The Devil's Chessboard

Kağan Aşık

April 2026

The Problem

Let's say you and a friend will be subjected to a test. A chessboard sits in a room, each square holding a coin in either heads or tails position. A voice announces a number from 1 to 64, and you flip one coin. Your friend, waiting outside, sees none of this: not the board's initial state, not the number, not which coin you flipped.

The question is: Can you develop a strategy that allows your friend to identify the number by looking only at the final state of the board, in every possible case? Both of you may agree on a strategy beforehand.

This problem is often dramatized so that if the second person guesses wrong, both are killed, hence the name *The Devil's Chessboard*. I prefer to skip that framing and let the mathematics speak for itself.

To be clear: everything except the initial board state and the announced number is known to us in advance, and we have all the time we need. Our goal is a strategy that works regardless of what the board looks like or what number the voice says.

First Analysis

Since there are 64 squares on a chessboard and each square has two possible states, the total number of configurations is an extremely large number, namely 2^{64}. For example, there is a configuration in which all squares are heads, one in which all are tails, or one in which the white squares are heads and the black squares are tails.

We want to encode 64 different numbers using these configurations. With so many configurations, this may seem quite easy at first glance. For instance, after listing the configurations, we could match the first 63 numbers with the first 63 configurations and assign all remaining configurations to correspond to the number 64. However, such an encoding does not satisfy the property we desire. Our goal is to be able to move, from any given configuration, to a configuration encoding any desired number in a single move. It is quite clear that this is not possible with the encoding described above.

Let us denote the set of all configurations by X. Define the *distance* between two elements of this set as the number of coins that must be flipped to obtain one from the other. This definition indeed induces a metric on X; in coding theory, this is called the **Hamming metric**.

Using this new terminology, we can express our goal as follows: we want to partition the set X into 64 disjoint classes such that the distance between the classes is 1. That is,

$$X = X_1 \cup \cdots \cup X_{64}$$

where the sets X_i are pairwise disjoint. Moreover, for every $u \in X$ and every $j \in \{1, \ldots, 64\}$, there exists an element $v \in X_j$ whose distance to u is 1.

Thus, if such a partition exists, you and your friend will agree on it in advance. When the first person enters the room, they will observe an element of X, namely the configuration denoted above by u. Then a voice will announce a number between 1 and 64; this corresponds to the value denoted above by j.

After that, following the predetermined method, the first person will flip a single coin to obtain another element of X, namely v. When the second person enters the room, the only thing they need to do is determine which set X_j contains v, and state the corresponding number j.

A Smaller Case: The 2×2 Board

A method frequently used by mathematicians is to consider simpler versions of a problem. In this case, the simplest version would be a 1×1 chessboard, but that is too trivial. As a next step, let us consider a 2×2 chessboard.

There is, in fact, a small issue here: formulating the analogue of the original problem for this smaller board. I propose the following formulation: on this board, there are again four coins placed randomly on the squares, and a voice announces a number between 1 and 4. Thus, this time our set of configurations has $2^4 = 16$ elements; let us again denote this set by X.

For now, instead of viewing the board as a square, let us think of it as a sequence of adjacent boxes containing the letters Y (heads) and T (tails). We define the following partition:

$$\begin{aligned} X_1 &= \{\texttt{YYTT}, \texttt{YTYY}, \texttt{TYTT}, \texttt{TTYY}\} \\ X_2 &= \{\texttt{YYTY}, \texttt{YTYT}, \texttt{TYTY}, \texttt{TTYT}\} \\ X_3 &= \{\texttt{YTTY}, \texttt{YYYT}, \texttt{TTTY}, \texttt{TYYT}\} \\ X_4 &= \{\texttt{YYYY}, \texttt{YTTT}, \texttt{TYYY}, \texttt{TTTT}\} \end{aligned}$$

It is easy to check that this partition works. For example, suppose the first person enters the room and sees the configuration TYTT, and then the voice says 2. In that case, they will flip the last coin to obtain TYTY. Since this element belongs to X_2, the second person will correctly conclude that the announced number was 2.

The main question is the following: how did I find these sets? Even in this relatively simple case, it is not very easy to arrive at the solution by trial and error. Once we understand the solution to the original problem, this construction will also become clear.

Solution I: Subsets of the Board

As you will see shortly, this is actually not a bad solution, but since it cleverly bypasses the mathematical core of the problem, I initially thought it was.

There are many ways to identify the squares on a chessboard. Normally, each square is determined by two coordinates: horizontal and vertical. Conventionally, the rows are denoted by the letters A, B, C, D, E, F, G, H, and the columns by the numbers 1, 2, 3, 4, 5, 6, 7, 8; a square is then expressed by a letter and a number, such as B5.

Alternatively, we can number the squares from 1 to 64, starting from the top-left corner and proceeding in the usual reading order. For the solution, we will introduce a new way of representation. To this end, let us assign names to the following six subsets of the board:

1. the upper half of the board
2. the left half of the board
3. rows A, B, E, F
4. rows A, C, E, G
5. columns 1, 2, 5, 6
6. columns 1, 3, 5, 7

The most important property of these subsets is that the collection of subsets to which a square belongs uniquely determines that square. For example, the only square that belongs to all subsets is the first square; the only square that belongs to none of them is the last square; and the only square that belongs to exactly the first and fifth subsets is square number 22. In short, we choose to represent each square by 6 coordinates taking values 0 or 1. The reason for this choice is that $2^6 = 64$.

