Internet Mathematics Vol. 1, No. 2: 177-192

Guessing Secrets with Inner Product Questions

Fan Chung, Ronald Graham, and Linyuan Lu

Abstract. We suppose we are given some fixed (but unknown) subset X of a set $\Omega = \mathbb{F}_2^n$, where $\mathbb{F}_2$ denotes the field of two elements. Our goal is to learn as much as possible about the elements of X by asking certain binary questions. Each "question" Q is just some element of Ω, and the "answer" to Q is just the inner product $Q \cdot x \in \mathbb{F}_2$ for *some* $x \in X$. However, the choice of x is made by a truthful (but possibly malevolent) adversary $\mathbf{A}$, whom we may assume is trying to choose answers so as to yield as little information as possible about X. In this note, we investigate several aspects of this problem. In particular, we are interested in extracting as much information as possible about X from $\mathbf{A}$'s answers. Although $\mathbf{A}$ can prevent us from learning the identity of any particular element of X, with appropriate questions we can still learn quite a bit about X. We determine the maximum amount of information that can be recovered under these assumptions and describe explicit sets of questions for achieving this goal. For the case that $|X| = 2$, we give an $O(n^3)$ algorithm for recovering the desired information. On the other hand, when $|X| \geq 3$, we show that no polynomial-time algorithm can exist for producing a secret set consistent with the answers given, unless $P = NP$.

1. Introduction

The following information-theoretic identification problem was introduced in [Chung et al. 01a, Chung et al. 01b]. A fixed (but unknown) subset X of some finite set Ω is given. A game is played between two players: the "seeker" $\mathbf{S}$ and the "adversary" $\mathbf{A}$. The goal of $\mathbf{S}$ is to learn as much as possible about X by asking $\mathbf{A}$ binary questions. In general, each question can be thought of as some map $Q : \Omega \to \{0, 1\}$, so that Ω is partitioned into $\Omega = Q^{-1}(0) \cup Q^{-1}(1)$.

1542-7951/04 $0.50 per page

For each Q, **A** is allowed to choose some element $x \in X$, and answers Q with the value $Q(x) \in \{0, 1\}$. In this case, **S** knows that $X \cap Q^{-1}(Q(x)) \neq \emptyset$. Of course, **A** could always use the *same* fixed element $x \in X$ to answer *every* question **S** asks, and then **S** would never learn anything about any of the *other* elements of X.

As noticed in [Chung et al. 01a, Chung et al. 01b], with $|X| = k$, **S** can always choose a sufficiently rich set of questions so that no matter how **A** selects the answers, the surviving set of possible k-element sets (or "k-sets") of secrets consistent with all the answers forms an *intersecting* k-uniform hypergraph, i.e., a family $\mathbf{F}$ of k-sets of Ω so that any $F, F' \in \mathbf{F}$ satisfy $F \cap F' \neq \emptyset$. Furthermore, this is the most that **S** can hope to achieve, i.e., once this state is reached, then **A** can prevent any additional k-set from being excluded as a possible secret set. Any set of questions which always results in an intersecting hypergraph will be called a *separating strategy* for **S**.

We point out here that we will only be concerned with *oblivious* strategies for **S**, i.e., those in which all questions must be specified before any answers are given. This can be contrasted to (more powerful) *adaptive* strategies in which the choice of questions can depend on earlier answers (for results on adaptive strategies, consult [Alon et al. 02] and [Chung et al. 01b]).

Our current interest in these questions arose from certain Internet routing algorithms in use by Akamai Technologies. In particular, it was desired to be able to associate with each client IP address the IP addresses of the nameservers the client is using (those are the "secrets"). Unfortunately, in neither the DNS nor the HTTP protocols do the client's IP address and the nameserver's IP address appear together. However, it is possible to gain a very limited amount of information about the nameserver's IP address (e.g., one bit of the address) by clever local routing. Of course, if the client has multiple nameservers, then one does not know which address is supplying the answers. It turns out that very similar problems have occurred previously in the literature in the study of asynchronous sequential machines [Friedman et al. 69] where the relevant concept is that of an (r, s)-separating system. This is just a family $\mathcal{F}$ of subsets of a set Ω such that for any pair of disjoint subsets X and Y of Ω, with $|X| = r$, $|Y| = s$, there is an $F \in \mathcal{F}$ such that F contains one of the sets X and Y, and is disjoint from the other one. Related questions have also arisen in the construction of hash functions and various authentication protocols (see [Alon et al. 02, Cohen et al. 01, Kőrner and Simonyi 88, Segalovich 94]).

In this note, we will take Ω to be $\mathbb{F}_2^n$ for some integer n, where $\mathbb{F}_2$ denotes the field of two elements. Each question Q will be specified by some element of $\mathbb{F}_2^n$, and an answer to the question Q will be the inner product $Q \cdot x$ for some $x \in X$ (all arithmetic will be performed in $\mathbb{F}_2$). One reason for restricting ourselves to

questions of this type is that they can be specified succinctly, that is, with n bits. Of course, the most general questions for $\mathbb{F}_2^n$ require 2^n bits to describe. The price you pay for this restriction is that **S** can no longer guarantee that the final surviving possible secret k-sets will be intersecting. Rather, the best that **S** can hope for, and in fact, which can always be achieved, is that a family of *larger* sets will be intersecting. We next describe these larger sets.

