Combinatorially Based Cryptography for Children (and Adults)

Michael R. Fellows
Department of Computer Science
University of Victoria
Victoria, B.C. V8W 3P6, Canada

Neal Koblitz
Department of Mathematics GN–50
University of Washington
Seattle, Washington 98195, U.S.A.

April 21, 1997

Abstract

In this paper we show how certain notions of modern cryptography can be presented to youngsters using combinatorial constructions. Among the topics discussed are the use of Boolean circuits for bit commitment protocols and hash functions, and the construction of a public key message transmission system using perfect codes in a graph. We also discuss how efforts such as this in popularizing mathematics for children are related to mathematics education reform.

1 Introduction

In this paper we show how ideas of cryptography can be presented to schoolchildren using combinatorial constructions. These topics can motivate the students by providing a stimulating context for logical and mathematical modes of thinking. Our hope is that this discussion will be useful in the enrichment and improvement of the school mathematics curriculum, and also contribute to the popularization of mathematics among children and the general public.

In §2 we describe some problems in discrete mathematics that have been successfully presented to young children. We characterize these topics as *pre-cryptography*, i.e., they involve certain elements of cryptography but do not yet constitute a cryptographic protocol.

§3 is a digression on the subject of good and bad pedagogy. Some of this section — particularly our critique of the computer craze — is intended to be controversial, perhaps even heretical.

§3 also contains an analysis of why topics in discrete math and cryptography can be especially useful in teaching mathematics to female and minority children in the U.S. and to children in the Third World.

In §4 we give examples of combinatorially based cryptography with props. However, our main interest in this paper is in propless cryptosystems, i.e., those which depend upon purely mental constructions (e.g., choosing a random number) rather than physical constructions (e.g., shuffling cards).

In §5 we present a cryptographic bit commitment protocol ("flipping a coin") based on Boolean circuits. Boolean circuits also give us hash functions for use, for example, in public key signature schemes. However, we have not yet been able to find a usable signature scheme based on combinatorial constructions.

In §6 we discuss a public key cryptosystem for message transmission based on the notion of a perfect code in a graph. We have several versions of this system, ranging from one that is simple enough for children in the primary grades to one that conceivably could be used in professional cryptography (at least, we have thus far been unable to break it).

In §7 we conclude with a brief discussion of the history of combinatorially based cryptosystems and directions for further work.

By its very essence, cryptography is a most excellent vehicle[1] for presenting fundamental mathematical concepts to children. Cryptography can be broadly defined as *mathematics/computer science in the presence of an adversary.* Implicit in any discussion of cryptography are elements of drama, of theater, of suspense. Few things motivate children as much as wanting to defeat the "bad guys" (or play the role of bad guys themselves).

Cryptography's ability to excite children has long been understood by advertisers of products like Rice Krispies and Crackerjacks. Many of us grew up quarreling with our siblings over who was going to get the decoder ring in the Crackerjacks box. Currently, some boxes of Rice Krispies have on the back a "secret algorithm" age guessing game based on binary representation of integers.[2] It is our hope that the charm and excitement of cryptography can provide a means to increase children's enjoyment and appreciation of mathematics. We use the term "Kid Krypto" to refer to this project.

[1] Akin to the telephone booth in Bill and Ted's Excellent Adventures.

[2] Just think how much better off the American educational system would be if the creative energy and ingenuity that goes into designing advertisements for TV and for cereal boxes could be harnessed and applied instead to pedagogical innovation!

2 Pre-Crypto

In order to present cryptography to children, there are certain "building block" ideas which are useful to develop first, and that are engaging in their own right. For example, there are many entertaining ways to introduce the notion of an algorithm, and of computational complexity. The idea of a one-way function, which plays a central role in modern cryptography, and the concept of an information hiding protocol can also be made accessible even to primary school students.

Among the ways to present the fundamental ideas of algorithmic procedure and computational complexity, we shall illustrate just a few of our favorites. The examples below have been tried out with children sometimes as young as 5 or 6.

2.1 Map Coloring

When giving this example to a class of young children, it is best to start with a story. You might tell of the poor Map-Colorer, trying to eke out a living with few crayons, and then pass out a map that needs to be colored. The definition of a proper coloring is visual, and can be illustrated with the maps at hand in the classroom. It is only a few minutes until most of the children understand the problem you have posed (finding out the minimum number of colors for the map you have passed out) and are puzzling away at it. It is a good idea to come to the classroom with plenty of copies of 3 or 4 different maps.

It is easy to generate a map that is two-colorable by overlaying closed curves. (Generating such a map is another topic the children may have fun thinking about). See Figure 1. In a typical classroom, children will figure out the algorithm for 2-coloring on their own, and they will see that it goes very quickly. It is easy enough to explain why it works: it has been called the "Have-to Algorithm" (if a country is red, then its neighbors have to be blue, and their neighbors have to be red, ...). Afterwards, you might distribute a map that requires 3 colors so that they can concretely contrast the 2-coloring experience with the apparent difficulty of finding a 3-coloring of a 3-colorable map.

Remark 1. Classroom experiences often lead to intriguing research questions that turn up in a playful vein. For example, the following question arose when the first author presented Map Coloring on one occasion. What is the minimum number of colors with which one can always color a planar map in a situation where one takes turns with an "incompetent helper" who is only assumed to color legally, but not necessarily judiciously? A bound of 33 colors was recently proved [11] for this problem.

Remark 2. As a variant, one can do graph-coloring (i.e., coloring of vertices) rather than map-coloring. Here is a story the second author used to introduce

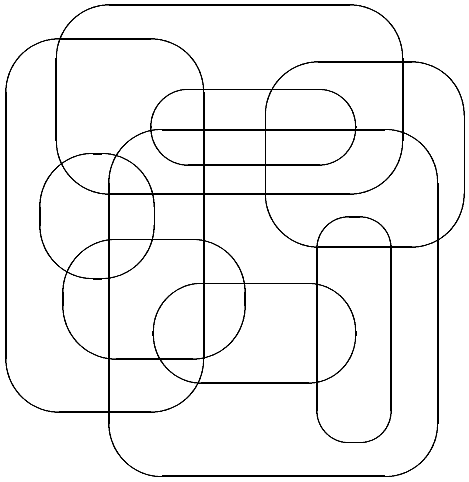

Figure 1: Example of a 2-colorable map generated by overlaying closed curves

graph-coloring in a 6th grade class. During the summer the merchants of Tourist Town decide to buy ice-cream machines, one for each street corner. The machines are inexpensive bottom-of-the-line devices, and each can dispense only one flavor of ice-cream. Suppose that they want to have enough different flavors at the different corners so that a tourist who doesn't happen to like the first flavor she comes to can continue walking in any direction, and the very next ice-cream machine will have a different flavor. What is the minimal number of flavors that must be ordered?

Note that one can introduce non-planar graphs without changing the story, by letting the town have "underpasses," i.e., streets which go by one another without intersecting.

2.2 Muddy City

Another excellent topic for children is the problem of computing a Minimum Weight Spanning Tree in a graph. Several efficient algorithms for solving this problem are known and are routinely covered in college level courses on design and analysis of algorithms. The story we use to present the problem is meant to be entertaining, but it should be noted that there are many practical applications of this problem.

The children are given a map of Muddy City and told the story of its woes

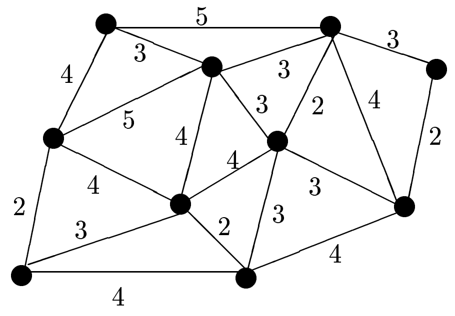

Figure 2: Muddy City

— cars disappearing into the mud after rainstorms, etc. The mayor insists that some of the streets must be paved, and poses the following problem. (1) Enough streets must be paved so that it is possible for everyone to travel from his or her house to anyone else's house — more precisely, from any street corner to any other street corner — by a route consisting only of paved roads, but (2) the paving should be accomplished at a minimum total cost, so that there will be funds remaining to build the town swimming pool. For the map shown in Figure 2 a solution of total cost 23 can be found.

The children typically work on the problem in small groups, with the objective of finding the best possible solution. As they obtain better and better solutions, the current best solution is posted in a place that everyone can see.

Remark. On occasion, the procedure does not work out exactly as planned. In San Vicente, El Salvador, where we presented Muddy City (*Ciudad Lodosa*) to a group of rural schoolgirls, we were unable to carry out this process of convergence to the best possible solution in the usual way, because, to our complete surprise, one of the *campesina* girls by the name of Abigail obtained the optimal solution within a few minutes. So what we did was to post the best non-Abigail solutions as they got closer and closer to her solution.

Sometimes students have been asked to describe their strategies and ideas as they worked and in a concluding discussion. In classrooms where the students kept mathematics journals, they also wrote down descriptions of the problem and of their ideas on how to solve it. These math journals can be a valuable part of an approach to teaching mathematics that emphasizes *mathematical communication*. Such an approach has been advocated by the National Council of Teachers of Mathematics in its reports on curriculum standards [19, 20].

As part of the wrap-up discussion, we sometimes presented Kruskal's algorithm, consisting simply of repeatedly paving a shortest street which does not form a cycle of paved streets, until no further paving is required. It is interesting that the children have often discovered some of the essential elements of Kruskal's algorithm and could offer arguments supporting them. (Rediscovering Kruskal's algorithm is not the point, of course.)

This problem can be presented to children of ages 5–6 by using maps with distances marked by ticks rather than numerals, so that the total cost of paving can be figured by counting rather than by sums.

2.3 Tourist Town

Minimum Dominating Set is another problem that can provide a nice illustration of the idea of computational complexity. Recall that a *dominating set* in a graph $G = (V, E)$ is a set of vertices $V' \subseteq V$ such that for every vertex x of G, either $x \in V'$ or x has a neighbor $y \in V'$.

The stories we have told for this problem generally run to the theme of *facilities location*. For example, in Tourist Town we now want to place ice-cream stands offering many flavors at street corners — but only at a few corners — so that no matter which corner you might be standing on, you need only walk at most one block to get an ice-cream. See Figure 3 for an example of a small, somewhat difficult graph for which the minimum size of a dominating set is 6.

