GUESSING SECRETS
WITH INNER PRODUCT QUESTIONS

Fan Chung* Ronald Graham † Linyuan Lu

Abstract

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. We would like to learn as much as possible about the elements of X by asking certain binary questions. Each "question" Q is 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 **A**, whom we may assume is trying to choose answers so as to yield as little information as possible about X. In this paper, we study various aspects of this problem. In particular, we are interested in extracting as much information as possible about X from **A**'s answers. Although **A** can prevent us from learning the identity of any particular element of X, with appropriate questions we can still learn a lot about X. We determine the maximum amount of information that can be recovered and discuss the optimal strategies for selecting questions. For the case that $|X| = 2$, we give an $O(n^3)$ algorithm for an optimal strategy. However, for the case that $|X| \geq 3$, we show that no such polynomial-time algorithm can exist, unless $P = NP$.

1 Introduction and background

The following information-theoretic identification problem was introduced in [1, 2]. A fixed (but unknown) subset X of some finite set Ω is given. A game is played between two players: the "seeker" **S** and the "adversary" **A**. The object of the seeker **S** is to learn as much as possible about the elements of X by asking **A** binary questions. Each question is in general just some map $Q : \Omega \to \{0, 1\}$, so that Ω is partitioned into $\Omega = Q^{-1}(0) \cup Q^{-1}(1)$. For each Q, **A** chooses some element $x \in X$, and answers Q with the value $Q(x) \in \{0, 1\}$. Thus, **S** knows that $X \cap Q^{-1}(Q(x)) \neq \emptyset$. Of course, **A** could always use the *same* element $x \in X$ to answer *every* question **S** asks, and so, **S** would never know anything about any of the *other* elements of X.

As shown in [1, 2], 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 of secrets (consistent with all the answers) forms an *intersecting* k-uniform hypergraph, i.e., a family **F** of k-sets of Ω so that any $F, F' \in$ **F** satisfy $F \cap F' \neq \emptyset$ (and this is the most that **S** can achieve). Any set of questions which always results in an intersecting hypergraph will be called a *separating strategy* for **S**.

In this paper, we will take Ω to be $\mathbb{F}_2^n$ for some integer n, where $\mathbf{F}_2$ denotes the field of two elements. Each "question" Q will just be some element of Ω, and an answer to Q will be the inner product $Q \cdot x$ (in $\mathbb{F}_2$) for some $x \in X$. (In fact, we will only need to use Q with rather small weight, where the weight $w(Q)$ is defined to be the number of coordinates of Q which are equal to 1). As we will see, with this restriction, S can no longer guarantee that any two surviving possible secret k-sets X and Y are intersecting. Rather, the best that S can hope for (and, in fact, which can always be achieved by what we call a *weak* separating strategy) is that two somewhat larger sets Odd(X) and Odd(Y) are always intersecting, where for $V = \{V_1, \ldots, V_k\} \subseteq \mathbb{F}_2^n$, we have

$$\mathrm{Odd}(V) := \{\sum_{i=1}^{k} \epsilon_i V_i \ : \ \epsilon_i = 0 \text{ or } 1 \text{ and } \sum_{i=1}^{k} \epsilon_i \text{ is odd}\}.$$

In the next section we show that the set of Q of weight at most $2k - 1$ forms a weak separating strategy for secret k-sets in $\mathbb{F}_2^n$.

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 when we consider sets of k secrets, say, $X = \{X_1, \ldots, X_k\}$. We remark that for *arbitrary* questions (i.e., functions from Ω to $\{0, 1\}$), we can guarantee that any two surviving k-sets are intersecting, i.e., have a nonempty intersection.

