Computer Networks 55 (2011) 2534–2544

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A scalable and robust key pre-distribution scheme with network coding for sensor data storage

Rongfei Zeng [a], Yixin Jiang [a], Chuang Lin [a], Yanfei Fan [b], Xuemin (Sherman) Shen [b,*]

[a] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[b] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

ARTICLE INFO

Article history:
Received 12 January 2011
Accepted 23 March 2011
Available online 4 May 2011
Responsible Editor: I.F. Akyildiz

Keywords:
Distributed sensor data storage
Key pre-distribution
Network coding
Matrix decomposition

ABSTRACT

In this paper, we propose a scalable and robust key pre-distribution scheme based on network coding and matrix decomposition for distributed sensor data storage. As a promising information dissemination technique, network coding is suitable for the key information exchange due to the reduced number of transmissions and the enhanced robustness. Matrix decomposition can help to exchange key information without any central nodes and location knowledge. Our scheme uses LU matrix decomposition to decompose a shared key into two vectors R and C. The vector R is held privately by the sensor, while the other vector C is protected and disseminated using network coding. The combination of network coding and matrix decomposition enhances the scalability, improves the communication performance, and increases the robustness for sensor data storage networks. Extensive theoretical analysis and simulative results both demonstrate the efficacy and efficiency of the proposed key pre-distribution scheme in distributed sensor data storage.

1. Introduction

Recently, security and privacy issues have gained increasing attention in distributed sensor data storage [1] which is extensively applied to the reliable and privacy-preserving access to confidential data in wireless sensor networks (WSNs). As an indispensable security and privacy component, key management is in charge of providing shared encryption and authentication keys for securing connections between vulnerable nodes in sensor data storage.

A variety of key management schemes have been proposed in the past few years [2–22]. The traditional trusted-server schemes rely on trusted online servers to establish shared keys between nodes. However, such central online infrastructure may not exist in sensor data storage networks. The self-enforcing schemes depend on asymmetric cryptography (e.g., Diffie–Hellman and RSA [2]) to negotiate pairwise keys, making the self-enforcing schemes infeasible in sensor data storage due to sensors' limited computational and energy resources [3–5]. The key pre-distribution schemes have been demonstrated to be more practical in WSNs [6,7]. Having some key information installed in nodes before deployment, the key pre-distribution schemes can be further classified into three types: the shared-key schemes, the location-aware schemes, and the pairwise schemes with structure key pool. Eschenauer et al. [7] first propose a probabilistic shared-key pre-distribution scheme, the security of which is further enhanced in [8–10]. The group-based and grip-based schemes leverage location knowledge to improve the security and performance [11–13]. The pairwise schemes with structure key pool adopt symmetric matrices, bivariate polynomials, and matrix decomposition in the key pre-distribution [2,14–19]. Recently, Oliveira

* Corresponding author. Tel.: +1 519 888 4567x32691; fax: +1 519 746 3077.
E-mail addresses: zengrf@csnet1.cs.tsinghua.edu.cn (R. Zeng), yxjiang@csnet1.cs.tsinghua.edu.cn (Y. Jiang), clin@csnet1.cs.tsinghua.edu.cn (C. Lin), yfan@bbcr.uwaterloo.ca (Y. Fan), xshen@bbcr.waterloo.ca (Xuemin (Sherman) Shen).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.03.018

et al. propose an efficient key pre-distribution scheme based on network coding (NC) [20].

However, the previous approaches cannot be directly applied to distributed sensor data storage, since sensor data storage networks have the following unique characteristics: (1) Vulnerable sensors may suffer from Byzantine failures and various attacks (e.g., collusion attacks and node-compromised attacks), thus the key pre-distribution scheme should be robust; (2) The communication bandwidth is more limited than other resources (e.g., memory) in sensor data storage, requiring the key pre-distribution scheme to be communication-efficient; and (3) The central online infrastructure and location information may not be available in sensor data storage [1,23]. In summary, the key pre-distribution scheme should be scalable, robust, efficient, and location-knowledge-free.

In this paper, we propose a robust and scalable key pre-distribution scheme based on network coding and matrix decomposition for distributed sensor data storage. As a promising information dissemination approach, NC is suitable for the key information exchange due to the reduced number of transmissions and the enhanced robustness. Matrix decomposition can be utilized to exchange key information without any central nodes and location knowledge. Our scheme uses LU matrix decomposition to decompose a shared key into two vectors R and C. The vector R is held privately by the sensor, while the other vector C is protected and disseminated using NC. In summary, the combination of network coding and matrix decomposition enhances the scalability, improves the communication performance, and increases the robustness for sensor data storage networks.

Our main contributions are threefold: (1) *Robustness*: We uses NC and matrix decomposition to enhance the robustness of key pre-distribution scheme for sensor data storage in hostile scenarios. The proposed scheme is proved to be secure against node-compromised attacks and resilient against collusion attacks; (2) *Efficiency*: It is demonstrated by theoretical analysis that the performance, such as the expected transmission number and local connectivity, are significantly improved. Compared with the traditional matrix decomposition scheme [14], our scheme remarkably reduces the communication overhead by 25% and guarantees the local connectivity as well; and (3) *Scalability*: The proposed scheme achieves security-enhanced and efficient key dissemination without the need for online central nodes and location information, making the scheme more scalable in distributed sensor data storage. Finally, to the best of our knowledge, this is the first research work to comprehensively consider the robustness, efficiency, and scalability in the key pre-distribution scheme for sensor data storage networks.

The remainder of the paper is organized as follows. Section 2 surveys related work, and Section 3 introduces the system model, threat model, and the preliminaries. In Section 4, we present the proposed scheme based on NC and matrix decomposition. We also present the features and extensions in Section 5. Security analysis and performance evaluations are given in Section 6 and Section 7, respectively. Section 8 concludes the paper.

2. Related work

Key management plays an important role in establishing secure communications in WSNs. In general, there are three types of key management schemes: the trusted-server schemes, the self-enforcing schemes, and the key pre-distribution schemes. Due to the lack of central online infrastructure, the trusted-server schemes cannot be directly applied to WSNs. The self-enforcing schemes depend on asymmetric cryptography, such as Diffie-Hellman and RSA [2], to negotiate pairwise keys. As a novel self-enforcing scheme, the ID-based cryptography enables two nodes to establish their shared keys by performing energy-consuming pairing operations [3–5]. With the increasing number of sensor nodes, the scalability of this scheme is a main concern due to the computation-limited sensors. Some recent work demonstrates that key pre-distribution schemes offer practical solutions to the key management problem in sensor networks [7]. The key pre-distribution schemes can be further categorized into the following types: (1) the shared-key schemes, (2) the location-aware schemes, and (3) the pairwise schemes with structured key pool.

Eschenauer et al. [7] first propose a probabilistic key pre-distribution scheme, which uniformly pre-distributes a large global set of keys so that each node has a key subset in the memory and two neighbors can negotiate a probabilistic key by intersecting their key subsets. Chan et al. extend this classic scheme and present a q-composite scheme to reduce the impact of compromised nodes [8]. The techniques, including key index notification, challenge-response, and pseudo-random key index transformation, have also been used to improve the security and performance in the key discovery phase [9,10].

