GUESSING SECRETS

Fan Chung* Ronald Graham Tom Leighton

University of California, San Diego
La Jolla, California
fan@ucsd.edu, graham@ucsd.edu

MIT
Cambridge, Massachusetts
ftl@math.mit.edu

Submitted: February 9, 2001; Accepted: February 15, 2001.
MR Subject Classifications: 05C05, 05C65, 68R05

Abstract

Suppose we are given some fixed (but unknown) subset X of a set Ω, and our object is to learn as much as possible about the elements of X by asking binary questions. Specifically, each *question* is just a function $F : \Omega \to \{0, 1\}$, and the *answer* to F is just the value $F(X_i)$ for *some* $X_i \in X$, (determined, for example, by a potentially malevolent but truthful, adversary). In this paper, we describe various algorithms for solving this problem, and establish upper and lower bounds on the efficiency of such algorithms.

1 Introduction

In this paper we consider a variant of the familiar "20 questions" problem in which someone (called the "Seeker") tries to discover the identity of some unknown "secret" by asking binary questions (e.g., see [15]). In our variation, there is now a set of $k \geq 2$ secrets. For each question asked, an "Adversary" gets to choose *which* of the k secrets to use in supplying the answer, which in any case must be truthful. We will describe a number of algorithms for dealing with this problem, although we still are far from a complete understanding of the situation. We will also describe the connection of these problems with some classic results of Erdős and Lovász [12] and others [13, 14] on 3-chromatic hypergraphs. Secret guessing problems of this type have arisen recently in connection with certain Internet traffic routing applications [20].

*Research supported in part by NSF Grant No. DMS 98-01446

2 The basic setup

To begin with we restrict ourselves to the case of $k = 2$. In this case, the Adversary **A** has a set $X = \{X_1, X_2\}$ of two secrets, taken from a universe Ω of N possible secrets. A question F is just a function $F : \Omega \to \{0, 1\}$. The adversary **A** has a choice of answering the question F with either of the values $F(X_1)$ or $F(X_2)$. The job of the Seeker **S** is to select questions so as to determine as much about the secrets as efficiently as possible. Observe that **S** can never hope to learn with certainty more than one of **A**'s secrets, since **A** can always answer *every* question using the *same* $X_i \in X$. So, how much can **S** be guaranteed of finding out about **A**'s secrets ?

To get a firmer grip on these questions, we will model our problem in terms of graphs. Let K_N denote the complete graph on the set of N vertices Ω. A pair of secrets $X = \{X_1, X_2\}$ corresponds to an *edge* X_1X_2 of K_N. Each question $F : \Omega \to \{0, 1\}$ induces a partition of $\Omega = F^{-1}(0) \cup F^{-1}(1)$. The answer $\alpha \in \{0, 1\}$ to the question F given by **A** implies that $X \cap F^{-1}(1 - \alpha) = \emptyset$. Thus, **S** can remove all the edges spanned by $F^{-1}(1 - \alpha)$ as possible candidates for $X = \{X_1, X_2\}$.

The process is complete and **S** is finished as soon as the set W of surviving edges is "intersecting", i.e., contains *no pair of disjoint edges.* For **S** can certainly reach this state (by repeatedly placing disjoint edges in different blocks of the partitions). It is equally clear that **A** can "protect" any intersecting set W by making sure not to discard any block of a partition which contains an edge of W. We will call a strategy "separating" if by using it, **S** can always reach an intersecting set of edges, no matter how **A** answers the questions.

For graphs, there are just two types of intersecting sets W. The first type is a *star*, i.e., a set of edges all sharing a common vertex X_0. In this case, **S** can assert that X_0 is indeed one of **A**'s secrets. The second type is a *triangle*, i.e., the complete graph K_3 with 3 edges on a set $\{X_0, X_1, X_2\}$ of size 3. In this case, all that **S** can assert is that **A**'s secret pair is one of the edges X_0X_1, X_0X_2 or X_1X_2 of the K_3. (In other words, **A** can choose the answer *majority* $\{F(X_1), F(X_2), F(X_3)\}$. By doing so, no edge of W is ever removed.) In particular, **S** cannot specify that any particular element of Ω is one of **A**'s secrets.

There are two kinds of strategies we will consider for **S**, namely *adaptive* and *oblivious*. In an adaptive strategy each question of **S** can depend on the answers to all preceding questions. On the other hand, in an oblivious strategy, all of **S**'s questions must be asked in advance of any of **A**'s answers.

We will give an adaptive separating strategy for **S** for which the number of steps required is reasonably close to the optimum. We will also give oblivious separating strategies with somewhat larger constants. In addition, we will discuss possible strategies when the questions are restricted in various ways, e.g., to be very compact. Finally we will examine the more complex situation in the case of $k \geq 3$ secrets.

3 Adaptive algorithms

In this section we focus on adaptive strategies, i.e., where future questions can depend on past answers. Let us say that a separating strategy has *length* t if **S** can force the surviving set W of edges to be intersecting in at most t steps, no matter how **A** selects answers. Define $f(N)$ to be the least value of t such that there exists a separating strategy of length t for the initial set Ω of size N.

Theorem 1

$$3\log_2(N) - 5 \leq f(N) \leq 4\log_2(N) + 3, \qquad N > 2.$$

Proof: For the lower bound, it suffices to observe that since the initial graph K_N has $\binom{N}{3}$ triangles, and at each stage, **A** can guarantee to save at least half of the existing triangles, and since the final set of edges can have at most one triangle, then any separating strategy will require at least $\log_2\binom{N}{3}$ steps which is at least $\log_2\binom{N}{3} > 3\log_2 N - 5$ for $N > 2$.

For the upper bound, we will derive recursive bounds on the minimum number of steps required to reach an intersecting set of edges starting from three special kinds of graphs. These are:

- $K(m,n)$ - the complete bipartite graph on m and n vertices;
- $\bar{K}(m,n)$ - the graph formed by joining every vertex of a complete graph $K(m)$ on m vertices to every vertex of an independent set of n vertices; and
- $K(m,m,n)$ - the complete tripartite graph on m, m and n vertices.

We denote these symbolically in Figure 1:

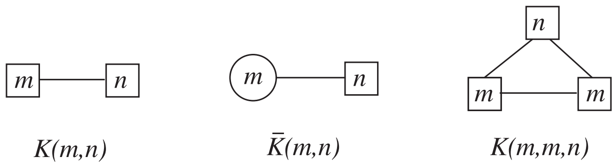

Figure 1: Three basic graphs

Denote the minimum number of steps in any adaptive separating strategy starting with these graphs by $f(m,n), \bar{f}(m,n)$ and $f(m,m,n)$, respectively. For convenience, we will assume that m and n are powers of 2, with $n \geq m > 1$. We will then use the monotonicity of the f's to obtain bounds for general m and n.

