Advances in Cryptology – Proceedings of CRYPTO '99
(August 15–19, 1999, Santa Barbara, California, USA)
M. Wiener (Ed.), vol. ???? of Lecture Notes in Computer Science, pages ???–???

The Hardness of the Hidden Subset Sum Problem and its Cryptographic Implications

Phong Nguyen and Jacques Stern

École Normale Supérieure
Laboratoire d'Informatique
45 rue d'Ulm, 75230 Paris Cedex 05
France
{Phong.Nguyen,Jacques.Stern}@ens.fr
http://www.dmi.ens.fr/~{pnguyen,stern}/

Abstract. At Eurocrypt'98, Boyko, Peinado and Venkatesan presented simple and very fast methods for generating randomly distributed pairs of the form $(x, g^x \mod p)$ using precomputation. The security of these methods relied on the potential hardness of a new problem, the so-called hidden subset sum problem. Surprisingly, apart from exhaustive search, no algorithm to solve this problem was known. In this paper, we exhibit a security criterion for the hidden subset sum problem, and discuss its implications on the practicability of the precomputation schemes. Our results are twofold. On the one hand, we present an efficient lattice-based attack which is expected to succeed if and only if the parameters satisfy a particular condition that we make explicit. Experiments have validated the theoretical analysis, and show the limitations of the precomputation methods. For instance, any realistic smart-card implementation of Schnorr's identification scheme using these precomputations methods is either vulnerable to the attack, or less efficient than with traditional precomputation methods. On the other hand, we show that, when another condition is satisfied, the pseudo-random generator based on the hidden subset sum problem is strong in some precise sense which includes attacks *via* lattice reduction. Namely, using the discrete Fourier transform, we prove that the distribution of the generator's output is indistinguishable from the uniform distribution. The two conditions complement each other quite well, and therefore form a convincing picture of the security level.

1 Introduction

In many discrete-log-based protocols, one needs to generate pairs of the form $(x, g^x \mod p)$ where x is random and g is a fixed base. ElGamal [9] and DSS [13] signatures, Schnorr's [18, 19] and Brickell-McCurley's [4] schemes for identification and signature are examples of such protocols. The generation of these pairs is often the most expensive operation, which makes it tempting to reduce the number of modular multiplications required per generation, especially for smartcards. There are basically two ways to solve this problem. One way is to generate separately a random x, and then compute $g^x \mod p$ using a precomputation method [3, 7, 12]. The other way is to generate x and $g^x \mod p$ together by a special pseudo-random number generator which uses precomputations. Schnorr was the first to propose such a preprocessing scheme [18]. The scheme had much better performances than all other methods but there was a drawback: the ouptut exponent x was no more guaranteed to be random, and therefore, each generation might leak information. Indeed, de Rooij [6] showed how to break the scheme. Schnorr later proposed a modified version [19], which was also broken by de Rooij [8].

At Eurocrypt'98, Boyko, Peinado and Venkatesan proposed new and very simple generators [2] to produce pairs of the form $(x, g^x \mod p)$, which could reduce even further the number of necessary modular multiplications. The security of these methods apparently depended on a new problem, the so-called hidden subset sum problem: given a positive integer M and $b_1, \ldots, b_m \in \mathbb{Z}_M$, find $\alpha_1, \ldots, \alpha_n \in \mathbb{Z}_M$ such that each b_i is some subset sum modulo M of $\alpha_1, \ldots, \alpha_n$. The problem borrows its name from the classical subset sum problem: given a positive integer M and $b, \alpha_1, \ldots, \alpha_n \in \mathbb{Z}_M$, find $S \subset \{1, \ldots, n\}$ such that $b \equiv \sum_{j \in S} \alpha_j \pmod{M}$. The most powerful known attack [5] against the subset sum problem reduces it to a shortest vector problem in a lattice built from $b, \alpha_1, \ldots, \alpha_n, M$. Provided a shortest vector oracle, the method succeeds with high probability if the density, defined as $d = n/\log_2 M$, is small, namely less than a constant approximately equal to 0.94. However, this method can hardly be applied to hidden subset sums: one cannot even build the lattice since the α_j's are hidden. Actually, apart from exhaustive search, no algorithm was known to solve the hidden subset sum problem. And thus, according to the authors of [2], the problem was potentially harder than the subset sum problem. Still, they suggested high values of parameters to prevent any subset sum attack, for unknown reasons. For these choices of parameters, the scheme was not suited for smartcards, and the speed-up over other methods was questionable.

It was therefore natural to ask whether or not, one could select small parameters in order to make the scheme very efficient, without affecting the security. More generally, Boyko *et al.* raised the following question: how hard is the hidden subset sum problem ? The present paper provides an answer. We exhibit a security criterion for the hidden subset sum problem which is twofold. On the one hand, we present an efficient lattice-based algorithm to solve the hidden subset sum problem. It relies on a systematic use of the powerful notion of an orthogonal lattice, which was introduced at Crypto'97 [14] by Nguyen and Stern as a cryptographic tool, and subsequently used in cryptanalysis [16, 15]. The algorithm is very different from known lattice-based methods to solve subset sums, but surprisingly, seems to generalize their results. More precisely, our algorithm is expected to succeed when the density $d = n/\log_2 M$ is very small. Unfortunately, this is exactly the case arising when one wants to make the scheme practical and more efficient than other exponentiation methods, in a smart-card environment. We have implemented the algorithm, and experiments have confirmed our analysis. On the other hand, we show that when the density is high, the pseudo-random generator based on the hidden subset sum problem is strong in some precise sense. Namely, using the discrete Fourier transform, we prove that the distribution of the generator's output is then statistically close to the uniform distribution. Such a result was already known (related results in [1, 10]), but our proof technique is different. Those results tend to prove that the hardness of the hidden subset sum problem is measured by the density, as for the subset sum problem.

The remainder of the paper is organized as follows. In section 2, we describe the generators of pairs $(x, g^x \mod p)$ proposed at Eurocrypt'98 in [2], and we clarify the relationships between the security of these schemes and the hidden subset sum problem. In section 3, we recall some facts on orthogonal lattices from [14]. Section 4 presents our lattice-based algorithm to solve hidden subset sum problems, and the experiments. In section 5, we discuss the hardness of the hidden subset problem, by measuring the randomness of the generator output.