As the typical location-aware schemes, group-based schemes and grip-based schemes leverage prior deployment knowledge to lower the impact of node-compromised attacks, increase the local connectivity, and reduce storage and computational overheads [11–13]. However, due to the randomness of deployment, obtaining such location information might not be feasible for sensor data storage.

The pairwise key pre-distribution schemes with structured key pool are proposed to improve the resilience against various attacks. Blom et al. propose a novel λ-secure key pre-distribution scheme based on a symmetric matrix of size $(\lambda+1)\times(\lambda+1)$ over $GF(q)$ [15]. Du et al. enhance the security of Blom's scheme by introducing multiple key spaces in [2]. In [30], nodes are organized into a n-dimensional hypercube, and Blundo's polynomial [18] is used in each dimension. The techniques of trivariate and multivariate symmetric polynomials are also applied to the key pre-distribution in WSNs [11,19]. Dai et al. use matrix decomposition and polynomial-based approach to guarantee that any two nodes can negotiate a pairwise key in [14]. Qamtepe et al. present two classes of combinatorial designs, i.e., symmetric balanced incomplete block designs and generalized quadrangles, in the key pre-distribution of WSNs [9]. Recently, network coding has also been applied to the pairwise key pre-distribution scheme

in WSNs environments [20]. In this novel scheme, a central trusted node distributes the XORed keys, and each sensor can establish pairwise keys by XORing the encoded keys with its own key. However, the trusted online nodes may suffer from a single point of failure. In addition, the requirement of central online nodes makes this approach inappropriate for distributed sensor data storage.

In various key pre-distribution schemes, most related to our proposal are the works of [14] and [20]. Compared to the scheme in [14], our scheme uses NC to reduce the expected transmission number by 25% in the key dissemination. Unlike the scheme in [20], our scheme utilizes matrix decomposition to enable a shared key to be calculated from two vectors, one of which is protected and disseminated using NC. The robustness of our scheme is thus enhanced by matrix decomposition and NC. Moreover, our key pre-distribution scheme does not need any central online nodes and location information, making our proposal more scalable than [20]. In summary, the proposed scheme is scalable, robust, efficient, and location-knowledge-free.

3. System model and preliminaries

3.1. System model

We consider the typical wireless data storage networks consisting of a large number of sensors, which are deployed in area of interest. From Fig. 1, some nodes sense the environments, occasionally generate data, and then distribute them to the neighbors called storage nodes within one-hop distance through unreliable channels depicted in real lines. Moreover, communications between remote sensors are through multi-hop transmissions shown in dashed lines. Without loss of generality, we assume that sensors are equipped with sufficient memory to store the sensed data and some related key information. However, sensors have limited power supply and constrained computational capability, and they do not have GPS modules and tamper-proof hardware. Finally, nodes are pre-installed with some key information by an offline server before deployment, which is an indispensable procedure of key pre-distribution schemes. However, it is noteworthy that there are no trusted online nodes in sensor data storage networks to help sensors negotiate pairwise keys after deployment.

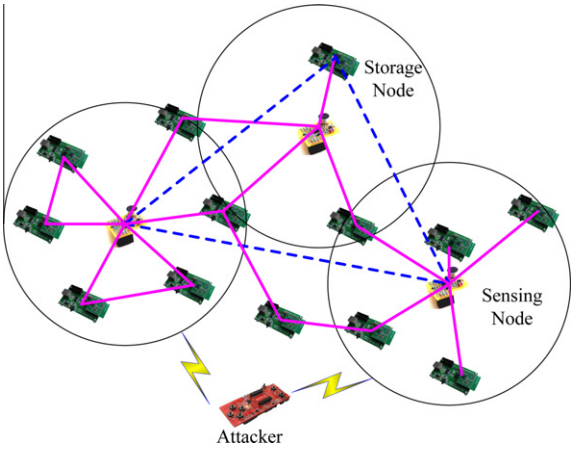


Fig. 1. The system model of sensor data storage.

3.2. Threat model

Generally, the possible threats come from Byzantine failures and malicious attacks (e.g., node-compromised attacks and collusion attacks) in sensor data storage networks. For Byzantine failures, some nodes may malfunction randomly during the whole lifetime, which should not interfere with others' key negotiations. For various attacks, the main goal of malicious attackers is to destroy the key pre-distribution scheme. They attempt to compromise as many storage nodes as possible to obtain the confidential key information. If a node is under the control of attackers, adversaries can read its memory and monitor all the incoming and outgoing communications. Attackers can also collaborate to launch collusion attacks, making the key pre-distribution scheme easy to be destroyed.

3.3. Design goal

The overall design goal is to enable any pair of nodes to efficiently and robustly negotiate a pairwise key in sensor data storage networks. Specifically, we want to achieve the followings: (1) *Robustness*: Since sensor data storage networks are hostile, the key pre-distribution scheme should be more resilient against node-compromised attacks and collusions than previous schemes. In other words, capturing several keys by malicious nodes may not lead to the derivation of other keys or the disruption of entire networks; (2) *Efficiency*: The scheme should be communication-efficient, meaning that the expected transmission number should not be high for sensor data storage. Moreover, the local connectivity must be guaranteed as well to ensure that any two neighbors can always establish a shared key; and (3) *Scalability*: The scheme should be deployed in a distributed manner without any location information and central online infrastructure.

3.4. Preliminaries

Network coding: Network coding first proposed by Ahlswede et al. [28] provides an efficient communication paradigm which allows intermediate nodes to mix the input messages and make the output messages be the mixture of the input ones, as shown in Fig. 2. This novel technique can improve the communication efficiency [24,27] and enhance the network security [25,26]. Among various encoding algorithms, random linear network coding (RLNC) and XOR network coding are widely applied in the recent years. In RLNC, the message $y(e) \in GF(q)$

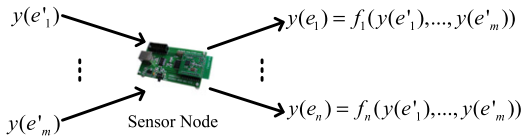


Fig. 2. Network coding.

carried on the outgoing edge e of node v can be computed as a linear combination of the message $y(e')$ on the incoming edge e', i.e., $y(e) = \sum_{e'} \beta_{e'} y(e')$. Similarly, the intermediate node v performs the XOR operations on the input messages to generate the message $y(e) = a_1 y(e'_1) \oplus \cdots \oplus a_m y(e'_m)$, $a_m \in \{0, 1\}$ for the outgoing edge e in XOR network coding. When a node collects enough encoded messages, it can easily recover the original symbols.

Matrix decomposition: LU matrix decomposition is the process of decomposing a symmetric matrix M of size $\lambda \times \lambda$ into two matrices, i.e.,

$$M = LU = \begin{pmatrix} l_{11} & & & 0 \\ l_{21} & l_{22} & & \\ \vdots & & \ddots & \\ l_{\lambda 1} & & \cdots & l_{\lambda\lambda} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1\lambda} \\ & u_{22} & \cdots & u_{2\lambda} \\ & & \ddots & \vdots \\ 0 & & & u_{\lambda\lambda} \end{pmatrix}, \tag{1}$$

where L is a lower triangular matrix, and U is an upper triangular matrix. The decomposition algorithms include the Doolittle algorithm, the Crout algorithm, etc. Refer to [29] for details.