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

†University of California, San Diego, La Jolla, California

For any k-set $X = \{X_1, \ldots, X_k\} \subseteq \Omega$, define

$$\mathrm{Odd}(X) = \{\sum_{i=1}^{k} \epsilon_i X_i \ : \ \epsilon_i = 0 \text{ or } 1 \text{ and } \sum_{i=1}^{k} \epsilon_i \text{ is odd}\}$$

For any two k-sets X and Y in Ω, we say that X and Y are *strongly disjoint* if $\mathrm{Odd}(X) \cap \mathrm{Odd}(Y) = \emptyset$.

LEMMA 2.1. *For k-sets X and Y in Ω, the following conditions are equivalent:*

(i) *Odd*$(X) \cap$ *Odd*$(Y) = \emptyset$;

(ii) $X_1 - Y_1 \notin \langle X_1 - X_2, \ldots, X_{k-1} - X_k, Y_1 - Y_2, \ldots, Y_{k-1} - Y_k \rangle$;

(iii) *There exists $F \in \Omega$ such that*

$$F \cdot X_1 \equiv F \cdot X_2 \equiv \ldots \equiv F \cdot X_k \not\equiv$$

$$F \cdot Y_1 \equiv F \cdot Y_2 \equiv \ldots \equiv F \cdot Y_k (mod\ 2);$$

Let $K_{k,k}$ be the labeled complete bipartite graph with vertex set X and Y, and with each edge $e = X_i Y_j$ labeled with $f(e) = X_i - Y_j$.

(iv) *For some spanning tree T of $K_{k,k}$, $0 \notin$ Odd$(f(e) \ : \ e \in T)$;*

(v) *For every spanning tree T of $K_{k,k}$, $0 \notin$ Odd$(f(e) \ : \ e \in T)$.*

Proof: (ii) $\Rightarrow$ (iii).
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 $W = \langle \Delta_1, \ldots, \Delta_{2k-2} \rangle$, say, $W = \langle \Delta'_1, \ldots, \Delta'_r \rangle$ where each Δ'_i is some Δ_j. Extend this together with Δ_0 to a basis for $\mathbb{F}_2^n$, say,

$$\mathbb{F}_2^n = \langle \Delta_0, \Delta'_1, \ldots, \Delta'_r, \Gamma_1, \ldots, \Gamma_{n-r-1} \rangle.$$

The matrix

$$\Delta' = \begin{bmatrix} \Delta_0 \\ \Delta_1 \\ \cdot \\ \cdot \\ \cdot \\ \Gamma_{n-r-1} \end{bmatrix}$$

is invertible, say, with inverse D, so that $\Delta' D = I_n$, the $n \times n$ identity matrix. Define

$$D(1) = \begin{bmatrix} D(1,1) \\ D(2,1) \\ \cdot \\ \cdot \\ \cdot \\ D(n,1) \end{bmatrix}, \text{ the first column of } D.$$

Thus

$$\Delta' \cdot D(1) = \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

Since all the $\Delta_i, i > 0$, are linearly dependent on $\Delta'_1, \ldots, \Delta'_r$ then $\Delta_i \cdot D(1) = 0, 1 \leq i \leq 2k-2$, and $\Delta_0 \cdot D(i) = 1$, as required for (iii). □

The proof that (iii) $\Rightarrow$ (ii) is immediate since if $\Delta_0 = \sum_{i>0} \epsilon_i \Delta_i$ and $F \cdot \Delta_i = 0$ for $i > 0$ then $F \cdot \Delta_0 = \sum_{i>0} \epsilon_i F \cdot \Delta_i = 0$, contradicting (iii).

(ii) $\Leftrightarrow$ (i)

$$\begin{aligned} & \Delta_0 \in \langle \Delta_1, \ldots, \Delta_{2k-2} \rangle \\ \Leftrightarrow \quad & X_1 - Y_1 = \sum_{i=1}^{k-1} \delta_i (X_{i-1} - X_i) + \sum_{j=1}^{k-1} \epsilon_j (Y_{j-1} - Y_j) \\ \Leftrightarrow \quad & X_1 - \sum_{i=1}^{k-1} \delta_i (X_{i-1} - X_i) = Y_1 + \sum_{j=1}^{k-1} \epsilon_j (Y_{j-1} - Y_j) \\ \Leftrightarrow \quad & \mathrm{Odd}(X) \cap \mathrm{Odd}(Y) \neq \emptyset \end{aligned}$$