2. Separating Strategies for k Secrets

The first issue we must address is the question of just how much separation can be achieved by inner product questions.

For any k-set $X = \{X_1, \ldots, X_k\} \subseteq \mathbb{F}_2^n$, define

$$\mathrm{Odd}(X) = \{\sum_{i=1}^{k} \epsilon_i X_i \ : \ \epsilon_i \in \mathbb{F}_2 \quad \text{and} \quad \sum_{i=1}^{k} \epsilon_i = 1\}$$

(where addition in the first sum is taken on $\mathbb{F}_2^n$, and addition in the second sum is in $\mathbb{F}_2$).

Lemma 2.1. *For k-sets $X = \{X_1, X_2, \ldots, X_k\}$ and $Y = \{Y_1, Y_2, \ldots, Y_k\}$ in $\mathbb{F}_2^n$, the following conditions are equivalent:*

(i) $X_1 + Y_1 \notin \langle X_1 + X_2, \ldots, X_{k-1} + X_k, Y_1 + Y_2, \ldots, Y_{k-1} + Y_k \rangle$
(where $\langle w_1, \ldots, w_i, \ldots, w_r \rangle$ denotes the vector space over $\mathbb{F}_2$ spanned by the w_i's);

(ii) *There exists $Q \in \mathbb{F}_2^n$ such that*

$$Q \cdot X_1 = Q \cdot X_2 = \cdots = Q \cdot X_k \neq Q \cdot Y_1 = Q \cdot Y_2 = \cdots = Q \cdot Y_k;$$

(iii) $Odd(X) \cap Odd(Y) = \emptyset$.

Proof. (i) $\Rightarrow$ (ii).
Define $\Delta'_0 = X_1 + Y_1$ and $\Delta_i = X_i + X_{i+1}, \Delta_{k-1+i} = Y_i + Y_{i+1}, 1 \leq i \leq k-1$. Choose a basis for $W = \langle \Delta_1, \Delta_2, \ldots, \Delta_{2k-2} \rangle$, say $W = \langle \Delta'_1, \ldots, \Delta'_r \rangle$ where each Δ'_i is some Δ_j. Since $\Delta'_0 \notin W$ by hypothesis, then the matrix

$$\Delta' = \begin{bmatrix} \Delta'_0 \\ \Delta'_1 \\ \cdot \\ \cdot \\ \cdot \\ \Delta'_r \end{bmatrix}$$

has rank $r+1$ over $\mathbb{F}_2$. Hence, there exists an $n \times (r+1)$ matrix D satisfying

$$\Delta' D = I_{r+1},$$

where I_{r+1} is the $(r+1) \times (r+1)$ identity matrix. In particular, the first column D_1 of D satisfies

$$\Delta' \cdot D_1 = \begin{bmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}.$$

Since all the $\Delta_i, 1 \leq i \leq 2k-2$, are linearly dependent on the Δ'_j, $1 \leq j \leq r$, then $\Delta_i \cdot D_1 = 0, 1 \leq i \leq 2k-2$, while $\Delta'_0 \cdot D_1 = 1$. Thus, (ii) holds as required.

(ii) $\Rightarrow$ (iii).

Suppose $\mathrm{Odd}(X) \cap \mathrm{Odd}(Y) \neq \emptyset$. Thus, there exist $\delta_i, \epsilon_i \in \mathbb{F}_2$ such that

$$\begin{aligned} \sum_{i=1}^{k} \delta_i &= 1 = \sum_{i=1}^{k} \epsilon_i \qquad \text{and} \\ \sum_{i=1}^{k} \delta_i X_i &= \sum_{i=1}^{k} \epsilon_i Y_i. \end{aligned}$$

However, we can write

$$\sum_{i=1}^{k} \delta_i X_i = X_1 + \sum_{i=1}^{k-1} \delta'_i (X_i + X_{i+1}),$$

where $\delta'_i = 1 + \sum_{j=1}^{i} \delta_i$, since $\delta'_{k-1} = \delta_k$. Thus, (using the same argument for the Y_i), we have

$$X_1 + \sum_{i=1}^{k-1} \delta'_i (X_i + X_{i+1}) = Y_1 + \sum_{i=1}^{k-1} \epsilon'_i (Y_i + Y_{i+1}), \qquad \delta'_i, \epsilon'_i \in \mathbb{F}_2.$$

This now immediately implies the negation of (ii).

(iii) $\Rightarrow$ (i).
Observe that

$$\begin{aligned}
& \Delta_0' \in \langle \Delta_1, \ldots, \Delta_{2k-2} \rangle \\
\Rightarrow \quad & X_1 + Y_1 = \sum_{i=1}^{k-1} \delta_i (X_i + X_{i+1}) + \sum_{i=1}^{k-1} \epsilon_i (Y_i + Y_{i+1}), \qquad \delta_i, \epsilon_i \in \mathbb{F}_2, \\
\Rightarrow \quad & X_1 + \sum_{i=1}^{k-1} \delta_i (X_i + X_{i+1}) = Y_1 + \sum_{i=1}^{k-1} \epsilon_i (Y_i + Y_{i+1}) \\
\Rightarrow \quad & X_1(1+\delta_1) + X_2(\delta_1 + \delta_2) + \cdots + X_{k-1}(\delta_{k-2} + \delta_{k-1}) + X_k \delta_{k-1} \\
& = Y_1(1+\epsilon_1) + Y_2(\epsilon_1 + \epsilon_2) + \cdots + Y_{k-1}(\epsilon_{k-2} + \epsilon_{k-1}) + Y_k \epsilon_{k-1} \\
\Rightarrow \quad & \mathrm{Odd}(X) \cap \mathrm{Odd}(Y) \neq \emptyset.
\end{aligned}$$

