Full version of the extended abstract which appeared in the
Proceedings of Eurocrypt '2000 (19th IACR Eurocrypt, May 14–18, 2000, Bruges, Belgium)
B. Preneel (Ed.), vol. ???? of Lecture Notes in Computer Science, pages ???–???

Noisy Polynomial Interpolation and Noisy Chinese Remaindering

(Full Version)

Daniel Bleichenbacher[1] and Phong Q. Nguyen[2]

[1] Bell Laboratories, Rm. 2A-366, 700 Mountain Av
Murray Hill, NJ 07974-0636, USA
bleichen@bell-labs.com and http://www.bell-labs.com/user/bleichen/

[2] École Normale Supérieure, Département d'Informatique,
45 rue d'Ulm, 75005 Paris, France
pnguyen@ens.fr and http://www.di.ens.fr/~pnguyen/

Abstract. The noisy polynomial interpolation problem is a new intractability assumption introduced last year in oblivious polynomial evaluation. It also appeared independently in password identification schemes, due to its connection with secret sharing schemes based on Lagrange's polynomial interpolation. This paper presents new algorithms to solve the noisy polynomial interpolation problem. In particular, we prove a reduction from noisy polynomial interpolation to the lattice shortest vector problem, when the parameters satisfy a certain condition that we make explicit. Standard lattice reduction techniques appear to solve many instances of the problem. It follows that noisy polynomial interpolation is much easier than expected. We therefore suggest simple modifications to several cryptographic schemes recently proposed, in order to change the intractability assumption. We also discuss analogous methods for the related noisy Chinese remaindering problem arising from the well-known analogy between polynomials and integers.

1 Introduction

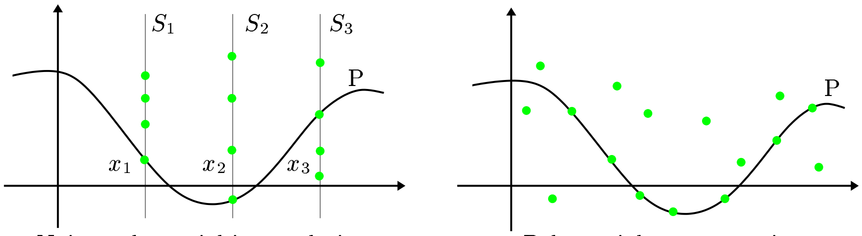
At STOC '99, Naor and Pinkas [26] introduced a new and useful primitive: oblivious evaluation of polynomials, where a polynomial P is known to Bob and he would like to let Alice compute the value $P(x)$ for an input x known to her in such a way that Bob does not learn x and Alice does not gain any additional information about P. The scheme they proposed is quite attractive, as it is much more efficient than traditional oblivious evaluation protocols, which leads to several applications. For instance, Gilboa [14] applied the scheme to two party RSA key generation. Naor and Pinkas mention other interesting applications in their paper [26], such as a method enabling two agencies each having a list of names, to find the common names on the lists without revealing other information.

Perhaps the only problem with the Naor-Pinkas scheme was a security issue, since the scheme used a new intractability assumption. The underlying computational problem, the so-called noisy polynomial interpolation problem, can be stated as follows:

Problem 1 (Noisy polynomial interpolation). Let P be a k-degree polynomial over a finite field $\mathbb{F}$. Given $n > k + 1$ sets $S_1, \ldots, S_n$ and n distinct elements x_1, $\ldots$, $x_n \in \mathbb{F}$ such that each $S_i = \{y_{i,j}\}_{1 \leq j \leq m}$ contains $m - 1$ random elements and $P(x_i)$, recover the polynomial P, provided that the solution is unique.

A simple counting argument suggests that $m^n \ll |\mathbb{F}|^{n-(k+1)}$ should be satisfied to ensure the unicity of the solution. Several generalizations are possible: for instance, one can assume that the sets S_i's have different sizes instead of m. A related problem is the following:

Problem 2 (Polynomial reconstruction). Given as input integers k, t and n points $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{F}^2$, output all univariate polynomials P of degree at most k such that $y_i = P(x_i)$ for at least t values of i.



Noisy polynomial interpolation   Polynomial reconstruction

The polynomial reconstruction problem is well-known because the generalized Reed-Solomon list decoding problem reduces to it. The best algorithm known to solve this problem is the recent algorithm of Guruswami and Sudan [17] (GS), which was inspired by previous work of Ar *et al.* [3] on a related problem. Its running time is polynomial in n, and the algorithm succeeds provided $t > \sqrt{kn}$, for any field $\mathbb{F}$ of cardinality at most 2^n. Naor and Pinkas remarked the existence of a simple reduction from noisy polynomial interpolation to polynomial reconstruction, which led them to conjecture that the noisy polynomial interpolation problem was as hard as the polynomial reconstruction problem.

This paper provides evidence that the conjecture is likely to be false. More precisely, we present new methods to solve noisy polynomial interpolation which (apparently) do not apply to polynomial reconstruction. In particular, we prove that the noisy polynomial interpolation problem can be transformed into a lattice shortest vector problem with high probability, provided that the parameters satisfy a certain condition that we make explicit. This result is qualitatively similar to the well-known lattice-based methods [20, 9] to solve the subset sum problem: the subset sum problem can be transformed into a lattice shortest vector problem with high probability, provided that a so-called low-density condition is satisfied. As with subset sums, experimental evidence suggest that most practical instances of the noisy polynomial interpolation problem with small m can be solved. It follows that noisy polynomial interpolation is much easier than expected (despite known hardness results [2, 24] on the lattice shortest vector problem), and thus, should be used cautiously as an intractability assumption.

Interestingly, the noisy polynomial interpolation and the polynomial reconstruction problems also appeared in password authentication schemes [25, 13]. Both schemes use Shamir's secret sharing scheme based on Lagrange's polynomial interpolation, where the shares are encrypted with low entropy secrets. Shamir's scheme achieves perfect security, but here, additional information is available to the attacker. A closer inspection shows that [13] is based on the noisy polynomial interpolation problem, and is therefore insecure for many choices of the parameters. For instance, the authors propose to use $n = 22, k = 14$ and $m \approx 256$ to protect a 112-bit key. But this configuration can be broken using a meet-in-the-middle attack (see Section 2.3) using $n' = 16$ in time 2^{64}. The solution described in [25] is much better as it is based on the hardness of the discrete log problem and a variant of the polynomial reconstruction problem.