To begin, let us first consider $f(m,n)$. **S**'s strategy will be to select a question (= partition) F which splits each of the two vertex sets in half. Symbolically, we show this in Figure 2

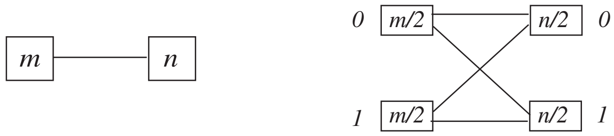

Figure 2: Splitting $K(m,n)$

where the 0's and 1's indicate the vertices in $F^{-1}(0)$ and $F^{-1}(1)$, respectively. Since this assignment is symmetrical then we can assume without loss of generality that **A** chooses the answer 0, so that all edges spanned by $F^{-1}(1)$ are eliminated. This leaves the graph in Figure 3

(i.e., the edges between the two lower-level boxes are gone). Next, suppose **S** specifies the partition shown below in Figure 4.

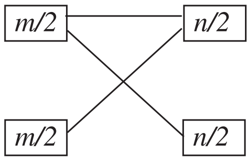

Figure 3: The remaining graph after splitting.

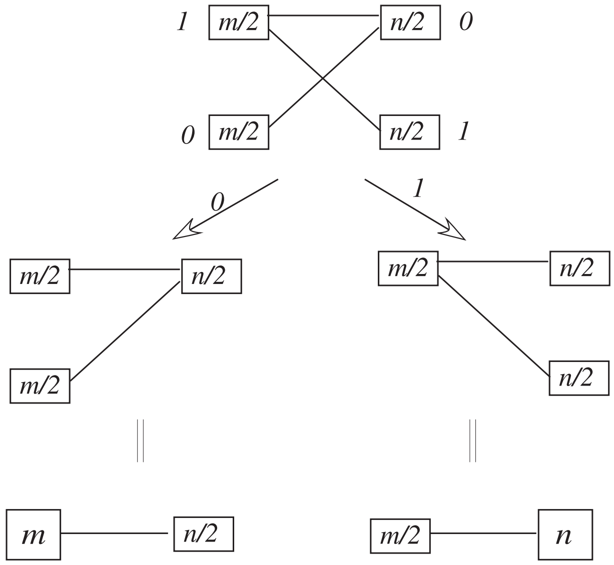

Figure 4: Reduction into bipartite graphs

If **A** answers 0, then we follow the left-hand branch labeled 0. Otherwise, we follow the right-hand branch. In each branch, we have simplified the presentation of the resulting graph by recognizing that it is a (smaller) complete bipartite graph. Hence, we have the recurrence

$$f(m,n) \leq 2 + \max\{f(m,n/2)), f(m/2,n)\} \tag{1}$$

Of course, $f(1,n) = f(m,1) = 0$, since $K(1,n)$ and $K(m,1)$ are both stars. It is now straightforward to show that this recurrence implies the bound

$$f(m,n) \leq 2(\log_2 m + \log_2 n - 1). \tag{2}$$

Next, we will treat $f(m,m,n)$, this time in a more abbreviated fashion. We begin with $K(m,m,n)$ where $n \geq m > 1$, with m and n powers of 2. **S**'s first question will split each of the three vertex sets in half as shown in Figure 5.

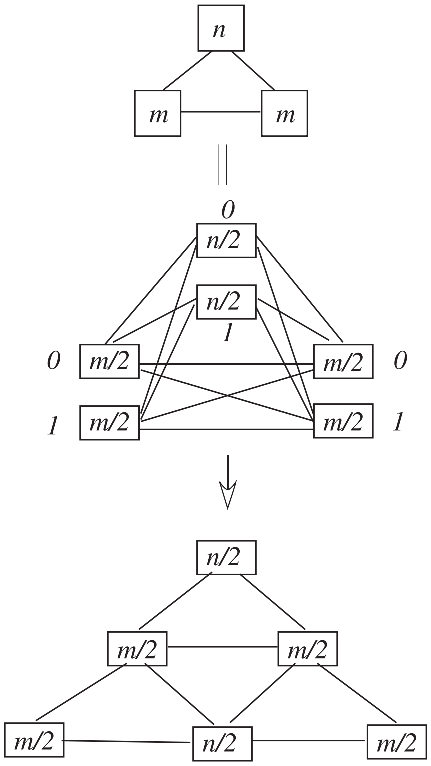

Figure 5: Splitting $K(m, m, n)$

By symmetry, we can assume without loss of generality that **A** selects the answer 0, resulting in the graph shown in Figure 5. In the next diagram (in Figure 6), we show the strategy tree for **S**'s next three questions.

Thus, we have the bound

$$\begin{aligned} f(m,m,n) &\leq 4 + \max\{f(m/2,m/2,n/2), f(2m,n/2), f(2n,m/2)\} \\ &\leq 4 + \max\{f(m/2,m/2,n/2), 2(\log_2 m, \log_2 n - 1)\} \end{aligned} \tag{3}$$

For the case that $m = 1$ we have the picture in Figure 7.

