Dong JW, Pei DY, Wang XL. A class of key predistribution schemes based on orthogonal arrays. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 23(5): 825–831 Sept. 2008

A Class of Key Predistribution Schemes Based on Orthogonal Arrays

Jun-Wu Dong[1,2] (董军武), Ding-Yi Pei[2] (裴定一), and Xue-Li Wang[3] (王学理)

[1] *College of Mathematics and Econometrics, Hunan University, Changsha 410082, China*

[2] *Institute of Information Security, Guangzhou University, Guangzhou 510006, China*

[3] *School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China*

E-mail: {djunwu1971, wangxuyuyan}@yahoo.com.cn; gztcdpei@scut.edu.cn

Received August 16, 2007; revised May 20, 2008.

Abstract Pairwise key establishment is a fundamental security service in sensor networks; it enables sensor nodes to communicate securely with each other using cryptographic techniques. In order to ensure this security, many approaches have been proposed recently. One of them is to use key predistribution schemes (KPSs) by means of combinatorial designs. In this paper, we use the Bush's construction of orthogonal arrays to present a class of key predistribution schemes for distributed sensor networks. The secure connectivity and resilience of the resulting sensor network are analyzed. This KPS constructed in our paper has some better properties than those of the existing schemes.

Keywords sensor network, key predistribution scheme, combinatorial design, orthogonal array, Bush's construction

1 Introduction

Distributed Sensor Networks (DSNs) have received a lot of attention recently due to their wide application in military as well as civilian operations. In the applications, we often accept the following assumptions on the model of DSNs: 1) many sensor nodes are dropped, in a random way, to the target area, so the network topology is unknown before the deployment; 2) the sensor nodes are typically low-cost, battery powered, and highly resource constrained, hence they should consume as little power as possible; 3) the sensor nodes have limited computation, storage, and communication capabilities, so that they can communicate with nodes only within a limited radius. We assume that the radio coverage area of each sensor node forms a circle of a fixed radius whose center is that node. We call this circle the neighborhood of the given sensor node. Once the sensor nodes are deployed, they scan their neighborhoods and find out their neighbors. The typical parameters of a sensor node, for example, are as follows: the size is 58mm × 47mm × 40mm, the weight is 35g (115g with batteries), and the node has a 16-bit microcontroller with 2Kbyte RAM, 60Kbyte flash–ROM, and 64Kbyte EEPROM.

In wireless distributed sensor networks, it is important for sensor nodes to communicate securely with each other. Of course, public key infrastructure (PKI) can be used to establish pairwise secret keys between sensor nodes. However, the operations, which are based on the complex arithmetic of big integers, have to be implemented in the low-level environments. Very recently, implementations of ECC and RSA in low-level environments such as 8-bit CPUs have been proposed in a reasonable timing. One may refer to [1–4], for example, for detailed results. With the improvement of hardware technique and the optimization of algorithm theory, it is possible to implement public key cryptography systems in such low-level environments. Yet, by now, it is not suitable to use PKI due to its expensive computational cost as well as storage consumption in each sensor node. Therefore it is natural to use the key predistribution schemes (KPSs), where a set of secret keys is installed in each node before the sensor nodes are deployed. If two adjacent sensor nodes have at least one common key, they can select it as the secret key and communicate securely by means of symmetric cryptography.

In general, a key predistribution scheme consists of three phases: key predistribution, shared key discovery, and path key establishment. First, a large pool of keys is specified, and each key is assigned a unique identifier. Then, every sensor node is loaded with a fixed number of keys chosen from the key pool, along with their key identifiers. After the deployment of the DSN, the shared key discovery phase comes, where any

Regular Paper

This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 60473017, 90604034 and 10771078.

two nodes in wireless communication range exchange their list of key identifiers to each other, and look for their common keys. If they share one or more common keys, they can pick it or one of them as their secret key/keys for cryptographic communication. The path key establishment phase works if there is no common key between a pair of nodes which need to have cryptographic communication. We call a successive sequence of nodes a path, where any two adjacent nodes (also in the radio coverage range) have at least one common key. If the sensor node i wants to communicate securely with the sensor node j, it needs to find a path between itself and the sensor node j. Thus messages from the sensor node i can reach the sensor node j securely.

