# Strategies Averting Sybil-type Attacks Based on the Blom-scheme in Ad Hoc Sensor Networks

Shiuh-Jeng Wang[1,*], Yuh-Ren Tsai[2] and Chung-Wei Chen [3]

[1]Department of Information Management,
Central Police University Taoyuan, Taiwan 333
[*]Correspondence: Email: sjwang@mail.cpu.edu.tw

[2,3] Institute of Communications Engineering
National Tsing Hua University
101, Sec. 2, Kuang-Fu Rd.,
Hsinchu 300, Taiwan

***Abstract*—We present a scheme based on the Blom scheme for resisting the Sybil type attacks. In this scheme, the authority pre-distributes the combination of the secret used in the Blom scheme. The attacker needs to compromise the specified number of nodes to compromise the whole systems. Furthermore, our scheme which provides a stronger relationship between ID and key would make authentication more reliable and resistant against the Sybil attacks**.

***Index Terms* - Ad hoc networks, Authentication, Sybil attack**

## I. INTRODUCTION

An ad hoc network is a network that is constructed using more than two radio sets that use the same wireless communication technology and can handle network procedures on their own. It is one type of distributed network system. One of its applications is the sensor network. The sensor network is usually used in battlefield situations to measure environmental conditions and gather reconnaissance information, such as temperature, and information about the movement of enemy troops [10]. Each radio set can construct a communication link with other radio sets in its radio communication range. For a radio set that is out of its radio communications range, it can use the radio sets that are within its radio communications range as relay stations. All the packets transferred to the radio set out of communications range hop through [10] [16] [17] [18] [20], the radio set within communications range. For simplicity, the radio set will henceforth be referred to as the "node" in the following section.

Because of the nature of the distributed network system, the ad hoc network is easily attacked. Therefore, we propose a scheme that resists attacks on the ad hoc network. Our scheme is based on the Blom scheme and the determinism property of the one-way hash function. The system pre-distributed the secret into each node individually. And nodes IDs will imply to these secret, which can be verified by other nodes. With the verification of IDs for each user, the Sybil attack can be detected and prevented in ad hoc networks. Although Du et al. [6] also provides a similar scheme, it can't efficiently resist attack, like the Sybil attack.

The rest of this paper is organized as follows: The security problems that are present in an ad hoc network are described in further detail in Sec. II. Then, we present our scheme consisting of two models parts (a) and (b) in Sec. III. After that, an analysis of our scheme is discussed in Sec. IV. Finally, a conclusion is given in Sec. V.

## II. SECURITY OF AD HOC NETWORKS

An ad hoc network is weak against an intruder's attack because it doesn't have a centralized authority to authenticate the packet validity and the *ID* of each member. Furthermore, the security protocol from the original centralized network doesn't work effectively on the ad hoc network due to the mobility of each node within the ad hoc network. In addition, the constraint of having only limited power in each node is another strategic failing of the ad hoc network. In this section, we will introduce the security problems inherent to this type of network and the solutions proposed by other scholars.

### *A. The Security Problems of an Ad Hoc Network*

An adversary's attacks against ad hoc networks can be classified into two main types, namely---the passive attack and the active attack. A passive attack means that the adversary attacks in such a way so as not to damage network operation. In contrast with a passive attack, a active attack *will* damage network operation.

Passive attacks enable the intruder to eavesdrop on messages as they propagate through a wireless environment, constituting a passive attack.

Active attacks impede the normal functioning of the internet. After capturing a node in an ad hoc network, the intruder can send legitimate messages, including routing messages, to every node in the ad hoc network by using

© 2008 ACADEMY PUBLISHER

the *ID* or the authentication information stored in the captured node. Active attack strategies can take one of the following different forms:

*i. Black Hole:*

When a node in the ad hoc network wants to find a route to a certain destination, it broadcasts the request packet, RREQ. Upon receiving the RREQ packet, the compromised node may send a fake RREP message back to tell the source node that the compromised node has the shortest route to the destination [4] [14]. Then the source node will send the data, hopping through the compromised node as a result of the fake RREP, and the compromised node will drop the data after receiving the data from the source node.

*ii. Denial of Service* (*DoS*)*:*

The DoS attack is when the networks can't provide normal service to the user because of some kind of attack.

*iii. RT Overflow:*

The compromised node sends a false routing message to make the safe node's routing table overflow. Then the safe node can't set up a link with the other normal nodes.

*iv. Impersonation:*

In this attack, the compromised node will use the fake *ID* to deceive the other normal nodes [13].

*v. Energy Consummation:*

When a packet hops through a node, it also expends the energy of that node. Therefore, if the compromised node sends a substantial number of RREQ's or other routing messages in the network all of the time, the life of that ad hoc network would become shorter than that of a normal ad hoc network because of the limited energy of every node.

*vi The Sybil Attack:*

A Sybil attack is a case that the compromised node claims more than one *ID* in the network [5] [11] [15]. The Sybil attack can reduce the effectiveness of fault-tolerant schemes such as distributed storage [3], dispersity [1] and multi-path routing [9]. This attack is not to be confused with the impersonation attack, which deals with identification fabrication only.

### *B. The Solutions Proposed at the Present Day*

There are many schemes which have been proposed for handling the authentication without the centralized authority. All these methods utilize three main strategies in order to handle the ad hoc network security problem---"traffic monitoring of the surrounding nodes", "cryptography method", and "multi-path".

When "monitoring the traffic", every node in the network monitors the traffic situation of the surrounding nodes [1] [3] [7]. If a node is considered a misbehaving node by all the members, the other members would then discard all the claims and disregard all behavior from that node [2] [14].

For the "multi-path" strategy, the node would find more than one way to the destination by using the routing algorithm. And the node would use multi-path to authenticate the destination in order to prevent the path composed by the intruder from being used.

The cryptography method is composed of three parts: "the trusted server", "the public key system", and "the key pre-distribution system". For "the trusted server", the system finds a node to be the authority. The node is then allowed to handle the key management of the whole network. For "the public key system", every node uses its public key system to encrypt data. For "the key pre-distribution system" the system stores the key in the node's memory, before the node is deployed in the ad hoc network,

The Blom scheme is a linear cryptography system [19]. It is also categorized under the key pre-distribution system. It uses a property of the linear equation to generate the same property from the public key without the complicated exponential computation. According to linear algebra theory, the intruder would be able to break the system security only if he/she captured any $\lambda$ nodes and compromised their secret, which consist of the columns that are pre-distributed into the nodes, where $\lambda$ is the rank of the secret matrix in the Blom scheme. Therefore, we define the security level of the Blom scheme as $\lambda$. Our proposed scheme is inspired by the Blom scheme, which we describe further in Section 3.

## III. OUR SCHEME

Our scheme has two parts, part (a) and part (b). Part (a) is based on the notion of single key space. Part (b) is based on the notion of multi-key space. After the presentation of our scheme, the soundness of our scheme will be proven. In addition, we further extend the methods of these two parts in order to retain more distinct secrets to assign to a node in ad hoc network systems. The assumptions with our system are shown as follows:

*Assumptions:*

$n$: the initial node number in an ad hoc network system
$ID_i$: the unique identity of a node $i$
$D_{\lambda x\lambda}$: a symmetric matrix, which is kept secret in this system
$G_{\lambda xn}$: a public matrix of $\lambda xn$, which is made up of $n$ independent column vectors, i.e. $G=\{g_1,g_2,g_3,...,g_n\}$ for each column vector $g_i$, $i=1, 2, ...,n$.
$A_{\lambda xn}$: a pre-distributed matrix, where $A=D_{\lambda x\lambda}\cdot G_{\lambda xn}$.and $A$ has $n$ column vectors, i.e. $A=\{c_1,c_2,c_3,...,c_n\}$ and column vector $c_i$, $i=1, 2, ...,n$.
$f$: a one way function
$PRG$ : a pseudo-random generator
$r$: a column number chosen to be the column combination pre-distributed into each node

### *A. Part (a): Single Key Space*

Part (a) is divided into two phases: the ***Key Pre-distribution Phase*** and the ***Common-Key-Set-Up Phase***. Initially, we assign each sensor node a unique *ID*. Subsequently, we define each column vector of matrix $A$ in the Blom scheme as $\{c_1, c_2, c_3, c_4, ..., c_n\}$. In ***Key Pre-distribution Phase***, the nodes have not yet entered the network or are not yet deployed in the network. During ***Key Pre-distribution Phase***, the authority would offer a one-way function $f$, and a pseudorandom generator,

© 2008 ACADEMY PUBLISHER

*PRG*. Then, the authority would load *f* and *PRG* into each node. After completing the above process, the authority would compute the result, *f*(*ID*), for each node individually. Then, that result is used as the initial state for the *PRG*. According to the output of the *PRG*, the authority then chooses *r* columns from set *C* and adds them together to form a totally new vector. Finally, the authority pre-distributes that new vector into the corresponding node.

After the nodes are deployed, if any two nodes want to set up a safe link, they will execute a ***Common-Key-Set-Up Phase***. During this phase, the two nodes will set up a secure path with each other. In order to do this, first, we define the columns of the public matrix *G* in the Blom scheme as, $g_1, g_2, g_3, g_4, g_5, \ldots, g_n$. In ***Common-Key-Set-Up Phase***, the two nodes deliver theirs *ID*s to each other. Then the nodes compute *f*(*ID*) according to the receiving ID and use the result of *f*(*ID*) as the initial state for the *PRG*, which is the same process used by the authority. From the output, the node will know the indices of the columns that make up the secret column in the other node. Then the node would compute the sum of the corresponding column in *G* to obtain a new vector. Subsequently, the node performs vector multiplication on the new vector and the vector that the authority pre-distributed into itself, then uses the result as the symmetric key to encrypt the data to be transferred. The other node would obtain the same key by executing the same process. The principle of our scheme within a single key space that produces the same key after ***Common-Key-Set-Up Phase*** can be proven by the mathematic induction method. To present our scheme formally, we indicate the motion between node *i* and *j* step by step as follows.

***Key Pre-distribution Phase***

Step 1 The system chooses *r* columns from the columns of *A* as per $PRG(f(ID_i))$, and the system computes the sum of these *r* columns as $c_i$'.

Step 2 The system pre-distributes $c_i$' to node *i*. The node *i* keeps $c_i$' secret.

Step 3 The system chooses another *r* columns from the columns of *A* as per $PRG(f(ID_j))$. Then the system computes the sum of these *r* columns as $c_j$'.

Step 4 The system pre-distributes $c_j$' to node *j*. The node *j* keeps $c_j$' secret.

***Common-Key-Set-Up Phase***

Step 1 Two nodes exchange their IDs.

Step 2 Node *i* computes $PRG(f(ID_j))$ and chooses the corresponding *r* columns in *G* according to the result of $PRG(f(ID_j))$. Then, the node computes the sum of these *r* columns as $g_j$'.

Step 3 The pre-distributed column, $c_i$' in the node *i* multiplies times $g_j$'.

Step 4 Node *j* does the same computation as in II using $ID_i$, and node *j* can then obtain $g_i$'.

Step 5 The pre-distributed column, $c_j$' in node *j* multiplies times $g_i$'.

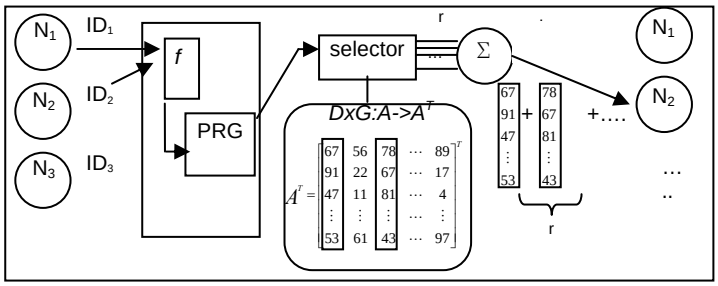


Figure 1. The ***Key Pre-Distributed Phase*** example of part (a) within single key space

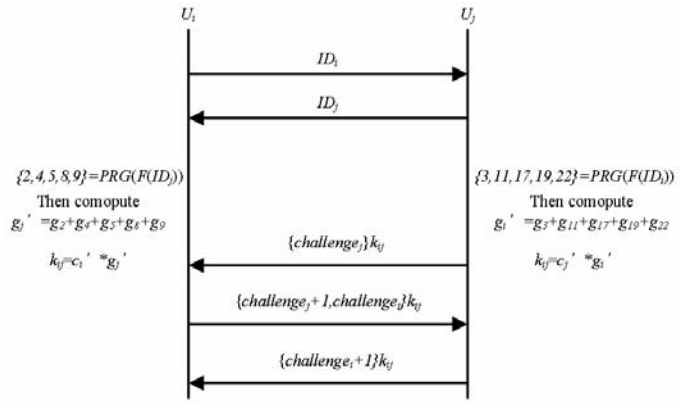


Figure 2. The ***Key Pre-Distributed Phase*** example of part(a)

© 2008 ACADEMY PUBLISHER

Fig. 1 illustrates the ***Key Pre-distribution Phase*** of part (a) in our scheme within single key space, shown on the last page.

Fig. 2 illustrates the ***Common-Key-Set-Up Phase*** of part (a) in our scheme within single key space, shown on the last page.

*B. Part (b): Multi-key space*

Next, we consider the case of multi-key space inspired by Du et al [6]. According to the reports in [21] and [22], a connection ratio of 100% is not necessary, as the verification procedure is always done in the overlapping key space for those nodes. However, an intruder could easily execute an impersonation attack or a Sybil attack because the key management mapping to each node is not one-to-one. The intruder could impersonate one or more users by capturing only a small portion of the network's members. In response to this concern, another scheme we have proposed, based on multi-space is explored in order to thoroughly repel such kinds of attacks. In other words, the impersonation attacks impacting Du et al [6] are prevented when our part (b) scheme is applied.

In part (b), the process is also divided into two phases: the ***Key Pre-distribution Phase*** and the ***Common Key Set-up Phase***. First the authority would generate $|S|$, which are the Blom scheme key spaces, by using the distinct symmetric secret matrix $D$. Because of the different key space, this means that there needs to be a set of $A$ matrices. If the nodes use a set of $A$ matrices to verify each other, the nodes would obtain a different symmetric key. In the ***Key Pre-distributed Phase***, similar to part (a) above, the authority would obtain a serial number using the node's unique *ID* as the input factor for the one-way function and the pseudorandom generator. According to the serial number, the authority chooses $k$ matrices of $A$ from the $S$. Then the authority chooses the column combination, like the one described above, from each matrix $A$ among the $k$ matrices of $A$ individually to pre-distribute into the node. After that deployment, any two nodes who want to verify each other would execute the ***Common Key Set Up Phase***. In the ***Common Key Set Up Phase***, the two nodes would know the combination situation of the columns which are pre-distributed among each chosen matrix $A$ in the other node according to the same computation that was executed by the authority by using the other node's *ID* as the input. Then, the node checks the serial number to find out whether the other node has the column combination from the same matrix $A$ that is also pre-distributed in itself. If the node finds that one or more of the matrices pre-distributed in the other node is overlapped with one of its own, he would then confirm that a safe link between the other node could be set up. It then executes the rest of the process from 3.1 by choosing one of the overlapped $A$ matrices. If the node can't find an overlapping matrix of $A$, the two nodes won't set up the safe link.

***Key Pre-distribution Phase***

Step 1 The system makes several distinct matrices $A$'s. We call these matrices $A$'s, a set, $S$.

Step 2 The system computes $PRG(f(ID))$ for each *ID* of each node individually. Part of this result could then be considered as a series of a set number of bits, each representing a matrix within the set of $S$. The other part of this result could be considered as another series of a different set number of bits, each representing an individual column combination of $S'$.

***Common-Key-Set-Up Phase***

Step 1 Two nodes exchange their IDs.

Step 2 These two nodes compute $PRG(f(ID))$ using the other node's ID. Upon completion, each node knows the overlap matrix, $A$, which is present in the other's node. Fig. 3 with a numeric example illustrates part (b) of our scheme within the multi-key space, shown on the last page.

Step 3 The two nodes then continue with the process of part (a) of Section 3.1 in order to set up the common key with overlap matrix $A$.

*C. The Extension of Part (a) and Part (b)*

In this section, we propose the extension of our schemes part (a) and part (b), of single key space and multi-key space, respectively. We also use the combination of the column vectors and pre-distribute into the nodes. However, each column vector chosen would assign its own particular coefficient, and that column will first multiply this coefficient. Then the rest of the procedure is similar to the scheme that we proposed in 3.1 and 3.2. The advantage of the extension is that the number of the column vector combination that the authority could use to pre-distribute into the nodes is much greater than the ones from the schemes in 3.1 and 3.2. It also means that the number of the network members using this extension would be much greater than the ones in schemes 3.1 and 3.2.

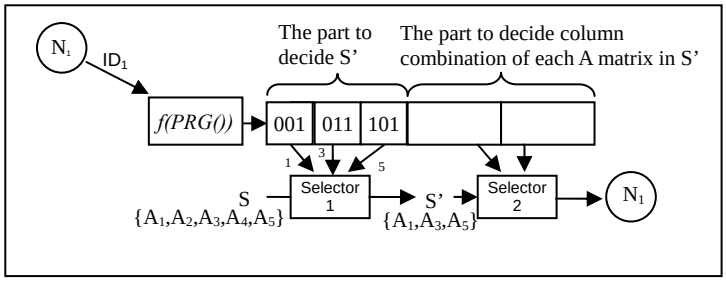


Figure 3. The *Key Pre-Distribution phase*: step 2 of part (b) of our scheme

© 2008 ACADEMY PUBLISHER

In the extension of part (a), there are two phases, ***Key Pre-distribution Phase*** and ***Common Key Set-up Phase***. In the ***Key Pre-distribution Phase***, without loss of generality, the authority would generate a one-way function, *f* and pseudorandom generator, *PRG*. Each node has a unique identification, *ID*. The authority computes *f*(*ID*), and forwards it to be initial state of the *PRG*. The output of the *PRG* for a node $N_i$ is the set of $\{i_1, i_2, i_3, i_4, \ldots, i_r; xi_1, xi_2, xi_3, \ldots, xi_r\}$, where $\{i_1, i_2, i_3, i_4, \ldots, i_r\}$ indicates the indices of the pre-distributed columns and $\{xi_1, xi_2, xi_3, \ldots, xi_r\}$ indicates the corresponding coefficients of the pre-distributed columns. After the deployment, the two nodes that want to set up the communication link would go into the next phase, the ***Common Key Set-up Phase,*** verifying their identities with each other. It is done prior to the data transmission between the two connecting nodes. In ***Common Key Set-up Phase***, the two nodes would broadcast their *IDs* first. In this way, the two nodes are capable of obtaining the result of *f*(*ID*) for the ID of the other node. Then, through the *PRG*, the node will obtain an output with serial numbers to show the public information of the other node's pre-distributed vector. The node would then choose the corresponding column from public matrix *G* according to the output of the *PRG* and then multiply the corresponding coefficient times the corresponding column from *G*.

In this scenario, we consider the case of multi-key space inspired by Du et al [6]. However, they achieve a higher connection ratio as compared to that of [6]. However, according to the reports in [21] and [22], it is not necessary to obtain the connection ratio up to 100%, as the verification procedure is always done in the overlapped key space for those nodes which want to successfully set-up the connection path. Yet, the intruder could still easily execute the impersonation attack and the Sybil attack because the key management mapping to each node is not one-to-one. The intruder could impersonate one or more users by capturing only a small portion of the network's members. In response to this concern, the extended version based on multi-space, from part (b), is explored to thoroughly repel these kinds of attacks. In other words, the impersonation attacks impacting Du et al [6] are the threats that are resolved when our scheme is applied.

In this extension of our scheme from part (b), the process is also divided into two phases: the ***Key Pre-distribution Phase*** and the ***Common Key Set-up Phase***. First, the authority would generate $|S|$ which are the Blom scheme key spaces, by using a set of matrices, $D, \{D_1, D_2, D_3, \ldots\}$. The different key space here means that there needs to exist a set of matrices of $A$, *i.e.* $\{A_1, A_2, A_3, \ldots\}$. If the nodes use a set of $A$ matrices to verify each other, the nodes would then obtain different symmetric keys. In the ***Key Pre-distribution Phase***, similar to the one in the extension above, the authority would obtain a serial number using the node's unique *ID* as the input factor for the one-way function and the pseudorandom generator. According to the serial number, the authority chooses $k$ matrices of $A$ from the $S$. Then the authority chooses the column combination, like the one described in the above extension, from each matrix $A$ among the $k$ matrices of $A$, individually, to pre-distribute into the node. After the deployment, any two nodes who want to verify each other's identity would execute the ***Common Key Set Up Phase***. During ***Common Key Set Up Phase,*** the two nodes know the combination situation of the column pre-distributed among each chosen matrix $A$ in the other node according to the same computation that was executed by the authority by using the other node's *ID* as the input. Then, the node checks the serial number to find out whether the other node has the column combination from the same matrix $A$ that is also pre-distributed inside it. If the node finds one or more matrices pre-distributed in the other node that overlaps with one of its own, the node would then confirm that a safe link between the other node could be set up and then it executes the rest of the process in the 3.1 by choosing one of the overlapped $A$ matrices. If the node couldn't find an overlapped matrix of $A$, the two nodes won't set up the safe link.

Our scheme extensions of part (a) & part (b) are as follows:

***Key Pre-distribution Phase***

I. According to the results of $PRG(f(IDi))$ and $PRG(f(IDi))$, the system chooses a subset of $S$, and then chooses the column combination from the column of $A_i$ and the corresponding coefficient.
II. $c'_i$ is pre-distributed into Node $i$, and $c_j'$ is pre-distributed into node $j$

***Common-Key-Set-Up Phase***

I. Two nodes exchange their IDs.
II. According to the ID, the other node would choose a suitable column combination in $G$, and obtain the coefficient to the column combination.
III. The two nodes multiply the pre-distributed column belonging to the overlapped matrix $A_i$ to the column combination corresponding to the other node with the coefficient to obtain the symmetric key.

## IV. Analyses & Comparisons

In this section we discuss our proposed scheme and other relevant schemes. We will then compare our scheme with the other relevant schemes in order to show the advantage of our scheme.

Because the secret column pre-distributed in our scheme depends on the nodes' individual IDs, the impersonation attack would be efficiently repelled. Our scheme is especially resistant to the Sybil attack, an attack that is more aggressive than the attack examples provided for in Du et al. [6]. The scheme in Du et al. [6] randomly chooses the secret column pre-distributed into each nodes. Therefore, the nodes in that situation are unable to authenticate the pre-distributed column in the other node. As a result, the intruder could deceive the other node using the column that the intruder compromised.