We have shown (i) $\Rightarrow$ (ii), (ii) $\Rightarrow$ (iii), and (iii) $\Rightarrow$ (i), so Lemma 2.1 is proved. □

Observe now that two k-sets X and Y can be "separated" by some question Q if and only if (ii) holds. In that case, whichever answer is given by $\mathbf{A}$, one of the two k-sets is eliminated of a possible k-set of secrets. Hence, by Lemma 2.1, X and Y *cannot* be separated by any inner product question if and only if $\mathrm{Odd}(X) \cap \mathrm{Odd}(Y) \neq \emptyset$. This proves

Theorem 2.2. *By using suitable inner product questions,* $\mathbf{S}$ *can guarantee that for the family* $\mathcal{X}$ *of surviving possible* k*-sets, i.e., consistent with all the answers given, the family* $Odd(\mathcal{X}) = \{Odd(X) : X \in \mathcal{X}\}$ *is an intersecting family. Furthermore, this is the most that* $\mathbf{S}$ *can guarantee. That is, for any family* $\mathcal{X}'$ *such that* $Odd(\mathcal{X}')$ *is intersecting,* $\mathbf{A}$ *can always answer in such a way that all* $X \in \mathcal{X}'$ *survive.*

We will call any such set of questions a *weakly* separating strategy for $\mathbf{A}$. Our next goal will be to exhibit a simple explicit weakly separating strategy.

For a positive integer m, define $\mathcal{F}(m) \subseteq \mathbb{F}_2^n$ by:

$$\mathcal{F}(m) = \{X \in \mathbb{F}_2^n : X \text{ has at least one and at most } m \text{ coordinates equal to } 1\}.$$

In other words, $\mathcal{F}(m)$ consists of all $X \in \mathbb{F}_2^n$ with "weight" $w(X)$ satisfying $1 \leq w(X) \leq m$.

Theorem 2.3. $\mathcal{F}(2k-1)$ *is a weakly separating strategy for secret sets of size* k*. However, this is not true for* $\mathcal{F}(2k-2)$*.*

Proof. Suppose $\mathcal{X}$ is a family of k-sets so that $\mathrm{Odd}(\mathcal{X})$ is not intersecting. Thus, there are $X = (X_1, X_2, \ldots, X_k) \in \mathcal{X}$, $Y = (Y_1, Y_2, \ldots, Y_k) \in \mathcal{X}$ such that $\mathrm{Odd}(X) \cap \mathrm{Odd}(Y) = \emptyset$. Adopting the notation of Lemma 2.1, then by Lemma 2.1, the matrix Δ' has row rank $r+1$. Hence, it also has column rank $r+1$, which implies there are $r+1$ columns of Δ', say $\Delta'(a_1), \ldots, \Delta'(a_{r+1})$ which are linearly independent over $\mathbb{F}_2$. Thus, there are $\epsilon_i \in \mathbb{F}_2$ such that

$$\sum_{i=1}^{r+1} \epsilon_i \Delta'(a_i) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{r+1}.$$

Since $\Delta_1, \Delta_2, \ldots, \Delta_{2k-2}$ are all linearly dependent on the $\Delta'_i, 1 \le i \le r$, there we have

$$\sum_{i=1}^{r+1} \epsilon_i \Delta(a_i) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{2k-1}$$

where $\Delta(j)$ denotes the j^{th} column of the matrix

$$\Delta = \begin{bmatrix} \Delta'_0 \\ \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_{2k-2} \end{bmatrix}.$$

Hence, the question $Q = (Q_1, Q_2, \ldots, Q_r)$ with

$$Q_j = \begin{cases} 1 & \text{if } j = a_i \text{ and } \epsilon_i = 1, 1 \le i \le r+1, \\ 0 & \text{otherwise,} \end{cases}$$

satisfies

$$Q \cdot \Delta'_0 = 1, \qquad Q \cdot \Delta_i = 0,\ 1 \le i \le 2k-2.$$

Thus, the question Q can be used to separate X and Y. However, Q only has weight $w(Q) \le r+1 \le 2k-1$. Applying the argument recursively, we see that we must eventually arrive at a family $\mathcal{X}'$ with $\mathrm{Odd}(\mathcal{X}')$ intersecting, using only questions with weight at most $2k-1$. This shows that $\mathcal{F}(2k-1)$ is a weakly separating strategy for secret sets of size k.

