COMBINATORIAL CRYPTOSYSTEMS GALORE! 1

Article · January 1994

DOI: 10.1090/conm/168/01688

CITATIONS
66

READS
397

2 authors:

Michael Fellows
Charles Darwin University
347 PUBLICATIONS **15,872** CITATIONS

SEE PROFILE

Neal Koblitz
University of Washington Seattle
140 PUBLICATIONS **15,218** CITATIONS

SEE PROFILE

COMBINATORIAL CRYPTOSYSTEMS GALORE! [1]

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

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

Abstract

Our paper addresses a fundamental (but naive) question in the foundations of cryptography: *Why haven't the hosts of well-known NP-hard combinatorial problems been of use in designing cryptosystems?* We offer three replies which differ radically from the conventional wisdom.

(1) *There is no good reason why NP-hard problems cannot serve as the basis for useful public-key cryptosystems.* In particular, we argue that a number of general arguments concerning this issue that are commonly found in the literature (in particular, those centering on Brassard's Theorem) are specious and circular.

(2) *There are plenty of public-key cryptosystems based on NP-hard combinatorial problems!* We describe a general method for constructing public-key cryptosystems based on virtually any kind of problem, yielding an interesting and natural class of public-key cryptosystems which we shall call CA (*combinatorially algebraic*). We show that NP can be characterized as precisely the class of problems which support public-key cryptosystems in CA. We show that there are public-key systems in CA that are *complete*, in the sense that they are *hardest to crack* for the class.

(3) *The distinction between combinatorial and algebraic problems is misleading and artificial.* Our constructions are based on ideals in polynomial algebras generated by a combinatorially derived basis, and seem to have a foot in both camps. We report a number of general theorems concerning this construction, and point to several directions that merit further investigation. In particular, we raise some issues which would appear to be crucial in any practical version of these systems.

[1] Invited address at the Second International Symposium on Finite Fields, Las Vegas, July 1993, to appear in the AMS Contemporary Mathematics Series.

1 Introduction

Following the early failure of knapsack-based public-key cryptosystems, a variety of explanations concerning this failure began to appear both in print and in the "folklore" of cryptography. The main point of much of this discussion seems to have been to argue that the many familiar NP-complete combinatorial problems (such as Knapsack) are unsuitable as a basis for cryptographic systems — that cryptography *must* be based on problems of "intermediate difficulty," such as factoring and discrete logarithms.

The following are two expressions of this "common knowledge" that have appeared in prominent surveys of the field.

> ... several other doubts were raised that applied specifically to knapsacks and other systems based on NP-complete problems. On a very abstract level there is an interesting result of Brassard [3] that says essentially that if breaking a cryptosystem is NP-hard, then NP=coNP, which would be a very surprising complexity theory result. Thus, if NP≠coNP, then breaking the Merkle-Hellman cryptosystem cannot be NP-hard, and so is likely to be easier than solving the general knapsack problem.

> There can be no hope to transform arbitrary problems in NP−P into public-key cryptosystems.

The first purpose of this paper is to refute the thesis that garden-variety hard combinatorial problems are unsuitable as a basis for cryptosystems. In particular, Brassard's Theorem provides no discernible basis for arguing against the possibility of useful combinatorially based cryptography.

Outsiders (and newcomers) to the field of cryptography are often struck by the fact that the hosts of natural NP-hard problems seem to be so barren of employment. Apart from the theoretical arguments against combinatorics-based cryptography cited above, one also encounters a relative paucity of concrete proposals for this kind of cryptosystem.

The second purpose of this paper is to describe a completely general way in which problems in NP can serve as the basis for public-key cryptosystems for which an effective cryptanalytic attack is at present unknown. These "combinatorially algebraic" (CA) cryptosystems appear to be of intrinsic mathematical interest as well as hard to crack. Somewhat tongue-in-cheek, we will describe a general *California-Style Cryptosystem* for which the first step begins as follows.

CA-style public-key cryptography

Step 1. In the privacy of your beach condominium, choose your favorite NP-complete problem and ... (to be continued).

We shall characterize NP as the class of decision problems suitable as a basis for California Cryptography. We also define a notion of *completeness* for the cracking problems in the class CA, and prove that cryptosystems may be CA-complete, i.e., hardest to crack.

The mechanics of California Crypto is concerned with ideals in polynomial rings, where the set of polynomials generating the ideal is chosen combinatorially. Because the mathematical objects and issues involved are quite standard, we believe that California Cryptography cannot be faulted for exoticism. Many of the associated complexity issues (e.g., the cracking problem) appear to be of natural mathematical interest, independently of cryptography.