We also discuss analogous methods for a related problem, the so-called noisy Chinese remaindering problem arising from the well-known analogy between polynomials and integers. Curiously, problems such as point counting on elliptic curves over finite fields and integer factorization of the form p^2q, can be viewed as generalized noisy Chinese remaindering problems. We explain why the lattice-based approach does not appear to be as useful in such settings.

The paper is organized as follows. In Section 2, we review simple methods for noisy polynomial interpolation. Section 3 is devoted to lattice-based methods. Cryptographic implications of these results are discussed in Section 4. In Section 5, we study analogous methods for the noisy Chinese remaindering problem. Due to lack of space, some details and proofs are omitted, but those can be found in the full version available on our webpages.

2 Simple methods for noisy polynomial interpolation

2.1 An error-correction method

When the noisy polynomial interpolation problem appeared in [26], the only known algorithm to solve it (apart from exhaustive search) was based on a simple reduction from noisy polynomial interpolation to polynomial reconstruction. More precisely, Naor and Pinkas noticed that by randomly choosing one element $y_{i,j}$ in S_i, one obtains an instance of the polynomial reconstruction problem with the n (randomly chosen) points $(x_i, y_{i,j})$. The solution P is of degree k, and we have $P(x_i) = y_{i,j}$ for approximately n/m values of i. Therefore the solution is expected to be outputted by the GS algorithm, provided that $\frac{n}{m} > \sqrt{kn}$, that is: $m < \sqrt{\frac{n}{k}}$. In fact, one can obtain a better reduction by taking all the points, which was apparently unnoticed. Indeed, if one picks all the nm points $(x_i, y_{i,j})$, then the solution P of degree k satisfies $P(x_i) = y_{i,j}$ for at least n values of (i, j). Hence, the GS algorithm will output P if $n > \sqrt{knm}$, that is:

$$m < \frac{n}{k}.$$

It is worth noting that this condition does not depend on the size of the finite field. The previous reductions do not use the specificity of the noisy polynomial interpolation instances. It is not known whether one can improve GS algorithm when applied to those particular instances, although [6] describes a simple algorithm achieving the same bound $m < n/k$. We now present methods to solve the problem when the condition $m < n/k$ is not satisfied.

2.2 A Gröbner basis method

A natural way to solve the noisy polynomial interpolation problem is reducing the problem to solving a system of polynomial multivariate equations. Write the unknown polynomial P as $P(X) = \sum_{i=0}^{k} a_i X^i$. For all i, there exists j such that $P(x_i) = y_{i,j}$, therefore:

$$\prod_{j=1}^{m} (P(x_i) - y_{i,j}) = 0.$$

One thus obtains n polynomial equations in the $k+1$ unknowns $a_0, \ldots, a_k$, in the field $\mathbb{F}$.

Gröbner basis is the usual way to solve such systems. However, the complexity of such techniques is super-exponential in k: in practice, it is likely that the method would be impractical if k is not very small (for instance, larger than 20). Theoretically, one could also apply the relinearization technique recently introduced by Kipnis and Shamir [19], at least in the case $m = 2$ (that is, a system of quadratic equations). At the moment, the behaviour of this new method is not completely understood, however latest results [10] suggest that the method is impractical for sufficiently large k, such as $k \geq 50$.

2.3 A meet-in-the-middle method

A meet-in-the-middle approach can be used to solve the noisy polynomial interpolation problem. Let $n' \leq n$ be the smallest integer for which we expect the solution to be unique. Define the Lagrange interpolation polynomials in $\mathbb{F}[X]$:

$$L_i(X) = \prod_{\substack{1 \leq j \leq n' \\ j \neq i}} \frac{X - x_j}{x_i - x_j}.$$

The degree of L_i is $n' - 1$. We are looking for coefficients c_i, such that

$$\deg \left(\sum_{i=1}^{n'} y_{i,c_i} L_i(X) \right) \leq k.$$

For all $\mathbf{c} = (c_1, \ldots, c_{\lfloor n'/2 \rfloor}) \in \{1, \ldots, m\}^{\lfloor n'/2 \rfloor}$ and $\tilde{\mathbf{c}} = (c_{\lfloor n'/2 \rfloor + 1}, \ldots, c'_n) \in \{1, \ldots, m\}^{\lceil n'/2 \rceil}$ we compute the polynomials $U_{\mathbf{c}}(X) = \sum_{i=1}^{\lfloor n'/2 \rfloor} y_{i,c_i} L_i(X)$ and $V_{\tilde{\mathbf{c}}}(X) = -\sum_{i=\lfloor n'/2 \rfloor + 1}^{n'} y_{i,c_i} L_i(X)$. We compare the two lists: If some $U_{\mathbf{c}}(X)$

and $V_{\tilde{\mathbf{c}}}(X)$ have identical coefficients for the terms $X^{k+1}, \ldots, X^{n'}$ then $U_{\mathbf{c}}(X) - V_{\tilde{\mathbf{c}}}(X)$ has degree at most k, and therefore, solves the problem.

The method requires the computation of $O(m^{\lceil n'/2 \rceil})$ polynomials $U_{\mathbf{c}}(X)$ and $V_{\tilde{\mathbf{c}}}(X)$. Since the values for $y_{i,j} L_i(X)$ can be precomputed and partial sums can be reused, the time complexity of this attack is $O(c(n'-k)m^{\lceil n'/2 \rceil})$, where c is the time for an addition in $\mathbb{F}$. The memory requirement of this algorithm is $O((\log q)m^{\lfloor n'/2 \rfloor})$, but an improved algorithm needing $O((\log q)m^{\lceil n'/4 \rceil})$ exists.