The other implications involving (iv) and (v) are easily established by similar arguments. □

Note that in the preceding proof, 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

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

Since $\Delta_1, \ldots, \Delta_{2k-2}$ are all linearly dependent on the

$\Delta'_i, 1 \le i \le r$, then in fact

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

where $\Delta(j)$ denotes the jth column of the matrix

$$\Delta = \begin{bmatrix} \Delta_0 \\ \Delta_1 \\ \cdot \\ \cdot \\ \cdot \\ \Delta_{2k-2} \end{bmatrix}$$

Hence, the vector $F = (F(1), \ldots, F(n))$ with

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

satisfies

$$\begin{aligned} F \cdot \Delta_0 &= F \cdot (X_1 - Y_1) = 1 \\ F \cdot \Delta_i &= 0, 1 \le i \le 2k-2. \end{aligned}$$

and has $w(F) \le r + 1 \le 2k - 1$.

In general, we will call a set $\mathcal{F}$ of inner product questions a *weak* separating strategy (for k-sets of secrets) if no matter how $\mathbf{A}$ chooses answers, any two surviving k-sets X and Y have $\mathrm{Odd}(X) \cap \mathrm{Odd}(Y) \ne \emptyset$ (i.e., the family of sets $\mathrm{Odd}(X)$ are intersecting). Thus,

$$\mathcal{F}(2k-1) = \{F \in \Omega \ : \ w(F) \le 2k-1\}$$

is a weak separating strategy (with $\sum_{i=1}^{2k-1} \binom{n}{i}$ questions).

This proves the following:

Lemma 2.2. *The set $\mathcal{F}(2k-1) = \{F \in \Omega \ : \ w(F) \le 2k-1\}$ is a weak separating strategy for k-sets of secrets.*

We point out that the bound of $2k-1$ above cannot be improved, as the following example shows.

Example. $n = 2k - 1$.

$$X = \left\{ \begin{array}{ccccccc|ccc} & & 1 & 2 & \ldots & k & & k+1 & \ldots & 2k-1 \\ \hline X_1 & = & 1 & 0 & \ldots & 0 & & 0 & \ldots & 0 \\ X_2 & = & 0 & 1 & \ldots & 0 & & 0 & \ldots & 0 \\ & & & & \ldots & & & & \ldots & \\ X_k & = & 0 & 0 & \ldots & 1 & & 0 & \ldots & 0 \end{array} \right.$$

$$Y = \left\{ \begin{array}{ccccc|cccc} & & 1 & \ldots & k-1 & k & k+1 & \ldots & 2k-1 \\ \hline Y_1 & = & 1 & \ldots & 1 & 0 & 1 & \ldots & 1 \\ Y_2 & = & 1 & \ldots & 1 & 1 & 0 & \ldots & 1 \\ & & & \ldots & & & & \ldots & \\ Y_k & = & 1 & \ldots & 1 & 1 & 1 & \ldots & 0 \end{array} \right.$$

Claim:

If $F \in \mathbb{F}_2^{2k-1}$ has $0 < w(F) < 2k-1$ then for some i and j, either $F \cdot X_i = 0, F \cdot X_j = 1$ or $F \cdot Y_i = 0, F \cdot Y_j = 1$. To see this, consider two cases:

(a) There exist $1 \le i, j \le k$ such that $F(i) \ne F(j)$. Then $F \cdot X_i \ne F \cdot X_j$.

(b) There exist $k \le i, j \le 2k-1$ such that $F(i) \ne F(j)$. Then $F \cdot Y_{i+1-k} \ne F \cdot Y_{j+1-k}$.

