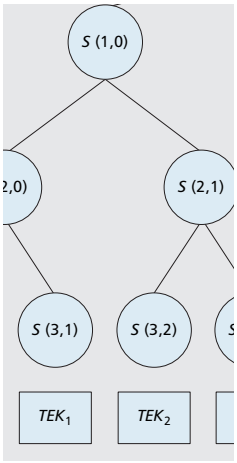
SELF-HEALING GROUP-WISE KEY DISTRIBUTION SCHEMES WITH TIME-LIMITED NODE REVOCATION FOR WIRELESS SENSOR NETWORKS

MINGHUI SHI AND XUEMIN (SHERMAN) SHEN, UNIVERSITY OF WATERLOO
YIXIN JIANG AND CHUANG LIN, TSTINGHUA UNIVERSITY



The authors introduce two novel group-wise key distribution schemes with time-limited node revocation for secure group communications in wireless sensor networks.

ABSTRACT

In this article two novel group-wise key distribution schemes with time-limited node revocation are introduced for secure group communications in wireless sensor networks. The proposed key distribution schemes are based on two different hash chain structures, dual directional hash chain and hash binary tree. Their salient security properties include self-healing rekeying message distribution, which features a periodic one-way rekeying function with efficient tolerance for lost rekeying messages; and time-limited dynamic node attachment and detachment. Security evaluation shows that the proposed key distribution schemes generally satisfy the requirement of group communications in WSNs with lightweight communication and computation overhead, and are robust under poor communication channel quality.

INTRODUCTION

Wireless sensor networks (WSNs) are an emerging class of networks with embedded systems [1]. A WSN is a collection of sensors, the scale of which can range from a few hundred to a few hundred thousand sensors. They are small in size and have wireless communication capability within short distances. A typical sensor node contains a power unit, a sensing unit, a processing unit, a storage unit, and a wireless transceiver. Each sensor node is usually specialized to monitor a specific environmental parameter such as temperature, light, sound, or acceleration. However, a WSN may be able to monitor multiple parameters by combining several kinds of sensor nodes.

Figure 1 shows a typical WSN architecture, which often contains one or more base stations providing centralized control. A base station typically serves as an access point for sensors or a gateway to another associated infrastructure such as data processing and management units. Individual sensors communicate locally with neighboring sensors and send their data over the peer-to-peer sensor network to the base station. Hence, there are three basic communication modes within WSNs: node to node, node to base station, and base station to node. Sensors do not rely on any predeployed network infrastructure, but communicate via an ad hoc wireless network.

Secure group communication, which occurs among a certain subnet (group) of sensor nodes and probably base stations, is increasingly used for efficient group-oriented applications in WSNs, such as mobile microrobots sent out for different application profiles in a battlefield and multiple sensor groups each with a specific sensing profile, mentioned earlier. Group communication also limits the propagation of the message flow within the group, which is beneficial to deliver messages efficiently and securely, and reduce network-wide power consumption.

Given the open nature of broadcast channels, the combination of group communication and WSNs is more susceptible to unauthorized access. Thus, confidentiality must be provided in group communications so that illegitimate nodes are prevented from having access to secret contents, whereas legitimate nodes can decrypt the data, which are broadcast to the entire network. To address these issues, the traffic encryption key (TEK), a symmetric secret key, is used to encrypt data at the source and decrypt them at the destination [2]. Furthermore, considering the dynamic network topology due to nodes' attachment and detachment, it is necessary to refresh the TEK to prevent a detached node accessing future communications and a newly attached node accessing prior communications. The group key manager[1] (GKM) located in the sensor net-

[1] *Generally speaking, there may be one or several GKMs responsible for distributing group or session keys to a large number of authorized group nodes via a broadcast message. We focus on the case of one group unless noted otherwise.*

1536-1284/07/$20.00 © 2007 IEEE

work controller is responsible for distributing rekeying messages to the nodes in the group secured by encrypting them using the key encrypting key (KEK) [3]. Based on the security requirement of the actual applications, a rekeying process may be triggered to update the TEK after each node attaches to or detaches from an active group session. This process ensures that a new node cannot decrypt previous group messages and prevents a detached node from eavesdropping on future group messages. Since each network topology change triggers a new rekeying process, the load of TEK refreshment messages may degrade performance and scalability in case of frequent network topology changes.

In this article we introduce two efficient self-healing group-wise key schemes with time-limited node revocation, which ensures forward/backward secrecy, certain collusion freedom, and group confidentiality in high packet loss environments. Based on the dual directional hash chain (DDHC) and hash binary tree (HBT), respectively, the proposed schemes offer a practical seal-healing method and an implicit node revocation[2] algorithm with lightweight computation and communication overhead to cope with dynamic network topology in WSNs. It is shown that, comparing with existing schemes, the DDHC/HBT mechanism can remarkably reduce both the computation and communication overhead at the GKM and the nodes, and thus improve the scalability and the performance of the key distribution scheme. Furthermore, the performance of the proposed schemes under poor broadcast channel condition is discussed. It is concluded that the proposed schemes can tolerate high channel loss rate, and hence can make a good trade-off between performance and security.

The rest of this article is organized as follows. We discuss the general issues in key management in WSNs and related work. We introduce the group-wise key distribution scheme based on the DDHC and the HBT, respectively. The security and the performance evaluations are presented, followed by the conclusion.