4. The proposed scheme

The proposed key pre-distribution scheme consists of two components: (1) link layer key (LLK) establishment: the key establishment between neighbors within one-hop distance; and (2) transport layer key (TLK) establishment: the key establishment between nodes with multi-hop distance.

4.1. Link layer key establishment

Based on RLNC and LU matrix decomposition, the LLK establishment scheme is composed of two phases: the key pre-distribution phase and the key agreement phase. Each node is identified by a unique global ID ranging from 1 to N, where N is the number of nodes in the networks.

- **Key pre-distribution phase**

Step 1 **(Generating matrices)**: We first randomly generates n symmetrical matrices M_i, $i = 1, \ldots, n$ of size $\lambda \times \lambda$, which satisfy $M_i^T = M_i$. Note that the element of M_i is randomly picked from a finite field $GF(q)$, where q is a prime number large enough for a cryptographic key.

Step 2 **(Decomposing matrices)**: The LU matrix decomposition technique in [29] is applied to each symmetric M_i to have $M_i = L_i \cdot U_i$, where L_i is a lower triangular matrix and U_i is the corresponding upper triangular matrix.

Step 3 **(Pre-distributing keys)**: Let $R_i(j)$ represent the jth row of L_i and $C_i(j)$ represent the jth column of U_i. For each node, we perform the following operations: (1) τ matrices are randomly chosen from $\{M_1, \ldots, M_n\}$ to constitute the set S_1; (2) For each matrix $M_i \in S_1$, we choose a random row $R_i(k)$ from L_i and its corresponding column $C_i(k)$ from U_i and store τ four-tuples $\langle i, k, R_i(k), C_i(k) \rangle$ at the sensor node for its future key negotiation; (3) We also randomly choose another σ matrices from the remaining matrices $\{M_1, \ldots, M_n\} \backslash S_1$ to constitute the set S_2; and (4) For the matrix U_j of $M_j \in S_2$, we compute

$$\widehat{C}_j(k) = a_{jk} \cdot C_j(k) + b_{jk} \cdot C(R), \quad k = 1, \ldots, \lambda \tag{2}$$

where a_{jk} and b_{jk} are two elements randomly chosen from $GF(q)$, and $C(R)$ is a random vector of size λ. We store the encoded columns with the coefficients, i.e., $< j, a_{jk}, b_{jk}, \widehat{C}_j(k) >, k = 1, \ldots, \lambda$, at the node for other nodes' key negotiations.

Note that all the steps in the key pre-distribution phase are performed offline before sensor deployment, thus the computational overhead is not a concern in this phase. Moreover, since $C(R)$ is unknown to the storage nodes, sensors can neither obtain matrix U_j nor the corresponding matrix M_j, which efficiently enhances the robustness. Finally, we can also pre-perform the above calculations for additional nodes before deployment and add them to the networks when required.

- **Key agreement phase**

When node A and its neighbor B want to negotiate a shared key before delivering packets, they should perform the key agreement as follows:

Step 1: Node A broadcasts a key negotiation request $\{N_A, IndexU_1, IndexC_1, \ldots, IndexU_\tau, IndexC_\tau, N_B\}_A$, where N_A and N_B are the IDs of communication nodes, $IndexU_i$ is the index of U_i decomposed from M_i, $IndexC_i$ is the index of U_i's column, and $\{\}_A$ denotes the message sent by node A. Meanwhile, node B broadcasts a similar request as well.

Step 2: Suppose that an intermediate node C, which is in the neighborhood of A and B and contains columns of the common matrix U_i, i.e., $IndexU_i \in \{IndexU_1, \ldots, IndexU_\tau\}_A \cap \{IndexU_1, \ldots, IndexU_\tau\}_B$, overhears these two requests. Let m denote node A's column index of the common matrix U_i and n denote node B's column index of U_i. The encoded columns of m and n stored at node C are respectively

$$\widehat{C}_i(m) = a_{im} \cdot C_i(m) + b_{im} \cdot C(R), \tag{3}$$

$$\widehat{C}_i(n) = a_{in} \cdot C_i(n) + b_{in} \cdot C(R). \tag{4}$$

From Eq. (3) and (4), we can easily eliminate $C(R)$ and get

$$\widehat{C}_i = a \cdot C_i(m) + b \cdot C_i(n), \tag{5}$$

where $\widehat{C}_i = (b_{in}\widehat{C}_i(m) - b_{im}\widehat{C}_i(n))/(b_{in}b_{im})$, $a = a_{im}/b_{im}$, and $b = -a_{in}/b_{in}$. Then, node C broadcasts the three-tuple $\{a, b, \widehat{C}_i\}_C$ to nodes A and B.

Step 3: After receiving the message $\{a, b, \widehat{C}_i\}_C$, node A can derive $C_i(n)$ by solving Eq. (5) with its own $C_i(m)$. Subsequently, node A computes the secret pairwise key K_{AB} as

$$K_{AB} = R_i(m) \times C_i(n). \tag{6}$$

Similarly, node B can get K_{BA} as well. Since all the matrices are symmetric, we have $K_{AB} = K_{BA}$. So far, a secret pairwise key is generated for nodes A and B.

4.2. Transport layer key establishment

Two remote nodes should negotiate a TLK before they prepare to communicate. For the TLK establishment, our primary idea is to combine RLNC with the traditional multi-path transmissions over edge-disjoint channels. Suppose that two nodes S and D need to establish a TLK, and there exist l edge-disjoint paths $P_1,\ldots,P_l$ constructed by the technique in [2]. Node S first chooses h ($1 < h < l$) random number $K_1, K_2,\ldots,K_h$ and then computes

$$\begin{pmatrix} \widetilde{K}_1 \\ \vdots \\ \widetilde{K}_l \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1h} \\ \vdots & \ddots & \vdots \\ a_{l1} & \cdots & a_{lh} \end{pmatrix} \begin{pmatrix} K_1 \\ \vdots \\ K_l \end{pmatrix} \tag{7}$$

where a_{ij} ($1 \leqslant i \leqslant l$, $1 \leqslant j \leqslant h$) is the coefficient randomly picked from $GF(q)$. Then, node S sends messages $\{a_{i1},\ldots,a_{ih},\widetilde{K}_i\}$ ($i = 1,\ldots,l$) to node D through these l edge-disjoint channels. After obtaining enough messages, node D can recover the original keys $K_1,\ldots,K_h$ by solving Eq. (7) using Gaussian eliminations, which is demonstrated to be feasible in sensor data storage [1]. Then, node D computes the TLK as $K_{SD} = h(K_1,\ldots,K_h)$, where $h(\cdot)$ is a public hash function known to all the sensors. It is noteworthy that the robustness of TLK establishment is guaranteed by RLNC, since any single sensor along the path cannot get the secret TLK.

When l edge-disjoint paths do not exist between a pair of remote nodes in some special scenarios, we propose another approach extended from our LLK scheme. Nodes S and D separately broadcast their key establishment requests. For intermediate nodes: (1) if they do not have any matrix required by the messages, they just forward the requests; (2) if they have the matrices indexed by any requests, they store the requests for an interval which is determined by the communication bandwidth and the size of WSNs; (3) if they receive two related requests and have the common matrix indexed by the requests, they compute $\widehat{C}_i$ as Eq. (5) and then deliver the message $\{a, b, \widehat{C}_i\}$ to nodes S and D. After obtaining such response, nodes S and D generate a secret pairwise key as Eq. (6).