In [5], Eschenauer and Gligor proposed a probabilistic key predistribution scheme. The main idea was to assign every sensor node randomly a set of keys from the given pool of keys before deployment, so any two sensor nodes have a certain probability of sharing at least one common key. Extensions and variations of this approach can be found in [6–8].

In order to construct deterministic key predistribution schemes for DSN, using combinatorial design is another strategy in this area. This idea was first proposed in Çamtepe and Yener[9]. Further study in this context can be found in [10–12].

A combinatorial design is a pair of sets $(X, \mathscr{B})$, where $X = \{x_1, x_2, \ldots, x_v\}$ is a finite set, the elements of which are called points, and $\mathscr{B} = \{B_1, B_2, \ldots, B_b\}$ is a finite set of subsets of X, called blocks. And the blocks of $\mathscr{B}$ satisfy some special intersectional properties.

Any combinatorial design can be used to establish a key predistribution scheme for a DSN. Let $X = \{x_1, x_2, \ldots, x_v\}$ and $\mathscr{B} = \{B_1, B_2, \ldots, B_b\}$, where each block B_j has k points of X. Let the sensor nodes be denoted by $N_1, N_2, \ldots, N_b$. For every $1 \leqslant i \leqslant v$, a key K_i is chosen randomly from some special key space. Hence there exists a 1-1 correspondence between X and $\{K_i \,|\, 1 \leqslant i \leqslant v\}$, and the symbol x_i can be called the label of K_i. Then for $1 \leqslant j \leqslant b$, the sensor node N_j receives the set of keys $\{K_i \,|\, x_i \in B_j\}$, that is, the block B_j is used for specifying which keys are given to the node N_j. Thus each node receives k keys.

The following two probabilities can be used to describe the property of a sensor network. 1) The connective probability p, it is defined by the probability that any pair of sensor nodes shares a link, i.e., the nodes of a pair have at least one common key. Suppose $B \in \mathscr{B}$ is a block, it is easy to see that the connective probability p, which is independent of the specified block B by the symmetry of combinatorial designs, can be defined as follows:

$$p = \frac{\#\{B' \in \mathscr{B} \,|\, B' \cap B \neq \emptyset\}}{b-1}.$$

Of course, this probability is expected to be as large as possible, since it measures the effectiveness of the sensor network. 2) The probability $\mathit{fail}(1)$. If a sensor node is detected as being compromised, then all the keys it possesses should no longer be used by any node in the sensor network. Suppose the sensor nodes N_i and N_j have at least one common key (which means that there is a link between the pair of N_i and N_j). If all the common keys of the pair of N_i and N_j are contained in the compromised sensor node, then N_i and N_j no longer communicate directly, i.e., the link between N_i and N_j is lost. And the probability of links being affected is defined as

$$\mathit{fail}(1) = \frac{\text{the lost connectivities}}{\text{the original connectivities}}.$$

Generally, we expect $\mathit{fail}(1)$ to be as small as possible, since it measures the resilience of the sensor network, when a random sensor node is compromised.

Çamtepe and Yener[9], constructed a key predistribution scheme for sensor network using finite geometry over projective planes. In [11], Lee and Stinson provided a key predistribution scheme by using a special combinatorial design $\mathrm{TD}(k, N)$. In this scheme, any two sensor nodes share at most one common key.

In [13], Dong *et al.* construct a key predistribution scheme based on 3-design, where the connective probability $p \to 1/2$ and $\mathit{fail}(1) \to 0$ as $b \to \infty$.