Related Work

WSNs are often deployed in open and hostile environments, and thus are subjected to great security risks. In order to protect confidentiality and integrity of the information, the sensor nodes should be securely associated with the neighbouring nodes and/or data sink via encrypted data link. Therefore, key management plays a critical role in establishing secure communications in WSNs. The key management usually includes the following three key distribution methods: *key distribution*, *key agreement*, and *key predistribution*. Traditional key distribution schemes require a trusted server to establish shared session keys between nodes. The key agreement scheme is usually based on asymmetric cryptography algorithms, which is not feasible for resource constrained sensors. Presently, the only practical scheme for key management in large sensor networks may be key predistribution, where key information is installed in each sensor node prior to deployment.

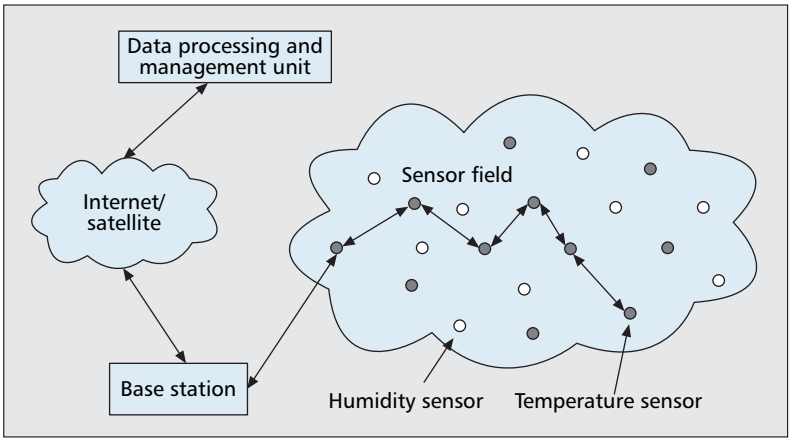


Figure 1. *The architecture of a wireless sensor network.*

Issues of Key Management in WSNs

Offering efficient key management in WSNs is challenging due to their constraints in hardware, deployment, network, and so on. The sensor network does not have fixed infrastructure and contains a very large number of entities with high density. A WSN is usually deployed randomly; therefore, designing a security scheme should not assume exact deployment knowledge of nodes. Sensor nodes may be deployed in public and hostile locations, and consequently exposed to physical attacks by an adversary, who may undetectably capture a sensor node and compromise the secret keys. Base stations in WSNs are centralized, powerful, and expensive. It is tempting to rely on them too much in functions. This may attract attacks on the base station and limit application of the security protocol. There are also some limitations and impairments in physical design of the sensor node and wireless network environment, such as the imperfect wireless channel and limited bandwidth, memory resources, and computation capacity. Therefore, some special security and performance requirements should be focused on in WSNs:

- Resilience against node capture: An adversary can mount a physical attack on a sensor node after deployment. It is required to estimate the fraction of total network communications compromised by such captured nodes.
- Resilience against node replication: An attacker may insert additional hostile nodes into a WSN. This is a serious attack since even a single compromised node might allow an adversary to populate the network with a clone of the captured node to such an extent that legitimate nodes could be outnumbered, and the adversary can thus gain full control of the network.
- Node revocation or participation: A new sensor may be deployed dynamically in a WSN, and a detected misbehaving node should also be able to be removed dynamically from the system.
- Scalability: When the number of sensors grows, security may be weakened. It is necessary to explore the maximum supported network size for a given deployment policy, since different key deployment policies will result in different

[2] *Node revocation can be described as follows. Let* U *be the set of all possible group nodes, and a subset of* U, R, *be the set of revoked nodes. The group node revocation is required to offer a secure way for the GKM to transmit rekeying messages over a broadcast channel shared by all nodes so that any nodes in* R *cannot decrypt it, even when, more strictly, they collude with each other.*

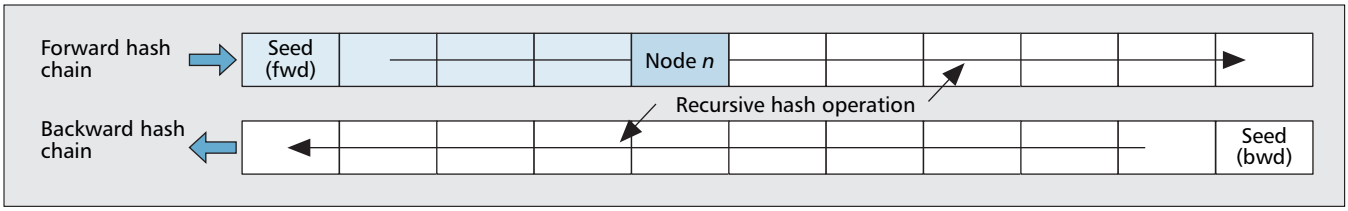


Figure 2. *Structure of dual directional hash chains.*

network scales, which significantly impacts the scalability of key schemes.

In practice, it is difficult to deal with all these constraints perfectly. A trade-off is usually made according to the actual application or purpose of the sensor network.

Study of Group-Wise Key Distribution Schemes

Due to the dynamic nature of group communications, the group key needs to be not only established at the initial phase but also refreshed from time to time. Typically, the additional security requirements for group-wise key distribution schemes include [4]:

- *Group confidentiality*: Nodes that are not part of the group should not have access to any key that can decrypt any data broadcasted to the group.
- *Forward secrecy*: Nodes that detach from the group should not have access to any future keys, which ensures that a detached node cannot decrypt further data.
- *Backward secrecy*: A new node that attaches to the session should not have access to any old key, which ensures that a node cannot decrypt data sent before it attaches to the group.
- *Collusion freedom*: Any set of fraudulent nodes should not be able to deduce the current active TEK.

