An optimization problem for combinatorial key predistribution

Xinrong Ma*[1], Douglas R. Stinson†[2], and Ruizhong Wei‡[3]

[1]Department of Mathematics, Soochow University, Suzhou 215006, P. R. China
[2]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
[3]Department of Computer Science, Lakehead University, Thunder Bay, Ontario P7B 5E1, Canada

Abstract

We consider an optimization problem motivated by the tradeoff between connectivity and resilience in key predistribution schemes (KPS) for sensor networks that are based on certain types of combinatorial designs. For a specific class of designs, we show that there is no real disadvantage in requiring the underlying design to be regular.

1 Introduction

In Section 1.1 we briefly state the problem we are solving in this paper. Section 1.2 provides some background and motivation for the interested reader. The proof of our main result is then given in Section 2.

1.1 The problem

Briefly, the problem we are solving is as follows. Suppose we have a set system on n points, having b blocks of size k, and no pair of points occurs in more than one block. Denote the number of blocks containing the ith

*X. Ma's research is supported by NSFC grant No. 11071183.
†D. Stinson's research is supported by NSERC discovery grant 203114-11.
‡R. Wei's research is supported by NSERC discovery grant 239135-11.

point by r_i, $1 \leq i \leq n$. We are interested in values taken on by the function

$$\frac{\left(\sum_{i=1}^{n} r_i(r_i - 1)\right)^2}{\sum_{i=1}^{n} r_i(r_i - 1)(r_i - 2)}.$$

when n, b and k are fixed positive integers. Our main result is that, for $bk > 2n$, this function is maximized when all the r_i's are equal.

1.2 Background and motivation

There has been considerable recent interest in using various types of set systems, including t-designs and various generalizations, to construct deterministic key predistribution schemes (KPS) for sensor networks. There will be a set of n keys chosen randomly by a trusted authority, and a k-subset of the n keys is assigned securely to each node in a sensor network. This assignment could be done randomly (as suggested by Eschenauer and Gligor [3]) or deterministically, using a suitable combinatorial design. For a nice introduction to the combinatorial approach, see Martin [6].

Before proceeding further, we present some basic definitions regarding set systems and designs. A *set system* or *design* is a pair $(X, \mathcal{A})$, where the elements of X are called *points* and $\mathcal{A}$ is a set of subsets of X, called *blocks*. The number of points will be denoted by n and the number of blocks by b. The *degree* of a point $x \in X$ is the number of blocks containing x. The design $(X, \mathcal{A})$ is *regular of degree* r if all points have the same degree, r. If all blocks have size k, then $(X, \mathcal{A})$ is said to be *uniform of rank* k. If a design is uniform of rank k and regular of degree r, then $bk = nr$.

A *configuration* is a design that is uniform of rank k and regular of degree r, and which satisfies the additional property that any two blocks intersect in at most one point. Some basic information about configurations can be found in [1, pp. 352–355]. Examples of configurations include generalized quadrangles, BIBDs with $\lambda = 1$, and transversal designs with $\lambda = 1$.

To construct a key predistribution scheme, we start with a design that is uniform of rank k. The correspondence between the design and the associated KPS is straightforward. The n points in the design correspond to the n keys in the key pool. The b blocks correspond to the b sensor nodes in the network. Each block specifies the k keys that are given to the corresponding node. More precisely, the points in the block are the *indices* of the keys given to the node (i.e., if a block contains a point i, then the corresponding node contains the ith key in the key pool, which we denote by key_i). Note that blocks are public, while the values of the keys are secret.

Two nodes N_1 and N_2 comprise a *link* if they can construct a pairwise key to enable secure direct communication between them. This can be done if and only if the two nodes N_1 and N_2 are within each other's communication range and they have at least one common key. Suppose that N_i and N_j have exactly $\ell \geq 1$ common keys, say $\{\mathsf{key}_{a_1}, \ldots, \mathsf{key}_{a_\ell}\}$, where $a_1 < a_2 < \cdots < a_\ell$ and $i < j$. Then they can each compute the same pairwise secret key,

$$K_{i,j} = h(\mathsf{key}_{a_1} \parallel \ldots \parallel \mathsf{key}_{a_\ell} \parallel i \parallel j),$$

using an appropriate public *key derivation function*, h, which has suitable input and output sizes. Such key derivation functions could be constructed from a secure cryptographic hash function.

