# PERFORMANCE ANALYSIS OF SHAMIR'S ATTACK ON THE BASIC MERKLE-HELLMAN KNAPSACK CRYPTOSYSTEM

(Extended Abstract)

J. C. Lagarias

AT&T Bell Laboratories

Murray Hill, New Jersey

**0. Abstract**

This paper gives a performance analysis of one variant of Shamir's attack on the basic Merkle-Hellman knapsack cryptosystem, which we call Algorithm S. Let $R = \dfrac{\#\ \text{plain text bits}}{\text{maximum}\ \#\ \text{cipher text bits}}$ denote the rate at which a knapsack cryptosystem transmits information, and let $n$ denote the number of items in a knapsack, i.e. the block size of plaintext. We show that for any *fixed* $R$ Algorithm $S$ runs to completion in time polynomial in $n$ on all knapsacks with rate $R_o \geq R$. We show that it successfully breaks at least the fraction $1 - \dfrac{c_R}{n}$ of such knapsack cryptosystems as $n \to \infty$, where $c_R$ is a constant depending on $R$.

## 1. Introduction

In 1978 Merkle and Hellman [11] proposed public key cryptosystems based on the knapsack problem. The simplest of these cryptosystems, the *basic knapsack cryptosystem*, works as follows. The *public information* is a set of nonnegative integers $\{a_i : 1 \leq i \leq n\}$ which are called *knapsack weights*. Messages are encrypted by first being broken into blocks $(x_1, \ldots, x_n)$ of $n$ binary digits. A block is encrypted as the integer $E$ given by

$$a_1 x_1 + \ldots + a_n x_n = E; \quad \text{all } x_i = 0 \text{ or } 1 . \tag{1.1}$$

The problem of solving (1.1) for $(x_1, \ldots, x_n)$ when given *arbitrary* $\{a_1, \ldots, a_n, E\}$ is known to be NP-hard. However in the basic Merkle-Hellman cryptosystem the knapsack items $\{a_i : 1 \leq i \leq n\}$ have a special structure which allows (1.1) to be solved easily; this structure is concealed by a *trapdoor*. The trapdoor information is any *decryption pair* $(W,M)$ of integers satisfying the following conditions:

(i) $1 \le W < M$ and $(W,M) = 1$.

(ii) $M > \underset{i}{MAX} \{a_i\}$.

(iii) The sequence $\{s_i : 1 \le i \le n\}$ defined by $0 \le s_i < M$ and

$$s_i \equiv Wa_i \ (mod\ M)$$

is *superincreasing*, i.e.

$$s_1 + \ldots + s_i < s_{i+1} \text{ for } 1 \le i \le n-1 .$$

(iv) The *size condition.* $s_1 + \ldots + s_n < M$ holds.

Given a decryption pair $(W,M)$ a ciphertext $E$ is decrypted by finding

$$E^* \equiv WE \ (mod\ M) ; \quad 0 \le E^* < M \tag{1.2}$$

and solving the $0-1$ integer programming problem

$$s_1 x_1 + \ldots + s_n x_n = E^* ; \quad \text{all } x_i = 0 \text{ or } 1 . \tag{1.3}$$

Under the conditions (i)–(ii) equations (1.1) and (1.3) have the same solution when $E$ and $E^*$ are related by (1.2). The equation (1.3) is easily solved in linear time using the superincreasing property of the $\{s_i\}$.

In order to produce knapsacks $\{a_i : 1 \le i \le s_n\}$ having such a trapdoor, Merkle and Hellman proceed as follows. Given an *expansion factor* $d \ge 1$ and a *block size n*, they pick an integer $M$ with

$$2^{dn} < M < 2^{dn+1} . \tag{1.4}$$

Next they pick a ''random'' superincreasing sequence $\{s_i, \ldots, s_n\}$ such that $s_1 + \ldots + s_n < M$. Finally they draw $W^*$ from $1 \le W^* < M$ with $(W^*, M) = 1$ using the uniform distribution and set

$$a_i \equiv W^* s_i \ (mod\ M) ; \quad 1 \le a_i < M .$$

It is easy to verify that (i)–(iv) hold in this case, provided $W$ is determined by $1 \le W < M$ and by

$$WW^* \equiv 1 \ (mod\ M)\ .$$

After the knapsack items $\{a_i : 1 \le i \le n\}$ are produced, Merkle and Hellman scramble their order using a permutation $\sigma \in S_n$, so that the *public keys* are $\{a_{\sigma(i)} : 1 \le i \le n\}$. In this case the permutation σ is also part of the trapdoor information.