Thus,

$$f(1,1,n) \leq 2 + f(1,1,n/2), \qquad f(1,1,1) = 0 \tag{4}$$

which implies

$$f(1,1,n) \leq 2\log_2 n. \tag{5}$$

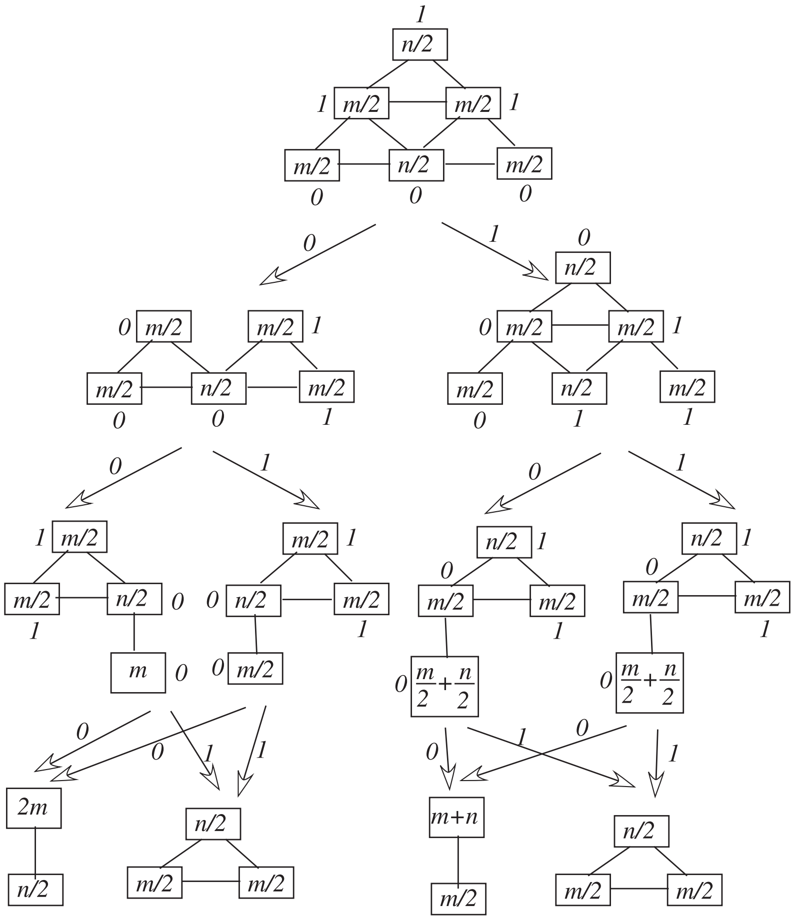

Figure 6: Three more steps

An easy calculation now shows that together with (2), we have

$$f(m,m,n) \leq 2(1+\log_2 m + \log_2 n). \tag{6}$$

Finally, we have $\bar{f}(m,n)$ with m a power of 2, and $n \geq m > 1$.

Thus,

$$\bar{f}(m,n) \leq 2 + \max\{\bar{f}(m/2, n+m/2), f(m/2, m/2, (n/2)^+)\} \tag{7}$$

where x^+ denotes the least power of 2 which is $\geq x$.

By (6), $f(m/2, m/2, (n/2)^+) \leq 2(\log_2 m + \log_2 n)$, so that

$$\bar{f}(m,n) \leq 2 + \max\{\bar{f}(m/2, n+m/2), 2(\log_2 m + \log_2 n)\}. \tag{8}$$

Therefore we have

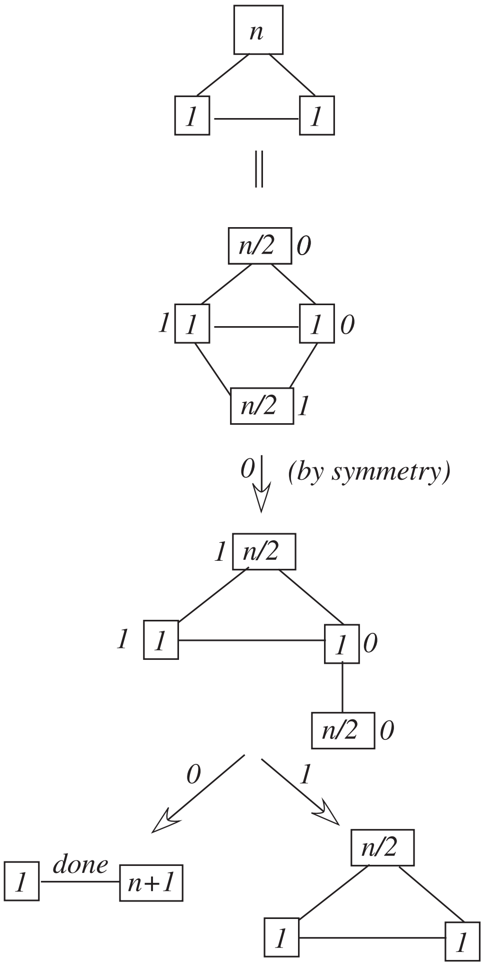

Figure 7: The case $m = 1$

$$\bar{f}(m, n) \leq 2 + 2(\log_2 m + \log_2(n + m/2)). \tag{9}$$

Finally, since our starting graph K_N can be reduced in one step to $\bar{K}(\lceil N/2 \rceil, \lfloor N/2 \rfloor)$ then $f(N)$, the number of steps required for any separating strategy is bounded by

$$\begin{aligned} f(N) &\leq 1 + \bar{f}(\lceil N/2 \rceil^+, \lfloor N/2 \rfloor) \\ &\leq 3 + 2(\log_2 N + \log_2(N/2 + N/2)) \qquad \text{by (9)} \\ &\leq 3 + 4 \log_2 N. \end{aligned} \tag{10}$$

This completes the proof for Theorem 1. □

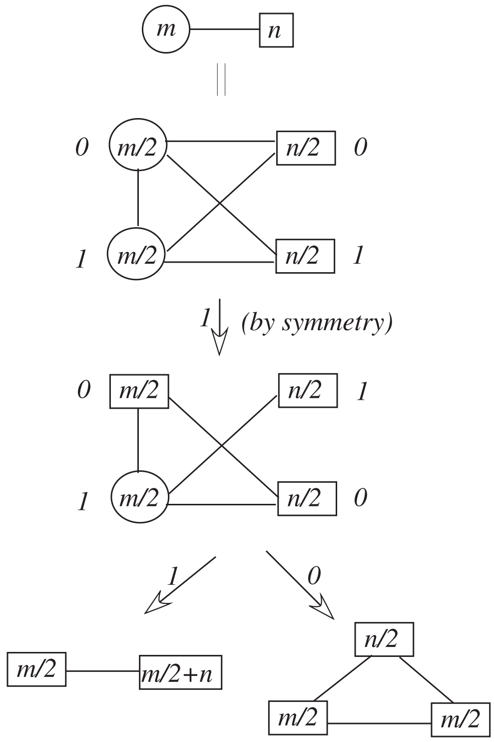

Figure 8: Reductions for $\bar{K}(m, n)$

We suspect that the truth here is

$$f(N) = (1 + o(1))4 \log_2 N.$$

4 Oblivious algorithms

In the case of oblivious algorithms (where all questions are asked before any answers are given), let $f_0(N)$ denote the minimum number of questions needed to separate the edges of K_N.

Theorem 2

$$f_0(N) \quad \leq \quad (c + o(1)) \log_2 N \tag{11}$$

where $c = 3/\log_2 8/7 = 15.57\ldots$.