2 Fast Exponentiation With Hidden Subset Sums

Let p be a prime number, and $g \in \mathbb{Z}_p^*$ of order M. In [2], several generators producing pairs $(x, g^x \mod p)$ were proposed. The simplest generator was the following one:

Preprocessing Step: Generate n random integers $\alpha_1, \ldots, \alpha_n \in \mathbb{Z}_M$. Compute $\beta_j = g_j^{\alpha}$ for each j and store both α_j's and β_j's in a table.

Pair Generation: Whenever a pair (x, g^x) is needed, randomly generate $S \subseteq \{1, \ldots, n\}$ such that $|S| = \kappa$. Compute $b = \sum_{j \in S} \alpha_j \mod M$. If $b = 0$, stop and start again. Compute $B = \prod_{j \in S} \beta_j \mod p$ and return (b, B).

Clearly, for any output (b, B), we have $B = g^b \mod p$. The other generators are just variants of the previous generator, using random walks. We will not discuss those, since the security of the generators relies on the same problem.

2.1 Parameters

The scheme needs to store n elements of $\mathbb{Z}_M$, and n elements of $\mathbb{Z}_p^*$. Recall that for DSS [13] and Schnorr [18, 19], M has 160 bits, while for ElGamal [9] and Brickell-McCurley [4], M has at least 512 bits. Each generation requires κ modular multiplications. When $\kappa \ll n/2$, we say that the underlying hidden subset sum problem is sparse. The parameters n and κ must be sufficiently large to prevent from birthday attacks. In [2], it was suggested to choose $n = 512$ and $\kappa = 64$. Comparisons with traditional precomputation methods were made, but only in the case of 512-bit exponents. Table 1 compares the scheme with several configurations of the simple exponentiation method with precomputation of [12]. It shows that for a 160-bit exponent, the generator with the proposed parameters is worse in all aspects. For a 512-bit exponent, it is better: with similar storage, one gains 14 multiplications. But with other precomputation methods, there is no security issue since the exponent is random. Another issue is the viability of the scheme for low-computing-power devices. For instance,

Table 1. A comparison of methods for generating pairs $(x, g^x \mod p)$ where p is a 512-bit prime. Storage requirements are in 512-bit numbers. Times are in multiplications per exponentiation.

Method	160-bit exponent		512-bit exponent	
	Storage	Time	Storage	Time
Hidden subset sum generator	672	64	1024	64
Lim and Lee [12]	30	58	62	153
	62	46	157	106
	508	27	1020	78

a storage of 672 represents 42 Kbytes, which is unacceptable for a smartcard. Thus, the parameters proposed in [2] are rather suited for server applications. In order to offer much better performances than other methods, one is tempted to decrease the parameters. We will discuss possible parameters when we present the experiments related to our attack.

2.2 Security Against Active Attacks

When the generator is used, the security seems to rely on the underlying hidden subset sum problem. Indeed, suppose for instance that the generator is used in Schnorr's [19] identification scheme. Let q be a 160-bit prime dividing $p-1$, where p is a 512-bit prime.

The prover has a secret key $s \in \mathbb{Z}_q^*$ and a public key $v = g^{-s} \mod p$, where g has order $q \geq 2^t$. He generates a random pair $(k, g^k \mod p)$ and sends $x = g^k$ to the verifier. The verifier returns a challenge $e \in \mathbb{Z}_t$. Then the prover sends $y = k + es \mod q$. Finally, the verifier checks whether $x = g^y v^e \mod p$. In an active attack, the verifier can issue many times the challenge $0 \in \mathbb{Z}_{2^t}$. He thus gets many outputs of the generator, as $y = k$. After solving the underlying hidden subset sum problem, he knows the hidden $\alpha_1, \ldots, \alpha_n$. He then issues the challenge $1 \in \mathbb{Z}_{2^t}$, to obtain $k + s \mod q$ for some unknown k a subset sum of the α_j's. If n and κ are not too large, he can exhaustively search for the $0,1$-coefficients of the α_j's to disclose k, and hence the secret key s.

Conversely, if the output of the hidden subset sum generator used is cryptographically pseudo-random, then the speeded-up versions of the following schemes are secure against polynomial time adaptive attacks, provided that the original schemes are secure: ElGamal, DSS and Schnorr signatures, Schnorr identification. (see [2]).

2.3 Security Against Passive Attacks

In [2] (Theorems 6 and 7, p.230), it was claimed that only the security against active attacks needed to assume the hardness of the hidden subset sum problem. However, it seems that the security against passive attacks actually relies on the potential hardness of a slight variant of the hidden subset sum problem, which we call the *affine hidden subset sum problem*: given a positive integer M, and $b_1, \ldots, b_m, c_1, \ldots, c_m \in \mathbb{Z}_M$, find integers $s, \alpha_1, \ldots, \alpha_n \in \mathbb{Z}_M$ such that each $b_i + sc_i$ is some subset sum modulo M of $\alpha_1, \ldots, \alpha_n$.

Assume for instance that the generator is used in Schnorr's signature scheme. We keep the notations of the previous section. The public key is $v = g^{-s} \mod p$. The signer generates a random pair $(k, g^k \mod p)$. He computes a hash $e = h(g^k \mod p, m)$ where m is the message, and $y = k + es \mod q$. The signature is the pair (y, e). Notice that $k = y - es \mod q$ is a hidden subset sum, where y and e are known and s is secret. Thus, a passive attacker is left with an affine hidden subset sum problem with the pairs $(y, -e)$ and the modulus q. If he can solve this problem, he recovers the secret key s.

The previous remark can be adapted to the following schemes: Schnorr's and Brickell-McCurley's identification, ElGamal and DSS signatures. For example, in the case of DSS, a signature is of the form (a, b) where $b = k^{-1}(m + as) \mod q$, s is the secret key and m is the hash. Note that $k = mb^{-1} + ab^{-1}s \mod q$ is a hidden subset sum. But mb^{-1} and ab^{-1} are known, so this is again an affine hidden subset sum problem, from which one can derive the secret key s.

We will see that our attack against the hidden subset sum problem can be adapted to the affine hidden subset sum problem. It appears that the complexity of these problems is similar.

