A Key Distribution Scheme Using Network Coding for Mobile Ad hoc Network

Jianwei Liu
School of Electrical and Information Engineering
Beihang University, Beijing, China
liujianwei@buaa.edu.cn

Ruiying Du
School of Computer Science
Wuhan University, Wuhan, China
duraying@gmail.com

***Abstract*— Network coding offers an excellent solution for maximizing throughput in various networks. Because of its simplicity and high efficiency, the idea of network coding can also be used for designing a lightweight key distribution schemes for wireless ad hoc network. This paper presents a key distribution scheme that exploits the inherent security properties of network coding. The new scheme relies on simple XOR network coding operations to provide data confidentiality and uses message authentication codes (MACs) to guarantee the integrity of the distributed keys. We also show that our scheme can resist a series of attacks suffered in wireless ad hoc network and has better performance in comparing with previous schemes proposed in the literature.**

Keywords-network coding; key distribution scheme; message authentication code(MAC); wireless ad hoc network

I. INTRODUCTION

Since network coding approach was first proposed by R. Ahlswede, Li, Cai and Yeung in their pioneering work in 2000[1], a few of further studies have begun to investigate how to exploit the network coding idea to design secure lightweight protocols for a lot of applications. A few of papers deal with network coding security problems. L. Lima, J. P. Vilela, P. F. Oliveira and J. Barros discussed the attacks and countermeasures in wireless network coding [2]. J. Dong et al identify some security threats and challenges in several network coding-based systems proposed for unicast in wireless network[3]. P. F. Oliveria and J. Barros proposed a secret key distribution protocol for wireless networks based on network coding[4] and C. Gkantsidis and P.R. Roddriguez proposed a large scale contents distribution scheme [5] in network scenarios. J. P. Vilela, L. Lima and J. Barros proposed a low-complexity cryptographic scheme [6] based on random linear network coding [7]. Z. Yu, Y. Wei, B. Ramkumar and Y. Guan proposed an efficient XOR network coding scheme to combat against pollution attacks[8]. S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Medard proposed the algorithms to resist Byzantine attacks[9].

While in mobile ad hoc network, its dynamic network topology, multi-hop, centerless and self-organizing properties pose even more serious security challenges than those in static networks [10]. One of the most important problems is how to distribute and update secret keys to ensure secure communication among all enrolled nodes. A network coding-based protocol is proposed for wireless sensor network [11]. In the scheme, the authors suppose there is a mobile node in the static sensor network. Obviously, the scheme can not meet the security requirements in ad hoc network, because all nodes in ad hoc network are mobile, and so-called neighbors of one node are not fixed any more.

Figure 1 shows a cluster-based topology of wireless general multi-hop ad hoc communication network, where the black dots indicate the clusterheads and the black lines indicates the communication route from node A to node B.

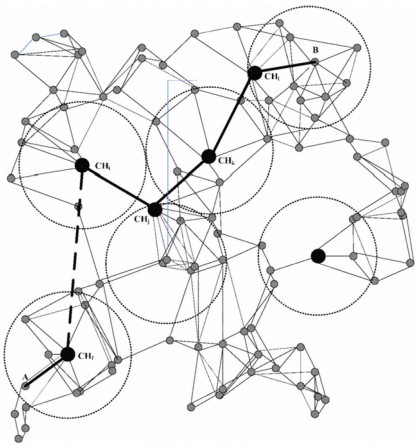

Figure 1 A cluster-based topology of ad hoc network

In this paper, we propose a new key distribution scheme for mobile ad hoc network. Our scheme is based on network coding paradigm. The scheme allows any two nodes to setup a shared key through a multi-hop route efficiently.

Our scheme adopts a trusted third party (TTP) to pre-install a secret key and all padded key materials of the other nodes to each ad hoc node in the initialization stage. Each node only knows its own secret key. Besides, it also keeps an encrypted version of keys of all other nodes pre-installed by TTP in the initialization stage. After the initialization stage, end-to-end key distribution can be performed efficiently based on network coding paradigm.

The rest of the paper is organized as follows: we present a security model and some reasonable assumptions in section II, and explain the symbols we use in the paper in section III. We propose our scheme in section IV and analyze its security and performance in section V. Section VI concludes the paper.