Proof: First we state a simple proof using the basic probabilistic method. For an integer t to be specified later, label each vertex S of Ω with a random binary t-tuple $\lambda(S) = (S(1), S(2), ..., S(t))$. The value of $S(i)$ will correspond to the part of the ith partition of $\Omega = F_i^{-1}(0) \cup F_i^{-1}(1)$ to which S belongs. The assignment λ separates the disjoint pairs $X = \{X_1, X_2\}$ and $Y = \{Y_1, Y_2\}$ provided for some $i, X_1(i) = X_2(i) \neq Y_1(i) = Y_2(i)$. There are 14 of the 16 possible assignments to these four coordinates for which this does *not* happen (X and Y are disjoint). Hence, the probability that λ does not separate X and Y is $\leq (7/8)^t$. Since there are just $1/2\binom{N}{2}\binom{N-2}{2}$ disjoint pairs X and Y in K_N, then some separating set of t questions must exist provided

$$(7/8)^t(1/2)\binom{N}{2}\binom{N-2}{2} < 1. \tag{12}$$

This is satisfied for $t = (c_1 + o(1))\log_2 N$ with $c_1 = 4/(\log_2 8/7) = 20.76\ldots$.

This bound can be improved by using the deletion method (see [5]) as pointed out by Noga Alon [1], or by using the inner product strategy as described in the next section. To apply the deletion method, we choose a random $t \times 2N$ binary array M. The probability that a given disjoint pair X and Y of pairs of elements of Ω' with $|\Omega'| = 2N$ are not separated by any particular row (= question) of M is $7/8$. Hence, the expected number of "bad" pairs X and Y is less than $\binom{2N}{2}^2(\frac{7}{8})^t$. We choose t large enough so that this expression is less than N. Thus, some $t \times 2N$ array M_0 has $< N$ bad pairs X and Y. Now, delete one column corresponding to one element from each of these bad pairs (of pairs). The resulting array M_1 has t rows and $\geq N$ columns with no bad pairs, i.e., all its disjoint pairs are separated by the rows of M_1. This gives an upper bound of $c\log_2 N$ where $c = 3/\log_2 8/7 = 15.57\ldots$. □

5 Inner product strategies

One disadvantage of the preceding approach is that the questions used to achieve the $O(\log N)$ bounds might in fact require $\Omega(N)$ bits for their description. We would like to have questions that can be represented very compactly, e.g., using just $O(\log N)$ bits.

One way to do this is as follows. Let us represent Ω as $GF(2)^n$, an n-dimensional vector space over $GF(2) = \{0, 1\}$ (so that $N = 2^n$). The allowable questions F will now just be vectors $F = (F(1), F(2), \dots, F(n)) \in GF(2)^n$. The *answer* to the question F will be $F \cdot X_i$, the inner product (mod 2) of F with some secret $X_i \in X$. We will call strategies for separating edges in this setting "inner product" strategies.

Theorem 3 *There is an inner product separating strategy for $\Omega = GF(2)^n$ with at most $3/(\log_2 8/7)\log_2 N$ questions, where $N = 2^n$.*

Proof: We again use the probabilistic method. We choose a random set of $3/(\log_2 8/7)n$ random inner product questions. A particular question F will separate the disjoint pairs $X = \{X_1, X_2\}$ and $Y = \{Y_1, Y_2\}$ provided

$$F \cdot X_1 \equiv F \cdot X_2 \not\equiv F \cdot Y_1 \equiv F \cdot Y_2 (\text{mod } 2)$$

For these disjoint pairs X and Y, define the three vectors

$$\Delta_1 = X_1 - X_2, \Delta_2 = X_2 - Y_1, \Delta_3 = Y_2 - Y_1,$$

and let Δ denote the $3 \times n$ array

$$\Delta = \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{pmatrix} = \begin{pmatrix} \Delta_1(1) & \Delta_1(2) & \dots & \Delta_1(n) \\ \Delta_2(1) & \Delta_2(2) & \dots & \Delta_2(n) \\ \Delta_3(1) & \Delta_3(2) & \dots & \Delta_3(n) \end{pmatrix}$$

Thus, F separates X and Y provided

$$F \cdot \Delta_1 \equiv 0, \quad F \cdot \Delta_2 \equiv 1, \quad F \cdot \Delta_3 \equiv 0,$$

i.e.,

$$F \cdot \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{pmatrix} = \begin{pmatrix} F \cdot \Delta_1 \\ F \cdot \Delta_2 \\ F \cdot \Delta_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{13}$$

Let us say that a column $\Delta(i) = \begin{pmatrix} \Delta_1(i) \\ \Delta_2(i) \\ \Delta_3(i) \end{pmatrix} = C(k)$ of Δ is of type k, $0 \leq k \leq 7$, if $k = \Delta_1(i) + 2\Delta_2(i) + 4\Delta_3(i)$ (i.e., the column $\Delta(i)$ is just the binary expansion of k), and let N_k denote the number of columns of Δ of type k. Thus,

$$\sum_{k=0}^{7} N_k = n.$$

The hypothesis that $X \cap Y = \emptyset$ implies

$$\begin{aligned} &N_2 + N_3 + N_4 + N_5 > 0 \\ &N_2 + N_3 + N_6 + N_7 > 0 \\ &N_1 + N_2 + N_5 + N_6 > 0 \\ &N_1 + N_2 + N_4 + N_7 > 0 \end{aligned} \tag{14}$$

Claim: At least $1/8$ of the 2^n possible $F \in \Omega$ satisfy (13).

Proof of Claim: There are several cases.

Case 1. Δ has three independent columns, say $\Delta(i)\Delta(j)$ and $\Delta(k)$. Since the linear span of these three columns contains each of the eight possible columns exactly once, then the Claim holds in this case. That is, for any choice of $F(t), t \neq i, j, k$, we can choose $F(i), F(j), F(k) \in \{0, 1\}$ so that (13) holds.

In particular, this implies that the Claim also holds when Δ has at least four distinct columns, say $\Delta(k_1), \Delta(k_2), \Delta(k_3), \Delta(k_4)$. For if no three columns were independent then we would have

$$\begin{aligned} \Delta(k_1) + \Delta(k_2) + \Delta(k_3) &= 0, \\ \Delta(k_1) + \Delta(k_2) + \Delta(k_4) &= 0 \end{aligned}$$

which implies that $\Delta(k_3) = \Delta(k_4)$, a contradiction.