3 Lattice Reduction and the Orthogonal Lattice

Throughout the paper, we call lattice any subgroup of $\mathbb{Z}^m$ for some integer m. If L is a lattice, we denote by $\det(L)$ its determinant (or volume), and $\Lambda(L)$ the Euclidean norm of a shortest non-zero vector of L. A classical result of Minkowski states that for any integer d, there is a constant $\gamma(d)$ such that for all d-dimensional lattice L:

$$\Lambda(L) \leq \gamma(d) \det(L)^{1/d}.$$

The smallest such constant is denoted by γ_d and called Hermite's constant of rank d. It is known that for sufficiently large d:

$$\sqrt{\frac{d}{2\pi e}} \leq \gamma_d \leq \sqrt{\frac{d}{\pi e}}.$$

As a result, it is convenient to assume that for a "random" d-dimensional lattice L, the quantity $\Lambda(L)/(\sqrt{d}\det(L)^{1/d})$ is roughly equal to some universal constant γ. The goal of lattice reduction is to find a *reduced basis*, that is, a basis consisting of reasonably short vectors. In the sequel, we will not need more precise definitions, or very precise approximations for the shortest vector. In practice, one hopes to obtain sufficiently reduced bases thanks to reduced bases in the sense of LLL [11], or its variants [17, 20].

Let L be a lattice in $\mathbb{Z}^m$. The orthogonal lattice $L^\perp$ is defined as the set of elements in $\mathbb{Z}^m$ which are orthogonal to all the lattice points of L, with respect to the usual dot product. We define the lattice $\bar{L} = (L^{\perp})^{\perp}$, which is the intersection of $\mathbb{Z}^m$ with the $\mathbb{Q}$-vector space generated by L: it contains L and its determinant divides the one of L. The result of [14] which are of interest to us is the following one:

Theorem 1. *If L is a lattice in $\mathbb{Z}^m$, then* $\dim(L) + \dim(L^{\perp}) = m$ *and* $\det(L^{\perp})$ *is equal to* $\det(\bar{L})$.

This suggests that if L is a "random" low-dimensional lattice in $\mathbb{Z}^m$, a reduced basis of $L^{\perp}$ will consist of very short vectors compared to a reduced basis of L. More precisely, one expects that any reduced basis of $L^{\perp}$ will consist of vectors with norm around $\gamma\sqrt{m - \dim L}\,\det(\bar{L})^{1/(m-\dim L)}$. Furthermore, one can note that computing a basis of the orthogonal lattice amounts to compute the integer kernel of an (integer) matrix, so that:

Theorem 2. *There exists an algorithm which, given as input a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice L in $\mathbb{Z}^m$, outputs a basis of the orthogonal lattice $L^{\perp}$, and whose running time is polynomial with respect to m, d and any upper bound of the bit-length of the $\|\mathbf{b}_j\|$'s.*

In fact, it was proved in [14] that one could directly obtain an LLL-reduced basis of the orthogonal lattice by a suitable LLL-reduction, in polynomial time.

4 A Lattice-based Attack

Let us first restate the hidden subset sum problem in terms of vectors. Given an integer M, and a vector $\mathbf{b} = (b_1, \ldots, b_m) \in \mathbb{Z}^m$ with entries in $[0..M-1]$, find integers $\alpha_1, \ldots, \alpha_n \in [0..M-1]$ such that there exist vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}^m$ with entries in $\{0, 1\}$ satisfying:

$$\mathbf{b} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n \pmod{M} \tag{1}$$

Throughout this section, we will assume that $(\mathbf{b}, M)$ is a correct input. That is, there exist integers $\alpha_1, \ldots, \alpha_n \in [0..M-1]$, vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}^m$ with entries in $\{0, 1\}$, and a vector $\mathbf{k} \in \mathbb{Z}^m$ such that:

$$\mathbf{b} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n + M\mathbf{k} \tag{2}$$

Our attack proceeds in three steps:

1. From $\mathbf{b}$, we determine the lattice $\bar{L}_{\mathbf{x}}$, where $L_{\mathbf{x}}$ is the lattice generated by the $\mathbf{x}_j$'s and $\mathbf{k}$.
2. From $\bar{L}_{\mathbf{x}}$, we derive the hidden coefficients $\mathbf{x}_j$'s.
3. Using $\mathbf{b}$, the $\mathbf{x}_j$'s and M, we finally recover the weights α_j's.

Note that this attack recovers all secret data, not just the α_j's. For the sake of simplicity, we will assume that $L_{\mathbf{x}}$ has dimension $n+1$, but the attack still applies when the dimension is less than $n+1$. In other words, we assume that the $\mathbf{x}_j$'s and $\mathbf{k}$ are linearly independent, which is a reasonable assumption since the $\mathbf{x}_j$'s are random. We now detail the three steps.

4.1 Disclosing the Hidden Lattice

The first step is based on the following observation, which is a simple consequence of (2):

Lemma 3. *Let* $\mathbf{u}$ *in* $\mathbb{Z}^m$ *be orthogonal to* $\mathbf{b}$*. Then* $\mathbf{p_u} = (\mathbf{u}.\mathbf{x}_1, \ldots, \mathbf{u}.\mathbf{x}_n, \mathbf{u}.\mathbf{k})$ *is orthogonal to the vector* $\mathbf{v}_\alpha = (\alpha_1, \ldots, \alpha_n, M)$*.*

Note that $\mathbf{v}_\alpha$ is independent of m, and so is the n-dimensional lattice $\mathbf{v}_\alpha^\perp$. We will see that, as m grows, most of the vectors of any reduced basis of the $(m-1)$-dimensional lattice $\mathbf{b}^\perp$ are shorter and shorter. For such vectors u, the corresponding vectors $\mathbf{p}_u$ are also shorter and shorter. But if $\mathbf{p}_u$ gets smaller than $\Lambda(\mathbf{v}_\alpha^\perp)$ (which is independent of m), then it is actually zero, that is, $\mathbf{u}$ is orthogonal to all the $\mathbf{x}_j$'s and $\mathbf{k}$. This leads to the following condition:

Condition 4. *Let* $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{m-1})$ *be a reduced basis of* $\mathbf{b}^\perp$*. Then the first* $m-(n+1)$ *vectors* $\mathbf{u}_1, \ldots, \mathbf{u}_{m-(n+1)}$ *are orthogonal to each* $\mathbf{x}_j$ *and* $\mathbf{k}$*.*

One cannot expect that more than $m-(n+1)$ vectors are orthogonal, because $\bar{L}_x$ has dimension $(n+1)$. If the condition is satisfied, the $(n+1)$-dimensional lattice $(\mathbf{u}_1, \ldots, \mathbf{u}_{m-(n+1)})^\perp$ contains each of the $\mathbf{x}_j$'s and $\mathbf{k}$. And one can see that it is in fact the lattice $\bar{L}_\mathbf{x}$, because they are orthogonal lattices of equal dimension, with one containing the other. Hence, the first step is as follows:

1. Compute a reduced basis $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{m-1})$ of the orthogonal lattice $\mathbf{b}^\perp$.
2. Compute a basis of the orthogonal lattice $(\mathbf{u}_1, \ldots, \mathbf{u}_{m-(n+1)})^\perp$ to obtain $\bar{L}_\mathbf{x}$.

This step is correct if and only if condition 4 is satisfied. We now precise in which case the condition is expected to hold. We first estimate the quantity $\Lambda(\mathbf{v}_\alpha^\perp)$. If the α_j's are uniformly distributed in $[0..M-1]$, then $E(\alpha_j^2) \approx M^2/3$ so that $\|\mathbf{v}_\alpha\|$ is roughly $M\sqrt{n/3}$ (we assume the variance is negligible). With overwhelming probability, the gcd of all the α_j's and M is equal to 1, implying that the lattice $\bar{\mathbf{v}}_\alpha$ is exactly $\mathbf{v}_\alpha$, and therefore: $\det(\mathbf{v}_\alpha^\perp) = \|\mathbf{v}_\alpha\| \approx M\sqrt{n/3}$. Since the α_i's are random, the n-dimensional lattice $\mathbf{v}_\alpha^\perp$ may be considered as random, so that:

$$\Lambda(\mathbf{v}^\perp) \approx \gamma\sqrt{n}\det(\mathbf{v}_\alpha^\perp)^{1/n} \approx \gamma M^{1/n}(n/3)^{1/(2n)}\sqrt{n}.$$

We then estimate $\|\mathbf{p}_u\|$ for some well-chosen vectors $\mathbf{u}$. If the coordinates of the $\mathbf{x}_j$'s are independently uniformly distributed in $\{0,1\}$ (the case of the actual sparse distribution is discussed in section 4.4), and so are the α_j's in $[0..M-1]$, the expectation of the square of each coordinate of $\alpha_1\mathbf{x}_1 + \cdots + \alpha_n\mathbf{x}_n$ is roughly:

$$n \times \frac{1}{2} \times \frac{M^2}{3} + (n^2-n) \times \frac{1}{4} \times \frac{M^2}{4} \approx \frac{1}{16}n^2M^2.$$

Hence $E(\|\mathbf{k}\|^2) \approx mn^2/16$, and we note that $E(\|\mathbf{x}_j\|^2) \approx m/2$. It follows that for any $\mathbf{u}$ (we again assume that the variance is negligible):

$$\|\mathbf{p_u}\| \approx \|\mathbf{u}\|\sqrt{n \times m/2 + mn^2/16} \approx n\sqrt{m}\|\mathbf{u}\|/4.$$

Besides, we observe that the lattice $\mathbf{b}^{\perp}$ contains a high-dimensional lattice of small determinant. Namely, it contains by (2) the $(m-n-1)$-dimensional lattice $(\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{k})^{\perp}$, which determinant is less than $\|\mathbf{k}\| \times \prod_{j=1}^{n} \|\mathbf{x}_j\| \approx n\sqrt{m}(m/2)^{n/2}/4$. Hence, the vectors of any reduced basis of $(\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{k})^{\perp}$ are expected to have norm around

$$\gamma\left[(m/2)^{n/2} n\sqrt{m}/4\right]^{1/(m-n-1)}\sqrt{m-n-1}.$$

Note that the expression is much smaller than $\gamma\|\mathbf{b}\|^{1/(m-1)}\sqrt{m-1}$ for large M, as $\|\mathbf{b}\| \approx M\sqrt{n}$. Therefore, the first $m-n-1$ vectors of any reduced basis of $\mathbf{b}^{\perp}$ are likely to be short lattice points of $(\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{k})^{\perp}$, and their expected length is given by the previous expression. For these vectors, the approximate length of the corresponding $\mathbf{p}_{\mathbf{u}}$ is:

$$\left[n\sqrt{m}/4 \times (m/2)^{n/2}\right]^{1/(m-n-1)} n\sqrt{m(m-n-1)}/4.$$

And condition 4 is likely to be satisfied if and only if this length is significantly smaller than $\Lambda(\mathbf{v}^{\perp})$, that is:

$$\left[n\sqrt{m}/4 \times (m/2)^{n/2}\right]^{1/(m-n-1)} n\sqrt{m(m-n-1)}/4 \ll M^{1/n}(n/3)^{1/(2n)}\sqrt{n}.$$

For sufficiently large m and n, the condition simplifies to:

$$\sqrt{mn(m-n-1)}/4 \ll M^{1/n} \tag{3}$$

The left-hand part is not large. In other words, this step is expected to succeed if the density $n/\log_2(M)$ is very small, so that $M^{1/n}$ is large.

4.2 Disclosing the Hidden Coefficients

In the second step, the lattice $\bar{L}_{\mathbf{x}}$ is known. The vectors $\mathbf{x}_j$'s are random and have entries in $\{0,1\}$, therefore these are short lattice points in $\bar{L}_{\mathbf{x}}$. Consider a short vector of some reduced basis of $\bar{L}_{\mathbf{x}}$. If its entries are all in $\{0,1\}$ or $\{0,-1\}$, it is very likely to be one of the $\pm\mathbf{x}_j$'s. Otherwise, its entries are probably in $\{0,\pm1\}$, as it must be shorter than the $\mathbf{x}_j$'s. To get rid of these vectors, we transform the lattice $\bar{L}_{\mathbf{x}}$: we double all the lattice points, and we add the vector $(1,1,\ldots,1)$. The new lattice is:

$$L'_{\mathbf{x}} = 2\bar{L}_{\mathbf{x}} + \mathbb{Z} \times (1,1,\ldots,1).$$

The vectors $2\mathbf{x}_i - (1,1,\ldots,1)$ belong to $L'_{\mathbf{x}}$, and their entries are ±1: they are short lattice points. We expect that there are no shorter vectors, since there is no obvious short combination of $(1,1,\ldots,1)$ with the previous parasite vectors when doubled. In other words, the vectors $\pm[2\mathbf{x}_i - (1,1,\ldots,1)]$ should appear in any reduced basis of the lattice $L'_{\mathbf{x}}$. We expect this step to succeed if our lattice reduction algorithm provides a sufficiently reduced basis.