To show that $\mathcal{F}(2k-2)$ need not be weakly separating, let $n = 2k-1$ and consider the two k-sets $X = \{X_1, X_2, \ldots, X_k\}$ and $Y = \{Y_1, Y_2, \ldots, Y_k\}$ defined by:

$$X_i = (X_i(1), X_i(2), \ldots, X_i(2k-1)),\ 1 \le i \le k,$$

$$\text{with } X_i(j) = \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{otherwise} \end{cases}$$

and

$$Y_i = (Y_i(1), Y_i(2), \ldots, Y_i(2k-1)),\ 1 \le i \le k,$$

$$\text{with } Y_i(j) = \begin{cases} 0 & \text{if } j = k-1+i, \\ 1 & \text{otherwise.} \end{cases}$$

First, observe that $x \in \mathrm{Odd}(X) \Rightarrow w(x)$ is odd and $y \in \mathrm{Odd}(Y) \Rightarrow w(y)$ is even. This shows that $\mathrm{Odd}(X) \cap \mathrm{Odd}(Y) = \emptyset$. We now claim that X and Y cannot be separated by any question Q with $w(Q) < 2k-1$. For suppose $Q = (Q_1, Q_2, \ldots, Q_{2k-1})$, and Q separates X and Y. This means that

$$Q \cdot X_1 = Q \cdot X_2 = \cdots = Q \cdot X_k \neq Q \cdot Y_1 = Q \cdot Y_2 = \cdots = Q \cdot Y_k.$$

However, this implies $Q_1 = Q_2 = \cdots = Q_k$ (because of the way the X_i are defined) and $Q_k = Q_{k+1} = \cdots = Q_{2k-1}$ (because of the way the Y_i are defined). Thus, the only such Q which can separate X and Y is the all 1s question, which has weight $2k-1$. This completes the proof of Theorem 2.3. □

3. Inverting the Answers for $k = 2$

In this section, we restrict our attention to the case $k = 2$ with $\mathcal{F}(3) = \{Q \in \mathbb{F}_2^n : 1 \le w(F) \le 3\}$ as our separating strategy. Since $|X| = 2 \Rightarrow \mathrm{Odd}(X) = X$, then any two surviving pairs must be intersecting, i.e., must share a common element. Thinking of pairs of elements of $\mathbb{F}_2^n$ as *edges* of a graph with vertex set $\mathbb{F}_2^n$, then the only possible intersecting sets are a *star* with some center X_0, or a *triangle* T on three vertices $\{X_1, X_2, X_3\}$. In the first case, it follows that X_0 must be one of **A**'s secrets. In the second case, **S** can only conclude that **A**'s secret pair is either $\{X_1, X_2\}$, $\{X_1, X_3\}$, or $\{X_2, X_3\}$ (and in particular, cannot assert that any specific element is in **A**'s secret set).

We will now describe a recursive algorithm ALG for inverting the answers to $\mathcal{F}(3)$ which runs in time $O(n^3)$ on $\Omega = \mathbb{F}_2^n$. We will assume (inductively on n) that ALG on $\mathbb{F}_2^n$ (denoted by ALG(n)) gives the following information on the surviving intersecting set E of edges:

(i) E is a *star* with some center X_0 (but no other information about the edges in the star), or

(ii) E is a *triangle* on the set $\{X_1, X_2, X_3\}$. In this case, all edges $X_iX_j, i \neq j$, have survived.

Note that in case (i), E might consist of a single edge, in which case X_0 could be either endpoint. Also note for $n \leq 3$, we can determine the information (star center or triangle) needed by ALG in $O(1)$ questions. (More precisely, since there are at most $2^3 = 8$ points, and so at most $\binom{8}{2} = 28$ edges, and $|\mathcal{F}(3)| \leq 7$, then testing each possible edge requires at most 196 questions.) The main idea of ALG(n) is the following: For some $n > 3$, we first ask all the $\binom{n}{1} + \binom{n}{2} + \binom{n}{3}$ questions in $\mathcal{F}(3)$.

We next partition $[n] := \{1, 2, \ldots, n\}$ into three subsets:

$$J_1 := \{1 \leq x < n/3\}, J_2 = \{n/3 \leq x < 2n/3\}, J_3 = \{2n/3 \leq x \leq n\},$$

and we define three (overlapping) complementary subsets $I_k \subseteq [n]$ by:

$$I_k = [n] \setminus J_k, \qquad 1 \leq k \leq 3.$$

We then examine the answers given to the questions which are supported entirely in I_k, $k = 1, 2, 3$. This corresponds to executing ALG on the smaller index set I_k, and we denote the output of these by ALG(I_k). By induction, this will be either a star center or a triangle for each value of k. We will then show that it will always be possible to put this partial information together and recover the full information (star center or triangle) required by ALG(n).

We first introduce some notation. If $X \in \mathbb{F}_2^n$, then we can write $X = X_1X_2X_3$, where X_k denotes the restriction of X to J_k. Further, we denote the restriction of X to I_1 by $X|I_1$, and we write this as $X|I_1 = *X_2X_3$ (where the * denotes the fact that I_1 omits the coordinates in J_1). Similarly, $X|I_2 = X_1 * X_3$ and $X|I_3 = X_1X_2*$.

Suppose E is the set of (intersecting) edges which survive after executing ALG(n), i.e., any $\{U, V\} \in E$ is consistent with the answers given to all the questions in $\mathcal{F}(3)$ asked by $\mathbf{S}$. How will such an edge show up in the output of ALG(I_1), for example? It is easy to see that if ALG(I_1) outputs $*A_2A_3$ as a star center, then either $U|I_1 = *A_2A_3$ or $V|I_1 = *A_2A_3$ (with similar remarks applying to I_2 and I_3). Also, it is clear that if ALG(I_1) outputs a triangle $\{*A_2A_3, *B_2B_3, *C_2C_3\}$, then both $U|I_1$ and $V|I_1$ are vertices of this triangle (with similar remarks applying to I_2 and I_3.)