In this paper, we shall construct a class of key predistribution schemes by means of a special type of orthogonal arrays, i.e., the orthogonal arrays generated by the Bush's construction. The main contributions of this paper are summarized as follows.

1) We propose a technique that can be used to construct deterministic KPSs from general combinatorial designs.

2) We construct a new class of key predistribution schemes. The most attractive feature of this key predistribution scheme is that when the number of sensor nodes b tends to ∞, we have p tending to some certain limit value $Pr(t)$ for every t, and $\mathit{fail}(1) \to 0$, hence we have various choices for different DSN.

3) We provide two theorems (Theorems 3.2 and 3.3), which are useful in deriving the formulas of the connective probability p and $\mathit{fail}(1)$ of the resulting key predistribution schemes for DSN.

The rest of this paper is arranged as follows. In Section 2, as the preliminaries, we discuss how to construct key predistribution designs for DSNs by use of orthogonal arrays, and how to compute the connective probability p and *fail*(1) of the resulting schemes by means of the properties of the corresponding combinatorial designs. The key predistribution scheme based on the Bush construction of orthogonal arrays will be presented in Section 3, and also the connective probabilities p and *fail*(1) of the scheme will be computed in this section. Some issues on implementation will be given in Section 4. We compare the schemes of this paper with some other known schemes in Section 5. Finally, we give the sketch of the proof of Theorems 3.2 and 3.3 in Appendix.

2 Preliminaries

Suppose that $(X, \mathscr{B})$ is a combinatorial design and t an positive integer, with the following properties. For every $1 \leqslant r \leqslant t-1$, any r-subset of X either occurs in exactly λ_r blocks of $\mathscr{B}$ or does not occur in any block of $\mathscr{B}$, and $\lambda_t = 1$. Such combinatorial designs can be used to construct key predistribution schemes for DSNs as described in the introduction. We can easily compute the connective probability p and *fail*(1).

Suppose $C \in \mathscr{B}$ is a block, and $\mu'_C(r)$ the number of blocks in $\mathscr{B}$ which intersects with C at some r-set for $1 \leqslant r \leqslant t-1$.

We have $\mu'_C(t-1) = \lambda_{t-1} - 1$, since $\lambda_t = 1$. It is easy to see that the following recursion relation holds for $1 \leqslant r \leqslant t-1$:

$$\mu'_C(r) = \lambda_r - 1 - \sum_{s=1}^{t-1-r} \binom{k-r}{s} \mu'_C(r+s), \tag{1}$$

from which we can calculate the number $\mu'_C(t-2)$, $\mu'_C(t-3), \ldots, \mu'_C(2), \mu'_C(1)$ successively.

Then we can calculate the connective probability p and *fail*(1) as the following:

$$p = \frac{\mu_C}{b-1}. \tag{2}$$

$$\mathit{fail}(1) = \frac{\sum_{r=1}^{t-1} \binom{k}{r} \lambda_r \mu'_C(r)}{b \sum_{r=1}^{t-1} \binom{k}{r} \mu'_C(r)}, \tag{3}$$

where

$$\mu_C = \sum_{r=1}^{t-1} \binom{k}{r} \mu'_C(r).$$

An orthogonal array of size N, with k constraints, s levels, strength t, and index λ, denoted $OA(N, k, s, t)$, is an $N \times k$ array with entries from a set S of $s \geqslant 2$ symbols, having the property that in every $N \times t$ submatrix, every $1 \times t$ row vector appears the same number of λ times.

It is easy to see that $\lambda = N/s^t$. It is customary to say that the orthogonal array has index unity when $\lambda = 1$. In this paper, we only consider the orthogonal arrays of index unity, and hence we have $N = s^t$. For more constructions and applications of orthogonal arrays one can refer to the book, Orthogonal Arrays, Theory and Application by A.S. Hedayat, N.J.A. Sloane, and John Stufken[17].

