Mathematical Problems of Computer Science 33, 127 134, 2010.

Data Encryption, Authentication and Key Predistribution Schemes for Wireless Sensor Networks

Ara Alexanian, Hakob Aslanyan and Armen Soghoyan

Yerevan State University

Abstract

This paper investigates some security aspects of wireless sensor networks. Particularly symmetric encryption, authentication and key predistribution schemes are proposed. Encryption and authentication schemes are based on a secret key which is a table of strong generators of a permutation group (as per Sims algorithm). The ciphertext of a plaintext of size N is a single permutation over $n = 2N + 1$ elements. Encryption/Decryption processes are simple and easy to perform for a wireless node with limited abilities.

Key predistribution scheme deals with all the node and network limitations and requirements existing in wireless sensor networks and gives a secure key predistribution scheme under the assumption that nodes are not physically captured for t_0 time after being deployed, where t_0 is time necessary for invoking the key establishment phase of proposed scheme, which is in practice should be some seconds.

1 Introduction

Wireless sensor networks (WSN) are composed of a large number of sensor nodes with sensing, processing and wireless communicating capabilities. Nodes are usually battery powered and spatially distributed on a given region without knowing the network topology. Nodes are doing the environmental monitoring by measuring variety of parameters such as temperature, vibration, radiation, chemical elements, etc. Data gathered by nodes are delivered to some base station in a multi-hop way [10]. WSN provides a real time environmental monitoring tool which could be used in variety of applications such as for monitoring hostile or disaster areas, toxic regions, etc. In most of those applications information authenticity and secrecy are important. Cryptographic tools being used in traditional networks usually do not suitable for WSNs because of nodes' limited battery, storage, processing and communicating abilities [19, 20]. Those limitations come from nodes' small sizes and low costs. Therefore development of new cryptographic tools respect of above mentioned limitations are of interest.

The survey of block ciphers suitable for WSNs based on existing literature is presented in [1]. Particularly it presents the comparison of RC5 [2], RC6 [3], Rijndael [4], MISTY1 [5], KASUMI [6] and Camellia [7]. Some other encryption schemes developed for wireless sensor networks are SPINS [8], TinySec [9], etc.

We propose a nice symmetric block encryption method based a on secret key which is a table of strong generators of a permutation group (as per Sims algorithm). The input plaintext is divided into the blocks, each of size N, and each block is encrypted separately into a single permutation over $n = 2N + 1$ elements.

The input string for decryption process is a sequence of permutations over n elements and the key is the above mentioned table. We treat each permutation separately and recover a block of size N from each permutation. The fact that the output of encryption process is a sequence of permutations leads to a very statistically uniform string (standard compression software, such as zip or rar, fails to compress the output generated by the proposed scheme).

The above scheme also gives a nice node authentication method. Details of which will be discussed in later sections.

Paper also considers the key predistribution in wireless sensor networks. Again, due to limited power and processing abilities WSN nodes fail to perform public key operations therefore they should be equipped with some secret information which allows them to establish symmetric communication keys with the neighboring nodes after deployment. The fact that nodes do not know their positions in network (the set of neighboring nodes) before being deployed and the limited memory makes the problem even harder. A key predistribution protocol must not only enable a newly deployed sensor network to initiate a secure infrastructure, but it must also allow the nodes deployed at a later time to join the network securely. More detailed description of key predistribution and existing works will be given in Section 3. Proposed scheme deals with all the node and network limitations and requirements existing in wireless sensor networks and gives a secure key predistribution scheme under the assumption that nodes are safe for t_0 time after being deployed (safe here means that nodes are not being captured physically). t_0 Is the time necessary for invoking key establishment part of proposed key predistribution protocol after node deployment, which is in practice should be some seconds.

2 Symmetric Encryption/Decryption Scheme

The encryption scheme is based on a secret key, which is a table of strong generators of a permutation group as per Sims algorithm. We present some definitions and the brief description of Sims algorithm in next section.