We allow some time for the children to puzzle over the map of Tourist Town, gradually producing more efficient solutions. Often, none of them is able to find the optimal solution with only six ice-cream stands. The children usually get an intuitive sense that Tourist Town is harder than Muddy City; the former does not seem to lend itself to solution by a quick and simple algorithm. The contrast between these two problems — one quickly solvable by a simple recipe and the other apparently much more difficult — provides a concrete introduction to the notion of computational complexity. We will return to the subject of dominating sets (of a special kind) in §6.

Remark. As in the case of Muddy City, here also children can sometimes confound one's expectations. Recently the second author presented what seemed to be a difficult Tourist Town example (having a solution of 10 ice-cream stands) to a 6th grade class[3] at Washington Middle School in downtown Seattle. The class was warned that finding the minimum number of ice-cream stands in this

[3] Although Kid Krypto can be done at any level K–12, a case can be made that one should particularly target the 6th and 7th grades. In the U.S., this is a key age group in need of motivation. It is around the age of puberty that girls are being told that boys don't like girls who are smarter than them at math and science; and boys also are coming under increasingly negative influences of their peers and the surrounding anti-intellectual mass culture. So at that stage it is especially important for us to try to influence the youngsters in the direction of being enthusiastic about their studies.

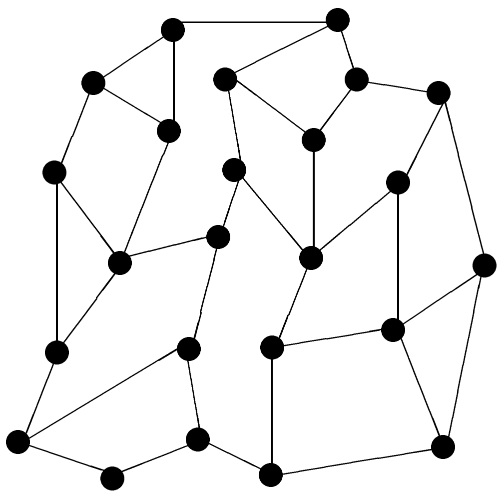

Figure 3: Map of Tourist Town

case was a *real hard problem*, and might take them hours at home. About 5 minutes later a youngster named Langston raised his hand and proudly displayed his completely correct optimal solution![4]

2.4 One-Way Functions

After explaining that no one knows a good algorithm for Tourist Town, one can show that there is, however, a simple algorithm for "working backwards," i.e., starting with a set of vertices V' that is to become an efficient solution and constructing a Tourist Town $G = (V, E)$ around it. Namely, one uses a two-step process. First, one forms a number of "stars" made up of "rays" (edges) emanating from the vertices in V'. (Two rays from different vertices in V' are allowed to have a common endpoint.) This graph clearly has V' as a solution.

[4] It is worth noting that we had specifically asked to visit classes in the non-advanced track. The children were about 80% Black and other minorities. No one had ever categorized these children as especially high-aptitude in mathematics. Yet here was Langston making us feel a little foolish by zeroing in so fast on an optimal solution.

It is a good idea to try out these ideas — and other ideas one might have for teaching math to children — in non-privileged classrooms, for example, in the non-advanced tracks of urban public schools. That gives a fairer test of how well the ideas work, and in some ways it can be especially rewarding. These children, after all, are not nearly as accustomed to enrichment presentations as are the children in the upper tracks and the wealthier schools.

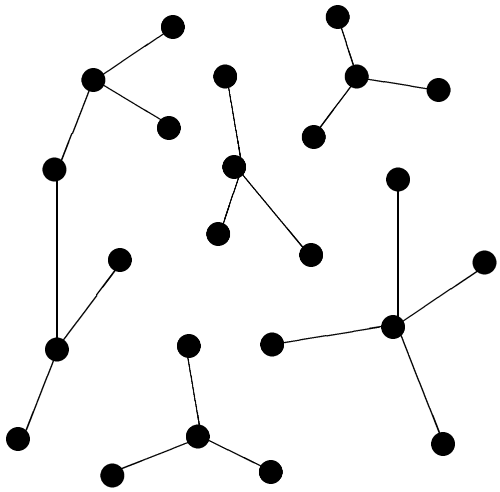

Figure 4: The first step in the construction of Tourist Town: a configuration of stars

Figure 4 shows this step in the case of the Tourist Town example in Figure 3. The second step is to “disguise” this easy-to-solve graph by adding more edges. This clearly does not increase the number of vertices required in a dominating set, but it does make the original built-in solution harder to see.

In this way it seems to be relatively easy to generate graphs on a small number of vertices (e.g. 25–30), having a known dominating set of size $6 \leq \gamma \leq 10$, for which it is relatively difficult to work out a solution of size γ by hand. However, no mathematical results are presently known that quantify the computational difficulty of problems such as this for graphs of small size.

This is a nice example of the idea of a one-way function. The children may look forward to trying out on their parents the process of creating a graph for which they secretly know a solution that their parents will find difficult to match.

Remark 1. If the two-step “hidden solution” construction described above is modified by

(1) in the first step, requiring that no two stars share a common vertex, and

(2) in the second step, requiring that the additional disguising edges

be added only between vertices not in V',

then the hidden solution will be a *perfect code* in $G = (V, E)$. (A more precise definition of a perfect code will be given later.) This modified construction is useful for the Perfect Code public key cryptosystem described in §6.

Remark 2. In presenting the Dominating Set problem to children in El Salvador, the authors had to confront an example of the general question of *cultural appropriateness* of the stories used to introduce these topics. We found that in El Salvador, as would be the case in many places in the world, the idea of minimizing the number of ice-cream stands makes no cultural sense whatsoever. The reason is that, in the first place, ice-cream sellers use movable carts, not fixed stands. Moreover, in any country with large unemployment, where much of the population depends on the so-called "informal economy" for their livelihood, there is always an overabundance of people available to sell ice-cream to tourists. The children would see no purpose in trying to minimize the number of ice-cream vendors.

So we changed the setting for the Dominating Set problem, presenting it by means of a story about minimizing the number of wells in order to achieve an efficient water supply for a village. Such a story is appropriate in a Third World context. Notice, however, that the story about wells would make no sense to children in industrialized countries.

2.5 Information Hiding Protocols

A simple illustration of an information hiding protocol is the following method for computing the average allowance of children in a classroom, without revealing any individual's allowance. More generally, the procedure described below can be used to find the average data for a group in a situation where the individuals in the group do not want their privacy compromised by revealing their own values for the numerical data.

The protocol goes as follows:

> The first child, Alicia, chooses a secret integer x, adds her allowance to it, and whispers the number $x + a_{\text{Alicia}}$ to her neighbor Berta.
>
> Berta adds her allowance to this number, and whispers the number $x + a_{\text{Alicia}} + a_{\text{Berta}}$ to Carmen.
>
> Carmen adds her allowance to this number, and ...
>
> ...
>
> ...
>
> The last child, Zinaída, adds her allowance and whispers the number $x + a_{\text{Alicia}} + a_{\text{Berta}} + a_{\text{Carmen}} + \cdots + a_{\text{Zinaida}}$ to Alicia.

Finally, Alicia subtracts x and divides the remainder by the number of children to determine the average for the group.

3 Digression on Math Education

In this section we discuss the educational context of the Kid Krypto project. Reform of mathematics education has been a subject of substantial public interest in recent years, as witnessed by a number of institutional reports and articles in the popular press. We will argue that topics in discrete math and cryptography have a great deal to offer in support of these reform efforts.

3.1 Traditional Grade School Mathematics Education

As viewed from the research milieu of mathematics, the following aspects of the traditional school mathematics curriculum are evident:
• A large number of repetitive short problems, each using a simple, low-level thought process.
• Immediate right/wrong gratification, as if there were an assumption that children are incapable of sustained efforts at mathematical problem-solving.
• No multi-layered problem-solving experiences.
• No problems that, instead of a single right answer, have *good* answers and *better* answers (as in some optimization problems).
• Archaic topics and terminology, some of which are virtually unchanged since the Middle Ages.
• Boring, concocted examples and applications.
• No frontiers, no discussion of the current limits of our knowledge of mathematics, no current events, no connections to the world of living mathematics.[5]
• Little independent activity, such as math projects, or even homework of a substantial nature.
• Silent individual seatwork, rather than communication of mathematical reasoning.
• Passivity: students are trained to follow a predictable short route to the correct answer, and do not contribute to developing the process of solution (e.g., setting up mathematical models of new situations, and formulating their own questions).
• No infinity, no logical paradoxes, no topology, no previews of coming attractions, nothing but "truths" which can be presented completely; no *science museum* mathematics.
• Narrow intellectual aspirations: a relentless focus on what the children in various opportunity tracks "need" to know, and on teaching geared to enhance their performance on short-answer standardized exams.

[5]That is why students when they reach the university are amazed to hear that there is such a thing as mathematical research with new developments all the time.

• Susceptibility to the commercial hype and gimmicry of textbook companies and others who are trying to sell something to the schools.[6]
• An atmosphere of anxiety and shame, which contrasts with the vivid curiosity and fascination with which children study such subjects as natural history (dinosaurs), astronomy (planets), and literature.

3.2 Current Reform Efforts

Fortunately, there are major efforts underway to change things. At present, the Curriculum Standards and the Teaching Standards documents of the National Council of Teachers of Mathematics [19, 20] (see also [17]) are the focus of much of the discussion of reform. In contrast to earlier curriculum documents in mathematics, which consisted of lists of specific topics for drill and test-oriented performance criteria, the new NCTM standards place an emphasis on *problem-solving, mathematical reasoning, communication, and real applications.* These "high-level" curriculum objectives represent an effort to orient mathematics education more towards mathematics as it is known by those who *do* mathematical science.

We believe that topics such as cryptography for children are an ideal vehicle for realizing these objectives. It is important to acknowledge, however, that there is much to be done in explaining to parents and educators why such enjoyable material is mathematics — when everyone "knows" that mathematics is arithmetic and algebra drill: hard work, intimidating, and boring. Below we address a few of the questions that have arisen in discussions with parents and teachers.