It is worth noting that the meet-in-the-middle method does not apply to the polynomial reconstruction problem. This is because the Lagrange polynomials $L_i(X)$ in this problem depend on the selection of the values $y_{i,j}$ used for the interpolation. Different $y_{i,j}$'s correspond to different x_i's and therefore different Lagrange polynomials. The meet-in-the-middle method takes advantage of the fact that the x_i's are known in advance.

Note that the meet-in-the-middle method can still be used if we have to compute $g^{f(x_0)}$ for some public x_0 and g, when given the $g^{y_{i,j}}$'s rather than the $y_{i,j}$'s. This is because polynomial interpolation is a linear function of the inputs $y_{i,j}$.

3 Lattice-based methods for noisy polynomial interpolation

We now describe lattice-based methods to solve noisy polynomial interpolation. To simplify the presentation, we assume in the whole section that the finite field $\mathbb{F}$ is a prime field $\mathbb{Z}_q$ (q being a prime number). The results extend to the general case by viewing $\mathbb{F}$ as a finite dimensional vector space over its prime field.

In this paper, we will call *lattice* any integer lattice, that is, any subgroup of $(\mathbb{Z}^n, +)$ for some n. Background on lattice theory can be found in several textbooks, such as [16, 35]. For lattice-based cryptanalysis, we refer to [18].

Our lattice-based methods build in polynomial time a lattice from a given instance of noisy polynomial interpolation. In this lattice, there is a particular lattice point, the so-called *target vector*, which is both unusually short and closely related to the solution of our problem. We will first give heuristic arguments suggesting that the target vector is the lattice shortest vector. Then we will modify our lattice to prove that the target vector is with high probability the shortest vector of the modified lattice, when the parameters satisfy a certain condition that we make explicit. The proofs are somewhat technical, but the underlying idea is similar to the one used to show that the low-density subset sum problem can be reduced with high probability to a lattice shortest vector problem [20, 9]. More precisely, we will estimate the probability that a fixed vector belongs to the lattice built from a randomly chosen instance of the problem. By enumerating all possible short vectors, we can then upper bound the probability that there exists a nonzero lattice point shorter than the target vector for a randomly chosen instance. From a practical point of view, one hopes to solve the problem by using standard lattice reductions algorithms [21, 30–32] as lattice shortest vector oracles.

3.1 Linearization of noisy polynomial interpolation

Let $L_i(X)$ be the Lagrange interpolation polynomial defined as

$$L_i(X) = \prod_{j \neq i} \frac{X - x_j}{x_i - x_j}.$$

The solution P satisfies: $P(X) = \sum_{i=1}^{n} P(x_i) L_i(X)$. We linearize the problem: letting $\delta_{i,j}$ equal to 1 if $P(x_i) = y_{i,j}$, and 0 otherwise, one obtains $P(x_i) = \sum_{j=1}^{m} \delta_{i,j} y_{i,j}$, hence:

$$P(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} \delta_{i,j} y_{i,j} L_i(X).$$

Since $P(X)$ has degree k, while L_i has degree $n-1$, we obtain $n-1-k$ linear equations in the nm unknowns $\delta_{i,j}$. As a linear system in the field $\mathbb{F}$, it is underdefined. However, one can also view the problem as a lattice problem for which lattice reduction might apply.[1]

The set L of integer row vectors $(d_{1,1}, d_{1,2}, \ldots, d_{n,m}) \in \mathbb{Z}^{nm}$ such that the polynomial $\sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} y_{i,j} L_i(X)$ has degree at most k is clearly a lattice in $\mathbb{Z}^{nm}$. The vector $(\delta_{1,1}, \delta_{1,2}, \ldots, \delta_{n,m})$ belongs to L, we call it the *target vector*. Its Euclidean norm is $\sqrt{n}$. To see how short this vector is compared to other lattice vectors, we need to analyze the lattice L. We wish to obtain results of the flavour of lattice-based algorithms to solve low-density subset sums [20, 9]: with high probability over a certain distribution of the inputs, and under specific conditions on the parameters, the target vector is the lattice shortest vector.

3.2 Volume of the lattice

The previous lattice is related to the lattices used by Ajtai [1] in his celebrated worst-case/average-case equivalence for certain lattice problems. More precisely, let A be a $n \times e$ matrix in $\mathbb{Z}_q$ where q is any integer. Let $L(A)$ be the set of n-dimensional integer row vectors $\mathbf{x}$ such that $\mathbf{x}A \equiv 0 \pmod{q}$. We call $L(A)$ the Ajtai lattice associated to A. It is easy to see that $L(A)$ is a n-dimensional lattice in $\mathbb{Z}^n$, from which one derives:

Lemma 3. *Let $A \in \mathcal{M}_{n,e}(\mathbb{Z}_q)$. Then the volume of $L(A)$ divides q^e. It is exactly q^e if and only if $\{\mathbf{x}A \ : \mathbf{x} \in \mathbb{Z}_q^n\}$ is entirely $\mathbb{Z}_q^e$.*

Proof. By definition, $L(A)$ is the kernel of the group homomorphism ϕ that maps any $\mathbf{x} \in \mathbb{Z}^n$ to $(\mathbf{x}A \bmod q) \in \mathbb{Z}_q^e$. Therefore the group quotient $\mathbb{Z}^n/L(A)$ is isomorphic to the image of ϕ. But since $L(A)$ is a full-dimensional lattice in $\mathbb{Z}^n$, its volume is simply the index $[\mathbb{Z}^n : L(A)]$ of $L(A)$ in $\mathbb{Z}^n$, from which both statements follow. $\square$

[1] If we used the field $\mathrm{GF}(q^a)$ rather than $\mathbb{Z}_q$ we would have $a(n-1-k)$ equations in nm unknowns over $\mathbb{Z}_q$ and the linear system might be solvable directly.