Case 2. Case 1 does not hold and $N_2 > 0$. Thus, Δ contains at least one column $\Delta(i) = C(k) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ of type 2. Since there can be at most $r \leq 2$ other column types in

Δ, then at least $(1/2)(1/2^r) \geq 1/8$ of the $F \in \Omega$ satisfy (13) (where the factor $1/2^r$ comes from the choices for the non-type 2 columns to contribute $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ and the $1/2$ comes from the number of ways of choosing an odd number of coordinates of F to be 1 in positions which have a type 2 column).

Case 3. Δ has just two different column types, and $N_2 = 0$. Since these two column types must satisfy (14) then they can only be the columns $\{C(1), C(3)\}, \{C(4), C(6)\}$, or $\{C(5), C(7)\}$. However, in each of these cases, the sum of the two columns in equal to $C(2)$, and so, at least $1/4$ of the linear combinations of the columns of Δ are $C(2)$'s, and consequently, this case is done.

Case 4. Δ has three distinct (dependent) column types and $N_2 = 0$. Thus, the three column types are $\{C(1), C(4), C(5)\}$ or $\{C(1), C(6), C(7)\}$. However, in both of these cases, (14) fails to be satisfied, so that this case cannot hold.

This proves the Claim.

Hence, for each choice of Δ (corresponding to X and Y with $X \cap Y = \emptyset$), the probability that t randomly chosen F's all fail to separate X and Y is $\leq (7/8)^t$. Since there are strictly fewer than 8^n choices for Δ (taking the symmetry of X and Y into account), then there must exist some set of t questions which separate all disjoint pairs of X and Y, provided

$$8^n (7/8)^t \leq 1,$$

i.e.,

$$t \geq (\log_2 8)/(\log_2 8/7)n = 3/(\log_2 8/7) \log_2 N.$$

This proves Theorem 3. $\square$

6 Constructive inner product strategies

One disadvantage of the approach taken in the preceding sections for showing the existence of small separating sets of questions is that they are non-constructive. That is, they do not give any information on how to actually produce the desired sets. We now remedy this defect, but at the cost of increasing the number of questions to $\Omega(\log^2 N)$.

For this construction, we choose a large prime $p \geq 49n^2$ and we form the (cyclic) sequence $Q = (q(0), q(1), ..., q(p-1))$ where

$$q(k) = \begin{cases} 1 & \text{if } k \text{ is a quadratic non-residue of } p, \\ 0 & \text{otherwise.} \end{cases}$$

The inner product questions for this construction will just be the p consecutive blocks $Q_x = (q(x+1), q(x+2), ..., q(x+n)), 0 \leq x < p$, where index addition is taken modulo p, i.e., $q(p) = q(0)$, etc. Note that $q(k)$ can be expressed as

$$q(k) = 1/2(1 - \chi_p^*(k))$$

where

$$\chi_p^*(k) = \begin{cases} -1 & \text{if } k \text{ is a quadratic non-residue of } p, \\ 1 & \text{otherwise.} \end{cases}$$

Note that χ_p^* differs from the usual non-trivial quadratic character χ_p of p only in that we have defined $\chi_p^*(0) = 1$, whereas by convention $\chi_p(0)$ is taken to be 0.

For a given disjoint pair $X = \{X_1, X_2\}$ and $Y = \{Y_1, Y_2\}$ in $\Omega = GF(2)^n$, define

$$\Delta_1 = X_1 - X_2, \quad \Delta_2 = Y_1 - X_2, \quad \Delta_3 = Y_2 - Y_1$$

and

$$\Delta = \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{pmatrix} = \begin{pmatrix} \Delta_1(1) & \Delta_1(2) & \dots & \Delta_1(n) \\ \Delta_2(1) & \Delta_2(2) & \dots & \Delta_2(n) \\ \Delta_3(1) & \Delta_3(2) & \dots & \Delta_3(n) \end{pmatrix}$$

As before, we want to show that (for p large enough) there will always be a block Q_x of Q of length n such that

$$Q_x \cdot \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{pmatrix} = \begin{pmatrix} Q_x \cdot \Delta_1 \\ Q_x \cdot \Delta_2 \\ Q_x \cdot \Delta_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

(which implies by the remarks in the preceding section that the Q_x are separating). Next, for $1 \leq k \leq 3$, define

$$\delta_k = \{i : \Delta_k(i) = 1\}.$$

Observe that

$$\frac{1}{2}(1 - \prod_{i\in\delta_k} \chi^*(x+i)) = 0$$

if an *even* number of terms $x+i$ are quadratic non-residues of p, and 1 otherwise. Hence, we have

$$\frac{1}{2}(1 - \prod_{i\in\delta_k} \chi^*(x+i)) = \begin{cases} 0 & \text{if } Q_x \cdot \Delta_k \equiv 0 \pmod 2, \\ 1 & \text{if } Q_x \cdot \Delta_k \equiv 1 \pmod 2. \end{cases}$$

Thus, the product

$$P(x) = (1 + \prod_{i\in\delta_1} \chi^*(x+i))(1 - \prod_{i\in\delta_2} \chi^*(x+i))(1 + \prod_{i\in\delta_3} \chi^*(x+i)) > 0$$

if and only if

$$\begin{pmatrix} Q_x \cdot \Delta_1 \\ Q_x \cdot \Delta_2 \\ Q_x \cdot \Delta_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

i.e., if and only Q_x separates X and Y. Note that we always have $P(X) \geq 0$. Now consider the sum $S = \sum_{x=0}^{p-1} P(X)$. We will show that if $p \geq 49n^2$ then $S > 0$. This will then imply that some $P(x) > 0$, and so, X and Y are separated by Q_x. Since X and Y were arbitrary, then the proof will be complete.