Since each I_k has fewer than n elements, we know by induction that each ALG(I_k) outputs either a star center or a triangle. Our argument will have

three cases, namely, if *none* of these outputs is a triangle, if exactly *one* of them is a triangle, or if *at least two* are triangles.

Case 1: ALG(I_1) outputs a star center $A = *A_2A_3$,
ALG(I_2) outputs a star center $B = B_1 * B_3$,
ALG(I_3) outputs a star center $C = C_1C_2 *$.

In this case, we will show that ALG(n) must also output a star center. Let us call two restrictions, say $*A_2A_3$ and $B_1 * B_3$, *compatible*, if they agree where they overlap, i.e., $A_3 = B_3$ in this case. We can define the *compatibility graph* H on the vertex set $\{*A_2A_3, B_1 * B_3, C_1C_2*\}$ with edges joining each pair of compatible vertices.

By the assumption in this case, for any surviving edge $\{U, V\} \in E$ in ALG(n) with $U = U_1U_2U_3$, $V = V_1V_2V_3$, we must have:

$$\begin{aligned} * A_2A_3 &\in \{* U_2U_3 \ , * V_2V_3\}, \\ B_1 * B_3 &\in \{U_1 * U_3 \ , V_1 * V_3\}, \\ C_1C_2 * &\in \{U_1U_2 * \ , V_1V_2 *\}. \end{aligned}$$

That is, the restriction $\{U|I_i, V|I_i\}$ of $\{U, V\}$ must survive in ALG(I_i), i=1,2,3. Thus, some point, say U, must be chosen for at least two of the choices above. Without loss of generality, let us assume

$$*A_2A_3 = *U_2U_3, B_1 * B_3 = U_1 * U_3,$$

i.e.,

$$U_2 = A_2, U_3 = A_3, U_1 = B_1, U_3 = B_3 = A_3.$$

This implies that $*A_2A_3$ and $B_1 * B_3 = B_1 * A_3$ are compatible, so that H will always have at least one edge which comes from an element of $\{U, V\}$ (in this case, $U = B_1A_2A_3$). If H has just one edge, then U must belong to every edge of E in ALG(n) so that U is a star center in this case.

Similarly, if H has three edges, then the vertices must have the form $*A_2B_3$, $B_1 * A_3$, B_1A_2*. However, this implies as before that $B_1A_2A_3 \in \{U, V\}$, and so is a star center in ALG(n).

So we are left with the case that H has exactly two edges, say the vertices of H are $*A_2A_3, B_1*A_3, C_1A_2*$ with $B_1 \neq C_1$. Hence, in this case, we can conclude that *either* $B_1A_2A_3$ is in $\{U, V\}$ and any other "mate" (i.e., the other element possibly paired with $B_1A_2A_3$) has the form C_1A_2*, *or* $C_1A_2A_3$ is in $\{U, V\}$ and any other mate has the form B_1*A_3. Our next task is to resolve this uncertainty.

Since $B_1 \neq C_1$, there must be some coordinate $i_0 \in J_1$ such that $B_1(i_0) \neq C_1(i_0)$ (i.e., B_1 and C_1 differ in their i_0^{th} coordinates). We now examine the set

of $|J_2||J_3|$ answers to the questions $Q_{i_0,j,k}$, $j \in J_2$, $k \in J_3$, where $Q_{i_0,j,k}$ denotes the vector in $\mathcal{F}(3)$ having 1's in position i_0, j and k.

Suppose for some $j \in J_2$, $k \in J_3$ that $B_1(i_0) + A_2(j) + A_3(k) = \alpha \in \{0,1\}$. Thus, $C_1(i_0) + A_2(j) + A_3(k) = 1 - \alpha$. If **A** answers $Q_{i_0,j,k}$ with the answer α, then any point of the form $B_1D_2A_3$ with $D_2(j) \neq A_2(j)$ is ruled out as a possible mate for $C_1A_2A_3$ (since in this case, both points $C_1A_2A_3$ and $B_1D_2A_3$ would give the answer $1-\alpha$ to this question). By the same token, if **A** answers $Q_{i_0,j,k}$ with the answer $1-\alpha$, then any point of the form $C_1A_2D_3$ with $D_3(k) \neq A_3(k)$ is ruled out as a possible mate for $B_1A_2A_3$. This now implies that at the end of this process, one of the points $B_1A_2A_3$ and $C_1A_2A_3$ will have all possible mates ruled out, except for its "trivial" mate, i.e., $B_1A_2A_3$ and $C_1A_2A_3$ are trivial mates.

For, if $B_1D_2A_3$ and $C_1A_2D_3$ survived as possible mates of $C_1A_2A_3$ and $B_1A_2A_3$, respectively, with $D_2(j_0) \neq A_2(j_0)$, $D_3(k_0) \neq A_3(k_0)$ for some $j_0 \in J_2$, $k_0 \in J_3$, then the question Q_{i_0,j_0,k_0} would clearly rule out one of these possibilities. So, if, for example, the only possible mate for $B_1A_2A_3$ is $C_1A_2A_3$ (whereas $C_1A_2A_3$ had more possible mates than $B_1A_2A_3$), then $C_1A_2A_3$ is the desired star center output of ALG(n). Even if $B_1A_2A_3$ and $C_1A_2A_3$ end up having only each other as possible mates (so that only one edge survived in E), the choice of $C_1A_2A_3$ works. This concludes the proof for Case 1.