The most studied adversarial model in wireless sensor networks is *random node compromise* [3], wherein an adversary compromises a fixed number of randomly chosen nodes in the network and extracts the keys stored in them. Under the assumption that these nodes are then removed from the network, it is obvious that any links involving the compromised nodes are broken. In addition, a link formed by two nodes corresponding to two blocks A_1, A_2, where $|A_1 \cap A_2| \geq 1$, will be *broken* if a node corresponding to a block $B \notin \{A_1, A_2\}$ is compromised, provided that $A_1 \cap A_2 \subseteq B$. More generally, if nodes corresponding to blocks $B_1, \ldots, B_s$ are compromised, then a link corresponding to two other blocks A_1, A_2 will be broken whenever

$$A_1 \cap A_2 \subseteq \bigcup_{i=1}^{s} B_i.$$

Here are three fundamental metrics that are relevant when evaluating a KPS for a sensor network.

Storage requirements

The number of keys stored in each node is equal to k, which is the rank of the underlying design. In general, we want to *minimize* storage.

Network connectivity

It is common to measure local connectivity of a network by computing the probability that a randomly chosen pair of nodes can compute a common key, i.e., that they have at least one common point. This probability will be denoted by Pr_1. In general, we want Pr_1 to be large.

Network resilience

Resilience against node capture is commonly measured by computing the probability that a random link is broken by the compromise of a

> set of s random nodes not in the link, for suitable values of s. We denote this probability by $\mathsf{fail}(s)$. In general, we want $\mathsf{fail}(s)$ to be small. For simplicity, we will restrict our attention in this paper to the value of $\mathsf{fail}(1)$.

There is an inherent tradeoff between connectivity and resilience. In [2], Dong, Pei and Wang suggested studying the quotient $\mathsf{Pr}_1/\mathsf{fail}(s)$ in order to quantify this tradeoff. We will mainly consider $\mathsf{Pr}_1/\mathsf{fail}(1)$, which we denote by ρ. The goal is to find schemes with high values of ρ (given a specific value for k).

For a configuration-based KPS, we have the following simple analysis from [5]. Every block intersects $k(r-1)$ blocks in one point and is disjoint from all the other blocks. Therefore

$$\mathsf{Pr}_1 = \frac{k(r-1)}{b-1}.$$

A link L is defined by two blocks that intersect in one point, say x. There are $r-2$ other blocks that contain x; the corresponding nodes will compromise the link L. Therefore,

$$\mathsf{fail}_1 = \frac{r-2}{b-2}.$$

The tradeoff $\rho = \mathsf{Pr}_1/\mathsf{fail}_1$ is therefore given by the following formula:

$$\rho = \frac{k(b-2)(r-1)}{(b-1)(r-2)} \approx k.$$

Next, we describe a possible relaxation of configuration-based KPS. Suppose we employ a set system with block size k where the maximum intersection of any two blocks equals 1. However suppose we no longer require that the design is regular. Denote the degree of point i by r_i, for $1 \leq i \leq n$. Then it is clear that

$$\sum_{i=1}^{n} r_i = bk.$$

For set systems of this type, it can be shown using formulas (4) and (10) in the paper [4] that

$$\mathsf{Pr}_1 = \frac{\sum_{i=1}^{n} r_i(r_i-1)}{b(b-1)}$$

and

$$\mathsf{fail}_1 = \frac{\sum_{i=1}^{n} r_i(r_i-1)(r_i-2)}{(b-2)\sum_{i=1}^{n} r_i(r_i-1)}.$$

Therefore,

$$\rho = \frac{(b-2)\left(\sum_{i=1}^{n} r_i(r_i-1)\right)^2}{b(b-1)\sum_{i=1}^{n} r_i(r_i-1)(r_i-2)}.$$

It seems to be an interesting question to determine if we can obtain an increased value of ρ by relaxing the conditions of a configuration in this manner, since there is no particular reason to require that the underlying design is regular. The main purpose of this paper is to show that this possible relaxation does not provide any benefit: assuming that $\sum_{i=1}^{n} r_i = bk > 2n$ is fixed, we prove that the value of ρ is maximized when $r_1 = \cdots = r_n = bk/n$.

Define the function

$$f(x_1,\ldots,x_n) = \frac{\left(\sum_{i=1}^{n} x_i(x_i-1)\right)^2}{\sum_{i=1}^{n} x_i(x_i-1)(x_i-2)}. \tag{1}$$

Observe that

$$\rho = \frac{(b-2)f(r_1,\ldots,r_n)}{b(b-1)}.$$

Treating b and k as constants and writing $S = bk$, the problem then is to maximize the value of $f(x_1,\ldots,x_n)$ subject to the constraint

$$\sum_{i=1}^{n} x_i = S, \tag{2}$$

where $x_1,\ldots,x_n > 0$. Here we are treating f as a function of n real variables.