Since the hypothesis implies $F(i) \ne F(j)$ for *some* i and j, then either (a) or (b) must occur (or otherwise $F(i) = F(k) = F(j)$ for $i \le k \le j$). Of course, the all 1's question of weight $2k-1$ does separate $\mathrm{Odd}(X)$ and $\mathrm{Odd}(Y)$. □

Observe that $x \in \mathrm{Odd}(X) \Rightarrow w(x)$ is odd and $y \in \mathrm{Odd}(Y) \Rightarrow w(y)$ is even, and consequently $\mathrm{Odd}(X) \cap \mathrm{Odd}(Y) = \emptyset$. This shows that for general n, if even a single inner product question F of weight $2k-1$ is omitted from $\mathcal{F}(2k-1)$, then the resulting set of questions does not form a weak separating strategy.

3 Inverting the answers for $k = 2$

In this section we restrict our attention to the case $k = 2$ with $\mathcal{F}(3) = \{F \in \Omega \ : \ w(F) = 3\}$ as our weak separating strategy. Since $|X| = 2 \Rightarrow \mathrm{Odd}(X) = X$ then any two surviving pairs must have a common element. Thinking of pairs of elements of Ω as *edges* of a graph with vertex set Ω, then the surviving edges must either form a *star* S with some center X_0, or a *triangle* T on 3 vertices $\{X_1, X_2, X_3\}$. In the first case, it follows that X_0 must be one of $\mathbf{A}$'s secrets. In the second case, we can only say that $\mathbf{A}$'s secret pair (= edge) must be one of the three edges of T (so, in particular, we cannot conclude that any specific element of Ω is a secret of $\mathbf{A}$).

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) that ALG(Ω) gives the following information on the surviving intersecting set S of edges:

(i) S is a *star* with center X_0, or

(ii) S is a *triangle* on the set $\{X_1, X_2, X_3\}$, and in particular, any edge X_iX_j survives.

As a special case of (i), S could contain exactly one edge. For example, for $n = 5$, suppose the answers to the total of $\binom{5}{1} + \binom{5}{2} + \binom{5}{3} = 25$ questions is given in lexicographic order is as follows:

$$1110011110001100010111001.$$

The unique solution to the above set of answers is the pair 01110 and 10100.

In this case, ALG randomly chooses one vector as the center of "star". The other one is simply ignored. However, it is possible (and easy) to modify the ALG algorithm to distinguish this special case. So whenever all secrets can be determined, it will produce both. We will omit the details here.

Let us decompose $[n] := \{1, 2, 3, \ldots, n\} = J_1 \cup J_2 \cup J_3$ where $J_1 = \{1 \leq x < n/3\}, J - 2 = \{n/3 \leq x < 2n/3\}, J_3 = \{2n/3 \leq x \leq n\}$ and define $I_i = [n] \setminus J_i, 1 \leq i \leq 3$.

We begin ALG by executing ALG(I_1), ALG(I_2) and ALG(I_3) (note that the weight 3 questions on I_i are also weight 3 questions on $[n]$).

First we introduce some notation. If $A \in \Omega = \mathbb{F}_2^n$ then A/I_1 denotes the *restriction* of X to I_1, and indicate this by writing $A/I_1 = -A_2A_3$. Similarly, we write the other two restrictions of A as $A/I_2 = A_1 - A_3$ and $A/I_3 = A_1A_2 - .$

Observe that if UV is an edge in S, the interesting set of surviving edges in Ω, then

(i) If ALG(I_1) has the answer that X/I_1 is a star with center $A = -A_2A_3$ 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).

(ii) If ALG(I_i) gives a triangle ABC then both U/I_i and V/I_i are vertices of ABC.