Using these subsets, we will assign a number between 1 and 64 to each configuration. For a given configuration, let us count the number of tails in each subset. If the number of tails in a subset is even, we assign the value 0 to that subset; if it is odd, we assign the value 1. The number corresponding to this configuration is then the number of the square that lies in exactly those subsets assigned the value 1 and not in those assigned 0.

Our new goal is now the following: given a configuration and a number, we want to flip a single coin to obtain a configuration corresponding to the given number.

In fact, this amounts to finding a square that belongs to certain subsets and not to others. Since our partition is constructed precisely in this way, we can always do this.

It is best to examine this through an example. Consider the configuration in which all white squares except the sixteenth square are heads, while the sixteenth square and all black squares are tails. In other words, there is an odd number of T's in the first and third subsets, and an even number in the others. Therefore, this configuration corresponds to the number 16, which lies exactly in the first and third subsets.

Now suppose the number announced by the voice is 15. This number belongs to the first, third, and sixth subsets. Therefore, we need to change the parity of the number of T's in the sixth subset while keeping the others unchanged. To achieve this, we should flip the coin on square number 63, which belongs only to the sixth subset.

As mentioned at the beginning of this section, this is a clever solution, but somewhat combinatorial. In particular, it requires both participants to memorize quite a lot of information. The most elegant solution, which we have saved for last, will require nothing beyond basic addition and multiplication, in fact even less: it suffices to know addition and multiplication modulo 2.

Solution II: Binary Encoding

We mention this solution because it contains the core idea of the most elegant solution. Here again, we will encode 64 numbers using configurations, but it will be more convenient to take these numbers to be between 0 and 63; of course, doing so does not change anything about the problem.

Let us place subsets of the set $[6] := \{0, 1, \ldots, 5\}$ into the squares of the board. Since there are exactly 64 subsets, each square will correspond to exactly one subset.

Now, we define the number corresponding to a given configuration as follows: for $i \in [6]$, if the number of tails in the squares whose assigned subsets contain i is even, let $s(i) = 0$; otherwise, let $s(i) = 1$. Then the number corresponding to this configuration is defined as

$$s(0) \cdot 2^0 + s(1) \cdot 2^1 + s(2) \cdot 2^2 + s(3) \cdot 2^3 + s(4) \cdot 2^4 + s(5) \cdot 2^5.$$

Note that this is precisely the binary expansion of a number between 0 and 63.

Now suppose we are given a number between 0 and 63, and consider its binary expansion:

$$a_0 \cdot 2^0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + a_3 \cdot 2^3 + a_4 \cdot 2^4 + a_5 \cdot 2^5 \qquad (a_i \in \{0, 1\}).$$

Let us determine the indices where the a_i's differ from the $s(i)$'s. This gives us a subset of $[6]$; denote this subset by A. If we flip the coin on the square corresponding to the subset A, then in the resulting configuration only the values $s(i)$ for $i \in A$ will change (from 0 to 1 or from 1 to 0).

Consequently, the number corresponding to the new configuration will be exactly the desired one.

Solution III: Linear Algebra over $\mathbb{F}_2$

We denoted the set of all possible configurations by X, and noted that this set has 2^{64} elements. Let us identify this set X with the vector space $V = \mathbb{F}_2^{64}$, a 64-dimensional vector space over the field $\mathbb{F}_2$. In doing so, we represent a square being heads by 0 and tails by 1. For example, the configuration that starts with a head in the top-left corner, then a tail next to it, and continues alternating as head, tail, ..., will be represented by

$$(0, 1, 0, 1, \ldots, 1, 0, 1).$$

Let us also identify the set of the 64 possible numbers that the voice may announce with a vector space $W = \mathbb{F}_2^6$.

We can summarize our goal as finding a function

$$f \colon V \to W$$

satisfying certain properties. The desired property is that, for any given $v \in V$ and $w \in W$, there exists a vector v' obtained by changing exactly one coordinate of v such that $f(v') = w$.

Let us list the elements of W as $(w_1, w_2, \ldots, w_{64})$, and define

$$f(a_1, a_2, \ldots, a_{64}) = a_1 w_1 + a_2 w_2 + \cdots + a_{64} w_{64}.$$

This is a surjective linear function. Hence, there exists a configuration corresponding to every number. But does it satisfy the desired property?

To see this, let $v = (a_1, a_2, \ldots, a_{64}) \in V$ and let $w_k \in W$ be given. Suppose $f(v) = w_l$. Then which coordinate of v should we change? Assume that

$$w_k + w_l = w_m.$$

Then we should change the m-th coordinate. Indeed, by linearity,

$$f\left(v + (0, \ldots, \underbrace{1}_{m}, \ldots, 0)\right) = f(v) + f(0, \ldots, 1, \ldots, 0) = w_l + w_m = w_l + w_l + w_k = w_k.$$

Thus, the desired result is obtained. □

Conclusion

I hope you enjoyed walking through this mathematical analysis with me. We started by stripping away the life-or-death story to expose the raw mathematical core of the chessboard. From there, we mapped out the sets of configurations to build a concrete strategy. By dropping down to a simpler 2×2 problem, we were able to see the hidden mechanics much more clearly, before zooming back out to explore three distinct solutions, each offering a completely different perspective on how to outsmart the Devil.