The interpretation of the *expansion factor* $d$ is that it is a measure of how much longer ciphertext messages are than plaintext messages, i.e.

$$d \cong \frac{\log_2(nM)}{n} \cong \frac{\text{maximum}\ \#\ bits\ in\ ciphertext\ E}{\#\ bits\ in\ plaintext\ block}\ .$$

The inverse quantity $R = d^{-1}$ is a measure of the average number of bits of plaintext transmitted per bit of ciphertext, i.e. $R$ is the *information rate*.

Merkle and Hellman's hope was that it would not be easily possible to recover the trapdoor information. However in 1981 Adi Shamir [13] discovered a strong attack on the basic Merkle-Hellman cryptosystem. He showed that his attack runs in time polynomial in $n$ as $n \to \infty$ when the modulus $M \le 2^{dn}$ and $d$ is *fixed*. (This running time is however, at least exponential in $d$.) He analyzed the performance of the key step in his attack (Step 2 following), assuming an unproved but plausible assumption (Hypothesis $U$ in Section 3B) and showed that this step succeeded with high probability provided $1 < d < 2$. As we indicate in Section 3B and [6], this key step depends on a rational vector constructed from the public keys having an ''unusually good'' simultaneous Diophantine approximation. The unproved assumption is that the ''unusually good'' simultaneous Diophantine approximation arising from superincreasing sequences behave similarly to ''random'' rational vectors having an ''unusually good'' simultaneous Diophantine approximation. Shamir also presented heuristic arguments that his attack works in general for $1 \le d < \infty$. (The condition $d \ge 1$ is required in order that encrypted messages be uniquely decipherable.)

The object of this paper is to outline a performance analysis of Shamir's attack that considers all steps in his attack, assumes no unproved hypothesis, and which applies to all expansion rates $d$ with $1 \le d < \infty$. In particular it asserts that a version of Hypothesis $U$ is true. The result contrasts with a similar heuristic put forward by Adleman [1] in connection with an attack on iterated knapsack cryptosystems, which does not seem to hold on numerical examples (c.f. [3]).

Some of the methods here can be applied to the analysis of other knapsack cryptosystems, c.f. [2], [4], [6], [8], [12].

## 2. Shamir's Attack on Basic Knapsack Cryptosystems

The object of Shamir's attack is to find a *decryption pair* $(W^*, M^*)$, which need not be the same as the pair $(W, M)$ used by the encrypter. This is possible because any basic knapsack cryptosystem has *infinitely many* decryption pairs: any pair with $\frac{W^*}{M^*}$ sufficiently close to $\frac{W}{M}$ will work.

Shamir's attack proceeds in several steps, which we sketch here.

*Algorithm S.*

*Step 1.* Estimate the modulus $M$ by $\tilde{M} = \max_{1 \le i \le n} \{a_i\}$ and estimate the expansion factor $d$ by $d^* = \frac{1}{n} \log_2 (n^2 \tilde{M})$.

*Step 2.* Set $g = d^* + 2$ or $g = 5$, whichever is larger. Guess the correct $g$ knapsack items $(a_1, \ldots, a_g)$ corresponding to the $g$ *smallest* superincreasing elements. (That is, run the following algorithm on all $\begin{bmatrix} n \\ g \end{bmatrix}$ possible $g$-tuples.) Solve the integer program (I.P.)

$$|x_i a_1 - x_1 a_i| \le B \;; \quad 2 \le i \le g \;, \tag{2.2}$$

$$1 \le x_1 \le B - 1 \;, \tag{2.3}$$

where

$$B = [2^{-n+g} M^*] \;. \tag{2.4}$$

If a solution $(x_1^{(0)}, \ldots, x_1^{(0)})$ is found, create two new integer programs by replacing (2.3) by the constraints

$$1 \le x_1 < x_1^{(0)}$$

and

$$x_1^{(0)} < x_1 \le B - 1 \;,$$

respectively. Solve these two I.P.'s and for each solution found, create two new I.P.'s by continuing to subdivide the $x_1$ regions according to the values of $x_1^{(i)}$ found. Do this until either $n \log_2 n$ distinct solutions are found, with at most $2n \log_2 n$ I.P.'s examined, or else until the process halts before this with a set of I.P.'s with no further solutions.