We consider three cases:

Case (a): ALG(I_1) yields a star with center

$$A = -A_2A_3,$$

ALG(I_2) yields a star with center

$$B = B_1 - B_3,$$

ALG(I_3) yields a star with center

$$C = C_1C_2 - .$$

Let us say two restrictions U/I_i and $V/I_j, i \neq j$, are *compatible* if all common coordinate positions of the two sequences are equal, e.g., $U/I_1 = -U_2U_3$ and $V/I_2 = V_1 - V_3$ are compatible if and only if $U_3 = V_3$. Define the *merge* $[A, B]$ of two compatible restrictions to be the (unique) n-tuple W in Ω which generates the corresponding restrictions A and B. For example, $[-D_2D_3, D_1 - D_3] = D_1D_2D_3$. Let G denote the *compatibility graph* on the vertex set of centers A, B and C in I_1, I_2 and I_3, respectively, with edges between every pair of compatible vertices.

Observe that G must have at least one edge, since for each edge UV in S (the set of surviving edges), there are only two choices for each I_i of which endpoint of UV to use for the restriction (= star center). Thus, some endpoint U or V must occur twice, say, $A = U/I_1$ and $B = U/I_2$. In this case, A and B are compatible, and $[A, B] = U$.

Claim: ALG(Ω) must give a star as the answer if ALG(I_i), $i = 1, 2, 3$, yield stars.

Proof: It suffices to show that for any edge in G, the merge $[A, B]$ is a point which must be in *every* surviving edge in S. Suppose $A = -QR, B = P - R$ and the merge $[A, B] = PQR$ is not in a surviving edge UV. One of the points in UV must have a projection in I_1 which is equal to A. Without loss of generality, we assume that $U/I_1 = A$, and $U/I_2 \neq B$, which implies that $U = P'QR$ with $P' \neq P$. We can also assume $V = PQ'R$ and $Q' \neq Q$ since each surviving edge must contain one point with projection B, and the merge $[A, B]$ is not in UV.

Now we consider the surviving edge VW. We must have $W/I_1 = A$ since otherwise $V/I_1 = A$, a contradiction. Similarly, considering UW, we conclude $W/I_2 = B$. This implies that $W = PQR$.

Now we consider $C = C_1C_2-$, which is the star center given by ALG(I_3). We have $C_1C_2- \in \{U/I_1, V/I_1\} = \{P'Q-, PQ'-\}$. Without loss of generality, we may assume $C_1C_2- = P'Q-$. We now consider the edge VW. We see that C_1C_2- cannot be one of $\{V/I_3, W/I_3\}$, which is impossible. This completes the proof of the claim.

We now consider the following cases:

(1) Suppose G has *one* edge, say,

Since $U = [A, B]$ is determined then every $U'V' \in S$ must project the same majority point to ABC. Hence,

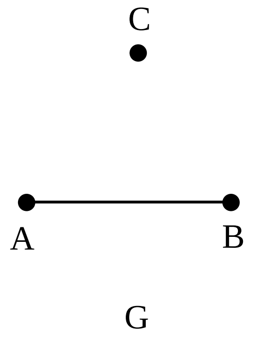

Figure 1: G has one edge.

$[A, B]$ is an endpoint of *every* edge in S, i.e., Ω has a star with center $[A, B]$.

(ii) Suppose G has *three* edges:

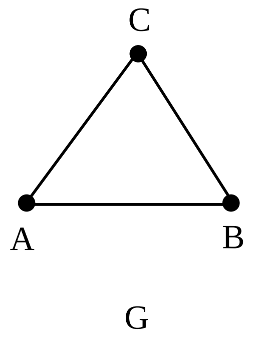

Figure 2: G has three edges.

Observe that in this case $[A, B] = [A, C] = [B, C]$. Thus, as in (i), we see that Ω must have a star with center $[A, B]$.

(iii) Suppose G has *two edges*, say,

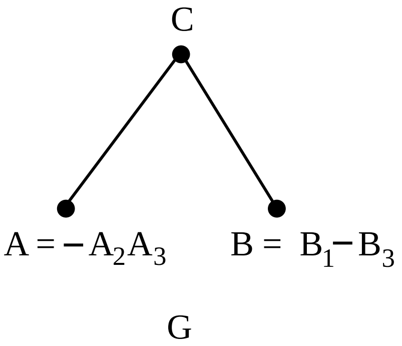

Figure 3: G has two edges.