Orthogonal arrays can be used to construct combinatorial designs. Regard symbols in different columns as different points in X, hence X has $v = ks$ elements; and take each row of the orthogonal array as block, so that each block has k elements. Therefore, we get a combinatorial design $(X, \mathscr{B})$, where $b = |\mathscr{B}| = N$.

3 New Scheme

Bush[15] provided a construction of orthogonal arrays of index unity with strength $t \geqslant 2$. We state it as follows.

Theorem 3.1. *If $q > 2$ is a prime power, then an $OA(q^t, q+1, q, t)$ of index unity exists whenever $q \geqslant t-1 \geqslant 0$.*

This orthogonal array can be used to construct a combinatorial design $(X, \mathscr{B})$, with $v = |X| = q(q+1) = q^2 + q$, $b = q^t$. It is easy to verify that this combinatorial design has the following special properties: every r-subset of X either occurs together exactly in $\lambda_r = q^{t-r}$ blocks or does not occur in any block of $\mathscr{B}$ for $1 \leqslant r \leqslant t$.

The correspondences between the parameters of a combinatorial design $(X, \mathscr{B})$ and the related key predistribution scheme for a DSN are summarized in Table 1.

Table 1. Parameters for Some t

KPS for a Distributed Sensor Network	Combinatorial Design	Parameter
Key Pool	Point Set X	
Sensor Nodes	Blocks	
Network Size	Number of Blocks	$b = q^t$
Size of Key Pool	Number of Points	$v = q^2 + q$
Number of Keys per Node	Block Size	$k = q + 1$

Now, we study the connective probability p of the KPS constructed by the above combinatorial design. Denote $Pr(t)$, the limit of p as $q \to \infty$, since this probability is relative to t.

We consider λ_r and $\mu'_C(r)$ $(1 \leqslant r \leqslant t)$ as polynomials of q, and by some more tedious and complex calculation, we can prove the following two theorems (we provide the sketch of the proofs in Appendix):

Table 2. Parameters for Some t

	p_3	p_4	p_5	p_6	f_3	f_4	f_5	f_6
$q = 11$	0.586 466	0.699 454	0.678 982	0.681 591	0.020 979	0.060 886	0.052 424	0.053 777
$q = 13$	0.573 770	0.694 118	0.671 019	0.674 219	0.015 385	0.052 252	0.043 985	0.045 441
$q = 17$	0.557 003	0.687 356	0.660 540	0.664 642	0.009 288	0.040 770	0.033 219	0.034 710
$q = 29$	0.533 869	0.678 543	0.646 121	0.651 711	0.003 337	0.024 635	0.019 085	0.020 346
$q = 59$	0.516 803	0.672 410	0.635 487	0.642 371	0.000 834	0.012 402	0.009 219	0.010 012
$q = 79$	0.512 577	0.670 939	0.632 852	0.640 084	0.000 469	0.009 319	0.006 853	0.007 481
$q = 109$	0.509 132	0.669 753	0.630 703	0.638 226	0.000 248	0.006 788	0.004 947	0.005 424
$q = 139$	0.507 168	0.669 082	0.629 477	0.637 170	0.000 153	0.005 338	0.003 870	0.004 255

Table 3. Experimental Results of p

q	29	59	79	109	139	179
$T = 1000$	0.677 047	0.674 985	0.672 533	0.670 681	0.668 899	0.666 657
$T = 2000$	0.679 295	0.674 217	0.668 439	0.671 441	0.669 170	0.669 315
$T = b$	0.678 543	0.672 410	0.670 939	0.669 753	0.669 082	0.668 539

Theorem 3.2. *Suppose that $t \geqslant 2$ is an integer, we have*

$$\begin{aligned} Pr(t) &= 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4!} + \cdots + \frac{(-1)^t}{(t-1)!} \\ &\to 1 - \frac{1}{e} \cong 0.632121, \quad \textit{when} \quad t \to \infty. \end{aligned} \tag{4}$$

where $e = 2.7182818\cdots$ is the constant of natural logarithm.