3.2.1 Map-coloring? This is mathematics?

Perhaps because it is so visual, this is one of the most popular topics of contemporary mathematics that one can do with young children. Presented with a hard instance to try to solve, children will often work on it for hours — it has proved to be an excellent topic for eliciting sustained concentration.

Moreover, the contrast between the easy algorithm for 2-coloring and the apparent difficulty of 3-coloring provides an opportunity on a naive level to share with children one of the most important unsolved problems in all of mathematics (the $P \neq NP$ conjecture).

[6] There are many drawbacks to the marketplace model of education. The private sector, oriented around the profit motive, has an excessive influence. There is a tendency towards fads and hype; the intrinsic value of a pedagogical idea is not as important as its saleability. Educational ideas that are not based on expensive gadgetry or new textbooks are not likely to be supported strongly. (See [12] for more discussion of this.) Short-range, anti-intellectual criteria prevail (boosting standardized test scores, competing with Japan). Finally, by analogy with the marketplace, where everything can be measured by a single numerical scale (money), there is pressure to adopt simplistic 1-dimensional criteria for success (number of correct multiple-guess answers, good grades, speed with which a student gets through certain material, and so on).

Yet this topic flies in the face of what most non-mathematicians have been conditioned to think of as mathematics. So one has to reassure them and calm their feelings of guilt that teaching math can be fun.

One way to do this is to describe the practical applications of graph coloring to such diverse tasks as the scheduling of committee meetings and the assignment of radio frequencies. These are not hard to explain, and seem to be particularly well received by parents and teachers. When concrete applications are presented to them, parents and teachers seem to be receptive to the idea that there are kinds of mathematics brought to the fore by computers that are different from the mathematics they are familiar with from their own school experiences.

3.2.2 Do children who will not be scientists really need to know this?

One can respond to this inevitable question with the rejoinder: "Who *needs* to read Huck Finn? Who *needs* to know about planets or dinosaurs?" When selecting material to be taught, there seems to be a tendency for mathematics to be judged by stingy criteria that are rightfully not applied to other sciences, to history or to literature.

What if literacy were taught *only* by means of parking tickets, job applications, tax forms and other material that people will *need* to read? That would be an accurate analogy to much of the traditional curriculum in mathematics.

3.2.3 What is your total philosophy of mathematics education?

The authors do not believe it is necessary to have a total philosophy of mathematics education before sharing some mathematical topics with children. Quite different pedagogical approaches might be appropriate to different age groups and in different social and educational contexts. For example, one might endorse an unstructured hands-on children's science museum approach to mathematics for grades K–6, while at the same time favoring the use of a more traditional, structured style of math teaching at the high school or college level for students who have already made the decision to pursue science or engineering.

3.2.4 Doesn't updating math education mean introducing computers?

On the contrary, it seems to us that there are dangers in the emphasis on using computers to teach math.

- They are expensive, and divert resources from other uses.
- Because of their cost, they further accentuate the division between have and have-not schools.
- People who make the decisions about purchasing the hardware and software are rarely able to evaluate carefully the claims of the salespeople. Because of

the tremendous amount of money that is at stake, people are often pressured into making poorly thought out decisions.

• Computers reenforce the fascination with gadgetry (as opposed to intellect) that is endemic in American popular culture.

• Computers are usually used in the classroom in a way that fosters a Golly–Gee–Whiz attitude that sees science as a magical black box, rather than as an area of critical thinking.

• The software is based on immediate gratification and very little creativity by the child.

• While physically playing an active role, in most cases the pupil is intellectually passive. That is, the pupil is programmed to follow a path already laid out in detail by others.

• Like a quack cure in medicine, perhaps the most harmful effect of the computer craze is that it diverts people from other, more solidly grounded approaches to treating what ails math education in America.

Most of the time, computers in the schools serve as little more than an expensive distraction. Many schools would probably be better off if they threw their computers into the dumpster.

It is regrettable that computers have been so aggressively marketed to teachers and school systems. In speaking to parents, teachers and school boards, many company representatives have taken the hard-sell approach: "If you don't buy our latest products you will be neglecting to prepare your children for the next century." Because of pressure from the companies and the media, computers have been fetishized to the extent that they threaten to become the Cargo Cult of the 21st century.[7]

The main beneficiaries of all the hype have been (1) computer hardware and software companies, and (2) educators who receive generous grants for the purpose of finding a way to use computers in the schools. It is quite possible that the Golly–Gee–Whiz–Look–What–Computers–Can–Do school of mathematical pedagogy will eventually come to be regarded as a disaster of the same magnitude as the "new math" rage of the 1960s.

[7]**Classical Cargo Cult** (see [15]): An isolated civilization comes into initial contact with European technology. Ignorant of modern science, they interpret the benefits of technology in terms of their familiar world and their familiar mode of operation. They pray and perform sacrifices, or do whatever seems to be necessary to induce the deities to bring them the Cargo.

Modern Cargo Cult: In the U.S., most of the general public — including a high percentage of teachers and an even higher percentage of school board members and educational bureaucrats — is *prescientific*, in the sense of having no rational understanding of the intellectual processes that go into scientific advances or their application to the real world. On the other hand, like the classical Cargo Cultists, they realize that technology is associated with economic wellbeing, and that something must be done so that youngsters will later be able to reap the benefits of the "computer age." The natural response, then, is to fetichize computers and fit them into the familiar world of traditional mindless school math.

3.3 Kid Krypto Is Best Done Without Computers

This is *crayon-technology* cryptography. The tools needed are: pencils, a lot of paper, crayons of different colors, and perhaps some pieces of string or sticks. There is no material obstacle to introducing Kid Krypto in poor school districts as well as rich ones — in Watts and Soweto as well as in Palm Beach and Scarsdale.

We see the absence of computers in Kid Krypto as a positive educational step. The public needs to understand that math and computer science is not about computers, in much the same way that cooking is not about stoves, and chemistry is not about glassware. That is,

$$\text{COMPUTER SCIENCE} \neq \text{COMPUTERS}.$$

The meaning of this inequality is: *What children need in order to become mathematically literate citizens in the computer age is not early exposure to manipulating a keyboard, but rather wide-ranging experience working in a creative and exciting way with algorithms, problem-solving techniques and logical modes of thought.*

3.4 Why Kid Krypto Is Especially Appropriate for Girls, Minority Children, and Third World Children

Among the organizations that have shown interest in our methods for presenting discrete mathematics and cryptography to children are the Kovalevskaia Fund (a foundation for women in science in developing countries) and the American Association of Historically Black Colleges. In this subsection we would like to argue that cryptography for children is especially appropriate in Third World countries and for female and minority students in the U.S.

First of all, in the case of minority and Third World communities, it has the obvious advantage of low cost. Kid Krypto is based on intellectual constructions rather than physical gadgets.

A second argument for Kid Krypto is based on the following analysis of how traditional math teaching reenforces white male supremacy:

• When taught as a boring, unpleasant subject in school, math education has much in common with a fraternity hazing. That is, it is a ritual that is pointless in itself (and painful), but which, if endured stoically, results in admission to an exclusive club — provided, of course, that the student is of the appropriate social and class background, race, and gender. Thus, a student who has reason to believe that he will be accepted in the club after the hazing is more inclined to endure the ritual than is a student who does not have that external incentive. This might be one reason why white male students on the average do better in math than female and minority children in middle school and high school (though *not* in the primary grades).

• To state this somewhat differently, if a student has many role models — people of the same race, gender, and class who show by their example that doing well in school math and science will result in a successful life — then the youngster will be motivated to do well even if the subject seems boring, difficult, and lifeless. On the other hand, a student who has few such role models is likely to work hard at a subject only if it seems intrinsically attractive and stimulating, i.e., only if it is well taught.

• If a subject is badly presented in school, then students will learn it only if they are strongly motivated for external reasons — parental and societal expectations, and encouragement from family and peers to do well in school math. On the other hand, if the subject is presented so as to be intrinsically interesting, then children are more inclined to work hard at it even in the absence of a lot of external motivation. It is well known that girls and minority children are less likely to have an environment of high expectations and encouragement to do well at math. Thus, for minority children and girls it is even more important than for white males that math be presented as something that is inherently interesting and that has clear connections with the real world and with human interaction.

• Although minority and economically disadvantaged children have even *more* need of stimulating ways to learn mathematics than privileged children, they seem to be given much *less* exposure to innovative teaching. A large part of the reason, according to a recent study undertaken for the National Science Foundation [16], is that in schools for the poor the teachers are under great pressure to teach for the standardized tests. As explained eloquently by Ruthie Green–Brown, principal of Camden High School:

> What is the result? We are preparing a generation of robots. Kids are learning exclusively through rote. We have children who are given no conceptual framework. They do not learn to think, because their teachers are straitjacketed by tests that measure only isolated skills. As a result, they can be given no electives, nothing wonderful or fanciful or beautiful, nothing that touches the spirit or the soul. Is this what the country wants for its black children?[8]

Cryptography can help answer Ms. Green–Brown's call for "fanciful" math topics that "touch the spirit or the soul." In addition to its challenging intellectual content, cryptography by its very essence entails human interaction and drama. No story has to be artificially constructed around a problem in cryptography, since, by definition, human drama is already implicit in any cryptography problem.

[8] Quoted in [13], p. 143; the student body of Camden High is almost entirely minority.

4 Cryptography With Props

Some cryptographic ideas can be effectively demonstrated by employing physical props. Such demonstrations can be useful in conveying the central concepts of cryptography to children and other mathematically unsophisticated audiences. Here is an elegant example, that was communicated to us by Adi Shamir.

A basic problem in cryptography is the Key Exchange problem. The objective is for two people, say Alice and Bobby, to agree upon an arbitrary sequence of bits that no one else knows, so that it can serve as a secret key with which to exchange messages (perhaps even a "one-time pad"). In the key exchange process, all communication between Alice and Bobby is in the open, so that an eavesdropper (Charlie) hears everything that one of them says to the other.