Then for any $UV \in S$, we must have either $U/I_1 = A, V/I_2 = B$ or $V/I_1 = A, U/I_2 = B$ (since otherwise, A and B would be compatible.)

If the star in Ω (which ALG(Ω) gives) has center Z and at least two edges $ZW_1, ZW_2, \ldots$, then for each edge ZW_i, Z must project to the same point of A and B (for otherwise A and B could be compatible). Let us simplify our notation and write

$$\begin{aligned} A &= -QR \\ B &= P - R' \\ C &= PQ- \qquad \text{where } R \neq R' \end{aligned}$$

Since $R \neq R'$, there must be some coordinate $k_0 \in J_3$ such that $R(k_0) \neq R'(k_0)$. Now, consider the answers to all the questions F_{i,j,k_0} where $i \in J_1, j \in J_2$ (i.e., take the inner product with the vector having 1's in positions i, j and k_0). The two possibilities for the star in Ω are that either the center is $X_0 = PQR$ and the other endpoints are of the form $PQ'R'$ with Q' arbitrary, or the center is $X_0' = PQR'$ and the other endpoints are of the form $P'QR$ with P' arbitrary.

We next ask each of the $|I_1|\ |I_2|$ questions F_{i,j,k_0} , where $i \in I_1, j \in I_2$. Suppose $P(i) + Q(j) + R(k_0) = \alpha$ so that $P(i) + Q(j) + R'(k_0) = 1 - \alpha$. *If* F_{i,j,k_0} is answered α by $\mathbf{A}$ (so that all pairs with three coordinate sums $1 - \alpha$ are eliminated, then exactly all the points $P'QR$ with $P'(i) \neq P(i)$ are ruled out as possible mates for X_0'. On the other hand, if F_{i,j,k_0} is answered $1 - \alpha$, then by the same argument any point $PQ'R'$ with $Q'(j) \neq Q(j)$ is ruled out as a possible mate for X_0. Note that X_0 is the trivial mate for X_0', and vice versa. At the end of the process, one of the two points X_0 and X_0' will have *all* its possible mates (except for its trivial mate) eliminated and the other one must be the surviving star center (and in addition, we know a lot about the endpoints of the star). If we make comparisons with the existing answers to all the questions, we can rule out the bogus star center in linear time (since each F_{i,j,k_0} rules out a value of $P(i)$ or $Q(j)$).

It is quite possible that both points have only their trivial mates surviving. In this case, the surviving set contains the only edge X_0X_0'. ALG will randomly choose one point as its center for later use (in the induction step).

Case(b) Two of ALG(I_i) give stars and the third gives a triangle, say, I_1 has a star with center $A = -A_2A_3$, I_2 has a star with center $B = B_1 - B_3$, and I_3 has a triangle in $P = P_1P_2-, Q = Q_1Q_2-, R = R_1R_2-$.

Suppose $UV \in S$.

(1) If A and B are *not* compatible then A and B are projections of both endpoints U and V. Since I_3 has *different* projections of both U and V then U and V can be recovered (and tested) from the $[\{A,B\},\{P,Q,R\}]$ merges.

(2) Suppose A and B are compatible. If $A = U/I_1$ and $B = V/I_2$, then again U and V can be recovered from the $[\{A,B\},\{P,Q,R\}]$ merges. On the other hand, all other edges, which project the same endpoint to A and B, must share the common vertex $[A,B]$.

Hence in any case, we have all the information to determine the center (if ALG(Ω) gives a star) or the three vertices of a triangle (if ALG(Ω) gives a triangle).

Case(c) At least two of ALG(I_2) give triangles, say, ALG(I_1) gives a triangle on $\{A,B,C\}$ and ALG(I_2) gives a triangle on $\{P,Q,R\}$. Then all possible endpoints of edges in S are contained in the set $[\{A,B,C\},\{P,Q,R\}]$, and each of these can be individually tested.