This paper is supported by the NSFC and 863 project of China

CHINACOM 2010, August 25-27, Beijing, China

DOI 10.4108/chinacom.2010.85

II. SECURITY MODEL

A. Network Topology Model

We consider a cluster-based ad hoc hierarchical network topology. A subset of the network nodes is selected to serve as the network backbone over which essential network control functions are supported. The approach to topology control is often called clustering, and consists of selecting a set of clusterheads in a way that every node is associated with a clusterhead, and clusterheads are connected with one another directly or by means of gateways, so that the union of gateways and clusterheads constitute a connected backbone. Once elected, the clusterheads and the gateways help reduce the complexity of maintaining topology information, and can simplify such essential functions as routing, bandwidth allocation, channel access, power control or virtual-circuit support. For clustering to be effective, the links and nodes that are part of the backbone (i.e., clusterheads, gateways, and the links that connect them) must be close to minimum and must also be connected [12].

Figure 2 illustrates a 2-layer hierarchical network topology for ad hoc network.

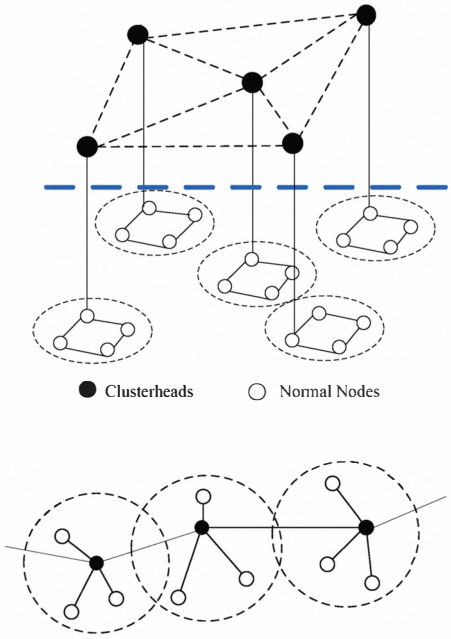

Figure 2 A hierarchical network topology

From Figure 2, we can learn that each clusterhead has control ability over all the other normal nodes within the cluster. The clusterheads are connected with each other to perform traffic delivery among nodes in different clusters. The characteristics of cluster-based topology of ad hoc network can be leveraged to distribute secret keys based on network coding paradigm.

B. Threat Model and Goal

We consider the security threats posed by an attacker in ad hoc network have the following characteristics:

1) He can eavesdrop every wireless link in the network;
2) He has full access to all data traffic and can perform analysis upon receiving the traffic.
3) He knows all the cryptographic algorithms used in the network, but he has limited computing resources and thus unable to break the cryptographic primitives.
4) He can inject bogus traffic, and modify traffic to launch impersonation attack.
5) He can capture some ad hoc nodes and extract authentication/encryption keys from the compromised nodes.

Our goal is to design a network coding-based scheme that can efficiently set up a secret key between two communication nodes, or set up a conference key among a group of nodes. We particularly address that the 2-layer topology should be adopted for ad hoc network, which can be greatly benefited from XOR operations in network coding paradigm.

C. Initial Assumptions

We make some reasonable assumptions for the scheme.

1) In the initialization stage, there exists an offline trusted third party (TTP) in the network.
2) Each ad hoc node has enough memory to store all the encrypted keys of whole network nodes.
3) One clusterhead knows all identifiers of nodes within its cluster and can route the traffic to other clusterhead, and the latter will deliver the data to the designated node in the other cluster.

Clearly, the first and the third assumptions are not difficult for us to understand. Some people may argue that the second assumption seems unreasonable, because terminals of ad hoc network have limited memory resource. Actually, unlike wireless sensor network, mobile ad hoc network usually has limited number of nodes under some military or industry scenarios, i.e., military rescue action and geological prospecting and exploration. We suppose that the secret key size is of 128 bits (16 bytes), and the node identifier is of 16 bits (2 bytes, which can represent to the maximum 65536 ad hoc nodes), then the memory for each node needs approximately 1M bytes. It is obviously affordable for a mobile ad hoc node with the technical advancement of storage device.

III. NOTATION AND SYMBOL

Before we begin to describe our proposed schemes, we explains the symbols used in the paper.

