ON THE CRYPTOCOMPLEXITY OF KNAPSACK SYSTEMS*

Adi Shamir
Department of Mathematics
Massachusetts Institute of Technology

Abstract

A recent trend in cryptographic systems is to base their encryption/decryption functions on NP-complete problems, and in particular on the knapsack problem. To analyze the security of these systems, we need a complexity theory which is less worst-case oriented and which takes into account the extra conditions imposed on the problems to make them cryptographically useful. In this paper we consider the two classes of one-to-one and onto knapsack systems, analyze the complexity of recognizing them and of solving their instances, introduce a new complexity measure (median complexity), and show that this complexity is inversely proportional to the density of the knapsack system. The tradeoff result is based on a fast probabilistic knapsack solving algorithm which is applicable only to one-to-one systems, and it indicates that knapsack-based cryptographic systems in which one can both encrypt and sign messages are relatively insecure. We end the paper with new results about the security of some specific knapsack systems.

1. Introduction

Cryptography, which has always been considered an esoteric mixture of art and science, is rapidly gaining respectability as an important branch of complexity theory. One of the major reasons for this change is our increasing ability to prove (or at least to give strong supporting evidence) that certain computational tasks are inherently difficult. Such results may be discouraging news for engineers, but in the context of cryptosystems they can make the construction of unbreakable super-codes possible. In fact, cryptosystems may turn out to be the most important positive application of the theory of lower bounds, since they seem to be the only case in which impossibly difficult computations are desirable.

In spite of this close relationship, the tools of standard complexity theory are not very well suited to the needs of cryptography. Even when we solve the major open problems (such as the $P \neq NP$ conjecture) we cannot claim that cryptosystems based on difficult (e.g., NP-complete) problems are secure, for the following reasons:

(i) The standard measures of worst-case and average-case complexities are completely inadequate. The existence of some heuristic technique which solves a positive fraction (say, 1/1000) of the instances in polynomial time is enough to make the cryptosystem useless, even if the average case complexity of the heuristic is exponential or if it fails to work correctly on the vast majority of the cases. What we need is a theory of "almost everywhere difficult" computational tasks.

(ii) Complexity theory usually considers the difficulty of a single isolated instance of a computational task. In cryptanalysis, we are often given a big collection of related problems (e.g., many cyphertexts generated by a common cryptosystem and key) to which we can apply statistical methods, analysis of repeated patterns, etc. It is not clear how to include such factors in the complexity-theoretic analysis of a cryptosystem.

(iii) One cannot take an arbitrary difficult computational task and transform it into a cryptosystem. In order to be useful in secret-communication systems, the encoding functions must be one-to-one, and in order to be useful in certain signature

* This research was supported by the Office of Naval Research under contract No. N00014-76-C-0366.

generating systems, the encoding functions must be onto (or almost onto - see below). These extra conditions (which are not usually dealt with in complexity theory) can have a major effect on the security of the cryptosystem.

In order to handle these problems, a new theory of cryptocomplexity must be developed, with a particular emphasis on the cryptocomplexity of NP-complete problems. In this paper we consider the special case of the knapsack problem (upon which many of the newer cryptosystems are based) in order to get sharper results, but we believe that some of the ideas and results can be extended to other NP-complete problems as well. An excellent survey of the problems and achievements in this area can be found in Lempel [7].

2. Definitions

A cryptosystem is a collection of pairs consisting of an encryption function E_K (which maps cleartexts into cyphertexts) and a decryption function D_K (which maps cyphertexts back to cleartexts), such that $D_K(E_K(M)) = M$ for every cleartext M and key K (this implies that E_K must be one-to-one). In classic cryptosystems, the two communicating parties share a common pair of encryption/decryption functions which enable them to communicate over insecure channels. In public-key cryptosystems [3], each user publicly reveals his encryption function but keeps his decryption function secret. When user A wants to send a message M to user B, he can compute $E_B(M)$ quickly, but only B can decrypt it back to M. If in addition $E_K(D_K(M)) = M$ for every message M and key K (i.e., if E_K and D_K are inverse permutations over the same message space) then A can sign a message M by computing $D_A(M)$; this signature can be easily authenticated by applying the publicly known E_A to it, but it cannot be forged on other messages.

In many cryptosystems, it is difficult to make the function E_A onto, and thus not all the possible messages can be signed. The density of a cryptosystem is defined as the fraction of the signable messages among all the messages. In high-density cryptosystems this ratio is close to one, and thus messages can be signed either directly or after a slight perturbation of some unimportant bits. In low-density cryptosystems, too many trial perturbations are necessary and signatures become impractical.

Public-key cryptosystems based on NP-complete problems use the asymmetric relation between problems and their solutions. The easy encryption functions assign to each solution (= cleartext) some problem (= cyphertext) for which it is the unique solution, and the difficult decryption functions solve these problems in order to recover the original cleartext. The most popular of these cryptosystems use the knapsack problem, for which we give a precise definition below.

A knapsack system K is a finite sequence of natural numbers (generators) $a_1,\ldots,a_n$. A knapsack problem (or instance) is a knapsack system + a target value b; the problem is to determine if b has a 0-1 valued representation $c_1,\ldots,c_n$ such that $\sum_{i=1}^{n} c_i a_i = b$ (in a modular knapsack problem, this equation should hold modulo a given modulus m). The knapsack problem is known to be NP-complete both in its modular and in its non-modular versions. The cleartexts in a knapsack system K are the representations $c_1,\ldots,c_n$ while the cyphertexts are the corresponding target values b. The system is one-to-one if every target value has at most one representation in K (the possible target values in the non-modular case are all the integers in the interval $[0,\sum_{i=1}^{n} a_i]$, and in the modular case all the integers in the interval $[0,m)$).

Knapsack systems seem to be an ideal source for encryption functions, since they are numeric (and thus easy to implement electronically), fast (need n-1 additions and possibly one modular reduction), and probably uniformly hard to invert (in the sense that there are extremely few knapsack systems for which fast inversion algorithms or heuristics are known).

3. Characterization of one-to-one or onto knapsack systems

As described in the introduction, the cryptographically interesting knapsack systems are those which are one-to-one or onto. In this section we consider the problem of characterizing these two sets of knapsack systems.