2.1 Generating Set of a Group And the Sims Algorithm

Definition 1. *Subset S of a group G is a generating set of G, if every element $a \in G$ can be expressed as the combination (under the group operation) of elements of the subset S and their inverses; $a = x_1^{\epsilon_1} x_2^{\epsilon_2} \dots x_k^{\epsilon_k}$, $\epsilon_i \in \{1, -1\}$, $x_i \in S$, $i = 1, 2, \dots, k$.*

According to above definition, it is possible to build all the elements of a group by doing all the possible multiplications of elements of subset S. Of course it is worth to do in case of finite groups. Algorithmically the presentation of a group as a generating set is not perfect as it is not simple to build all the possible multiplications of elements of generating set S, even if it is a finite set. Also expression of an element of a group is not unique and it is not possible to count the order of a group by having the set of generators.

The another variant of generating set which does not have the above faults is the *strong generating set* of a group, which plays an important role in our scheme. Sims algorithm gives us a way to build such a *strong generating set* of a group by having its generating set.

According to Kelly's theorem considering the group of permutations only is enough.

2.1.1 Sims Algorithm.

The input of Sims algorithm is a set of permutations S over n elements, which is a generating set (Definition 1) for some subgroup $G \leq S_n$. It stars a work with an empty $n \times n$ table (Table 1), where cells below the diagonal are not used (when we refer to cells we mean the cells over the diagonal), all the diagonal cells contain the trivial permutation, which maps any i into i, $i = 1, \ldots, n$ and the rest of the cells are empty. During the work Sims algorithm repeatedly performs an operation called *cascade*. *cascade* is applied on some permutation a when table is partially filled in (some cells may have already contain a permutation). It either fills an empty cell of a table by a correct permutation, i.e. the permutation in the cell with row number i and column number j maps l into l, $l = 1, \ldots, i-1$ and i into j, or goes throw all the rows of a table and gets the representation of a by the elements of the table, one from each row. Algorithm has two stages. During the first stage *cascade* is being applied to all the permutations from S. As a result some of the cells will be assigned a 'correct' permutation. In the second stage *cascade* is applied to all the pairs of permutations which have been filled in a table during the first stage. After the second stage algorithm finishes the work and the table contains the ***strong generating set*** of G. Each element of G is presentable as a multiplication of elements of resulting table by taking one permutation from each row starting from first one. The representation of any element a of G could be achieved by ap

	1	2	3	...	n
1	e				
2	■	e			
3	■	■	e		
⋮	■	■	■	⋱	
n	■	■	■	■	e

Table 1.The empty table Sims algorithm starts with.

Definition 2. *The generating set given by the output table of Sims algorithm is a strong generating set.*

[1]The facts from group theory presented in this section could be found in different books of algebra [18, 21, 22] etc.

2.2 Encryption

The input information is a binary string divided into consecutive blocks consisting of N bytes each. Each block is being encrypted separately. The encryption scheme is based on a secret key, which is a table of strong generators of a permutation group (as per Sims algorithm). Each permutation is a permutation over $n = 2N + 1$ elements. Also we do not use the last row of a table, it contains trivial permutation only.

Encryption of a block is done as follows. We read the first byte in the block and treat it as an unsigned integer, i.e. an integer x_1 such that $0 \leq x_1 \leq 255$. We compute $s_1 = x_1 \text{mod}(n-1)+2$, where $x_1 \text{mod}(n-1)$ stands for the remainder when x_1 is divided by $n-1$. We have that $2 \leq s_1 \leq n$ and we choose the permutation that occupies the $s_1 - th$ cell in the first row of the table. Then we compute $s_2 = x_1 \text{mod}(n-2)+3$ and choose the permutation that occupies the $s_2 - th$ cell in the second row of the table. Thus we fixed two permutations from the first two rows of the table using the first byte of the input block. We continue this way choosing two permutations for each byte of the block, i.e. for the k^{th} byte x_k we compute $s_{2k-1} = x_k \text{mod}(n-2k+1)+2k$ to choose a permutation in the $(2k-1)^{th}$ row and $s_{2k} = x_k \text{mod}(n-2k)+2k+1$ to choose a permutation in the $2k^{th}$ row. As a result we fix $2N = n-1$ permutations from all rows of the table. We multiply all the fixed permutations starting from the first row (as in Sims algorithm) and obtain a single permutation, which is the output code for the block. Thus the output code for the entire input string is a sequence of permutations over n elements.

2.3 Decryption

The input string is a sequence of permutations over n elements and the secret key is the table from above. We take the permutations one by one and perform the following action to recover the original bytes. One block of N bytes is being recovered from one permutation.