Putting all the preceding cases together we have the following recursive bound on $c(n)$, the number of comparisons needed on the $\binom{n}{3}+\binom{n}{2}+\binom{n}{1}$ answers to determine the required output for ALG(Ω).

$$(3.1) \quad c(n) \le 3\cdot c(\frac{2n}{3}) + \binom{9}{2}(\frac{n}{3})^3 + o(n^3)$$

The term $3\cdot c(\frac{2n}{3})$ comes from the execution of the three ALG(I_i). The term $\binom{9}{2}(\frac{n}{3})^3$ is an upper bound of the number of comparisons needed to make the final resolution, the most costly being the individual testing of each of the possible $\binom{9}{2}$ pairs of points arising from $[\{A,B,C\},\{P,Q,R\}]$ in Case (c). This implies the bound

$$c(n) \le 12n^3 + o(n^3).$$

We have implemented this algorithm. It takes only a few seconds to process $n = 256$-bit binary strings on a Pentimum II PC (under Linux). It seems to demand a lot of memory when n is large. On our Department Unix machine, it can handle $n = 400$ with ease.

4 NP-hardness for $k = 3$

In this section we will show that no weak separating strategy $\mathcal{F}$ can have a polynomial-time algorithm for inverting $\mathbf{A}$'s answers when $k = 3$, unless $P = NP$.

Suppose $\mathcal{F}$ is some weak separating strategy. We can assume without loss of generality that $\mathcal{F}$ also includes all the weight 2 questions $F_{i,j}$, $i,j \in [n]$. Let G be a fixed graph with vertex set $[n]$ and edge set E. We will suppose that $\mathbf{A}$ answers the questions $F \in \mathcal{F}$ as follows:

$$\mathbf{A}(F) = \begin{cases} 1 & \text{if } F = F_{i,j} \text{ and } ij \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Claim: G is 4-chromatic if and only if there is a valid secret triple of the form $\{0,u_1,u_2\}$, where 0 denotes the vector $(0,0,\ldots,0) \in \mathbb{F}_2^n$.

Proof of Claim:

Assume G is 4-chromatic. Then E can be decomposed into two bipartite graphs G_1 and G_2 on $[n]$ with edge sets E_1 and E_2, respectively (see Fig. 4).

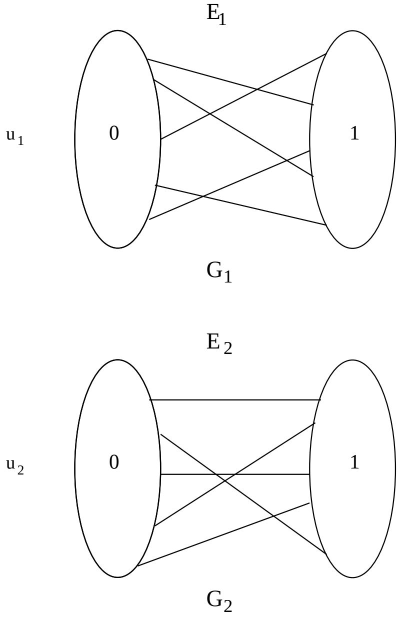

Figure 4: Partition into two bipartite graphs.

Define the vectors $u_i \in \mathbb{F}_2^n$, $i = 1,2$, by

$$u_i(k) = \begin{cases} 0 & \text{if } k \in A_i \\ 1 & \text{if } k \in B_i. \end{cases}$$

It is clear that with this choice that $\{0,u_1,u_2\}$ is a valid secret triple for $\mathbf{A}$, i.e., all the answers $\mathbf{A}(\mathcal{F})$ given are consistent with $\mathbf{A}$ having the secret triple $\{0,u_1,u_2\}$.