From the probabilistic point of view, we have:

Theorem 1: A random modular knapsack system with n generators and modulus m is likely to be one-to-one when $n < \frac{1}{2}\log_2 m$ and non one-to-one otherwise.

Proof (Sketch): For randomly chosen modular knapsack systems, the 2^n target values corresponding to the 2^n possible representations are distributed very uniformly (with possible repetitions) in the [0,m) interval. When successive representations are enumerated (e.g., in lexicographic order on the n-bit sequences) and their corresponding target values are marked on the interval, the first repeated marking of a point is likely to occur around the $\sqrt{m}$ stage (this is a variant of the birthday problem in probability theory - see [4]). Thus if $2^n < \sqrt{m}$ a repeated marking is not likely to occur, while if $2^n > \sqrt{m}$ it is. Interpreting a repeated marking as a knapsack system which is not one-to-one and taking binary logarithms of these inequalities give the desired result. Q.E.D.

Theorem 2: A random modular knapsack system with n generators and modulus m is likely to be onto when $n > \log_2 m + \log_2\log_2 m - \log_2\log_2 e$ and non-onto otherwise.

Proof (Sketch): Using the random marking paradigm again, we would like to know how many points should be marked at random (with possible repetitions) on the [0,m) interval before all the points are likely to get marked at least once. The probability that a particular point remains unmarked after one marking stage is $1 - 1/m$, and after 2^n marking stages it is

$$P_{m,n} = (1 - 1/m)^{2^n} .$$

The expected number of unmarked points at the end of the marking process is $m\cdot P_{m,n}$, and thus the knapsack system is likely to be onto when $m\cdot P_{m,n} < 1$ and non-onto when $m\cdot P_{m,n} > 1$. To find an explicit relation between m and n, we evaluate

$$m\cdot P_{m,n} = m\cdot(1 - \frac{1}{m})^{2^n} = m\cdot\left[(1 - \frac{1}{m})^m\right]^{2^n/m} \approx m\cdot e^{-2^n/m} .$$

The turning point is when

$$m\cdot e^{-2^n/m} = 1 \quad \text{or} \quad m = e^{2^n/m}.$$

Taking repeated logarithms, we get

$$\log_2 m = (2^n/m)\log_2 e$$

and then

$$\log_2\log_2 m = n - \log_2 m + \log_2\log_2 e ,$$

which, after rearranging the terms, gives the desired result. Q.E.D.

Example: A 200 element modular knapsack is likely to be one-to-one when its modulus and generators are over 400 bits long; it is likely to be onto when its modulus and generators are less than 192 bits long.

These results show that the expected complexity of solving instances of a random knapsack system with n generators which are k bits long is at most exponential in min(n,k), and thus systems in which n and k are very different should be avoided. The reason is that if $n >> k$, we can eliminate all but $O(k+\log k)$ of the n generators and still expect to be able to represent every target value; if $k >> n$, we can consider just the 2n lowermost bits in the generators and still expect each representable target value to have only its original representation.

Although these theorems give us some insight into the average-case behavior of knapsack systems, they obviously do not enable us to decide whether a particular knapsack system is one-to-one, or onto. The difficulty of these decision problems is considered in the next group of results:

Theorem 3: Deciding whether a given knapsack system is one-to-one is co-NP complete.

Proof: The decision problem is clearly in co-NP, since there is always a short proof that a given system is not one-to-one (namely, the two representations involved).

To show completeness, we reduce the partition problem to the non-one-to-one problem. The partition problem (which is NP-complete -- see [6]) is to decide whether the equation $\sum_{i=1}^{n} c_i a_i = 0$ has a solution in which $c_i \in \{-1,+1\}$ for all i.

Lemma 4: A knapsack system $a_1,\ldots,a_n$ is not one-to-one if and only if the equation $\sum_{i=1}^{n} c_i a_i = 0$ has a non-trivial solution in which $c_i \in \{-1,0,+1\}$ for all i.

Proof: If the knapsack system is not one-to-one, then some target value b has two different representations $c_1',\ldots,c_n'$ and $c_1'',\ldots,c_n''$:

$$\sum_{i=1}^{n} c_i' a_i = b , \quad \sum_{i=1}^{n} c_i'' a_i = b$$

in which each $c_i = c_i' - c_i''$ is -1, 0 or +1, and at least one of the c_i is not 0.

Conversely, if the equation has a non-trivial solution $c_1,\ldots,c_n$, let us define

$$c_i' = \begin{cases} 1 & \text{if } c_i = 1 \\ 0 & \text{otherwise} \end{cases} \qquad c_i'' = \begin{cases} 1 & \text{if } c_i = -1 \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that $c_i = c_i' - c_i''$ for all i, and thus the equation $\sum_{i=1}^{n} c_i a_i = 0$ implies

$$\sum_{i=1}^{n} c_i' a_i = \sum_{i=1}^{n} c_i'' a_i ,$$

which gives two different representations for the common value of these terms. Q.E.D.

Proof of Theorem 3 (continued): Given a partition problem $a_1,\ldots,a_k$, we would like to construct a knapsack system $a'_1,\ldots,a'_n$ in such a way that $\sum_{i=1}^{k} c_i a_i = 0$ has a solution with $c_i \varepsilon \{-1,+1\}$ iff $\sum_{i=1}^{n} c'_i a'_i = 0$ has a non-trivial solution with $c'_i \varepsilon \{-1,0,+1\}$. The $n = 2k-1$ numbers a'_i are defined as follows:

$$a'_i = \begin{cases} 2^{3k}\cdot a_i + 2^{3i} & 1 \leq i \leq k-1 \\ 2^{3k}\cdot a_i + \sum_{i=1}^{k-1} 2^{3i} & i = k \\ 2\cdot 2^{3(i-k)} & k+1 \leq i < 2k-1 \end{cases}$$

The easiest way to understand this reduction is to consider the numbers a'_i as bit strings. The bit strings of numbers a'_i in the first group are composed of prefixes (into which the bit strings of the original a_i are left-shifted) plus 3k-bit suffixes (in which single characteristic bits are turned on). The number a'_k is similarly defined, except that all the characteristic bits of the previous $k-1$ numbers are turned on in its suffix. Finally, the numbers a'_{i+k} in the last group have empty prefixes, and their suffixes are (numerically) twice as big as those of the corresponding a'_i numbers.

