to know what it is. We next discuss a concept called "oblivious transfer," with which one can construct noninteractive zero-knowledge proofs. Finally, we use oblivious transfer to give a zero-knowledge proof of factorization.

Map coloring. Our first example is the following. It is now known that any planar map can be colored with 4 colors. Some maps can be colored with 3 colors and others cannot. Suppose Pícara is given a complicated map, which after much effort she is able to find a way of coloring with only 3 colors (red, blue, green). How can she convince Vivales that she has done this, without giving him a clue that would help him color the map?

We first translate this problem into the language of graphs.

Definition. A *graph* is a set V, whose elements are called "vertices," and a subset E of the set of all (unordered) pairs of elements of V. The elements of E are called "edges." An "edge" $e = \{u, v\}$, where $u, v \in V$, should be visualized as a line joining the vertices u and v.

Definition. We say that a graph is *colorable* by the colors r, b, g, if there exists a function $f : V \rightarrow \{r, b, g\}$ such that no vertices joined by an edge have the same color, i.e., $\{u, v\} \in E \Longrightarrow f(u) \neq f(v)$.

The 3-colorability problem consists in determining, given a graph, whether or not it is colorable by r, b, g.

To translate the map-coloring problem to a graph-coloring problem, simply take V to be the set of countries (visualized now as points), and "connect" two countries with an edge if and only if they have a common boundary.

The 3-colorability problem has two nice properties which make it a convenient choice for discussions of many questions: (1) it is easy to visualize; and (2) it is NP-complete (see the discussion of the knapsack in §4). The NP-completeness property implies that, if you have a zero-knowledge verification of 3-colorability, then you can get a zero-knowledge verification for any NP-problem by "reducing" it to 3-colorability.

However, this does not mean that, once a zero-knowledge verification has been constructed for a certain NP-complete problem P_1 (say, 3-colorability), it is then superfluous to construct a zero-knowledge proof for another NP-problem P_2. On the contrary, in the process of reducing P_2 to P_1, one generally increases the size of the input data substantially. Thus, a much more efficient zero-knowledge verification is likely to result by working directly with P_2 rather than by reducing P_2 to P_1 and then using the earlier verification of P_1. For example, we shall later give a direct zero-knowledge proof of possession of a discrete logarithm. It would be inefficient in the extreme to construct such a zero-knowledge proof by first reducing possession of a discrete log to 3-colorability of some graph.

Zero-knowledge proof of 3-colorability. Suppose that Pícara is given a graph. We shall visualize the vertices as small balls containing little colored lights and joined by bars wherever there is an edge. The light in each vertex can flash either red, blue or green. Pícara has (1) a device A which sets each vertex to flash whichever of the three colors she chooses, and (2) a device B