Case 2: ALG(I_1) outputs a star center $A = *A_2A_3$,
ALG(I_2) outputs a star center $B = B_1 * B_3$,
ALG(I_3) outputs a triangle $\{P_1P_2*, Q_1Q_2*, R_1R_2*\}$.

Suppose $\{U,V\} \in E$. There are two possibilities.

(i) $A_3 \neq B_3$. Then $*A_2A_3$ and $B_1 * B_3$ are restrictions of *different* points in $\{U,V\}$, for example, $U|I_1 = *A_2A_3$, $V|I_2 = B_1 * B_3$. Since $\{P_1P_2*, Q_1Q_2*, R_1R_2*\}$ contains the I_3 restrictions of both U and V, then there is only a small number of possible pairs to test (on all answers to the questions in $\mathcal{F}(3)$) to discover the surviving edges, which still could be a star or a triangle.

(ii) $A_3 = B_3$. Then we claim $B_1A_2A_3$ must belong to every surviving edge in E for ALG(n). For suppose not. Then the two restrictions $*A_2A_3$ and B_1*A_3 must come from different points, say $U|I_1 = *A_2A_3$ and $V|I_2 = B_1 * A_3$. Since ALG(I_3) yields the triangle $\{P_1P_2*, Q_1Q_2*, R_1R_2*\}$, then each of the pairs $\{P_1P_2*, Q_1Q_2*\}$, $\{P_1P_2*, R_1R_2*\}$, and $\{Q_1Q_2*, R_1R_2*\}$ must be a possible surviving pair for ALG(I_3). However, each of these pairs must be equal to $\{U|I_3, V|I_3\}$. This implies that two of P_1P_2, Q_1Q_2, R_1R_2 must be equal, which is a contradiction. Thus, in this case, $B_1A_2A_3$ is a star center for ALG(n), and case (ii) is done.

Case 3: At least two of the restricted outputs are triangles, say

$$\begin{aligned} \mathrm{ALG}(I_1) &= \{*A_2A_3, *B_2B_3, *C_2C_3\} \\ \mathrm{ALG}(I_2) &= \{P_1 * P_3, Q_1 * Q_3, R_1 * R_3\}. \end{aligned}$$

In this case, all possible secrets can be found by combining compatible pairs from $\mathrm{ALG}(I_1)$ and $\mathrm{ALG}(I_2)$. Each can be tested against all the answers to see if it survives in $\mathrm{ALG}(n)$. This completes Case 3, and the proof of the induction step of the algorithm. If $c(m)$ denotes the number of comparisons needed for $\mathrm{ALG}(n)$, then it follows from the preceding analysis that

$$c(n) \leq 3 \cdot c(\frac{2n}{3}) + \binom{9}{2}\binom{n}{3} + o(n^3), \tag{3.1}$$

which implies $c(n) \leq 54n^3 + o(n^3)$.

Our implementation of this algorithm (available upon request) easily handles values of n around 500 in a few seconds on a standard Unix workstation.

4. Finding a Valid Secret Set

The algorithm described in the preceding section will always identify either a star center (which must belong to every valid secret pair) or a triangle (in which case, the valid secret pairs correspond to the three edges of the triangle). In the first case, however, it does not automatically produce a specific secret pair, i.e., a viable mate for the star center. It is easy to extend the algorithm to achieve this goal, however, as follows. Suppose X is identified as the star center. Let Y denote some potential mate for X, i.e., the pair $\{X, Y\}$ satisfies all the answers $\mathcal{A}(Q)$ given by $\mathbf{A}$ to each question $Q \in \mathcal{F}(3)$. Now, if $Q \cdot X \neq \mathcal{A}(Q)$, then we must have $Q \cdot Y = \mathcal{A}(Q)$. This represents a linear constraint on the coordinates of Y. The collection of all the linear constraints arising from all Q such that $Q \cdot X \neq \mathcal{A}(Q)$ forms a system of linear equations in the coordinates of Y. It is now straightforward to find (by Gaussian elimination, for example) the desired Y (and conclude that no such Y exists in case X, in fact, was not a star center). This augmented algorithm clearly still runs in polynomial time.

For $k \geq 3$, however, the solution appears to be quite different. Namely, there is strong evidence that no such polynomial time algorithm exists for finding a valid secret k-set. To see this, we focus on the case $k = 3$.