This raises a third issue concerning which we hope to stimulate discussion. What exactly does "combinatorial" mean in the context of cryptography? We argue that the distinction between "combinatorial" and "algebraic" (or "number theoretic") is artificial and unproductive. In fact, our proposed cryptosystems represent an algebraicization of combinatorics that is similar in some respects to a number of productive new approaches in complexity and combinatorics (such as [1] and [15]).

We cannot yet say whether the cryptosystems we describe have potential practical merit. Some of the issues involved are discussed in §4.3. In particular, we do not know if any variants of these systems are random-self-reducible or are likely to be hard-on-average. Moreover, in their present form the systems we describe are very inefficient (requiring a polynomial amount of work to encrypt a single bit).

Our main purpose is to contribute to clarifying and widening the discussion about the kinds of problems that might serve as the basis for modern cryptosystems.

2 California Crypto

Let X represent your favorite NP-complete problem, as in the scenario (Step 1) suggested in the previous section. For concreteness in what follows, let us suppose you have chosen X to be Graph 3-Colorability. We further fix attention on the field F_2 of two elements, over which we will consider various polynomial ideals. Suppose that our message consists of a single bit (an element of the field F_2).

The following in an overview of how to implement a public-key cryptosystem based on X.

The *public key* is the graph $G = (V, E)$.

The *private key* is a proper 3-coloring of G.

The *probabilistic encryption* of the message $m \in F_2$ consists in creating a polynomial q with the properties:

(1) If one evaluates q according to any substitution t for the variables of q that corresponds canonically to a proper 3-coloring of G, one obtains $q(t) = m$.

(2) If one evaluates q according to any substitution s that does not correspond canonically to a proper 3-coloring of G, then the result $q(s)$ is randomly distributed.

The encryption method is based on a polynomial-sized set $B = \{q_i\}$ of *basis polynomials* canonically associated to G. These polynomials have the properties: (1) a substitution vector t is in the affine variety $V(B)$ if and only if t canonically corresponds to a proper 3-coloring of G; and (2) knowledge of such an F_2-point t of $V(B)$ is equivalent to knowledge of a proper 3-coloring of G. To send a message m we randomly construct a polynomial p in the ideal $J(B)$ generated by B. Then the ciphertext polynomial q is set equal to $p + m$. The construction of the polynomial p is discussed briefly in §3. For this initial discussion, we make the following assumption:

> $\star$ A random polynomial $p \in J(B)$ of size and degree specified by security parameters can be efficiently generated.

The recipient Alice, who knows a 3-coloring of G and hence an F_2-point t of $V(B)$, simply computes $q(t) = p(t) + m = m$.

We next describe the CA cryptosystems based on Graph 3-Colorability and Circuit Satisfiability in more detail, i.e., we describe the construction of $B = \{q_i\}$.

Example 1. A public-key system based on Graph 3-Colorability

Let $G = (V, E)$ be the public key. We describe a basis B of polynomials over the set of variables $T = \{t_{u,i} : \ u \in V, \ 1 \le i \le 3\}$. Let $B = B_1 \cup B_2 \cup B_3$ where

$$B_1 = \{t_{u,1} + t_{u,2} + t_{u,3} + 1 : u \in V\};$$

$$B_2 = \{t_{u,1}t_{u,2} + t_{u,2}t_{u,3} + t_{u,1}t_{u,3} : u \in V\};$$

$$B_3 = \{t_{u,1}t_{v,1} + t_{u,2}t_{v,2} + t_{u,3}t_{v,3} : uv \in E\}.$$

The vertex coloring which assigns the color i_u $(1 \le i_u \le 3)$ to the vertex u $(u \in V)$ is associated to the substitution t which sets the variable t_{u,i_u} equal to 1 and the other two of the variables $\{t_{u,1}, t_{u,2}, t_{u,3}\}$ equal to 0. It is easy to see that a substitution t is in the affine variety $V(B)$ (that is, $q(t) = 0$ for every polynomial $q \in B$) if and only if t corresponds to a proper 3-coloring of G.

Example 2. A public-key system based on Circuit Satisfiability

Let C be a boolean circuit (the public key). We assume that C consists of *and*, *or* and *not* gates, with the *and* and *or* gates having fan-in and fan-out equal to 2, and the *not* gates having fan-in and fan-out equal to 1. (We allow that some of the fan-out lines go nowhere, i.e., are ignored.) The circuit C has a single output line, corresponding to a variable z.