*Step 3.* For each solution $(x_1^{(i)},\ldots,x_n^{(i)})$ found in Step 2, examine the $n^7$ rationals

$$\theta_j^{(i)} = \frac{x_1^{(i)}}{a_i} + j \frac{1}{n^7 2^n \tilde{M}} ; \quad 1 \le j \le n^7 . \tag{2.5}$$

Find $\theta_j^{(i)} = \dfrac{W_j^*}{M_j^*}$ in lowest terms using the Euclidean algorithm. Check if $(W_j^*, M_j^*)$ is a decryption pair for $\{a_1,\ldots,a_n\}$. If so, the algorithm *succeeds*. If not, continue.

□

The rationale for this procedure is as follows. The decryption congruence

$$W\, a_i \equiv s_i \ (mod\ M) \tag{2.6}$$

is equivalent to the equality

$$Wa_i - M\, k_i = s_i \tag{2.7}$$

for some non negative integer $k_i$. Then (2.7) gives:

$$\frac{W}{M} - \frac{k_i}{a_i} = \frac{s_i}{Ma_i} . \tag{2.8}$$

Hence we obtain

$$\frac{k_i}{a_i} - \frac{k_1}{a_1} = \frac{s_1}{Ma_1} - \frac{s_i}{Ma_i} , \tag{2.9}$$

so that

$$k_i a_1 - k_1 a_i = \frac{1}{M} (s_1 a_i - s_i a_1) . \tag{2.10}$$

Since any superincreasing sequence $\{s_i\}$ with $\sum_{i=1}^{n} s_i < M$ has

$$0 \le s_i \le 2^{-n+i} M , \tag{2.11}$$

the bounds (2.10), (2.11) yield for $1 \le i \le g$ that

$$|k_i a_1 - k_1 a_i| \le 2^{-n+g} \tilde{M} . \tag{2.12}$$

So in this case the integer program (2.2) and (2.3) has at least one solution $(x_1,\ldots,x_g) = (k_1,\ldots,k_g)$. It turns out that for $g \ge d^* + 2$, a ‘‘random’’ integer program of the form (2.2), (2.3) can be expected to have no integer feasible solution; in this sense our particular I.P. is ‘‘unusual’’ in having a solution. The condition $d^* \geq 5$ is a technical one.

Now suppose Step 2 succeeds in finding a solution $(x_1^{(i)}, \ldots, x_n^{(i)})$ with $x_1^{(i)} = k_1$. Then (2.8) and (2.11) give

$$0 \leq \frac{W}{M} - \frac{k_1}{a_1} \leq \frac{1}{2^{n+1} a_1} \ . \qquad (2.13)$$

Also suppose that

$$\frac{1}{n^2} M \leq a_1 \leq M \ . \qquad (2.14)$$

Then (2.13) becomes

$$0 \leq \frac{W}{M} - \frac{k_1}{a_1} \leq \frac{n^2}{2^n M} \ . \qquad (2.15)$$

In this case Step 3 is bound to find a pair $\frac{W_j^*}{M_j^*}$ with

$$\left| \frac{W_j^*}{M_j^*} - \frac{W}{M} \right| \leq \frac{1}{n^5 2^n M} \ . \qquad (2.16)$$

Set $(W^*, M^*)$ equal to this $(W_j^*, M_j^*)$. Then if $\lambda$ is defined by $M^* = \lambda M$, we have

$$W^* = \lambda (W + \varepsilon)$$

where (2.16) gives

$$|\varepsilon| \leq n^{-5} 2^{-n} \ . \qquad (2.17)$$

Hence

$$\begin{aligned} W^* a_i - M^* k_i &= \lambda \left[ (W a_i - M k_i) + \varepsilon a_i \right] \\ &= \lambda (s_i + \varepsilon a_i) \ , \end{aligned}$$

where

$$|\varepsilon a_i| \leq n^{-5} 2^{-n} M \ . \qquad (2.18)$$

It turns out that

$$s_i^* = s_i + \varepsilon a_i \tag{2.19}$$

will be a superincreasing sequence for ‘‘almost all’’ superincreasing sequences (Corollary 3.6) and hence $(W^*, M^*)$ will then be the desired decryption pair.