In addition, the lossy channel usually causes scheme failure if nodes cannot communicate with the GKM due to communication interruption. The dynamics of the network topology also increase service disruption probability, since some nodes may lose connections temporarily. Hence, it is required to offer a reliable rekeying process with minimum number and size of rekeying messages. The rekey scheme should also require neither a large number of storage keys nor high computation overhead at the GKM or the nodes in the group.

A straightforward approach to key establishment is to use the key distribution method on top of a pre-installed key in the sensor nodes to establish group-wise keys. A lightweight key management system [5] considered a WSN where a group of sensor nodes are deployed in different phases, and proposed a group-wise key distribution scheme through links secured with pair-wise keys. Other approaches include using secure but costly asymmetric cryptography. Burmester-Desmedt [6] and IKA2 [7] used a Diffie-Hellman-based group key transport protocol. These two algorithms were further improved by ID-STAR [8], which adopted identity-based cryptography where sensor nodes' public keys can be derived from their identities. It is also possible to use an existing pair-wise key structure to establish group-wise keys.

The rekeying mechanism is another critical security function for group communications in WSNs. Inefficient rekeying eventually causes WSNs to not work as planned. In order to tackle the scalability problem of rekeying operation with highly dynamic network topology, a number of efficient approaches have recently been proposed (LKH [9], Subset Difference [10], etc.). Considering the interdependency of rekeying messages, a group key distribution scheme with revocation can be classified into two distinct classes: stateful or stateless. In a stateful scheme [9], a legal node's state in the current rekey affects its ability to decrypt future group keys. A stateless scheme relies only on the current rekeying message and the node's initial configuration [11]. A non-revoked node can decrypt the new TEK independent of previous rekeying messages without contacting the GKM, even if the node is offline for certain sessions. This property makes a stateless scheme more useful in scenarios where some nodes are not constantly online or suffer from burst packet losses.

The scheme with stateless node revocation was first investigated in [11], which requires $O(tn^2\log t)$ storage keys and $O(t^2n\log^2 t)$ messages, and allows the GKM to revoke any number of nodes, while at most t of them could collude to obtain the TEK. Subsequently, two stateless revocation schemes, CS and SD, were proposed in [12]. Given N nodes with $\log N$ keys, the CS scheme can revoke any R nodes with $O(R\log(N/R))$ messages. The SD scheme reduces the message number to $O(R)$, while the node storage overhead is increased to $O(\log^2(N))$ with $O(\log N)$ cryptographic operations.

The lossy channel usually results in scheme failure if nodes cannot communicate with the GKM. The dynamics of the network topology also increase service disruption probability, since some nodes may lose connection temporarily. Therefore, in addition to node revocation capacity, some recent work also addressed self-healing issue so that a group node could recover the missed session keys from the latest rekeying message on its own. Based on two-dimensional t-degree polynomials, Staddon *et al.* [13] first presented a self-healing group key distribution scheme, which was further improved by Liu and Ning [14].

Self-Healing Group-Wise Key Distribution Schemes with Time-Limited Node Revocation

We introduce two efficient self-healing group key schemes with time-limited node revocation based on the DDHC and HBT.[3] It is defined that a

[3] *Due to page limits, we go through the major features of the two schemes.*

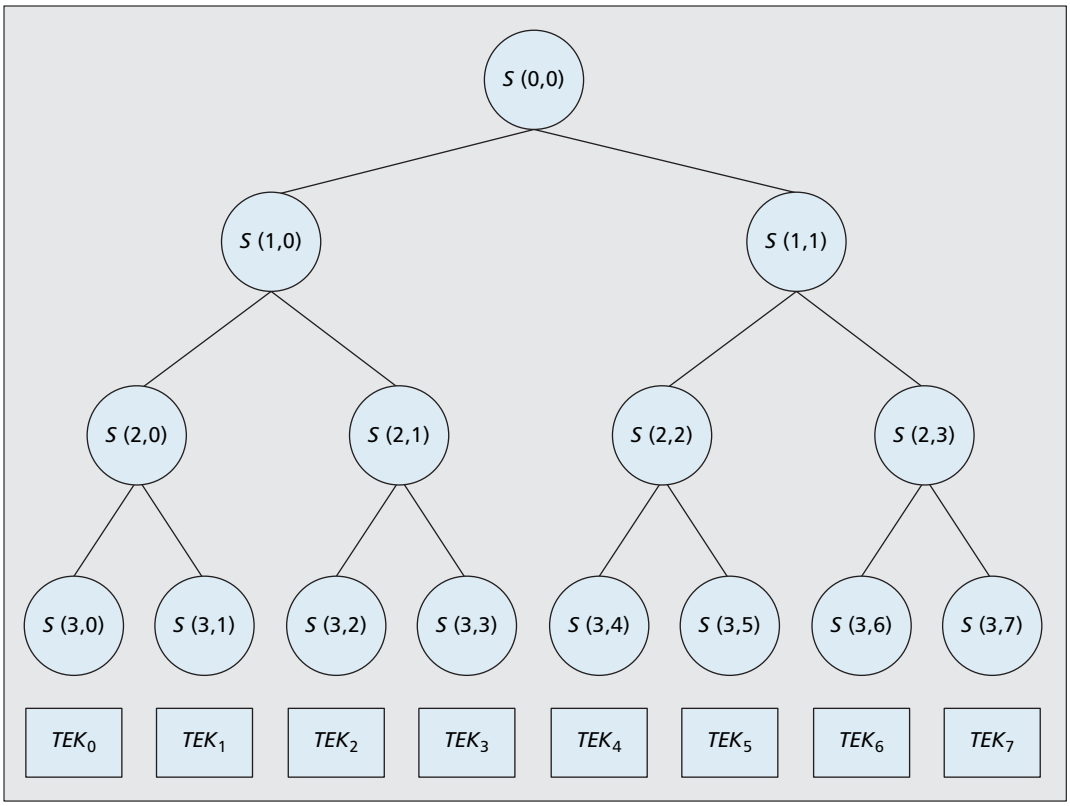


Figure 3. *Example of a hash binary tree with* D = *3.*

A non-revoked node can decrypt the new TEK independently from previous re-keying messages without contacting the GKM, even if the node is off-line for certain sessions.

sender may transmit a broadcast message to receivers (group members) directly or indirectly, and the life cycle of a wireless network is divided into time intervals called *sessions* of fixed duration.

