ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

Secure Network Coding Approach With Distributed Reprogramming Protocol For Cluster Based Ad-hoc Networks In Dynamic Key Management Of Wireless Sensor Networks

Priyanka Manhas[1], Parminder Kaur[2]

Student, Department of Computer Science & Engineering, Chandigarh University (Gharuan, Mohali), India[1]

Assistant Professor, Department of Computer Science & Engg., Chandigarh University (Gharuan, Mohali), India[2]

Abstract: network coding provide set of secure protocols that rely on simple network coding operations to provide a robust and low-complexity solution for sharing secret keys among sensor nodes, including pairwise keys, cluster keys, key revocation, and mobile node authentication. We consider the problem of secret key distribution in a sensor network with multiple scattered sensor nodes. Our results include performance evaluation in terms of security metrics and a detailed analysis of resource utilization. For security reasons, every code update must be authenticated to prevent an adversary from installing malicious code in the network. All existing reprogramming protocols are based on the centralized approach in which only the base station has the authority to initiate reprogramming. it is desirable and sometimes necessary for multiple authorized network users to simultaneously and directly reprogram sensor nodes without involving the base station, which is referred to as distributed reprogramming. In this case, the network owner can also assign different reprogramming privileges to different users. Motivated by this consideration, we develop a secure and distributed reprogramming protocol named *SDRP*, The protocol uses identity-based cryptography to secure the reprogramming and to reduce the communication and storage requirements of each node. we address key management in cluster-based mobile ad hoc networks (MANETs). Ensuring secure communication in an ad hoc network is extremely challenging because of the dynamic nature of the network and the lack of centralized management. This scheme is implemented via a combination of ID-based multiple secrets and threshold cryptography. It eliminates the need for certificate-based authenticated publickey distribution and provides an efficient mechanism for key update and key revocation schemes, which leads to more suitable, economic, adaptable, scalable, and autonomous key management for mobile ad hoc networks. Many schemes, referred to as static schemes, have adopted the principle of key predistribution with the underlying assumption of a relatively static short-lived network. An emerging class of schemes, dynamic key management schemes, assumes long-lived networks with more frequent addition of new nodes, thus requiring network rekeying for sustained security and survivability. The theoretical analysyes demonstrates the security properties of the protocol, but we also implement it in a network of resource limited sensor nodes to show its high efficiency in wireless sensor networks.

Keywords: Authentication, ID-Based Cryptography, Key Management, Mobile Ad Hoc Network, Network Coding, Reprogramming, Secret Key Distribution, Sensor Networks, Secret Sharing ,Security, Wireless Sensor Networks (Wsns)

I INTRODUCTION

Secret key distribution schemes that specifically exploit the availability of mobile nodes thus far seem elusive. To the best of our knowledge, the same could be said about the use of network coding [8], [9] (or, equivalently, algebraic mixing of data packets) toward accomplishing secret key distribution tasks. The concept of weak network coding security is introduced in [10] to describe the level of secrecy that is achievable for message transmission in a multicast scenario where an attacker only observes linear combinations of data packets and not the data packets themselves. Our contribution differs from the work in [10] in several important aspects: 1) we focus on uniformly distributed secret keys, for which strong information-theoretic security guarantees can be given and 2) we target a large class of sensor networks where we make explicit use of mobility and of the broadcast property of the wireless medium. In addition to the basic secret key distribution scheme, we also include:

- *Authentication, clustering, and key revocation:* we provide additional protocols that cover authentication of the mobile node, generation of cluster keys, and revocation in the case of compromised sensor nodes.
- *Key renewal for robustness and scalability in dynamic environments:*if the network topology changes rapidly or new nodes enter the network, new keys can be safely distributed with a simple procedure even when the sets of prestored keys have been depleted.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

- *Performance evaluation:* we provide a thorough analysis of the security performance of our scheme by discussing its behavior under various attack models and proving mathematically that the encrypted keys stored by the mobile node are information-theoretically secure.