In Shamir's protocol we suppose that three playing cards of different value (say, Jack, Queen, and King) are repeatedly shuffled and dealt to Alice, Bobby, and the eavesdropper Charlie. Alice and Bobby agree in advance that, if they can both determine which of them has the higher card without Charlie knowing, then the next secret bit in the sequence will be:

$$\begin{cases} 1, & \text{if Alice has the higher card;} \\ 0, & \text{if Bobby has the higher card.} \end{cases}$$

Each time the cards are dealt, Alice tells Bobby one of the two cards that she does *not* have in her hand (she chooses one of the two possibilities at random). If Bobby has that card, he says so, and they move on to the next shuffle without having exchanged a secret bit. However, if Bobby does not have that card, he says "Charlie has that card," at which point Alice and Bobby both know which card everyone has, whereas Charlie has learned nothing. In that case, Alice and Bobby have agreed upon the next secret bit. Notice that the probability is that it will take $2n$ shuffles to produce a sequence of n bits.

Various other cryptographic concepts, such as oblivious transfer and multi-party secure computation, can be demonstrated by means of ordinary playing cards [4]. Note that these familiar physical objects have a number of cryptographically useful properties "built in": they have a convenient means of randomization (shuffling), they are uniquely identifiable, and when face down they are all indistinguishable. A number of research problems arise in constructing cryptosystems based on such physical primitives (see [4] for further discussion).

But a word of caution is in order. When designing a story or game to present a cryptographic concept, it is important not only to be entertaining but also to avoid any blatant fallacy. Children can often be quite perceptive in spotting a logical flaw. As an example of a well-intentioned but specious construction we cite the article "How to Explain Zero-Knowledge Protocols to Your Children" that was presented at the Crypto meeting in Santa Barbara in 1989 [1]. In a cleverly and humorously written essay, the authors explain how in the Strange

Cave of Ali Baba one could verify a claim of knowing the password to open the door without the secret being revealed. The cave had two passageways, and only someone who knew the password could enter one passageway, unlock the door, and emerge from the other passageway. The verifier would ask the person claiming to have the secret (the "prover") to go down either of the passageways (the verifier would not know which). Then the verifier would ask the prover to emerge from one of the two passageways picked at random. The verifier would know that if the prover did not possess the password, then with probability 1/2 he could not do this. So after k repetitions, in all of which the prover passes the test, the verifier can be certain with probability $1 - 2^{-k}$ that he does in fact have the password. There is only one objection to all of this, an objection that a precocious child might easily raise: Why not just send the prover down one passageway and demand that he emerge from the other one? There is no satisfactory answer to this question, and so the whole story loses its value as an example of cryptography for children.

For the remainder of this paper we shall be working with propless cryptosystems, i.e., cryptosystems that rely upon mental rather than physical constructions and assumptions. That is, our cryptosystems will depend upon the intractability of combinatorial problems rather than the existence of a perfect shuffling of cards, fair dice, fair coins, or an impenetrable door in a cave.

5 The Peruvian Coin Flip and Related Protocols

One of the key issues we must face in designing crayon-technology cryptosystems is: What interesting functions can 8-year olds or 12-year olds (for instance) compute reliably? That is, what sort of by-hand computing do we have available to work with?

With a little thought, we can see that interesting computations *can* be performed by children to provide the computational engines for cryptosystems. For example, the outputs of Boolean circuits can be computed; finite-state automata and Mealy machines can be operated. Cellular automata, if they are not too complicated, are also a possibility. Simple rewrite systems are another candidate for accessible calculations. The following protocol is based on Boolean circuits.

This protocol was first demonstrated by the authors with children in Peru (hence the name). The idea of trying out a crayon-technology cryptosystem in Peru seemed natural for several reasons. In the first place, the improvement of mathematics education is currently a hot topic of discussion among educators in Peru, as in much of the Third World. In the second place, developing countries (and international science development organizations such as the Kovalevskaia Fund) have a special interest in the possibility of enhancing math and computer science education in situations where computers are not available

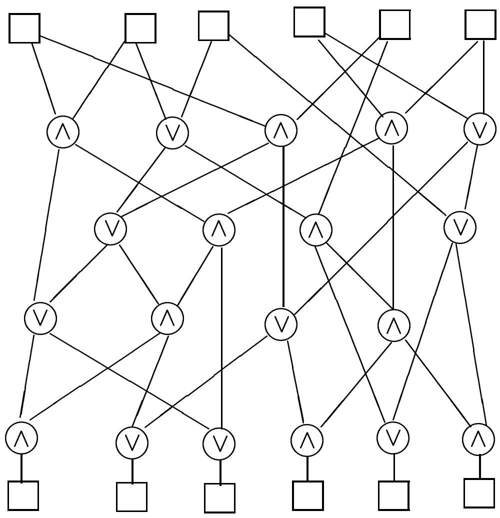

Figure 5: A Boolean circuit for the Peruvian coin-flip

and even money for textbooks is scarce.

5.1 The Coin Flip

We first told a story to explain how the need for such a coin-flip protocol might arise. The women's soccer teams of Lima and Cuzco have to decide who gets to be the home team for the championship game. Alicia, representing Lima, and Berta, representing Cuzco, cannot spend the time and money to get together to flip a coin. So they agree to the following arrangement.

Working together by telephone, they construct a Boolean circuit made up of and-gates and or-gates (for simplicity, we do not allow any large gates or not-gates). See Figure 5 for an example. In the construction process, each has an interest in ensuring enough complexity of the circuit so that the other will be unable to cheat (see below). The final circuit is public knowledge.

Alicia selects an arbitrary input string, which she keeps secret. She puts the string through the circuit, and sends Berta the output. Berta must then try to guess the *parity* of Alicia's input, i.e., the sum of its bits mod 2. If she guesses right, then the teams play in Cuzco. If her guess is wrong (which Alicia must demonstrate to her by revealing the input string), then they play in Lima.

Nothing in this description is hard to convey to a child of age 8 or above. Moreover, when we explain to the children the building blocks for the protocol

($\wedge$-gates and $\vee$-gates), we are talking about a really basic concept — perhaps *the* most basic concept — in formal logical thought. There is certainly as much justification for teaching children about $\wedge$-gates and $\vee$-gates as for long division and addition of fractions!

Remark. An alternative construction would be for Alicia and Berta each to construct a circuit with n input bits and m output bits. Both circuits would be public knowledge. Then Alicia would put her secret input through both circuits, and the final output would be the XOR of the outputs produced by the two circuits. This variant is "cleaner" in the sense that it avoids some interaction (i.e., an expensive telephone call between Lima and Cuzco during which Alicia and Berta construct a common circuit). Moreover, it is more convenient when one has a large group of people (as in a chess tournament or a classroom demonstration), since in this version each player makes a single circuit once and for all, after which the player can "toss coins" with several different people by simply exchanging copies of the circuits, without having to design a new circuit each time.

5.2 Cheating

Berta can cheat if she can invert the circuit, i.e., find the input (or inputs) that produce a given output. Alicia can cheat if she can find two inputs of opposite parity that produce the same output. It seems likely that both forms of cheating are infeasible if the circuit is large and complex.

If the circuit maps many-to-one, we claim that the ability to cheat in Berta's role implies the ability to cheat in Alicia's role. Namely, we have

Proposition 5.1 *Suppose we have a family $\mathcal{C}$ of many-to-one Boolean circuits, with the property that for any output the proportion of inputs in its preimage of given parity (odd or even) is bounded from below. Further suppose that one has an algorithm that inverts any circuit of $\mathcal{C}$ in time bounded by $f(n)$, where n is the size of the circuit. Then in time bounded by $k\big(f(n)+p(n)\big)$ (where p is a polynomial and k is a security parameter) one can find two inputs of opposite parity that give the same output.*

Proof. This result — both the statement and the proof — is completely analogous to the well-known result in number theoretic cryptography that states that the ability to take square roots modulo a composite number n implies the ability to factor n. Namely, to find the two desired inputs, select one input at random, and then apply the inversion algorithm to its output. With probability bounded from below, the inversion algorithm will give a second input of different parity for the same output. $\square$

On the other hand, we can entirely prevent Alicia from being able to cheat by choosing a circuit that maps inputs to outputs injectively, i.e., it effects an imbedding of $\{0,1\}^n$ into $\{0,1\}^m$. If we suppose that the circuit is complicated enough to behave like a random map, then the next proposition shows that it suffices to choose m somewhat larger than $2n$.

Proposition 5.2 *The probability that a random map from $\{0,1\}^n$ to $\{0,1\}^m$ is injective, is asymptotic to $1 - 2^{-(m-2n+1)}$ as $m - 2n \longrightarrow \infty$.*

Proof. This is a variant of a well-known combinatorial result (the "birthday paradox"). $\square$

5.3 An Open Question

In presenting the Peruvian coin-flip to a middle school audience, the authors encountered the situation where children attempted to evaluate an n-input/n-output circuit *upside down.* This leads to the following natural question, to which we do not know the answer. Let us suppose that all gates of our circuit have fan-out (as well as fan-in) of 2. (An alternative would be to allow large gates, i.e., gates with arbitrary fan-in and fan-out.) In addition, let us put $\vee$'s and $\wedge$'s in the input gates in an arbitrary way, with the understanding that such a gate (with a fan-in of 1) leaves the input bit unchanged. Under these assumptions the circuit makes sense if the child turns it upside down, of course with each $\vee$-gate now becoming a $\wedge$-gate and vice-versa. A natural question is whether it makes much difference (to a cheater) whether the circuit is right side up or upside down. More precisely, can one find a family of n-input/n-output circuits which are easy to invert, but which when turned upside down are hard to invert? Can the problem of inverting the circuits in some presumably hard-to-invert family $\mathcal{C}$ be shown to be polynomial time equivalent to the problem of inverting the upside down circuits of $\mathcal{C}$?

5.4 Hash Functions and Signature Schemes

Now suppose that the input is much longer than the output. If $n \gg m$ in the Boolean circuit, then the map from strings of n bits to strings of m bits may be used as a *hash function.* Roughly speaking, a hash function is a map $f : x \mapsto y$ from a very long input x to a much shorter output y that has the following property:

> it is not computationally feasible to find two different
> inputs x and x' such that $f(x') = f(x)$.