To estimate S, we expand each term $P(x)$ into a sum of eight terms, sum each of these over x, and use a variant of the powerful character sum estimate of Weil to bound all the non-trivial terms. The trivial terms in the expansion are 1, and we will see that its sum $\sum_{x=0}^{p-1} 1 = p$ will be a dominant term. The other sums will have the forms

$$\pm\sum_{x=0}^{p-1} \prod_{i\in\delta_u} \chi_p^*(x+i), \quad \pm\sum_{x=0}^{p-1} \prod_{i\in\delta_u} \chi_p^*(x+i) \prod_{j\in\delta_v} \chi_p^*(x+j),$$

$$\text{and} \quad \pm\sum_{x=0}^{p-1} \prod_{i\in\delta_u} \chi_p^*(x+i) \prod_{j\in\delta_v} \chi_p^*(x+j) \prod_{k\in\delta_w} \chi_p^*(x+k) \quad \text{for distinct } u, v, w.$$

Recall the Weil estimate:

Theorem ([8]) For distinct $a_1, a_2, \ldots, a_s$ residues modulo a prime p, and $s \geq 1$,

$$|\sum_{x=0}^{p-1} \prod_{k=1}^{s} \chi_p(x+a_k)| \leq (s-1)\sqrt{p}. \tag{15}$$

A simple modification of (15) with χ_p^* replacing χ_p gives under the same hypothesis the estimate

$$|\sum_{x=0}^{p-1} \prod_{k=1}^{s} \chi_p^*(x+a_k)| \leq s\sqrt{p}. \quad \text{for} \;\; p \geq s^2. \tag{16}$$

Notice that the only sums which occur with a minus sign are those involving a product over δ_2. None of these products can "collapse" to 1 (i.e., every factor $\chi_p^*(t)$ occurs an even number of times) since the assumption that X and Y are disjoint implies that $\Delta_2 \neq \bar{0}, \Delta_1 + \Delta_2 \neq \bar{0}, \Delta_2 + \Delta_3 \neq \bar{0}$ and $\Delta_1 + \Delta_2 + \Delta_3 \neq \bar{0}$. Each of the products corresponds to a polynomial of degree at most n since there are only n distinct terms of the form $\chi_p^*(x+i)$. Thus, (16) implies

$$S > p - (3n\sqrt{p} + 3 \cdot n\sqrt{p} + 1 \cdot n\sqrt{p}) = p - 7n\sqrt{p} \geq 0 \;\; \text{for} \;\; p \geq 49n^2.$$

This proves the theorem. □

We believe that this construction may well be valid for much smaller values of p, e.g., $p = cn^{3/2}$ or perhaps even $p = cn \log n$ (or $p = cn$ for large c?). We have performed some limited computational experiments which are consistent with this belief. To prove such statements, however, would require much more careful analysis of the terms of S, and more powerful character sum estimates than are currently available.

It is possible that the same kind of analysis can be done using "quasi-random" subsets of the integers modulo p (or for general composites m) in place of quadratic residues. These are subsets of Z/mZ which share many of the properties of random subsets of Z/mZ (e.g., see [9, 10] for a discussion). We plan to explore this approach in the future.

7 Invertible strategies

It turns out that all the preceding strategies suffer from one slight (!) defect. Namely, it is not at all obvious how to deduce the sought-after secret (or the 2-out-of-3 secrets) from

A's answers, even when we know that the questions do separate. In other works, even knowing that the surviving edges are intersecting, how do we identify the resulting star or triangle? In this section we present an even simpler (though larger) set of inner product questions for $\Omega = GF(2)^n$ for which there is a polynomial-time algorithm for recovering the secrets. For this construction, we take for our set of inner product questions all vectors $V \in \Omega$ with at most three non-zero coordinates. An easy case analysis shows that this set is separating. To invert, we outline a recursive algorithm due to Lincoln Lu [19]. Suppose we have an algorithm ALG$(2k)$ for inverting the answers for an initial set $\Omega_k = GF(2)^{2k}$ which requires $f(2k)$ steps. We assume ALG$(2k)$ produces as its output *either* one secret X_i and a matrix of consistent linear constraints which the other secret X_2 must satisfy, *or* a triple $\{X_1, X_2, X_3\}$ from which any pair is valid. We will use it three times to invert answers for $\Omega_{3k} = GF(2)^{3k}$ as follows.

Define three subsets of the coordinate set $\{1, 2, \ldots, 3k\}$
$A_1 = \{1, 2, \ldots, 2k\}$, $A_2 = \{k+1, \ldots, 3k\}$, $A_3 = \{1, \ldots, k\} \cup \{2k+1, \ldots, 3k\}$. Apply ALG(2k) to each of the sets A_i, $1 \leq i \leq 3$. The result for each will be a small set of possibilities which must all be consistent with the actual pair of secrets chosen for Ω_{3k}. In particular, it is not hard to see that some secret X_i must be represented in at least two of the three cases and since the union of any pair of the A_i is $\{1, 2, \ldots, 3k\}$, then we can write down *all* bits of candidates for possible solutions of ALG$(3k)$. (In fact, all the solutions of ALG$(3k)$ must be contained in the set of candidates.) Then we check all the questions on each of the candidates Y_1, computing at the end the companion matrix of linear constraints not satisfied by Y_1. For those Y_1 having solvable companion matrices, we can then deduce the solutions for ALG$(3k)$.

Lincoln Lu has written a very slick recursive program for implementing this algorithm. Although the upper bound for the complexity of this algorithm is of $O(n^4)$ (because of the steps involving Gaussian elimination for sparse matrices), the actual running time seems to be much faster in all the examples that we tested.

8 More Secrets

We next consider the situation in which **A** has $k = 3$ secrets $X = \{X_1, X_2, X_3\} \subseteq \Omega$, where we assume that $|\Omega| = N$. As usual, we will restrict ourselves to the situation that **S** must use binary questions to gain information about X. From a graph-theoretic point of view, we begin with $K_N^{(3)}$, the complete triple system (= 3-uniform hypergraph) on Ω. Each question F of **S** is a partition of Ω into two sets $\Omega = F^{-1}(0) \cup F^{-1}(1)$. **A** then selects one of the sets $F^{-1}(\alpha)$ and all the triples in the complement $F^{-1}(1-\alpha)$ are discarded. The process terminates as soon as **S** can guarantee that the surviving triples form an intersecting family of triples, i.e., $T \cap T' \neq \emptyset$ for any two surviving triples T and T'.

For $k = 2$, it was easy to see that there were just two types of intersecting sets, namely stars and triangles. We call the first type *extendible*, since there is no bound on the possible degree of the star. On the other hand, the triangle is a non-extendible intersecting family (of edges).

For $k = 3$, the situation is more complicated. We will describe the various possibilities that the vertex set is $\Omega = 1, 2, ..., N$. We first list the extendible intersecting families of triples.

(i) $1xy$ (in other words, all the triples containing a fixed element, here called 1)

(ii) $12x, 13y, 23z$ (in other words, all triples containing at least two elements from a fixed triple).

(iii) $134, 135, 145, 234, 235, 245, 12x$

(iv) $134, 156, 235, 236, 245, 246, 12x$

(v) $134, 136, 156, 235, 236, 246, 12x$

We next list the non-extendible (i.e., maximal) intersecting families of triples.