When the base station wants to disseminate a new code image to certain sensor nodes, it transmits the signed code image to those nodes via multihop routing, and those nodes only accept the code image signed by it. Unfortunately, the centralized approach is vulnerable to the single point of failure and not reliable because reprogramming becomes impossible when the base station fails or when some nodes lose connections to the base station. Also, it is inefficient, weakly scalable, and vulnerable to potential attacks along the long communication path [18]. The base station has to be online and accessible to any user at any time during the network operation.Even worse, there are some WSNs that do not have any base station. Examples of such networks include a WSN deployed along an international border to monitor weapon smuggling and human trafficking. Having a base station in these WSNs introduces a very attractive attack target. Obviously, for such networks, it is necessary to have authorized network users to be able to carry out reprogramming in a distributed manner. Another advantage of distributed reprogramming is that, while multiple authorized users are supported, each user may have a different privilege of reprogramming sensor nodes. This is particularly important in large-scale sensor networks owned by an owner and used by different users from both public and private sectors. we propose an ID-based multiple secrets key management (IMKM) protocol to address all the above concerns. Our scheme is a comprehensive solution for inter and intra-cluster key management, including key revocation, key update, and group key agreement. objective of key management is todynamically establish and maintain secure channels among communicating parties. Communication keys may be pair-wise keys used to secure a communication. channel between two nodes that are in direct or indirect communications [1–4], or they may be group keys shared by multiple nodes [5,6]. Network keys (both administrative and communication keys) may need to be changed (rekeyed) to maintain secrecy and resilience to attacks, failures, or network topology changes.

The rest of the paper is organized as follows: section 2 describe the detailed description of SDRP requirements and security analyses with design considerations of distributed reprogramming. In section 3,we propose protocol analyses ID based multiple secret key management(IMKM). System model with detail description explained in section 4.in section 5,we illustrate the difference of static and dynamic key management scheme with example of LOCK for dynamic key management scheme.finally,paper conclude in section6 with future directions.

II SDRP REQUIREMENTS AND SECURITY ANALYSES WITH DESIGN CONSIDERATIONS OF DISTRIBUTED REPROGRAMMING:

The distributed reprogramming approach is more suitable for WSNs. It allows authorized network users to simultaneously and directly update code images on the nodes without involving the base station. Unfortunately, to the best of our knowledge, distributed reprogramming in WSNs has so far received no attention, despite a rich literature on the centralized approach. Similar to the centralized reprogramming protocols, a secure distributed reprogramming protocol should satisfy the following requirements:

SDRP requirements:

- *Freshness:* An earlier version of a program image cannot be installed over the program with the same or greater version number, ensuring that a node always installs the newest version of a program image.
- *Node compromise tolerance:* A compromised node must be prevented from causing an uncompromised node to violate the aforementioned security requirements.
- *Authenticity and integrity of code images:* The source of a program image must be verified by a sensor node prior to installation, ensuring that only a trusted source can install a program. In addition, integrity means that an updated program image cannot be modified undetectably Other than meeting the aforementioned requirements, a distributed reprogramming protocol should also have the following properties.
- *Distributed:* The authorized network users are able to simultaneously and directly update code images on the nodes without involving the base station. At the same time, the protocol should prevent unauthorized users from updating sensor nodes.
- *Partial reprogram capability:* To prevent sensor nodes from being totally controlled by network users, the special modules (e.g., authentication module for each new program image) on each sensor node cannot be overwritten by anyone except the network owner.
- *Supporting different user privileges:* To ensure smooth functioning for a WSN, the level of each user privilege should be limited by the network owner. For example, a user is only allowed to reprogram the sensor nodes set with specified identities or/and within a particular localized area during his subscription period.
- *Scalability:* First, the protocol needs to be efficient even in a large-scale WSN with thousands of sensor nodes, and second, the protocol should be able to support a large number of users.
- *User traceability:* In most application scenarios, traceability is highly desirable, particularly for reprogramming.
- *Being efficient:* Mobile devices, particularly sensor nodes, usually have limited resources (e.g., CPU processing power, memory, bandwidth, and energy). Thus, energy efficiency (with respect to both

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

communication and computation) and small storage overhead should be given priority to cope with the resource-constrained nature of WSNs.