To show that $a'_1,\ldots,a'_{2k-1}$ has the desired properties, assume that $\sum_{i=1} c_i a_i = 0$ has a solution with $c_i \varepsilon \{-1,+1\}$. We define the coefficients c'_i in the following way:

$$c'_i = \begin{cases} c_i & \text{if } i \leq k \\ 0 & \text{if } i > k \text{ and } c_{i-k} \neq c_k \\ -1 & \text{if } i > k \text{ and } c_{i-k} = c_k = +1 \\ +1 & \text{if } i > k \text{ and } c_{i-k} = c_k = -1 . \end{cases}$$

To show that $\sum_{i=1}^{2k-1} c'_i a'_i = 0$, we consider separately the prefix and suffix regions in this equation. The prefix parts are not empty only in $a'_1,\ldots,a'_k$. Since they are equal to $a_1,\ldots,a_k$ and the coefficients $c'_1,\ldots,c'_k$ are equal to $c_1,\ldots,c_k$, the prefix regions in the equation sum up to zero by assumption. In the suffix parts, the i^{th} group of 3 successive bits is not empty only in a'_i (where it is 001), in a'_k (where it is 001) and in a'_{i+k} (where it is 010); our choice of c'_i, c'_k and c'_{i+k} makes sure that all these 3-bit groups of bits add up to zero, and thus the suffixes as a whole add up to zero in $\sum_{i=1}^{2k-1} c'_i a'_i$.

To show the converse, let $c'_1,\ldots,c'_{2k-1}$ be any non-trivial solution of $\sum_{i=1}^{2k-1} c'_i a'_i = 0$, with $c'_i \varepsilon \{-1,0,+1\}$. Taking $c_i = c'_i$ for $i = 1,\ldots,k$ and considering the prefix parts of these equations, it is easy to see that $\sum_{i=1}^{k} c_i a_i = 0$. What remains to be done is to show that this solution is legal, i.e., that none of the c_i is 0.

Since the maximum summed value in each group of 3 successive bits in the suffix is 4 (in binary $001 + 001 + 010 = 100$), there can be no carry from one group to the next, and thus each group must sum up to zero independently. The only coefficients (c'_i, c'_k, c'_{i+k}) which can make the groups 001 in a'_i, 001 in a'_k and 010 in a'_{i+k} sum up to zero are $(0,0,0)$, $(+1,+1,-1)$, $(-1,-1,+1)$, $(+1,-1,0)$ and $(-1,+1,0)$. For different values of i, different coefficient triplets can be chosen from this set, provided that the common c'_k entries get consistent values in all the triplets. Since $c'_1,\ldots,c'_{2k-1}$ is non-trivial, there is at least one i for which $(c'_i, c'_k, c'_{k+i}) \neq (0,0,0)$, and thus $c'_k \neq 0$, which implies that $c'_i \neq 0$ for all $1 \leq i \leq k$. Q.E.D.

<u>Corollary 5</u>: Deciding whether a given modular knapsack system is one-to-one is co-NP complete.

<u>Proof</u>: When $m > \sum_{i=1}^{n} a_i$ the modulus is irrelevant, since no modular reductions can ever take place. Q.E.D.

The situation with respect to the onto property is quite different:

<u>Theorem 6</u>: Deciding whether a given knapsack system is onto is doable in polynomial time.

<u>Proof</u>: We show by induction on n that the whole interval $[0, \sum_{i=1}^{n} a_i]$ is representable in the knapsack system $a_1,\ldots,a_n$ (in which the a_i are arranged in non-decreasing order) iff $a_j \leq 1 + \sum_{i=1}^{j-1} a_i$ for all $1 \leq j \leq n$ (this condition is clearly decidable in polynomial time). It is true for any single element knapsack system since $[0,a_1]$ is representable iff $a_1 \leq 1$.

Let $a_1,\ldots,a_{n+1}$ be an n+1 element knapsack system in which $a_j \leq 1 + \sum_{i=1}^{j-1} a_i$ for all $1 \leq j \leq n$. If $a_{n+1} > 1 + \sum_{i=1}^{n} a_i = b$, then the target value b cannot be represented although it is in the interval. On the other hand, if $a_{n+1} \leq b$, then every target value in $[0, \sum_{i=1}^{n+1} a_i]$ is either in the subinterval $[0,b-1]$ (where it is representable with $c_{n+1} = 0$ by the induction hypothesis) or in the subinterval $[a_{n+1}, a_{n+1} + b - 1]$ (where it is representable with $c_{n+1} = 1$). Note that the two subintervals may overlap, and thus some target values may have representations of both forms. Q.E.D.

The complexity of the modular onto decision problem is still open. No simple characterization was found, and in fact it is not even clear whether the problem is in NP union co-NP. We conjecture that the problem is very difficult, perhaps even π_2-complete (it is in π_2 since it has the form "for all target values, there exists a representation" – see [13] for more details).

These results show that we cannot effectively characterize all the cryptographically-useful knapsack systems. All we can hope for is to characterize certain subsets of them, such as the Merkle-Hellman and the Graham-Shamir one-to-one knapsack systems described later in this paper.

An interesting open problem is whether knapsack problems remain NP-complete when restricted to knapsack systems which are one-to-one. The same question can be asked about the complexity of other similarly restricted NP-complete problems (propositional formulas with at most one satisfying model, graphs with at most one hamiltonian cycle, etc.). As far as we know, none of these problems have been shown to be either NP-complete or polynomially solvable.

4. Properties of one-to-one knapsack systems

A basic property of one-to-one knapsack systems, which will be used in the sequel, is:

Theorem 7: Let $a_1,\dots,a_n$ be a one-to-one (modular or non-modular)knapsack system, let i be an arbitrary index between 1 and n, and let b and $b+a_i$ be two target values whose representations (if they exist) are denoted by $c_1,\dots,c_n$ and $c_1',\dots,c_n'$, respectively. Then:

(i) If both b and $b+a_i$ are representable, then $c_i = 1$ if and only if $c_i' = 0$.

