A Polynomial-Based Key Predistribution Scheme for Wireless Sensor Networks Using Matrix Decomposition

Hong Zheng
School of Automation Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan, China
Email: macrozheng@uestc.edu.cn

Hangyang Dai
School of Automation Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan, China
Email: daihang1981@sina.com

***Abstract*— Key predistribution is one of the most challenging issues for secure communication in wireless sensor networks. But most of existing schemes are not scalable due to their linearly increased communication and key storage overhead. Furthermore, these protocols cannot provide sufficient security when the number of compromised nodes exceeds a critical value. In this paper, we propose a polynomial-based key predistribution scheme using matrix decomposition. Our scheme guarantees that any two sensor nodes can find a shared key between themselves. The analysis in this paper indicates that the existing schemes require a large number of keys in each sensor node to display a comparable connectivity as the proposed scheme which allows almost 100% connectivity regardless of the number of keys, and it develops an efficient encoding mechanism to optimize the network-wide storage usage. On the other hand, compared with existing schemes, our scheme has better performance in network resilience.**

***Keywords*-wireless sensor works; security; matrix decomposition; polynomial-based key predistribution**

I. INTRODUCTION

Wireless sensor networks (WSNs) [1] increasingly become viable solutions to many challenging problems for both military and civilian applications, including target tracking, battlefield surveillance, intruder detection and scientific exploration. Wireless sensor networks consist of a large number of sensor nodes which have limited energy resources, computation ability and wireless communication range. Generally, WSNs are deployed in hostile environments and operated in unattended mode. Hence, security mechanisms that provide confidentiality and authentication are critical for the operation of many sensor applications. As the basic requirement for providing security functionality, key management plays a central role in data encryption and authentication. The key problem in key management is how to secure the communication between the sensor nodes, i.e. how to set up secret keys between communication nodes. However, due to resource and energy constrains in sensor nodes, many ordinary security mechanisms such as public key-based authentication and corresponding key management schemes are impractical and infeasible for WSN.

In this paper, we present a key predistribution scheme based on polynomial-based key predistribution [2] [3] and matrix decomposition [4]. It guarantees that any two sensor nodes can find a common key between themselves by using a pool of polynomial formed in the symmetric matrix format and matrix decomposition. Our analysis shows that the existing schemes require a large number of keys in each sensor node to display a comparable connectivity as the proposed scheme which allows almost 100% connectivity regardless of the number of keys, and it enhances the secret of the exchanged row information of the polynomials. In addition, when network-wide storage usage is reduced, the security tolerance has been greatly improved.

The remainder of this paper is organized as follows. Section 2 describes the existing key distribution schemes for wireless sensor networks. Section 3 introduces the polynomial-based key predistribution approach in detail. Section 4 presents the key predistribution scheme with matrix decomposition. Section 5 deals with the detailed performance analysis and comparisons. Finally, Section 6 concludes this paper and points out the future research directions.

II. RELATED WORKS

Based on the characteristic of WSN, the practical key management scheme for WSN would be key predistribution approach. Key predistribution approach which belongs to symmetric encryption algorithm is that key information is distributed to all sensor nodes prior to deployment.

Eschenauer and Gligor [5] (E-G scheme) proposed a random key predistribution scheme which is based on Random Graph Theory [6] and Probability Theory. According to this scheme, each sensor node receives a random subset of keys from a large key pool as the node's key ring before deployment, and stores them in its memory. After nodes have been deployed in the designated area, two neighboring nodes can find at least one common key in their key rings and use the key as their shared key. Chan et al. [7] improved E-G scheme

978-1-4244-1674-5/08 /$25.00 ©2008 IEEE

and developed q -composite key establishment scheme and random pairwise key scheme. The q -composite key establishment scheme requires that two sensor nodes share at least q pre-computed keys as the basis to establish a pairwise key between the two sensor nodes. In the random pairwise key scheme, random pairwise keys are established between a specific sensor node and a random subset of other nodes. Du et al. [3] proposed another key predistribution scheme which substantially improves the resilience of the network. This scheme exhibits a threshold: when the number of compromised nodes is smaller than the threshold, the probability that any node other than the compromised nodes is affected is close to zero.