We take the first permutation and apply *cascade* step of Sims algorithm to obtain the representation of the permutation by strong generators. Thus, we get the cell numbers for the permutations from the table, one for each row. If s_1 is the number of the cell in the first row and s_2 is the number of the cell in the second row then we recover the first byte x_1 using the Chinese reminders theorem; x_1 is the solution of the system

$$\{x_1 = (s_1 - 2)\text{mod}(n-1) x_1 = (s_2 - 3)\text{mod}(n-2)$$

As $n-1$ and $n-2$ are relative primes the Chinese theorem gives us the x_1. All solutions of the system differ by an integer, which is a multiple of $(n-1)(n-2)$ and we are guaranteed that there is a solution in $[0, 255]$ as per encryption scheme.

The next byte x_2 is being obtained the same way from the second permutation. This time the system is as follows

$$\{x_2 = (s_3 - 4)\text{mod}(n-3) x_2 = (s_4 - 5)\text{mod}(n-4)$$

system for the k^{th} permutation is

$$\{x_k = (s_{2k-1} - 2k)\text{mod}(n-2k-1) x_k = (s_{2k} - 2k - 1)\text{mod}(n-2k)$$

Thus, one by one we recover initial N bytes.

2.4 Node Authentication

Of course this scheme may be used to encrypt information being exchanged between nodes. But also it gives a nice method of authentication of the nodes. Initially each node is given one table (or a set of tables) that generates a subgroup(s) of the symmetric group. The tables for different nodes may coincide, but also can be different depending on the unique identification code assigned to each node. The given node is supplied by permutations for each other node.

such that they belong to the subgroups of those nodes. If the node A wants to establish a link to the node B it sends the permutation that belongs to the subgroup generated by the table of the node B to B. The node B performs *cascade* to this permutation and verifies that A is a decent node and not a substitute attacking the system. B may ask for more than one permutations from A to prove its decency. The probability of successful attack is very close to zero. After successful verification the nodes may start information exchange either insecure or secure (if we use the above scheme A and B must be supplied by the same table for communication generating all permutations).

2.5 Security Analysis of Scheme

One should choose N in a way to reduce the redundancy as much as possible. On practice it would be better to use only a part of the table, excluding some number of the lower rows of the table, which have very few cells. The fact that the output is a sequence of permutations leads to a very statistically uniform string (standard compression software such as zip or rar fails to compress the output generated by the above scheme). It is also a good idea to compress the input string first and then encrypt. The above method can be modified to be used as a public-key system. One can use a table that do not generate all permutations over n elements, but a subgroup of the symmetric group. There are many variations of this scheme and you may use your imagination.

3 Key Predistribution Scheme

Key predistribution problem in wireless sensor networks is to assign a secret information to each node of a network, before being deployed, such that after deployment node can establish secure communication keys with all of its neighboring nodes securely. Also nodes added at a later time should be able to join the network securely.

3.1 Problem Definition and The Related Work

The following node and network limitations challenge and formalize the key predistribution problem in WSNs.

- The limited computational and power resources of sensor nodes often makes impractical the use of public-key algorithms, such as Diffie-Hellman key agreement [11] or RSA signatures [12].
- Usually nodes are deployed in public areas, which enables the possibility of physical attack. Because of low cost nodes are usually not tamper resistant, which gives a chance to an attacker to read the secret information from captured node. Hence, the captured nodes should not contain information about the rest of the communication links of the network. Also clones of the captured node, made by an attacker, should not be able to join to the network.
- Nodes are usually deployed randomly (e.g. from an airplane), hence the key predistribution scheme should not assume prior knowledge of which nodes will be neighbors in network.
- Nodes added at a later time should be able to join the network securely, therefor the bootstrapping information can not be simply erased from nodes after deployment to prevent compromise in the event of capture.

- The key storage on a node is very limited and usually it is not possible to keep a unique communication key for each of the other node of network.
- The scheme resilience should not depend on the size of the network (the number of nodes). The size of the network may grow during the time, therefor the scheme should support networks with any size.

Random key predistribution scheme is presented in [13]. Before being deployed each node is assigned to a random set of k keys from big key pool S. After deployment nodes having a common key would be able to establish a secure communication key. Key pool size $|S|$ and the key ring size k of a node should be chosen in a way such that after node deployment the networks will be connected with hign probability. [14] Proposes the similar scheme as [13]. The difference is that pair of nodes establishes a communication key if they share more than q keys. [15] Proposes the combinatorial scheme to choose the key ring in a way such that each pair of nodes shares exactly one key. All those scheme are vulnerable to node capture. Captured node contains k different keys of key pool and enables the node replication attack. However [16, 17] are some of the existing surveys of key predistribution schemes proposed for wireless sensor networks.