This rationale indicates that Algorithm $S$ can only fail in the following ways:

(1) The bound $\frac{1}{n^2} M \le a_1 \le M$ can fail to hold.

(2) Step 2 may fail to find a solution $(x_1^{(i)}, \ldots, x^{(1)})$ with $x_1^{(i)} = k_1$.

(3) Step 3 may fail for all $j$ because all sequences $\{s_i^*\}$ given by (2.19) aren’t superincreasing.

We analyze these possibilities in Section 3.

Before proceeding we bound the running time of Algorithm $S$. The integer programs encountered in Step 2 all have $g$ variables, which we regard as fixed while the number of items $n \to \infty$. Such I.P.’s can be solved in polynomial time in the input length $L$ using an algorithm of H. W. Lenstra, Jr. [10]. The running time bound of Lenstra’s algorithm has (apparently) the form $O(L^{F(g)})$ where $F(g)$ grows exponentially in $g$. Kannan ([5], Theorem 1) has announced a faster algorithm, which runs in time $O(g^{9g} L \log L)$. Using Kannan’s algorithm, it is easy to obtain the following result.

*Lemma 2.1. Algorithm S runs to completion in time* $0(n^{g+10} L \log L)$ *where* $L = g \log \tilde{M}$.

Note here than when the information rate $R_0 \le R$ then the expansion factor $d^* \le d \le R^{-1}$ is bounded above. Hence $g = d^* + 2$ is fixed and Lemma 2.1 gives a bound for the running time which is polynomial in the input length $L$.

## 3. Performance Analysis

We assume the following probablistic model. The modulus $M$ is *fixed*. The multiplier $W$ is drawn uniformly from the set of all $W$ with $1 \le W \le M$ with $(W, M) = 1$. The superincreasing sequence $\{s_i, \ldots, s_n\}$ drawn uniformly from all superincreasing sequences with $\sum_{i=1}^{n} s_i < M$. We *define* $d$ by $M = 2^{dn}$. Let $G(n, M)$ denote the number of choices for $(W;\ s_1, \ldots, s_n)$. Our main result is:

*Theorem 3.1. For any fixed information rate R with* $O < R < 1$ *there is a constant* $c_R$ *such that for fixed M Algorithm S fails on at most* $(1 - \frac{c_R}{n})\ G(n,M)$ *choices* $(W;\ s_1,\ldots,s_n)$*, provided* $M = 2^{dn}$ *with* $1 \le d \le R^{-1}$.

We now indicate the main steps in the proof, corresponding to the three types of failure mentioned at the end of Section 2.

### A. Bounding the knapsack item $\mathbf{a_1}$

In this step we consider the smallest element $s_1$ of the superincreasing sequence as fixed. Now $Wa_1 \equiv s_1\ (mod\ M)$ so that

$$W^* s_1 \equiv a_1\ (mod\ M)\ . \tag{3.1}$$

where $WW^* \equiv 1\ (mod\ M)$. There are $\phi(M)$ choices for $W^*$ with $(W^*,M) = 1$ and we want to show that at most $O\left[\frac{1}{n}\ \phi(M)\right]$ such choices give $|a_1| \le \frac{1}{n^2}\ M$, provided $M \ge 2^n$. We use:

*Lemma 3.2. Let* $B(M,T) = |\ \{x : (x,M) = 1 \text{ and } 1 \le x \le T\}|$
*Then*

$$B(M,T) \le \frac{T}{M}\ \phi(M) + O\left[M^{\frac{c_o}{\log\ \log\ M}}\right] .$$

This suffices if $(s_1,M) = 1$ and in fact gives $O\left[\frac{1}{n}\ \phi(M)\right]$. There are some complications if $(s_1,M) > 1$. We may assume $(s_1,M) \le n$ since the fraction of superincreasing sequences with $(s_1,M) > n$ is $O\left[\frac{1}{n}\right]$ of the total, as can be inferred using Theorem 3.5 below.