It will be convenient to write $f(x_1,\ldots,x_n) = p^2/q$, where

$$p = p(x_1,\ldots,x_n) = \sum_{i=1}^{n} x_i(x_i-1) \tag{3}$$

and

$$q = q(x_1,\ldots,x_n) = \sum_{i=1}^{n} x_i(x_i-1)(x_i-2). \tag{4}$$

2 The proof

In this section, we will prove our main theorem.

Theorem 2.1. *Suppose that $x_1,\ldots,x_n > 0$ and $\sum_{i=1}^{n} x_i > 2n$. Then, the maximum value of $f(x_1,\ldots,x_n)$, as defined in (1), subject to the constraint (2), is attained when $x_1 = x_2 = \cdots = x_n = S/n$.*

First, using Lagrange multipliers, we prove the following:

Lemma 2.2. *The maximum value of $f(x_1, \dots, x_n)$, subject to the constraint (2), is attained when the x_i's have at most two distinct values.*

Proof. Define $g(x_1, \dots, x_n, \lambda) = f(x_1, \dots, x_n) - \lambda(x_1 + \dots + x_n - S)$. It is straightforward to compute the partial derivative

$$\frac{\partial f}{\partial x_j} = \frac{p}{q}\left(q(4x_j - 2) - p(3{x_j}^2 - 6x_j + 2)\right) - \lambda, \tag{5}$$

where p and q are given by (3) and (4), respectively. Now suppose that $\frac{\partial f}{\partial x_j} = \frac{\partial f}{\partial x_k} = 0$. Then, from (5), we have

$$q(4x_j - 2) - p(3{x_j}^2 - 6x_j + 2) = q(4x_k - 2) - p(3{x_k}^2 - 6x_k + 2),$$

which simplifies to give

$$4q(x_j - x_k) = 3p(x_j - x_k)(x_j + x_k - 2).$$

Therefore, if $x_j \neq x_k$, it follows that

$$x_j + x_k = \frac{4q}{3p} + 2. \tag{6}$$

Now suppose there are three different values taken on by the x_i's, say $x_j \neq x_k \neq x_\ell \neq x_j$. Then, from (6) we obtain

$$x_j + x_k = \frac{4q}{3p} + 2 = x_k + x_\ell,$$

so $x_j = x_\ell$, which is a contradiction. □

In the proof of the Lemma 2.2, we do not need the condition $S > 2n$. It seems that the above result is the best that we can do in general (i.e., without this condition) from the following example.

Example 2.1. *Suppose $n = 4$ and $S = 5$. If $x_1 = x_2 = x_3 = x_4 = 5/4$, then $f(x_1, \dots, x_4) = -5/3$. However, if $x_1 = x_2 = \frac{5-\sqrt{15}}{4}$ and $x_3 = x_4 = \frac{5+\sqrt{15}}{4}$, then $f(x_1, \dots, x_4) = 40/3$.*

In the remainder of this section, we assume that $S > 2n$. Now we suppose that there are u x_i's having the value X and v x_i's having the value Y, where $u + v = n$ and $uX + vY = S$. Then let

$$g(X, Y) = f(x_1, \dots, x_n) = \frac{(uX(X-1) + vY(Y-1))^2}{uX(X-1)(X-2) + vY(Y-1)(Y-2)}.$$

We will prove that $g(X,Y)$ is maximized when $X = Y = S/n$. In what follows, we assume without loss of generality that $v \geq u$. Also, denote $a = S/n$. By the condition $S > 2n$, we have $a > 2$.

Let $x = X - a$ and $y = Y - a$. Then $ux + vy = 0$. Suppose we define $x = t$ and $y = -\frac{u}{v}t$. Since $X, Y > 0$, we have $-a < t < \frac{v}{u}a$.

Now consider the function

$$F(t) = g(X,Y) - g(a,a) = g\left(t + a, a - \frac{u}{v}t\right) - g(a,a).$$

Our goal is to show that $F(t) < 0$ for $-a < t < \frac{v}{u}a$.