Dual Directional Hash Chain and Binary Hash Tree

We first introduce the concept of a one-way hash function, which is the foundation of the DDHC. A hash function Hash(.) takes a binary string of arbitrary length as input, and outputs a binary string of fixed length. A one-way function H satisfies the following two properties:

- Given x, it is easy to compute y such that y = Hash(x).
- Given y, it is computationally infeasible to compute x such that y = Hash(x).

The security features of the proposed group-wise key distribution schemes are based on the one-way property of the hash function.

Dual Directional Hash Chain — A one-way hash chain, as illustrated by forward or backward hash chains in Fig. 2, is formed by recursively hashing x and lining them up in sequence. Let us take the forward hash chain as an example. Due to the one-way property of the hash function, given any point node n in the chain, it is computationally infeasible to calculate the elements on its left, but easy to compute those on its right.

A DDHC (Fig. 2) is composed of two one-way hash chains of equal length, a forward hash chain and a backward hash chain. It can be derived as follows:

- Generating two random key seed values, seed (fwd) and seed (bwd), for the forward and backward hash chains with size z + 1, respectively
- Repeatedly applying the same one-way function on each seed to generate two hash chains of equal length

Hash Binary Tree — The generation of an HBT requires two (left and right) hash functions. The HBT in the proposed group-wise key distribution scheme is constructed from a hash function Hash(.) by applying one of two cyclic bit shift functions, LeftShift(.) and RightShift(.) before the hash function, that is, Hash(LeftShift(.)) and Hash(RightShift(.)). As shown in Fig. 3, it is an HBT with depth equal to 3. S(1,0) generated by computing Hash(LeftShift(S(0,0))), and S(1,1) is generated by computing Hash(RightShift(S(0,0))). All other elements are generated similarly.

DDHC based Group-Wise Key Distribution Scheme

Initial System Setup — At the initial phase, the GKM first selects a secret seed as the end element of the *RK* chain. Then the GKM generates a one-way hash chain and uses the last hash value as the first element of the *RK* chain, as shown in Fig. 4. The length of the *RK* chain is sufficient to cover the session line of the life cycle of the multicast group. The rekeying message is broadcast within the sensor network from time to time. Each legitimate node within the multicast group is able to compute TEK, which encrypts and decrypts the multicast messages from the received *RK*. The GKM generates a sufficiently long DDHC chain, as shown in the figure.

At the initial phase, the GKM first selects a secret seed as the end element of the RK chain. Then the GKM generates a one way hash chain and uses the last hash value as the first element of the RK chain.

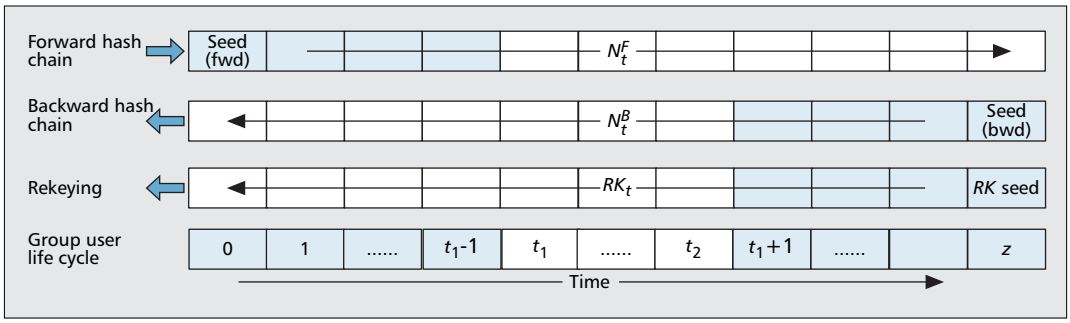


Figure 4. *Time-limited node revocation based on DDHC.*

A main or master KEK is shared between the GKM and each node for *InitGroupKey* message encryption and authentication. In order to perform self-healing recovery of a rekeying message, a node has a small buffer that can store up to *l RK*s. Assume a node is legitimate between time window $[t_1, t_2]$. In the initiation stage, the GKM sends each sensor node the element in the forward hash chain at time t_1, the element in the backward hash chain at time t_2, and the lth *RK*, which are encrypted by the corresponding KEK associated with the sensor node.

Time-Limited Node Revocation Scheme — The application of the DDHC in the time-limited node revocation mechanism is shown in Fig. 4. The TEK at time t is composed by the corresponding elements in the forward hash chain, backward hash chain, and rekeying chain,

$$TEK_t = f(N_t^F, N_t^B, RK_t),$$

where $f(.)$ is a one way function.

Due to the one-way property of the DDHC, a sensor node can only access the TEKs between t_1 and t_2, since computing the TEK requires both corresponding elements in forward and backward hash chains. The sensor nodes can feasibly obtain both values within the time window $[t_1, t_2]$ from information sent by the GKM in the initiation stage. For any time out of the time window, the sensor node cannot compute both elements in the DDHC; therefore, it cannot achieve the TEK. Thus, an implicit time-limited node revocation is achieved. Each sensor node can only access a predefined contiguous range of the TEKs between $[t_1, t_2]$.