In preceding schemes, the key performance's indices: network connectivity, resilience against node capture and memory usage are not satisfactory. Sometimes, the high network connectivity is achieved by reducing the ability of resilience against node capture and increasing the memory usage of the network. The proposed approach in our scheme allows the preferable trade-off among network connectivity, security against node capture and memory usage to a great extent.

III. The Polynomial-Based Key Predistribution Scheme

To predistribution pairwise keys, the key predistribution server first randomly generates a bivariate t degree polynomial $f(x,y)=\sum_{i,j=0}^{t} a_{ij}x^i y^j$ over a finite field F_q, where q is a prime number that is large enough to accommodate a cryptographic key, such that it has the property of $f(x,y)=f(y,x)$ .Then, for each node i , the setup server computes a polynomial share of $f(x,y)$, that is $f(i,y)$ and stores it in sensor node i . For any two sensor nodes i , j , node i can compute the pair-wise key $f(i,j)$ by evaluating $f(i,y)$ at point j , and node j can compute the pair-wise key $f(j,i)$ by evaluating $f(j,y)$ at point i . From the property of symmetry of $f(x,y)$, $f(i,j)=f(j,i)$. So the pair-wise key between nodes i and j can be established.

In this scheme, each sensor node needs to store a bivariate t-degree polynomial's coefficients, which would occupy $(t+1)\log_2 q$ storage space. The security proof in [2] ensures that this scheme is unconditionally secure and t -collision resistant. In other words, the coalition of no more than t compromised sensor nodes knows nothing about the pairwise keys between any two non-compromised sensor nodes.

IV. The Proposed Scheme

In this section we briefly describe how the proposed key predistribution scheme works. Based on E-G scheme, the proposed scheme uses matrix decomposition technique and the polynomial-based key predistribution approach. It makes sure that any two sensor nodes can find one shared key between themselves along with mutual authentication. Moreover, compared with E-G scheme, our scheme has the stronger ability in resilience against node capture.

A. Preliminaries

Firstly, we define some important properties of matrix used in our scheme.

Definition 1. If a square matrix K has the property $K^T=K$, where transpose of matrix K is denoted by K^T , we say that K is a symmetric matrix. A symmetric matrix means that $K_{ij}=K_{ji}$, where K_{ij} is the element in the i th row and j th column of matrix K .

Definition 2. In order to realize node-to-node mutual authentication, we introduce LU matrix decomposition which is one of triangle decomposition approaches. LU decomposition is to decompose a $m\times m$ matrix K into two matrices such that $K=LU$, where L is a $m\times m$ lower triangular matrix and U is an $m\times m$ upper triangular matrix, respectively.

Secondly, we list the symbols used in the following sections in Table I.

TABLE I. THE SYMBOLS USED IN THE FOLLOWING SECTIONS

Symbol	meaning
P	network connectivity
S	size of the key pool
k	the number of keys of each node in E-G scheme or key capacity of each node in our scheme
x	the number of compromised nodes
τ	the number of polynomials in each node
ω	size of polynomial pool
t	the security threshold

B. The Proposed Key Predistribution Scheme

The proposed key predistribution scheme consists of four steps:

Step1. Generate a large pool of polynomial. In this scheme the setup server randomly generates a large pool of bivariate t-degree polynomials over the finite field F_q , and each polynomial has a unique ID. Each node can pick a subset of polynomials from the large pool of polynomials.

Step2. Form a symmetric matrix using the pool of polynomials. The proposed scheme uses a pool of polynomials formed in a $m\times m$ symmetric matrix.

Step3. Decompose the formed symmetric matrix. We apply LU matrix decomposition to the symmetric matrix. This approach lets a pair of nodes find a common polynomial, and two sensor nodes can find one shared key using the polynomial-based key predistribution discussed in Section 3.The formed symmetric matrix is decomposed into a $m\times m$ lower triangular matrix L and an $m\times m$ upper triangular matrix U .