The key predistribution scheme presented in next section deals with all of above mentioned node and network limitations and security requirements. Under the only assumption that nodes are safe for t_0 time after being deployed where t_0 is the time necessary for invoking key establishment after node deployment, which should be some seconds in practice.

3.2 The Scheme

Let E is the symmetric encryption scheme used by network, and each node is equipped with encryption/decryption algorithms of scheme E. We denote by $E(k, m)$ the ciphertext of plaintext m encrypted with scheme E and key k. And $E^{-1}(k, c)$ stands for decryption.

Key predistribution stage. In a bootstrapping stage each node i is being assigned to a unique key K_i, the master key of network K and the unique key encrypted by the master key $E(K, K_i)$.

Communication key establishment stage. After being deployed each node i broadcasts a network joining request and as a response receives $E(K, K_{j_1}), \ldots, E(K, K_{j_s})$ from its neighboring nodes $j_1, \ldots, j_s$. $E(K, K_{j_1}), \ldots, E(K, K_{j_s})$ can be decrypted with master key K hence node i will have a unique keys $K_{j_1}, \ldots, K_{j_s}$ of its neighbors, which can be used to establish a secret communication keys $K_{ij_1}, \ldots, K_{ij_s}$ with them.

After key establishment. After communication key establishment stage nodes erase the following information; master key of the network K, response data of neighbors $E(K, K_{j_1}), \ldots, E(K, K_{j_s})$ and the unique keys $K_{j_1}, \ldots, K_{j_s}$ of neighbors. hence they keep only; own unique key, own unique key encrypted by the master key $E(K, K_i)$ and the established communication keys $K_{ij_1}, \ldots, K_{ij_s}$. Unique key and the encrypted unique key stay for supporting the later node deployment.

3.3 Security Analysis of Scheme

Let communication key establishment stage last t_0 time, which in fact depends on different parameters. such as wireless communication speed of nodes, number of neighboring nodes, complexity of networks' encryption scheme E, etc. Anyway it should be some seconds in real network. In a presented scheme after t_0 time nodes do not contain any secret information which can be used by an attacker (in case of node capture) to obtain the communication/unique keys of nodes other than the captured one.

One can approximate the value of t_0 and fix some $t_0' > t_0$ after which all the deployed nodes will erase the above mentioned data weather they established communication keys with neighbors or not. This will be needed for nodes which are accidentally dropped so far from network that have no neighboring nodes to perform a key establishment. Nodes added at a later time has the same bootstrapping information, and can join the network by the same way.

References

[1] Yee Wei Law, Jeroen Doumen and Pieter Hartel, "Survey and benchmark of block ciphers for wireless sensor networks", *ACM Transactions on Sensor Networks TOSN*, vol. 2, number 1, pp. 65-93, 2006.

[2] R. Rivest, "The RC5 Encryption Algorithm", 1994 Leuven Workshop on Fast Software Encryption, Springer-Verlag, pp. 86-96, 1995.

[3] R. Rivest, M. Robshaw, R. Sidney and Y. Yin, "The RC6^{TM} Block Cipher. Specification version 1.1", 1998.

[4] J. Daemen and V. Rijmen, AES Proposal: Rijndael, 1999.

[5] M. Matsui, "New Block Encryption Algorithm MISTY", *Fast Software Encryption, 4th International Workshop, FSE 97*, vol. 1267 of LNCS.,Springer-Verlag, pp. 54-68, 1997.

[6] 3rd Generation Partnership Project, "Specification of the 3GPP Confidentiality and Integrity Algorithms Document 2", KASUMI Specification. ETSI/SAGE, Specification Version: 1.0, 1999.

[7] K. Aoki, T. Ichikawa, M. Matsui, S. Moriai, J. Nakajima and T. Tokita, "Specification of Camellia. A 128-Bit Block Cipher", Nippon Telegraph and Telephone Corporation and Mitsubishi Electric Corporation, Specification Version 2.0, 2001.