One of the main applications of hash functions is in signature schemes. Suppose that Alice sends Bob a long message x (say, 10^6 bits), and they have both agreed to use a hash function f, where $f(x)$ has about 500 bits. Alice wants

Bob to be able to convince himself that it was truly Alice who sent the message x, and that this message has not been tampered with.

We will illustrate with the RSA signature scheme from adult cryptography (which is not accessible to children because of the number theory required and the need for computer manipulation of very large integers). So let us suppose that Alice and Bob belong to an RSA signature network. This means that each user (in particular, Alice) has a public key (n, e) consisting of a composite number $n = p{\cdot}q$ (where p and q are primes of roughly 300 bits) and an encryption exponent e. Only the particular user Alice knows the factorization of her n and the decryption exponent d (her private key) that satisfies $ed \equiv 1 \pmod{(p-1)(q-1)}$. After sending Bob the message x, Alice "signs" the message in the following way: first she hashes it using f; then she raises $y = f(x)$ to the d-th power modulo n, and sends the result y' to Bob. After receiving the message x, Bob also computes $y = f(x)$, and then raises y' to the e-th power modulo n. If the result agrees with y, then he knows that Alice must in fact have sent him the message x. He knows this because (1) no adversary would have been able to tamper with the message x without changing the hash y; and (2) no one other than Alice would know the deciphering exponent d that is "undone" by raising to the e-th power.

It would be nice to have a signature scheme to use with children in conjunction with the Boolean circuit hash function. One could easily devise interesting stories and games around tamper-proof messages. But unfortunately, none of the signature schemes known to the authors are accessible to children.

Open problem. Find an efficient, secure, and accessible combinatorially based signature scheme.

Prize. In the tradition of Paul Erdös, we are prepared to put money where our mouths are. For the first person to solve this open problem, the second author will donate \$100 in that person's name to his/her favorite charity or educational organization; and the first author will inscribe the person's name in the Highest Honor List of the SIGACT Compendium Project (of the Committee on Education of the ACM Special Interest Group on Algorithms and Computation Theory). In order to qualify, the solution must be elegant and accessible to schoolchildren.

6 Perfect Code Cryptosystems

The public key system which we will describe in this section can be designed with different levels of accessibility and security. The simplest version, which will be described first, can be mastered by a child who understands only (1) the simplest properties of graphs, and (2) addition (say, modulo 2 or modulo 26). We shall next describe a more complicated version, appropriate for older

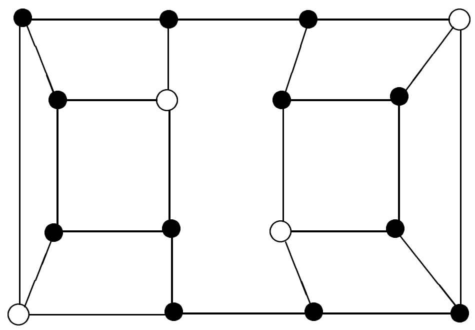

Figure 6: Example of a perfect code in a 3-regular graph

children. Then we discuss the most general version; we know of no algorithm to crack this last version in polynomial time.

We begin by considering a special kind of dominating set in a graph called a *perfect code*. In what follows, if u is a vertex of a graph $G = (V, E)$, then the notation $N[u]$ (the "neighborhood" of u) denotes the set of vertices which share an edge with u (including u itself).

Definition 6.1 *A set of vertices $V' \subseteq V$ in a graph $G = (V, E)$ is said to be a perfect code if for every vertex $u \in V$ the neighborhood $N[u]$ contains exactly one vertex of V'.*

Figure 6 shows an example of a graph with a perfect code. The vertices of the perfect code are indicated by open circles.

Remark 1. Jan Kratochvíl has shown that the problem of determining whether a graph has a perfect code is NP-complete for r-regular graphs, for all $r \geq 3$ [14].

Remark 2. An interesting detour for the children along the way to our cryptosystem might be to investigate error-correcting codes. For example, let n be of the form $2^k - 1$, and let G be the hypercube graph, whose vertices are $\{0,1\}^n \subset \mathbf{R}^n$ and whose edges are the edges of the n-dimensional unit hypercube. Then a *binary Hamming code* of length $n = 2^k - 1$ and dimension $d = 2^k - k - 1$ corresponds to a perfect code of 2^d vertices in G. For example, when $k = 2$, the (unique) Hamming code is the pair of opposite vertices (0,0,0) and (1,1,1) on the ordinary cube.

6.1 Version 1 of the Perfect Code Cryptosystem

This version is accessible to elementary school children. Suppose that the children have already mastered the Pre-Crypto topic *construction of a graph that has a well-disguised perfect code* (see the first remark of §2.4). Now Alice wants to be able to receive an encrypted bit from Bobby. She constructs a graph $G = (V, E)$ with a perfect code V'. The graph G is her public key. Her private key is V'.

To send a bit b, Bobby makes a random assignment of 0's and 1's to all of the vertices of G except one. He then assigns either a 0 or 1 to the last vertex in such a way that the sum mod 2 over all of the vertices is equal to b. Next, he replaces the bit c_u assigned to each vertex u by a new bit c'_u determined by summing (mod 2) all of the bits previously assigned to the neighboring vertices: $c'_u = \sum_{v \in N[u]} c_v$. He finally returns the graph to Alice with the bits c'_u annotating the vertices.

To decipher the message, Alice takes the sum of c'_u over the perfect code; that is, she has $b = \sum_{u \in V} c_u = \sum_{u \in V'} c'_u$, where the last equality follows from the definition of a perfect code.

6.2 Version 1′

The same as Version 1, but we make it more interesting by working modulo 26, so that Bobby can send Alice an enciphered letter $b \in \{A = 0, \ldots, Z = 25\}$.

Remark. Even if G is a complicated graph, both Versions 1 and 1′ of this cryptosystem can be broken in polynomial time using linear algebra (Gaussian elimination) modulo 2 (respectively, modulo 26). This will be shown later as a special case of a more general result. However, young children have no more knowledge of how to do this than we adults have of how to factor integers in polynomial time. So with a judicious choice of G, Versions 1 and 1′ appear to be accessible starting in elementary school and secure at least through middle school or high school.

Note that we are introducing a new, relativized notion of security of a cryptosystem. If the techniques needed to implement the system are accessible to a certain group of people, whereas the math needed to crack it is not, then we say that the system is *secure for the given class of people.*

6.3 Version 2

We next describe a more elaborate version which is probably accessible and secure in high school.

First we need some definitions and notation. Suppose that F is a ring (for example, the ring of integers or the ring of integers modulo m),and we are working with polynomials over F in a certain set of variables. Given a subset of those variables, we define the *value of a polynomial on the subset* to be the value obtained if the variables in the subset are set equal to 1 and the rest of the variables are set equal to 0.

In particular, given a graph $G = (V, E)$, we assign a variable denoted x_u to each vertex $u \in V$. Suppose that G has a perfect code V'. Given any polynomial $f \in F[\{x_u\}]$, we define its value at the perfect code V' to be the value obtained by setting

$$x_u = \begin{cases} 1, & \text{if } u \in V'; \\ 0, & \text{otherwise.} \end{cases}$$

Notice that even though Bobby does not know Alice's perfect code, he knows that for any vertex u the expression $\sum_{v \in N[u]} x_v$ has value 1 at her perfect code. By combining such expressions, he can form a very complicated polynomial f whose value at Alice's perfect code is known to him — and can be determined by Alice, who knows which variables to set equal to 1 and which to set equal to 0 — but presumably cannot be found by an eavesdropper who knows neither Alice's perfect code nor the manner in which Bobby formed f from the sums of the form $\sum_{v \in N[u]} x_v$. This is the idea of Version 2.

More precisely, suppose Bobby wants to send a message b, which is a certain integer modulo m. For some k, Bobby chooses an arbitrary set I of subsets of vertices $S \subseteq V$, $\#S \leq k$, and a corresponding set of integers c_S such that $\sum_{S \in I} c_S \equiv b \pmod{m}$. He then forms the following polynomial over the ring $F = \mathbf{Z}/m\mathbf{Z}$:

$$f = \sum_{S \in I} c_S \prod_{u \in S} \sum_{v \in N[u]} x_v .$$

Since each inner sum evaluates to 1 at any perfect code, the whole expression obviously evaluates to $\sum c_S = b$.

Bobby does three things to disguise the manner in which f was formed: (1) he combines terms; (2) he replaces all higher powers of a variable by its first power; and (3) he deletes any monomial in which two variables occur that correspond to vertices whose distance from one another in the graph G is ≤ 2 (because even without knowing Alice's perfect code, he knows that those vertices could not both belong to it, and hence the monomial must evaluate to 0).

Remark. Versions 1 and $1'$ are special cases of Version 2 where $k = 1$, i.e., I consists of one-element sets $S = \{u\}$. Then

$$f = \sum_{u \in V} c_u \sum_{v \in N[u]} x_v = \sum_{u \in V} c'_u x_u, \qquad \text{where} \qquad c'_u = \sum_{v \in N[u]} c_v .$$

6.4 Breaking Versions 1, $1'$ and 2

Given a polynomial f in the variables x_u, we want to find a relation of the form

$$f = \sum_{\#S \le k} c_S \prod_{u \in S} \sum_{v \in N[u]} x_v,$$

which holds after replacing all higher powers of a variable by the first power and deleting monomials which contain two variables corresponding to vertices at a distance ≤ 2 in G. We regard the c_S as unknowns, and equate coefficients of each monomial on the left and right. There are $\sum_{j=0}^{k} \binom{n}{j}$ unknowns c_S (here $n = \#V$ is the size of the graph), and there are an equal number of monomials of total degree $\le k$ (in which the variables occur at most to the first power), and hence at most the same number of equations. (Actually, there will be fewer equations, because we drop any monomial in which two variables occur corresponding to vertices that are at a distance ≤ 2 from one another.) We know that the system of linear equations has a solution, because the f in Version 2 was constructed as such a sum of products. The solution can be found by Gaussian elimination. (In practice, the system of equations will probably be sparse, in which case special methods are available.)

Notice that if k is unbounded, then the time required to do the linear algebra is not polynomial in the size n of the graph. However, the time is polynomial in the size of the polynomial f that Bobby sends to Alice, unless he has some way of producing sparse polynomials f (polynomials f with mostly zero coefficients). We will turn to the construction of sparse polynomials when we discuss the more general version of the Perfect Code cryptosystem.

