Structures for Communication-Efficient Public Key Revocation in Ubiquitous Sensor Network[⋆]

Abedelaziz Mohaisen[1], DaeHun Nyang[1,⋆⋆], YoungJae Maeng[1], and KyungHee Lee[2]

[1] Information Security Research Laboratory, Inha University
Incheon 402-751, Korea
asm@seclab.inha.ac.kr, nyang@inha.ac.kr,
brendig@seclab.inha.ac.kr

[2] Department of Electrical Engineering, The University of Suwon
Suwon 445-743, Korea
khlee@suwon.ac.kr

Abstract. In this paper we discuss the uprising problem of public key revocation. The main problem in key revocation includes the relatively large memory and communication required to store and transmit the revoked list of keys. This problem becomes serious as the sensor network is subjected to several constraints. In this paper, we introduce several efficient representation mechanisms for representing a set of revoked identifiers of keys. We discuss several network and revocation scenarios and introduce the corresponding solution for each. To demonstrate the value of our proposed approaches, practical simulation results and several comparisons with the current used revocation mechanism are included.

Keywords: Key revocation, sensor network, communication efficiency, complete subtree cover, bit vector, dynamic run-length encoding.

1 Introduction

The security of wireless sensor network (WSN) has been the issue of critical and challenging research. Due to the limited resources of WSN, public key (PK) algorithms has been discarded from being a solution due to their known computational requirements that enables vulnerability for DoS attacks [1]. In lieu, symmetric key (SK) algorithms have been studied where several SK-related problems have been researched. Because of the weak infrastructure of the WSN that does not permit any traditional symmetric key distribution mechanism such like the key distribution center (KDC), the key predistribution (KPD) in which keys or keying material are assigned for each sensor node in a pre-deployment phase has been studied and several schemes have been introduced [2,3,4,5]. However, the recent advancement of operating public key algorithms such like

[⋆] 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).

[⋆⋆] Corresponding author.

H. Zhang et al. (Eds.): MSN 2007, LNCS 4864, pp. 822–833, 2007.

RSA [6] and ECC [7] on typical sensor nodes has shown a great feasibility that promises a good solution for many of the current security problems in sensor network [8,9,10,11]. Once PK algorithms are deployed in WSN, both of the resiliency and connectivity problems will not be a problem any more [12,13]. However, to make PK algorithms more efficient, different key management services will be required. This mainly includes PK authentication and revocation. As the the cost of communication is much greater than the cost of the local computation on sensor nodes, any efficient security service of PK in WSN should be as light as possible [14]. In this paper, we investigate the problem of the second issue (i.e., PK revocation) and introduce communication efficient schemes to solve this problem.

In the traditional networks, digital certificates (DCs) are used for the purpose of key distribution and authentication. In addition, the certificate revocation is used to revoke a certain certificate for any reason. The digital certificate (e.g. X.509 [15]) includes certificate identifier or simply serial number, *public key which is 1024 bits in case of RSA*, certificate attributes, and digital signature of the certificate's contents. To revoke a PK which is associated with a certificate, the identifier that represents the undesired PK's certificate is published through the entire network. If the number of certificates to be revoked is more than one certificate, their IDs are listed in a Certificate Revocation List (CRL). In the CRL, the IDs of the certificates associated with PKs to be revoked are represented as in Eq. (1) considering the definition of naïve representation in 1. Obviously, the resulting overhead of such representation increases sharply as the number IDs to be revoked increase.

In WSN, we assume that both of the keys and their associated certificates which are for a specific node hold the node's ID. Through the rest of this paper, PK and DC (or digital certificate) are used interchangeably to refer to the revoked identifier.

2 Contributions and Organization

In this paper, we investigate efficient communication schemes for DC revocation and consider the naïve representation scheme defined in 1 as reference work with which our work is compared. Our contribution thereafter includes three parts: (i) introducing a scheme based on the complete subtree cover among the subset coverage framework to represent a set of certificates' identifiers. (ii) Introducing a novel Bit Vector Scheme (BVS) for DC revocation. We study the case of dynamic and static revocation approaches which are equivalent to multiplies and single revocation for same entity (i.e., PK or DC). (iii) Study the probability of specific pattern occurrence in the revocation list. To eliminate extra overhead, we introduce a dynamic run length encoding algorithm that uses pattern-based parameters for efficient compression. In order to demonstrate the performance of the proposed schemes, we introduce a detailed analysis and a practical simulation followed by an extensive comparison with the related works.

The rest of this paper is orgainzed as follows: in section 3, we list the related works on public key and its revocation in sensor networks. In section 4 we introduce the definitions used through the paper. For the details of our contribution, we discuss the naïve and subtree cover mechanisms in section 5. In section 6 we introduce the bit vector scheme with different revocation scenarios. For the efficiency of bit vector compression, we introduce the dynamic run length encoding scheme in 7. We discuss the simulation results in section 8 followed by concluding remarks in section 9.

3 Related Works

The recent results of operating PK algorithms on typical sensor nodes have shown very relevant computational efficiency on contrast of what has been considered as impractical operation for long time. For example, in Gura's *et al.* work [8], practical measurements for elliptic curve cryptography (ECC) [7] and RSA[6] signatures verification have been obtained. It was shown that the verification of ECC signature consumes 1.62 seconds on the typical 8-bit ATmega128 processor which operates at 8 Mega Hertz. In addition, a reduction in the required memory for the code and the data implementation of these algorithms is introduced on other processing platforms (i.e., CC1010 that operates at 14.7456 Mega Hertz). As an extension, Gura *et al.* considered a better implementation of the above algorithms for more efficient energy consumption in [10]. Also, Watro *et al.* developed another limited PK architecture with a practical evaluation of consumed resources per sensor node in so what called TinyPK [11]. The key distribution in TinyOS [16] based on ECC [7] with real measurement and evaluation was taken into account in Malan's *et al.* work [9]. Most recently (February 2007), Ning *et al.* released an update for TinyECC which is an efficient implementation of ECC on TinyOS considering several WSN platforms including MICAz, TelosB and Imote2 motes [17]. To provide PK services such like authentication, Du *et al.* studied the usage of Merkle Authentication Trees [18] and the deployment knowledge to authenticate the public key with memory/communication trade-offs [12]. Also, Nyang *et al.* provided a MAC-based cooperative public key authentication with trade-offs in provided security level and required resources [13].

To the best of our knowledge, PK revocation in WSN has not been studied yet. Thus, our work is the first that formulates the problem of communication efficiency of PK revocation and introduces solutions for this problem.

4 Definitions

In this section, we introduce four definitions which our work is mainly based on. For the Complete Subtree cover (CS) [19,20], we only use the reduction method of identifiers representation and do not use the key generation or assignment mechanisms. In CS, we consider a complete binary tree $\mathcal{T}$ with leaves that represent the different network nodes' DCs. These nodes are represented as $\mathcal{N} : |\mathcal{N}| = n$. The different path from the root to the leaf represents a corresponding DC's ID. The path itself in the tree is represented as left branch of zero and right branch of one relatively from the corresponding parent. The set of node's identifiers to be revoked is $\mathcal{R} = \{v_1, v_2 \ldots v_r\} : |\mathcal{R}| = r$. In definition 2, we are interested in the COVER which is the reduced representation for the set of IDs as we will show in 5.

Definition 1 (Naïve Representation Method). *For a space S that permits $2^{|S|}$ possible IDs representation, the naïve method for the identifiers representation in performed by listing these identifiers at same length resulting that the size of the list is the* *length of the list multiplied by S. The general representation of this method is shown in Eq. (1).*

Definition 2 (Complete Subtree Cover). *The CS cover for a group of IDs associated with $\mathcal{R} \subset \mathcal{T}$ is obtained by finding the set of nodes $V_1 \ldots V_t \subset \mathcal{T}$ such that $\{v_{i1}, v_{i2} \ldots v_{ix}\}$ that represent a complete subtree are rooted at V_i for each i such that $0 < i < t$. Also, $\overline{V_i} \cap \overline{V_j} = \phi$ for any $i \neq j$ and $\overline{V_1} \cup \overline{V_2} \cdots \cup \overline{V_t} = \mathcal{R}$ considering that $\overline{V_i}$ is a representative group for the set of nodes rooted at V_i. Here, cover ID for nodes rooted at 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 Scheme). *The Bit Vector Scheme is a relative representation mechanism for sequential identifiers representation aimed to reduce the length of the CRL. In BVS, for a network of n nodes, a bit vector $\mathcal{S}$ n bits is constructed. In $\mathcal{S}$, the i^{th} bit indicates the validity of a key associated with the identifier i. Also, a '0' valued location in $\mathcal{S}$ represents valid or unrevoked PK and '1' represents the revoked PK. In the node's side, keys identifiers to be revoked can be extracted from offset of the 1's occurrence.*

Definition 4 (Dynamic Run-Length Encoding). *The run length encoding (a.k.a. RLE [21]) is a data encoding algorithm that maps a plain binary string $\mathcal{P} : \mathcal{P} = p_1 p_2 p_3 \ldots p_a$ into an encoded binary string $\mathcal{C} : \mathcal{C} = c_1 c_2 c_3 \ldots c_b$. Considering the instances $\xi_i, \xi_j \in \mathcal{C}$ and $\Psi_i, \Psi_j \in \mathcal{P}$, $\xi_i = \xi_j \iff \Psi_i = \Psi_j$. The notation hRLE-l includes both h and l as the header and the length that determines the prosperities of plain word W_p of length $\|W_p\|$ and the coded word W_c of length $\|W_c\|$ as follows: (a) $MAX(\|W_p\|) = 2^{l-1}$. (b) $MAX(\|W_c\|) = l$. (c) header bit of $W_p = h$. Further examples on this mapping with different h and l are shown in Fig. 2(a), Fig. 2(b), and Fig. 2(c).*

5 Naïve Versus CS Cover for Revoked ID Representation

In the CRL, if the number of the DCs to be revoked is r in a network of n nodes, the required overhead therein is $C = r \times \lceil \log n \rceil$ bits excluding other information such as signature and the CRL's attributes. In this section, we introduce the CS Cover-based schemes to reduce the representation length of the CRL.

5.1 Naïve Representation

Based on Definition 1, for a set of nodes $\mathcal{R}$ of size r in a network of n nodes, the required representation for these identifiers is $r \log_2 n$ bits. In other word, once these identifiers are to be revoked, the message in Eq. (1) is to be sent. In Eq. (1), $b_{(i)(j)}$ represents the i^{th} bit in the j^{th} revoked identifier representation.

$$\text{RL} = \left\{ \begin{matrix} b_{(1)(k_1)} & b_{(2)(k_1)} & b_{(3)(k_1)} & \cdots & b_{(n-1)(k_1)} & b_{(n)(k_1)} \\ b_{(1)(k_2)} & b_{(2)(k_2)} & b_{(3)(k_2)} & \cdots & b_{(n-1)(k_2)} & b_{(n)(k_2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{(1)(k_r)} & b_{(2)(k_r)} & b_{(3)(k_r)} & \cdots & b_{(n-1)(k_r)} & b_{(n)(k_r)} \end{matrix} \right\} \quad (1)$$

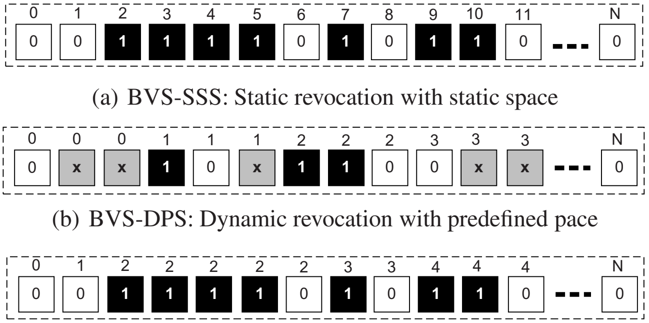
5.2 Complete Subtree in DC Revocation

The complete subtree cover concept in Definition 2 can be used directly to reduce the representation size for the list of the key's to be revoked. As an example, if the set of identifiers $\mathcal{R}$ to be revoked is $\{0100, 0101, 0110, 0111, 1010, 1011 \text{ and } 1111\}$. For the first four IDs, the furthest common parent which includes all of those IDs and no other leaf IDs is 01. For the next two, it is 101 and for the last ID, it is 1111 since it has no neighbored IDs to be revoked. Thus, The final list representation is $\{01{,}101{,}1111\}$ which includes all of the required IDs. From the simulation, CS probably reduce the overhead when r is about 5% of n but it greatly reduces the overhead when r is large enough (say, $r > 20\%$ of n). This efficeincy of reduction is shown in Fig. 5(d). Technically, additional overhead equivelent to $\log_2 n * \log_2(\log_2 n)$ is required for shortened covers' separation.

6 Bit Vector Scheme for Efficient DC Revocation

6.1 Motivation: Static Revocation with Static Space (BVS-SSS)

The bit vector which is a relative representation mechanism for the different DCs' IDs including those to be revoked can be used for CRL length reduction. In BVS, as in definition in 3, for the network of n nodes, a bit vector of length n bits in which the $i^{\mathbf{th}}$ bit indicates the validity of the DC of the sensor node with ID i is generated. For the bits of the bit vector, '0' valued location represents unrevoked DC and '1' represents the revoked DC as shown in Fig. 1(a) and the revocation list (RL) expressed in Eq. (2). Since each DC has only one revocation chance with predefined representation space, we call this scenario as the Static with Static Space (BVS-SSS).


0 1 2 3 4 5 6 7 8 9 10 11 N
0 0 1 1 1 1 0 1 0 1 1 0 0
0 0 0 1 1 1 2 2 2 3 3 3 N
0 x x 1 0 x 1 1 0 0 x x 0
0 1 2 2 2 2 2 3 3 4 4 4 N
0 0 1 1 1 1 0 1 0 1 1 0 0

(a) BVS-SSS: Static revocation with static space

(b) BVS-DPS: Dynamic revocation with predefined pace

(c) BVS-DDS: Dynamic revocation with dynamic space

Fig. 1. Different scenarios of BVS

Obviously, in the actual DC revocation systems, if a DC is revoked another DC with other ID will replace it. Therefore, in the following, we extend the BVS-SSS to cover the multi-revocation using two approaches.

$$\mathrm{RL} = b_{(0)}, b_{(1)}, \ldots, b_{(N-2)}, b_{(N-1)}. : b_{(i)} = \begin{cases} 1 & \text{If } i \text{ is revoked} \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

6.2 Dynamic Revocation with Steady Predefined Space (BVS-DPS)

In this approach, we provide each sensor node's DC with a predefined number of bits (BVS-DPS) to handle a number of revocations. For each node, this number is the same to provide an auto-separation mechanism. This procedure provide a better solution than the naïve, however, the representation has a low efficiency when the number of revoked DCs is small.

Let $b_{(i)(j)}$ be the j^{th} revocation chance for the i^{th} node, the RL general representation is shown in Eq. (3) where each bit has three possible statuses. An illustrating example is shown in Fig. 1(b).

$$\mathrm{RL} = \left\{ \begin{matrix} b_{(0)(1)} & b_{(0)(2)} & \cdots & b_{(0)(c)} \\ b_{(1)(1)} & b_{(1)(2)} & \cdots & b_{(1)(c)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{(n-1)(1)} & b_{(n-1)(2)} & \cdots & b_{(n-1)(c)} \end{matrix} \right\} : b_{(i)(j)} = \begin{cases} 0 & \text{if revoked} \\ 1 & \text{if unrevoked} \\ \text{x} & \text{if unused} \end{cases} \tag{3}$$

6.3 Dynamic Revocation with Dynamically Extendable Space (BVS-DDS)

To overcome the efficiency problem of the above representation, let us consider the BVS-SSS with a slight modification. Initially, the bit vector is initialized by '0's when no DC is revoked. Once a DC is revoked, a '1' is attached immediately before the corresponding DC's '0' offset. Based on that, one revocation adds only one bit to the bit vector. When we would like to check the validity of the i^{th} DC, firstly we find the i^{th} block that represent sensor node i and count the number of '1's which means that the $(n+1)^{\text{th}}$ DC for node i is valid. The i^{th} block for node i is composed of the i^{th} '0' in the bit vector and previous '1's before the $(i-1)^{\text{th}}$ '0'. As in the first scenario, each node can have a limited and pre-defined number of revocation chances. An example that shows how this scheme works is in Fig. 1(c).

Unlike other scenarios (i.e., BVS-SSS, BVS-DPS), BVS-DDS is fully dynamic in that it can support a infinite times of revocations where the CRL cost is typically as much as the number of revoked DCs added to an initial overhead. To introduce a dynamic naïve revocation scheme, a larger space to represent IDs is required. While 14 bits are enough to represent 10,000 DC for 10,000 with their associated PKs, 18 bits are required to provide 10 revocation chances for the same size. In BVS-DPS, an equivalent number of bits per node is required.

7 Compression: Dynamic RLE with Predefined Parameters

Technically, knowledge or even probable knowledge represented in probability of the presence for a specific pattern in the RL in BVS-SSS, BVS-DPS, BVS-DDS, CS makes it possible to use the encoding mechanism (RLE) in 4 for an efficient compression. The compression efficiency is due to a long $W_p \in \mathcal{P}$ which are replaced with shorter $W_p \in \mathcal{C}$.

W_p	W_c	$P_r(W_p)$	W_p	W_c	$P_r(W_p)$
1	0000	$p_0^0 p_1^1$	000001	0101	$p_0^5 p_1^1$
01	0001	$p_0^1 p_1^1$	0000001	0110	$p_0^6 p_1^1$
001	0010	$p_0^2 p_1^1$	00000001	0111	$p_0^7 p_1^1$
0001	0011	$p_0^3 p_1^1$	00000000	1	$p_0^8 p_1^0$
00001	0100	$p_0^4 p_1^1$			

(a)

W_p	W_c	$P_r(W_p)$	W_p	W_c	$P_r(W_p)$
0	0000	$p_0^1 p_1^0$	111110	0101	$p_0^1 p_1^5$
10	0001	$p_0^1 p_1^1$	1111110	0110	$p_0^1 p_1^6$
110	0010	$p_0^1 p_1^2$	11111110	0111	$p_0^1 p_1^7$
1110	0011	$p_0^1 p_1^3$	11111111	1000	$p_0^0 p_1^8$
11110	0100	$p_0^1 p_1^4$			

(b)

W_p	W_c	$P_r(W_p)$	W_p	W_c	$P_r(W_p)$
1	00	$p_0^0 p_1^1$	0	00	$p_0^1 p_1^0$
01	01	$p_0^1 p_1^1$	10	01	$p_0^1 p_1^1$
00	1	$p_0^2 p_1^0$	11	1	$p_0^0 p_1^2$

(c)

Fig. 2. (a) RLE with $h = 1$ and $l = 4$ (1RLE4) (b) RLE with $h = 0$ and $l = 4$ (0RLE4) (c) RLE with $h = 1, h = 0$ and $l = 2$ (1RLE4, 0RLE4)

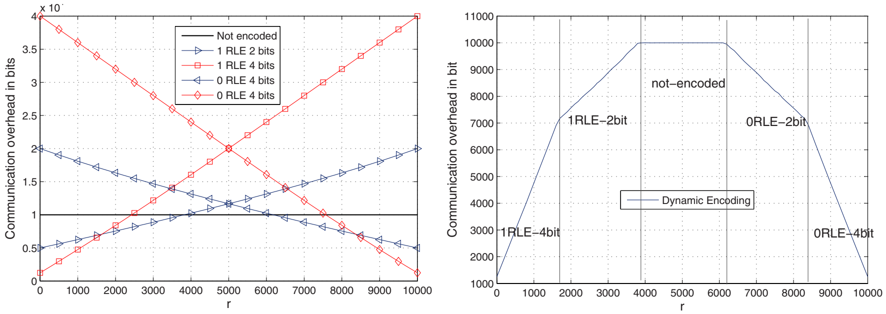
7.1 Parameters Assignment

To get a desirable performance that expresses a correlated efficiency with the number of revoked nodes r, we apply a dynamic encoding by changing l and h in the encoding algorithm. For the relatively small number of revoked DCs, both 1RLE-4b and 1RLE-2b can be used since it is high probability for long consequent zeros to appear. Similarly, for a relatively large number of revoked DCs, 0RLE-2b, 0RLE-4b can be used. Finally, for the case where the number of revoked DCs is similar to the number of unrevoked ones, it is more efficient to keep the RL not encoded. To find out the exact percents where the dynamic encoding parameters should be changed, we encode the same string represent different r as percent of n using the different possible parameters and manually calibrate the points. Fig. 3(a) shows the encoded string using different parameters where the intersections that relies below N is used. Fig. 3(b) shows the RLE for dynamically assigned percentages.

7.2 Encoding Efficiency

To find out the performance of the encoding, the probability of 1s and 0s occurrence need to be measured. Herein, we describe the analysis of efficiency for the BVS-SSS where the other two schemes follow the same analysis.

Let X be a random variable (which is in fact a Bernoulli Random Variable - BRV [22]) that describes the above occurrence of 1 and 0 in S that represents the revocation bit vector. Based on the above notation and structure of BVS-SSS in Fig. 1(a) and Eq. (2), the probability p_0 is defined as $P_r(X = 0)$ and the probability p_1 is defined as $P_r(X = 1)$. Both of these probabilities are shown in Eq. (4).



(a) RLE with different parameters

(b) Dynamic RLE

Fig. 3. (a) Different simulation running averages for the CS versus the naïve scheme (b) Dynamic RLE with dynamically chosen parameters

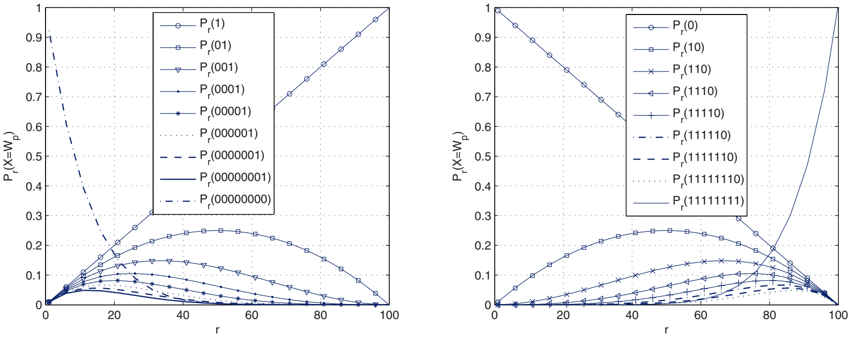
$$P_r(X=0) = p_0 = \left(\frac{n-r}{n}\right), \; P_r(X=1) = p_1 = 1 - p_0 = \left(\frac{r}{n}\right) \tag{4}$$

Note that, always $p_0 + p_1 = 1$, $0 \leq p_0 \leq 1$, and $0 \leq p_1 \leq 1$ for any n and r. In addition, any sample point with k number of 1s and $(\|W_p\| - k)$ 0s is assigned a probability $p(k, \|W_p\| - k)$ as in 5

$$p(k, \|W_p\| - k) = p_0^{(\|W_p\|-k)} p_1^k \tag{5}$$

Based on the mapping of the RLE defined earlier and the above probabilities p_0, p_1, and $p(k, \|W_p\| - k)$, the following probability samples: $P_r(X_i = 1) = (\frac{n-r}{n})^0(\frac{r}{n})^1 = p_0^0 p_1^1$, $P_r(X_i, X_{i+1} = 01) = (\frac{n-r}{n})^1(\frac{r}{n})^1 = p_0^1 p_1^1$, $P_r(X_i, \ldots, X_{i+2} = 001) = (\frac{n-r}{n})^2(\frac{r}{n})^1 = p_0^2 p_1^1$, $\ddots$, and $P_r(X_i, \ldots, X_{i+7} = 00000000) = (\frac{n-r}{n})^8(\frac{r}{n})^0 = p_0^8 p_1^0$ where i is the offset in $\mathcal{S}$ for the beginning of W_p. The above are for 1RLE-4 where other RLE encoding follows the same probability representation. From the above family of probabilities, a conclusive representation of any pattern occurrence probability is shown in Fig. 2(a), Fig. 2(b), and Fig. 2(c) for the corresponding pattern with the given W_p, k, l, and h. The resulting probability for some pattern occurrence that demonstrates the efficiency of the encoding is shown in Fig. 4(a) and Fig. 4(b). The overall performance of the dynamic encoding is dependent upon the variation of the parameters (h, l) which always guarantees a high probability for a desirable long and compressible pattern to occur.

Finally, the entropy $H(X)$ of $\mathcal{S}$ per symbol is shown in Eq. (7) which is typically a binary entropy function (as X that describes $\mathcal{S}$ is a Bernoulli Random Variable) resulting that the required bits per symbol in $\mathcal{S}$ are always less than or equal to 1. That means, based on r, there exist an algorithm (as shown in the simulation results) that is able to reduce the length of the bit vector into a shorter compressed one with an efficiency correlated with $H(X)$ in 8.



(a) $P_r(X = W_p)$ in 1RLE-4

(b) $P_r(X = W_p)$ in 0RLE-4

Fig. 4. Probability of occurrence for some pattern that determines the efficiency of encoding with the specified parameters under r number of revoked positions in $\mathcal{S}$ for a sample of $n = 100$

$$H(X) \overset{\Delta}{=} -\sum_{i=0}^{1} P_r(X = i) \log_2 P_r(X = i) \tag{6}$$

$$= -\left[\left(\frac{n-r}{n}\right) \log_2 \left(\frac{n-r}{n}\right) + \left(\frac{r}{n}\right) \log_2 \left(\frac{r}{n}\right)\right] \leq 1 \tag{7}$$

$$= [(e-1)\log_2(1-e) - e \log_2 e] \text{ for some } e : 0 < e \leq 1, e = r/n \tag{8}$$

8 Simulation Results and Analysis

In this section, we justify our schemes' performance by simulating the following schemes: naïve, naïve Encoded, CS, BVS-SSS, BVS-DDS, BVS-DPS. To handle the randomness of the revoked DC ID, we use a random identifier selector that indicates the current compromised ID from the non-compromised pool. $n = 10,000$ sensor nodes and $c = 10$.

Simulation results: The dynamic encoding altering points to change the encoding parameters are calibrated using the intersections of the RLE encoding using different length and heading parameters as of Fig. 3(a) and Fig. 3(b). The communication overhead of the BVS-DDS versus the naïve is shown in Fig. 5(a) and Fig. 5(b). Note that, our BVS-DDS scheme provides a high efficiency since it does not include any non-required bits. Fig. 5(c) shows the resulting performance of the CS in which we performed the simulation on different random samples (i.e., 5 for each) and considered the average (AVG $= \frac{1}{5}\sum_{i=1}^{5}$ CS Overhead$_i$). To show the random behavior of the CS, we executed our simulator for the three times in which the average of 5 times is considered. Note that the performance of the CS provides a higher efficiency when r is large enough.

To show the regions in which the naïve solution provides a better performance than the BVS (i.e. $r \leq \frac{n^{'}}{\lg n-1}$ in case of BVS-DPS and BVS-DDS), we executed the simulator for different network sizes. Fig. 5(d) shows that the BVS provides a better performance than the naïve when r is greater than 7% of n. Moreover, additional numerical results are shown in Table 1.

Table 1. Communication overhead in bit for revoking different percents of network size using different schemes. $N = 10000$ nodes, $-_{\mathbf{C}}$ indicates the usage of RLE.

Scheme	01%	05%	10%	20%	40%	50%
Naïve	1,400	7,000	14,000	28,000	56,000	70,000
Naïve$_{\mathbf{C}}$	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-DPS	1,800	9,000	18,000	36,000	72,000	90,000
Naïve-DDS	18,000	90,000	180,000	360,000	720,000	900,000
BVS-SSS	10,000	10,000	10,000	10,000	10,000	10,000
BVS-SSS$_{\mathbf{C}}$	1,597	2,994	4,753	7,569	10,000	10,000
BVS-DPS	10,100	10,500	11,000	12,000	14,000	15,000
BVS-DDS	11,000	15,000	20,000	30,000	50,000	60,000
BVS-DDS$_{\mathbf{C}}$	3,756	13,778	20,000	27,508	37,498	41,320

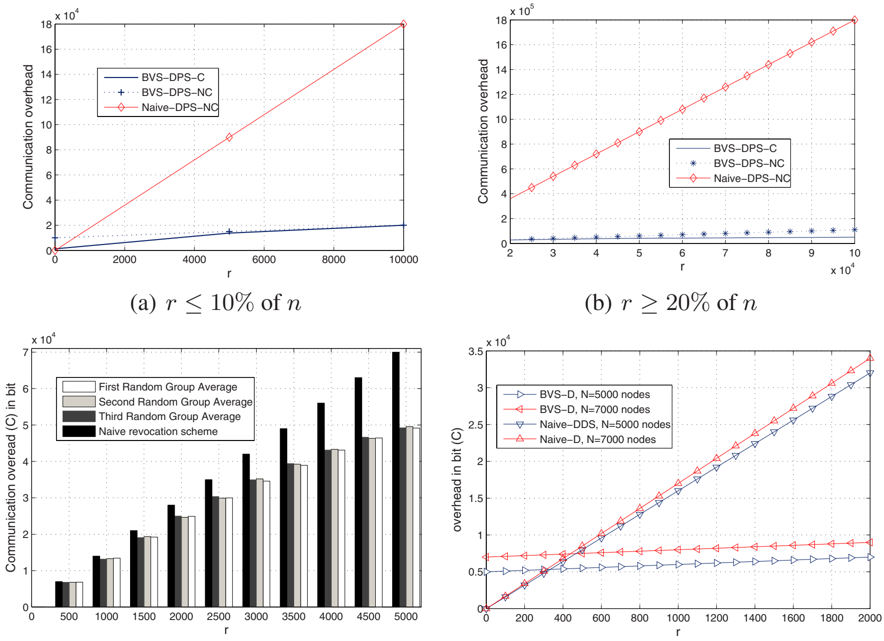
Analysis: For the part of the BVS-SSS, BVS-DPS, and BVS-DDS, the dynamic encoding algorithm with h, l parameters on intervals provides a high efficiency based the early discussed probability of pattern occurrence. On the other hand, due to the non-systematic occurrence of similar bits in both naïve and CS algorithm, the RLE may not provide a high efficiency. That means, applying the dynamic RLE for the CS or the naïve will replace a small string in the $\mathcal{P}$ with a longer ones $\mathcal{C}$ with high probability.

Another notable feature is that the CS algorithm provides a probabilistic communication reduction. If the set of DCs to be revoked is in a consequent order with fewer gaps, CS will provide a high representation efficiency and reduction in the overall communication. Otherwise, the communication will be greater. In the worst case, it will be the same like the naïve representation. Table 1 shows a numerical results that consider a uniformly random distribution for the compromised IDs of DCs. The efficiency is dependent on the random manner of the compromising and the used RLE parameters.

For the BVS-DDS, the communication overhead is $f_r = r + n$ bit where n is the real network size which is equivelant to the initial overhead in bits. In the case of Dynamic naïve solution, the required overhead is $f_r = r * \lg n^{'}$ where $n^{'}$ herein is the expanded space that permits a multiple revocation for a given ID as discussed earlier

Table 2. Overhead Comparison

Scheme	**Communication Overhead**	**Scheme**	**Communication overhead**
Naïve	$r \log_2 n$	**CS**	$r \log_2 \frac{n}{r} + (\log_2 n)(\log_2 \log_2 n)$
BVS-SSS	n	**Naïve-DPS**	$r \log_2 n^{'} = r \log_2 cn$
BVS-DPS	$n^{'} = cn$	**Naïve-DDS**	$r \log_2 n^{'} = r \log_2 cn$
BVS-DDS	$n + r$		



(a) $r \leq 10\%$ of n

(b) $r \geq 20\%$ of n

(c) CS versus the naïve scheme

(d) Naïve DPS versus BVS-DPS

Fig. 5. The communication overhead of our schemes with different scenarios

(i.e., $n^{'} = c * n$ where c is the possible revocation chances for a DC). BVS-SSS provides a better efficiency than the naïve scheme for $r \geq \frac{n^{'}}{\lg n^{'} - 1}$ even without any compression. For the CS communication overhead is, at the worst case, $r \log_2 \frac{n}{r}$ bits [20] in addition to the sets separation bits which we discussed earlier. A concluding overhead comparison is shown in Table 2.

9 Conclusion and Future Works

We introduced two schemes for communication efficient DC revocation in WSN. The first one relies on the complete subtree and the second is bit vector scheme. Our solutions showed a relevant reduction in the communication. The CS provides high probability for reduction in normal cases. The upper bound for the BVS is constant and depends initially on the networks size. Using encoding schemes like RLE can be behind a more reduction in the communication overhead. We tried different scenario of dynamic and static BVS. Trying other encoding/compression schemes and finding other applications for the bit vector representation and/or the CS might be future work. The deployment knowledge as a direct communication reduction method for the communication overhead is valuable to be studied.

References

1. Deng, J., Han, R., Mishra, S.: Defending against path-based dos attacks in wireless sensor networks. In: SASN, pp. 89–96 (2005)
2. Chan, H., Perrig, A., Song, D.X.: Random key predistribution schemes for sensor networks. In: IEEE Symposium on Security and Privacy, p. 197 (2003)
3. 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)
4. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: ACM CCS, pp. 41–47 (2002)
5. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: ACM CCS, pp. 52–61 (2003)
6. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key crypto-systems. CACM 26, 96–99 (1983)
7. Koblitz, N., Menezes, A., Vanstone, S.A.: The state of elliptic curve cryptography. Des. Codes Cryptography 19, 173–193 (2000)
8. Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C.: Comparing elliptic curve cryptography and rsa on 8-bit cpus. In: CHES, pp. 119–132 (2004)
9. Malan, D.J., Welsh, M., Smith, M.D.: A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In: First IEEE Int. Conf. on Sensor and Ad Hoc Comm. and Networks, pp. 71–80 (2004)
10. Wander, A., Gura, N., Eberle, H., Gupta, V., Shantz, S.C.: Energy analysis of public-key cryptography for wireless sensor networks. In: PerCom, pp. 324–328 (2005)
11. Watro, R.J., Kong, D., fenCuti, S., Gardiner, C., Lynn, C., Kruus, P.: Tinypk: securing sensor networks with public key technology. In: SASN, pp. 59–64 (2004)
12. Du, W., Wang, R., Ning, P.: An efficient scheme for authenticating public keys in sensor networks. In: MobiHoc, pp. 58–67 (2005)
13. Nyang, D., Mohaisen, A.: Cooperative public key authentication protocol in wireless sensor network. In: Ma, J., Jin, H., Yang, L.T., Tsai, J.J.-P. (eds.) UIC 2006. LNCS, vol. 4159, pp. 864–873. Springer, Heidelberg (2006)
14. Pottie, G.J., Kaiser, W.J.: Wireless integrated network sensors. Commun. ACM 43, 51–58 (2000)
15. Housley, R., Polk, W., Ford, W., Solo, D.: Rfc 3280: Internet x.509 public key infrastructure: Certificate and certificate revocation list (crl) profile (2002)
16. Levis, P., Madden, S., Gay, D., Polastre, J., Szewczyk, R., Woo, A., Brewer, E.A., Culler, D.E.: The emergence of networking abstractions and techniques in tinyos. In: NSDI, pp. 1–14 (2004)
17. Ning, P., An Liu, P.K.: Tinyecc: Elliptic curve cryptography for sensor networks (version 0.3), software package (2007)
18. Merkle, R.C.: Protocols for public key cryptosystems. In: IEEE S&P, pp. 122–134 (1980)
19. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
20. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
21. Golomb, S.W., Peile, R.E., Scholtz, R.A.: Basic Concepts in Information Theory and Coding: The Adventures of Secret Agent 00111. Springer, Heidelberg (1994)
22. Trivedi, K.S.: Probability and Statistics with Reliability, Queuing and Computer Science Applications. John Wiley and Sons Inc, New York, USA (2001)