Using some Maple computations, we have $F(t) = \frac{A(t)}{B(t)}$, where

$$\begin{aligned} A(t) \quad &= \quad (-u^2ta - 2u^2t^2 + u^2t^2a + u^2ta^2 - 2uvt^2 + ut^2va - ua^3v + uva \\ &\quad\quad - v^2ta^2 - v^2a^3 + v^2ta + v^2a)t^2u \end{aligned}$$

and

$$B(t) = -(a-2)(t^3u^2 - 3ut^2va + 3uvt^2 - ut^3v - v^2a^3 + 3v^2a^2 - 2v^2a).$$

We will now examine the behaviour of $A(t)$ and $B(t)$ in the interval $\left(-a, \frac{v}{u}a\right)$.

Lemma 2.3. *$B(t) > 0$ when $t > -a$.*

Proof. We have

$$B(t) = (a-2)\left((v-u)ut^3 + 3(a-1)uvt^2 + v^2a(a-2)(a-1)\right).$$

Let

$$d(t) = \frac{B(t)}{a-2} = (v-u)ut^3 + 3(a-1)uvt^2 + v^2a(a-2)(a-1).$$

If $u = v$, then $d(t) > 0$, so we assume that $v > u$. Then

$$d'(t) = 3(v-u)ut^2 + 6(a-1)uvt.$$

$d'(t)$ has two roots: 0 and $-\frac{2(a-1)}{v-u}$. Since $v - u > 0$, we have that $d'(t) > 0$ when $t > 0$ or $t < -\frac{2(a-1)}{v-u}$, and $d'(t) < 0$ when $-\frac{2(a-1)}{v-u} < t < 0$. Therefore $d(t)$ is an increasing function for $t > 0$ or $t < -\frac{2(a-1)}{v-u}$, and a decreasing function if $-\frac{2(a-1)}{v-u} < t < 0$.

Since $a > 2$, we have

$$\begin{aligned}
d(0) &= v^2a(a-2)(a-1) > 0, \quad \text{and} \\
d(-a) &= (u-v)ua^3 + 3(a-1)uva^2 + v^2a(a-1)(a-2) \\
&= u^2a^3 + 2uva^3 - 3uva^2 + v^2a(a-1)(a-2) \\
&> u^2a^3 + 4uva^2 - 3uva^2 + v^2a(a-1)(a-2) \\
&> 0.
\end{aligned}$$

From the above, we see that $d(t)$ has one real root, which is less than 0, and therefore the conclusion follows since $d(-a) > 0$. □

Next we consider $A(t)$. Let $h(t) = A(t)/(t^2u)$; then

$$\begin{aligned}
h(t) &= -u^2ta - 2u^2t^2 + u^2t^2a + u^2ta^2 - 2uvt^2 + ut^2va - ua^3v + uva \\
&\quad - v^2ta^2 - v^2a^3 + v^2ta + v^2a \\
&= (au^2 - 2u^2 - 2uv + uva)t^2 + (u^2a^2 - au^2 + av^2 - v^2a^2)t - uva^3 \\
&\quad + uva - v^2a^3 + v^2a.
\end{aligned}$$

From a Maple computation, $h(t)$ has two roots:

$$t_1 = \frac{C+D}{2u(a-2)} \quad \text{and} \quad t_2 = \frac{C-D}{2u(a-2)},$$

where

$$C = (v-u)a(a-1)$$

and

$$D = a\sqrt{v^2 - 4uav - 6uv - 2v^2a + u^2a^2 - 2u^2a + 2ua^2v + u^2 + v^2a^2 + 8uv/a}.$$

The leading coefficient of $h(t)$ is

$$\begin{aligned}
au^2 - 2u^2 - 2uv + uva &= u^2(a-2) + uv(a-2) \\
&= u(u+v)(a-2) \\
&> 0.
\end{aligned}$$

Also,

$$h(0) = av(u+v)(1-a^2) < 0.$$

Therefore $h(t)$ has two real roots and hence D is a positive real number.

Lemma 2.4. *$h(t) < 0$ for $t_2 < t < t_1$ and $A(t) < 0$ for $t_2 < t < t_1, t \neq 0$.*

Proof. From the above discussion, we have $h(t) < 0$ for $t_2 < t < t_1$. Since $A(t) = ut^2h(t)$, it follows that $A(t) < 0$ for $t_2 < t < t_1, t \neq 0$. □

Lemma 2.5. $t_1 > \frac{v}{u}a$.