4.3 Recovering the Hidden Weights

Now that $\mathbf{k}$ and the $\mathbf{x}_j$'s are known, equation (2) reads as a modular linear system:

$$\mathbf{b} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n \pmod{M}$$

The only unknowns are the α_j's. If m is sufficiently large, this system is likely to have a unique solution. One way to solve this system is to use an orthogonal lattice. Denote by $x_{i,j}$ the j-th coordinate of $\mathbf{x}_i$. Also denote by b_i the i-th coordinate of $\mathbf{b}$. Let $m' \leq m$. Consider the lattice L generated by the rows of the following matrix:

$$\begin{pmatrix} b_1 & x_{1,1} & x_{2,1} & \dots & x_{n,1} & M & 0 & \dots & 0 \\ b_2 & x_{1,2} & x_{2,2} & \dots & x_{n,2} & 0 & M & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \vdots & \vdots & \ddots & \ddots & 0 \\ b_{m'} & x_{1,m'} & x_{2,m'} & \dots & x_{n,m'} & 0 & \dots & 0 & M \end{pmatrix}$$

We note that $L^{\perp}$ must contain a point of the form $(-1, \alpha_1, \dots, \alpha_n, ?, \dots, ?)$, since the α_j's satisfy the system. It follows that in any basis of $L^{\perp}$, there exists a linear combination (of the basis elements) for which the first coordinate is equal to -1. Such a combination can easily be computed from an extended gcd algorithm applied to the list formed by the first coordinate of each basis element. The element obtained is of the form $(-1, \beta_1, \dots, \beta_n, ?, \dots, ?)$. Clearly, the vector $(\beta_1, \dots, \beta_n)$ modulo M satisfies the first m' solutions of the system. If m' is sufficiently large, it must be the unique solution $(\alpha_1, \dots, \alpha_n)$. Hence, in order to solve the system, it suffices to compute a basis of the orthogonal lattice $L^{\perp}$, which can be done in polynomial time.

4.4 Sparse Hidden Subset Sums

If the hidden subset sum is sparse, that is $\kappa \ll n/2$, the condition (3) gets slightly better. Indeed, when one picks at most κ weights in each subset sum, one can show that $E(\|\mathbf{k}\|^2) \approx m\kappa^2/16$ and $E(\|\mathbf{x}_j\|^2) \approx m\kappa/n$. It follows, after a few computations, that the attack is expected to succeed if:

$$\left[\kappa\sqrt{m}/4 \times \sqrt{m}(m\kappa/n)^{n/2}\right]^{1/(m-n-1)} \kappa\sqrt{m(m-n-1)}/4 \ll M^{1/n}(n/3)^{1/(2n)}\sqrt{n}.$$

For sufficiently large m and n, the condition simplifies to:

$$\kappa\sqrt{m(m-n-1)}/4 \ll M^{1/n}\sqrt{n} \tag{4}$$

4.5 Affine Hidden Subset Sums

In the case of affine hidden subset sums, equation (2) becomes:

$$\mathbf{b} + s\mathbf{c} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n + M\mathbf{k} \tag{5}$$

Only $\mathbf{b}$, $\mathbf{c}$ and M are known. The attack can be adapted as follows. Clearly, lemma 3 remains correct if we take for $\mathbf{u}$ a vector orthogonal to $\mathbf{b}$ and $\mathbf{c}$. Step 1 thus becomes:

1. Compute a reduced basis $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{m-2})$ of the orthogonal lattice $(\mathbf{b}, \mathbf{c})^{\perp}$.
2. Compute a basis of the orthogonal lattice $(\mathbf{u}_1, \ldots, \mathbf{u}_{m-(n+1)})^{\perp}$ to obtain $\bar{L}_{\mathbf{x}}$.

The difference with the hidden subset problem is that, this time, the vector $\mathbf{k}$ can be much bigger, due to the presence of s. More precisely, we have $s \approx M/2$ and $\|\mathbf{c}\| \approx M\sqrt{m/3}$, so that $\|\mathbf{k}\| \approx M\sqrt{m/12}$. In the appendix, we discuss how to modify the previous arguments to explain why the condition is still expected to be satisfied. Loosely speaking, when $\mathbf{u}$ is short, the vector $\mathbf{p_u}$ cannot be guaranteed to be short, but all its entries except the last one are short, which suggests it cannot be a non-zero vector of $\mathbf{v}_{\alpha}^{\perp}$. Step 2 remains unchanged. And in step 3, we solve a similar linear system which is induced by (5). Therefore, the only difference when attacking affine hidden subset sums is that the underlying condition is less likely to be satisfied.

4.6 Experiments

We implemented our attack using the NTL [21] library version 3.1 developped by V. Shoup. We used two reduction algorithms: the LLL [11] algorithm to compute orthogonal lattices, and Schnorr's [17] Korkine-Zolotarev reduction algorithm with blocksize 20 to obtain better reduced bases. The implementation is fast: when $m \leq 300$ and M is no larger than 512 bits, the attack performed in less than 15 minutes on a 333MHz Ultrasparc-IIi. Heuristically, our attack works when the density is much smaller than 1, but only experiments can tell us what is exactly the limit. We stress that our implementation has not been optimized, which means that it might be possible to go a little bit further than what we obtained. For instance, one might try improved reduction algorithms such as [20]. In all our experiments, the attack worked as soon as step 1 was correct.