Letting $L_i(x) = \sum_{w=0}^{n-1} \ell_{i,w} x^w$, the lattice L of Section 3.1 is equal to $L(A)$, where $\mathbb{F} = \mathbb{Z}_q$ and A is the following matrix of dimension $nm \times n-1-k$.

$$A = \begin{pmatrix} y_{1,1}\ell_{1,k+1} & \cdots & y_{1,1}\ell_{1,n-1} \\ \vdots & & \vdots \\ y_{i,j}\ell_{i,k+1} & \cdots & y_{i,j}\ell_{i,n-1} \\ \vdots & & \vdots \\ y_{n,m}\ell_{n,k+1} & \cdots & y_{n,m}\ell_{n,n-1} \end{pmatrix}$$

Lemma 4. *Assume that for all $1 \leq i \leq n$ there exists $1 \leq w_i \leq m$, such that $y_{i,w_i} \neq 0$. Then* $\operatorname{rank}(A) = n-1-k$.

Proof. Remember, that for all $c_1, \ldots, c_n \in \mathbb{F}^n$ and $f(x) = \sum_{i=1}^{n} c_i L_i(x)$ we have $f(x_i) = c_i$. Hence, $\sum_{i=1}^{n} c_i L_i(x) = 0$ implies $c_1 = \cdots = c_n = 0$. This shows that the $n \times n$ matrix $(\ell_{i,j})_{1 \leq i \leq n; 0 \leq j \leq n-1}$ is nonsingular. In particular, the last $n-1-k$ columns are linearly independent and thus the matrix $(\ell_{i,j})_{1 \leq i \leq n; k+1 \leq j \leq n-1}$ has rank $n-1-k$. We assumed that $y_{i,w_i} \neq 0$ and therefore the matrix $A_0 = (y_{i,w_i}\ell_{i,j})_{1 \leq i \leq n; k+1 \leq j \leq n-1}$ has rank $n-1-k$ too. Since A_0 is a submatrix of A it follows that A has rank $n-1-k$ too. □

A consequence of this lemma is that the set $\{0, \ldots, q-1\}^{nm}$ contains exactly $q^{nm-n+1+k}$ lattice points and hence the volume of $L(A)$ is q^{n-1-k}. Therefore, if γ_d denotes Hermite's constant of order d, we have:

$$\lambda_1(L) \leq \sqrt{\gamma_{nm}} q^{\frac{n-1-k}{nm}},$$

where $\lambda_1(L)$ is the first minimum of L (the length of a shortest non-zero lattice point). The best asymptotic estimate known of Hermite's constant is the following (see [7]):

$$\frac{d}{2\pi e} + \frac{\log(\pi d)}{2\pi e} + o(1) \leq \gamma_d \leq \frac{1.744d}{2\pi e}(1 + o(1)).$$

It follows that one expects the target vector to be the shortest lattice vector if

$$\sqrt{n} \ll \sqrt{\frac{nm}{2\pi e}} q^{\frac{n-1-k}{nm}}.$$

This condition is very heuristic, as the lattice L cannot be considered as a "random" lattice.

3.3 Structure of the lattice

We now give a different heuristic argument to guess when the target vector is the shortest vector. The argument is inspired by lattice-based attacks against low-density subset sums (see [20, 9]). If we denote by $N(n, r)$ the number of integer points in the n-dimensional sphere of radius $\sqrt{r}$ centered at the origin, we have the following elementary result :

Lemma 5. *Let A be a $nm \times e$ matrix in $\mathbb{Z}_q$ (q prime) chosen at random with uniform distribution. Then:*

$$\Pr\left(\lambda_1(L(A)) < \sqrt{n}\right) \leq \frac{N(nm,n)}{q^e}.$$

Proof. Let $\mathbf{x} = (x_1, \ldots, x_{nm}) \in \mathbb{Z}_q^{nm}$ be a non-zero vector. The probability that $\mathbf{x}A \equiv 0 \pmod{q}$ for a uniformly chosen matrix $A = (a_{i,j})_{1 \leq i \leq nm, 1 \leq j \leq e}$ is q^{-e}. Indeed, there exists $i_0 \in \{1, \ldots, nm\}$ such that $x_{i_0} \neq 0$. Then, for any choice of $(a_{i,j})_{i \neq i_0, 1 \leq j \leq e}$, there exists a unique choice of $(a_{i_0,j})_{1 \leq j \leq e}$ such that $\mathbf{x}A \equiv 0 \pmod{q}$, which gives the expected probability. Since the number of possible $\mathbf{x}$ is less than $N(nm,n)$, the result follows. $\square$

It follows that one expects the target vector to be the shortest lattice vector when $N(nm,n) \ll q^{n-1-k}$. Numerical values of $N(nm,m)$ can be computed by recursion. And sharp theoretical estimates of $N(nm,m)$ can be obtained using the power series $h(x) = 1 + 2\sum_{k=1}^{\infty} x^{k^2}$ (see [23, Lemma 1]). However, the condition is still heuristic, since in our case, the matrix A cannot be considered as uniformly distributed. In particular, it does not seem easy to compute the probability that a fixed vector belongs to the lattice $L(A)$ for a randomly chosen instance of noisy polynomial interpolation.

3.4 Reduction by lattice improvement

To achieve a reduction from noisy polynomial interpolation to the lattice shortest vector problem, we consider a certain sublattice. The improvement is based on a property of the target vector which has not been used so far: for all i_1 and i_2, $\sum_{j=1}^{m} \delta_{i_1,j} = \sum_{j=1}^{m} \delta_{i_2,j} = 1$. This leads us to define the lattice Λ as the set of lattice points $(d_{1,1}, d_{1,2}, \cdots, d_{n,m}) \in L$ such that for all i_1 and i_2:

$$\sum_{j=1}^{m} d_{i_1,j} = \sum_{j=1}^{m} d_{i_2,j}. \tag{1}$$

Since Λ is the intersection of the full-dimensional lattice L (in $\mathbb{Z}^{nm}$) with a $(nm-n+1)$-dimensional vector subspace, Λ is a $(nm-n+1)$-dimensional lattice in $\mathbb{Z}^{nm}$, which can be computed in polynomial time.