(vi) $123, 124, 125, 134, 135, 145, 234, 235, 245, 345$ (i.e., all the triples from a fixed 5-element set).

(vii) $123, 145, 167, 246, 257, 347, 356$ (i.e., the 7 lines of a projective plane $PP(2)$ of order 2).

(viii) Any set of 10 triples from $\{1, 2, 3, 4, 5, 6\}$ which doesn't contain a triple and its complement, and which is not (i) or (ii). By results of Frankl, Ota and Tokushige [13], there are

5 non-isomorphic such families.

(ix) $123, 145, 167, 124, 126, 146, 246, 247, 256, 356$

Since there are cN^7 different copies of $PP(2)$ in Ω then any separating algorithm will require at least $7\log_2 N + O(1)$ steps. In the other direction, it can be shown by probabilistic methods that there is an oblivious algorithm with $5/(\log_2 32/31)\log_2 N < 110\log_2 N$ questions which separates all pairs of disjoint triples in Ω. At present, we have no better upper bound for a corresponding adaptive algorithm (although a much better bound must certainly exist).

We next turn to the general case of k secrets. As before, **S**'s goal is to reach an intersecting family of k-sets, where we start with $K_N^{(k)}$, the complete k-uniform hypergraph on Ω, and we follow the usual partition-and-choose process by **S** and **A** as before. It is easy to see that **A** can preserve any given intersecting family by appropriate choices, namely, always choosing any block of a partition which contains one of the k-set's of the family. While there is a fairly large literature on intersecting families of k-graphs (often called *k-cliques*), relatively little is known.

Let H denote an intersecting family of k-sets in Ω. We say that H is *non-extendible* if any k-set in Ω which is not in H is disjoint from some k-set in H. We say that H has *covering number* k (written $\tau(H) = k$) if any set in Ω which hits every k-set in H must have size at least k. Finally, we say that H has *chromatic number* 3 (written $\chi(H) = 3$) if for any partition of Ω into two sets, $\Omega = \Omega_0 \cup \Omega_1$, some Ω_i contains a k-set of H. The following (strict) implications are well known (see [12]) for intersecting k-graphs H:

$$\chi(H) = 3 \;\Rightarrow\; H \text{ is non-extendible} \;\Rightarrow\; \tau(H) = k.$$

To see the first implication, for example, suppose $\chi(H) = 3$ but there is some k-set $X \subseteq \Omega$ not in H which hits every k-set in H. Color all the points in X red, and all the other points in $\Omega \setminus H$ blue. Since X hits every k-set in H, then every k-set in H has a red point. Thus, H has no red k-set (the only one in Ω is X) and no blue k-set, which contradicts the assumption that $\chi(H) = 3$. A classic result of Erdős and Lovász [12] shows that the number $e(H)$ of edges in an intersecting k-graph H with $\chi(H) = 3$ is bounded. In fact, they show

$$k!(e-1) < \max\{e(H) : \tau(H) = k\} \ \leq \ k^k.$$

The lower bound was recently improved by Frankl, Ota and Tokushige [13] to $((k+1)/2)^(k-1)$. In fact, Erdős and Lovász [12] show that 3-chromatic intersecting k-graphs must have many edges. Their result was improved by Beck [6] and then by Radhakrishnan and Srinivasan [22] recently who showed that for an intersecting k-graph H,

$$\min\{e(H) \ : \ \chi(H) = 3\} \ > \ 0.17\sqrt{\frac{k}{\log k}}\, 2^k.$$

However, if we only require that $\tau(H) = k$, then $e(H)$ can be much smaller. A celebrated (\$500) conjecture of Erdős asserted that

$$\min\{e(H) \ : \ \tau(H) = k\} \ = \ O(k).$$

This was finally proved by Jeff Kahn [16] by a highly non-trivial probabilistic argument.

If we restrict H further, requiring it to be non-extendible, then it is conjectured that the same bound should hold:

Conjecture: (Kahn [16])

$$\min\{e(H) \ : \ H \text{ is non-extendible}\} = O(k).$$

It is known in this case that the following hold.

$$\min\{e(H) \ : \ H \text{ is non-extendible}\} \ \leq \ (1+o(1))k^2, \quad \text{for } k \text{ a prime power (Füredi [14])}$$

$$\min\{e(H) \ : \ H \text{ is non-extendible}\} \ \leq \ k^5, \quad \text{for all } k, \text{ (Blokhuis[7])}.$$

Recall for comparison the classic theorem of Erdős-Ko-Rado [11] which asserts

$$\max\{e(H) \ : \ H \text{ is intersecting }\} \ = \ \binom{N-1}{k-1} \quad \text{for } N \geq 2k.$$

The extremal k-graphs here have $\tau(H) = 1$. We also mention the related bounds of Erdős and Lovász [12], on $v(H)$, the maximum possible number of vertices of an intersecting k-graph H with $\chi(H) = 3$:

$$\frac{1}{2}\binom{2k-2}{k-1} \leq \max\{v(H) \ : \ \chi(H) = 3\} \ \leq \ \frac{1}{2}\binom{2k-1}{k-1}$$

(which are not so far apart!)

9 Concluding remarks

There are numerous questions about guessing secrets that remain open, in particular for the general case of $k \geq 3$ secrets. Here we mention several suggestions by Noga Alon [1] which could provide interesting directions for further work.

The problem of guessing secrets is closely related to the study of small sample spaces supporting k-wise independent (or nearly independent) random variables, which has a rich literature [2, 21, 3, 4]. The problem of interest there is to find a sample space as small as possible, and n binary random variables defined on it, with the property, called k-wise independence, that for any choice of k random variables $X_1, \ldots, X_k$, the probabilities satisfy:

$$Prob(X_1 \ldots X_k = a_1 \ldots a_k) = \frac{1}{2^k}$$

for each of the 2^k binary k-tuples, denoted by $a_1 \ldots a_k$. A somewhat weaker property, called almost k-wise independence, only requires that

$$(1-\epsilon)\frac{1}{2^k} < Prob(X_1 \ldots X_k = a_1 \ldots a_k) < (1+\epsilon)\frac{1}{2^k}.$$

Our problem of guessing secrets for the case of two secrets can be viewed as finding a small sample space satisfying a still weaker condition that the probability of any 4 random variables assuming the values 0011 or 1100 is nonzero. Therefore, the constructions of small sample spaces for almost k-wise independent random variables in, for example, [3] can be used for constructing efficient oblivious algorithms. By using these sample spaces, we can get upper bounds for the minimum number $f_0^{(k)}(N)$ of questions required for an oblivious algorithm giving a separating strategy of guessing k secrets in a space of size N of the form

$$f_0^{(k)}(N) \leq c_k \log N$$

where c_k depends exponentially on k. Moreover, this gives an explicit construction for such oblivious algorithms.