Remark. In implementing these cryptosystems, the youngsters have to build up complicated f, using the distributive law and gathering similar terms so as to disguise the way f was formed. In this way Kid Krypto might add some excitement to the subject of polynomials, which is often presented in school in a dry, unmotivated manner. The decision as to what version of Perfect Code cryptography to use — how complicated to make the possible f — depends on the age of the children and their ability to keep track of a lot of data.

6.5 3-Regular Graphs

One way to keep the level of difficulty under control is to use only regular graphs of degree 3. Then there are exactly 4 variables in each neighborhood. The class of 3-regular graphs is still plenty complicated to support these cryptosystems — as mentioned before, determining whether a given 3-regular graph has a perfect code is NP-complete. We now describe a simple one-way construction (different from the method with "stars" that was described in §2.4) that gives a large class of 3-regular graphs having perfect codes. The construction is based on covering spaces of K_4, the complete graph on 4 vertices.

The construction is as follows. Let $n = 4n_0$ be the size of the 3-regular graph to be constructed. Select four sets of n_0 vertices each, which we denote A, B, C, D. Then randomly create six one-to-one correspondences between the sets: $A \approx B$, $A \approx C$, $A \approx D$, $B \approx C$, $B \approx D$, $C \approx D$. Draw edges between vertices that are associated under any of these six bijections. Let $G = (V, E)$ be the resulting graph. Notice that each neighborhood $N[u]$ contains exactly one vertex from each of the sets A, B, C, D; thus, each of these sets is a perfect code in G. The construction is completely general: every covering space of K_4 can be produced in this way. It is not known whether the problem of recovering such a vertex set partition for a graph that is known to be a cover of K_4 is difficult in the sense of average-case complexity. The problem of deciding whether an arbitrary graph is a cover of K_4, however, has been shown to be NP-complete [14].

6.6 Version 3

We now discuss a much more general version of Perfect Code cryptography, which may be secure even in the sense of adult cryptography. Before describing the cryptosystem, we give some definitions and prove some results about the most general types of invariant polynomials.

In the definition below, we use the term "invariant" to refer to a polynomial $\widetilde{f}$ in n sets of variables which has the following property. If n variables are chosen, one from each set, and are set equal to 1, and if all of the remaining variables are set equal to 0, then the resulting value of $\widetilde{f}$ does not depend on which variable was chosen from each of the n sets of variables.

Let $G = (V, E)$ be a graph of size $n = \#V$, and let F be a ring. Let $\{x_v\}$ be a set of variables indexed by V, and let $\{\widetilde{x}_{u,v}\}$ be a set of variables indexed by the set of ordered pairs $u, v \in V$ for which $v \in N[u]$. Let B denote the F-module generated by all monomials in the $\widetilde{x}_{u,v}$ having the properties that (1) each variable occurs at most to the first power, and (2) if $\widetilde{x}_{u,v}$ and $\widetilde{x}_{u',v'}$ both occur, then $u \neq u'$. Let φ denote the map from B to $F[\{x_v\}_{v\in V}]$ that takes $\widetilde{x}_{u,v} \mapsto x_v$, then replaces every higher power of a variable x_v by x_v to the first power, and finally replaces a monomial by 0 if it contains two variables x_u and x_v corresponding to vertices u and v that are at a distance ≤ 2 in G.

An element $\widetilde{f} \in B$ is said to be an *invariant prepolynomial on* G if it has the following property: If $g: V \longrightarrow V$ is an arbitrary map such that $g(u) \in N[u]$ for all $u \in V$, then the value of $\widetilde{f}$ at the set of variables $\left\{\widetilde{x}_{u,g(u)}\right\}_{u\in V}$ is independent of the map g. A polynomial $f \in F[\{x_v\}]$ is said to be an *invariant polynomial on* G if it is the image of an invariant prepolynomial $\widetilde{f}$ under the map φ.

Let $\widetilde{A}(k) = \widetilde{A}(G, k) \subseteq B$ be the F-module of invariant prepolynomials on G of degree $\leq k$, and let $A(k) = A(G, k) = \varphi\big(\widetilde{A}(G, k)\big)$. Set $\widetilde{A} = \widetilde{A}(n)$, $A = A(n)$, i.e., the F-modules of all invariant (pre)polynomials. Set $\widetilde{m}(k) = \operatorname{rank} \widetilde{A}(k)$, $m(k) = \operatorname{rank} A(k)$, $\widetilde{m} = \widetilde{m}(n)$, $m = m(n)$. When we want to indicate the dependence on G we write $m(G,k)$, etc.

For fixed r, we let $\widetilde{m}(n,k)$ denote $\widetilde{m}(G,k)$ for any r-regular graph $G = (V,E)$ on n vertices. Note that, by definition, $\widetilde{m}(G,k)$ depends only on the set $\{\#N[u] \mid u \in V\}$, e.g., the set $\{r+1, r+1, \ldots, r+1\}$ in the case of an r-regular graph. That is, $\widetilde{m}(G,k)$ does not depend on the particular structure of the graph G. Moreover, the definition of an invariant prepolynomial makes sense for any n-tuple of natural numbers (i.e., any n sets of variables), whether or not a graph G exists with the particular n-tuple as its set $\{\#N[u] \mid u \in V\}$. For example, $\sum_{u \in V}(\#N[u]-1) = 2\#E$ cannot be odd. However, we shall use the notation $\widetilde{m}(n,k)$ even when no graph G exists (e.g., r and n are both odd).

Proposition 6.1 $\widetilde{m} = 1 + \prod_{u \in V}\left(1 + \#N[u]\right) - \prod_{u \in V} \#N[u]$. *In particular, if G is an r-regular graph, then* $\widetilde{m} = 1 + (r+2)^n - (r+1)^n$.

Proof. Let $C \subset B$ be the F-module spanned by monomials of degree n. Then the proposition states that $\operatorname{rank} \widetilde{A} = 1 + \operatorname{rank} B - \operatorname{rank} C$. (To see the product formulas for $\operatorname{rank} B$ and $\operatorname{rank} C$, note that for each $u \in V$ one has $\#N[u]$ choices of variable $\widetilde{x}_{u,v}$ when forming a monomial in C, and $\#N[u]+1$ choices when forming a monomial in B, including the option of taking none of the variables.) We have an exact sequence of linear maps

$$0 \longrightarrow F \longrightarrow \widetilde{A} \xrightarrow{\alpha} B \xrightarrow{\beta} C \longrightarrow 0,$$

where the map from F to $\widetilde{A}$ is the inclusion of constant polynomials; α is defined by $\widetilde{f} \mapsto \widetilde{f} - \widetilde{f}(g)$, where $\widetilde{f}(g)$ denotes the value of f corresponding to (any) map g (see the definition of an invariant prepolynomial); and β takes a polynomial f to $\sum_g \left(f(g) \prod_{u \in V} \widetilde{x}_{u,g(u)} \right)$. It is easy to check that the sequence is exact. In any exact sequence, the alternating sum of the ranks is equal to zero. Hence $1 - \operatorname{rank} \widetilde{A} + \operatorname{rank} B - \operatorname{rank} C = 0$, proving Proposition 1. $\square$

We now define a generalization of the binomial coefficients as follows. For any integer parameters a, b and for integers $0 \leq k \leq n$, let $C(n,k,a,b)$ denote the entries in the generalized Pascal triangle formed using the recurrence relation

$$C(n,k,a,b) = C(n-1,k,a,b) + a\,C(n-1,k-1,a,b)$$

with the following boundary conditions along the sides of the triangle:

$$C(n,0,a,b) = 1, \qquad C(n,n,a,b) = b^n.$$

Of course, the usual binomial coefficients are $\binom{n}{k} = C(n,k,1,1)$.

Proposition 6.2 *Suppose that G is an r-regular graph of size n. For $1 \le k \le n$ let $m^*(n,k)$ denote 1 plus the corank of the module of invariant prepolynomials of degree $\le k$ in the module of all prepolynomials in B of degree $\le k$. Then $m^*(n,k) = C(n,k,r,r+1)$.*

Proof. Obviously, $m^*(n,0) = 1$. That $m^*(n,n) = (r+1)^n$ follows from Proposition 6.1. We next establish a recurrence relation for $\widetilde{m}(n,k)$, from which the desired recurrence relation for $m^*(n,k)$ will follow.

Fix a vertex u_0, and let $y_1, \dots, y_{r+1}$ be the variables $\widetilde{x}_{u_0,v}$ for $v \in N[u_0]$. Then any element of $\widetilde{A}(n,k)$ can be constructed as follows. Let f_{r+1} be an arbitrary prepolynomial in B of degree $\le k-1$ in which the variables $y_1, \dots, y_{r+1}$ do not appear; such f_{r+1} form a module of rank $\sum_{j=0}^{k-1} \binom{n-1}{j}(r+1)^j$. Let f_0 be an arbitrary invariant prepolynomial of degree $\le k$ in which the variables $y_1, \dots, y_{r+1}$ do not appear; such f_0 form a module of rank $\widetilde{m}(n-1,k)$. Finally, for $1 \le i \le r$ let f_i be an arbitrary invariant prepolynomial of degree $\le k-1$ which evaluates to 0 for any map g (see the definition of an invariant prepolynomial) and in which the variables $y_1, \dots, y_{r+1}$ do not appear; for each i such f_i form a module of rank $\widetilde{m}(n-1,k-1)-1$. Then

$$f_0 - f_{r+1} + f_{r+1}y_{r+1} + \sum_{i=1}^{r}(f_{r+1}+f_i)y_i$$

is an invariant prepolynomial of degree $\le k$ on the graph G, and any invariant prepolynomial of degree $\le k$ can be uniquely obtained in this way. Thus,

$$\widetilde{m}(n,k) = \widetilde{m}(n-1,k) + r\big(\widetilde{m}(n-1,k-1)-1\big) + \sum_{j=0}^{k-1}\binom{n-1}{j}(r+1)^j. \quad (1)$$

Now

$$m^*(n,k) = 1 + \sum_{j=0}^{k}\binom{n}{j}(r+1)^j - \widetilde{m}(n,k) \quad (2)$$

by the definition of $m^*(n,k)$, and also

$$\sum_{j=0}^{k}\binom{n}{j}(r+1)^j = \sum_{j=0}^{k}\binom{n-1}{j}(r+1)^j + \sum_{j=1}^{k}\binom{n-1}{j-1}(r+1)^j =$$

$$= \sum_{j=0}^{k}\binom{n-1}{j}(r+1)^j + (r+1)\sum_{j=0}^{k-1}\binom{n-1}{j}(r+1)^j. \quad (3)$$

Combining (1)–(3) gives the desired recurrence relation for $m^*(n,k)$:

$$m^*(n,k) = m^*(n-1,k) + r\,m^*(n-1,k-1),$$

and Proposition 6.2 is proved. □

Remark 1. It is not the case that $\widetilde{A}(n,k)$ is spanned by homogeneous polynomials. Here is the simplest counterexample. Take $r=1$, $n=2$, i.e., G consists of two vertices u, v connected by an edge. The homogeneous invariant prepolynomials of degrees 0, 1, and 2 are spanned by: 1; $x_{u,u}+x_{u,v}$ and $x_{v,u}+x_{v,v}$; and $x_{u,u}x_{v,u}+x_{u,u}x_{v,v}+x_{u,v}x_{v,u}+x_{u,v}x_{v,v}$. But $x_{u,u}+x_{u,v}x_{v,u}+x_{u,v}x_{v,v}$, which is also invariant, is not in the span of these four polynomials.