We note that although TLK establishment schemes have been proposed in a few previous work, we effectively enhance the robustness of TLK establishment by combining RLNC with the traditional TLK scheme or extending the proposed LLK scheme.

5. The features and extensions

5.1. Node authentication

An extension of our approach is the node authentication protocol used to validate the authenticity of sensor nodes. The procedure of node authentication is detailed as follows:

Step 1: The verifier V broadcasts an authentication request $\{IndexU_1, IndexC_1,\ldots,IndexU_\tau, IndexC_\tau, N_V\}$, where $IndexU_i$ and $IndexC_i$ are respectively the indexes of U_i and U_i's column associated with S_1.

Step 2: The neighbors, who have the matrices required by the authentication request, respond to the verifier V. They individually compute a linear combination of the columns by eliminating $C(R)$, i.e.,

$$\widehat{C} = a_1 C_{x1}(d_{x1}) + \cdots + a_j C_{xj}(d_{xj}), \tag{8}$$

encrypt the corresponding matrix indexes with the first element of $\widehat{C}$, i.e., $E = En_{\widehat{C}^1}(IndexU_{x1}\|\cdots\|IndexU_{xj})$, and send $\{N_x, IndexU_{x1},\ldots,IndexU_{xj}, a_1,\ldots,a_j, E\}_x$ back to the verifier V.

Step 3: After receiving the acknowledgments, the verifier V checks the authenticity by decrypting E and comparing it with the matrix indexes.

In Fig. 3, we present an instance for a node to check the legitimacy of its neighbors. The verifier V first sends an authentication request to inform its neighbors that it contains columns $\{C_1(1), C_3(4), C_4(1)\}$. After obtaining the authentication request, nodes A, B and C respectively respond to node V. For node A containing encoded matrices $\{\widehat{U}_3, \widehat{U}_4\}$, it computes $\widehat{C} = C_3(4) + C_4(1)$, encrypts the corresponding matrix indexes with $\widehat{C}$'s first element $\widehat{C}^1$, i.e., $E = En_{\widehat{C}^1}(3\|4)$, and sends $\{A,3,4,1,1,E\}_A$ back to node V. Node V checks whether the equation of E is satisfied. If the equation holds, then node A is proved to be authentic. For nodes B and C, the response and verification procedures are similar. It should be noted that an attacker can successfully pass the authentication procedure if it compromises a sensor and gets all the data from the compromised node.

5.2. A simplified LLK establishment

Though our scheme can be easily implemented with linear operations in sensors, the proposed LLK establishment scheme can still be simplified for the scenarios with computationally-weak nodes. In Step 3 of the key pre-distribution phase, we can protect U_j using Vernam cipher and compute $\widehat{C}_j(k)$ as

$$\widehat{C}_j(k) = C_j(k) \oplus C(R), \quad k = 1,\ldots,\lambda. \tag{9}$$

When an intermediate node overhears two related requests, it performs only the XOR operations on two encoded columns and get

$$\widehat{C}_i = C_i(m) \oplus C_i(n). \tag{10}$$

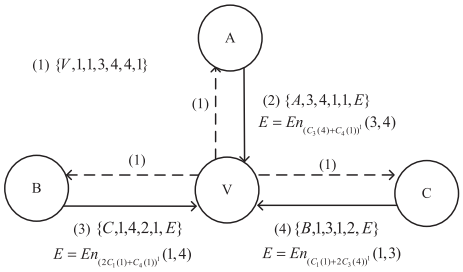


Fig. 3. Node authentication.

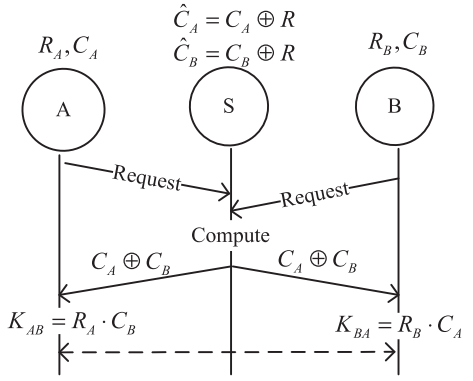


Fig. 4. LLK establishment procedure with XOR NC.

Similarly, the communication node can also derive other nodes' column by performing the XOR operations on $\widehat{C}_i$ with its own column. Then, a secure pairwise key can be established according to Eq. (6). The detailed procedure is illustrated in Fig. 4. It should be mentioned that XOR network coding based on the operations over $GF(2)$ can also be extended to the TLK establishment and node authentication procedure to reduce the computation cost. The reduction of computation cost is achieved by sacrificing partial security, which is demonstrated to be acceptable in some special scenarios [20].

6. Security analysis

In this section, we analyze the robustness of the proposed scheme from three aspects: (1) the security impact of compromising a node; (2) the reliability against compromising a certain matrix; and (3) the resilience against collusion attacks. We also compare our scheme with other counterparts.

6.1. Security impact of compromising a node

Unlike the previous assumption that sensors are tamper-proof, our assumption is that sensors are prone to node compromising and collusions, both of which may incur the leakage of confidential key information. When a node is captured, the attacker can obtain two types of secrets: vectors of L_i and U_i for its own key establishment and RLNC-encoded matrix $\widehat{U}_j$ for other nodes' key negotiations. For the former type of information, since each sensor stores one vector of L_i and the corresponding vector U_i for each $M_i \in S_1$, attackers could easily derive M_i after compromising sufficient nodes, which is a universal problem of key pre-distribution schemes. In the following subsection, we will discuss the resilience of the proposed scheme in this scenario. For the latter type of confidential information, each node stores the entire RLNC-encoded matrix $\widehat{U}_j$, the information-theoretical security of which is shown in the following theorem:

Theorem 1. *For the matrix M_j, the elements are randomly selected from GF(q). Let $H(\cdot)$ denote the entropy function of information theory, and then we have $H(C_j(k)) = H(C_j(k)|\widehat{C}_j(1), \ldots, \widehat{C}_j(\lambda))$.*

Theorem 1 shows that the probability of guessing a given column is unaltered with some additional RLNC-encoded information. In other words, it is difficult to discover the matrix U_j given the RLNC-encoded matrix $\widehat{U}_j$. Thus, storing the RLNC-encoded matrices at sensor nodes will not impair the security of the proposed scheme. Even though attackers can get the matrix U_j, they are also required to have the corresponding matrix L_j to compute M_j, which is also a difficult task.

6.2. The reliability against compromising a certain matrix

Though it is impossible that compromising one node will destroy our scheme, malicious nodes can still obtain all the keys with small probability after compromising a large number of sensors. In this subsection, we analyze the reliability in terms of the average number of compromised nodes required to obtain a certain matrix and have the following theorem:

Theorem 2. *The average number of nodes required to obtain a given matrix is $E(N_{th}) = \frac{\lambda(2-p)H_\lambda}{n(1-p)^2}$, where H_λ is the harmonic number and $p = \frac{n-1}{n}$.*

Proof. To get the matrix M_i, $E(N_{th})$ sensors on average should cover the entire vectors of L_i and U_i decomposed from M_i. In our scheme, each node randomly selects τ matrices and then stores one vector of L_i and the corresponding vector of U_i for each matrix. Thus, we can consider the whole procedure as two independent steps, i.e., the random selection of matrices and vectors, so that we have

$$E(N_{th}) = E(N_v) \cdot E(N_m), \tag{11}$$

where $E(N_v)$ is the average number of nodes required to cover all the vectors of L_i and U_i for the matrix M_i if sensors contain the vectors of L_i and U_i, and $E(N_m)$ denotes the average number of selected sensors, among which there must be a sensor containing the information related to M_i.