(ii) If b is representable but $b+a_i$ is not, then $c_i = 1$.

(iii) If $b+a_i$ is representable but b is not, then $c_i' = 0$.

Proof: (i) If $c_i = c_i' = 0$, we can add a_i to the representation of b (i.e., change c_i from 0 to 1) in order to get a second (and different) representation for $b+a_i$. If $c_i = c_i' = 1$, we can subtract a_i from the representation of $b+a_i$ (i.e., change c_i' from 1 to 0) in order to get a second (and different) representation for b. Both cases clearly contradict the assumption that the knapsack system is one-to-one.

(ii) If b had a representation in which $c_i = 0$, then by changing c_i to 1 we would get a representation of $b+a_i$, and a contradiction.

(iii) If $b+a_i$ had a representation in which $c_i' = 1$, then by changing c_i' to 0 we would get a representation of b. Q.E.D.

This theorem shows that in one-to-one knapsack systems, the sequence of c_i values in the representations of successive multiples of some fixed generator a_i ($0\cdot a_i$, $1\cdot a_i$, $2\cdot a_i$, ...) is extremely uniform. Denoting by ? the undetermined value of c_i when the multiple is unrepresentable, a typical sequence is:

010101?0101???01010101??01....

This result is particularly important in modular systems, since the set of (modular) multiples of a_i contains all the possible target values whenever a_i and m are relatively prime.

We now turn to consider some transformations which can be applied to knapsack systems without changing their one-to-one or onto character:

Lemma 8: If $a_1,\dots,a_n$ (m) is a one-to-one (onto) modular knapsack system and d is relatively prime to m, then the augmented knapsack system $da_1,\dots,da_n$ (m) is also one-to-one (onto).

Proof: Since d is relatively prime to m, it has a modular inverse, $dd^{-1} \equiv 1 \pmod{m}$, and thus

$$\sum_{i=1}^{n} c_i(da_i) \equiv b \pmod{m} \Leftrightarrow \sum_{i=1}^{n} c_i a_i \equiv bd^{-1} \pmod{m}.$$

Consequently, the number of representable target values in the two systems is the same, and multiple representations occur in one system iff they occur in the other (with target values b and bd^{-1}, respectively). Q.E.D.

Lemma 9: If $a_1,\dots,a_n$ (m) is a one-to-one (onto) modular knapsack system, then the knapsack system obtained by replacing any subset of the a_i's by their complements $m-a_i$ is also one-to-one (onto).

Proof: It is enough to show that a single complementation of a_1 to $m-a_1$ leaves the system one-to-one (onto). If b has two representations $c_1,\dots,c_n$ and $c_1',\dots,c_n'$ in the new system, then

$$b \equiv c_1(m-a_1) + \sum_{i=2}^{n} c_i a_i \equiv c_1'(m-a_1) + \sum_{i=2}^{n} c_i' a_i \pmod{m},$$

and thus

$$b + (c_1 + c_1')a_1 \equiv c_1' a_1 + \sum_{i=2}^{n} c_i a_i \equiv c_1 a_1 + \sum_{i=2}^{n} c_i' a \pmod{m},$$

i.e., c_1', $c_2,\dots,c_n$ and c_1, $c_2',\dots,c_n'$ are two different representations of a common target value in the old system. Similarly, b can be represented in the new system iff $b + a_1$ can be represented in the old system, since

$$b + a_1 = \sum_{i=1}^{n} c_i a_i \pmod m \Leftrightarrow$$

$$b = (1 - c_1)(m - a_1) + \sum_{i=2}^{n} c_i a_i \pmod m$$

Q.E.D.

We end this section with the following technical observation:

Lemma 10: If $a_1,\ldots,a_n$ (m) is a one-to-one modular knapsack system in which u target values are unrepresentable, then $m = 2^n + u$.

Proof: The 2^n possible representations generate 2^n different representable target values; all the other $m - 2^n$ possible target values are unrepresentable. Q.E.D.

5. Permutation knapsack systems

The ideal knapsack system from the cryptographic point of view is one which is both one-to-one and onto (i.e., a permutation). While each one of the two properties is believed to be very hard to check, their intersection is surprisingly easy:

Lemma 11: The knapsack system $a_1,\ldots,a_n$ defines a permutation if and only if (under some reordering) each a_i is exactly 2^{i-1}.

Proof: An easy extension of the proof of Theorem 6.

Theorem 12: The modular knapsack system $a_1,\ldots,a_n$ (m) defines a permutation if and only if $m = 2^n$ and (under some reordering) each a_i has the following form: $n - i$ arbitrary leading bits followed by 1 followed by $i - 1$ trailing zeroes.

Proof: By Lemma 10, $m = 2^n$. At least one of the generators (say, a_1) must be odd, since if all the generators (and the modulus) were even, odd target values could not be represented in the system. This odd generator is relatively prime to m, and thus multiplying all the generators by $a_1^{-1} \pmod m$ creates a new normalized knapsack system $1, a_2', \ldots, a_n'$ (m) which is also one-to-one and onto by Lemma 8.

All the multiples $0 \cdot 1$, $1 \cdot 1$, $2 \cdot 1$, $3 \cdot 1$, ... of the first generator are representable, and thus by Theorem 7 the coefficient c_1 alternates between 0 in the representations of even numbers and 1 in the representations of odd numbers. Since the generators $a_2',\ldots,a_n'$ have trivial representations of the form 0...010...0, they must all be even.

The proof can now proceed by induction. Since the generators, the modulus and the representable numbers are all even in the subsystem $a_2',\ldots,a_n'(m)$, we can divide them by 2. Applying the characterization in the theorem to the permutation system $a_2'/2,\ldots,a_n'/2$ (m/2), we know that each $a_i'/2$ ($2 \leq i \leq n$) ends with a 1 followed by $i - 2$ trailing zeroes, and thus each a_i' ends with a 1 followed by $i - 1$ trailing zeroes. Since $a_i \equiv a_1\, a_i' \pmod{2^n}$ and a_1 is odd, this also characterizes the structure of the original generators a_i.