We will be able to compute the probability that a (fixed) short vector satisfying (1) belongs to Λ, which was apparently not possible for L. The probability is with respect to the natural distribution induced by the definition of noisy polynomial interpolation, which is the following:

- Let $x_1, \ldots, x_n$ be distinct elements of $\mathbb{F} = \mathbb{Z}_q$, and g be a function from $\{1, \ldots, n\}$ to $\{1, \ldots, m\}$.
- Choose uniformly at random a k-degree polynomial P in $\mathbb{F}[X]$.
- For all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\} \setminus g(i)$, choose uniformly at random an element $y_{i,j}$ in $\mathbb{F}$, and let $y_{i,g(i)} = P(x_i)$.

Recall that the noisy polynomial interpolation problem is to recover either g or P, given k, the $y_{i,j}$'s and the x_i's. The (secret) function g indicates which $y_{i,j}$ is equal to $P(x_i)$.

Let $\mathbf{d} = (d_{1,1}, \ldots, d_{n,m}) \in \mathbb{Z}^{nm}$ be a vector satisfying (1). We define $p(\mathbf{d})$ as the probability that $\mathbf{d}$ belongs to the lattice Λ, that is, the probability that $\deg(\sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} y_{i,j} L_i(X)) \leq k$, with respect to the previous distribution. Let $t(\mathbf{d})$ be the number of indices i for which there exists at least one nonzero $d_{i,j}$ modulo q with $j \neq g(i)$:

$$t(\mathbf{d}) = \left| \{1 \leq i \leq n : \exists j \in \{1, \ldots, m\} \setminus g(i) \text{ such that } d_{i,j} \not\equiv 0 \bmod q\} \right|.$$

The following technical lemma gives a formula for $p(\mathbf{d})$. It shows that the heuristic assumptions made in Section 3.2 and Section 3.3 are correct for all vectors $\mathbf{d}$ where $t(\mathbf{d}) \geq n - k - 1$, but $p(\mathbf{d})$ is larger than expected when $t(\mathbf{d}) < n - k - 1$. As we will see later the effect of those vectors is often negligible. A proof can be found in appendix A.

Lemma 6. *Let* $\mathbf{d} \in \mathbb{Z}^{nm}$ *satisfying* (1). *Then:*

$$p(\mathbf{d}) = q^{-\min(t(\mathbf{d}), n-k-1)}.$$

It follows that $p(\mathbf{d}) > \frac{1}{q}$ if and only if $t(\mathbf{d}) = 0$ (recall that $n > k + 1$). But if $\mathbf{d}$ satisfies (1) and $t(\mathbf{d}) = 0$, then either $\mathbf{d}$ is a multiple (possibly zero) of the target vector, or at least one of $\mathbf{d}$'s entries is a nonzero multiple of q, implying $\|\mathbf{d}\| \geq q$. By enumerating all possible $\mathbf{d}$'s, we finally obtain a reduction:

Theorem 7. *Let* $\sqrt{r} < q$. *Let a noisy polynomial interpolation instance be chosen uniformly at random as described above and let* Λ *be the sublattice built from the instance. Then the expected number of nonzero vectors* $E(r, n, m)$ *contained in* Λ *not equal to the target vector or a multiple of it with norm* $\leq \sqrt{r}$ *is:*

$$E(r, n, m) = \sum_{\lambda = -\lfloor r/n \rfloor}^{\lfloor r/n \rfloor} \sum_{w=1}^{n} R(w, r, \lambda, n, m) \; q^{-\min(w, n-k-1)},$$

where $R(w, r, \lambda, n, m)$ *denotes the number of vectors* $\mathbf{d} = (d_{1,1}, \ldots, d_{n,m}) \in \mathbb{Z}^{mn}$ *such that* $t(\mathbf{d}) = w$, $\|\mathbf{d}\| \leq \sqrt{r}$ *and* $\sum_{j=1}^{m} d_{i,j} = \lambda$.

If $E(n, n, m) < 1$ then $E(n, n, m)$ is a nontrivial upper bound on the probability that Λ contains a nonzero vector shorter than the target vector. The proof of Theorem 7 and numerical methods to compute $E(r, n, m)$ are given in appendix B. The results are more complicated than low-density subset sum attacks for the following reasons. In low-density subset sum attacks, one can compute fairly easily an upper bound of the probability that a fixed nonzero short vector (different from the target vector) belongs to a certain lattice built from the subset sum instance (see [20, 9]). And the bound obtained is independent of the vector. It then remains to estimate the number of possible short vectors, by bounding the number of integer points in high-dimensional spheres (using techniques of [23]).

Here, we have an exact formula for the probability instead of an upper bound, but the formula depends on the vector, for it involves $t(\mathbf{d})$. This leads to more complicated enumerations and asymptotic formulas. Hence, we cannot give a criterion as "simple" as the low-density criterion for subset sum, to indicate when the reduction is expected to hold. However, for some special cases we have some preliminary results:

Lemma 8. *Let $n \geq 2$, $m \geq 2$ and $n^2 < q$. Let $0 < x < 1$ and $h(x) = 1 + 2\sum_{k=1}^{\infty} x^{k^2}$. Then:*

$$\frac{N(n, \lfloor n/2 \rfloor) + 2^{n+1} - 3}{q^{n-1-k}} \leq E(n, n, 2) \leq \frac{N(n, \lfloor n/2 \rfloor) + 2^{n+1}}{q^{n-1-k}} + 2n^2/q + 4n/q \tag{2}$$

$$E(n, n, m) \leq \frac{N(nm, n)}{q^{n-1-k}} + 3x^{-n}\left(\left(1 + \frac{h(x)^m}{q}\right)^n - 1\right) \tag{3}$$

The proof of Lemma 8 can be found in Appendix D. Note that $h(x)$ can be approximated numerically. The result for the case $m = 2$ are much stronger than the result for a general m. From a practical point of view, we can alternatively compute the upper bound $E(r, n, m)$ numerically for any given choice of the parameters. And the bound seems to be sharp in practice.

The following table shows for some values of m, n, q the largest k, such that the expected number of vectors with norm shorter or equal to $\sqrt{n}$ is smaller than 1. We compare this to the largest $\tilde{k}$ for which we would expect the target vector to be the shortest vector in the original lattice without improvement.