SECURITY ANALYSES OF SDRP:

A. Authenticity and Integrity of Code Images the nodes can authenticate each hash packet in page 0 once they receive such packets, based on the security of the Merkle hash tree. The hash packets include the hash values of the data packets in page 1. Therefore, after verifying the hash packets, a node can easily verify the data packets in page 1 based on the oneway property of hash functions. Likewise, once the data packets in page i are verified, a sensor node can easily authenticate the data packets in page $i + 1$, where $i = 1, 2, \ldots, Y - 1$. In summary, if an adversary injects a forged modified program image, each receiving node can detect it easily because of the (immediate) authentication of reprogramming packets.

B. Ensurance of Freshness: Obviously, there are two cases for the network users to administrate the program update of a WSN. In the first case, each network user has the privilege to reprogram the sensor nodes in different zones (or different sets of sensor nodes according to their identities), and there exists no sensor node which is allowed to be reprogrammed by two network users. In step 1) of the sensor node verification phase, a sensor node first checks whether the version number from the received message m is valid. Only if it is valid, the verification procedure goes to the next step. Therefore, the use of the version number of the updated program image can ensure the freshness of SDRP. The other case is that a sensor node may be assigned to multiple network users by the network owner. A feasible approach for achieving the freshness is that a timestamp is used instead of the version number of the updated code image. In step 1) of the sensor node verification phase, a sensor node first checks whether the timestamp included in the message*m*is fresh. This can ensure that a node always installs the most recent version of a program. In this case, we assume that the WSN is loosely synchronized via some existing efficient time synchronization mechanism

C. Resistance to Node- and User-Compromised Attacks: the adversary cannot impersonate any authorized network user by compromising sensor nodes. In other words no matter how many sensor nodes are compromised, a benign sensor node will not grant the adversary any reprogramming privilege.

D. Distributed: Here, it is demonstrated that the network owner can enforce strict reprogramming so that the reprogramming privilege is only accessible to users willing to register. In addition, it is clear that the authorized users are able to carry out reprogramming in a distributed manner.

E. Supporting Different User Privileges: nobody except the network owner can modify *Prij* contained in the signature message and then pass the verification from the sensor nodes.where *Prij* is reprogramming previelage.

F. User Traceability: In many application scenarios, traceability is highly desirable, particularly for reprogramming, where it is used for collecting the network users' activities for some purposes. For instance, with the knowledge of the network users' reprogramming history, the network owner is able to find out which nodes are frequently reprogrammed, and then can improve the network deployment.

Design Consideration Of Distributed Reprogramming

centralized reprogramming protocol involves only two kinds of participants, the base station (administered by the network owner) and all sensor nodes. Only the base station can reprogram sensor nodes. Different from the centralized approach, a distributed reprogramming protocol consists of three kinds of participants, the network owner, authorized network users, and all sensor nodes. Here, the network owner can be offline. Also, after the users register to the owner, they can enter the WSN and then have predefined privileges to reprogram the sensor nodes without involving the owner. To provide secure and distributed reprogramming, a naïve solution is to pre-equip each sensor node with multiple publickey/ reprogramming-privilege pairs, each of which corresponds to one authorized user. This scheme allows a network user to sign a program image with his private key such that each sensor node can verify whether the program image originates from an authorized user. However, this solution is not applicable to WSNs due to the following facts. First, resource constraints on sensor nodes often make it undesirable to implement such an expensive algorithm. For the RSA-1024 public-key cryptosystem (1024-b keys), the length of each public key is more than 1026 b. Additionally, for the ECC-160 [29] public-key cryptosystem (160-b keys), the length of each public key is 1120 b. Assuming that the length of reprogramming privilege is 32 B and either RSA-1024 or ECC-160 is used, the length of each public-key/reprogramming-privilege pair is more than 160 B. This means that not too many public-key/reprogramming privilege pairs can be stored in a sensor node.