We describe a basis B of polynomials in a set of variables that are in one-to-one correspondence with the lines of C (including the input lines). For each gate g in the circuit we describe a small set of polynomials B_g which enforce the proper operation of the gate. Then

$$B = \left(\bigcup_g B_g \right) \cup \{z + 1\}.$$

If g is an *or* gate with input variables s, t and output variables u, v then the enforcement set B_g is

$$B_g = \{\ u + v, \quad st + uv + s + t\ \}.$$

(For an *and* gate take $B_g = \{\ u + v, \quad st + uv\}$; for a *not* gate with input variable s and output variable u take $B_g = \{s + u + 1\}$; and for an input gate with input variable s and output variables u_i take $B_g = \{s + u_i\}_i$.)

Another example of a CA cryptosystem, based on the NP-complete problem Perfect Codes in Graphs, can be found in [11]. After examining a few such examples, one sees that these cryptosystems are quite easy to develop, starting from virtually any problem. For these cryptosystems, the description of the generating set of polynomials resembles a typical direct NP-completeness reduction from the problem X to CNF Satisfiability, only couched in terms of the annihilation of polynomials rather than the satisfaction of clauses. How far can we go with this?

We use the field F_2, as before, and we let $\Sigma = \{0, 1\}$. Let $R \subseteq \Sigma^* \times \Sigma^*$. We say that R is *P-honest* if there is a polynomial q such that $(x, y) \in R$ implies $|x| \leq q(|y|)$ and $|y| \leq q(|x|)$. We say that R is *P-checkable* if there is a polynomial-time algorithm to recognize R. The *domain* of R is the set $dom(R) = \{x : \exists y \ \ (x, y) \in R\}$.

Definition. A *combinatorially algebraic* (CA) public-key cryptosystem consists of:
(1) A P-honest, P-checkable relation R.
(2) A polynomial-time algorithm $\mathcal{B}$ that on input x produces a set $B(x) = \{q_i : i \in I\}$ of polynomials in a set of variables $t_1, \ldots, t_m$, such that

$(x, y) \in R$ if and only if $|y| = m$ and $\forall q_i \in B(x)\ q_i(y) = 0$ in the field F_2, where the evaluation $q_i(y)$ is defined by setting t_i equal to the value of the i^{th} bit of y.

A *public key* for CA[R] is an $x \in dom(R)$. A *private key* for x is a y such that $(x, y) \in R$.

We have the following relationship between CA and NP.

Theorem 1. A set $X \in$ NP if and only if X is the domain of a CA public-key cryptosystem R.

Proof. In one direction the theorem is trivial; a certificate for the membership claim $x \in X$ can be a y such that $(x, y) in R$, and checking consists in generating the polynomials $B(x)$ for x and seeing that y annihilates them. In the nontrivial direction, we may assume that R has the further property that for $(x, y) \in R$ the length of y depends only on the length of x. Given x, in polynomial time using the techniques in the proof that the Circuit Value Problem is logspace complete for P, we can construct a decision circuit C_x such that $C_x(y) = 1$ if and only if $(x, y) \in R$. The basis polynomials are then generated as in Example 2 above. □

We see now that, contrary to the statement in [20] cited in §1, *every* problem X in NP has a naturally associated public-key cryptosystem (perhaps several, since we may have $X = dom(R_1) = dom(R_2)$). Note that if $X \in$ P and the relation R is natural, then the cryptosystem can be cracked in polynomial time (even under the above assumption $\star$). It is natural to ask whether there is a *hardest* public-key cryptosystem in CA. We first define what can naturally be regarded as the cracking problem under assumption $\star$ (see further discussion of this in §4.1).

Definition. For a public-key cryptosystem CA[R] ∈CA the *cracking problem* is specified as follows:
Input: The public key x, and the ciphertext polynomial q.
Promise: For some y, $(x, y) \in R$, and either $q \in J(B(x))$ or $q + 1 \in J(B(x))$.
Question: Is $q \in J((B(x))$?

In other words, the cracking problem is to find the value of q under a "valid" substitution y, i.e., one for which $(x, y) \in R$.

Definition. We say that a public-key cryptosystem CA[R] in the class of cryptosystems CA is CA-*complete* if for all CA[R'] ∈CA, the cracking problem for CA[R'] reduces in polynomial time to the cracking problem for CA[R].