Table 1 lists the symbols and their corresponding meanings in our scheme.

TABLE I. NOTATION AND SYMBOLS

Symbol	Description
K_i	the secret key of i-th Ad hoc node
ID_i	the identifier of i-th Ad hoc node
R	random number generated by TTP
$h(x)$	secure hash function used to generate a MAC
r_i	random challenge generated by i-th Ad hoc node

MAC_i	message authentication code using i-th node's key
$\|\|$	message concatenation operation
H_l	l-th clusterhead
$P, \|P\|$	global key pool and its size
N	maximum number of Ad hoc nodes
SK	shared secret key between two Ad hoc nodes

IV. Our New Key Distribution Scheme Using Network Coding For Manet

In this section, a new key distribution scheme is proposed based on network coding paradigm. Because the XOR operations are used in the scheme, it requires only a few lightweight computations and provides a level of security of probabilistic key sharing scheme [13].

We will describe these 3 phases in detail in the following.

A. The Framework for Key Distribution in Ad hoc Network

Before describing our proposed scheme, we first propose a framework for securely distributing secret keys in mobile Ad hoc network.

Our proposed scheme includes 3 phases. The first one is the *initialization phase*. The second one is the *key distribution phase*. And the third one is the *key updating phase*.

- *Initialization phase*: In this phase, we suppose there is a offline trusted third party (TTP) in Ad hoc network, which is responsible for security parameter setup, such as generating secret key for each node, and choose cryptographic hash functions and algorithms. The TTP will initialize every Ad hoc node and injects the security data into its memory. Once this phase is finished, all network nodes are ready for deployment.
- *Key distribution phase*: Two kinds of protocols will be executed based on whether two communication nodes belong to a same cluster or not. If the two nodes belong to the same cluster, then key distribution can be easily done by the aid of the clusterhead. Whereas, if the two nodes belong to different clusters, the key distribution will be realized by the aid of two different clusterheads, which take the effect of gateways.
- *Key updating phase*: When Ad hoc network topology changes dynamically or there are new nodes entering the network, new keys should be securely distributed efficiently. When an Ad hoc node wants to update its current secret key, it needs to send a update request to its clusterhead. Then key updating procedure will be executed with the aid of clusterheads.

B. Detailed Procedure of Our Key Distribution Scheme

We assume that 2-layer hierarchical topology model is adopted, and the clusterheads can be elected through a recommendation algorithm automatically [12], and every Ad hoc node is associated with a clusterhead. Thus, once any two noses want to setup a common secret key and communicate securely, they must first contact their own clusterheads. The clusterheads taking the effect of gateways can compute and deliver data between the two communication nodes.

There are two cases here. The first case is two nodes are associated with one same clusterhead. The second case is two nodes are associated with two different clusterheads. Therefore, we will propose two different key distribution protocols to meet the above two cases in the following.

***Initialization phase*:** The offline trusted third party (TTP) in the network generates a random number R meeting *Bernoulli* (1/2) distribution, a secret key $K_i \in P$, where P is the large key pool generated by TTP, and the corresponding identifiers $ID_i, i \in \{0,\cdots,N-1\}$ for each Ad hoc node. TTP stores K_i and a list of an encrypted version of the other node's keys $K_j \oplus R, j = 1,\cdots,N, j \neq i$ into node i alone with all corresponding identifiers of the Ad hoc nodes. Then TTP choose a secure hash function $h(x)$.

Note that, after the initialization phase, each node only knows its own secret key and doesn't know the secret keys of other nodes. This will minimize the risk of secret key leakage when one node is captured and compromised.

***Key distribution phase*:** After Ad hoc node deployment, there are two cases that have already been addressed above.

(i) Case 1: Two Ad hoc nodes belong to the same clusterhead. Figure 3 shows the case.