The other direction (showing that any modular knapsack system with this structure defines a permutation) is left for the reader as an easy exercise. Q.E.D.

As noted by many researchers, the complexity of inverting permutation encryption functions cannot be any higher than $\Delta = NP \cap co\text{-}NP$, and thus it is not likely to be NP-complete. (The uniqueness of the solution enables us to give short proofs both for the question "is there a solution satisfying property P" and to its converse "do all the solutions satisfy ~P".) In the special case of knapsack-based encryption functions, we can use the characterization theorems in order to get the stronger result:

Corollary 13: All the knapsack problems generated by (modular or non-modular) permutation knapsack systems are polynomially solvable.

Proof: In the non-modular case, there is only one permutation knapsack system of each size, and it defines an identity mapping. In the modular case, there are $2^{n(n-1)/2}$ non-isomorphic permutation knapsack systems of size n (they differ in the leading bits of the generators), but their structure makes it easy to determine successive c_i's in the representation of each b ($c_1 = 0$ iff b is divisible by 2, $c_2 = 0$ iff $b - c_1 a_1$ is divisible by 4, etc.). Q.E.D.

Thus, unlike the Rivest-Shamir-Adleman factorization cryptosystems (see [11]), no single knapsack-based cryptosystem can both encode and sign arbitrary messages.

What happens when the onto condition is allowed to have a few exceptions? An illustrative result is:

Theorem 14: The modular knapsack system $a_1,\ldots,a_n$ (m) is one-to-one, and onto with a single exception b_0, if and only if $m = 2^n + 1$ and (under some reordering) each a_i is either $a_1 \cdot 2^{i-1} (\mathrm{mod}\, m)$ or $m - a_1 \cdot 2^{i-1} (\mathrm{mod}\, m)$.

Proof: We first show that all the generators a_i are relatively prime to the modulus $m = 2^n + 1$. If $(a_i, m) > 1$, then the cyclic sequence of numbers $b_o + 1 + 0 \cdot a_i$, $b_o + 1 + 1 \cdot a_i$, $b_o + 1 + 2 \cdot a_i$, ... $(\mathrm{mod}\, m)$ contains fewer than m distinct elements, and in particular it does not contain b_o. Consequently, all these numbers are representable, and by Theorem 7 the c_i in their representations alternate between 0 and 1. The length of the cycle must therefore be even, but this contradicts the fact that this length must also be a divisor of the odd modulus m.

We can now multiply the generators by a_1^{-1} in order to normalize the knapsack system to $1, a'_2, \ldots, a'_n (m)$. Since m is odd, exactly one of each $a'_i, m - a'_i$ pair is even and thus by Lemma 9 we can transform this knapsack system to $1, a''_2, \ldots, a''_n\ (m)$ in which all the generators except the first are even, and in which they are listed in non-decreasing order. If we can show that $a''_i = 2^{i-1}$ for all i, we get the desired characterization of the original knapsack system by unwinding the two normalizing transformations we applied to it.

The sequence of c_1 values in the representations of m successive multiples of the generator 1 has the following form:

0101...01?0101...01.

Consequently, the subsystem $a''_2, \ldots, a''_n$ represents all the even target values in the range $[0, b_o)$ and all the odd target values in the range (b_o, m). Since the generators $a''_2, \ldots, a''_n$ themselves are even, they are all smaller than b_o.

If there is any odd target value in $[0, m)$ which is representable in $a''_2, \ldots, a''_n\ (m)$, let b_1 be the smallest. By changing some c_i from 1 to 0 in the representation of b_1, we get a representation of the smaller odd target value $b_1 - a_i$ (modular wraparound cannot occur since $a_i < b_o < b_1$) — a contradiction. Thus $a''_2, \ldots, a''_n$ can represent only even target values in $[0, m)$ and $b_o = m - 1 = 2^n$.

If (without modular reduction) $\sum_{i=2}^{n} a''_i \geq m$, there is some j such that $b_2 = \sum_{i=2}^{j-1} a''_i < m$ and $b_3 = b_2 + a''_j \geq m$. As a sum of even generators $b_2\ (\mathrm{mod}\, m)$ is even, but $b_3\ (\mathrm{mod}\, m)$ (which is actually $b_2 + a_j - m$) is odd. This contradicts the assumption that no odd number in the interval $[0, m)$ can be represented in $a''_2, \ldots, a''_n\ (m)$.

This leaves us with a one-to-one knapsack system $a''_2, \ldots, a''_n$ in which all the even numbers are representable and in which no modular reductions can ever take place (since $\sum_{i=2}^{n} a''_i < m$). By Lemma 11, the only knapsack system having this property is $a''_i = 2^{i-1}$, $2 \leq i \leq n$. Q.E.D.

A similar (but somewhat more complicated) characterization was found for modular one-to-one knapsack systems which are onto with two exceptions. We haven't carried out this detailed analysis any further, but we conjecture that all the modular one-to-one knapsack systems with sufficiently high density are recognizable in polynomial time, and that all their associated knapsack problems are solvable in polynomial time. The characterization problem is likely to get harder and harder as the density decreases, since in the absence of the density condition it is co-NP complete.

6. A new complexity measure for cryptographic systems

As described in the introduction, both the worst-case and the average-case complexity measures are inadequate in the context of cryptography, since the security of a cryptographic system depends on the complexity of the easiest (rather than the most difficult) instances of the underlying problem. When a cryptanalyst tries to decode a batch of intercepted messages, he does not crack them in sequence, regardless of computational effort. Instead, he determines a threshold of effort beyond which individual decoding attempts are abandoned. The cryptanalysis is likely to succeed if a sufficiently high percentage of the possible messages is decodable within this time threshold. Note that the actual complexity of the unsolved cases (which can have a strong influence on the average-case complexity) does not matter in this model.

The new complexity measure, which we call median complexity, is essentially a worst case measure on the easier half of the instances. More formally, we define:

Definition: A problem P(x) (where P is a predicate or a function on instances x) has a median complexity C(n) if there exists an algorithm A for it such that for each size n, A solves at least half the instances x whose size is n within time C(n).