For the other direction, assume $\{0,v_1,v_2\}$ is a valid secret triple for $\mathbf{A}(\mathcal{F})$. Reverse the preceding construction and construct two bipartite graphs G_i as follows:

Define $A_i := \{k : v_i(k) = 0\}, B_i := \{k : v_k(k) = 1\}$ and $E_i = \{\{a,b\} : a \in A_i, b \in B_i\}$. Thus,

$\mathbf{A}(F_{i,j}) = 1$ if and only if $ij \in E_1$ or E_2, and the proof of the Claim is complete. $\square$

Observe that if $\{x, y, z\}$ is a valid secret triple for some set of answers $A(F), F \in \mathcal{F}$, then any 3-element subset of $\mathrm{Odd}(x, y, z) = \{x, y, z, x + y + z\}$ also is.

Now suppose we have a polynomial-time algorithm ALG which can invert the answers $\mathbf{A}(F)$, defined as before according to edge set of G, producing a solution $\{x, y, z, x + y + z\}$ as an Odd set satisfying the answers $\mathbf{A}(F)$. Thus $\mathcal{F}$ must have polynomial size. Now, if $0 \in \{x, y, z, x + y + z\}$, then by the preceding remarks G must be 4-chromatic. On the other hand, since $\mathcal{F}$ is a weak separating strategy, then any two satisfying Odd quadruples must intersect. So we can assume that $0 \notin \{x, y, z, x + y + z\}$.

Now, if G is 4-chromatic then there must be a satisfying Odd set of the form $\{0, u, v, u + v\}$, which intersects $\{x, y, z, x + y + z\}$. Hence, u, v or $u + v$ must be equal to one of x, y, z or $x + y + z$. However, we can efficiently test whether there is a satisfying triple of the form $\{0, x, x'\}$ as follows. Namely, sequentially check each $F \in \mathcal{F}$ with $\mathbf{A}(F) = 1$ to see whether $F \cdot x \equiv 1$. If not, then add the (algebraic) constraint $F \cdot x' \equiv 1$ to a matrix M of such constraints on x'. At the end, we can use Gaussian elimination on M to decide whether or not an appropriate x' exists (this can be done in polynomial time in n).

G is 4-chromatic if and only if this process succeeds for *some* element of $\{x, y, z, x + y + z\}$ paired with 0. However, it is known [3] that determining if a graph is 4-chromatic is NP-hard. Hence, the existence of our hypothetical polynomial time algorithm ALG would imply that $P = NP$, an assertion not widely believed.

Of course, similar conclusions apply for $k > 3$.

5 Concluding remarks

There are still quite a few aspects of this problem which need greater understanding. For example, probabilistic arguments show (see [1]) that (in the case of $k = 2$) there are weak separating strategies for $\mathbb{F}_2^n$ with only $O(n)$ questions. (We achieve $O(n^3)$ in this paper using $\mathcal{F}(3)$.) We do not currently know of any *constructive* way of producing such strategies.

In the more general case [1] where $\Omega = \{1, 2, \ldots, N\}$ and questions are just functions mapping Ω to $\{0, 1\}$, it is possible to construct separating strategies (using properties of quadratic residues modulo primes) of size $O(\log^2 N)$ but we have no idea how to *invert* these answers. Finally, in this setting we have no non-trivial estimate for how much better an *adaptive* separating strategy is then our usual non-adaptive (i.e., oblivious) algorithms. As proved in [1], for $k = 2$, any adaptive separating strategy for Ω must have at least $3 \log_2 N - 5$ questions, and need not have more than $4 \log_2 N + 3$ questions (for $N > 2$). Surely no oblivious algorithm can be this good! Of course, for $k > 2$, we know even less.

References

[1] F. Chung, R. Graham and F. T. Leighton, Guessing Secrets (Extended Abstract), *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms* (2001), 723-726.

[2] F. Chung, R. Graham and F. T. Leighton, Guessing secrets, *Electronic Journal of Combinatorics* **8** (2001), #R13.

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