### B. Bounding failure in Step 2

This is done in two stages. First, we show that for a fixed $K$ the *expected* number of solutions of a ‘‘random’’ integer program of the form (2.2), (2.3) having a solution $(x_1^{(0)},\ldots,x_g^{(0)})$ with $x_1^{(0)} = K$ is bounded above by a constant depending on $g$ (but not on $n$). Second, we show that the set of all sequences $(W,s_1,\ldots,s_n)$ mapping onto a *fixed* $a_1$ have images $(a_2,\ldots,a_g)$ hitting at least a positive fraction (depending on $g$) of all ‘‘random’’ I.P.’s of the form (2.2), (2.3). Consequently the expected number of solutions to such ‘‘special’’ integer programs is bounded above by a (larger) constant depending on $g$.

Shamir's analysis avoided this second stage by assuming:

*Hypothesis U. The integer programs of the form* (2.2), (2.3) *arising from basic knapsack cryptosystems have, up to a multiplicative constant depending on g, the same expected number of solutions* as a *''random'' integer program* (2.2), (2.3) *having at least one solution*.

The first stage is handled by:

*Theorem 3.3.* (a) *Let* $a_1$ *and* $K$ *be fixed. Let* $a_1 = \lambda M$ *with* $\frac{1}{n^2} \le \lambda \le 1$. *The number* $(a_2, \ldots, a_g)$ *of integer programs*

$$|a_i x_1 - a_1 x_i| \le \lambda 2^{-n+g} a_1 \ ; \quad 2 \le i \le g \ ,$$

*with* $0 \le a_i < a$, *having at least one solution* $(x_1^{(0)}, \ldots, x_g^{(0)})$ *with* $x_1^{(0)} = K$ *is bounded above by a constant depending on g times* $\lambda^g 2^{(g-1)dn}$.

(b) *The expected number of solutions to such an integer program is bounded above by a constant depending on g but not on n, provided* $g \ge d + 2$ *and* $g \ge 5$.

This is proved by reformulating it as the equivalent $(g-1)$-dimensional simultaneous Diophantine approximation problem

$$\left| \frac{a_i}{a_1} - \frac{x_i}{x_1} \right| \le \frac{2^{-n+g}}{x_1} \ , \ 2 \le i \le g$$

and applying the results of [7]. Here (a) uses an easy counting argument and (b) is difficult to prove.

The second stage is supplied by the following lemma.

*Lemma 3.4. Let* $a_1$ *be fixed, with* $a_1 = \lambda M$ *for* $\frac{1}{n^2} \le \lambda \le 1$. *The number of distinct* $(a_2, \ldots, a_g)$ *(mod* $a_1$*) arising as the image of some* $(W, s_1, \ldots, s_g)$ *with* $Wa_1 \equiv s_1$ *(mod M) is either zero or at least a constant depending on g times* $\lambda^g 2^{(g-1)dn}$.

To prove this we hold $W$ and $s_1$ fixed and vary $s_2, \ldots, s_g$, using the bounds on the number of superincreasing sequences given in Theorem 3.5 below. We also need the result that the number of ways of extending a given superincreasing sequence $(s_1,\ldots,s_g)$ with $s_i < 2^{-n+i}M$ to a superincreasing sequence $(s_1,\ldots,s_n)$ with $\sum_{i-1}^{n} s_i \le M$ is roughly a constant for ‘‘almost all’’ $(s_1,\ldots,s_g)$, c.f. Corollary 3.6.

**C. Bounding failure in Step 3**

Here we use good estimates for the number of superincreasing sequences with various restrictions on their elements $(s_1,\ldots,s_n)$. Let $S_n(M)$ denote the number of superincreasing sequences with $\sum_{i=1}^{n} s_i < M$. If we set

$$d_i = s_i - (s_1+\ldots+s_{i-1}) \ ; \quad 1 \le i \le n \ .$$

then

$$s_1+\ldots+s_n = 2^{n-1}d_1 + 2^{n-2}d_2+\ldots+2d_{n-1} + d_n$$

we find $S_n(M)$ is the number of integer solutions to

$$d_i \ ; \quad 1 \le i \le n \ , \tag{3.1}$$

$$M > 2^{n-1}d_1+\ldots+2d_{n-1} + d_n \ .$$

The system (3.1) cuts out a simplex $\Omega_n(M)$ in $\mathbb{R}^n$ with volume $\frac{2^{-\binom{n}{2}}}{n!} M^n$. Consequently this is about the number of integer points we expect to satisfy (3.1) for large enough $M$.