An obvious generalization of this definition is to replace the "half" by a fraction parameter $0 \leq \alpha \leq 1$; the resultant percentile complexity $C(n,\alpha)$ gives for each n and α the complexity of the easiest α of the instances of size n.

Example: Consider the following factorization problem: For each input x (a natural number in binary notation), we have to print out "PRIME" if x is a prime number, and some non-trivial factor if x is composite. The median complexity C(n) of this problem is O(1), since for half the numbers of each size, 2 is a non-trivial factor. The algorithm can be extremely slow on the odd inputs (say, using exhaustive search) without affecting this median complexity. What is the percentile complexity of this factoring problem?

The new complexity measure enables us to define the following important property of problems:

Definition: A problem is uniform if there is a polynomial q which bounds its worst-case complexity to $q(\frac{1}{\alpha_0})C(n,\alpha_0)$ for every fraction α_0 and $C(n,\alpha)$ algorithm that solves it.

This definition states that for each fixed fraction α_0, the worst case complexity C(n,1) is not much bigger than the complexity of the easiest α_0 of the instances, and thus a difficult uniform problem is not likely to have sizeable easy subsets of instances. An example of a problem with a uniform behavior was recently discovered by Rabin [10]. He considered the (apparently difficult) problem of taking square roots modulo a composite number m, and showed how to transform each instance of this problem into any other instance, with uniform probability distribution. By trying $O(\frac{1}{\alpha_0})$ random transformations, any $C(n,\alpha_0)$ algorithm for taking the square roots of a fraction α_0 of the numbers modulo m, can be made a probabilistic $O(\frac{1}{\alpha_0}C(n,\alpha_0))$ algorithm for taking the square roots of all the numbers modulo m.

Randomly chosen modular knapsack systems with many more generators than bits (see Theorem 2) seem to have a similar uniform behavior, although in a less rigorous sense of the word. The transformation in this case consists of subtracting a random subset of the first half of the generators from the target value b, and representing the new target value b' in terms of the second half of the generators. If each half of the generators is an onto knapsack system, then b can be transformed to any other target value b' (typically with a fairly uniform distribution), and any such b' has a representation. The coefficients of the generators in both halves give us a legal representation of the original target value b.

In the next section we demonstrate the usefulness of the new complexity measure by showing that the median complexity of solving modular knapsack problems is inversely proportional to the density of their associated systems, and thus all the very-high-density one-to-one modular knapsack systems (which we were unable to characterize explicitly) are cryptographically insecure. Both the worst-case and the average-case complexity measures seem to be inadequate for this purpose.

7. Tradeoffs between the density and security of modular knapsack systems

A simple way of motivating our result is as follows. If a complicated but smoothly-behaving function f has to be evaluated at many points, it makes sense to precompute a table of values of f at a sufficiently dense grid of points. When an actual argument b is given, we look up the values of f at the closest grid neighbors of b, and use them in order to interpolate f(b). If the function has some isolated discontinuities, this technique can be applied only to those arguments b whose grid cell does not contain a discontinuity. We thus get a triple tradeoff between the number of discontinuities, the grid size, and the fraction of the arguments b for which f(b) can be interpolated from the table. By fixing this fraction to one half, we get a relationship between the number of discontinuities and the complexity of precomputing the table (as determined by the grid size).

A similar situation exists in high-density one-to-one modular knapsack systems. Theorem 7 shows that the relationship between target values and representations in these systems is very smooth, with potential discontinuities only at the unrepresentable target values (whose exact locations are usually unknown). If b and b' are two sufficiently close locations in the 0-1-? c_i sequence (which we introduced after Theorem 7), we expect their values to be equal if $|b - b'|$ is even, and opposite if $|b - b'|$ is odd. Consequently, we can interpolate the value of c_i at b from the value of c_i at its closest grid neighbor b' whenever the two locations are in the same continuity interval between successive question marks. If these question marks are few and far between, we can use a small number of grid points in order to correctly interpolate the value of c_i at most of the locations.

We can now describe the density-security tradeoff result in detail. Given a one-to-one modular knapsack system $a_1,\dots,a_n$ (m) in which $m >> u = m - 2^n$, we would like to find a representation for a given target value b. We make the simplifying assumption that $(a_i, m) = 1$ for all i, and thus each generator can be normalized to 1. (Generators with a small gcd can be handled by similar techniques, while generators with a large gcd, if there aren't too many of them, can be handled by brute-force methods once the coefficients c_i of the other generators are determined.)

We proceed in n stages. At the j^{th} stage, we change the equation $\sum_{i=1}^{n} c_i a_i = b \pmod m$ into $\sum_{i=1}^{n} c_i(a_j^{-1} a_i) = a_j^{-1} b \pmod m$ in which the j^{th} generator $a_j^{-1} a_j$ is 1 and the target value is $a_j^{-1}b$, but in which the coefficients c_i remain unchanged. Successive multiples of the j^{th} generator become successive integers, and thus the coefficient c_j in the representation of $a_j^{-1}b$ is just the $a_j^{-1}b$-th entry in the appropriate 0-1-? sequence.

We next choose a random set of r representations, and enumerate their corresponding target values in the augmented system (the exact value of r will be determined later). Since these representations are in a one-to-one correspondence with the representable target values, we get a uniformly distributed grid of r locations along the 0-1 parts in the 0-1-? sequence, whose values are known. The desired c_j value is then interpolated in the usual way, and the tentative collection of coefficients $c_1,\dots,c_n$ calculated in the n stages is eventually verified by direct substitution.

The most time-consuming part in this process is to find the closest grid neighbor of the location we want to represent. We use a variant of Horowitz and Sahni's algorithm [5]. We divide the (modified) generators into two halves, and prepare for each half a random list of $O(\sqrt{r})$ representation/target value pairs. If x and y are target values in the first and second list, respectively, then $x + y \pmod m$ is a target value whose representation in the complete system is the concatenation of x's and y's representations in their respective half systems. Among the O(r) possible sums we can find the one that best approximates a given value z in the following way. We first sort each one of the two lists into increasing target value order, using a linear time bucket sort. Starting with $x_{min} + y_{max}$, we replace x by its list-successor whenever the sum is smaller than z, and replace y by its list-predecessor whenever the sum is bigger than z (always recording the best approximation found so far). We stop when we hit z or when one of the two lists is exhausted. In our modular case we have to repeat the process twice, with $z = a_j^{-1}b$ and $z = a_j^{-1}b + m$, but the total time complexity is still $O(\sqrt{r})$.