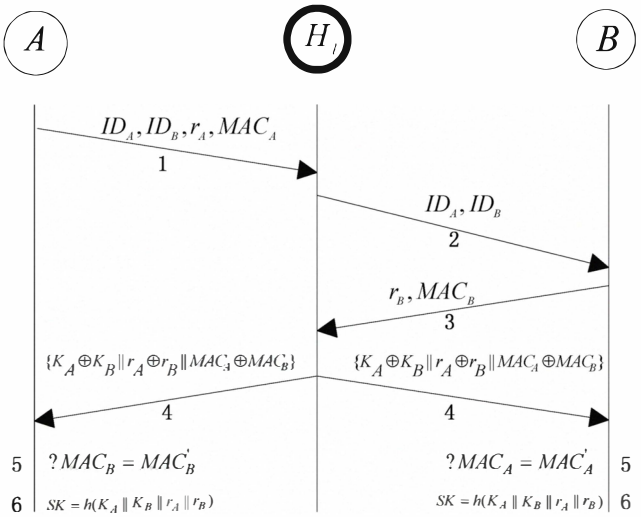

Figure 3

1) Node A sends a challenge random r_A, a message authentication code $MAC_A = h(r_A \| K_A)$ and ID_A, ID_B to its clusterhead $H_l, l \in \{1,\cdots,N\}$, where N is the current maximum number of clusterheads in Ad hoc network.
2) When clusterhead H_l receive the message from node A, it first checks if node A and node B are associated with it. If the two nodes belong to the same cluster, then H_l records r_A, MAC_A, ID_A, ID_B and delivers ID_A, ID_B to node B.

3) When node B receives ID_A, ID_B, node B knows A wants to communicate with it. Then it sends a random challenge r_B and $MAC_B = h(r_B \| K_B)$ to H_l.

4) H_l first performs a simple table look-up and computes $(K_A \oplus R) \oplus (K_B \oplus R) = K_A \oplus K_B$, and it then uses network coding paradigm to broadcast the value of $\{K_A \oplus K_B \| r_A \oplus r_B \| MAC_A \oplus MAC_B\}$.

5) Upon receiving the message, node A computes $K_A \oplus \{K_A \oplus K_B\} = K_B^{'}$, $r_A \oplus \{r_A \oplus r_B\} = r_B^{'}$ and $MAC_A \oplus \{MAC_A \oplus MAC_B\} = MAC_B$, and then computes $MAC_B^{'} = h(r_B^{'} \| K_B^{'})$; Node B computes $K_B \oplus \{K_A \oplus K_B\} = K_A^{'}$, $r_B \oplus \{r_A \oplus r_B\} = r_A^{'}$ and $MAC_B \oplus \{MAC_A \oplus MAC_B\} = MAC_A$, and then computes $MAC_A^{'} = h(r_A^{'} \| K_A^{'})$.

6) Node A verifies to confirm if $MAC_B^{'} = MAC_B$; Node B verifies to confirm if $MAC_A^{'} = MAC_A$. If they are equal, then both node A and node B will compute a shared secret key $SK = h(K_A \| K_B \| r_A \| r_B)$.

(ii) Case 2: Two Ad hoc nodes belong to two different clusterheads. Figure 4 shows the protocol.

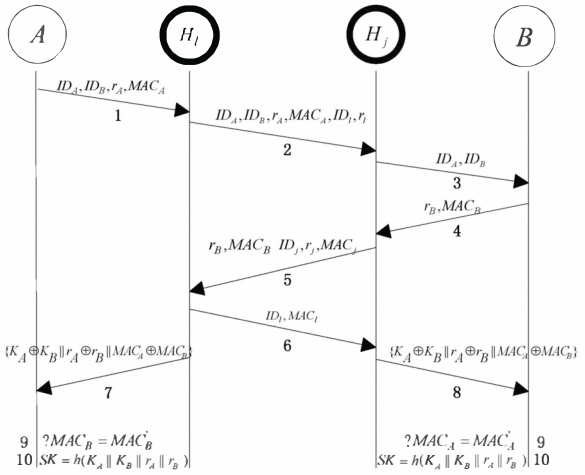

Figure 4 Protocol for two ad hoc nodes associating with two clusterheads

1) Node A initiates the protocol by sending a challenge random r_A, a message authentication code $MAC_A = h(r_A \| K_A)$ and ID_A, ID_B to its clusterhead $H_l, l \in \{1, \cdots, N\}$. This step is same as that of the first protocol.

2) Upon receiving the message from node A, H_l first checks if node A and node B are associated with it. If not, H_l records ID_A, ID_B, r_A, MAC_A and broadcasts a 6-tuple $ID_A, ID_B, r_A, MAC_A, r_l, ID_l$ to the other clusterheads, where r_l is the new random challenge generated by H_l.

3) Suppose clusterhead H_j receives the 6-tuple $ID_A, ID_B, r_A, MAC_A, r_l, ID_l$ broadcasted from H_l, H_j knows that node A want to communicate with is subordinate node B. Then H_j records the 6-tuple $ID_A, ID_B, r_A, MAC_A, r_l, ID_l$ and broadcasts ID_A, ID_B to node B.

4) Upon receiving ID_A, ID_B, node B knows A wants to communicate with it, Then it sends a random challenge r_B and $MAC_B = h(r_B \| K_B)$ to H_j.

5) Upon receiving r_B, MAC_B, H_j first performs a simple table look-up and computes $MAC_j = h(r_l \| K_j \oplus (K_l \oplus R)) = h(r_l \| K_j \oplus K_l \oplus R)$. Then it generates a new random challenge r_j and sends a 5-tuple $r_B, MAC_B, ID_j, r_j, MAC_j$ to H_l.

6) Upon receiving the 5-tuple, H_l performs a simple table look-up and computes $MAC_j^{'} = h(r_l \| K_l \oplus (K_j \oplus R)) = h(r_l \| K_l \oplus K_j \oplus R)$. Then H_l check if $MAC_j = MAC_j^{'}$. If the two values are equal, then H_l authenticates H_j. H_l performs a simple table look-up and computes $MAC_l = h(r_j \| K_l \oplus (K_j \oplus R)) = h(r_l \| K_l \oplus K_j \oplus R)$, then unicasts ID_l, MAC_l to H_j.

7) At almost the same time, H_l computes $(K_A \oplus R) \oplus (K_B \oplus R) = K_A \oplus K_B$, then it computes the value $\{K_A \oplus K_B \| r_A \oplus r_B \| MAC_A \oplus MAC_B\}$ and uses network coding paradigm to broadcasts the value to node A.

8) Upon receiving ID_l, MAC_l, H_j performs a simple table look-up and computes $MAC_l^{'} = h(r_j \| K_j \oplus (K_l \oplus R)) = h(r_l \| K_j \oplus K_l \oplus R)$. Then H_j check if $MAC_l = MAC_l^{'}$. If the two values are equal, then H_j authenticates H_l. H_l computes $(K_A \oplus R) \oplus (K_B \oplus R) = K_A \oplus K_B$, then it computes the value $\{K_A \oplus K_B \| r_A \oplus r_B \| MAC_A \oplus MAC_B\}$ and broadcasts the value to node A.

9) Upon receiving the message, node A computes $K_A \oplus \{K_A \oplus K_B\} = K_B^{'}$, $r_A \oplus \{r_A \oplus r_B\} = r_B^{'}$ and $MAC_A \oplus \{MAC_A \oplus MAC_B\} = MAC_B$, and then computes $MAC_B^{'} = h(r_B^{'} \| K_B^{'})$; Node B computes

$K_B \oplus \{K_A \oplus K_B\} = K_A^{'}$, $r_B \oplus \{r_A \oplus r_B\} = r_A^{'}$ and $MAC_B \oplus \{MAC_A \oplus MAC_B\} = MAC_A$, and then computes $MAC_A^{'} = h(r_A^{'} \| K_A^{'})$.

10) Node A verifies to confirm if $MAC_B^{'} = MAC_B$; Node B verifies to confirm if $MAC_A^{'} = MAC_A$. If they are equal, then both node A and node B will compute a shared secret key $SK = h(K_A \| K_B \| r_A \| r_B)$.

C. Updating of the Key

As we know, the new shared key established between arbitrary two Ad hoc nodes is $SK = h(K_A \| K_B \| r_A \| r_B)$, in which two random numbers are included. This guarantees the shared key is fresh for every protocol execution. When one node want to update his shared key with the other nodes, he can simply initiate a new protocol execution once again, then he will get the new shared keys with the designated nodes, as the two random numbers r_A and r_B have changed.

V. PERFORMANCE ANALISIS

A. Security Analysis

There are numerous kinds of attacks in the mobile ad hoc network, almost all of which can be classified as the following two types:

- *External attacks*, in which the attacker aims to cause congestion, propagate fake routing information or disturb nodes from providing services.
- *Internal attacks*, in which the adversary wants to gain the normal access to the network and participate the network activities, either by some malicious impersonation to get the access to the network as a new node, or by directly compromising a current node and using it as a basis to conduct its malicious behaviors.

In the above two categories, external attacks are similar to the normal attacks in the traditional wired networks in that the adversary is in the proximity but not a trusted node in the network, therefore, this type of attack can be prevented and detected by the security methods such as membership authentication or firewall, which are relatively conventional security solutions. However, due to the pervasive communication nature and open network media in the mobile ad hoc network, internal attacks are far more dangerous than the internal attacks: because the compromised nodes are originally the benign users of the ad hoc network, they can easily pass the authentication and get protection from the security mechanisms. As a result, the adversaries can make use of them to gain normal access to the services that should only be available to the authorized users in the network, and they can use the legal identity provided by the compromised nodes to conceal their malicious behaviors. Therefore, we should pay more attention to the internal attacks initiated by the malicious insiders when we consider the security issues in the mobile ad hoc networks. In the following, we discuss the main attack types that emerge in the mobile ad hoc networks[10].

An internal attack from a compromised malicious nodes is a severe threat against a mobile ad hoc network. An internal attack is either an authorised user misusing his/her privileges or an external attacker gaining access to the equipment and performing the attack with internal user privileges.

The main security threats in Ad hoc networks can be listed as follows:

- *Eavesdropping attack*, which means the attacker can listen to all the traffic over the wireless medium. If the attacker can not get any keys from the key pool, it can only eavesdrop the user's identifiers, message authentication codes and random challenges, while these data except the identifiers will be changed during the next protocol execution. The attacker can not obtain any useful secret information by eavesdropping the traffic.
- *Impersonation attack*, which means the attacker can intercept traffic on wireless link and impersonate a legitimated user by replaying some intercepted private information. If the attacker could just intercept the data traffic and simply replay them during the current session, it can not obtain any benefit from the transaction. If the attacker wants to replay the intercepted data from the last protocol execution, then it will be easily detected by verifying the *MAC* values on both Ad hoc nodes and clusterheads, because the attacker can not get the correct keys.
- *Node-compromising attack*, which means some Ad hoc nodes or clusterheads could be captured after deployment. Afterwards, the attacker can gain access to the memory of the Ad hoc node or a clusterhead. We always reasonably assume that the Ad hoc nodes are equipped with tamper-detection devices and once a node is captured, the keys in its memory will be erased automatically. If this assumption can not be met, the protocols proposed in this paper are not secure anymore. Therefore, we strongly suggest that the tamper-detection devices should be adopted by Ad hoc terminals when the protocol is used.
- *Brute-force attack*, which means an adversary could launch a attack against the XOR and hash operations used in the protocols. As we know, the keys stored in the Ad hoc nodes are XOR-ed by a random number R, which can be considered as a encryption operation using one-time padding cipher. It has been proved that the one-time padding cipher can achieve information-theoretic security. Besides, one of our initial assumptions is that the hash function used in the protocol is secure. An adversary can not find collisions of the *MACs*. Thus, the protocols proposed in paper are secure against brute-force attack.

B. Analysis of Computation Overhead

1) Memory reqiurements: As we know that each Ad hoc node or clusterhead has to store some permanent N identifiers, one node key K_i and $(N-1)$ XOR-ed keys, and some temporary data including *MAC* data, identifiers and challenges. To store the protocol data, each node requires approximately $|S| \approx N \times (|K_i| + |ID_i|)$ bits, where $|K_i|$ is the key length and $|ID_i|$ is the length of *ID* bit string. For example, if we assign $|ID_i|$ =8 bits, the maximum number N of Ad hoc nodes would be 256. Table II shows the required memory, which is very reasonable under current technology.