Step4. Predistribute keys. Every sensor node is randomly distributed one row from the matrix L and one column from the matrix U , respectively. In this step both the row and column which are distributed to each sensor node have a same number. (i.e. L_{ri} : i th row of L and U_{cj} : j th column of U)

Since key predistribution has been finished, any two sensor nodes can find a shared key according to the following method: assume any two sensor nodes A (contains L_{ri} and U_{ci}) and B (contains L_{rj} and U_{cj}), they first exchange their rows, and a vector product in (1):

node A: $L_{rj} \times U_{ci} = K_{ji}$ (1)

node B: $L_{ri} \times U_{cj} = K_{ij}$

K is a symmetric matrix, thus $K_{ij} = K_{ji}$. K_{ji} (or K_{ij}) is used as a common polynomial between the two sensor nodes. Then using the polynomial-based key predistribution discussed in Section 3, node A and node B always find a shared key between themselves.

Furthermore, for storing the row and column information in the sensor node, we apply an encoding technique to increase the efficiency of memory usage throughout the network. Each row of L or each column of U matrices has two parts: one non-zero-element part and another zero –element part (might be absent). Therefore, to store one row of L and one column of U in each sensor node, we only need to store each element in the non-zero-element part and one value specifying the number of following zeros in zero-element part. The non-zero-element part could actually contain one or more zeros which are treated as non-zero values and stored accordingly. When the size of the network is large or the dimension of the symmetric matrix is large, the storing method is extremely effective.

For enhancing the secret of the exchanged row information, a new polynomial exchange method is used in this phase:

- Node A sends L_{ri} to node B.
- Node B obtains K_{ij} by multiplying U_{cj} with L_{ri} received from node A, $L_{ri} \times U_{cj} = K_{ij}$.
- B sends L_{rj} , K_{ij} and its id S_B to A.
- A obtains K_{ji} by multiplying U_{ci} with L_{rj} received from B, $L_{rj} \times U_{ci} = K_{ji}$.
- A sends $K_{ji}(S_B)$ to S_B which is the encrypted id of S_B by K_{ji} .
- B uses K_{ij} to decrypt $K_{ji}(S_B)$, and it could be sure that the calculated polynomial of A matches with the calculated polynomial of B. And then B uses K_{ij} to generate MAC (Message Authentication Code) and sends K_{ij} (CLR), MAC (K_{ij} , S_B ||CLR) to A. This CLR message is the confirmation that B agrees with A for the locally computed polynomial. RC5 [11] could be used to calculate the MAC using K_{ij} . RC5 is a symmetric block cipher designed to be suitable for both software and hardware implementation. It is a parameterized algorithm, with a variable block size, a variable number of rounds and a variable length of key. Now the shared secret polynomial $K_{ij} = K_{ji}$ is secure. The secure exchange framework is shown in Fig.1.

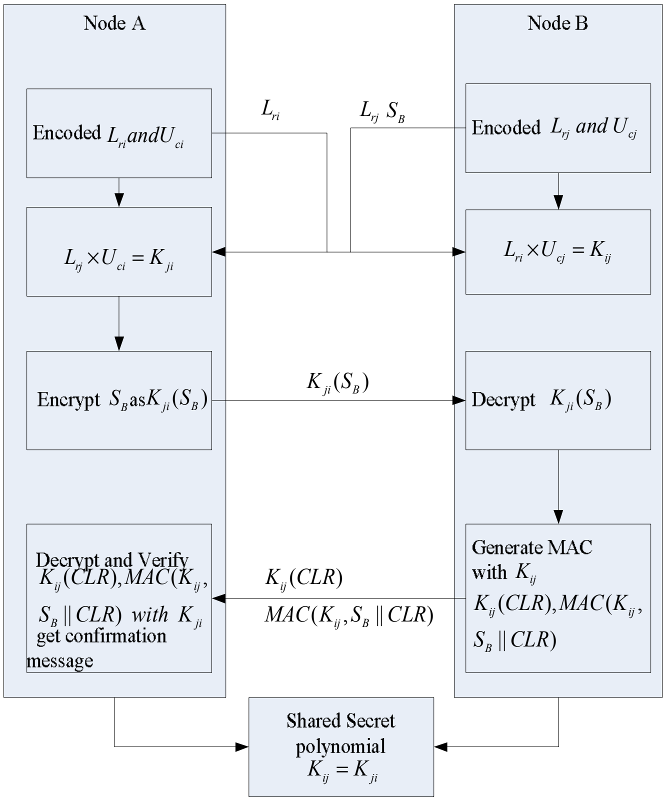
Node A
Node B
Encoded $L_{ri} and U_{ci}$
L_{ri}
L_{rj} S_B
Encoded L_{rj} and U_{cj}
$L_{rj} \times U_{ci} = K_{ji}$
$L_{ri} \times U_{cj} = K_{ij}$
Encrypt S_B as $K_{ji}(S_B)$
$K_{ji}(S_B)$
Decrypt $K_{ji}(S_B)$
Decrypt and Verify $K_{ij}(CLR), MAC(K_{ij},$ $S_B \| CLR)$ $with$ K_{ji} get confirmation message
$K_{ij}(CLR)$
$MAC(K_{ij}, S_B \| CLR)$
Generate MAC with K_{ij} $K_{ij}(CLR), MAC(K_{ij},$ $S_B \| CLR)$
Shared Secret polynomial $K_{ij} = K_{ji}$