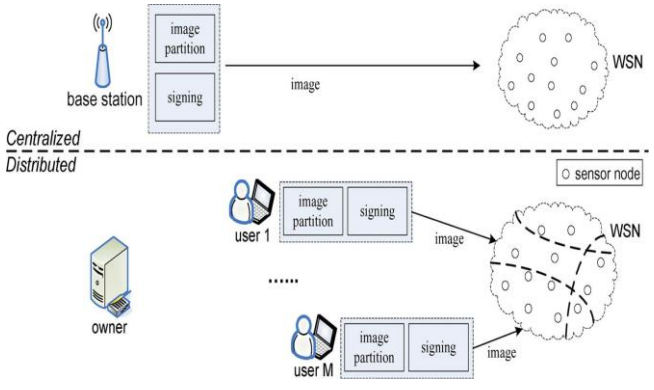

Fig. 1 Design Consideration Of Distributed Reprogramming

III PROTOCOL ANALYSES ID BASED MULTIPLE SECRET KEY MANAGEMENT(IMKM)

We separate our discussion into two parts: share key distribution and group key distribution.

A. Security Analysis

A.1. Share Key Distribution

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

We compare the security of our IMKM scheme to that of RSA certificate-based cryptography (RCBC), such as MOCA [11], URSA [12] or AKM [15] and ID-based cryptography, such as IBC-K [16].

- These five approaches are all based on (t, n) threshold schemes. The master secret key is spread over the initial set of nodes, enhancing intrusion tolerance. This makes the service robust in the sense that an adversary must compromise a minimum of t nodes in order to recover the master secret key. This also reduces vulnerability, as the service is available as long as t correctly-behaving shareholders are within reach. However, when adversaries compromised more than t–1 D-CAs or D-PKGs they are then able to construct the CA's private key, in the case of RCBC, or the PKG's master secret key, in the case of IBC-K. In contrast, in the case of IMKM, the master secret key is generated by the collaboration of all uncompromised D-PKGs. Therefore, the overall system security is still guaranteed even when t shareholders are compromised.
- The RCBC and IBC-K master secret keys are generated by a centralized authority and remain static afterward. Nevertheless, the primary function of the centralized PKG in IMKM is to preload each node with public/ private key pairs and all system parameters, except for the master secret key. IMKM is a fully distributed key management scheme; all the uncompromised DPKGs are required to participate in the construction of the master secret key, therefore, even compromise of the PKG does not reveal the master secret key. In addition, with the IBC-K scheme, the personal private key of a newly joined node is accumulated from t shareholders, therefore, a secure channel is required, such as physical contact or a dedicated communication channel, otherwise, the system is vulnerable to passive eavesdropping or man-in-the-middle attacks.

A.2. Group Key Distribution

The proposed authenticated group key agreement (AGKA) protocol satisfies the following security attributes [43]-[44]: Implicit Key Authentication: The pair-wise key is computed by each CH's ephemeral and long-term private keys, as described in session IV-A.2. Therefore, the CHs are assured that no other CHs can obtain the pair-wise keys, except for their partners that have the private keys. Our group key is computed using each participant's pairwise keys, so it inherits an implicit key authentication property.Only those who have all the correct corresponding pair-wise keys can generate the group session key. Known Session Key Security: Each execution of the protocol computes a unique session key, which depends on the ephemeral key, *Li.* Consequently, compromise of past session keys does not result in the compromising of future session keys. Backward and Forward Secrecy: Forward secrecy prevents a user who has left a secure group from accessing future secure keys. Backward secrecy prevents a newly joined user from accessing past secure keys. Our AGKA protocol supports both of these properties. No Key-compromise Impersonation: Suppose the longterm private key of a member is compromised, an adversarycan then impersonate that member in this protocol; however, the adversary cannot impersonate other members.No Unknown Key-share: In an unknown key share attack, an adversary convinces a group of entities that they share a key with it, whereas, in fact, the key is shared between the group and some other party. This attack is unlikely to work unless the adversary obtains the pair-wise keys of some entity. No Key Control: all members determine the group session key in the protocol, so that no single party can control the outcome. No single party can restrict the range of the group key to some predetermined value.