The computation of $E(N_v)$ is very similar to the classic coupon collector problem which describes the "collect all coupons and win" contest. The problem can be represented as follows. For a matrix, there exist λ different vectors, from which vectors are collected with replacement. Each sensor node randomly chooses a vector with probability $\frac{1}{\lambda}$. Then, the average number of trials required to collect all the vectors is

$$E(N_v) = \frac{\lambda}{\lambda} + \frac{\lambda}{\lambda - 1} + \cdots + \frac{\lambda}{1} = \lambda H_\lambda, \tag{12}$$

where H_λ is called the harmonic number.

The $E(N_m)$ can be calculated as follows. From n different matrices, each sensor randomly chooses a matrix with probability of uniform distribution. Then, the average number of trials required to collect a given matrix M_i is

$$E(N_m) = \sum_{k=1}^{\infty} k \left(\frac{n-1}{n}\right)^{k-1} \frac{1}{n}. \tag{13}$$

It also can be represented as

$$nE(N_m) = p^0 + 2p^1 + 3p^2 + \cdots, \tag{14}$$

where $p = \frac{n-1}{n}$. Furthermore, we can get $E(N_m)$ by subtracting $npE(N_m)$ from $nE(N_m)$. Thus, $E(N_m)$ can be denoted as

$$E(N_m) = \frac{2-p}{n(1-p)^2}. \tag{15}$$

Finally, according to Eq. (12) and (15), we have

$$E(N_{th}) = \frac{\lambda(2-p)H_\lambda}{n(1-p)^2}, \tag{16}$$

where H_λ is the harmonic number and $p = \frac{n-1}{n}$. □

6.3. The resilience against collusion attacks

When the number of compromised nodes exceeds the threshold, i.e., $N_x \geqslant E(N_{th})$, some additional keys between uncompromised sensors might be exposed, and the robustness of the scheme diminishes. In this subsection, we analyze the resilience of the proposed scheme against collusion attacks in terms of the additional key exposure probability and have the following theorem:

Theorem 3. *Let the number of compromised nodes be N_x, $N_x \geqslant E(N_{th})$, then the probability of additional keys being exposed among uncompromised nodes satisfies $Pr = \sum_{j=\lambda}^{x} C_x^j \theta^j (1-\theta)^{x-j}(1-\xi^j)^\lambda$, where $\theta = \frac{\tau}{n}$ and $\xi = \frac{\lambda-1}{\lambda}$.*

Proof. Let $\mathcal{S}_x$ be the event that x nodes are compromised, $\mathcal{M}_i$ be the event that the matrix M_i is obtained by attackers, and $\mathcal{A}_i$ be the event that the key is calculated from the compromised matrix M_i. Then, the resilience against collusion attacks, i.e., the probability of obtaining a shared key between uncompromised nodes, can be expressed as

$$Pr = Pr(\mathcal{A}_1 \cup \mathcal{A}_2 \cup \cdots \cup \mathcal{A}_n | \mathcal{S}_x). \tag{17}$$

According to the definition of independent events, Pr can also be denoted as

$$Pr = \sum_{i=1}^{n} Pr(\mathcal{A}_i|\mathcal{S}_x) = n \cdot \frac{1}{n} Pr(\mathcal{M}_i|\mathcal{S}_x), \tag{18}$$

where $Pr(\mathcal{M}_i|\mathcal{S}_x)$ is the probability that M_i is obtained by attackers when x nodes are compromised. Since each node randomly selects τ matrices to constitute the set S_1 and stores one vector of L_i and the corresponding vector of U_i for each matrix in S_1, a given matrix M_i is chosen with probability $\theta = \frac{\tau}{n}$, and $j, j \geqslant \lambda$ sensors containing the information related to M_i can cover the entire vectors of L_i and U_i with probability $(1-\xi^j)^\lambda$, where $\xi = \frac{\lambda-1}{\lambda}$. Thus, we have

$$Pr(\mathcal{M}_i|\mathcal{S}_x) = \sum_{j=\lambda}^{x} C_x^j \theta^j (1-\theta)^{x-j}(1-\xi^j)^\lambda. \tag{19}$$

From Eq. (18) and (19), we have

$$Pr = \sum_{j=\lambda}^{x} C_x^j \theta^j (1-\theta)^{x-j}(1-\xi^j)^\lambda, \tag{20}$$

where $\theta = \frac{\tau}{n}$ and $\xi = \frac{\lambda-1}{\lambda}$. □

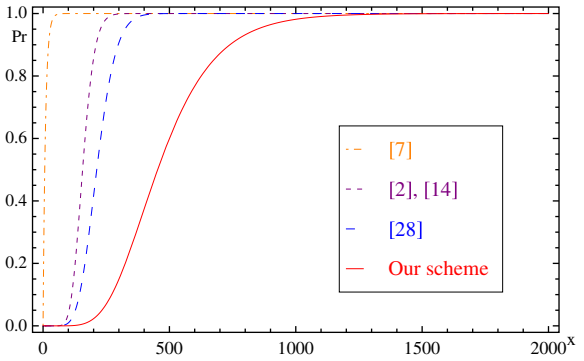


Fig. 5. The resilience against collusion attacks.

Compared with our scheme, Eschenauer et al.'s scheme [7] has the additional key exposure probability

$$Pr_1 = 1 - \left(1 - \frac{\tau}{S}\right)^x, \tag{21}$$

where τ denotes the number of keys that each node selects from a global key set of size S, and x is the number of compromised nodes. Du et al.'s scheme [2] and Dai et al.'s scheme [14] have the same additional key exposure probability when x sensors have been compromised. This probability can be denoted as

$$Pr_2 = \sum_{j=\lambda+1}^{x} C_x^j \left(\frac{\tau}{n}\right)^j \left(1 - \frac{\tau}{n}\right)^{x-j}. \tag{22}$$

Hypercube-2D [30] adopts t-degree bivariate polynomials to achieve the key agreement. In this scheme, the probability of exposing a polynomial is given by

$$Pr_3 = \sum_{j=t+1}^{A} \frac{C_A^j C_{N_A}^{x-j}}{C_N^x}, \tag{23}$$

where A is the number of nodes sharing a t-degree bivariate polynomial, and N is the total number of nodes.

Fig. 5 shows the numerical results of different impressive work with the following settings: $\tau = 10$, $S = 100$, $n = 100$, $\lambda = 15$, $t = 10$, $A = 100$, and $N = 2000$. It can be seen that the schemes in [2,7,14] have high key exposure probability, and Hypercube-2D [30] improves the results. The additional key exposure probability of our scheme becomes much lower than that of other schemes with the increasing number of compromised nodes. The numerical results demonstrate that the proposed scheme significantly outperforms the others in terms of the resilience against collusions.

7. Performance evaluation

In this section, we evaluate the performance in terms of the local connectivity, communication overhead, memory usage, node authentication probability, energy consumption, and computation cost, all of which are significant for the key pre-distribution in sensor data storage.

7.1. Local connectivity

Local connectivity is defined as the probability that a node can establish a LLK with its neighbors. Let the transmission range of sensor be r and the average distance between two neighbors A and B be d, i.e., $|AB| = d$. We assume that all the sensors are uniformly distributed into a region of area A, thus the density of sensor is denoted as $\rho = \frac{N}{A}$. According to the communication model in [2], we can express the intersected area S as

$$S = 2\left(\frac{\pi r^2}{2\pi}\theta - \frac{1}{2}\cdot\frac{1}{2}d_1\right), \tag{24}$$

where $d_1 = 2\sqrt{r^2 - (1/2)^2}$ and $\theta = 2\arccos(d/2r)$. Then, the number of nodes in the intersected region can be denoted as $N_S = \rho S$.

During LLK establishment, nodes A and B separately send the key negotiation request containing τ matrix indexes and vector indexes. The probability that two neighbors have the common matrix and the average number of common matrix are respectively

$$P_C = \sum_{i=1}^{\tau} \frac{C_\tau^i C_{n-\tau}^{\tau-i}}{C_n^\tau} = 1 - \frac{C_{n-\tau}^\tau}{C_n^\tau}, \tag{25}$$

$$E(N_c) = \sum_{i=1}^{\tau} i\frac{C_\tau^i C_{n-\tau}^{\tau-i}}{C_n^\tau}. \tag{26}$$

Since $iC_\tau^i = \tau C_{\tau-1}^{i-1}$, Eq. (26) can also be denoted as

$$E(N_c) = \frac{\tau}{C_n^\tau}\sum_{i=1}^{\tau} C_{\tau-1}^{i-1}C_{n-\tau}^{\tau-i} = \frac{\tau C_{n-1}^{\tau-1}}{n}. \tag{27}$$

As previously mentioned, N_S nodes exist in the intersected region, and each node contains σ matrices for others' key negotiations; thus, the probability of establishing a secret key between nodes A and B can be denoted as

$$P_K = \begin{cases} 1 - \left(\frac{C_{n-\sigma}^\gamma}{C_n^\gamma}\right)^{N_S} & \text{if } \gamma + \sigma \leqslant n, \\ 1 & \text{otherwise}, \end{cases} \tag{28}$$

where $\gamma = \lfloor E(N_c) \rfloor$. Note that Eq. (28) also indicates the relation between n and N.

If $\gamma + \sigma > n$ is satisfied, two neighbors can always establish a shared key. According to Eq. (28) and (27), we can have

$$\tau^2 + \sigma n - n^2 > 0, \tag{29}$$

which is the condition that any two neighbors can establish a shared key with probability one.

In Table 1, we present the local connectivity of our scheme. It can be seen that the local connectivity is increased with the decrease of distance d, and it is also a monotonic increasing function of τ. Another impressive result is that the local connectivity almost approaches one although Eq. (29) is not satisfied, indicating that Eq. (29) is a sufficient but not necessary condition of $P_K = 1$.

Compared with our scheme, Du et al.'s scheme [2] has the local connectivity $P_{K1} = 1 - C_n^\tau C_{n-\tau}^\tau/(C_n^\tau)^2$, the numerical results of which are also presented in Table 1. The local connectivity of Du et al.'s scheme is a little smaller than that of our scheme with comparable parameters. The improvement of our scheme is achieved by sacrificing the memory usage, which is acceptable for sensor data storage. Moreover, it should be noted that the local connectivity of our scheme is mainly determined by two parameters: τ and σ, while the local connectivity of Du et al.'s scheme is decided by the parameter τ. We can adapt our scheme to various situations by adjusting two parameters τ and σ, which enhances the scalability and feasibility of the proposed scheme.

In addition, both of the local connectivity in [8] and [20] are one due to the central mobile node and location knowledge, respectively. Without the prior location information, our scheme can also achieve the same performance.

7.2. Communication overhead

The contribution of RLNC in our scheme is that it can effectively improve the communication efficiency which is a primary concern for sensor data storage. In Fig. 6, we present an example to demonstrate that our scheme outperforms others in terms of the expected transmission number. In Du et al.'s and Dai et al.'s schemes, two neighbors are required to exchange their key information after sending the requests, and this procedure needs four transmissions in total. In our scheme, we only need three transmissions via an intermediate node. Compared to Dai et al.'s scheme [14], our scheme roughly reduces the communication overhead by 25%. We also present another instance to further demonstrate the communication efficiency in Fig. 7. In Du et al.'s and Dai et al.'s schemes, they both require eight transmissions for two pairs of neighbors to negotiate communication keys. In our scheme, if both nodes A and B have $C_M \oplus C_N$ and both nodes M and N contain $C_A \oplus C_B$, the intermediate node S requires sending message $C_A \oplus C_B \oplus C_M \oplus C_N$ in one transmission instead of four transmissions. Thus, the expected transmission number is significantly reduced. Additionally, the more such cases exist in our scheme, the less communication resources are required. In conclusion, the proposed scheme is communication-efficient.

Table 1
Local connectivity.

	τ	5	6	7	8	9	10
Our Scheme	$d = 18$	0.927	0.927	0.996	1.000	1.000	1.000
	$d = 17$	0.992	0.992	1.000	1.000	1.000	1.000
	$d = 16$	0.999	0.999	1.000	1.000	1.000	1.000
Reference [2]	$n = 20$	0.806	0.923	0.978	0.996	1.000	1.000

Note: numerical results derived from Eq. (28) with the following parameters: $n = 20$, $r = 10$, $\sigma = 4$ and $\rho = 1$.

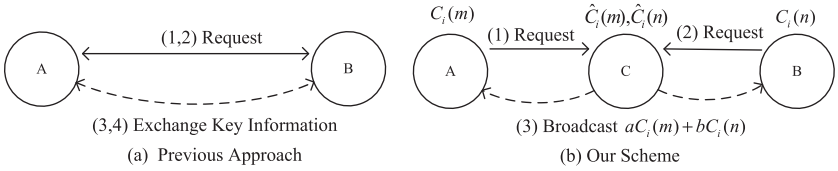


Fig. 6. Key information exchange: (a) Du et al.'s and Dai et al.'s schemes (b) our scheme.

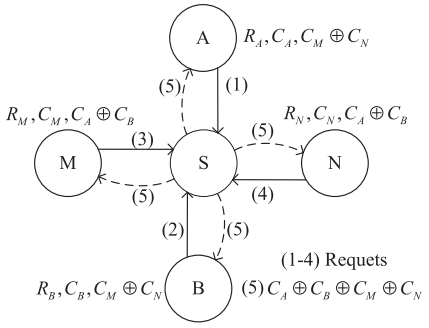


Fig. 7. Efficient LLK agreement using XOR NC.

In the following, we present the upper bound of communication overhead in our scheme. Since the nodes' ID and index are denoted in binary, we can have *sizeof(ID)* = $\log_2 N$, *sizeof(IndexU$_i$)* = $\log_2 n$, and *sizeof(IndexC$_i$)* = $\log_2 \lambda$. During the key establishment, two neighbors are required to broadcast their requests, and an intermediate node responsible for their key negotiation responds to the requests. The communication overheads in these two steps are respectively $2(\log_2 N + \tau(\log_2 n + \log_2 \lambda))$ and $(\lambda + 2)\kappa$, where κ is the size of M_i's element picked from $GF(q)$. Since the former part is so small that it can be omitted, the overall communication overhead is about $(\lambda + 2)\kappa$ for each shared key. In sensor data storage with N nodes, each sensor has $\rho\pi r^2$ neighbors, and the number of LLKs is $N\rho\pi r^2/2$ in the entire networks. Thus, the upper bound of communication overhead in our scheme can be approximately denoted as $(N\rho\pi r^2\kappa\ (\lambda + 2))/2$.

7.3. Memory Usage

In our scheme, each sensor is required to store two types of vectors, among which some are related to set S_1 for its own key establishment and others are related to set S_2 for others' key negotiations. For each matrix $M_i \in S_1$, each sensor is required to store a row of L_i and the corresponding column of U_i; thus, the memory usage for its own key establishment is $2\lambda\tau\kappa$. In addition, a node is required to store λ columns and 2λ coefficients for each matrix in S_2. The memory usage for these data can be expressed as $(\sigma\lambda^2 + 2\lambda\sigma)\kappa$. Thus, the overall memory usage of each sensor is $\lambda(2\tau\kappa + \sigma\lambda\kappa + 2\sigma)$, which can be further optimized with the technique in [14].

We present an instance to demonstrate the feasibility of our scheme in terms of memory usage. Let $\lambda = 15$, which is extremely large for most of applications, $\kappa = 1$ byte, $\tau = 7$, and $\sigma = 4$. The overall memory usage is about 1*K*, which is acceptable for sensors. In addition, the local connectivity in the setting approximately approaches one according to Table 1. Consequently, our scheme is feasible for sensor data storage from the aspect of memory usage.

7.4. Node authentication probability

In this subsection, we present the performance of node authentication and have the following theorem:

Theorem 4. *In the node authentication, a node can be verified by its neighbors with the probability*

$$P_{NA} = 1 - \left(1 - \sum_{i=2}^{min\{\tau,\sigma\}} \frac{C_\tau^i C_{n-\tau}^{\sigma-i}}{C_n^\sigma}\right)^{N_S}.$$

Proof. The probability of a node being authenticated by a given neighbor can be denoted as $P_{Au} = \sum_{i=2}^{m} \frac{C_\tau^i C_{n-\tau}^{\sigma-i}}{C_n^\sigma}$, where $m = min\{\tau,\sigma\}$, and the probability that a node cannot be verified by any neighbors is $P_{NA} = (1 - P_{Au})^{N_S}$; thus, Theorem 4 holds. □

7.5. Energy consumption

Energy consumption is the function of the expected transmission number and data length in each transmission. As mentioned above, the communication overhead for each key establishment is approximately $(\lambda + 2)\kappa$ bits in our scheme. Then, the average energy consumption is about $(\lambda + 2)\kappa(En_{tr} + 2En_{rec}) + En_{com}$, where En_{tr}, En_{rec}, and En_{com} are the average energy consumptions for transmission, reception, and computation, respectively. Since $En_{tr} = 2En_{rec}$ and $En_{tr} \ll En_{com}$ according to [31], the average consumption for each key is about $4(\lambda + 2)\kappa En_{rec}$. Then, the overall energy consumption can be approximately computed as $2N\rho\pi r^2\kappa En_{rec}\ (\lambda + 2)$ for sensor data storage. In the traditional matrix decomposition scheme [14], the overall energy consumption is about $(2(\lambda + 2)\kappa(En_{tr} + En_{rec}) + En_{com})\cdot(N\rho\pi\ r^2/2) \approx 3\ N\rho\pi r^2(\lambda + 2)\kappa En_{rec}$. Thus, we have less energy consumptions than the traditional matrix decomposition scheme.

7.6. Computation cost

Recall that all the calculations in the key pre-distribution phase are performed by the offline server, and we mainly analyze the computation cost of sensor node in the key agreement phase. For each pairwise key, the sensor node needs to calculate Eq. (5) and (6), which involves λ additions,

($\lambda + 3$) multiplications and 3 divisions. We can rely on look-up tables, rather than additions, to perform the multiplication and division. Moreover, we have the simplified LLK scheme for the scenarios with computationally-weak nodes. Thus, our proposal is acceptable for distributed sensor data storage in terms of computation cost.

8. Conclusions and future work

Key pre-distribution plays a key role in securing communications among vulnerable nodes in distributed sensor data storage, where the robustness, efficiency, and scalability are critical features. In this paper, we have proposed a novel key pre-distribution scheme based on network coding and LU matrix decomposition for distributed sensor data storage. Compared with the previous counterparts, our scheme is remarkably robust against node-compromised attacks and much resilient against collusion attacks. Moreover, the proposed scheme can improve the performance such as the expected transmission number and local connectivity. Finally, our scheme is much scalable for distributed sensor data storage networks.

For our future work, we will extend the scheme to group key distribution and multi-hop secret key distribution in hostile or volatile networks with limited transmission capabilities.

Acknowledgment

This work is supported by the National Grand Fundamental Research 973 Program of China (Nos. 2011CB302703, and 2010CB328105, and the National Natural Science Foundation of China (Nos. 60970101, 60932003, 61033001, 61073174, and 61061130540).