Theorem 3.3. 1) $\lim_{q\to\infty} \mathit{fail}(1) = 0$.

2) *$p/\mathit{fail}(1) \cong O(q^2)$ if $t = 3$, and $p/\mathit{fail}(1) \cong O(q)$ otherwise, where $O(q^2)$ and $O(q)$ means no more than some certain constant multiple of q^2 and q respectively.*

Example 1. For $t = 2$, 3, 4, 5, 6, we compute the explicit formulas of the connective probability p and $\mathit{fail}(1)$ as follows:

	p	$\mathit{fail}(1)$
$t = 2$	1	$\frac{1}{q}$
$t = 3$	$\frac{q^2+3q+2}{2(q^2+q+1)}$	$\frac{3}{q(q+2)}$
$t = 4$	$\frac{2q^2+q+3}{3q^2+3}$	$\frac{3q^3-3q^2+13q-1}{4q^4+2q^3+6q^2}$
$t = 5$	$\frac{5q^4+10q^3+7q^2+10q+8}{8(q^4+q^3+q^2+q+1)}$	$\frac{8q^5+18q^4-26q^3+73q^2-15q+2}{15q^6+15q^5+6q^4+24q^3}$
$t = 6$	$\frac{19q^4+16q^3+19q^2+6q+30}{30(q^4+q^2+1)}$	$\frac{45q^7+30q^6+145q^5-230q^4+511q^3-186q^2+51q-6}{4(19q^8+16q^7+19q^6+6q^5+30q^4)}$

Table 2 lists the connective probability p and $\mathit{fail}(1)$ for some q, and $t = 3$, 4, 5, 6, where the subscript denotes the value of t.

4 Implementation

Suppose that q is a prime in this section. The size of q is determined by the number of keys $k = q + 1$ per node. Since the elements from different columns of the orthogonal array are considered as different elements of X, we can denote X to be

$$X = \{a_{ij} \mid 0 \leqslant i \leqslant q - 1, \quad 0 \leqslant j \leqslant q\}.$$

To get a block B of $\mathscr{B}$, we generate a polynomial $\phi(x)$ of degree $t-1$, compute $y_i = \phi(i)$ for $0 \leqslant i \leqslant q-1$, and denote the coefficient of x^{t-1} in ϕ by y_q, and then we have

$$B = \{a_{y_0,0}, a_{y_1,1}, \ldots, a_{y_{q-1},q-1}, a_{y_q,q}\}.$$

If we generate all the polynomials of degree $t - 1$ in $\mathbb{F}_q[x]$, we get all the blocks $\mathscr{B}$ of the DSN.

We may find all the blocks $\mathscr{B}$ by searching if q is not too large. When q grows larger, the number of blocks $b = q^t$ in $\mathscr{B}$ may become too large for application. In this case we can only use a part of the blocks chosen randomly from $\mathscr{B}$. If we take $t = 4$, for example, then

Table 4. Experimental Results of $fail(1)$

q	29	59	79	109	139	179
$T = 1000$	0.024 499	0.012 488	0.009 423	0.006 846	0.005 380	0.004 112
$T = 2000$	0.024 730	0.012 464	0.009 287	0.006 795	0.005 345	0.004 154
$T = b$	0.024 635	0.012 402	0.009 319	0.006 788	0.005 338	0.004 155

the experiments (Tables 3, 4, where T is the number of blocks) show that the parameters p and $fail(1)$ have only a small disturbance when only a part of blocks is used.

After the deployment of the distributed sensor network, any two nodes in the wireless communication range exchange their list of key identifiers to each other, and look for their common keys. If they have common keys, they can pick one of them as their secret key for cryptographic communication.