Suppose $\mathcal{F}$ is some weak separating strategy for $\mathbb{F}_2^n$ with $k = 3$. Let $Q_{i,j}$ denote the weight 2 question (=vector) having 1s in positions i and j with $i < j$. Denote by $\mathcal{F}^+$ the strategy $\mathcal{F} \cup \{Q_{i,j} : 1 \leq i < j \leq n\}$. Let G denote a given

graph with vertex set $[n] = \{1, 2, \ldots, n\}$ and edge set $E(G)$. We now suppose that the answer $\mathcal{A}_G(Q)$ that $\mathbf{A}$ gives to each question $Q \in \mathcal{F}^+$ is as follows:

$$\mathcal{A}_G(Q) = \begin{cases} 1 & \text{if } Q = Q_{i,j} \text{ and } \{i,j\} \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

Claim 4.1. *There is a valid secret triple of the form $\{0, X, Y\}$ if and only if G is 4-chromatic.*

Proof. First, assume G is 4-chromatic. There we can partition the vertex set $[n] = C_1 \cup C_2 \cup C_3 \cup C_4$ so that every edge in $E(G)$ has endpoints in different C_i. Define two points $X = (X(1), X(2), \ldots, X(n)) \in \mathbb{F}_2^n$, $Y = (Y(1), Y(2), \ldots, Y(n)) \in \mathbb{F}_2^n$, as follows:

$$X(i) : = \begin{cases} 1 & \text{if } i \in C_1 \cup C_2 \\ 0 & \text{if } i \in C_3 \cup C_4 \end{cases}$$

$$Y(i) : = \begin{cases} 1 & \text{if } i \in C_1 \cup C_3 \\ 0 & \text{if } i \in C_2 \cup C_4. \end{cases}$$

It is now easy to check that $\{0, X, Y\}$ is a valid secret triple for the answers $\mathcal{A}_G(Q)$ supplied by $\mathbf{A}$.

For the other direction, assume $\{0, P_1, P_2\}$ is a valid secret triple for $\mathbf{A}$'s answers. Define the two sets

$$A_i := \{j \colon P_i(j) = 1\},$$

and let $B_i := [n] \setminus A_i$, $i = 1, 2$. Let $G_i = G_i(A_i, B_i)$ denote the complete bipartite graph on the vertex sets A_i and B_i, $i = 1, 2$. Then $\mathcal{A}(Q_{i,j})$ can only be 1 if $\{i, j\}$ is an edge in either G_1 or G_2 (or both). In particular, since $\{0, P_1, P_2\}$ is valid, every edge of G must be an edge of G_1 or G_2, which implies G is 4-chromatic. This proves Claim 4.1. □

Observe that if $\{X, Y, Z\}$ is a valid secret triple, then *any* 3-element subset of $\mathrm{Odd}(X, Y, Z) = \{X, Y, Z, X + Y + Z\}$ also is.

Now suppose there is a polynomial time algorithm which can produce some solution $\{X, Y, Z,\ X + Y + Z\}$ as an Odd 4-set satisfying the specified answers $\mathcal{A}_G(\mathcal{F}^+)$. Thus, $\mathcal{F}^+$ must have polynomial size. If $0 \in \{X, Y, Z, X + Y + Z\}$, then by the preceding remarks, we can conclude that G is 4-chromatic.

On the other hand, suppose $0 \notin \{X, Y, Z, X + Y + Z\}$. Since $\mathcal{F}$ (and therefore $\mathcal{F}^+$) is a weak separating strategy, then any two satisfying Odd 4-sets must intersect. Thus, if G is 4-chromatic, then there must be a satisfying Odd 4-set of the form $\{0, C, D, C + D\}$, and this set must intersect $\{X, Y, Z, X + Y + Z\}$.

Since $0 \notin \{X, Y, Z, X+Y+Z\}$, then at least one of C, D, or $C+D$ must be equal to one of X, Y, Z or $X+Y+Z$. However, each one of these possibilities can be checked in polynomial time by the method previously described for finding a star center mate when $k = 2$. Namely, for example, assume $C = X$. So we must check whether there exists a W so that $\{0, X, W\}$ is a valid triple, etc. Consequently, we see that G is 4-chromatic if and only if the process succeeds if *some* element of $\{X, Y, Z, X+Y+Z\}$ can be successfully paired with 0 (and generating a compatible third element W).

Thus, our hypothesized polynomial time algorithm for finding a valid secret triple from these answers $\mathcal{A}_G(\mathcal{F}^+)$ could be used to determine whether or not G is 4-chromatic. However, it is well known [Garey and Johnson 79] that this implies $P = NP$, a statement not widely believed.

This is the basis for our skepticism in the existence of a general polynomial time algorithm for producing valid secret k-sets for $k \geq 3$.

5. Concluding Remarks

We conclude with several remarks for our original secret guessing game with $\Omega = [N]$, and any $Q : \Omega \to \{0, 1\}$ allowed as a question. Let $f_k(N)$ denote the minimum number of queries needed for any **adaptive** separating strategy (with k-element secret sets), and let $g_k(N)$ denote the corresponding minimum for *nonadaptive* strategies. It was shown in [Chung et al. 01a] that

$$3\log_2 N - 5 \leq f_2(N) \leq 4\log_2 N + 3, N > 2.$$

We still do not know the truth here.

For nonadaptive strategies, it can be shown that

$$3.5276\log_2 N < g_2(N) < \frac{3}{\log_2(8/7)}\log_2 N < 15.573\log_2 N.$$

The lower bound is due to N. Alon [Alon et al. 02] (and uses coding theory and linear programming bounds). The upper bound comes from a probabilistic construction.

In [Alon et al. 02], Alon, Guruswami, Kaufman, and Sudan give a beautiful construction of an explicit set of $O(\log N)$ queries for which the desired intersecting graph (star center or triangle) can be recovered in time $O(\log^3 N)$. They use techniques based on small ϵ-biased spaces and list decoding. Is it possible to construct such small sets of queries which can be "inverted" in time $O(\log N)$?

We remark that a straightforward probabilistic argument shows that

$$g_3(N) \leq \frac{5}{\log_2(32/31)}\log_2 N < 109.16\log_2 N.$$

Very recently, Vukičević [Vukicevic 02] has shown this can be reduced for adaptive strategies to

$$f_3(N) < \frac{2 + \frac{1}{2}\log_2 43}{\frac{3}{2} + \log_2 \frac{32}{31}} \log_2 N + O(1).$$

(The coefficient is ≈ 104.4.)

We should point out here that even the problem of explicitly describing the possible families of intersecting k-sets is not simple when $k > 2$. Of course, for $k = 2$, we just have either a star or a triangle. We can represent these as:
(i) $1x$ where 1 denotes some fixed element of Ω, and x denotes any other element of Ω, and
(ii) $12, 13, 23$ where $1, 2$, and 3 denote any three distinct elements of Ω.

The first case is an example of an *extendible* intersecting family, while the second is an example of a *nonextendible* intersecting family.

For $k = 3$, the situation is much more complicated. The complete list of maximal intersecting families of 3-uniform hypergraphs is given in the following list (where (i)-(vii) are extendible):

(i) $1xy$ (i.e., all triples containing some fixed element)

(ii) $12x, 13y, 14z$

(iii) $123, 14x, 24y, 34z$

(iv) $145, 234, 235, 12x, 13y$

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

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

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

(viii) $123, 145, 167, 246, 257, 347, 356$, the seven lines (= triples) of $PP(2)$, the projective plane of order 2.

(ix) $123, 145, 167, 246, 247, 346, 356$

(x) Any set of 10 triples from $\{1, 2, 3, 4, 5, 6\}$ which does not contain a triple and its complement, and which is not one of the previous cases. There are five of these, as was shown by Frankl, Ota, and Tokushige [Frankl et al. 96].

This list incorporates the helpful remarks of Matthew Cook [Cook 02], who pointed out that the earlier version of this, given in [Chung et al. 01a], was incomplete. At present, a characterization of maximal intersecting k-uniform hypergraphs seems out of reach for $k > 3$.

Of course, there remain many variants of these problems which have not been investigated. For example, what happens if we allow queries with more than two possible answers? Or what if the adversary is allowed to be untruthful some number (or fraction) of times? Or what if the adversary chooses answers probabilistically? Clearly much remains to be done.

Acknowledgments

An extended abstract of this paper has appeared in the *Proceedings of the Thirteenth SIAM-ACM Symposium on Discrete Algorithms*, (2002), 247–253. The research of Fan Chung and Linyuan Lu was supported in part by NSF Grant No. DMS 0100472 and ITR 0205061.

References

[Alon et al. 02] N. Alon, V. Guruswami, T. Kaufman, and M. Sudan. "Guessing Secrets Efficiently via List Decoding." In *Proceeding of the Thirteenth SIAM-ACM Symposium on Discrete Algorithms*, pp. 254–262, Philadelphia: SIAM, 2002.

[Alon 02] N. Alon. Private communication, 2002.

[Chung et al. 01a] F. Chung, R. Graham, and F. T. Leighton. "Guessing Secrets (Extended Abstract). In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms,* pp. 723–726, Philadelphia: SIAM, 2001.

[Chung et al. 01b] F. Chung, R. Graham, and F. T. Leighton. "Guessing Secrets." *Electronic Journal of Combinatorics* 8 (2001), #R13.

[Cohen et al. 01] G. D. Cohen, S. B. Encheva, and H. G. Schaathun. "On Separating Codes." Technical Report, ENST, 2001.

[Cook 02] Matthew Cook. Private communication, 2002.

[Edelman 99] A. Edelman. "Akamai Technologies: A Mathematical Success Story." *SIAM News* 32:1 (1999), 12–13. Available from World Wide Web (http://www.siam.org/siamnews/12-99/akamai.pdf), 1999.

[Frankl et al. 96] 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.

[Friedman et al. 69] A. D. Friedman, R. L. Graham, and J. D. Ullman. "Universal Single Transition Time Asynchronous State Assignments." *IEEE Trans. Comput.* C-18 (1969), 541–547.

[Garey and Johnson 79] M. R. Garey and D. S. Johnson. *Computer and Intractability, A Guide to the Theory of NP-Completeness.* San Francisco: W. H. Freeman and Co., 1979.

[Kőrner and Simonyi 88] J. Kőrner and G. Simonyi. "Separating Partition Systems and Locally Different Sequences." *SIAM J. Discrete Math.* 1 (1988), 355–359.

[Segalovich 94] Y. L. Segalovich. "Separating Systems." *Problems of Information Transmission* 30:2 (1994), 105–123.

[Vukicevic 02] D. Vukičević. "A Note on Guessing Secrets." Preprint, 2002.

Fan Chung, University of California, San Diego, La Jolla, California (fan@euclid.ucsd.edu)

Ronald Graham, University of California, San Diego, La Jolla, California (graham.ucsd.edu)

Linyuan Lu, University of California, San Diego, La Jolla, California (llu@math.ucsd.edu)

Received June 16, 2003; accepted June 19, 2003.