Fig.1 Secure exchange framework to establish the shared secret polynomial

V. Performance Analysis and Comparison

In this section, we evaluate the key performance's indices of our scheme, and compare the scheme with E-G schemes. We focus on analyzing network connectivity, resilience against node capture and storage analysis.

A. Analysis of Network Connectivity

Similar to the analysis in E-G scheme, network connectivity P in the proposed scheme is the probability of sharing at least one key between any two sensor nodes. In the proposed scheme, network connectivity P is approximated as (2).

$$P = 1 - \Pr[\text{a pair of nodes do not share any one key}]$$

$$= 1 - \frac{(1-\frac{k}{S})^{2S-2k+1}}{(1-\frac{2k}{S})^{S-2k+\frac{1}{2}}} \tag{2}$$

In Section 4 we discuss that any two sensor nodes can always find a shared key (a common polynomial) between themselves using LU matrix decomposition. In other words, the probability of no sharing a common key between any two sensor nodes is zero.

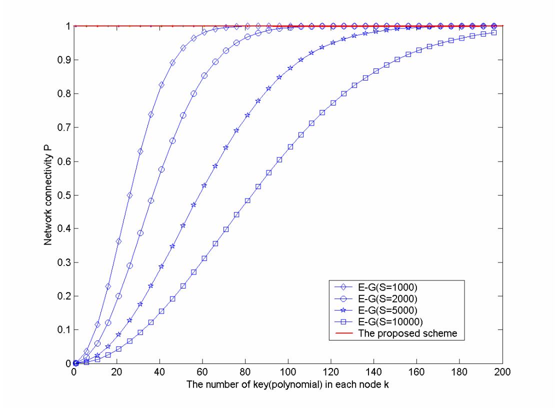

Figure 2.Network connectivity comparison between the proposed scheme and E-G scheme

Fig.2 compares network connectivity P of the proposed scheme with that of E-G scheme. In the simulation, we assume that the capacitance of each node is 200 and the size of key pool is 1000, 2000, 5000 and 10000. The result illustrates that the proposed scheme has 100% connectivity regardless of the number of keys per node. In addition, keys in our scheme occupy less memory space of sensor nodes.

B. *Analysis of Resilience Against Node Capture*

In wireless sensor networks an adversary can easily realize information of compromised nodes, intentionally provide misleading information to the entire network, and threaten the whole network's security. In this section we evaluate how the proposed scheme improves WSN's resilience by calculating the fraction of compromised communication (shared keys) among non-compromised nodes. In addition, we plan to compare our scheme with E-G scheme in this performance.

In E-G scheme, the probability of compromising the shared keys between any two non-compromised nodes is following (3):

$$P_{compromised} = 1-(1-\frac{k}{S})^{x} \qquad (3)$$

In the proposed scheme, polynomials which are predistributed to each node are randomly selected from the polynomial pool. When x nodes have been compromised, the probability of compromising the shared keys between any two non-compromised nodes is equal to the probability of compromising the shared polynomials between any two non-compromised nodes. At the same time the probability can be also regarded as the proportion of the entire network insecurity.