Example 3. The "generic" CA-system Polly Cracker

The relation R in this case is the set of pairs (x, y) that are P-honest for some fixed polynomial, where x is a set of polynomials $\{q_i\}$, and y is a substitution vector such that $q_i(y) = 0$ for all i.

Informally speaking, the Polly Cracker cryptosystem works as follows. Alice randomly

chooses a secret fixed vector y and a set of polynomials $\{q_i\}$ which vanish on y. Her public key is $\{q_i\}$. To send her a bit m, Bob generates a sum $p = \sum g_i q_i$, and sends Alice the polynomial $p + m$.

In the *cracking problem* for this generic CA cryptosystem, the basic problem in both the *promise* and the *question* is Ideal Membership. The decision problem "$q \in J$?" and the search problem "Find the coefficients g_i of the q_i" have been studied extensively; in the special case $q = 1$ the search problem is called "effective Nullstellenzatz." See [6, 14, 7, 16]. Upper bounds for the degree of the g_i and for the field extension degree of a point in the variety $J(B)$ are exponential or superexponential. Hence, it seems likely that Ideal Membership is in neither NP nor coNP.

Theorem 2. The public-key cryptosystem CA[Polly Cracker] is CA-complete.

Proof. This is quite easy, since all other cryptosystems in CA can be viewed as special cases. □

We note that completeness in this sense does not imply (and the reductions would not be expected to respect) any hard-on-average properties that the cryptosystems may have, and thus it is not the appropriate criterion for identifying hard to crack cryptosystems. Nevertheless it would be interesting to know if other combinatorial cryptosystems are CA-complete.

3 Security of CA-Cryptosystems

We know of two ways to break such a cryptosystem: (1) by solving the underlying combinatorial problem (3-Coloring, Circuit Satisfiability, Perfect Code, or whatever); and (2) by linear algebra. The linear algebra cryptanalysis works as follows. Let T denote the set of variables in $B = \{q_i\}$, and let q be a polynomial in T of degree d over F_2 such that either q or $q+1$ belongs to $J(B)$. One attempts to find coefficient polynomials g_i of degree $d - \deg(q_i)$ such that $\sum g_i q_i = q$ or $q + 1$. One does this by regarding the coefficients of the monomials in the g_i as unknowns, and equating nonconstant monomial terms of $\sum g_i q_i$ and q.

How time-consuming is this linear algebra? If M is the cardinality of B and N is the number of variables T, then the time is clearly polynomial in $M\binom{d+N}{N}$. But the time for the linear algebra is not necessarily polynomial in the length of the input q. Namely, one can construct "sparse" polynomials q that have very few nonzero monomial terms compared with an average polynomial in T of degree d. For instance, one might construct polynomials q that have only $e^{O(d)}$, rather than $O(N^d)$, nonzero monomial terms. For more details (in the case of CA[Perfect Code]), see [12].

4 Combinatorially-Based Cryptography: Prospects Revised

By exhibiting a "galore" of public-key cryptosystems based on familiar hard problems of combinatorics, we have called into question the "folklore" that such problems are unsuitable for cryptography. At this point we should reexamine the theoretical basis for this folklore.

As mentioned before, the theoretical argument started with a theorem of Brassard [3] which stated, roughly speaking, that cryptanalysis of a system based on a one-way function cannot be NP-hard unless NP=coNP.

We note at the outset that an important distinction must be made between two types of one-way functions. One type of one-way function is the encryption function, whose inversion is the *cracking problem* in the sense of [20]. The other type of one-way function is the underlying function in the construction of the trapdoor for the system.

In combinatorial cryptosystems of the class CA, the first type of one-way function — the encryption function — is a hybrid. Although one starts by constructing an instance of a combinatorial problem, the actual cracking problem is algebraic. As explained above, it is a special case of Ideal Membership. Thus, nothing can be concluded about the cracking problem even if the underlying combinatorial problem is NP-hard.

4.1 Inapplicability of Brassard's Theorem

We next deal with the second type of one-way function — the construction used to form the trapdoor for the system. This is the type of one-way function considered in Brassard's original article [3], where RSA and discrete logarithm were used as examples. For instance, the underlying function in RSA is the multiplication map on $\mathcal{P} \times \mathcal{P}$, where $\mathcal{P}$ is the set of primes. In Brassard's theorem this one-way function must satisfy the condition that its image is in coNP. Although the coNP condition tends to hold for number-theoretic functions (e.g., the RSA and discrete log examples in [3]), below we shall see that in general the condition most likely does not hold for combinatorial one-way functions. Thus, it is circular to argue from Brassard's Theorem that number-theoretic trapdoor constructions are more suitable than combinatorial constructions.