IV SYSTEM MODEL

A. Energy Consumption Model

Sensors consume energy for sensing, receiving, transmitting, data processing, and also during the idle mode when no data sensing, processing or exchange happens. Data transmission can be accomplished with a fixed or adjustable power. Utilizing the power adjustment mechanism can benefit the energy conservation in the network [5] at the expense of a more complex hardware. The proposed analysis in this paper is mainly based on adjustable transmission power. Extending the analysis to multilevel transmission power (e.g., for Mica Mote sensors [17]) or fixed transmission power is straight forward and will be briefly discussed in Section 7. Assuming adjustable transmission power proposed in [5], the transmission energy for one packet can be modeled as etðdÞ ¼ kd_ þ c; ð1Þ where _ stands for the path loss exponent and d refers to the distance between the sensor and destination. Also, k and c represent the loss coefficient and the overhead energy for one packet, respectively. Usually, _ is considered to be two for small distances and four for large distances [3], [18]. Assuming that the intracluster distances are small and intercluster distances are large, similar to [3], we use _ ¼ 2 for intracluster and _ ¼ 4 for intercluster transmissions.

Consumed energy for receiving, er, is almost independent of the distance between the transmitter and receiver and depends on the electronic parts of the receiver. In addition, we assume that the idle mode consumed power, used to keep the radio part on to listen to the channel, is almost fixed.

B. Clustering Model

Clustering is proposed for WSNs to decrease the energy consumption and ease the network management, [1], [2]. Usually, the nodes within a cluster send their data to the CH and then CH performs necessary data processing andaggregation before relaying it toward the data sink. Here, we assume that N sensors are deployed randomly over the cluster. First, we focus on the case where all transmissions are in single-hop mode. Later on, the multihop mode for intercluster transmissions (CH to sink) will be discussed. For multihop intercluster transmissions, it is assumed that a CH forwards its data toward the sink through other CHs, i.e., other nodes are not involved in intercluster data communication. It is also assumed that

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

the network has a static clustering, meaning that the shape of the clusters are fixed during the network operation [14], [15]. While dynamic clustering allows for a more flexible network design, its overhead to form the clusters is considered a serious drawback [3], [16]. For many practical situations, therefore, static clustering is an attractive solution [13].

C. Lifetime Definition

The network lifetime is the time interval over which the network can operate effectively. Clearly, a more specific definition of the lifetime is possible only based on the network application. Hence, different lifetime definitions are used in the literature. For instance, in [26], and in [6] the network connectivity is used to define the lifetime. Another commonly used definition is based on the percentage of the dead nodes [8], [27], i.e., the network lifetime is the moment when the number of live nodes drops below ð1 _ _ÞN where 0 _ _ < 1 and N stands for the total number of nodes in the network. Here, we adopt the last definition for the cluster lifetime. The lifetime definition based on the percentage of the dead nodes also includes the lifetime definition based on the first
node death [28], [29]. Moreover, since the number of dead nodes can reflect the quality of the network coverage and/ or connectivity [30], this lifetime definition can also be considered as an approximation of the lifetime based on the network coverage and/or connectivity

D. Event Occurrence Model

There exist three main models for packet generation in WSNs, namely, event-driven, time-driven, and querydriven [19]. In the time-driven case, sensors send their data periodically to the sink. Event-driven networks are used when it is desired to inform the data sink whenever a random events occurs. In query-driven networks, sink sends a request of data gathering when needed.Based on the network application, data characteristic, and the type of data inquiry, usually one of these models is utilized to characterize the sensors traffic generation. Here, the proposed analysis mostly concentrates on the eventdriven networks. As discussed in Section 7, for the purpose of our analysis, event-driven networks comprise a rather general case. In fact, time-driven networks can be studied as a special case of the proposed analysis.

E. Media Access Control (MAC) Protocol

It is common to use time division multiple access (TDMA) technique for MAC in clustered WSNs [16], [31]. As an example, low-energy adaptive clustering hierarchy (LEACH) uses TDMA to enable multiple access within a cluster [3]. In this case, CH usually coordinates the time scheduling among the nodes. To avoid intercluster interference, code division multiple access (CDMA) might also be used [3]. Here, we assume that TDMA is used in the clusters. Thus, collisions are avoided and transmissions can be assumed ideal [31].