Suppose that one shared key between any two non-compromised nodes is K, the polynomials in polynomial pool are $P_{1,}P_{2,}\cdots,P_{\omega}$. Let C_x be the event that x nodes are compromised, A_i be the event that the shared key K is calculated by P_i which has been disclosed. In condition of x nodes have been compromised, the probability of compromising the shared key K is:

$$P(C_{compromised} \mid C_x) = P(A_1 \cup A_2 \cup \cdots \cup A_\omega \mid C_x) \qquad (4)$$

Due to reason that $P_{1,}P_{2,}\cdots,P_{\omega}$ are exclusive events, so

$$P(A_1 \cup A_2 \cup \cdots \cup A_\omega \mid C_x) = \sum_{i=1}^{\omega} P(A_1 | C_x) = \omega \cdot P(A_1 \mid C_x) \qquad (5)$$

$$P(A_1 \mid C_x) = \frac{P((K \in P_1) \cap (P_{1compromised}) \cap C_x)}{P(C_x)} \qquad (6)$$

The shared key K between any two non-compromised nodes may be calculated by any one polynomial, so

$$P(K \in P_1) = \frac{1}{\omega} \qquad (7)$$

According to the (4)(5)(6)(7), Total Probability Theorem and Bernoulli Probability, hence：

$$P(C_{compromised} \mid C_x) = P(P_{1compromised}) = \sum_{i=t+1}^{x} \binom{x}{i} \left(\frac{\tau}{\omega}\right)^{i} \left(1-\frac{\tau}{\omega}\right)^{x-i} \qquad (8)$$

Equation (8) can be used as computing the fraction of compromised communication (shared keys) among non-compromised nodes. In (8), τ is decided by key capacity k of each node and the security threshold t . τ is far less than m in order that network obtains a bigger security threshold. The related parameters of analysis and simulations can be calculated in the Table II.

TABLE II. LIST OF PARAMETERS

<table>
<tr><th colspan="4">E-G scheme</th></tr>
<tr><td colspan="2">k</td><td colspan="2">S</td></tr>
<tr><td colspan="2">100</td><td colspan="2">100000</td></tr>
<tr><td colspan="2">200</td><td colspan="2">100000</td></tr>
<tr><th colspan="4">The proposed scheme</th></tr>
<tr><td>τ</td><td>ω</td><td>t</td><td>k</td></tr>
<tr><td>2</td><td>1000</td><td>99</td><td>100</td></tr>
<tr><td>2</td><td>500</td><td>199</td><td>200</td></tr>
</table>

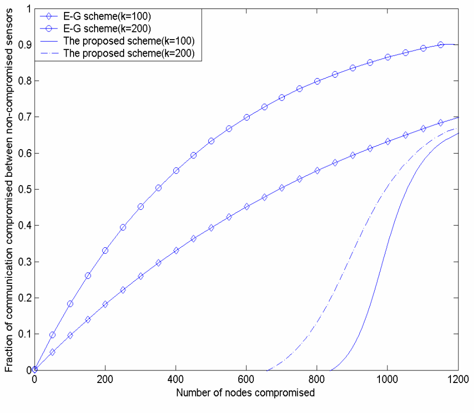

Figure 3. Network resilience: comparison the proposed scheme with E-G scheme

Fig.3 shows the relationship between the fraction of compromised communication for non-compromised sensor nodes and the number of compromised nodes under different parameters in Table II. The result shows that the less the key capacity k of each node, the lower the probability of disclosed communication for uncompromised sensor nodes. On the other hand, any node capture in E-G scheme would leak the entire network, whereas the communication links of our scheme cannot be disclosed as long as the number of compromised nodes is smaller than the threshold security property. Therefore the proposed scheme has observably improved the resilience against attacks.