Consider, for example, CA[3-Colorability]. Let A_n be the set of $n \times n$ adjacency matrices (i.e., symmetric matrices of 0's and 1's). Let $A'_n \subset A_n$ be the subset of matrices with the following property: there exists a partition $n = n_1 + n_2 + n_3$ such that there are only 0's in the three square blocks along the main diagonal of size $n_1 \times n_1$ (the intersection of the first n_1 rows and first n_1 columns), $n_2 \times n_2$ (the intersection of the next n_2 rows and next n_2

columns), and $n_3 \times n_3$ (the intersection of the last n_3 rows and last n_3 columns). Let S_n be the symmetric group, and set $S = \cup_n (A'_n \times S_n)$. Let $\varphi : \ S \longrightarrow \{\text{graphs}\}$ be the map which, given $(M, \sigma) \in A'_n \times S_n$, constructs the graph on vertices $u_1, \ldots, u_n$ with adjacency matrix M and then relabels the vertices according to the permutation σ. Clearly φ gives a one-way construction of 3-colorable graphs; its image consists of all 3-colorable graphs.

Theorem 4. If φ satisfies the coNP condition in Brassard's Theorem, then NP=coNP.

Proof. If the image of φ were in coNP, then NP=coNP because the problem of determining whether a graph is 3-colorable is NP-complete. □

4.2 Other Combinatorial Systems

In general, one-way constructions of trapdoors that are combinatorially based seem not to satisfy the coNP condition in Brassard's Theorem. In other words, Brassard's Theorem is not applicable to such systems. Theorem 4 above showed this in the case of a typical CA-system. We now illustrate this point in the case of two other combinatorially based cryptosystems that have been proposed (see [13, 8]).

4.2.1 Reversible cellular automata

Following [13], let $\{\mathcal{A}_i\}$ be a set of reversible cellular automata that are easy to invert. The one-way map that Kari's cryptosystem is based upon is the map from finite sequences of the $\mathcal{A}_i$ to reversible cellular automata $\mathcal{A}$ given by composition:

$$(\mathcal{A}_{i_1}, \ldots, \mathcal{A}_{i_l}) \qquad \mapsto \qquad \mathcal{A} = \mathcal{A}_{i_1} \circ \cdots \circ \mathcal{A}_{i_l}.$$

Alternately, one might use the restriction of this function to sequences of some bounded length $l \leq L$. In either case, the coNP condition in Brassard's theorem probably does not hold for this one-way function: how could one possibly have a witness that a given $\mathcal{A}$ does *not* decompose into such a product? Unlike in the RSA example in Brassard's paper — where the one-way function is multiplication restricted to the prime numbers — reversible cellular automata do not have "unique factorization." That is, in RSA the prime factorization of n into a product of > 2 primes is a witness that n is not in the image of $(p, q) \mapsto p \cdot q$; but in the case of cellular automata, even if one expresses $\mathcal{A}$ as a composite of $> L$ elementary $\mathcal{A}_i$ or expresses $\mathcal{A}$ as a composite of elementary automata that are not among the $\{\mathcal{A}_i\}$, one cannot conclude that $\mathcal{A}$ is not in the image of the above one-way map.

4.2.2 Rewrite systems

Following [8], let G be an arbitrary group given by a finite set of generators $\{c_1, \ldots, c_n\}$ and a finite set of relations $\{R_1, \ldots, R_m\}$. Without loss of generality we may assume that for each relation R_j, its inverse R_j^{-1} is also included among the $R_1, \ldots, R_m$. Let u_0 and u_1 be fixed elements of G.