What we really want to know, however, is not the ranks of spaces of invariant prepolynomials, but rather the ranks of spaces of invariant polynomials. We do not have exact formulas, since even in the case of r-regular graphs these ranks depend not just on n, k and r but also on the particular structure of G. Proposition 6.3 below gives an estimate for these ranks.

Remark 2. The drop in rank from $\widetilde{A}$ to A can be dramatic. For example, if G is a 2-fold covering of K_{r+1} (see §6.5), then the rank $\widetilde{m}$ of the space of all invariant prepolynomials is $(r+2)^{2r+2}-(r+1)^{2r+2}+1$, while the rank m of the space of all invariant polynomials is only $2r+4$. For instance, for $r=3$ we have $\widetilde{m}=325090$, $m=10$.

Before stating the proposition on the ranks of the spaces of invariant polynomials, we introduce some notation. Let $m_0(G,k)$ denote the rank of the space of all invariant polynomials on G of degree $\le k$. Note that $m_0(G,k)\ge m(G,k)$, with strict inequality in cases where one has invariant polynomials of degree $\le k$ that are of the form $\varphi(\widetilde{f})$ only for invariant prepolynomials $\widetilde{f}$ of degree $> k$. Next, let r_1 denote the number of vertices in the smallest neighborhood in G, i.e.,

$$r_1=\min_{u\in V}\#\{v\in V\;\big|\;v\in N[u]\},$$

and let r_2 denote the number of vertices in the largest neighborhood of radius 2, i.e.,

$$r_2=\max_{u\in V}\#\{v\in V\;\big|\;\mathrm{dist}(u,v)\le 2\}.$$

Proposition 6.3 (a)

$$\sum_{j=0}^{k}\binom{n}{j}\ge m_0(G,k)\ge m(G,k)\ge\sum_{j=0}^{k-1}r_2^{-j}\binom{\frac{n-r_1}{r_2}}{j},$$

where $\binom{x}{j}$ is interpreted to be equal to zero if $x\le j-1$.

(b) *If $\mathcal{C}$ is a family of graphs of size $n\longrightarrow\infty$ in which r_2 and $k/\sqrt{n}$ are bounded from above by an absolute constant, then*

$$1+\frac{n^k}{(k-1)!}\ge m_0(G,k)\ge m(G,k)\ge c\frac{n^{k-1}}{(k-1)!}$$

for $G\in\mathcal{C}$, $k\le n/2$, and for some absolute constant $c>0$.

Proof. Part (b) is an immediate consequence of part (a): for $k \le n/2$ the inequality on the left follows trivially from the inequality on the left in part (a); the inequality on the right follows because the last summand on the right in part (a) is bounded from below by

$$\frac{n^{k-1}}{(k-1)!}\left(1-\frac{(k-1)r_2}{n}\right)^{k-1},$$

and the factor appearing with $n^{k-1}/(k-1)!$ is bounded from below by a constant if $k = O(\sqrt{n})$.

We now prove part (a). The upper bound follows simply because, if we ignore both the invariant condition and the collapsing to zero that occurs when a monomial has variables corresponding to vertices at a distance ≤ 2, then the number of monomials of degree j is equal to the number of subsets of j vertices, i.e., $\binom{n}{j}$. To prove the lower bound, let u_0 be a vertex whose neighborhood has only r_1 vertices. Then for each $j \le k-1$ we bound from below the number of monomials of degree j whose variables x_u correspond to distinct vertices $u \notin N[u_0]$ such that no two of the u are at a distance ≤ 2 from one another. The number of such subsets $\{u_1, \ldots, u_j\}$ of j vertices is

$$\ge \frac{(n-r_1)(n-r_1-r_2)(n-r_1-2r_2)\cdots(n-r_1-(j-1)r_2)}{j!} = r_2^{-j}\binom{\frac{n-r_1}{r_2}}{j}.$$

Given any linear combination of the monomials $x_{u_1}\cdots x_{u_j}$, we take the prepolynomial $\widetilde{f}$ to be the corresponding linear combination of the monomials $\widetilde{x}_{u_1,u_1}\cdots\widetilde{x}_{u_j,u_j}$. Then $\widetilde{f}\left(1-\sum_{v\in N[u_0]}\widetilde{x}_{u_0,v}\right)$ is an invariant prepolynomial of degree $\le k$ whose image under φ, i.e., $f\left(1-\sum_{v\in N[u_0]}x_v\right)$, is an element of $A(G,k)$. Since the resulting invariant polynomials are distinct, this gives us the lower bound in the proposition. □

6.7 Construction of Polynomials for Version 3

The proof of Proposition 6.2 contains a recursive recipe for constructing an arbitrary invariant prepolynomial of degree $\le k$ on an arbitrary graph of size n. Choose a vertex u_0, whose neighborhood consists of $r+1$ vertices (but G is no longer assumed to be r-regular); let y_i be the corresponding variables. Assume by induction that we can construct invariant prepolynomials in which the variables $y_i = \widetilde{x}_{u_0,v}$ do not occur. Construct an arbitrary (not necessarily invariant) prepolynomial f_{r+1} of degree $\le k-1$ in which the variables y_i do not appear. Now choose an invariant prepolynomial f_0 of degree $\le k$ and invariant prepolynomials f_i for $1 \le i \le r$ of degree $\le k-1$, as in the proof of Proposition 6.2. Then the desired invariant prepolynomial is

$$f_0 - f_{r+1} + f_{r+1}y_{r+1} + \sum_{i=1}^{r}(f_{r+1}+f_i)y_i =$$

$$= f_0 + \sum_{i=1}^{r} f_i y_i - f_{r+1}\left(1 - \sum_{i=1}^{r+1} y_i\right).$$

In sending a message to Alice, Bob's last step is to apply the map φ from prepolynomials in the variables $\widetilde{x}_{u,v}$ to polynomials in the variables x_v, and also to add a constant so that Bob's final polynomial f evaluates to the message b that he wants to send to her. Notice that as he constructs his invariant prepolynomial by the above recipe, Bob can easily keep track of how the prepolynomial evaluates at an arbitrary map g in the definition of invariance — and hence how the later polynomial in the x_v will evaluate at a perfect code — without knowing Alice's perfect code.

Remark. To reduce running time and space, in practice Bob should not wait until the end of the procedure to apply φ. Rather, at each step he should map the variables $\widetilde{x}_{u,v}$ to x_v and collapse whatever terms he can.

Suppose Charlie wants to crack the cipher. He can do this using linear algebra as follows. First, Charlie goes through the same recipe as Bob a large number of times, constructing $N \gg m(G,k)$ random invariant polynomials f_i. Since he has far more polynomials than the rank of the space $A(G,k)$ of all such polynomials, it is almost certain that his set of invariant polynomials spans $A(G,k)$, and so Charlie can find a linear combination $\sum c_i f_i$ that equals Bob's polynomial f. This involves equating coefficients of all of the monomials, and solving the linear equations for the unknowns c_i.

For variable k, the running time of Charlie's cracking algorithm is not polynomial in the size n of the graph. But it is clearly polynomial in $m(G,k)$, which, by Proposition 6.3, has order of magnitude roughly $\binom{n}{k}$. Note that a random implementation of the recipe by Bob will lead to an invariant polynomial with roughly $m(G,k)$ nonzero monomial terms, i.e., his ciphertext will have length of this order; and so the running time of such an implementation must also be at least $m(G,k)$. For purposes of adult cryptography, one does not want a system which can be broken in time polynomial in the ciphertext length (or in the length of time needed to create the ciphertext).

However, this objection to the cryptosystem is no longer valid if Bob uses a "sparse" implementation of the recipe that gives sparse invariant polynomials, i.e., polynomials with a relatively small number of nonzero monomial terms. Let $r+1$ be an upper bound on the size of the neighborhoods of the vertices of G. We now describe a random process to construct invariant prepolynomials of degree k that requires only $O\big((r+1)^k\big)$ operations.

The description is recursive. Let u_0 be a randomly chosen vertex, and let $y_1, \ldots, y_{r'}$, where $r' \le r+1$, be the variables $\widetilde{x}_{u_0,v}$ for $v \in N[u_0]$. We assume, by induction, that we can construct invariant prepolynomials of degree $k-1$ in time $O\big((r+1)^{k-1}\big)$. Let $f_1, \ldots, f_{r'}$ be r' such prepolynomials of degree $k-1$. Then $\sum_{i=1}^{r'}(f_i - a_i)y_i$ is the desired invariant prepolynomial of degree k, constructed in time $O\big((r+1)^k\big)$, where the a_i are constants chosen so that the different $f_i - a_i$ evaluate to the same value.