The linear binary error-correcting codes used in the constructions of these sample spaces can be used to provide explicit, oblivious inner product strategies with $O(\log N)$ questions for the case of two secrets. Indeed, it suffices to find a family of t binary vectors F of length $n = \log_2 N$, so that the matrix consisting of their columns generates a binary linear error correcting code consisting of N codewords provides a separating strategy if for any three vectors of length n, denoted by Δ_1, Δ_2 and Δ_3, whose sum (over $GF(2)$) is not the zero vector, and with Δ_2 different from Δ_1, Δ_3 there is a vector $f \in F$ whose inner products with the vectors Δ_i are $0, 1, 0$, respectively. Noga Alon [1] pointed out that the t columns of the generating matrix of any linear binary codes of dimension n and length t in which the weight of every nontrivial code word deviates from half the length by less than $1/14$ the length provides such an F. The known constructions in [3, 21] gives an explicit, oblivious, inner product strategy with $t = O(\log N)$ queries. In fact, the construction described in Section 6 here can be obtained from one of the codes of [4] in the same manner.

By using results from coding theory (or by applying some probabilistic arguments, together with an argument similar to the one used in the study of perfect hash families), the following lower bound for $f_0^{(k)}(N)$ can be derived:

$$f_0^{(k)}(N) \geq c \cdot 2^{2k} \log_2 N,$$

where c is an absolute positive constant.

On the other hand, an easy probabilistic argument shows that (non-explicitly)

$$f_0^{(k)}(N) \leq c'k \cdot 2^{2k} \log_2 N,$$

for some absolute positive constant c'. The same bound follows also from the result in [18].

For the adaptive case, and k secrets, one can derive the lower bound

$$f(N) \geq \Omega((2^{2k}/\sqrt{k}) \cdot \log_2 N)$$

using the bound of Erdös and Lovász mentioned at the end of Section 8.

Applying results from coding theory (using the linear programming bounds together with combinatorial arguments), the following bounds can be obtained for oblivious algorithms for $k = 2$ and 3:

$$f_0^{(2)}(N) > 3.5276 \log_2 N,$$

$$f_0^{(3)}(N) > 15.1862 \log_2 N.$$

One can also study the preceding questions in the cases that questions can have more than two possible answers. Of course, this makes it easier for **S** to deduce information about **A**'s secrets. For example, if **S** can ask just a single question with a 2-bit answer in the inner product scenario, then **S** can always identify some secret of **A** (i.e., **S** can resolve the 2-out-of-3 ambiguity). On the other hand, suppose **A** has a set of $r(t-1)+1$ secrets from which to choose to answer **S**'s question, but each question can now have one of t different answers. Then by a simple majority strategy, **A** can make sure that **S** will never be able to claim that any particular r-element set $T \subset \Omega$ contains one of **A**'s secrets. The preceding analyzes can also be carried out for these cases as well, although not as much is known here. One could also look at other variants, e.g., suppose **A** is allowed to *lie* a certain number (or fraction) of times. Now what should **S** do? These results and many others we hope will be addressed in a subsequent paper.

References

[1] N. Alon, personal communication.

[2] N. Alon, L. Babai and A. Itai, A fast and simple randomized algorithm for the maximal independent set problem, *J. Algorithms*, **7** (1986), 567-583.

[3] N. Alon, J. Bruck, J. Naor, M. Naor and R. Roth, Construction of asymptotically good, low-rate error-correcting codes through pseudo-random graphs, IEEE Transactions on Information Theory, 38 (1992), 509-516.

[4] N. Alon, O. Goldreich, J. Hastad and P. Peralta, Simple constructions of almost k-wise independent random variables, *Random Structures and Algorithms*,**3** (1992), 289-304.

[5] N. Alon and J. H. Spencer, *The Probabilistic Method*, Wiley and Sons, New York, 1992.

[6] J. Beck, On 3-chromatic hypergraphs, *Discrete Math.* **24** (1978), 127-137.

[7] A. Blokhuis, More on maximal intersecting families of finite sets, *J. Combin. Theory* (A) **44** (1987), 299-303.

[8] A. Burgess, On character sums and primitive roots, *Proc. London Math. Soc.* **12** (1962), 179-192.

[9] F. R. K. Chung and R. L. Graham, Quasi-random subsets of Z_n, *J. Comb. Th.* (A) **61** (1992), 365-388.

[10] F. R. K. Chung and R. L. Graham, Quasi-random set systems, *J. Amer. Math. Soc.* **4** (1991), 151-196.

[11] P. Erdős, C. Ko and R. Rado, Intersection theorems for systems of finite sets, *Quart. J. Math. Oxford ser.* (2) **12** (1961), 313-320.

[12] P. Erdős and L. Lovász, Problems and results on 3-chromatic hypergraphs and some related questions, *Infinite and Finite Sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th Birthday)*, vol. II, pp. 609-627. *Colloq. Math. Soc. Janos Bolyai*, vol 10, North-Holland, Amsterdam, 1975.

[13] P. Frankl, K. Ota and N. Tokushige, Covers in uniform intersecting families and a counterexample to a conjecture of Lovász, *J. Comb. Theory* (A), **74** (1996), 33-42.

[14] Z. Füredi, On maximal intersecting families of finite sets, *J. Combin. Theory* (A) **52** (1989), 1-9.

[15] *I've Got a Secret*, a classic '50's television gameshow, see http://www.timvp.ivegotse.html

[16] J. Kahn, On a problem of Erdős and Lovász II: $n(r) = O(r)$, *J. Amer. Math. Soc.* **7** (1994), 125-143.

[17] D. E. Knuth, *The Art of Computer Programming, vol. 3, Sorting and Searching*, Addison Wesley, 1973.

[18] D. J. Kleitman and J. Spencer, Families of k-independent sets, Discrete Mathematics 6 (1973), 255-262.

[19] L. Lu, personal communication.

[20] B. Maggs, personal communication.

[21] J. Naor and M. Naor, Small-bias probability spaces: Efficient constructions and applications, *22nd STOC*, (1990), 213-223.

[22] J. Radhakrishnan and A. Srinivasan, Improved bounds and algorithms for hypergraph 2-coloring, *Random Structures and Algorithms* **16** (2000), 4-32.