Suppose that plaintext message units are sequences of N bits. By letting 0 and 1 correspond to u_0 and u_1, respectively, we may regard a plaintext message as a word of length N in the u_0, u_1 (in which u_0 and u_1 appear only to the first power). The one-way function consists of successively inserting various R_j in words written in terms of the c_i. That is, one starts with the word $u_{\epsilon_1} \cdots u_{\epsilon_N}$ (where $\epsilon_1 \cdots \epsilon_N$ is an arbitrary sequence of N bits), written as a word in the $2n$ symbols c_i and c_i^{-1} (with any adjacent $c\,c^{-1}$ canceled). For any pair of natural numbers j, k with $j \leq m$, let $\sigma_{j,k}$ denote the following operation on a word in the c_i, c_i^{-1}: insert R_j between the $(k-1)$-th symbol and the k-th symbol of the word (insert it at the end of the word if k is greater than the length of the word), and then cancel any terms of the form $c\,c^{-1}$. The one-way function is the map from $\{0,1\}^N \times \{\text{finite sequences of pairs } j,k\}$ to words in c_i, c_i^{-1} given by

$$\epsilon_1 \cdots \epsilon_N; j_1, k_1, \ldots, j_M, k_M \qquad \mapsto \qquad (\sigma_{j_1,k_1} \circ \cdots \circ \sigma_{j_M,k_M})(u_{\epsilon_1} \cdots u_{\epsilon_N}).$$

If the image of this one-way function satisfied the coNP condition in Brassard's theorem, then, in particular, there would exist a witness of nonmembership in the image of the zero-plaintext u_0^N. That is, every $g \in G$ for which the word $u_0^{-N} g$ is not equivalent to 1 in the group would have a witness that certifies this in polynomial time. Such an eventuality is highly unlikely, in view of the undecidability of the word problem in group theory (see [18]).

4.3 Questions Concerning Implementation

1. Can CA-systems be made efficient? Working over larger fields than F_2 might help. In addition, one could have a ciphertext polynomial convey more than just one field element. But at present we do not see how to make CA-systems competitive with the commonly used public-key systems.

2. What one-way constructions lead to hard instances of the NP problem X for use in CA$[X]$? A difficulty here (as with some proposed implementations of zero-knowledge protocols) is that it is not known whether invulnerable generators for hard problems exist (see [2, 10]). What is known is that if any NP-complete problem has an invulnerable generator, then so does every NP-complete problem [10].

3. For what CA-systems is the cracking problem hard-on-average? Are there

suitable versions that are random-self-reducible (see [9])? If one uses arbitrary degree d polynomials in $J(B)$, then one has random-self-reducibility. However, to avoid a polynomial time linear algebra attack, one should use sparse polynomials, as mentioned above. The sparseness property is not preserved under addition, so this destroys the random-self-reducibility.

These questions seem to be interesting targets for further investigation.

References

[1] N. Alon and M. Tarsi, Chromatic numbers and orientations of graphs, *Combinatorica*, to appear.

[2] A. Z. Broder, A. M. Frieze, and E. Shamir, Finding hidden hamiltonian cycles, *Proc. Symp. Theory of Computing* (1991), 182–189.

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

[4] E. F. Brickell, Personal conversation, 1988.

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

[6] L. Caniglia, A. Galligo, and J. Heintz, Borne simple exponentielle pour les degrés dans le théorème des zéros sur un corps de caractéristique quelconque, *C. R. Acad. Sci. Paris* **307** (1988), 255–258.

[7] J. Canny, Personal correspondence, 1993.

[8] 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.

[9] J. Feigenbaum, S. Kannan, and N. Nisan, Lower bounds on random-self-reducibility (extended abstract), *Fifth Annual Structures in Complexity Theory Conference, IEEE Comput. Soc. Press* (1990), 100–109.

[10] J. Feigenbaum, R. J. Lipton, and S. R. Mahaney, A completeness theorem for almost everywhere invulnerable generators, *Technical Memorandum, AT&T Bell Laboratories* (February 1989).

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

[12] M. R. Fellows and N. Koblitz, Combinatorially based cryptography for children (and adults), to appear.

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

[14] J. Kollár, Sharp effective Nullstellensatz, *J. Amer. Math. Soc.* **1** (1988), 963–975.

[15] C. Lund, L. Fortnow, H. Karloff and N. Nisan, Algebraic methods for interactive proof systems, *Proc. 31st IEEE Symposium on Foundations of Computer Science* (1990), 2–10.

[16] E. Mayr and A. Meyer, The complexity of the word problem for commutative semigroups and polynomial ideals, *Advances in Math.* **46** (1982), 305–329.

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

[18] P. S. Novikov, On the algorithmic unsolvability of the word problem in group theory, *Trudy Mat. Inst. im. Steklov* **44** (1955), 143 pp.

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

[20] A. L. Selman, Complexity issues in cryptography, *Computational Complexity Theory*, *Proc. Symp. Appl. Math.* **38** (1988), 92–107.

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