5 Comparisons

In [9], Çamtepe and Yener constructed a key predistribution scheme for sensor network by use of finite geometry over projective planes. Let q be a prime power, X the projective plane $PG(2, \mathbb{F}_q)$, $\mathscr{B}$ the set of projective lines in $PG(2, \mathbb{F}_q)$, then it is easily seen that $(X, \mathscr{B})$ is a $2 - (q^2 + q + 1, q + 1, 1)$ design. Each pair of lines has exactly one common point, so the connective probability $p = 1$. However, sine $b = |\mathscr{B}| = q^2 + q + 1$, the number of keys per node is $k = q + 1 \approx \sqrt{b}$. When the size of DSN is large, it may be impossible for its heavy storage requirement.

For example, suppose that we want to construct a key predistribution scheme, by the Çamtepe and Yener's method, for a DSN having 1000000 nodes. Then the smallest prime power q such that $q^2 + q + 1 \geqslant 1000000$ is $q = 1009$. The resulting KPS would assign 1010 keys to every node.

In our schemes, let $t = 4$, the smallest prime q such that $b = q^4 \geqslant 1000000$ is that $q = 37$. If we take $q = 37$, then the resulting key predistribution scheme for a DSN can support more than 1000000 sensor nodes and each node stores $k = q + 1 = 38$ keys, which is much less than that in Çamtepe and Yener's scheme.

In [11], Lee and Stinson constructed a key predistribution scheme by using transversal design $\mathrm{TD}(k, N)$.

Let $k \geqslant 2$ and $N \geqslant 1$. A transversal design $\mathrm{TD}(k, N)$ is a triple $(X, \mathscr{B}, \mathscr{G})$ such that the following properties are satisfied: 1) X is a set of kN elements called points; 2) $\mathscr{G}$ is a partition of X into k subsets of size N called groups; 3) $\mathscr{B}$ is a set of k-subsets of X called blocks; 4) any group and any block contain exactly one common point; and 5) every pair of points from distinct groups is contained exactly in one block. For further introduction, or construction to transversal designs, one can refer to [14, 16].

A class of transversal design $\mathrm{TD}(k, N)$, where N is a prime and $k < N$, was constructed in [11]. In the resulting scheme, every two sensor nodes share at most one common key, the number of nodes $b = N^2$, the connective probability $p' = k/(N + 1)$ and $Fail(1) = (N - 2)/(N^2 - 2)$. We compare the scheme based on this class of $\mathrm{TD}(k, N)$ with the scheme of this paper with $t = 3$. For a given prime power q, let $k = q + 1$ and N be the largest prime such that $N^2 \leqslant q^3$. The comparison between these two schemes is given in Table 5.

Table 5 shows that when these two schemes have the same number k of keys per node and approximately the same number b of nodes, the scheme of this paper has greater connective probability p and lower $fail(1)$.

In [13], Dong *et al.* constructed a key predistribution scheme based on 3-design. Table 6 gives some basic parameters of this scheme, where b_2, p'' and $FAIL(1)$ denote the corresponding parameters of the scheme in [13].

From Example 1, we know that when $t = 2$, our scheme has the maximal connective probability $p = 1$, which is similar to that of the scheme in [9]. And from Tables 2 and 6, we know that, when $t = 3$, our scheme has the similar parameters as that of the scheme in [14].

Table 5. Comparison of the Case $t = 3$ and $\mathrm{TD}(k, N)$

q	b	v	k	p_3	f_3	N	p'	$Fail(1)$
11	1 331	132	12	0.586 466	0.020 979	31	0.375 000	0.030 240
13	2 197	182	14	0.573 770	0.015 385	47	0.318 182	0.022 198
17	4 913	306	18	0.557 003	0.009 288	67	0.264 706	0.014 486
29	24 389	870	30	0.533 869	0.003 337	151	0.197 368	0.006 535
59	205 379	3 540	60	0.516 803	0.000 834	449	0.133 333	0.002 217
79	493 039	6 320	80	0.512 577	0.000 469	701	0.113 960	0.001 422
109	1 295 029	11 990	110	0.509 132	0.000 248	1 129	0.097 345	0.000 884
139	2 685 619	19 460	140	0.507 168	0.000 153	1 637	0.085 470	0.000 610

Furthermore, by selecting properly parameter t, the scheme in this paper may have better connective probability p or lower $fail(1)$ than that of [9], and [13].

Table 6. Some Parameters from [13]

q	b_2	p''	$FAIL(1)$
11	1 342	0.630 872	0.021 625
13	2 210	0.611 589	0.015 788
17	4 930	0.586 123	0.009 475
29	24 418	0.551 050	0.003 376
59	205 438	0.525 271	0.000 838
79	493 118	0.518 903	0.000 471
109	1 295 138	0.513 718	0.000 249
139	2 685 758	0.510 765	0.000 153

6 Conclusions

In this paper, we construct a class of key predistribution schemes by using the Bush construction of orthogonal arrays. Compared with the schemes from [11], our scheme has higher connective probability and lower $fail(1)$. And compared with schemes from [13], by selecting properly parameter t, our scheme can have higher connective probability p. Furthermore, our schemes can provide various choices for different DSN's, since there are various choices of t.

Acknowledgment The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] Gura N, Patel A, Wander A, Eberle H, Shantz S C. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Proc. CHES*, Cambridge, Boston, USA, *LNCS* 3156, 2004, pp.119–132.

[2] Dong J, Zou H, Pei D. The design and implementation of ECC card. *Journal of Computer Applications*, 2005, 25(11): 2549–2553. (In Chinese)

[3] Adam D W, Daniel V B, Christof P. Elliptic curve cryptography on smart cards without coprocessors. In *Proc. the Fourth Smart Card Research and Advanced Applications Conference*, Bristol, UK, 2000, pp.71–92.

[4] Guajardo J, Blümel R, Krieger U, Paar C. Efficient implementation of elliptic curve cryptosystems on the TI MSP 430x33x family of microcontrollers. In *Proc. PKC2001*, *LNCS* 1992, 2001, pp.365–382.

[5] Eschenauer L, Gligor V B. A key management scheme for distributed sensor networks. In *Proc. the 9th ACM Conference on Computer and Communications Security*, Washington DC, USA, 2002, pp.41–47.

[6] Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks. In *Proc. the IEEE Symposium on Security and Privacy*, Washington DC, 2003, pp.197–213.

[7] Du W, Deng J, Han Y, Varsheney P. A pairwise key predistribution scheme for wireless sensor networks. In *Proc. the 10th ACM Conference on Computer and Communications Security* (*CCS*), Washington DC, USA, October 2003, pp.42–51.

[8] Liu D, Ning P. Establishing pairwise keys in distributed sensor networks. In *Proc. the 10th ACM Conference on Computer and Communications Security* (*ACMCCS*), Washington DC, USA, 2003, pp.52–61.

[9] Çamtepe S A, Yener B. Combinatorial design of key distribution mechanisms for wireless sensor networks. Technical Report TR–04–10, RPI Dept. Computer Science, April 2004.

[10] Lee J, Stinson D R. Deterministic key predistribution schemes for sensor networks. In *Proc. SAC*, Nicosia, Cyprus, *Lecture Notes in Computer Science*, 3357, 2004, pp.294–307.

[11] Lee J, Stinson D R. A combinatorial approach to key predistribution for distributed sensor networks. In *Proc. IEEE Wireless Computing and Networking Conference* (*WCNC 2005*), New Orleans, LA, USA, 2005, pp.1200–1205.

[12] Wei R, Wu J. Product construction of key distribution schemes for network. In *Proc. SAC 2004, Lecture Note in Computer Science*, 3357, Springer, 2005, pp.280–293.

[13] Dong J, Pei D, Wang X. A key predistribution scheme based on 3-designs. In *Proc. Inscrypt 2007*, *LNCS* 4990, 2008, pp.81–92.

[14] Pei D. Authentication Codes and Combinatorial Designs. Chapman & Hall / CRC, 2006.

