Even Data Bases That Lie
Can Be Compromised

Richard A. DeMillo, David Dobkin,
Richard J. Lipton

Research Report #67

May 1976

DeMillo is at the Department of Electrical Engineering, University of Wisconsin-Milwaukee, Milwaukee, Wisconsin 53201.

This work was supported in part by National Science Foundation Grant DCR-74-24193 and by US Army Research Office Grant DAHCO-74-G-01-79.

Protecting information in data bases from access by naive or malicious users is a fundamental consideration in system design. A data base whose information can be deduced by a finite mechanism is said to be *compromised*. Dobkin, Jones, and Lipton [2] show that users can compromise data bases simply by asking a series of statistical queries.

Consider, for example, a data base of employee salaries. Normal system protocol may restrict users to querying the system in the following way:

$$\text{"What is the median salary of } \{\zeta_k\}\text{?"} \qquad (1)$$

where $\{\zeta_k\}$ is a set of k employees. The system answers such a query with the value of the median salary but not who earns it. This data base is compromised if a user can determine some employee's salary.

Even if a user may ask only questions of form (1), he can always compromise the data base in

$$\frac{3}{2}\,(k+1) + 1$$

queries [2]. Notice that k, the size of the median sample is fixed; without this restriction the problem is trivial since

"What is the median salary of Jones?"

would then be a legal query.

It is therefore dramatically clear that protecting a data base against compromise is very difficult. A natural scheme for avoiding this difficulty is to perturb slightly the responses to queries. This leads us to ask what happens if a data base can "lie" [1]. A data base that lies is not bound to give truthful answers to queries. For example, asked the median salary of Brown, Jones, Smith, Green, and Black, the system may decide to return Black's salary whether or not it is actually the median

This approach seems entirely reasonable; indeed, we can imagine a data base that tells much more bizarre lies to confuse the malicious user. Our main result is therefore surprising:

> Even a data base that lies by answering a question about medians with any value stored in the data base can always be compromised -- and in relatively few steps.

We now give a more precise statement of this result. Let $k > 0$ be the fixed sample size permitted by a query. A query is then:

"What is the median salary of the following list of k employees?" (2)

The data base can return *any* of the values of these k employees. Notice that the rule used to select which value to return need not be regular in any way; it can be random, nondeterministic, or even time-varying. Define the function m(k) as follows:

$$m(k) = \begin{cases} k^2 + 1 & \text{if } k \text{ is a prime power} \\ 4(k^2+1) & \text{otherwise.} \end{cases}$$

Theorem: In no more than m(k) queries of the form (2) it is always possible to compromise a data base.

For this short note, we will not be more precise about the concept of "compromise," and the reader will not be led astray by the intuitive notion. The interested reader will find a precise definition in Dobkin, Jones, and Lipton [2].

Before presenting the proof of this theorem, we strengthen it slightly by further restricting the user. Restricting the overlap of queries is an intuitively reasonable way to protect a data base against compromise. By an overlap restriction we mean that no pair of queries can contain lists of employees with more than one employee in common. Thus the two queries

"What is the median salary of Jones, Brown, Smith?"

"What is the median salary of Jones, Brown, Green?"

are not allowed since Jones and Brown are common to both lists. If queries can have no overlap, compromise is impossible. So our restriction to an overlap of one is a severe restriction on the user, although we will prove that compromise is still possible in m(k) queries.

Proof of theorem: For convenience, let $s_1,\ldots,s_n$ be the salaries of the n employees of the data base, and assume that all salaries are distinct.† If k is the query size, then a query is completely described by a set Q of integers between 1 and n. A response to a query Q is any answer s_i, provided only that i is in Q.

† It can be shown that this can be assumed without loss of generality in data bases that are "statistically reasonable" in a precise sense, which we do not define here.

The key to our argument lies in finding sets of queries $Q_1,\dots,Q_M$, where $M = m(k)$, that satisfy the following properties:

a) For $i \neq j$, Q_i and Q_j satisfy the overlap restriction ($Q_i \cap Q_j$ contains at most one element).

b) Each set Q_i contains only queries about the first M-1 employees ($Q_i \subseteq \{1,\dots,M-1\}$).

A proof of the existence of such sets is sketched in the appendix. Once the magic step of finding $Q_1,\dots,Q_M$ has been taken, the rest of the proof is easy. The user simply generates queries that correspond to $Q_1,\dots,Q_M$; these satisfy the overlap restriction by property (a). Suppose that the data base returns s_{a_i} as the answer to the *ith* query, so that the only information the user gets is that a_i is in Q_i. We claim that some answer is given twice, i.e. for some i,j ($i \neq j$)

$$s_{a_i} = s_{a_j}.$$

By property (b), $1 \leq a_i \leq M-1$, but since there are all together M responses $a_i = a_j$ for some i,j ($i \neq j$) by the pigeon-hole principle.[†] Suppose that s_{a_i} is the answer to both queries Q_i and Q_j, so that uniqueness requires that a_i must be in Q_i and in Q_j. Finally property (a) allows us to determine a_i. Therefore, the value of the salary of employee a_i must be s_{a_i}, and the user has compromised the data base. □

[†] If $p+1$ objects are placed in p containers, then some container must have received at least two objects.

This theorem should be contrasted with the results of Kam and Ullman [3], who require a great deal of information about the values stored in the data base and rather unnatural queries to achieve a base that cannot be compromised -- and even then, only statistically. Our theorem has much in common with other seemingly anomalous facts about what can be inferred from rules that are apparently irregular or difficult to understand. Consider, for example, that a notationally complex generating function is not enough to guarantee randomness in the sequence it generates [5]; much more careful analysis is needed to aid one's intuitions. Similarly naive strategies for delivering responses to queries, even when the strategies are bizarre, do not guarantee secure data bases; a determined user does not need any understanding of the way in which responses to his queries are selected to compromise a data base. This is further evidence that protecting data bases is a subtle problem, whose solution will require careful analysis of the underlying principles and issues of data base security.

Acknowledgement: The authors would like to thank Professors D. Kleitman and R. Rivest for helpful comments.

Appendix

We prove here the existence of the required sets $Q_1,\ldots,Q_M$. We first consider the simpler case in which k is a prime or a power of a prime. By the existence of projective planes of order k [3], there are k^2+k+1 sets such that

1) each set is contained in $\{1,\ldots,(k^2+k+1)\}$,

2) each pair of sets intersects at exactly one point,

3) each set has $k+1$ elements.

Now, remove one set from this system of k^2+k+1 sets and delete its elements from the system. Since $k+1$ points are deleted in this fashion, we can remember, if necessary, the remaining points to obtain a system of k^2+k sets of points from $\{1,\ldots,k^2\}$. Since $k^2+k \geq k^2+1 \geq m(k)$, from the remaining system of sets, $m(k)$ query sets can be constructed that satisfy our requirements.

For arbitrary k we proceed by embedding in the next largest prime power $k' \geq k$. Since k' is $\leq 2k$ our claim follows.

References

1] Conway.
Private communication.

2] D. Dobkin, A. Jones, and R. Lipton.
Secure data bases: Protection against user inference.
Yale Computer Science Research Report.

3] M. Hall.
Combinatorial Theory.
Ginn and Blaisdell, 1967.

4] J. Kam and J. Ullman.
Security in statistical data bases.
Princeton University, Department of Electrical Engineering, Technical Report 207, 1975.

5] D. Knuth.
The Art of Computer Programming. Volume 2: Semi-numerical Algorithms.
Addison-Wesley, 1969.