C. Memory Usage Analysis

In this subsection, as any two sensor nodes establish the shared key by using polynomial-based key predistribution, the network -wide memory usage of our scheme is mostly polynomials' cost. We propose an efficient method to store the row and column information of L and U matrices. Analyzed in Section 4, our scheme only needs to store each element in the non-zero-element part and one value specifying the number of following zeros in zero-element part of L and U matrices. This technique is specially suitable for large wireless senor networks. Some notations will be used to estimate the storage efficiency:

- m ---the number of bits for each polynomial information in L or U matrix
- N ---the maximum number of sensor nodes that are to be deployed in the network
- h_i ---the total number of nonzero elements in a row of L and in a column of U stored in sensor node with id i
- z ---the number of bits needed so that the largest number of zero elements in zero-element part in a row of L or in a column of U could be represented

Now the memory usage to store polynomial information in each sensor node is

$$\lambda_i = (h_i \times m + 2 \times z) \tag{9}$$

The network-wide memory usage is computed by (10) .

$$\begin{aligned}\Gamma_{total} &= \sum_{i=1}^{N} \lambda_i = m \times \sum_{i=1}^{N} h_i + N \times (2 \times z) \\ &= m \times \frac{N \times (N+1)}{2} + N \times (2 \times z) \\ &= m \times \frac{N \times (N+1)}{2} + N \times (2 \times ceil(\log_2(N-1)))\end{aligned} \tag{10}$$

Where $z = ceil(\log_2(N-1))$ as the largest number of zeros in a row or a column could be represented by the $ceil$ (rounded up) value of $\log_2(N-1)$.

In our scheme, the major memory saving is done by encoding the zeros in the zero-element parts of the L and U matrices. Hence, the memory usage memory saving could be computed by (11).

$$\begin{aligned}\Gamma_{saving} &= 2 \times m \times \frac{N \times (N-1)}{2} - N \times (2 \times z) \\ &= m \times N \times (N-1) - N \times (2 \times ceil(\log_2(N-1)))\end{aligned} \tag{11}$$

In E-G scheme, to maintain the certain network connectivity, which is the probability that two neighboring sensor nodes can establish a direct shared key, the number of keys can not be too small. However, large number of keys means the adversary can obtain more secrets each time he compromises one more node. The contradictive memory requirements make it difficult to optimize both of security and network connectivity given fixed memory resource. A merit of our scheme is that the memory usage is unrelated with network connectivity, and any two sensor nodes always find a shared key between them.

VI. CONCLUSION

Security mechanism of wireless sensor networks which is varied from wireless ad hoc networks has its unique characteristic. In recent years, key distribution [8] has been one of the hot issues in security research. In this paper, we have proposed a new key predistribution scheme to make sure that any two sensor nodes can find one shared key using the polynomial-based key predistribution and LU matrix decomposition. Our scheme significantly increases network connectivity, security and the network-wide memory. In the future, we will plan to develop the group-based matrix decomposition for the large distributed wireless sensor networks. Also we will investigate how the computational costs could be reduced to increase the efficiency of our scheme.

REFERENCE

[1] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E, "A survey on sensor networks", IEEE Communications Magazine, Vol.40(8), pp.102-114, 2002.

[2] C.Blundo, A.DeSantis, A.Herzberg, etal, "Perfectly-secure key distribution for dynamic conference", Advances in Cryptolopy-CRYPTO'92, LNCS 740, pp.471-486, 1993.

[3] Donggang Liu, Peng Ning, "Establishing Pairwise Keys in Distributed Sensor Networks", in Proceeding of the 10th ACM Conference on Compute and Communications Security, New York, USA, pp.51-61, 2003.

[4] Horn, Roger A. Johnson, Charles R., Matrix analysis, Cambridge University Press, 1985.

[5] L.Eschenauer,V.D.Gligor,"A Key-management Scheme For Distributed Sensor Networks", in Proceeding of the 9th ACM Conference on Computer and Communications Security, New York, USA, pp.41-47, 2002.

[6] J.Spencer, "The Strange Logic of Random Graphs, Algorithms and Combinatorics 22", Springer-Verlag 2000, ISBN 3-540-41654-4.

[7] Haowen Chan, Adrian Perrig ,Dawn Song, "Random Key Predistribution Schemes for Sensor Networks", in Proceeding of 2003 IEEE Symposium on Research in Security and Privacy, pp.197-213,2003.

[8] Seyita Camtepe, Bulent Yener, "Key Distribution Mechanisms for Wireless Sensor Networks : A Survey", Rensselaer Polytechnic Institute TR-05-07, March 2005.