We first experimented the attack on hidden subset sums. If M is a 160-bit number (resp. 512-bit) , the attack works up to $n \approx 45$ (resp. 90) with $m \approx 90$ (resp. 200). We were not able to attack larger values of n, even with larger values of m (up to 400). For affine hidden subset sums, results are not as good: if M is a 160-bit number (resp. 512-bit), the attack works up to $n \approx 35$ (resp. 60) with $m \approx 90$ (resp. 150). These results show that the conditions of validity for the attack which we gave previously are quite pessimistic. In particular, it appears that the attack is effective against small values of n, which are required in a smartcard environment. Analysing table 1, we find that in the smartcard case, the HSS generator cannot be more efficient than the method of LL [12] for 160-bit and 512-bit exponents.

However, there is quite a gap between the largest parameters that our attack can handle and the parameters suggested in the scheme. When $M^{1/n}$ is very small, even very short vectors can be orthogonal to $\mathbf{v}_{\alpha}$, so that step 1 is highly unlikely to suceed. This is for instance the case with $n = \log_2 M$. For such a

n, our attack cannot even exploit the sparsity of the subset sums, and the best attack remains the birthday attack. It follows that if one is willing to pay with storage by choosing a sufficiently large value of n to foil the attack, then one can choose a small κ to reduce significantly the computation time. This does not seem to be very useful in the 160-bit case, as LL's method offers very good performances. But it improves the situation for 512-bit exponents. Hence, the hidden subset sum generator appears to be useful only for server applications, with exponents of at least 512 bits.

5 The Randomness of the Hidden Subset Sum Generator

We analyze the distribution of the output of the hidden subset sum generator, and discuss its implications on the security of the scheme. For fixed M, the distribution is exponentially close (with respect to n) to the uniform distribution. We provide a proof in two cases: when the $0,1$-coefficients are balanced, and when they are not. It was pointed out to us that such results were already known (technical result in [1], and a particular case is treated in [10]), but since our proof is quite different, we include it in appendix B. Our technique is based on the discrete Fourier transform, which might be of independent interest. The following result is proved in the extended version of [1] (Lemma 1, p12):

Theorem 5. *There exists a $c > 0$ such that for all $M > 0$, if $\alpha_1, \ldots, \alpha_n$ are independently and uniformly chosen from $[0..M-1]$, then the following holds with probability at least $1 - 2^{-cn}$:*

$$\sum_{a=0}^{M-1} \left| P\left(\sum_{j=1}^{n} x_j \alpha_j = a \right) - \frac{1}{M} \right| \leq 2^{-cn}\sqrt{M},$$

where P refers to a uniform and independent choice of the x_j's in $\{0,1\}$.

This means that for fixed M, with overwhelming probability on the choice of the α_i's, the distribution of the hidden subset sum generator is statistically close to the uniform distribution. And the result remains valid when one considers a polynomial number of samples instead of just one sample. More precisely, it is well-known that if for some distributions D_1 and D_2, the statistical difference of D_1 and D_2 is less than ε, then the statistical difference of D_1^m and D_2^m is less than $m\varepsilon$, where the notation D^m means picking m elements independently at random from D. For instance, it can be proved by introducing hybrid distributions consisting of k elements picked from D_1 and $m-k$ picked from D_2.

Thus, for a fixed M, with overwhelming probability on the choice of the α_i's, the distribution obtained by picking independently at random a polynomial number (in n) of outputs of the hidden subset sum generator corresponding to the α_i's is statistically close to the uniform distribution. In particular, a polynomial-time adversary cannot distinguish the two distributions. But a successful attack against a scheme (for instance, DSS) using the hidden subset sum generator would serve as a distinguisher for those distributions, assuming that the underlying scheme is unbreakable. Note that it was the case of our lattice-based attack.

Hence, the hidden subset sum generator is provably secure in this sense when the density is high. But this not very interesting from a practical point of view. Because when the density is high and the $0,1$-coefficients are balanced, the scheme does not save over the obvious square-and-multiply generator. However, for the moment, we do not know what happens precisely with the actual distribution of the generator, that is, when the subset sums are sparse. Our technique is able to deal with the case of unbalanced coefficients (see section B.2), but we are unable to extend it to the sparse distribution. Maybe the technique of [1] will be more useful.

6 Conclusion

Boyko *et al.* proposed several methods to produce simple and very fast methods for generating randomly distributed pairs of the form $(x, g^x \mod p)$ and $(x, x^e \mod N)$ using precomputation. For discrete-log-based schemes, the security of these generators against active attacks relied on the presumed hardness of the hidden subset sum problem. We showed that the security against passive attacks relied on a variant of this problem, which we called the affine hidden subset sum problem. We provided a security criterion for these problems, based on the density. On the one hand, we presented an effective lattice-based algorithm which can heuristically solve these problems when the density is very small. Experiments have confirmed the theoretical analysis, and show that the practical interest of the proposed schemes is marginal. When applied to protocols such as DSS, ElGamal, or Schnorr, the proposed methods cannot be significantly more efficient on smartcards than traditional exponentiation methods based on precomputation, without being vulnerable to attacks. On the other hand, we showed that when the density is high, the distribution of the output of the generator was exponentially close to the uniform distribution, which provides undistinguishability against polynomial-time adversaries. The two conditions complement each other, but there is still a gap. It would be interesting to reduce the gap, either by improving the attack, or the hardness results. In particular, it would be nice to obtain a hardness result of practical interest for the actual hidden subset sum generator which uses sparse subset sums.

Acknowledgements. We would like to thank the anonymous referees for their helpful comments.

References

1. M. Ajtai. Generating hard instances of lattice problems. In *Proc. 28th ACM STOC*, pages 99–108, 1996. Extended version at `http://www.eccc.uni-trier.de/eccc/`.
2. V. Boyko, M. Peinado, and R. Venkatesan. Speeding up discrete log and factoring based schemes via precomputations. In *Proc. of Eurocrypt' 98*, volume 1403 of *LNCS*, pages 221–235. Springer-Verlag, 1998.