[8] A. Perrig, R. Szewczyk , V. Wen and et al., "SPINS: security protocols for sensor networks", *Journal of Wireless Networks*, vol. 8, number 2, pp. 521-534, 2002.

[9] C. Karlof ,N. Sastry and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks", *Proceedings of the 2nd International Conference & Embedded Networked Sensor Systems*, Baltimore, USA, 2004.

[10] J.N. Al-Karaki and A.E. Kamal, "Routing techniques in wireless sensor networks: A survey", *IEEE Wireless Communications.*, vol. 11, number 6, pp. 6-28, 2004.

[11] W. Diffie and M. E. Hellman. "New Directions in Cryptography", *IEEE Transactions on Information Theory*, vol. IT-22, number 6, pp. 644-654, 1976..

[12] R. L. Rivest, A. Shamir and L. M. Adleman. "A method for obtaining digital signatures and public-key cryptosystems". *Communications of the ACM*, vol. 21, number 2, pages 120-126, 1978.

[13] L. Eschenauer and V. D. Gligor. "A key management scheme for distributed sensor networks", *Proceedings of the 9th ACM Conference on Computer and Communication Security*, pages 41-47, 2002.

[14] H. Chan, A. Perrig and D. Song. "Random key predistribution schemes for sensor networks", *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pp. 197-213, 2003.

[15] S. A. Camtepe and B. Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks, *IEEE/ACM Transactions on Networking (TON)*, vol. 15, number 2, pp. 346-358, 2007.

[16] Y. Xiao, V. Krishna Rayi, Bo Sun, X. Du, Fei Hu and M. Galloway, "A survey of key management schemes in wireless sensor networks", *Computer Communications*, vol. 30, number 11-12, pp. 2314-2341, 2007.

[17] H. Chan, A. Perrig, and D. Song, "Key Distribution Techniques for Sensor Networks', *Wireless Sensor Networks*, T. Znati et al., eds, 2004.

[18] A. Alexanyan, Algebra (Groups, Rings, Fields), Yerevan University Publisher, In Armenian, 2006,.

[19] M. Brown, D. Cheung, D. Hankerson, J. L. Hernandez, M. Kirkup and A. Menezes, "PGP in constrained wireless devices", *9th USENIX Security Symposium*, 2000.

[20] D. W. Carman, P.S. Kruus and B. J. Matt., "Constraints and approaches for distributed sensor networks security", NAI Lab, September. number 00-010, 2000.

[21] Serege Lang, Algebra, Springer Science+Business Media, Inc., 2002.

[22] B.L Van der Varden, Algebra, Springer Verlag, 1968.

Անլար սենսորային ցանցերի տվյալների ծածկագրման, ինքնության հաստատման և բանալիների բաշխման սխեմաներ համար

Ա. Ալեքսանյան, Հ. Ասլանյան և Ա. Սողոյան

Ամփոփում

Աշխատանքում դիտարկված են անլար սենսորային ցանցերի ապահովությանը վերաբերող որոշ խնդիրներ: Մասնավորապես առաջարկվում են տվյալների սիմետրիկ ծածկագրման, ինքնության հաստատման և բանալիների բաժանման սխեմաներ: Ծածկագրման և ինքնության հաստատման սխեմաների բանալի է հանդիսանում տեղադրությունների խմբի ուժեղ ծնիչների բազմության աղյուսակը (ինչպես Սիմսի ալգորիթմում): Մուտքային տվյալների N չափանի (N բայթ պարունակող) բլոկը ծածկագրելուց ստացվում է $n = 2N + 1$ տարրերից կազմված մեկ Տեղադրություն: Ծածկագրման և վերծանման գործողությունները պարզ են և կարող են կատարվել սահմանափակ հնարավորություններ ունեցող անլար սենսորային հանգույցի կողմից: Բանալիների բաժանման սխեման հաշվի է առնում անլար սենսորային ցանցերում առկա բոլոր սահմանափակումներն ու պահանջները և տալիս է բանալիներ բաժանման ապահով սխեմա անելով միայն մեկ ենթադրություն, այն է՝ հանգույցները տեղադրվելուց հետո t_0 ժամանակի ընթացքում չեն ենթարկվում ֆիզիկական հարձակման: Այստեղ t_0 -ն ցանցի հանգույցների միջև հաղորդակցման բանալիների հաստատման համար անհրաժեշտ ժամանակն է, որը իրականում տևում է վայրկյաններ: