Secret Key Revocation in Sensor Networks*

YoungJae Maeng, Abedelaziz Mohaisen, and DaeHun Nyang

Information Security Research Laboratory
Graduate School of IT & Telecommunication - INHA University
253 YongHyun-dong, Nam-gu, Incheon 402-751, Korea
{brendig,asm}@seclab.inha.ac.kr, nyang@inha.ac.kr
http://seclab.inha.ac.kr

Abstract. Many challenging security-related issues have been studied in wireless sensor networks to provide a demanded quality and security for deliverable data. Yet, one of these issues which are not handled is the secret key revocation. In a semi-dynamic, resource-constrained and long-living sensor network with self organization features, traditional revocation methods are not desirable and somehow impractical. Through this paper, we discuss the rising issue of key revocation due to the pre-distribution and provide several techniques, structures and algorithms for several network and security conditions and requirements. In addition to the saving of the resources represented by the communication, computation and memory, we provide an extension for special-case networks in which our work can provide a higher performance.

1 Introduction

Wireless sensor networks have received a wide attention of research due to its promising variety of applications that include both the civilian and military purposes. Civilian applications include environmental and habitat monitoring, acoustic and seismic detection, medical and process monitoring and smart spaces. The military applications include battlefield surveillance, location determination and others [4,11]. Though the lifetime maximization, self configuration and mobility are critical issues of the study, the security is still one of the hottest issues to be researched. In the security regard, due to its computational feasibility and resources consumption, the secret key cryptography using the same key for both the sender and receiver of the data is mainly used on the sensor nodes with limited resources. As these nodes construct a network without a pre-dedicated infrastructure and with limited self-resources, the key pre-distribution received a big interest of research [3,6,7,9].

Most of the key pre-distribution schemes consider the distribution itself and did not handle uprising security issues such like the key revocation [3,6,7,9].

* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement)(IITA-2006-C1090-0603-0028).

J. Indulska et al. (Eds.): UIC 2007, LNCS 4611, pp. 1222–1232, 2007.

Upon the compromise of a secret key, the revocation is required to cancel a key or even to replace it with another one which is still secret for the attacker. The problem of the key revocation in sensor network is represented by the resources (specially, the communication and memory) required for exchanging (or even distributing) the list of revoked keys indicted by their identifiers. One of the solutions that are typically used is to publish the identifiers of the keys to be revoked as it is. That is to say, for a network of size N and r number of keys to be revoked, $k \times \lg N$ bits are required as a memory and communication overhead. Given that a single Kilo bit transmission over a 100 link requires an equivalent energy of performing 3 Mega instructions on a typical sensor network [14], this type of representation is not an efficient to be used on the sensor networks the early mentioned limited resources.

In this paper, we introduce the researches performed on the key pre-distribution to determine the required element to be revoked followed by our contributions. On the side of the paper structure, §2 background works, §3 lists our contributions and definitions, §4 presents our contributions in details, §5 is on the analysis and evaluation, §6 extends our work for further applications and §7 includes the conclusion and future works.

2 Basic and Relying Schemes

Two of the early works in [1,2] are widely known for its novelty. For network of N nodes, in the first work by Blom *et al.* [1] a symmetric matrix of size $N \times N$ is required to store the different N^2 keys of the entire network. Node $s_i \in N$ has row and column in the matrix. If two nodes s_i, s_j would like to communicate, they use the entries $\mathbf{E_{ij}}$ in s_i side and $\mathbf{E_{ji}}$ in s_j side which are equal. To reduce the memory requirements, a slight modification is introduced by Du *et al.* in [6]. The following are defined, a public matrix $\mathbf{G}$ of size $(\lambda + 1) \times N$ and a private symmetric matrix $\mathbf{D}$ of size $(\lambda+1) \times (\lambda+1)$ where $\mathbf{D}$ entries are randomly generated. Also, $\mathbf{A} = (\mathbf{D} \cdot \mathbf{G})^{\mathbf{T}}$ of size $N \times (\lambda+1)$ is defined. For a node s_i, row $\mathbf{R_i}$ in $\mathbf{A}$ and column $\mathbf{C_i}$ in $\mathbf{G}$ are dedicated. When two nodes s_i, s_j would eventually communication securely, they exchange their $\mathbf{C_i}, \mathbf{Cj}$ and $k_{ij} = \mathbf{R_i} \cdot \mathbf{C_j}$ is computed by s_i and $k_{ji} = \mathbf{R_j} \cdot \mathbf{C_i}$ by s_j and used as shared key. The second work by Blundo *et al.* [2], a Symmetric Bivariate Polynomial (SBP) is used to distribute keys for N nodes. The SBP is in the form of $f(x, y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j, (a_{ij} = a_{ji})$ of degree $t \leq N$. For a node s_i with identifier i, the share $g^i(y) = f(i, y)$ is calculated and loaded to its memory generate secure keys. Similarly, for two nodes s_i, s_j, $k_{ij} = g^i(j), k_{ji} = g^j(i)$ are are evaluated locally and used respectively.

The early scheme of key pre-distribution specifically for WSN is introduced by Eschenauer-Gligor (EG) [7]. Each node is let to randomly pick a key ring S_k of size k from big keys pool of size P guaranteeing a probabilistic connectivity $p_{actual} = 1 - \frac{((P-k)!)^2}{(P-2k)!P!}$. If two nodes s_i, s_j share a key $k : k \in S_{k_i} \cap S_{k_j}$ they use it a secret key. Otherwise, a path discovery phase via intermediate nodes is performed. To improve the resiliency, Chan *et al.* proposed the q-composite scheme [3]. Using the same procedure of EG, a key between two nodes s_i, s_j is available

iff $S_{k_i} \cap S_{k_j}$ is a set of q keys. If $\{k_1, \ldots, k_q\} \in \{S_{k_i} \cap S_{k_j}\}$, $\mathbf{hash}(\mathbf{k_1}||\mathbf{k_2}, \ldots, ||\mathbf{k_q})$ is used as k_{ij}, k_{ji}. Otherwise, intermediate node(s) are used.

In addition to [1], Du *et al.* proposed two schemes in [6,5]. In the first work they introduced a location based scheme by avoiding the unnecessary memory, communication, and computation with reasonable connectivity [5]. In [6], a multi-space matrix scheme based on [1,7] is introduced. A τ number of private matrices $\mathbf{D}$ is selected randomly out of ω pre-constructed matrices providing connectivity of $p_{actual} = 1 - \frac{((\omega-\tau)!)^2}{(\omega-2\tau)!)\omega!}$. Different $\mathbf{A}$'s are created using the different $\mathbf{D}$s. τ rows of the different $\mathbf{A}$s are selected and assigned for each node. For (s_i, s_j), If they have a common space $\tau_{i,j} : \tau_{i,j} \in \tau_i, \tau_j$, the rest of Blom's is performed, else, an intermediate space is used to construct a key path. At the same time, Liu *et al.* proposed several schemes [9,10] for key distribution and mainly based on [2]. In [9], Blundo's scheme is used by assigning more than polynomial for each node similar to EG scheme [7]. In [10], a two dimensional grid deployment structure is used where nodes are deployed on different intersection points of the grid and different polynomials are assigned for the different rows and columns of the grid. Two nodes establish direct key if $R_i = R_j$ or $C_i = C_j$. Else, an intermediate node is used in indirect key establishment phase.

Since we interested in the item to be revoked and not the keying material itself, based on [12] which ideally matches with other schemes in [3,6,7,9] in this regard, the identifier which is used to identify the node itself is also used as an associated identity for the corresponding keying material. Given that, once the identifier of the a given node, the secret key associated with the node in [7,3] is revoked, the polynomial share associated with a node [9,10,12] is revoked, and the matrix's row or column associated with a node [6,5] is revoked. To make the above survey clear, the conclusive comparison in Table 1

Table 1. Resources usage vs. a provided connectivity and revocation capabilities

Scheme	Comm.	Com.	Memory	Conn.	Revocation
GBS [10]	c	SBP Evaluation	ID+2 SBP	$\frac{2}{N^{1/2}-1}$	NO
EG [7]	$C \lg(S_k)$	$\frac{(2C+p-p_k)}{2} \lg C$	S_k keys	$1 - \frac{((P-k)!)^2}{(P-2k)!P!}$	NO
CPS [3]	Constant	c	S_k keys	$\frac{m}{N}$	NO
DDHV [6]	$C \lg(n \times \tau)$	2 vectors mult.	$\tau + 1$ vectors	$1 - \frac{((\omega-\tau)!)^2}{(\omega-2\tau)!\omega!}$	NO
HGBS[12]	c	SBP Evaluation	ID+n SBP	1	NO

3 Contribution, Definitions and Terms Elucidation

3.1 Contributions

In this paper, our contribution is as follows:

– Introducing the concept of the symmetric key revocation in sensor networks and the hurdle of the revocation resulting from the representation of the nodes or keys identifiers' representation.

- Introducing the Bit Vector Scheme as an efficient mechanism for symmetric key representation in a key revocation list.
- Employing the Complete Subtree Cover [13,8] for groups representation as a possible method for reducing the revocation list's size.
- Study the impact of simple and dynamic encoding such like the run length encoding on the representation in a simplified version of bit-compression.
- Conclude our contributions by a comparison between our work and the only possible representation method which is the naïve scheme.

3.2 Definitions and Terms Elucidation

In this section, we introduce two definitions that our work relies on. The first one is the complete subtree (CS) [13,8]. In our work, we don't use the key derivation of CS but we just use the representation method to reduce the overhead of set of identifiers representation. In addition, other broadcast encryption method which is the subtree difference (SD) can be used. In CS, we consider a complete binary tree $\mathcal{T}$ with leaves that represent the different network nodes (in this case, the key's/keying material's identifier) $\mathcal{N} : |\mathcal{N}| = N$. The different path from the root to the leaf represents a corresponding leaf identifier (ID). The path itself is represented as left of zero and right of one (relatively from the parent). The set of node's identifiers (that represents the corresponding keying material) to be revoked is $\mathcal{R} = \{v_1, v_2 \dots v_r\} : |\mathcal{R}| = r$. In definition 2, we are interested in the COVER which is the set of reduced IDs which are required to represent a set of revoked identifiers.

Definition 1 (Naïve Method for Revoked Identifiers Representation). For a space S that permits $2^{|S|}$ possible identifiers, the naïve method for the identifiers representation is by listing these identifiers of at same length resulting that the size of the list is the length of the list multiplied by S.

Definition 2 (Complete Subtree Cover for Representation Reduction). In the above tree, the CS cover of a group of leaves $\mathcal{R} \subset \mathcal{T}$ is the mechanism of finding the set of nodes $V_1 \dots V_t \subset \mathcal{T}$ (represented by its ID's beginning from the root) such that the group of nodes $v_{i1} \dots v_{it}$ that represent a complete subtree are rooted at V_i for all i such that $0 < i < t$, $V_i \cap V_j = \phi$ for any $i \neq j$ and $V_1 \cup V_2 \dots \cup V_t = \mathcal{R}$ considering that V_i is a representative group for the set of nodes rooted at V_i. Here, the set ID of V_i is the binary string constructed by concatenating bits assigned to each branch from the root to the V_i. Note that the length of ID is always less than or equal to $\lg N$.

Definition 3 (Bit Vector for Efficient Representation). The Bit Vector Scheme which is a relative representation mechanism for sequential identifiers including those to be revoked and mainly used to reduce the CRL length. In BVS, for a network of N nodes, a bit vector S of length N is constructed. In S, the $i^{\mathbf{th}}$ bit indicates the validity of a key associated with the identifier i. Also, a '0' valued location in S represents valid (un-revoked) key and '1' represents the revoked key. In the node's side, the keys' identifiers to be revoked can be extracted from offset of the 1's occurrences.

Definition 4 (Run-length encoding RLE). Data encoding algorithm that maps a plain binary string $\mathcal{P} : \mathcal{P} = p_1p_2p_3 \dots p_a$ into an encoded binary string $\mathcal{C} : \mathcal{C} = c_1c_2c_3 \dots c_b$. Considering the instances $\xi_i, \xi_j \in \mathcal{C}$ and $\varphi_i, \varphi_j \in \mathcal{P}$, $\xi_i = \xi_j$ if and only if $\varphi_i = \varphi_j$. The notation h-RLE-l determines both h and l as the header and the length where h expresses the input plain string's header bit and l determines the length of the input and the output as follows: maximum length of input length is 2^{l-1}, length of the encoded output is l and the plain input's header is h and other bits before it (if any) are equal to h's binary compliment.

4 Putting It All Together: Symmetric Key Revocation

4.1 On Symmetric Key Revocation

As we early mentioned, the symmetric key pre-distribution process uses immediate cryptographic secret key [3,7] or secret key's generation material in what so called keying material. This material can be a share of polynomial [2,9,10] or a vector from a matrix [1,6]. In this paper, we are interest in the identifier of the any type of the secret information and not the key itself. As early mentioned, fortunately in most of key pre-distribution mechanisms, the keying material or the key itself are associated with a node with a unique identifier. If we would like to revoke a given key or keying material, then it is enough to publish the identifier of the associated node in the revocation list to be revoked. In the following subsections, we show how the early definition of the BVS, CS and the RLE encoding are used in this revocation process. In the remaining part of this paper, we use the expression key or key to express the key itself or the keying material as well.

4.2 BVS for Symmetric Key Revocation

Scenario I: Static Revocation: The Bit Vector Scheme (BVS) which is a relative representation mechanism for the different keys including those to be revoked can be used for revocation list's reduction. Based on definition 3, it is enough to make a given position of the bit vector string as zero to indelicate a non-revoked identifier that has the offset value of the bit. In the same way, it is enough to make a given position of the bit vector as one to indicate a revoked identifier of the offset of the valued bit. In the node's side, the key's identifiers to be revoked can be extracted from offset of the 1's occurrences. Since this type of substitution considers only one chance for each identifier to be represent as revoked or not, we denote this initial representation as a static scheme (BVS-S). This type of revocation is good for static networks with almost same age and almost same compromise chances. An example of how this scheme works is shown in Fig. 1❶.

Obviously, Not all of the networks follows the same policy of the above one. In a more dynamic network that permits a frequent join-disjoin activities, if a key is revoked another key with other ID will replace it (or might be with the same

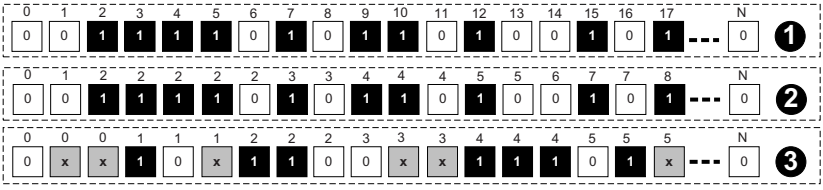


Fig. 1. Different Scenarios of the Bit Vector Scheme for Key Revocation

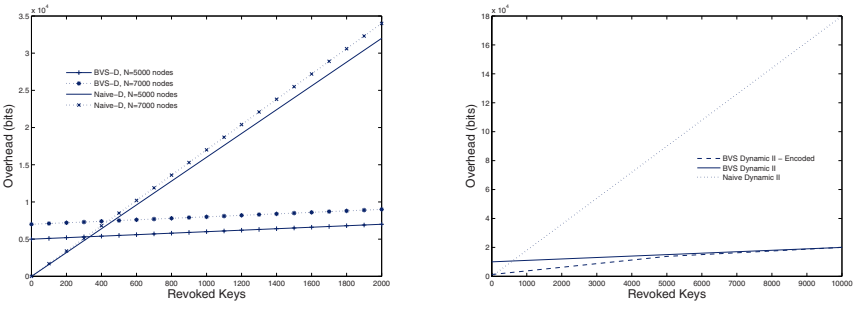
ID and different parameters). Thus, possibly we would like to extend the BVS-S to cover the multi-revocation case. In the following, we provide two extensions with scalability/memory requirements trade-off.

Scenario II: Dynamic Revocation

Dynamic with Pre-defined Length (BVS-D-I): In the first approach, we provide each sensor node's key with a pre-dedicated (say 10) number of bits to handle a number of revocations for each. The equal number of bits for each sensor node will provide an auto-discrimination mechanism. However, it provides a low efficiency especially when the number of revoked keys is small (in what we call a loosely representation). An example of how does this scheme works is shown in Fig. 1❸.

Dynamic with Expendable Length (BVS-D-II): To overcome this problem, let's consider the BVS-S with some modification. Initially, the bit vector is initialized by 0's when no keys are revoked. Once a key is revoked, a '1' is attached immediately before the corresponding key's '0' offset.

Thus, one revocation adds only one bit to the bit vector. When a check of whether $i^{\mathbf{th}}$ certificate is valid or not is required, first find out the i^{th} block that represent sensor node i and count the number of '1' (say, n) which means that the $(n+1)^{\mathbf{th}}$ issued key for node i is valid. The $i^{\mathbf{th}}$ block for node i is composed of the $i^{\mathbf{th}}$ '0' in the bit vector and proceeding '1's after the $(i-1)^{\mathbf{th}}$ '0'. As in the first scenario, each node can have a limited and pre-defined number of revocation chances (say 10 as before) which can be overflowed. An example that shows how this scheme works is shown in Fig. 1❷. Unlike the other scenarios, BVS-D-II is fully dynamic in that it can support (relatively) infinite times of revocation where the list cost is typically as much as the number of revoked keys plus an additional initial overhead. For the naïve CRL, a bigger range of IDs is provided to permit a 10 revocation chances for each key. While 14 bits are enough to represent 10,000 keys, 18 bits are dedicated to provide 10 revocations for each. In BVS-D-I, an extended number of bits per node is required even they are not used.



(a) Dynamic I: BVS vs. Naïve (b) Dynamic II: BVS vs. Naïve

Fig. 2. Communication overhead in bit using (a) Naïve solution vs. BVS-D-I with different network size (b) naïve vs. encoded and un-encoded BVS-D-II

4.3 CS Cover for Symmetric Key Revocation

The Complete Subtree (CS) of definition 2 can be used directly to reduce the the representation size for the list of the key's to be revoked. As an example, if the set of IDs to be revoked r is $\{0011, 0100, 0101, 0110, 0111, 1011, 1101$ and $1110\}$, the final list representation of the reduced list is {0011,10,101,1101,1110} which includes all of the required IDs. From the simulation results, CS probably reduce the overhead when r is about 5% of the network size but it greatly reduces the overhead when r is large enough (say, $r > 20\%$). This efficeincy of reduction is shown in Fig. 3(a).

4.4 On the Encoding

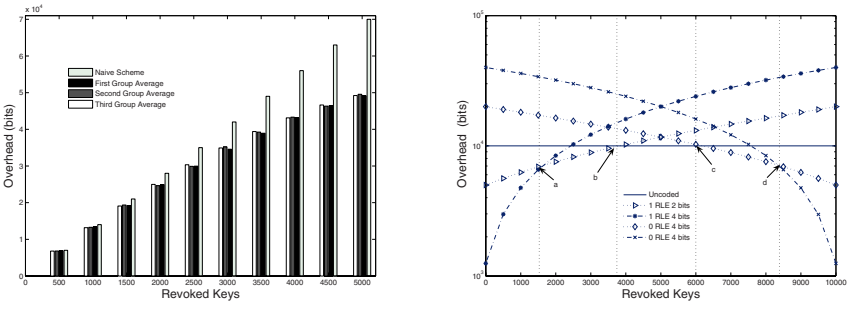
Technically, the in advance knowledge of multiple occurrences of same bits in BVS (for the basic BVS, BVS-D-I, BVS-D-II and even in CS) makes it possible to use the encoding mechanism (RLE) in definition 4 for an efficient compression. The compression efficiency is due to that the long bit vector's fraction of the same bits (consequent ones or zeros) can be eventually represented in a shorter encoded string. The heading and the length parameters of the RLE encoding algorithms determines the structure of the input and the length of the input and output as well. Table 2 shows the two possible heading (i.e. 1 or zero) with two different length (2 and 4).

To get a desired performance of the encoding algorithm, we apply a dynamic encoding using different parameters for h and l at different points (i.e. a, b, c, d in Fig. 3(b))[1]. For the relatively small number of revoked keys (e.g. $r < 40\%$ of the network size or the number of all possible keys in the network), both 1RLE-4b and 1RLE-2b can be used since it's highly probable for long consequent zeros to occur. For a number of revoked certificates r such $40\% < r < 60\%$, the number

[1] The values of these parameters and its percentages are driven from the simulation as shown in Fig. 3(b).

Table 2. RLE with different parameters

1RLE4				0RLE4				1RLE2		0RLE2	
P	C	P	C	P	C	P	C	P	C	P	C
1	0000	000001	0101	0	0000	111110	0101	1	00	0	00
01	0001	0000001	0110	10	0001	1111110	0110	01	01	10	01
001	0010	0000001	0111	110	0010	11111110	0111	00	1	11	1
0001	0011	0000000	1	1110	0011	11111111	1				
00001	0100			11110	0100						



(a) Several CS vs Naïve (b) Dynamic Encoding

Fig. 3. Overhead in bit using (a) The CS cover using different random generation rounds vs naïve solution (b) Dynamic encoding that considers alternation points for the RLE parameters

of the zeros and ones has the same occurrence probability. Thus, we use the non-encoded vector (i.e full use of N bits. For the remaining part, when $60\% < r < 100\%$ of the number of the nodes' certificates, a reverse encoding can be used (i.e. 0RLE-2b,0RLE-4b). To find out the exact percents where the dynamic encoding parameters should be changed, we encode the same string represent different r percents using the different possible parameters and manually calibrate the points at which the encoding algorithm's alternation points. Fig. 3(b) shows the encoded string using different parameters where the intersections that relies below N (i.e a, b, c, d) are used.

5 Analysis and Evaluation

To justify the performance of the proposed scheme, simulating the following schemes was carried out: naïve, naïve Encoded, CS, BVS-S, BVS-D, BVS-S encoded and BVS-D encoded for the both dynamic scenarios. To handle the revocation randomness, we use a random identifier selector that indicates the current compromised ID from the non-compromised pool (which is the worst

case). $N = 10,000$ sensor nodes (the same certificates of the static) and $c = 10$. We noticed the following from the simulation results:

Note 1: the CS scheme to reduce the overhead of the communication and the representation does not follow a constant form. While a 24,917 bits are enough when the number of revoked keys is 2000 from a network size of 10000 nodes which is corresponding to 28000 bits in the naïve scheme, 49140 bits are required when the revoked keys are 5000 at which 70000 bits are required for the naïve scheme. From that we get that the reduction rate at the first case is $\frac{28000-24917}{28000} \times 100\% = 11.012\%$ while it is for the second case as $\frac{70000-49140}{70000} \times 100\% = 29.8\%$. This result is based on the average driven from figure 3(a) however as discussed earlier, this percentage has a range of random variation based on Table 3.

Note 2: the dynamic encoding is achievable through a pre-deployment obtained parameters which permits the node to flip the parameters for different length and headings as shown in Fig. 3(b). Note that, the final overhead of the compressed encoded pattern is always less than the original plain representation of the BVS or any other used scheme.

Note 3: a comparison between our dynamic BVS's two different scenarios and the naïve scheme are shown in Fig. 2. Fig. 2(a) shows that the growth of the overhead is almost constant due to the growth of the network size once it is given and mainly depends on the number of revoked keys in BVS-D-I (where the overhead is typically equal to $d \times N$ where d is the number of revocation chances given for each key) while the naïve scheme depends on both the number of revoked keys and the network size with a linear increment based on the number of revoked keys with a higher slop (which is equal to $\lg N$.) based on the network size which means that the overhead is typically equal to $r \times \lg(d \times N)$ bits.

Note 4: the encoding has parameters which are pre-defined before the installation of the system by considering the size of the network and the algorithm to be used which are dedicated in advance. The RLE dynamic encoding algorithm provides best performance when used with our proposed BVS since it is highly possible to find out the probability of a given pattern to occur in the bit vector with higher probability than others given the number of revoked keys. Based on Fig. 2(b), the overhead using BVS-D-II is equal to $r + N$ which is the best possible reduced overhead that ever achieved for a dynamic revocation. The same as with the BVS-D-I, the naïve solution requires $r \times \lg(d \times N)$ bits.

6 Application: Key Establishment

One of the interesting applications for the BVS in sensor networks is the key establishment. Most of the key establishment procedures relies on exchanging a list of identifiers for keying material or the identifiers of the keys itself as in [3,6,7,9]. For example, in [7] if two nodes would like to establish a key, they

Table 3. The communication overhead in bit for revoking percents of network size using different revocation schemes. $N = 10000$ nodes, r is a percent of the overall network size, **-C** indicates the usage of dynamic RLE.

Scheme	$r = 01\%$	$r = 05\%$	$r = 10\%$	$r = 20\%$	$r = 40\%$	$r = 50\%$
Naïve	1,400	7,000	14,000	28,000	56,000	70,000
Naïve-C	2,595	12,876	26,132	52,104	103,403	130,070
CS	1,355	6,805	13,310	25,004	42,850	49,408
Naïve-D-I	1,800	9,000	18,000	36,000	72,000	90,000
Naïve-D-II	18,000	90,000	180,000	360,000	720,000	900,000
BVS-S	10,000	10,000	10,000	10,000	10,000	10,000
BVS-S-C	1,597	2,994	4,753	7,569	10,000	10,000
BVS-D	10,100	10,500	11,000	12,000	14,000	15,000
BVS-D-II	11,000	15,000	20,000	30,000	50,000	60,000
BVS-D-II-C	3,756	13,778	20,000	27,508	37,498	41,320

Table 4. Overhead comparison between naïve revocation and BVS-based

Scheme	**Original**	**BVS**	**Advantageous**
EG scheme [7]	$k\lvert \lg P\rvert$	P	$k \geq \frac{P}{\lvert \lg P\rvert}$
Liu-Ning-EG [9]	$n^{'}\lvert \lg n\rvert$	n	$n^{'} \geq \frac{n}{\lvert \lg n\rvert}$
DDHV-EG[6]	$\tau\lvert \lg \omega\rvert$	ω	$\tau \geq \frac{\omega}{\lvert \lg \omega\rvert}$
Chan Scheme[3]	m	m	-

first exchange their keys' identifiers list to determine whether they own a common key or not. Similarly, in [9,6] identifiers of the polynomial or the matrix are exchanged respectively (refer to section 2 for the technical details). To show our schemes let us consider [6] where other schemes follow the same way. For the original method, τ matrices are selected from ω number of different matrices for each node. The communication overhead to exchange the identifiers is typically $\tau\lvert \lg \omega\rvert$, however, in BVS ω bits are required. Therefore, our representation will provide better performance as long as $\tau \geq \frac{\omega}{\lvert \lg \omega\rvert}$. For [9], our performance is better when $n^{'} \geq \frac{n}{\lvert \lg n\rvert}$ and so on. Table 4 shows a comparison for the storage/communication overhead between our BVS and the original used mechanisms in key establishment for several schemes.

7 Conclusion and Future Work

In this paper we discussed the uprising problem of key revocation in sensor networks. The main problem is mainly in the required communication overhead for the representation of the different keys to be revoked. We introduced a set of schemes for the efficient revocation based on both the bit vector and the complete subtree cover. Our introduced work provided a reasonable resources saving. We

tested different conditions of the dynamic revocation and the impact of encoding which provided an efficient dynamic compression based on the status of the bit vector and the number of revoked keys. On the other side, we consider the general case of a plat network; however, possible deployment knowledge structures and schemes will be considered in future works. It will be interesting to find and apply different compression algorithms on our specific structured bit vector. As well, several applications will be studied for the BVS.

References

1. Blom, R.: An optimal class of symmetric key generation systems. In: Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques, New York, NY, USA, pp. 335–338. Springer-Verlag, New York, Inc. (1985)
2. Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: CRYPTO, pp. 471–486 (1992)
3. Chan, H., Perrig, A., Song, D.X.: Random key predistribution schemes for sensor networks. In: IEEE Symposium on Security and Privacy, p. 197 (2003)
4. Culler, D., Estrin, D., Srivastava, M.B.: Overview of sensor networks. In: IEEE Computer Society, pp. 41–49 (2004)
5. Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P.K.: A key management scheme for wireless sensor networks using deployment knowledge. In: INFOCOM (2004)
6. Du, W., Deng, J., Han, Y.S, Varshney, P.K., Katz, J., Khalili, A.: A pairwise key predistribution scheme for wireless sensor networks. ACM Trans. Inf. Syst. Secur. 8(2), 228–258 (2005)
7. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: ACM CCS, pp. 41–47 (2002)
8. Fiat, A., Naor, M.: Broadcast encryption. In: CRYPTO, pp. 480–491 (1993)
9. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: ACM CCS, pp. 52–61 (2003)
10. Liu, D., Ning, P., Zhu, S., Jajodia, S.: Practical broadcast authentication in sensor networks. In: MobiQuitous, pp. 118–132 (2005)
11. Mainwaring, A.M., Culler, D.E., Polastre, J., Szewczyk, R., Anderson, J.: Wireless sensor networks for habitat monitoring. In: WSNA, pp. 88–97 (2002)
12. Mohaisen, A., Nyang, D.: Hierarchical grid-based pairwise key pre-distribution scheme for wireless sensor networks. In: EWSN, pp. 83–98 (2006)
13. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: CRYPTO, pp. 41–62 (2001)
14. Pottie, G.J., Kaiser, W.J.: Wireless integrated network sensors. Commun. ACM 43(5), 51–58 (2000)