First, we define the size of a matrix $A$ as $\lambda xn$ in our

© 2008 ACADEMY PUBLISHER

scheme. Then, the number of the column vector combinations from our part (a) scheme is $C_r^n$; the number of the column vector combinations from our part (b) scheme is $C_k^{|s|}C_r^n$; however, the number of the column vector combination in the extension of part (c) of our scheme is unlimited.

Our schemes have the same security level as the Blom scheme. In addition, our scheme provides a strong relationship between the node identity and the secret column vector. The nodes can verify each other by checking the combination of the column vector in the other node. In particular, our scheme from part (b) effectively resists the Sybil attack, yet the scheme in Du et al. fails to do so at all.

The relevant comparisons of the connectivity, computation and storage among our scheme, Gligor et al. [8], Newsome et al. [15], and Du et al. [6] are shown in Table 1.

## V. CONCLUSION

The security of ad hoc networks is becoming more and more of a serious problem, no matter what the usage level of any given ad hoc network is. Therefore, we have proposed a scheme that provides a solution for some of the challenging attacks that are currently plaguing ad hoc networks. Our scheme uses the concept of linear combination and principles of algebra theory to own more secret columns associated with the phase of pre-distribution when compared to past research. Accordingly, we offer more secure consideration in order to prevent a variety of Sybil attacks. In our scheme, we also provide an authentication mechanism for when a new node wants to join this system. Another advantage is that our scheme enlarges the key space pre-distributed into the nodes. It is an efficient and secure authentication scheme that sets up more reliable communication links between all the nodes in any ad hoc systems.

## VI. ACKNOWLEDGMENT

This work was supported in part by the National Science Council of the Republic of China under the Grant NSC 95-2221-E-015-002-MY2 and by the iCAST project sponsored, National Science Council under the Grants NSC 95-3114-P-001-001-Y02 and NSC 96-3114-P-001-002-Y.

TABLE 1.
THE COMPARISON OF RELEVANT WORKS.

| | Computation | Connectivity | Storage |
|---|---|---|---|
| Gligor et al. [8] | 0 | $1-\frac{(C_{k_p}^{\|U\|})(C_{k_p}^{\|U\|-k_p})}{(C_{k_p}^{\|U\|})^2}$ | $64k_p$bits |
| Newsome et al. [15] | $\|f\|+\|PRG\|$ | $1-\frac{(C_{k_p}^{\|U\|})(C_{k_p}^{\|U\|-k_p+th})}{(C_{k_p}^{\|U\|})^2}$ | $64k_p+[f]+[PRG]$ |
| Du et al. [6] | $(\lambda-1)Addition+\lambda Multiplication$ | $1-\frac{(C_k^S)(C_k^{S-k})}{(C_k^S)^2}$ | $64k\lambda$bits |
| Our scheme part (a) | $[(\lambda-1)+r*\lambda]Addition+\lambda Multiplication+\|f\|+\|PRG\|$ | 1 | $64\lambda$bits+$[f]$+$[PRG]$ |
| Our scheme part (b) | $[(\lambda-1)+r*\lambda]Addition+\lambda Multiplication+\|f\|+\|PRG\|$ | $1-\frac{(C_k^S)(C_k^{S-k})}{(C_k^S)^2}$ | $64k\lambda$bits+$[f]$+$[PRG]$ |
| Our scheme part (c) | $[(\lambda-1)+r*\lambda]Addition+2\lambda Multiplication+\|f\|+\|PRG\|$ | $1-\frac{(C_k^S)(C_k^{S-k})}{(C_k^S)^2}$ | $64k\lambda$bits+$[f]$+$[PRG]$ |

*Addition*: The addition computation

*Multiplication*: The multiplication computation

$|f|$: The computation of the one-way function

$|PRG|$: The computation of the pseudorandom generator

$r$: The number of columns chosen in the column combination

$\lambda$: The security factor in the Blom scheme that the Du et al scheme. and our scheme use

$|U|$: The size of the key pool in the Gligor et al. and the Newsome et al..

$k_p$: The number of the keys pre-distributed into each node using Gligor et al. or Newsome et al.

$S$: The number of the Blom scheme key space using Du et al. and our scheme.

$k$: The number of the key space pre-distributed into each node using Du et al. and our scheme

$[f]$: The storage of the one-way function

$[PRG]$: The storage of the pseudorandom generator

© 2008 ACADEMY PUBLISHER

## REFERENCES


[1] A. Banerjea, "A Taxonomy of Dispersity Routing Schemes for Fault Tolerant Real-Time Channels," *Proceedings of ECMAST*, vol. 26, pp. 129–148, May 1996.

[2] S. Buchegger and J. L. Boudec, "Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robust-Ness in Mobile Ad Hoc Networks," *10th Euromicro Workshop on Parallel, Distributed and Network-based Processing* (*PDP*), 2001.

[3] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, October 2000.

[4] H. Du, W. Li, and D. P. Agrawal, "Routing Security in Wireless Ad Hoc Networks," *IEEE Communication Magazine*, vol.40, no.10, October 2002.

[5] J. R. Douceur, "The Sybil Attack," *Proceedings of the1st International Workshop on Peer-to-Peer Systems* (*IPTPS '02*), March 2002.

[6] W. Du, J. Du, and Y.S. Han, "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks," *Proceedings of 10th ACM Conference on Computer and Communication Security* (*CCS'03*), pp. 42-51, October 2003.

[7] W. Du, J. Du, Y.S. Han, and P. K. Varshney, "A Witness-Based Approach for Data Fusion Assurance in Wireless Sensor Networks," *Proceedings of IEEE 2003 Global Communications Conference* (*Globecom'2003*), San Francisco, CA, December 2003.

[8] 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*, pp. 41-47, November 2002.

[9] K. Ishida, Y. Kakuda, and T. Kikuno, "A Routing Protocol for Finding Two Node-Disjoint Paths in Computer Networks," *Proceedings of International Conference on Network Protocols*, pp. 340-347 November 1992.

[10] D. B. Johnsom and D. A. Maltz," Dynamic Source Routing," http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt, April 2003.

[11] H. Kellerer, U. Pferschy, D. Pisinger, "Knapsack Problems," Springer-Verlag Berlin Heidelberg, 2004.

[12] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor: Attacks and Countermeasures," *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 113-127, May 2003.

[13] J. Lundberg, "Routing Security in Ad Hoc Networks," Helsinki University of Technology, http://citeseer.nj.nec.com/400961.html.

[14] S. Marti, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proceedings of MOBICOM 2000*, pp. 255-265, August 2000.

[15] J. Newsome, "The Sybil Attack in Sensor Networks: Analysis & Defense," *Proceedings of 21st Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM'03*), April 2003.

[16] C. Perkins and E .Royer, "Ad hoc On-Demand Distance Vector Routing," *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Application*, February 1999.

[17] C. E. Perkins, "Ad Hoc Networking", Addison-Wesley, Reading, MA, 2001.

[18] C. E. Perkins, P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing for Mobile Computer," *Processing of ACM SIGCOMM'94*, pp. 234-244, September 1994.

[19] R. Blom, "An Optimal Class of Symmetric Key Generation Systems," *Proceedings of EUROSRYPT'84*, pp. 335-338, 1984.

[20] Rice University Monarch Project, "Mobile Networking Architectures," Home Page Available from http://www.monarch.cs.rice.edu.

[21] E. Shih, S. Cho, N, Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks," *Proceedings of ACM MOBICOM'01*, Rome, Italy, pp. 272-286, July 2001.

[22] Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Network: A Survey," *Computer Networks*, vol. 38, pp. 393-422, 2002.


© 2008 ACADEMY PUBLISHER