TABLE II. MEMORY REQUIREMENTS FOR THE PROTOCOLS

	Memory Size		
$\|K_i\|$	***N=256***	***N=512***	***N=1024***
32 bits	1280 Bytes	2624 Bytes	5376 Bytes
64 bits	2304 Bytes	4672 Bytes	9472 Bytes
128 bits	4352 Bytes	8768 Bytes	17664 Bytes
256 bis	8848 Bytes	16960 Bytes	34048 Bytes

2) Analysis of Computation Overhead: As shown in Figure 1, when the first protocol is executed, both node A and node B need to compute 2 *MAC* values, 1 hash value, and to do 3 XOR operations. Clusterhead needs to do 4 XOR operations only. When the second protocol as shown in Figure 2 is executed, the computation overhead for both node A and node B is the same as the first protocol, but the clusterheads need to do 4 XOR operations and compute 1 *MAC* value. Actually, the computation overhead of XOR operations in a computer system can be negligible, therefore the main computation overhead of the protocols comes from the hash operations. Because neither symmetric-key cryptosystem nor public-key cryptosystem is used, the protocols proposed are absolutely light-weight and computationally efficient.

VI. CONCLUDING REMARKS

Wireless Ad hoc network are vulnerable to various attacks, such as eavesdropping, impersonation, and node-compromising and brute-force attacks. In this paper, we propose two light-weight key distribution protocols based on network coding paradigm. The security and memory requirements of the protocols are also analyzed. The new protocols provide a lightweight solution for distributing keys and ensuring communication confidentiality and authentication of nodes against eavesdropping and impersonation attacks. The memory requirement for the protocols is reasonable under modern integrated circuit technology. The computation overhead for each node in our new protocols is lower and thus they are very lightweight. We also pointed out that our protocols can not resist node-compromising attack. Therefore, we suggest a tamper-free device should be used at the Ad hoc terminals when the protocols are implemented.

As a future research, we plan to extend our approach to combine network coding and cryptographic primitives to prevent node-compromising attack in an effective way.

ACKNOWLEDGMENT

This work is partially supported by the National Science Foundation (NFS) of China under grant NFS-60672102 and Foundation of "863" high technology program under grant 2009AA01Z418.

The authors would like to thank Chi Zhang, Miao Pan, and Yongsheng Huang, Jinyuan Sun, Yang Song, Hao Yue from University of Florida for valuable discussions.

REFERENCES

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] L. Lima, J. P. Vilela, P. F. Oliveira and J. Barros. "Network Coding Security: Attacks and Countermeasures", *Cryptography and Security*, Sep 2008.

[3] J. Dong, R. Curtmola, R. Sethi, C. Nita-Rotaru. "Toward Secure Network Coding in Wireless Networks: Threats and Challenges", In *Proceedings of 4th Workshop on Secure Network Protocols (NPSEC) in conjunction with IEEE ICNP*, Orlando, Florida, Oct 2008.

[4] Paulo F. Oliveira, JoÃ£o Barros, "Network Coding Protocols for Secret Key Distribution", *Proc. of the International Symposium on Information* Security (IS'07), 2007.

[5] C. Gkantsidis, P. R. Rodriguez, "Network Coding for Large Scale Content Distribution", in *Proceedings of IEEE Inforcom 2005*.

[6] J. P. Vilela, L. Lima and J. Barros, "Lightweight Security for Network Coding", *Proceedings of International Conference On Communications* (ICC2008), Beijing, 2008.

[7] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," IEEE Transactions on Information Theory, vol. 49, no. 2, pp. 371–381, 2003.

[8] Z. Yu, Y. Wei, B. Ramkumar and Y. Guan, "An Efficient Scheme for Securing XOR Network Coding against Pollution Attacks", in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, April, 2009.

[9] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Medard, "Resilient Network Coding in the Presence of Byzantine Adversaries", *Proceeedings of INFOCOM2007*.

[10] B. Wu, J. Chen, J. Wu, and M. Cardei "A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks", Chapter 12, Wireless/Mobile Network Security. Y. Xiao, X. Shen, and D.-Z. Du (Eds.), 2006 Springer.

[11] P. F. Oliveira, R. A. Costa, and J. Barros, "Mobile Secret Key Distribution with Network Coding", *Proc. of the International Conference on Security and Cryptography* (SECRYPT'07), 2007.

[12] L. Bao and J.J. Garcia-Luna-Aceves, "Topology Management in Ad Hoc Networks", The ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), Annapolis, Maryland, June 1-3, 2003.

[13] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks Wireless Sensor Networks," *ACM Transactions on Information and System Security*, vol.8, no.2, pp.228-258, 2005.