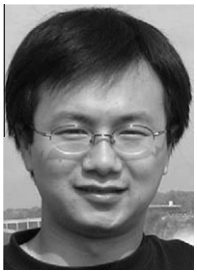
References

[1] Q. Wang, K. Ren, W. Lou, Y. Zhang, "Dependable and secure sensor data storage with dynamic integrity assurance", in: Proc. IEEE INFOCOM, 2009, pp. 954–962.
[2] W. Du, J. Deng, Y.S. Han, P.K. Varshney, J.K. Katz, A. Khalili, A pairwise key predistribution scheme for wireless sensor networks, in: ACM Trans. Inform. System Security, vol. 8, no. 2, 2005, pp. 228–258.
[3] D. Boneh, M. Franklin, "Identity-based encryption from the Weil pairing", in Proc. Crypto, 2001, pp. 213–229.
[4] Y. Zhang, W. Liu, W. Lou, Y. Fang, Securing mobile ad hoc networks with certificateless public keys, IEEE Trans. Dependable Secure Comput. 3 (4) (2006) 386–399.
[5] N. Potlapally, S. Ravi, A. Raghunathan, N. Jha, A study of the energy consumption characteristics of cryptographic algorithms and security protocols, IEEE Trans. Mobile Comput. 5 (2) (2006) 128–143.
[6] Y. Jiang, C. Lin, M. Shi, X. Shen, Seal-healing group key distribution with time-limited user revocation for wireless sensor networks, Ad Hoc Netw. Elsevier 5 (1) (2007) 14–23.
[7] L. Eschenauer, V.D. Gligor, A key management scheme for distributed sensor networks, in: Proc. ACM Conf. Comput. Commun. Security, 2002, pp. 41–47.
[8] H. Chan, A. Perrig, D. Song, Random key predistribution schemes for sensor networks, in: Proc. IEEE Symp. Security Privacy, 2003, pp. 197–213.
[9] R.D. Pietro, L.V. Mancini, A. Mei, Efficient and resilient key discovery based on pseudo-random key pre-deployment, in: Proc. IEEE IPDPS, 2004, pp. 217–224.
[10] S. Zhu, S. Xu, S. Setia, S. Jajodia, Establishing pairwise key for secure communication in ad-hoc network: a probabilistic approach, in: Proc. IEEE ICNP, 2003, pp. 326-335.
[11] Y. Zhou, Y. Fang, A two-layer key establishment scheme for wireless sensor networks, IEEE Trans. Mobile Comput. 6 (9) (2007) 1009–1020.
[12] W. Du, J. Deng, Y.S. Han, S. Chen, P.K. Varshney, A key management scheme for wireless sensor networks using deployment knowledge, in: Proc. IEEE INFOCOM, 2004, pp. 586–594.
[13] Y. Zhou, Y. Zhang, Y. Fang, LLK: a link-layer key establishment scheme in wireless sensor networks, in: Proc. IEEE WCNC, 2005, pp. 1912–1926.
[14] H. Dai, H. Xu, A key predistribution scheme with matrix decomposition for secure wireless sensor networks, in: Proc. IEEE World Congr. Intell. Control Automat., 2008, pp. 1724–1727.
[15] R. Blom, An optimal class of symmetric key generation systems, in: Proc. ACM EUROCRYPT, 1985, pp. 335–338.
[16] D. Liu, P. Ning, R. Li, Establishing pairwise keys in distributed sensor networks, ACM Trans. Inform. Syst. Security 8 (1) (2005) 41–47.
[17] S.A. Camptepe, B. Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks, IEEE/ACM Trans. Netw. 15 (2) (2007) 346–358.
[18] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, Perfecly-secure key distribution for dynamic conferences, in: Proc. ACM Adv. Cryptology, 1992, pp. 471-486.
[19] F. Delgosha, F. Fekri, A multivariate key establishment scheme for wireless sensor networks, IEEE Trans. Wireless Commun. 8 (4) (2009) 1814–1824.
[20] P.F. Oliveira, J.W. Daly, A network coding approach to secret key distribution, IEEE Trans. Inform. Forensics Security 3 (3) (2008) 414–423.
[21] Y. Jiang, C. Lin, X. Shen, M. Shi, A DoS and fualt tolerant authentication protocol for group communications in ad hoc networks, Comput. Commun., Elsevier 30 (1) (2007) 2428–2441.
[22] Y. Jiang, C. Lin, X. Shen, M. Shi, Multiple key sharing distribution scheme with (n,t) threshold memchanismfor NEMO group communications, IEEE J. Selected Areas Commun. 24 (9) (2006) 1738–1747.
[23] Y. Wu, P. Chou, S. Kung, Minimum-energy multicast in mobile ad hoc networks using network coding, IEEE Trans. Commun. 53 (11) (2005) 1906–1918.
[24] P.A. Fragouli, Y. Wu, Network coding for internet and wireless networks, IEEE Signal Process. Mag. 24 (5) (2007) 77–85.
[25] P. Zhang, Y. Jiang, C. Lin, Y. Fan, X. Shen, P-coding: secure network coding against eavesdropping attacks, in: Proc. IEEE INFOCOM, 2010, pp. 1–9.
[26] Y. Fan, Y. Jiang, H. Zhu, X. Shen, An efficient privacy-preserving scheme against traffic analysis attacks in network coding, in: Proc. IEEE INFOCOM, 2009, pp. 2213–2221.
[27] S. Fragouli, J.Y. Boudec, J. Widmer, Network coding: an instant primer, ACM SIGCOMM Commun. Rev. 36 (1) (2008) 63–68.
[28] R. Ahlswede, N. Cai, S.R. Li, R.W. Yeung, Network information flow, IEEE Trans. Inform. Theor. 46 (4) (2000) 1204–1216.
[29] R.L. Burden, J.D. Faires, Numerical analysis, eighth ed., Brooks-Cole, 2004.
[30] D. Liu, P. Ning, R. Li, Establishing pairwise keys in distributed sensor networks, ACM Trans. Inform. Syst. Security 8 (1) (2005) 41–77.
[31] G. Mathur, P. Desnoyers, D. Ganesan, P. Shenoy, Ultra-low power data storage for sensor networks, in: Proc. of IEEE IPSN, 2006, pp. 374–381.

Rongfei Zeng received a B.S. degree (2002) from Northeastern University (China) in Computer Science and Technology. Currently, he is a Ph.D. candidate at Computer Science Department, Tsinghua University, China. His current research interests include wireless network security and performance evaluations.

Yixin Jiang is an associate professor in Tsinghua University. In 2007–2009, he was a Post Doctorial Fellow with University of Waterloo. He received the Ph.D. degree (2006) from Department of Computer Science and Technology, Tsinghua University, China. In 2005, he was a Visiting Scholar with the Department of Computer Sciences, Hong Kong Baptist University. In 2009, he was a Visiting Scholar with the Department of Computer Science and Engineering, the Chinese University of Hong Kong. He has served as the Technical Program Committee (TPC) member for main network conferences, such as IEEE ICCCN, IEEE GLOBECOM, IEEE ICC, IEEE WCNC, etc. He is a member of IEEE CISTC. His current research interests include network coding, clouding computing, security and privacy in wireless communication and mobile computing. He has received Excellent Backbone Talents Fund Award, Outstanding Doctoral Graduate Award, and Excellent Doctoral Thesis Award of Tsinghua University.

Chuang Lin (IEEE SM'04) is a professor of the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He received the Ph.D. degree in Computer Science from the 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 300 papers in research journals and IEEE conference proceedings in these areas and has published three books. Professor Lin is a member of ACM Council, a senior member of the IEEE and the Chinese Delegate in TC6 of IFIP. He serves as the Technical Program Vice Chair, the 10th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS 2004); the General Chair, ACM SIGCOMM Asia workshop 2005; the Associate Editor, IEEE Transactions on Vehicular Technology; the Area Editor, Journal of Computer Networks; and the Area Editor, Journal of Parallel and Distributed Computing.



Yanfei Fan received the M.E. degree (2005) from Tsinghua University, China, and the B.E. degree (2002) from Beijing University of Posts and Telecommunications, China, all in Computer Science. He is currently pursuing his Ph.D. degree in the Department of Electrical and Computer Engineering at University of Waterloo, Canada. His research interests include network coding, security in wireless communication and mobile computing.

Xuemin (Sherman) Shen (IEEE M'97-SM'02-F'09) received the B.Sc. (1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a University Research Chair Professor, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on mobility and resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks and vehicular ad hoc and sensor networks. He is a co-author of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control and filtering. He is a Distinguished Lecturer of IEEE Communications Society. He serves as the Tutorial Chair for IEEE ICC'08, the Technical Program Committee Chair for IEEE Globecom'07, the General Co-Chair for Chinacom'07 and QShine'06, the Founding Chair for 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; Associate Editor for IEEE Transactions on Vehicular Technology; KICS/IEEE Journal of Communications and Networks, Computer Networks; ACM/Wireless Networks; and Wireless Communications and Mobile Computing (Wiley), etc. He has also served as Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. He received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004 and 2008 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 and 2007 from the Faculty of Engineering, University of Waterloo. He is a registered Professional Engineer of Ontario, Canada.