R^3 = h_R(y), k_R^4 = h_R(k_R^3) = h_R^2(y), \cdots, k_R^{m+1} = h_R(k_R^m)$.

$\mathcal{B}$ continues to select a random s_F and compute the forward hash chain $k_F^j = h_F(k_F^{j-1}) = h_F(h_F(k_F^{j-2})) = \cdots = h_F^j(s_F), 1 \leq j \leq m$. $\mathcal{B}$ sets the j-th group session key as $gK_j = k_R^{m-j+1} + k_F^j, 1 \leq j \leq m-1$, and leaves gK_m undefined (gK_m should be $x + h_F^m(s_F)$ by definition). $\mathcal{B}$ selects m random polynomials $f_1(x), \cdots, f_m(x) \in F_q[x]$, each of degree t. For each node $i \in \{1, 2, \cdots, t\}$, $\mathcal{B}$ computes the personal secret $S_i = \{f_1(i), \cdots, f_m(i)\}$. For each session $1 \leq j \leq m-1$, $\mathcal{B}$ computes the broadcast key update message $B_j = R_j \cup R'_j \cup \{b_j(x)\}$, where $R_j \subseteq R_m$, and $b_j(x) = k_R^{m-j+1}.r_j(x) + f_j(x)$, with R'_j and $r_j(x)$ being constructed in exactly the same manner as in our scheme. To compute the broadcast key update message for session m, $\mathcal{B}$ selects a random $k_R^1 \in F_q$, and computes $B_m = R_m \cup \{b_m(x)\}$, where $b_m(x) = k_R^1.r_m(x) + f_m(x)$, with $r_m(x) = (x-1)(x-2)\cdots(x-t)$. Then $\mathcal{B}$ sets View of the protocol as

$$\mathsf{View} = \left\{ \begin{array}{c} k_R^{m+1} \\ s_F \\ \{f_j(1), f_j(2), \cdots, f_j(t)\}, j = 1, \cdots, m \\ B_j, j = 1, \cdots, m \\ gK_1, \cdots, gK_{m-1} \end{array} \right\}.$$

Finally, $\mathcal{B}$ gives View to $\mathcal{A}$, which in turn outputs gK'_m, its guess for the actual group session key gK_m. $\mathcal{B}$ outputs $gK'_m - h_F^m(s_F)$. It is easy to see that for any session $j, 1 \leq j \leq m-1$, the simulation by $\mathcal{B}$ in constructing View is perfect with respect to the original scheme. We next show that the simulation for session m (where a random k_R^1 is used) is also perfect to $\mathcal{A}$. To see this, $\mathcal{A}$ has $b_m(x)$ (from B_m) and $\{f_m(1), \cdots, f_m(t)\}$ at its disposal. First, from $f_m(1), \cdots, f_m(t)$, $\mathcal{A}$ cannot determine $f_m(.)$ as it only has t points over the t-degree polynomial which has $t+1$ unknown coefficients, thus $\mathcal{A}$ cannot compute any $f_m(i), i \notin \{1, \cdots, t\}$. Here, we need to stress that $b_m(x)$ does not help $\mathcal{A}$ to determine $f_m(x)$. The reason is that $b_m(x)$ at the points of $\{1, \cdots, t\}$ equals $f_m(1), \cdots, f_m(t)$, respectively, thereby revealing no more information on $f_m(x)$. Second, on $b_m(x)$, there are two cases to be considered.

1. $i \in \{1, \cdots t\}$: from $b_m(x)$, $\mathcal{A}$ can evaluate $b_m(i)$ which equates $f_m(i)$ regardless of k_R^1. $\mathcal{A}$ thus can check against $f_m(i)$ which is already at its disposal. The simulation is perfect to $\mathcal{A}$.
2. $i \notin \{1, \cdots, t\}$: from $\mathcal{A}$'s point of view, $b_m(i) = k_R^1.r_m(i) + f_m(i)$ is random, because in each $b_m(i)$ there are two unknown variables k_R^1 and $f_m(i)$; for every value of k_R^1, there is a corresponding value $f_m(i)$. Hence the adversary $\mathcal{A}$ who cannot break the polynomial $f_m(.)$ has no way to distinguish whether or not the genuine k_R^1 is used in $b_m(i)$. The simulation is thus again perfect to $\mathcal{A}$.

Combining together the above arguments, we conclude that $\mathcal{B}$ inverts h_R with the same advantage as $\mathcal{A}$ breaks t-revocation.

$\diamondsuit$ *t-wise Forward Secrecy.* An intuition on t-wise forward secrecy is that if a set of nodes U join the system in session j, they are given $k_F^j = h_F^j(s_F)$. To compute

the group key for an earlier session $j' < j$, they need $k_F^{j'}$. In our scheme, the only information relates to $k_F^{j'}$ are $k_F^j, k_F^{j+1}, \cdots, k_F^m$. Computing $k_F^{j'}$ from these group keys of later sessions clearly involves inverting the one-way cryptographic hash function $h_F(.)$. As it is straightforward to construct an adversary $\mathcal{B}$ for inverting $h_F(.)$, based on an adversary $\mathcal{A}$ that breaks t-wise forward secrecy (similar to the above proof), we omit the details of the proof. □

6 Conclusion

We studied hierarchical self-healing group key distribution for heterogenous WSNs. In particular, we formulated a model for hierarchical self-healing group key distribution, and proposed concrete schemes that achieve provably computational security and high efficiency.

Acknowledgement

This work is supported by A*STAR project SEDS-0721330047.

References

1. Blundo, C., D'Arco, P., Santis, A., Listo, M.: Definitions and Bounds for Self-healing Key Distribution. In: Díaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) ICALP 2004. LNCS, vol. 3142, pp. 234–245. Springer, Heidelberg (2004)
2. Blundo, C., D'Arco, P., Santis, A., Listo, M.: Design of Self-healing Key Distribution Schemes. Designs, Codes and Cryptography 32(1-3), 15–44 (2004)
3. Blundo, C., Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 471–486. Springer, Heidelberg (1993)
4. Chan, H., Perrig, A., Song, D.: Random Key Pre-distribution Schemes for Sensor Networks. In: IEEE Symposium on Security and Privacy, pp. 197–213 (2003)
5. Du, W.L., Deng, J., Han, Y.S., Varshney, P.K.: A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In: ACM Conference on Computer and Communication Security, CCS 2003, pp. 42–51 (2003)
6. Dutta, R., Change, E.C., Mukhopadhyay, S.: Efficient Self-healing Key Distribution with Revocation for Wireless Sensor Networks Using One Way Key Chains. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 385–400. Springer, Heidelberg (2007)
7. Eschenauer, L., Gligor, V.D.: A Key-Management Scheme for Distributed Sensor Networks. In: ACM Conference on Computer and Communication Security, CCS 2002 (2002)
8. Huang, D., Mehta, M., Medhi, D., Harn, L.: Location-aware key management scheme for wireless sensor networks. In: 2nd ACM workshop on Security of Ad Hoc and Sensor Networks
9. Liu, D., Ning, P.: Improving Key Pre-distribution wih Deployment Knowledge in Static Sensor Networks. ACM Transactions on Sensor Networks (2005)
10. Liu, D., Ning, P., Du, W.L.: Group-based Key Pre-distribution in Wireless Sensor Networks. In: ACM Workshop on Wireless Security (2005)

11. Liu, D., Ning, P., Sun, K.: Efficient Self-Healing Group Key Distribution with revocation Capability. In: ACM Conference on Computer and Communication Security, CCS 2003 (2003)
12. Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J.D.: SPINS: Security Protocols for Sensor Networks. Wireless Networks Journal (WINE) (September 2002)
13. Staddon, J., Miner, S., Franklin, M., Balfanz, D., Malkin, M., Dean, D.: Self-healing Key Distribution with Revocation. In: IEEE Symposium on Security and Privacy, S&P 2002, pp. 241–257 (2002)
14. Yarvis, M., et al.: Exploiting Heterogeneity in Sensor Networks. In: IEEE INFOCOM 2005 (2005)
15. Zhu, S., Setia, S., Jajodia, S.: LEAP: Efficient Security Mechanisms for Large-scale Distributed Sensor Networks. In: ACM Conferenc on Computer and Communication Security, CCS 2003, pp. 62–72 (2003)