Our algorithm successfully finds the representation of an (original) target value b only if it is successful in all its n stages. At each stage there is a certain fraction of target values for which the interpolation gives erroneous results, but by making this fraction smaller than 1/2n, we can guarantee that no more than half of the target values will be incorrectly processed at some stage.

What remains to be done is to show that for an appropriately chosen number r of grid points, each stage can be made correct for all but a 1/2n fraction of the target values.

<u>Definition</u>: Given a 0-1-? sequence, the <u>radius</u> of location b in it is the shortest cyclic distance from b to a question mark.

<u>Lemma 15</u>: For every 0-1-? of length m with u question marks, the fraction of the locations which have radiuses smaller than $\frac{1}{4n} \cdot \frac{m}{u}$ is at most $\frac{1}{2n}$.

<u>Proof</u>: In the interval of radius $\frac{1}{4n} \cdot \frac{m}{u}$ around each question mark there can be at most $\frac{1}{2n} \frac{m}{u}$ locations, and thus the number of locations which are that close to any one of the u question marks is at most $\frac{1}{2n} m$. Q.E.D.

By choosing a random grid of, say, $r = 1000 \cdot n \cdot u$ locations along the 0-1-? sequence, we get an average grid separation of $\frac{1}{1000n} \cdot \frac{m}{u}$, and thus practically all the locations whose radiuses are bigger than $\frac{1}{4m} \cdot \frac{m}{u}$ will be closer to a grid point than to a question mark. Although some grids are better and some grids are worse in this respect, the average grid quality (which is what we are interested in when we consider probabilistic algorithms) is excellent and does not depend on the knapsack system or on the target value involved.

Using this r value, we get:

Theorem 16: The probabilistic median complexity of solving instances of a one-to-one modular knapsack system with modulus m, n generators and u unrepresentable target values is at most $O(n^{3/2}u^{1/2})$.

Proof: Each one of the n phases takes $O(\sqrt{1000 \cdot n \cdot u})$ time, and the total is thus $O(n^{3/2}u^{1/2})$. For every knapsack system with the above parameters, the algorithm succeeds (with a very high probability that does not depend on the system) in finding the representations of at least one half of the possible target values. Q.E.D.

This algorithm is considerably faster than the best known $O(2^{n/2})$ algorithm [5] whenever the knapsack system is very-high-density ($m \approx 2^n >> u$), and it indicates that knapsack systems in which one can both encrypt and sign messages may be dangerously overloaded. Although the result does not directly apply to medium-density ($m \approx 2^n \approx u$) knapsack systems, it seems best to use different types of knapsack systems for these two tasks, such as a low-density ($m \approx u >> 2^n$) Merkle-Hellman system [9] for encryption and an onto but non one-to-one system (Shamir [12]) for signature generation.

8. Some specific knapsack-based cryptographic systems and their security

In their paper, Merkle and Hellman define a particular class of one-to-one knapsack systems, which can be used in order to encrypt messages in public-key cryptosystems.

Definition: A sequence $a_1,\ldots,a_n$ of natural numbers is superincreasing if for each $1 \leq j \leq n$ $a_j > \sum_{i=1}^{j-1} a_i$.

Definition: A knapsack system $a_1,\ldots,a_n$ is a Merkle-Hellman system if there are two relatively prime numbers m and w such that $w^{-1}a_1 \pmod m$ $,\ldots,$ $w^{-1}a_n \pmod m$ is a superincreasing sequence whose sum is smaller than m.

A simple example of a superincreasing sequence is $1,2,4,8,\ldots,2^n$. Considered as a knapsack system, it is always one-to-one, and there is an easy algorithm for solving all its instances by successive subtractions. By Lemma 8, the Merkle-Hellman systems are also one-to-one and their instances can be solved by the same algorithm (if m and w are known).

The procedure Merkle and Hellman recommend for choosing a knapsack system is:

(i) Choose a superincreasing sequence $a'_1,\ldots,a'_{100}$ in which each a'_i is randomly chosen from the interval $[(2^{i-1}-1)\cdot 2^{100}+1,\ 2^{i-1}\cdot 2^{100}]$ (it is a $99+i$ bit natural number).

(ii) Choose a random m from $[2^{201}+1, 2^{202}-1]$.

(iii) Choose a random w from $[2, m-2]$, and repeat this step if $(m,w) > 1$.

(iv) Compute $a_i = w \cdot a'_i \pmod m$ for $1 \leq i \leq 100$.

(v) Publish $a_1,\ldots,a_{100}$ in the communication network directory, keeping m and w secret.

The security of this cryptosystem depends on the fact that the a_i generators are pseudo-random numbers, and only the legitimate user (who knows m and w) can decode messages in the shortcut method.

The starting point for our analysis of the security of these systems was the following challenge in Merkle and Hellman's paper:

> "Attempts to break the system can start with simplified problems (e.g., assuming m is known). If even the most favored of certificational attacks is unsuccessful, then there is a margin of safety against cleverer, wealthier, or luckier opponents. Or, if the favored attack is successful, it helps to establish where the security really must reside. For example, if knowledge of m allows solution, then an opponent's uncertainty about m must be large."

Theorem 17 (Adi Shamir, Richard Zippel): The knowledge of m makes any standard-parameter Merkle-Hellman system highly vulnerable to cryptanalysis.

Proof: The key observation is that a'_1 and a'_2 in the unknown superincreasing sequence are much smaller than m (for the recommended parameters, they are 100, 101 and 202 bits long, respectively). Since m is known, the cryptanalyst can calculate $q = a_1/a_2 \pmod m$. But $a_i = w \cdot a'_i$ and thus $q = a'_1/a'_2 \pmod m$ or $a'_1 = a'_2 \cdot q \pmod m$.