A missing entry in the column k says that for this particular choice of m and n the problem is very likely not solvable with the lattice based method for any k. We have chosen m and n such that the meet-in-the middle method has a time complexity of 2^{80}. We have chosen $q > 2^{80}$, so that elements of $\mathbb{Z}_q$ can be used to represent 80 bit keys for symmetric ciphers.

m	n	$\log_2(q)$	k	$\tilde{k}$
2	160	80	155	152
3	115	80	110	108
4	105	80	100	98
16	44	80	40	39
256	20	80	–	–

3.5 Non-prime fields

When $\mathbb{F}$ is a field of the form $\mathrm{GF}(q^a)$ with $a > 1$, Lemma 6 still holds if one replaces q by q^a, with the same definition of $t(\mathbf{d})$ (that is, the number of indices i for which there exists at least one nonzero $d_{i,j}$ modulo q with $j \neq g(i)$), so that $p(\mathbf{d}) = q^{-a \min(t(\mathbf{d}), n-k-1)}$. Theorem 7 and Lemma 8 need to be modified accordingly. It follows that the lattice-based approach is useful only when the characteristic of $\mathbb{F}$ is sufficiently high ($q > \sqrt{n}$), so that any vector $\mathbf{d}$ satisfying $t(\mathbf{d}) = 0$ is strictly longer than the target vector.

3.6 Experiments

We implemented the improved lattice-based method on a 500 MHz 64-bit DEC Alpha using Victor Shoup's NTL library [34]. For a randomly chosen instance, we built the corresponding sublattice Λ. For lattice reduction, we successively applied three different types of reduction : plain LLL [21], Schnorr's BKZ reduction [30, 31] with block size 20, and when necessary, Schnorr-Hörner's pruned BKZ reduction [32] with block size 54 and pruning factor 14. We stopped the reduction as soon as the reduced basis contained the target vector.

To fix ideas on the efficiency of the lattice-based method, we chose $n = 160$ and $m = 2$, with a prime field of size 80 bits. The error-correction method succeeds only if $k \leq 80$. The meet-in-the-middle method requires at least $2^{k/2}$ operations. And the Gröbner basis approaches are very unlikely to be practical. Numerical values given by theorem 7 (see Section 3.4) suggest that the noisy polynomial interpolation problem can be reduced to a lattice shortest vector problem, as while as $k \leq 155$. The lattice dimension is then 160. Our implementation was able to solve noisy polynomial interpolation up to $k = 154$. For $k \leq 152$, only BKZ-20 reduction was necessary, and the total running time was less than 4 hours. For $153 \leq k \leq 154$, an additional Schnorr-Hörner pruned BKZ reduction was necessary: 1 hour for $k = 153$, and 8 hours for $k = 154$. We do not know if the theoretical value of $k = 155$ can be reached in practice: the corresponding lattice problem is hard because there are many lattice points almost as short as the target vector. The situation might be similar to lattice-based subset sum attacks: when the subset sum density is very close to the critical density, and the lattice dimension is large, the lattice problem is hard. It is worth noting that to ensure the unicity of the solution, one should have $k \leq 156$. This suggests that the lattice-based method is likely to solve most instances of practical interest for small m. We also made a few experiments with $m > 2$. A BKZ-20 reduction can solve in one day the problem with $n = 115$, $k = 101$, $m = 3$ and $n = 105$, $k = 80$, $m = 4$. For such parameters, the meet-in-the-middle method requires at least 2^{80} operations.

4 Cryptographic implications

We showed that when the parameters satisfy a certain relation, there exists a provable reduction from noisy polynomial interpolation to the lattice shortest vector problem. This results in an attack which is much more effective than previously known methods based on list decoding algorithms, due to the strength of current lattice reduction algorithms. We could not apply the same method to the polynomial reconstruction problem. This suggests (but does not prove) that the polynomial reconstruction problem is harder than the noisy polynomial interpolation problem, so that Conjecture 3.1 in [26] about the hardness equivalence[2] of the two problems does not hold.

It follows that cryptographic protocols should – if possible – be based on the polynomial reconstruction problem rather than the noisy polynomial interpolation problem. Such a change is possible for the oblivious polynomial evaluation of Naor and Pinkas [26]. There are two players Alice and Bob. Bob's secret input is a polynomial $P(x)$, which he hides in a bivariate polynomial $Q(x, y)$, such that $Q(0, y) = P(y)$. Alice has a secret value α and would like to learn $P(\alpha)$. Alice chooses a polynomial $S(x)$ with $S(0) = \alpha$. In a crucial step of the protocol Alice would like to learn $Q(x_i, S(x_i))$ without revealing $S(x)$. This is done by sending x_i and a list of random values $y_{i,j}$, except that one value $S(x_i)$. Bob computes $Q(x_i, y_{i,j})$ for all these values and A retrieves the answer she is interested in using a 1-out-of-m oblivious transfer. The privacy of Alice depends on the difficulty to find $S(x)$ given x_i and $y_{i,j}$, *i.e.* the noisy polynomial interpolation problem. However, the protocol can be changed by using the values $Q(x_{i,j}, y_{i,j})$ for distinct $x_{i,j}$'s rather than $Q(x_i, y_{i,j})$ [29].

Another way to prevent lattice-based attacks is to use a field where computing discrete logarithms is intractable, and to publish the powers $g^{y_{i,j}}$ rather than the values $y_{i,j}$. It is then still possible to perform a polynomial interpolation, that is to compute $g^{f(x_0)}$, given sufficiently many values $g^{f(x_i)}$. In fact, the meet-in-the middle method is the only algorithm known to us that is applicable in this case and it can only be used for the noisy polynomial interpolation problem but not for the polynomial reconstruction problem. A protocol using the polynomial interpolation problem combined with the discrete logarithm problem is described in [25].