3. E. Brickell, D.M. Gordon, K.S. McCurley, and D. Wilson. Fast exponentiation with precomputation. In *Proc. of Eurocrypt'92*, volume 658 of *Lecture Notes in Computer Science*, pages 200–207. Springer-Verlag, 1993.
4. E. F. Brickell and K. S. McCurley. An interactive identification scheme based on discrete logarithms and factoring. *Journal of Cryptology*, 5(1):29–39, 1992.
5. M.J. Coster, A. Joux, B.A. LaMacchia, A.M. Odlyzko, C.-P. Schnorr, and J. Stern. Improved low-density subset sum algorithms. *Computational Complexity*, 2:111–128, 1992.
6. P. de Rooij. On the security of the Schnorr scheme using preprocessing. In *Proc. of Eurocrypt'91*, volume 547 of *LNCS*, pages 71–80. Springer-Verlag, 1991.
7. P. de Rooij. Efficient exponentiation using precomputation and vector addition chains. In *Proc. of Eurocrypt'94*, volume 950 of *Lecture Notes in Computer Science*, pages 389–399. Springer-Verlag, 1995.
8. P. de Rooij. On Schnorr's preprocessing for digital signature schemes. *Journal of Cryptology*, 10(1):1–16, 1997.
9. T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, 31:469–472, 1985.
10. R. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. *Journal of Cryptology*, 9(4):199–216, 1996.
11. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.
12. C.H. Lim and P.J. Lee. More flexible exponentiation with precomputation. In *Proc. of Crypto'94*, volume 839 of *Lecture Notes in Computer Science*, pages 95–107. Springer-Verlag, 1994.
13. National Institute of Standards and Technology (NIST). *FIPS Publication 186: Digital Signature Standard*, May 1994.
14. P. Nguyen and J. Stern. Merkle-Hellman revisited: a cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations. In *Proc. of Crypto'97*, volume 1294 of *LNCS*, pages 198–212. Springer-Verlag, 1997.
15. P. Nguyen and J. Stern. Cryptanalysis of a fast public key cryptosystem presented at SAC '97. In *Proc. of SAC '98*, LNCS. Springer-Verlag, 1998.
16. P. Nguyen and J. Stern. The Béguin-Quisquater server-aided RSA protocol from Crypto '95 is not secure. In *Proc. of Asiacrypt '98*, volume 1514 of *LNCS*, pages 372–379. Springer-Verlag, 1998.
17. C.-P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
18. C.P. Schnorr. Efficient identification and signatures for smart cards. In *Proc. of Crypto'89*, volume 435, pages 239–252. Springer-Verlag, 1990.
19. C.P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.
20. C.P. Schnorr and H.H. Hörner. Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In *Proc. of Eurocrypt'95*, volume 921 of *LNCS*, pages 1–12. Springer-Verlag, 1995.
21. V. Shoup. Number Theory C++ Library (NTL) version 3.6. Can be obtained at `http://www.shoup.net/ntl/`.

A The Condition for Affine Hidden Subset Sums

We explain why the modified step 1 of the attack is still expected to succeed against affine hidden subset sums when the density is very small. This time, the vector $\mathbf{k}$ is long:

$$\|\mathbf{k}\| \approx M\sqrt{m/12}.$$

Therefore, we no longer know a high-dimensional sublattice of $(\mathbf{b},\mathbf{c})^\perp$ with small determinant. Still, we can hope that the first vectors of a reduced basis of $(\mathbf{b},\mathbf{c})^\perp$ will have norm around $(\|\mathbf{b}\| \times \|\mathbf{c}\|)^{1/(m-2)}\sqrt{m-2} \approx (mM^2/3)^{1/(m-2)}\sqrt{m-2}$ which is small for large m. But the explanations of section 4.1 regarding condition 4 are no longer convincing, because $\mathbf{p_u}$ cannot be guaranteed to be short, even if $\mathbf{u}$ is very short (which it is). Recall that

$$\mathbf{p_u} = (\mathbf{u}.\mathbf{x}_1, \ldots, \mathbf{u}.\mathbf{x}_n, \mathbf{u}.\mathbf{k}).$$

All the dot products $\mathbf{u}.\mathbf{x}_j$'s are still small, but the last coordinate $\mathbf{u}.\mathbf{k}$ might be large, since $\mathbf{k}$ is long. However, we still expect that $\mathbf{p_u}$ cannot be a non-zero vector of $\mathbf{v}_\alpha^\perp$ if $\mathbf{u}$ is very short, because most of its coordinates are very small.

To see this, let A be a bound for all the $|\mathbf{u}.\mathbf{x}_i|$'s, and B a bound for the last coordinate $|\mathbf{u}.\mathbf{k}|$. Denote by S the set of all vectors $(y_1, \ldots, y_{n+1}) \in \mathbb{Z}^{n+1}$ where all the y_i's are positive with $y_{n+1} \leq B$ and the remaining y_i's are less than A. There are $A^n B$ vectors in S. Now, consider the dot product of an element of S with $\mathbf{v}_\alpha$. This dot product is in absolute value less than $(nA + B)M$, so that there are most $2(nA + B)M$ different possible values. It follows by the pigeon-hole principle that if $A^n B > 2(nA + B)M$, there must be a collision, that is, there must exist two distinct vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ in S that have the same dot product with $\mathbf{v}_\alpha$, which yields by difference an non-zero orthogonal vector to $\mathbf{v}_\alpha$. The first n entries of this vector are less than A in absolute value, and the last entry is less than B in absolute value. This vector might be $\mathbf{p_u}$. But if $A^n B \ll 2(nA + B)M$, one does not expect any collision, and therefore $\mathbf{p_u}$ is unlikely to be a non-zero vector orthogonal to $\mathbf{v}_\alpha$. The parameter B has limited influence on this condition, it is the value of A which is preponderant. In other words, when $\mathbf{u}$ is short, $\mathbf{p_u}$ is not necessarily short, but all its entries except the last one (which corresponds to $\mathbf{k}$) are small. This makes a small bound for A and a large one for B, and therefore, the condition $A^n B \ll 2(nA + B)M$ is nevertheless satisfied when the density is small.

B A Fourier Analysis of the Hidden Subset Generator

We compare the distribution of the output of the hidden subset sum generator with the uniform distribution, in two cases: when the $0,1$-coefficients are balanced, and when they are not.