Consider now the set $S = \{k \cdot q \pmod m \mid 1 \leq k \leq 2^{101}\}$. Since $a'_2 \leq 2^{101}$, $a'_2 \cdot q \pmod m$ (Which is equal to a'_1) is in this set. The 2^{101} modular multiples in S are very evenly distributed in $[0, m-1]$, and thus the smallest number among them is likely to be around $m/2^{101} \approx 2^{202}/2^{101} = 2^{101}$. But a'_1 is known to be smaller than or equal to 2^{100}, and thus a'_1 itself is likely to be the smallest number in S.

Consequently, all we need in order to find (a candidate for) a'_1 is to find the minimum value of $k \cdot q \pmod m$ when k ranges over $[1, 2^{101}]$ and q,m are known. Efficient methods for solving this number-theoretic problem (using the continued fraction approximation of the ratio q/m) can be found in [2] and [8]. Q.E.D.

It is easy to see that for other choices of the parameters, this cryptanalytic attack has a good chance of success only when $a'_1 \cdot a'_2$ is not much bigger than m, which suggests one way in which the security of the Merkle-Hellman systems can be enhanced. Another way was discovered (independently) by Ron Graham and Adi Shamir. The danger represented by the extremely small first few elements of the superincreasing sequence can be eliminated by adding some high-order noise bits in the following way:

$a'_i = (\text{random bits})(0^{n-i}10^{i-1})(0..0)(\text{random bits})$.

The block of zeroes is $\log_2 n$ bits wide, and its purpose is to "absorb" the addition carries generated by the low order random bits.

Sums of subsets of these numbers are decodable with no arithmetic operations at all (the characteristic 1 bits in the a'_i keep track of the summands). To hide the structure of these numbers, we multiply them by $w \pmod m$, and publish the resultant generators $a_1, \ldots, a_n$ (keeping w and m secret). This variant of Merkle and Hellman's scheme resists the cryptanalytic attack described in Theorem 17, and it thus seems to be safer, faster and simpler to implement than the original variant.

Another way of using knapsack systems for cryptographic purposes was first suggested to the author by Benjamin Arazi [1]. To encrypt a (numeric) message M in such a system, the sender computes the sum R of a random subset of the published generators, and sends M + R over the communication channel. The huge number of possible R values makes it impossible for an eavesdropper to compute M from M + R without knowing the secret structure embedded in the published knapsack system.

We propose using a modified Graham-Shamir system, in which the characteristic 1's in the a'_i numbers are changed to 0's; the resultant $\bar{a}'_i$ numbers are structurally indistinguishable, and the generators $\bar{a}_i$ in the published system are defined in the usual way as $\bar{a}_i = w \cdot \bar{a}'_i \pmod m$. The number of generators k in this scheme need not be related to the message size n.

Upon receiving an encrypted message $M + \sum_{i=1}^{k} r_i \bar{a}_i$ (where the r_i are unknown random bits), the user calculates $w^{-1}[M + \sum_{i=1}^{k} r_i \bar{a}_i] \pmod m$ which is equal to $w^{-1}M + \sum_{i=1}^{k} r_i \bar{a}'_i \pmod m$. Due to the special structure of the $\bar{a}'_i$ numbers, the sum $\sum_{i=1}^{k} r_i \bar{a}'_i$ has in its middle a "transparent window" of n successive zero bits, which enables n successive bits of $w^{-1}M$ to be "seen through" in $w^{-1}M + \sum_{i=1}^{k} r_i \bar{a}'_i$. By restricting w^{-1} to have the appropriate size, these visible bits can be made the high-order bits of $w^{-1}M$, from which the n bits of M can be extracted by a single division-with-remainder operation.

We (informally) claim that this random addition scheme is at least as secure as the Graham-Shamir scheme. The knapsack systems used in the random addition scheme seem to be tougher, since they are not necessarily one-to-one, and even their publisher need not know any easy way for solving their instances (in fact, none may exist). Furthermore, given a Graham-Shamir knapsack system $a_1, \ldots, a_n$, it is easy to see that the numbers $\bar{a}_i = 2a_i - a_{i+1}$ all have the "transparent window" structure, and thus $\bar{a}_1, \ldots, \bar{a}_{n-1}$ can be used in a random addition scheme. This transformation makes any cryptanalytic attack on the published generators in random addition systems a cryptanalytic attack on Graham-Shamir systems (but not necessarily vice-versa).

An important advantage of random addition systems is that the random sums $\sum_{i=1}^{k} r_i \bar{a}_i$ do not depend on the messages, and thus can be prepared in advance and added to M in a single step. This enables an extremely fast response time to encryption requests, which is important in many real-time applications.

Acknowledgments: I would like to thank Len Adleman, Abraham Lempel, Gary Miller, Michael Rabin and Ron Rivest for many fruitful discussions.

Bibliography

[1] B. Arazi, private communication.

[2] J. Cassels, "An Introduction to Diophantine Approximation", Cambridge University Press, 1965.

[3] W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Trans. Information Theory, November 1976.

[4] W. Feller, "An Introduction to Probability Theory and Its Applications", John Wiley & Sons, 1968.

[5] E. Horowitz and S. Sahni, "Computing Partitions with Applications to the Knapsack Problem", JACM, April 1974.

[6] R. Karp, "Reducibility Among Combinatorial Problems", in "Complexity of Computer Computations" (ed. R. Miller and J. Thatcher), Plenum Press, 1972.

[7] A. Lempel, "Cryptology in Transition: A Survey", to appear in Computing Surveys.

[8] W. LeVeque, "Fundamentals of Number Theory", Addison-Wesley, 1977.

[9] R. Merkle and M. Hellman, "Hiding Information and Receipts in Trap Door Knapsacks", IEEE Trans. Information Theory, September 1978.

[10] M. Rabin, "Digitalized Signatures and Public Key Functions As Intractable as Factorization", to appear in CACM.

[11] R. Rivest, A. Shamir and L. Adleman, "A Method For Obtaining Digital Signatures and Public-Key Cryptosystems", CACM, February 1978.

[12] A. Shamir, "A Fast Signature Scheme", MIT/LCS/TM-107, July 1978.

[13] L. Stockmeyer, "The Polynomial-Time Hierarchy", TCS, October 1976.