V DIFFERENCE OF STASTIC AND DYNAMIC KEY MANAGEMENT

	STATIC KEY MANAGEMENT	**DYNAMIC KEY MANAGEMENT**
NETWORK LIFE	Assume short lived	Assumed long lived
KEY ASSIGNMENT	Once predeployment	Multiple times post deployment
KEY GENERATION	Once predeployment	Multiple times post deployment
KEY DISTRIBUTION	All nodes predistributed to nodes prior to deployment	Subsets of keys are re-distributed to some nodes as needed
KEY POOL	Very large key pool,static administrative key values	Small size pool,dynamic administrative key values
RE-KEYING COST	May be practically infeasible with respect to number of messages	Requires few messages
COMMUNICATION COST	Not applicable for administrative keys	Re-keying overhead
STORAGE COST	More keys per node	Fewer keys per node
NETWORK RESILIENCE	High as long as number of nodes captured is small. Once threshold is exceeded, resilience falls sharply	High,largely independent of number of nodes captured as long as rekeying is performed is performed in a timely manner
NETWORK CONNECTIVITY	Less connected due to large key pool. Connectivity improves with increasing number of keys per node	More connected due to small-size key pool
HANDLING NODE ADDITION	New node preloadedwith keys from static pool may decrease network resilience to node capture	New node receives new set of keys, other nodes may be rekeyed-less impact on network resilience to node capture

Overview Of Lock: An Example Of Dynamic Key Management Schemes

LOCK is an EBS-based dynamic key management scheme for clustered sensor network. The physical network model is a three-tier wireless sensor network with the base station (BS)at the top, followed by cluster leader nodes (CLs) , then regular sensor nodes. In LOCK, no pre-deployment information is assumed about the expected locations of the nodes. LOCK uses two layers of EBS administrative keys. The upper layer (level 1) is *EBSb* that enables the base station to manage the cluster leaders as a group. The lower layer (level 0) involves an *EBSCi* for each cluster *Ci*. A cluster leader, *Ci*, is a member in both the upper *EBSb* as well as the lower *EBSCi*. We assume that the capture of a CL is as likely as any other sensor node. Administrative keys of each *EBSCi*, in turn, are used to construct (and refresh) cluster session keys used by the cluster leader to communicate with the sensor nodes within the cluster. The CL is considered a regular member in its *EBSC* that knows as many cluster administrative keys as any other node in the same cluster (k keys out of $k + m$ keys). Accordingly, the capture of a CL does not provide the attacker with any more cluster keys than the capture of a regular sensor node (compare with SHELL [6] where CLs store sensor node keys and hence their capture may cause more harm). During the initialization phase, sensor nodes of each cluster establish a set of backup keys (one chain of keys for each cluster) shared with the base station and unknown to their (or any other) cluster leader. Key generation of *EBSC* keys is performed by a group of sensor nodes within

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

the cluster called key generation nodes (KGNs), which are usually selected by the CL. Key distribution is performed by the CL. The capture of a sensor node (or a KGN) is handled solely by a local rekeying mechanism within the cluster to exclude the captured node from EBSC. LOCK does not involve the base station or any intercluster leader communication to generate new cluster keys as in SHELL. The capture of a cluster leader is first handled by the base station through EBS rekeying at the clusterleader level through EBS eviction. Sensor nodes within the cluster whose leader is captured also use EBS re-keying to exclude such node form the cluster. The base station might later deploy a replacement cluster leader or redistribute nodes. Deploying a new cluster leader for a cluster whose leader has been captured requires thecluster's sensor nodes to authenticate the new leader using the backup keys shared with the base station. It is worth mentioning that unlike other dynamic schemes, the capture of any node in LOCK (including cluster leaders) does not affect the normal operation of other clusters. Since location information is not necessarily available in most sensor networks given the mass deployment in many applications, we propose the use of key polynomials in LOCK to improve network resilience to collusion instead of location-based key assignment as in SHELL. As shown in the next section, using such a technique can significantly enhance network resilience to node capture while using lower degree polynomials compared to static keying. In the next section, we discuss general performance and security of static and dynamic key management schemes.