During the system life cycle, when a node attaches to an active group, the GKM assigns the pair of the element in the forward hash chain at t_1 and the element in the backward hash chain at t_2 to the new node according to its prearranged life cycle $[t_1, t_2]$. Due to the property of the DDHC, once the node's life cycle is expired, it is forced to detach from the multicast session without need for direct intervention of the GKM.

Self-Healing Rekeying Mechanism — The GKM broadcasts the *RK*, which is encapsulated in the rekeying (*RefreshKey*) message at a defined time interval so that the legitimate sensor nodes are able to renew the TEK. Due to the one-way property of the *RK* sequence, the *RefreshKey* message does not need message authentication code since the receiver can verify if the received *RK* belongs to the same key sequence by checking if its hash value equals the previous *RK*. Such implicit authentication notably decreases the message size.

In the rekeying phases, all *RK*s are released to all nodes by the GKM in reverse order (i.e., RK_o will be released for session 0, RK_1 for session 1, …, RK_n for session n, etc.). Therefore, given current RK_j in the hash chain, nodes can only compute previous keys recursively. Since most sensor nodes work in wireless and likely hostile environments, it is possible that a sensor node does not receive the *RK*s all the time. A self-healing rekeying mechanism offers equivalent reliable *RK* transmission over a lossy broadcast channel.

Consider that each rekeying message contains only one *RK* in the current session. Although rekeying messages may be lost during transmission, self-healing can be achieved. The lost *RK*s in previous rekeying messages can be recovered using the one-way hash function and the last received *RK*. Consequently, the TEK can be successfully derived by each node. Thus, the proposed self-healing scheme can efficiently tolerate high packet loss or error up to the size of the *RK* buffer. On the other hand, if the channel condition is good but the application of the sensor network is not delay-sensitive, receiving or sending the rekeying message every time is not necessary, and energy consumption can be reduced.

HBT-Based Group-Wise Key Distribution

Earlier, we proposed a DDHC-based self-healing group key distribution scheme with time-limited node revocation. To further improve the security and performance with high computation efficiency (fewer hash operations), here we propose a second scheme in which the HBT is adopted to generate all pre-assigned seeds. Each TEK is linked to a leaf node in the HBT, and all leaf nodes are derived using a hash algorithm on these seeds.

Initial System Setup — The GKM generates an HBT with the scale according to the maximum number, or life cycle, of the multicast session. Without loss of generality, we assume that the maximum number of group sessions, or life cycle time unit, is m. Correspondingly, the depth of the HBT is $D = \lceil \log_2 m \rceil$. Then the derivation algorithm of the HBT can be illustrated in detail as follows.

The GKM randomly generates an initial seed $S(0,0)$ that is sufficiently large (e.g., 256 bits).

The GKM generates two left and right intermediate seeds in the first level by applying the left and right hash functions to the initial seed $S(0,0)$, respectively, as shown earlier, repeatedly executing and operations until all seed values in the tree depth $D = \lceil \log_2 m \rceil$ are generated. Each TEK is related to a leaf node in the HBT, as demonstrated in Fig. 3. That is, TEK_1 is associated with $S(D,0)$, TEK_2 with $S(D,1)$, and so on. The HBT in Fig. 3 can satisfy a group communication with maximum eight sessions. All leaf nodes in the HBT can be derived by applying a hash algorithm on the root seed value $S(0,0)$ in the HBT.

Time-Limited Node Revocation Mechanism — In the group-wise key distribution scheme based on the HBT, TEK_t at time t is composed by the corresponding element of the leaf node in the HBT and current RK,

$$TEK_t = f(S(D, t-1), RK_t),$$

where f is a one-way function. Therefore, the accessibility of a TEK can be controlled by knowledge of the elements of the leaf nodes. When a sensor node attaches to an active group, the GKM distributes the elements of the leaf nodes to the sensor nodes corresponding to the allowable time window $[t_1, t_2]$.

However, the distribution of sending each element is not efficient for storage and wireless bandwidth. Since any node down a branch node can feasibly be computed as shown earlier, subtrees for a node with an allowable time window $[t_1, t_2]$ need to be found that can cover all the leaf nodes in the time window [15]. The pre-assigned seed set includes the subroot nodes in all such subtrees. All the leaf node seeds in the range of $[t_1, t_2]$ can be derived by repeatedly applying the hash function on such pre-assigned seeds. For instance, as shown in Fig. 3, for a node with allowable time window [3, 6], the GKM will assign seeds $\{S(2,1), S(2,2)\}$ to this node via a secure channel. The GKM does not need to assign seeds $\{S(3,2), S(3,3), S(3,4), S(3,5)\}$ to the node directly, since it can calculate these seeds by applying hash functions on $\{S(2,1), S(2,2)\}$. Therefore, the storage requirement for the group node is greatly reduced. Once a group node's life cycle is expired, it autonomously exits the group session without the GKM's direct intervention.

Self-Healing Rekeying Mechanism — As described earlier, the self-healing method for group rekeying distribution is based on a one-way hash chain. Each legitimate node can derive the allowable time window as $TEK_t = f(S(D, t-1), RK_t)$. $S(D, t-1)$ does not need to be transmitted at each rekeying. Each node can individually compute them according to the pre-assigned seeds and the current time. RK_t is encapsulated in the rekeying message, which is periodically sent by the GKM to all users. With a similar mechanism, self-healing can be achieved since the lost in RKs previous rekeying messages can be recovered using the hash function and the latest received RK. The TEK will be successfully derived by each node on its own.