5 Noisy Chinese remaindering

There is a well-known analogy between polynomials and integers: the polynomial degree corresponds to the integer size; Lagrange's interpolation corresponds to Chinese remainders; and polynomial evaluation corresponds to the modulo operation (in fact, a polynomial P evaluated at x_0 can also be viewed as the remainder of $P(x)$ modulo the linear polynomial $x - x_0$). We refer to [15] for some examples. The noisy polynomial interpolation and polynomial reconstruction problems then become the following ones:

Problem 9 (Noisy Chinese remaindering). Let $0 \le N \le B$, and $p_1, \ldots, p_n$ be coprime integers. Given n sets $S_1, \ldots, S_n$ where each $S_i = \{r_{i,j}\}_{1 \le j \le m}$ contains $m - 1$ random elements in $\mathbb{Z}_{p_i}$ and $N \mod p_i$, recover the integer N, provided that the solution is unique (*e.g.*, $m^n B \ll \prod_{i=1}^{n} p_i$).

---

[2] In fact, Conjecture 3.1 relates the hardness of polynomial reconstruction and an easier version of noisy polynomial interpolation.

Problem 10 (Chinese remaindering with errors). Given as input integers t, B and n points $(r_1, p_1), \ldots, (r_n, p_n) \in \mathbb{N}^2$ where the p_i's are coprime, output all numbers $0 \leq N < B$ such that $N \equiv r_i \pmod{p_i}$ for at least t values of i.

We refer to [15] for a history of the latter problem, which is beyond the scope of this article. We will only mention that the best decoding algorithm known for the problem is the recent lattice-based work of Boneh [6], which improves previous work of Goldreich *et al.* [15]. The algorithm works in polynomial time and solves the problem provided that a certain condition is satisfied. The exact condition is analogous to the bound obtained by GS algorithm for polynomial reconstruction.

We note that there are two well-known problems for which the general noisy Chinese remaindering problem (in which one allows different sizes for the sets S_i's) arises. The first problem is point counting on elliptic curves over finite fields. The best general algorithm for this problem is the Schoof-Elkies-Atkin (SEA) algorithm [33, 12, 4, 5] (see [22] for implementation issues). Let E be an elliptic curve over a finite field of cardinality q. Hasse's theorem states that the cardinality of E is of the form $q+1-t$ where $|t| \leq 2\sqrt{q}$. The SEA algorithm tries to determine this t, using Chinese remainders. However, in practice, it turns out to be too expensive to compute the exact value of t modulo sufficiently many coprime numbers. Therefore, one actually determines many coprime numbers of two kinds: for the first kind of numbers, t modulo such numbers is exactly known; for the second kind of numbers, the value of t modulo such numbers is constrained to a small number of values. This is exactly a noisy Chinese remaindering problem. To solve this problem, current versions of SEA apply a meet-in-the-middle strategy. The second problem is integer factorization of numbers of the form $N = p^2 q$. It has been noticed for some time (see for instance [28]) that for any number r, the Jacobi symbol $\left(\frac{r}{N}\right)$ is equal to the Legendre symbol $\left(\frac{r}{q}\right)$. It follows that for any number r, $q \bmod r$ is limited to half of $\mathbb{Z}_r$, and such a half can be determined. The problem of computing q can thus be viewed as a noisy Chinese remaindering problem. However, the S_i's are so dense that this formulation is likely to be useless.

We briefly review methods for noisy Chinese remaindering, analogous to the ones we described for noisy polynomial interpolation. One can first use the analog of the meet-in-the-middle method of Section 2.3. One can also use the reduction to Chinese remaindering with errors and the algorithm of [6], in a way analogous to Section 2.1. But the following simpler method achieves the same results.

5.1 Coppersmith's method

We obtain an analogous method to the Gröbner basis approach by translating the problem in terms of polynomial equations. The solution N satisfies for each i the following equation:

$$\prod_{j=1}^{m} (N - r_{i,j}) \equiv 0 \pmod{p_i}.$$

Using Chinese remainders and collecting all equations, one obtains a univariate polynomial equation of degree m in the unknown N modulo $\prod_{i=1}^{n} p_i$. We then apply the following lattice-based result by Coppersmith [8] :

Theorem 11. *Let $P(x)$ be a polynomial of degree δ in one variable modulo an integer M of possibly unknown factorization. In time polynomial in $(\log M, 2^{\delta})$, one can find all integers x_0 such that $P(x_0) \equiv 0 \pmod{M}$ and $|x_0| \leq M^{1/\delta}$.*

In time polynomial in $(\sum_{i=1}^{n} \log p_i, 2^m)$, we can thus find the solution N to noisy Chinese remaindering, provided that : $B^m \leq \prod_{i=1}^{n} p_i$. This condition is analogous to the condition $m < n/k$ we obtained by applying GS algorithm to the noisy polynomial interpolation problem. The method is mentioned in [6].

5.2 Lattice-based methods

Let $P = \prod_{i=1}^{n} p_i$. By analogy to the lattice-based method of section 3, we define interpolation numbers L_i in $\{0, \ldots, P-1\}$ by : $L_i \equiv 1 \pmod{p_i}$ and $L_i \equiv 0 \pmod{\prod_{j \neq i} p_j}$. The solution N of noisy Chinese remaindering satisfies:

$$N \equiv \sum_{i=1}^{n} (N \bmod p_i) L_i \pmod{P}.$$

We linearize the problem: letting $\delta_{i,j}$ equal to 1 if $N \equiv r_i \pmod{p_i}$, and 0 otherwise, one obtains

$$N \equiv \sum_{i=1}^{n} \sum_{j=1}^{m} \delta_{i,j} r_{i,j} L_i \pmod{P}.$$

This equation basically says that N is a small subset sum of the $r_{i,j}L_i$'s modulo P. It is thus natural to consider the $(nm+1)$-dimensional lattice L spanned by the rows of the following matrix:

$$\begin{pmatrix} P & 0 & \ldots & \ldots & 0 \\ r_{1,1}L_1 & B & 0 & \ldots & 0 \\ r_{1,2}L_1 & 0 & B & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ r_{n,m}L_n & 0 & \ldots & 0 & B \end{pmatrix}$$