VI CONCLUSION

We presented a secret key distribution scheme for large sensor networks. Unlike [3] and [6], this is not a probabilistic scheme. More specifically, any two nodes that can reach each other can communicate securely with probability one, using a small number of prestored keys albeit at the expense of a mobile node for bootstrapping.We presented several security extensions that exploit network coding to provide secret key distribution in large and dynamic sensor networks. a number of secure reprogramming protocols have been proposed, but none of these approaches support distributed operation. Therefore, in this paper, a secure distributed reprogramming protocol named SDRP has been proposed. In addition to analyzing the security of SDRP, this paper has also reported the evaluation results of SDRP in an experimental network of resource-limited sensor nodes, which shows that SDRP is feasible in practice. In some applications, data are also required to be kept confidential due to the possibility of message interception. In future work, we will study how to support confidentiality in distributed reprogramming.

REFERENCES

[1] Y. Zhou, M. Hart, S. Vadgama, and A. Rouz, "A Hierarchical Clustering Method in Wireless Ad Hoc Sensor Networks," Proc. IEEE Int'l Conf. Comm. (ICC '07), pp. 3503-3509, June 2007.

[2] C. Liu, K. Wu, and J. Pei, "An Energy-Efficient Data Collection Framework for Wireless Sensor Networks by Exploiting Spatiotemporal Correlation," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 7, pp. 1010-1023, July 2007.

[3] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Trans. Wireless Comm., vol. 1, no. 4, pp. 660-670, Oct. 2002.

[4] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," IEEE Trans. Mobile Computing, vol. 3, no. 4, pp. 366-379, Oct.-Dec. 2004.

[5] W.B. Heinzelman, "Application-Specific Protocol for Wireless Networks," PhD dissertation, Massachusetts Inst. of Technology, 2000.

[6] P. F. Oliveira, R. A. Costa, and J. Barros, "Mobile secret key distribution with network coding," presented at the Int. Conf. Security Cryptography, Jul. 2007.

[7] P. F. Oliveira and J. Barros, "Network coding protocols for secret key distribution," in *Proc. Int. Symp. Information Security*, Nov. 2007, pp. 1718–1733.

[8] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 2, pp. 228–258, 2005.

[9] D. Malan, M. Welsh, and M. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," presented at the 1st IEEE Int. Conf. Sensor and Ad Hoc Communications and Networks, Santa Clara, CA, Oct. 2004. [Online]. Available: citeseer. ist.psu.edu/malan04publickey.html.

[10] A. Perrig, R. Szewczyk, J. D. Tygar, V.Wen, and D. E. Culler, "SPINS: Security protocols for sensor networks," *Wireless Netw.*, vol. 8, no. 5, pp. 521–534, 2002.

[11] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. 9th ACMConf. Computer and Communications Security*, New York, 2002, pp. 41–47.

[12] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *Proc. 10th ACM Conf. Computer and Communications Security*, New York, 2003, pp. 62–72.

[13] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Trans.Ind. Electron.*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[14] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3557–3564, Oct. 2010.

[15] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun, "Distributed collaborative control for industrial automation with wireless sensor and actuator networks," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 4219–4230, Dec. 2010.

[16] X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Building-environment control with wireless sensor and actuator networks: Centralized versus distributed," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3596–3605, Nov. 2010.

[17] J. Carmo, P. Mendes, C. Couto, and J. Correia, "A 2.4-GHz CMOS shortrange wireless-sensor-network interface for automotive applications," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1764–1771, May 2010.

[18] Crossbow Technology Inc., Milpitas, CA, Mote In-Network Programming User Reference, 2003.

[19] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proc. ACM SenSys*, 2004, pp. 81–94.

[20] V. Naik, A. Arora, P. Sinha, and H. Zhang, "Sprinkler: A reliable and energy efficient data dissemination service for extreme scale wireless networks of embedded devices," *IEEE Trans. Mobile Comput.*, vol. 6, no. 7, pp. 777–789, Jul. 2007.

[21] TinyOS: An open-source OS for the networked sensor regime. [Online]. Available: http://www.tinyos.net/

[22] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proc. NSDI*, 2004, p. 2.

[23] J. Deng, R. Han, and S. Mishra, "Secure code distribution in dynamically programmable wireless sensor networks," in *Proc. ACM/IEEE IPSN*, 2006, pp. 292–300.

[24] P. K. Dutta, J. W. Hui, D. C. Chu, and D. E. Culler, "Securing the deluge network programming system," in *Proc. ACM/IEEE IPSN*, 2006, pp. 326–333.