Performance Evaluation of Proposed Group-Wise Key Distribution Schemes

Security Performance Evaluation

We evaluate the proposed schemes to see if they satisfy the security requirements for secure group communications described earlier. The confidentiality of the key distribution information is protected by the preshared key between the sensor nodes and the KGM. A sensor node not belonging to the group cannot generate an effective TEK even with the broadcast RK.

The group key distribution mechanism in the proposed schemes can ensure the refreshment of the TEK by periodic rekeying when a node attaches to or detaches from an active group session. In the DDHC-based scheme, as indicated in Fig. 4, a sensor node is restricted to access the group communication in the shaded range. The forward hash chain guarantees backward secrecy. A new node that participates in the group communication at time t_1 cannot calculate the previous hash keys before time t_1 because of the property of a one-way hash function. Similarly, the backward hash chain guarantees forward secrecy. Once a node detaches from the group session at time t_2, it cannot compute the subsequent hash keys after time t_2. TEK_t is computed as the combination of corresponding elements in the forward hash chain, backward hash chain, and RK chain at time t. In the HBT-based scheme, subroot nodes only are sent to the sensor node to generate subtrees. TEK_t is computed as the combination of the elements in the leaf node in the subtrees and RK chain at time t. Within the allowable time window, the sensor node has all the required information to compute the TEK, and is able to send and receive multicast messages. It is not feasible for the sensor node to compute the undistributed elements in forward or backward hash chains, or the element in the leaf node, respectively, in the time out of the time window. Take the example in Fig. 3 and assume the allowable time window is between 3 and 6. Due to the property of the one-way function, the sensor node is not able to know $S(2,1)$ from $S(3,3)$. Therefore, it cannot compute TEK_2 at $S(3,2)$. Thus, both proposed schemes meet the security requirements for forward and backward secrecy with time-limited node revocation.

Compared to the group-wise key distribution scheme based on the DDHC, the group-wise key distribution scheme based on the HBT offers stronger collusion freedom. In the DDHC-based scheme, a sensor node is able to compute part of the elements in either a forward or backward hash chain beyond the time window $[t_1, t_2]$. For example, sensor node A is assigned elements at the far right of the DDHC in Fig. 4, and sensor node B is assigned elements at the far left of the same DDHC. If they exchange the information of the elements in the DDHC, sensor node A is able to compute the whole forward hash chain using the information held by B, and B is able to compute the whole backward hash chain using the information held by A. On the other hand, the sensor node cannot determine the position

if the channel condition is good, but the application of the sensor network is not delay sensitive, receiving or sending the re-keying message every time is not necessary, and energy consumption can be reduced.

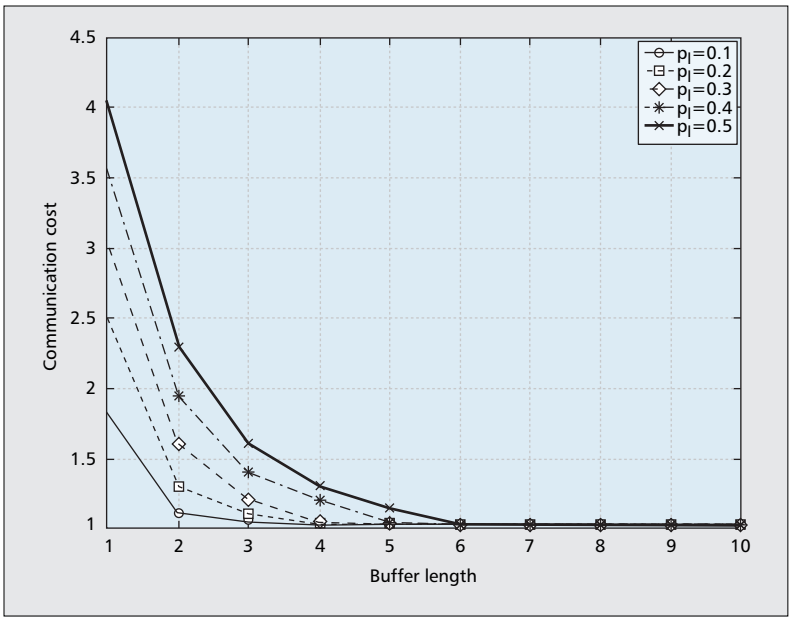


Figure 5. *Normalized communication costs vs. key buffer length.*

of the elements it received in the DDHC easily. Serious bleach of the DDHC requires the exact ones of many sensor nodes that are holding the information at the very ends of the DDHC. Therefore, in rare cases it is possible that two sensor nodes could combine the information they have and get access to TEKs outside of their allowable time window. In the HBT-based scheme, any two nodes of the same level in the HBT are isolated by the one-way property of the hash function. Therefore, a sensor node can only compute the elements in those leaf nodes that are derivable from the seeds the sensor has received. Sensor nodes are no longer able to cooperate with each other to access TEKs beyond the allowable time window.

Overhead Performance Evaluation

Besides the security requirements, we discuss the operation overhead in terms of storage, computation, and communication. The computation overhead comparison is shown in Table 1. For each group node in the HBT, the minimum number of hash operations is 1, while the maximum number of hash operations is $\lceil \log_2 m \rceil$, since the maximum depth of the HBT is $\lceil \log_2 m \rceil$. On average, the computation overhead is $O(\log_2 m)$. Accordingly, for the DDHC based scheme, the minimum number of hash operation is 2, while the maximum number of hash operations is $2(m-1)$. Assume that the lifecycle of each group node is uniformly distributed in [1, m]. On the average, the computation overhead is $m - 1$, that is, $O(m)$. Therefore, the HBT based algorithm has higher computation efficiency.