[15] Bush K A. Orthogonal arrays of index unity. *Annals of Mathematical Statistics*, 1952, 23: 426–434.

[16] Street A P, Street D J. Combinatorics of Experimental Design. Oxford: Clarendon Press, 1987.

[17] Hedayat A S, Sloane N J A, Stufken J. Orthogonal Arrays, Theory and Application. Springer, 1999.

Jun-Wu Dong is a Ph.D. candidate majoring in cryptology at Hunan University. Now he is an associate professor in the College of Mathematics and Information Sciences at Guangzhou University. His current research interests include: wireless sensor network, combinatorial design and Boolean function.

Ding-Yi Pei graduated in mathematics from the University of Science and Technology of China for undergraduate study and graduate study in 1964 and 1967, respectively. In 1977∼1986, he was an associate researcher in the Institute of Applied Mathematics, Chinese Academy of Science. In 1986∼2006, he is a professor of the Mathematics and Information Science Institute at Guangzhou University. Since 2007, he is the president of the Chinese Association for Cryptologic Research. His research area covers number theory and cryptology.

Xue-Li Wang is a professor of Mathematical College of the South China Normal University. His research area covers number theory, modular form and cryptology.

Appendix. Proofs of Theorems 3.2 and 3.3

Proof of Theorem 3.2 (*Sketch*). Suppose $t \geqslant 2$ is an integer, we consider $\lambda_r = q^{t-r}$ and $\mu'_C(r)$ $(1 \leqslant r \leqslant t)$ as polynomials of q. By (1), we have

$$\deg\left(\mu'_C(r)\right) = \begin{cases} t-r, & r \neq t-2; \\ 1, & r = t-2; \end{cases} \tag{A1}$$

for $1 \leqslant r \leqslant t-1$. Furthermore, we have $\mu'_C(t-1) = q-1$, and $\mu'_C(t-2) = (t-2)(q-1)$.

For every $1 \leqslant r \leqslant t$, let a_r be the leading coefficient of $\mu'_C(r)$. Then we have the following relation of the leading coefficients a_r of $\mu'_C(r)$:

$$\begin{aligned} a_{t-r} + \frac{a_{t-r+1}}{1!} + \frac{a_{t-r+2}}{2!} + \frac{a_{t-r+3}}{3!} \\ + \cdots + \frac{a_{t-2}}{(r-2)!} + \frac{a_{t-1}}{(r-1)!} = 1. \end{aligned} \tag{A2}$$

Finally, by induction on r, we can prove that

$$a_{t-r} = \begin{cases} 1, & \text{if } r = 1, \\ 0, & \text{if } r = 2; \\ \dfrac{1}{2!} - \dfrac{1}{3!} + \cdots + \dfrac{(-1)^{r-1}}{(r-1)!}, & \text{if } r \geqslant 3; \end{cases} \tag{A3}$$

for every $1 \leqslant r \leqslant t-1$.

By (A1), the degrees of the numerator and denominator of the connective probability p are equal. Since the limit of connective probability as $q \to \infty$, denoted by $Pr(t)$ as above, it can be calculated by

$$Pr(t) = \frac{a_{t-1}}{(t-1)!} + \frac{a_{t-2}}{(t-2)!} + \frac{a_{t-3}}{(t-3)!} + \cdots + \frac{a_2}{2!} + \frac{a_1}{1!}, \tag{A4}$$

which will complete the proof of Theorem 3.2. □

Proof of Theorem 3.3 (*Sketch*). If $t = 3$, we can compute the formulas of the connective probability p and *fail*(1) as in the table of Example 1, and the theorem holds in this case.

If $t \neq 3$, by computing the degrees of the numerator and denominator of *Fail*(1), we get that the degree of the numerator equals the degree of the denominator -1. As to the connective probability p, the degree of the numerator equals the degree of the denominator, which will complete the proof of Theorem 3.3. □