Proof. We have

$$\begin{aligned} t_1 - \frac{v}{u}a &= \frac{C+D}{2u(a-2)} - \frac{2va(a-2)}{2u(a-2)} \\ &= \frac{1}{2u(a-2)}\left(a(3v - ua + u - va) + D\right). \end{aligned}$$

If $3v - ua + u - va \geq 0$, then it is clear that $t_1 - \frac{v}{u}a > 0$, so we assume that $3v - ua + u - va < 0$. We compute

$$\begin{aligned} D^2 - a^2(3v - ua + u - va)^2 &= a^2(4v^2a - 8v^2 + 4uva - 12uv + 8uv/a) \\ &= a^2\left(v^2(4a-8) + uv(4a - 12 + 8/a)\right) \\ &> 0, \end{aligned}$$

where we use the fact that $4a - 12 + 8/a > 0$ when $a > 2$. This implies that

$$(D - a(3v - ua + u - va))(D + a(3v - ua + u - va)) > 0.$$

Since the first factor is positive, the second factor is also positive, and the conclusion follows. □

A similar technique can be used to prove the following lemma.

Lemma 2.6. $t_2 < -a$.

Proof. We have

$$\begin{aligned} t_2 + a &= \frac{C-D}{2u(a-2)} + \frac{2ua(a-2)}{2u(a-2)} \\ &= \frac{1}{2u(a-2)}\left(a(ua - v - 3u + va) - D\right) \end{aligned}$$

We have that

$$\begin{aligned} ua - v - 3u + va &= (u+v)a - v - 3u \\ &> 2(u+v) - v - 3u \\ &= v - u \\ &\geq 0. \end{aligned}$$

Also,

$$\begin{aligned} a^2(ua - v - 3u + va)^2 - D^2 &= a^2(8v^2 - 4uva + 12uv - 4v^2a - 8uv/a) \\ &= a^2\left(v^2(8-4a) - uv(4a - 12 + 8/a)\right) \\ &< 0. \end{aligned}$$

This implies that

$$(a(ua - v - 3u + va) - D)(a(ua - v - 3u + va) + D) < 0.$$

Since the second factor is positive, the first factor is negative, and the conclusion follows. □

We can now complete the proof of our main theorem.

Proof of Theorem 2.1. By Lemma 2.2, $f(x_1, \ldots, x_n)$ is maximized only if there are at most two different values of the x_i's, which we denoted by X and Y. If $X \neq Y$, then Lemmas 2.4, 2.5 and 2.6 proved that $A(t) < 0$ for $-a < t < \frac{v}{u}a$, $t \neq 0$. Therefore, from Lemma 2.3, it follows that $F(t) < 0$ for $-a < t < \frac{v}{u}a$, $t \neq 0$. Thus, the maximum value of $F(t)$ for $-a < t < \frac{v}{u}a$ is $F(0) = 0$. This means the two values X and Y are the same, which completes the proof. □

3 Conclusion

The proof of our main theorem is rather technical. A simpler and more illuminating proof would be very nice.

Another direction for future research would be to generalize our results to other types of designs used for constructing KPS in sensor networks. A very general class of designs, called *partially balanced t-designs*, were introduced in [7] and were evaluated for their suitability in constructing KPS. It turns out that a partially balanced 2-design (as defined in [7]) is just a configuration. This was one motivation for studying the class of designs considered in the current paper. A natural extension would be to consider relaxations of partially balanced t-designs when $t > 2$.

References

[1] C.J. Colbourn and J.H. Dinitz, eds. *Handbook of Combinatorial Designs, Second Edition*, Chapman & Hall/CRC, 2007.

[2] J.-W. Dong, D.-Y. Pei and X.-L. Wang. A class of key predistribution schemes based on orthogonal arrays. *Journal of Computer Science and Technology* **23** (2008), 825–831.

[3] L. Eschenauer and V. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security.* ACM Press, 2002, pp. 41–47.

[4] K. Henry, M.B. Paterson and D.R. Stinson. Flexible choice of network size in key predistribution schemes based on transversal designs. *Lecture Notes in Computer Science* **8282** (2014) 89–117 (SAC 2013 Proceedings).

[5] J. Lee and D.R. Stinson. On the construction of practical key predistribution schemes for distributed sensor networks using combinatorial designs. *ACM Transactions on Information and System Security*, **11(2)** (2008), article No. 1, 35 pp.

[6] K.M. Martin. On the applicability of combinatorial designs to key predistribution for wireless sensor networks. *Lecture Notes in Computer Science* **5557** (2009), 124–145 (IWCC 2009).

[7] M.B. Paterson and D.R. Stinson. A unified approach to combinatorial key predistribution schemes for sensor networks. *Designs, Codes and Cryptography* **71** (2014), 433–457.