	Maximum	Minimum	Average
DDHC	$2(m-1)$	2	$O(m)$
HBT	$\lceil \log_2 m \rceil$	1	$O(\log_2 m)$

Table 1. *Computation overhead of the proposed group-wise key distribution schemes.*

Table 2 shows a concise comparison among the proposed schemes and other similar two self-healing key distribution methods [13, 14] in terms of communication and storage overhead. Our schemes behave similar to the scheme in [14], and better than the scheme in [13] in storage overhead. The broadcast communication cost of the two proposed schemes is $O(t\log q)$, while the cost of the scheme in [13] is $O((mt^2 + mt)\log q)$, and that for [14] is $O((mt + m + t)\log q)$, where q is related to the number of revoked nodes, k is related to the time window, and m is the life cycle (total session number) of the group communication. Obviously, the communication performance in our scheme is improved to a large extent, since the size of broadcast packet is reduced to $O(t \log q)$. Especially, the communication cost is independent of session number m. Thus, the optimized outcome is more distinct, especially when m becomes larger. The unicast communication overhead follows the same trend. Table 2 indicates that the two proposed schemes perform better in communication and storage overhead than the two schemes in [13, 14] do.

Additionally, the proposed HBT-based scheme reduces the communication and storage overheads without sacrificing any security property. However, the HBT is a two-dimensional data structure, so its implementation is expected to be a bit more complex than that of the DDHC-based scheme.

The effectiveness of the self-healing mechanism is also evaluated. Since both schemes share a similar concept, we focus on the scheme based on the DDHC. We first discuss the relationship between *RK* buffer size and communication overhead between the GKM and nodes, as shown in Fig. 5 with packet loss rate varying from 0.1 to 0.5. It can be seen that the key buffer length in each node determines the communica-

	Storage overhead	Communication overhead (broadcast)	Communication overhead (unicast)	Implementation complexity
HBT	Low	Low	Low	Medium
DDHC	Low	Low	Low	Low
[13]	High	High	High	—
[14]	Low	Medium	Low	—

Table 2. *Storage/communication overhead and implementation complexity.*

tion cost. A larger *RK* buffer can significantly reduce the communication overhead. Figure 6 depicts the computation cost as a function of the *RK* buffer length, where path loss p_l varies from 0.1 to 0.5 for $n = 500$. It can be seen that the computation cost of each node is low, since it only computes less than two hash functions per *RK* refreshment even in the worst case, $p_l = 0.5$.

From Figs. 5 and 6, the desirable number of RK buffers should be greater than 10 so that the normalized communication or computation cost is lower and in the range of 1∼1.5, which indicates that the proposed scheme is efficient in terms of communication and computation overhead, even in high packet loss or error rate environments.

Conclusion

We have proposed two novel group-wise key distribution schemes based on the DDHC and HBT, respectively, for secure group communications in WSNs. The proposed schemes offer self-healing group key distribution, which features periodic rekeying with implicit authentication and efficient tolerance for lost rekeying messages; and time-limited group node revocation so that forward and backward secrecy can be ensured. Performance and security evaluations demonstrate that storage, computation, and communication overheads of the two proposed schemes are quite low. The HBT-based key distribution scheme has stronger collusion resistance capability with a slight increase of implementation complexity as the trade-off. Both group-wise key distribution schemes are suitable for WSNs with frequent dynamic network topology changes.

Acknowledgment

This research has been supported in part by the NSFC under contracts no.60573144, 60218003, 60429202, 60673187, 60432030, and 90412012, Intel IXA University Research Plan, and a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship.

References

[1] I. F. Akyildiz *et al.*, "A Survey on Sensor Networks," *IEEE Commun. Mag.*, vol. 40, 2002, pp. 102–14.
[2] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) Specification," RFC 2093, 1997.
[3] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) Architecture," RFC 2094, 1997.
[4] Y. Challal and H. Seba, "Group Key Management Protocols: A Novel Taxonomy," *Int'l. J. Info. Tech.*, vol. 2, 2005, pp. 105–19.
[5] B. Dutertre, S. Cheung, and J. Levy, "Lightweight Key Management in Wireless Sensor Networks by Leveraging Initial Trust," Tech. rep., vol. SRI-SDL-04-02, 2004.
[6] M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System," *Eurocrypt '94*, 1994.
[7] M. Steiner, G. Tsudik, and M. Waidner, "Key Agreement in Dynamic Peer Groups," *IEEE Trans. Parallel and Distrib. Sys.*, 2000.
[8] D. Carman, B. Matt, and G. Cirincione, "Energy-Efficient and Low-Latency Key Management for Sensor Networks," *23rd Army Sci. Conf.*, 2002.
[9] C. K. Wong, M. G. Gouda, and S. S. Lam, "Secure Group Communications Using Key Graphs," *IEEE/ACM Trans. Net.*, vol. 8, 2000, pp. 16–30.
[10] Y. Nakamura and H. Kikuchi, "Efficient Key Management Based on the Subset Difference Method for Secure Group Communication," *Proc. 19th Int'l. Conf. Advanced Info. Net. and Apps.*, vol. 1, 2005, pp. 707–12.