The polynomials coming from these prepolynomials are not satisfactoryas ciphertext, because of the possibility that Charlie can guess the vertex u_0,[9] and then work backwards inductively and recover Bob's construction. However, if Bob takes the sum of a large number of invariant polynomials constructed in this way (with many different u_0), then it is not clear (at least not to the authors) how Charlie could proceed. For example, suppose Bob lets u_0 range over a large proportion of the vertices in the graph, and comes back to some u_0 two or three times. That is, Bob takes as his ciphertext the sum of $O(n)$ polynomials constructed as in the preceding paragraph. Then the running time to form the ciphertext is $O\big(n(r+1)^k\big)$.

For fixed r and variable n and k, the rank $m(G,k)$ is not polynomial in $n(r+1)^k$, by Proposition 6.3. Hence, the cracking method by general linear algebra that we described before is no longer polynomial time in the ciphertext length, or in the length of time required to form the ciphertext.

For example, suppose one uses 3-regular graphs (see §6.5) and takes $n = 100$, $k = 7$. Then a random invariant polynomial will almost certainly have $> 10^8$ nonzero monomial terms, by Proposition 6.3(a) (where we set $r_1 = 4$, $r_2 = 10$), and Charlie's linear algebra would be infeasible with this order of magnitude of unknowns. On the other hand, $n(r+1)^k \approx 10^6$, so one could feasibly encrypt messages with sparse polynomials.

Open problems. 1. Find a polynomial time algorithm to crack this version of Perfect Code cryptography, i.e., an algorithm with expected running time (for fixed r) of the form $O\big(n^\alpha \exp(\beta k)\big)$, where α and β are constants.

2. A weaker but also interesting result would be to show that the cracking problem is randomized fixed-parameter tractable (see [6] and [8]), i.e., find an algorithm with expected running time of the form $O\big(n^\alpha f(k)\big)$, where $f(k)$ is an arbitrary function of k (but α, of course, does not depend on k).

Remark. Under the assumption that it is computationally infeasible to break the Perfect Code cryptosystem without knowing a perfect code, the above encryption function can also be used for a zero-knowledge proof for Perfect Code. That is, if Alice claims to know a perfect code, Bob can verify her claim by sending a sequence of encrypted randomly chosen messages for Alice to decrypt.

[9] He need only consider those vertices u_0 for which all monomials contain some x_v for $v \in N[u_0]$.

6.8 Invariant Polynomials for Other NP-Complete Problems

It is possible to construct similar cryptosystems based on invariant polynomials associated to other NP-complete problems. For example, suppose that $G = (V, E)$ is a graph with a three-coloring $c: V \longrightarrow \{1,2,3\}$, and we are working on $\mathbf{Z}/2\mathbf{Z}$ (i.e., we want to encrypt a single bit). Consider polynomials in the variables $x_{v,i}$ for $v \in V$ and $i = 1,2,3$. The analogue of the Perfect Code building block expression $\sum_{v\in N(u)} x_v$ in §6.3 and §6.7 is the set of expressions

$$x_{v,1}+x_{v,2}+x_{v,3}, \qquad v\in V; \qquad\qquad x_{v,1}x_{v,2}+x_{v,1}x_{v,3}+x_{v,2}x_{v,3}, \qquad v\in V;$$

$$x_{u,1}x_{v_1}+x_{u,2}x_{v,2}+x_{u,3}x_{v,3}, \qquad u,v\in V, \quad u\neq v.$$

Bob knows the value of each of these expressions at Alice's three-coloring, i.e., at the subset of variables $\{x_{v,c(v)}\}_{v\in V}$, without knowing the three-coloring. Namely, the first type of expression must evaluate to 1, and the other two types of expression must evaluate to 0. Moreover, one can assign the values 0 and 1 to the variables $x_{v,i}$ in a way that is consistent with these values for the building block expressions only if one knows a three-coloring (namely, the three-coloring determined by the variables that are assigned the value 1).

7 Combinatorially Based Cryptography as a Research Project

Most public key cryptosystems are based on the presumed difficulty of certain number theoretic tasks. Among the combinatorially based systems that have been proposed, the most famous was the Merkle–Hellman Knapsack [18], which used the Subset Sum problem. At first, there was a lot of optimism about the Knapsack cryptosystem, because of its relative efficiency and also because its security seemed all but guaranteed, due to the NP-completeness of Subset Sum. However, the system was actually based on a subproblem of Subset Sum, and just a few years after the system was introduced cryptographers were stunned by the news that Shamir had succeeded in cracking the Merkle–Hellman Knapsack by solving the subproblem in polynomial time [22]. Soon after, Brickell [3] and others showed how to undermine the security of variants and generalizations of the Merkle–Hellman construction. The dramatic breaking of most knapsack-based cryptosystems in the 1980s (see [21] for a survey) seems to have caused combinatorially based cryptosystems to fall into disfavor.

An additional possible reason for skepticism about such systems is connected with a theorem of Brassard [2], which states, roughly speaking, that the cracking problem for a cryptosystem based on a one-way function cannot be NP-hard unless NP=coNP. This has been interpreted as an indication that one-way functions based on combinatorics are poor candidates for cryptography, since combinatorial problems tend to have either polynomial-time or NP-hard complexity (while such famous number theoretic problems as factoring and discrete logarithm seem to fall somewhere in the middle).

However, such an interpretation of Brassard's theorem may be premature. In several cases — such as that of the Perfect Code system in §6 — Brassard's theorem does not apply to the combinatorial one-way function upon which the cryptosystem is based (because a key condition in the theorem is not satisfied); moreover, the actual cracking problem for the system is not pure combinatorics but rather a hybrid combinatorial/algebraic problem. See [9] for more details. In short, it is too early to predict whether or not combinatorially based systems have a future in adult cryptography.

In the meantime, at the very least, such cryptosystems are a source of engaging projects in the classroom, as we have seen. In addition to its value in working with children, Kid Krypto leads to some interesting and amusing research problems, some examples of which have been given above. One challenging research project would be to develop combinatorial implementations (hopefully accessible to children) of certain cryptographic protocols which at this point are not yet part of the Kid Krypto repertoire. We have already offered a prize for a signature scheme. It would also be worthwhile to find elegant and accessible ways to present oblivious transfer, secure 2-party computation, secret sharing, zero-knowledge proofs, etc.

Thus, Kid Krypto gives us a reason to have another look at various proposals for cryptosystems based on simple combinatorics. For example, the public key system using reversible cellular automata proposed in [10] may have merit for Kid Krypto. Another combinatorially based cryptosystem was proposed by a group of researchers at Madras Christian College in India and the Hanoi Mathematical Institute in Vietnam. In [5], they show that a rewrite system — based on the word problem in a group — can be used to construct a public key system. It would be interesting to try to adapt these ideas for Kid Krypto.

It is worthwhile to develop a variety of examples of Kid Kryptosystems. In that way one can convey some of the richness and interconnectedness of mathematics, and at the same time give oneself flexibility when using Kid Krypto in the classroom.

References

[1] T. Berson, L. Guillou, and J.-J. Quisquater, How to explain zero-knowledge protocols to your children, *Advances in Cryptology — Crypto '89*, Spinger-Verlag, 1990, 628–631.

[2] G. Brassard, A note on the complexity of cryptography, *IEEE Trans. Information Theory* **IT-25** (1979), 232–233.

[3] E. F. Brickell, Breaking iterated knapsacks, *Advances in Cryptology — Crypto '84*, Springer-Verlag, 1985, 342–358.

[4] C. Crépeau and J. Kilian, Discreet solitary games, Manuscript, April 1992.

[5] Do Long Van, A. Jeyanthi, R. Siromoney, and K. G. Subramanian, Public key cryptosystems based on word problem, *ICOMIDC Symposium on the Mathematics of Computation*, Ho Chi Minh City, April 1988.

[6] R. G. Downey and M. R. Fellows, Fixed-parameter tractability and completeness I: basic results, to appear.

[7] M. R. Fellows and N. Koblitz, Kid krypto, to appear in *Advances in Cryptology — Crypto '92*, Springer-Verlag, 1993.

[8] M. R. Fellows and N. Koblitz, Fixed-parameter complexity and cryptography, to appear in *Proc. Tenth International Symposium on Applied Algebra, Algebraic Algorithms, and Error Correcting Codes*, San Juan de Puerto Rico, 1993.

[9] M. R. Fellows and N. Koblitz, On combinatorially based cryptosystems, in preparation.

[10] J. Kari, Cryptosystems based on reversible cellular automata, Manuscript, August 1992.

[11] H. Kierstead and T. Trotter, Planar graph coloring with an uncooperative partner, Manuscript, April 1992.

[12] N. Koblitz, The profit motive: the bane of mathematics education, *Humanistic Mathematics Network Journal*, No. 7 (1992), 89–92.

[13] J. Kozol, *Savage Inequalities: Children in America's Schools*, Crown Publishers, 1991.

[14] J. Kratochvíl, Perfect codes in general graphs, Monograph, Czechoslovakian National Academy of Sciences, Prague, 1991.

[15] P. Lawrence, Cargo cults, *The Encyclopedia of Religion*, Vol. 3, New York: Macmillan, 1987, 74–81.

[16] G. Madaus, study quoted in *Science News* **142** (Oct. 24, 1992), 277.

[17] Mathematical Sciences Education Board and National Research Council, *Measuring Up: Prototypes for Mathematics Assessment*, National Academy Press, 1993.

[18] R. C. Merkle and M. E. Hellman, Hiding information and signatures in trapdoor knapsacks, *IEEE Trans. Information Theory* **IT-24** (1978), 525–530.

[19] National Council of Teachers of Mathematics, *Curriculum and Evaluation Standards for School Mathematics*, 1989.

[20] National Council of Teachers of Mathematics, *Professional Standards for Teaching Mathematics*, 1991.

[21] A. Odlyzko, The rise and fall of knapsack cryptosystems, *Cryptology and Computational Number Theory*, *Proc. Symp. Appl. Math.* **42** (1990), 75–88.

[22] A. Shamir, A polynomial time algorithm for breaking the basic Merkle–Hellman cryptosystem, *IEEE Trans. Information Theory* **IT-30** (1984), 699–704.

[23] W. P. Thurston, Mathematical education, *Notices Amer. Math. Soc.* **37** (1990), 844–850.