B.1 The Basic Case

Let M be an integer, and let $\alpha_1, \ldots, \alpha_n$ be independently and uniformly chosen from $[0..M-1]$. We first investigate the basic case where a linear combination

$$\sum_{j=1}^{n} x_j \alpha_j \pmod{M}$$

is produced with the x_j's independently and uniformly chosen from $\{0,1\}$. We use the Fourier transform over the abelian group $\mathbb{Z}_M$. The characters $\chi_k(t) = e^{\frac{2\pi i k t}{M}}$ form an orthonormal basis of $L^2(\mathbb{Z}_M)$, endowed with the uniform probability measure and therefore any element f of $L^2(\mathbb{Z}_M)$ can be recovered from its Fourier coefficients $\hat{f}(k) = \frac{1}{M}\sum_{q=0}^{M-1} f(q)e^{-2\pi i k q/M}$, through the inverse Fourier formula:

$$f = \sum_{k=0}^{M} \hat{f}(k)\chi_k.$$

We now evaluate the expectation of each χ_k with respect to the image probability of the product probability over $\{0,1\}^m$ induced by the transformation: $(x_1,\ldots,x_n) \longmapsto \sum_{j=1}^n x_j\alpha_j$. We get for $k \neq 0$:

$$E(\chi_k) = E\left(e^{2\pi i k \sum_{j=1}^n x_j\alpha_j/M}\right) = \prod_{j=1}^n \frac{1}{2}\left(1 + e^{2\pi i k\alpha_j/M}\right).$$

Since $|1+e^{i\theta}|^2 = (1+\cos\theta)^2 + \sin^2\theta = 2 + 2\cos\theta$, it follows that:

$$\sum_{k=1}^{M-1} |E(\chi_k)|^2 = \sum_{k=1}^{M-1}\prod_{j=1}^n \frac{1+\cos(2\pi k\alpha_j/M)}{2}.$$

We estimate this expression, with respect to a uniform choice of the α_j's in $[0..M-1]$:

Lemma 6. *Let k be an integer in $[1..M-1]$. If α is a random integer uniformly chosen from $[0..M-1]$ then:*

$$E[\cos(2k\pi\alpha/M)] = E[\sin(2k\pi\alpha/M)] = 0.$$

Proof. Let $\theta = 2k\pi/M$. By definition, the two expectations are respectively the real and imaginary part of: $E = \frac{1}{M}\sum_{\alpha=0}^{M-1} e^{i\alpha\theta}$. But since $k \in [1..M-1]$, the complex $e^{i\theta}$ is an M-th root of unity different from 1. Therefore the geometric sum is actually equal to zero, which completes the proof. □

Corollary 7. *Let $\varepsilon > 0$. If the α_j's are independently and uniformly chosen from $[0..M-1]$, then the following holds with probability at least $1-\varepsilon$:*

$$\sum_{k=1}^{M-1}\prod_{j=1}^n \frac{1+\cos(2\pi k\alpha_j/M)}{2} \leq \frac{M2^{-n}}{\varepsilon}.$$

Proof. Denote by X the left-hand random variable. By independence of the α_j's, the previous lemma shows that:

$$E(X) = \sum_{k=1}^{M-1}\prod_{j=1}^n \frac{1}{2} \leq \frac{M}{2^n}.$$

And the result follows by Markov inequality. □

Now assume that the α_j's satisfy the inequality of the previous proposition for some $\varepsilon > 0$. Then:

$$\sum_{k=1}^{M-1} |E(\chi_k)|^2 \leq \frac{M2^{-n}}{\varepsilon}.$$

This in turn means that $f - \hat{f}(0) = \sum_{k=1}^{M} \hat{f}(k).\chi_k$ has expectation bounded by $||f||.\sqrt{M2^{-n}/\varepsilon}$, where $||f||$ denotes the L^2-norm of f with respect to the uniform probability on $\mathbb{Z}_M$. This reads:

$$|E(f) - \hat{f}(0)| \leq ||f||.\sqrt{M2^{-n}/\varepsilon},$$

and establishes a bound on the difference between the expectation $E(f)$ and the expectation of the same function f taken over the uniform probability on $\mathbb{Z}_M$.

Applying to a given event X, and using corollary 7, we obtain:

Theorem 8. *Let $\varepsilon > 0$. If $\alpha_1, \ldots, \alpha_n$ are independently and uniformly chosen from $[0..M-1]$, then the following holds with probability at least $1 - \varepsilon$: for any subset X of $\mathbb{Z}_M$ with uniform probability δ, the probability δ' of X with respect to the image probability of the product probability over $\{0,1\}^m$ induced by the transformation $(x_1, \ldots, x_n) \longmapsto \sum_{j=1}^{n} x_j\alpha_j$, differs from δ by an amount bounded by:*

$$\sqrt{\delta M 2^{-n}/\varepsilon}.$$

B.2 The Case of Unbalanced Coefficients

We now assume that the coefficients x_j's are unbalanced and chosen according to a distribution where zeros are more likely to appear. We consider the probability ditsribution on $\{0,1\}$ where one receives probability p and zero $1-p$ and we endow $\{0,1\}^n$ with the product probability P_p. It is easy to show that the results of the previous section go through *mutatis mutandis*. The main difference is that the expectation of χ_k becomes

$$\prod_{j=1}^{n} (1-p) + pe^{\frac{2\pi i k\alpha_j}{M}}$$

An easy computation shows that this amounts to replacing the term $\frac{1}{2}[1 + \cos(2\pi k\alpha_j/M)]$ by $(1-p)^2 + 2p(1-p)\cos(2\pi k\alpha_j/M) + p^2$ in the expression of $|E(\chi_k)|^2$. Lemma 6 shows that the cosine term has zero mean, with respect to a uniform choice of α_j from $[0..M-1]$. It follows that the previous term has expectation equal to $(1-p)^2 + p^2$. We finally get:

Theorem 9. *Let $\varepsilon > 0$. If $\alpha_1, \ldots, \alpha_n$ are independently and uniformly chosen from $[0..M-1]$, then the following holds with probability at least $1 - \varepsilon$: for any subset X of $\mathbb{Z}_M$ with uniform probability δ, the probability δ' of X, with respect to the image probability of P_p induced by the transformation $(x_1, \ldots, x_n) \longmapsto \sum_{j=1}^{n} x_j\alpha_j$, differs from δ by an amount bounded by*

$$\sqrt{\delta M((1-p)^2 + p^2)^n/\varepsilon}.$$