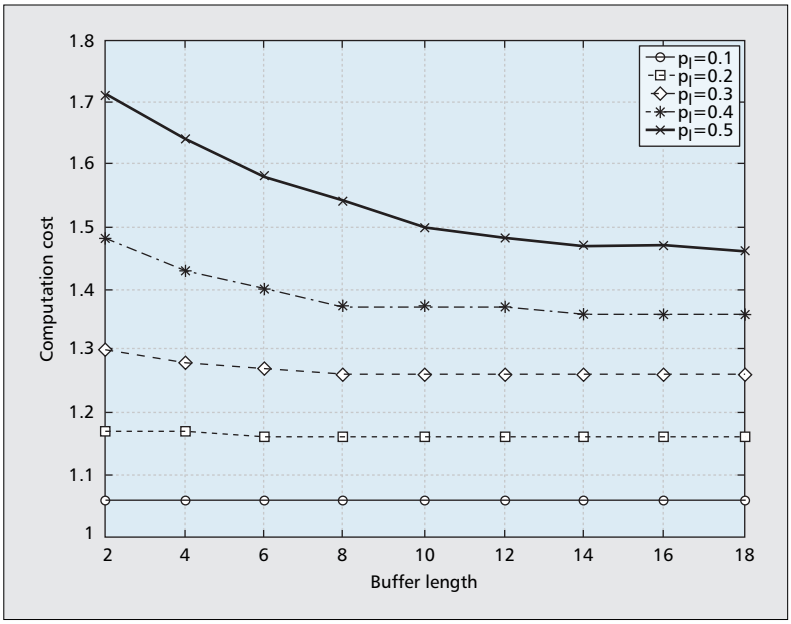


Figure 6. *Computation costs vs. key buffer length.*

[11] A. Fiat and M. Naor, "Broadcast Encryption," *Proc. Advances in Cryptology '93*, vol. 773, 1994, pp. 480–91.
[12] D. Naor, M. Naor, and J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," *Proc. Advances in Cryptology '01)*, *LNCS* 2139, 2001, pp. 41–62.
[13] J. Staddon *et al.*, "Self-Healing Key Distribution with Revocation," *Proc. IEEE Symp. Sec. and Privacy*, 2002, pp. 241–57.
[14] D. Liu, P. Ning, and K. Sun, "Efficient Self-Healing Group Key Distribution with Revocation Capability," *Proc. 10th ACM CCS*, 2003, pp. 231–40.
[15] Y. Jiang *et al.*, "Hash-Binary-Tree Based Group Key Distribution with Time-Limited Node Revocation," Tech. rep., 2006.

Biography

MINGHUI SHI (mshi@bbcr.uwaterloo.ca) received a B.S. degree (1996) from Shanghai Jiao Tong University, China, and an MASc. degree (2002) and a Ph.D. degree (2006) from the University of Waterloo, Ontario, Canada, all in electrical and computer engineering. He is currently with McMaster University, Ontario, Canada as an NSERC postdoctoral fellow and a research associate with the Centre for Wireless Communications, University of Waterloo. His current research interests include network security and mobility management in wireless LAN/cellular network integration, vehicular communications networks, and delay-tolerant networks.

YIXIN JIANG (yxjiang@csnet1.cs.tsinghua.edu.cn) received a Ph.D. degree (2006) from Tsinghua University, China. and an M.E. degree (2002) from Huazhong University of Science and Technology, both in computer science. In 2005 he was a visiting scholar with the Department of Computer Sciences, Hong Kong Baptist University. His current research interests include security and performance evaluation in wireless communication and mobile computing. He has published more than 20 papers in research journals and IEEE conference proceedings in these areas.

XUEMIN (SHERMAN) SHEN (xshen@bbcr.uwaterloo.ca) received a B.Sc. (1982) degree from Dalian Maritime University, China, and M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey, all in electrical engineering. He is with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, where he is a professor and the associate chair for graduate studies. His research focuses on mobility and resource management in interconnected wireless/wireline networks, UWB wireless communications systems, wireless security, and ad hoc and sensor networks. He is a coauthor of two books, and has published more than 200 papers and book chapters on wireless communications and networks, control, and filtering. He serves as Technical Program Committee Chair for

IEEE GLOBECOM '07, General Co-Chair for Chinacom '07 and QShine '06, and is Founding Chair of the IEEE Communications Society Technical Committee on P2P Communications and Networking. He also serves as a Founding Area Editor for *IEEE Transactions on Wireless Communications*; Editor-in-Chief for *Peer-to-Peer Networking and Application*; and as an Associate Editor for *IEEE Transactions on Vehicular Technology*, *KICS/IEEE Journal of Communications and Networks* (on computer networks); *ACM/Wireless Networks*; and *Wireless Communications and Mobile Computing* (Wiley). He has also served as Guest Editor for *IEEE JSAC*, *IEEE Wireless Communications*, and *IEEE Communications Magazine*. He received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 from the Faculty of Engineering, University of Waterloo. He is a registered Professional Engineer of Ontario, Canada.

CHUANG LIN [SM] (clin@csnet1.cs.tsinghua.edu.cn) is a professor of the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He received a Ph.D. degree in computer science from Tsinghua University in 1994. His current research interests include computer networks, performance evaluation, network security analysis, and Petri net theory and its applications. He has published more than 260 papers in research journals and IEEE conference proceedings in these areas, and has published three books. He is the Chinese Delegate in TC6 of IFIP. He serves as Technical Program Vice Chair for the 10th IEEE Workshop on Future Trends of Distributed Computing Systems; General Chair, ACM SIGCOMM Asia Workshop 2005; Associate Editor, *IEEE Transactions on Vehicular Technology*; Area Editor, *Journal of Computer Networks*, and Area Editor, *Journal of Parallel and Distributed Computing*.