*Theorem 3.5.* $S_n(M) = \frac{2^{-\binom{n}{2}}}{n!} M^n + O\left[\frac{2^{-\binom{n}{2}}}{n!} M^n \left(\frac{2^n n^4}{M}\right)\right]$

In order to make Step 3 successful, we want relatively large perturbations $\varepsilon\, a_i$ in (2.19) to not ruin the superincreasing property. This is equivalent to requiring that all the $d_i$ in (3.1) be ‘‘large.’’ Now the integer program

$$d_i > B \ ; \quad 1 \le i \le n \ ,$$

$$M \ge 2^{n-1}d_1+\ldots+2d_{n-1} + d_n \ .$$

has the same number of integer feasible solutions as

$$d_i^* > 0 \ ; \quad 1 \le i \le n \ ,$$

$$M - (2^n - 1) B > 2^{n-1} d_1^* + \ldots + 2 d_{n-1}^* + d_n^* \ ,$$

which has $S_n(M - (2^n - 1) B)$ solutions. Hence as long as

$$S_n(M - 2^n B) \sim S_n(M) \tag{3.2}$$

as $M = 2^{dn}$ with $n \to \infty$, we have ''almost all'' superincreasing sequences have all $d_i \ge B$. The following Corollary of Theorem 3.5 shows that we may choose $B = n^{-2} 2^{-n} M$.

*Corollary 3.6. If $M = 2^{dn}$ with $d > 1$ then as $n \to \infty$ at most $O\left(\frac{1}{n} S_n(M)\right)$ superincreasing sequences with $\sum_{i-1}^{n} s_i < M$ and $d_i = s_i - (s_1 + \ldots + s_{i=1})$ have some*

$$d_i < n^{-2} 2^{-n} M \ . \tag{3.3}$$

Now the bounds (2.19) and (2.17) give

$$|\varepsilon\, a_i| \le n^{-5} 2^{-n} M \tag{3.4}$$

Comparing this with (3.3) we easily check that ''almost all'' superincreasing sequences remain superincreasing when perturbations of size $n^{-5} 2^{-n} M$ are allowed.

## References


[1] L. Adleman, On Breaking Generalized Knapsack Cryptosystems, Proc. 15th Annual ACM Symposium on Theory of Computing, 1983, pp. 402-412.

[2] E. Brickell, Solving Low Density Knapsacks, in: *Advances in Cryptology*, *Proceedings of Crypto-83* (D. Chaum, Ed.), Plenum Publ. Co., New York 1984.

[3] E. Brickell, J. C. Lagarias and A. M. Odlyzko, Evaluation of Adleman's Attack on Multiply Iterated Knapsacks (Abstract), *Advances in Cryptology Proceeding of Crypto-83* (D. Chaum, Ed.), Plenum Publ. Co., New York 1984.

[4] Y. Desmedt, J. Vandewalle, R. Govaerts, A Critical Analysis of the Security of Knapsack Public Key Cryptosystems, preprint.

[5] R. Kannan, Improved Algorithms for Integer Programming and Related Lattice Problems, Proc. 15th Annual ACM Symposium on theory of Computing, 1983, pp. 193-206.

[6] J. C. Lagarias, Knapsack Public Key Cryptosystems and Diophantine Approximation (Extend Abstract), *Advances in Cryptology, Proceedings of Crypto-83* (D. Chaum, Ed.), Plenum Publ. Co., New York, 1984, pp. 3-24.

[7] J. C. Lagarias, Simultaneous Diophantine Approximation of Rationals by Rationals, preprint.

[8] J. C. Lagarias and A. M. Odlyzko, Solving Low Density Subset Sum Problems, Proc. 24th IEEE Symposium on Foundations of Computer Science, 1983, pp. 1-10.

[9] A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovasz, Factoring polynomials with rational coefficients, Math. Annalen. 261 (1982), pp. 515-534.

[10] H. W. Lenstra, Jr., Integer programming with a fixed number of variables, Math. of Operations Research, to appear.

[11] R. Merkle and M. Hellman, Hiding Information and Signatures in Trapdoor Knapsacks, IEEE Trans. Information Theory IT-24 (1978), pp. 525-530.

[12] A. M. Odlyzko, Cryptanalytic attacks on the multiplicative knapsack cryptosystem and on Shamir’s fast signature scheme, IEEE Trans. Information Theory, to appear.

[13] A. Shamir, A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem, Proc. 23rd Annual Symposium on Foundations of Computer Science, 1982, pp. 145-152.