The lattice L is the set of integer row vectors $(M, d_{1,1}B, d_{1,2}B, \ldots, d_{n,m}B) \in \mathbb{Z}^{nm+1}$ such that $M \equiv \sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} r_{i,j} L_i \pmod{P}$. It contains the *target vector* $(N, \delta_{1,1}B, \delta_{1,2}B, \ldots, \delta_{n,m}B)$, which has norm $\sqrt{N^2 + nB^2} \leq B\sqrt{n+1}$. Since the previous matrix is triangular, the volume of L is simply $P \times B^{nm}$. It follows that the target vector is expected to be the shortest vector of L when

$$B\sqrt{n+1} \ll (PB^{nm})^{1/(nm+1)} \approx P^{1/(nm+1)} B,$$

that is $\sqrt{n} \ll P^{1/(nm+1)}$. The condition should however be taken with care, as the lattice L cannot be considered as random. For instance, note that any sufficiently short linear relation between $r_{i,1}, r_{i,2}, \ldots, r_{i,m}$ gives rise to a shorter lattice point. It can be proved that such a case occurs when one of the p_i's is small or one of the $|S_i|$'s is big (using the notion of orthogonal lattice [27], see Appendix C). As with noisy polynomial interpolation, one can improve the lattice L by considering the sublattice Λ of points $(M, d_{1,1}B, d_{1,2}B, \ldots, d_{n,m}B) \in L$ such that, for all i_1 and i_2, $\sum_{j=1}^{m} d_{i_1,j} = \sum_{j=1}^{m} d_{i_2,j}$. However, the previous obstruction still holds (see details in Appendix C). Thus, the lattice-based approach is unlikely to be useful for elliptic curve point counting or integer factorization. Still, the reduction can be proved for certain choices of the parameters, for we have the following analog of lemma 6.

Lemma 12. *Let $\mathbf{d} = (M, d_{1,1}B, d_{1,2}B, \ldots, d_{n,m}B) \in \mathbb{Z}^{nm+1}$ satisfying* (1) *and shorter than the target vector. Assume that $B(m+1)\sqrt{n+1} < P/2$. Then:*

$$p(\mathbf{d}) \leq q^{-\min(t(\mathbf{d}), n-k)},$$

where $q = \min p_i$, k is the least positive integer such that $B(m+1)\sqrt{n+1} < \frac{q^k}{2}$, and $t(\mathbf{d}) = \left|\{1 \leq i \leq n : \exists j \in \{1, \ldots, m\} \setminus g(i) \text{ such that } d_{i,j} \not\equiv 0 \bmod p_i\}\right|$.

This lemma is useful, when none of the $|S_i|$'s are big and none of the p_i's are small (which is not the case arising in elliptic curve point counting or integer factorization) in which case one can obtain a provable reduction to the lattice shortest vector problem roughly similar to Theorem 7 since one can upper bound the probability that there exists a nonzero vector strictly shorter than the target vector. In particular, by taking all the p_i's of the same size (such as 32 bits), it is easy to build instances for which the lattice-based approach can experimentally solve noisy Chinese remaindering with a bound B much larger than with Coppersmith's method.

6 Conclusion

We presented various methods to solve the noisy polynomial interpolation problem. In particular, we proved the existence of a reduction from the noisy polynomial interpolation problem to the lattice shortest vector problem, for many choices of the parameters. This reduction appears to be very efficient in practice: experimental evidence suggest that many instances can be solved using standard lattice reduction algorithms. We therefore suggested simple modifications to several cryptographic schemes for which the security assumption relied on the computational hardness of noisy polynomial interpolation. We also briefly discussed analogous methods to solve the related noisy Chinese remaindering problem. The lattice-based approach is the best known method for certain choices of the parameters, but unfortunately not in applications such as elliptic curve point counting or integer factorization. There are several open problems, such as:

– Is there a better[3] reduction from noisy polynomial interpolation or Chinese remaindering to the lattice shortest vector problem ?
– Is there a lattice-based method to solve the polynomial reconstruction problem ?

Acknowledgments

We thank Amin Shokrollahi and Karin Melnick for their help and ideas with error correcting algorithms. We are grateful to Dan Boneh for several enlightening discussions, and for informing us on references [3, 15, 6]. We also thank Louis Granboulan and Andrew Odlyzko for their comments.

References

1. M. Ajtai. Generating hard instances of lattice problems. In *Proc. of 28th STOC*, pages 99–108. ACM, 1996. Available at [11] as TR96-007 report.
2. M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proc. 30th ACM STOC*, 1998. Available at [11] as TR97-047.
3. S. Ar, R. Lipton, R. Rubinfeld, and M. Sudan. Reconstructing algebraic functions from mixed data. *SIAM J. of Computing*, 28(2):488–511, 1999.
4. A. O. L. Atkin. The number of points on an elliptic curve modulo a prime. Email on the Number Theory mailing list, 1988.
5. A. O. L. Atkin. The number of points on an elliptic curve modulo a prime. Email on the Number Theory mailing list, 1991.
6. D. Boneh. Finding smooth integers in short intervals using CRT decoding. In *Proc. of 32nd STOC*. ACM, 2000.
7. J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, 1998. Third edition.
8. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. of Cryptology*, 10(4):233–260, 1997.
9. M.J. Coster, A. Joux, B.A. LaMacchia, A.M. Odlyzko, C.-P. Schnorr, and J. Stern. Improved low-density subset sum algorithms. *Comput. Complexity*, 2:111–128, 1992.
10. N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Proc. of Eurocrypt'2000*, LNCS. Springer-Verlag, 2000.
11. ECCC. `http://www.eccc.uni-trier.de/eccc/`. The Electronic Colloquium on Computational Complexity.
12. N. D. Elkies. Explicit isogenies. Draft, 1991.
13. C. Ellison, C. Hall, R. Milbert, and B. Schneier. Protecting secret keys with personal entropy. *Future Generation Computer Systems*, 16(4):311–318, 2000. Available at `http://www.counterpane.com/`.
14. N. Gilboa. Two party RSA key generation. In *Proc. of Crypto '99*, volume 1666 of *LNCS*, pages 116–129. Springer-Verlag, 1999.
15. O. Goldreich, D. Ron, and M. Sudan. Chinese remaindering with errors. In *Proc. of 31st STOC*. ACM, 1999. Also available at [11].

[3] holding for more or all choices of the parameters.

16. M. Gruber and C. G. Lekkerkerker. *Geometry of Numbers.* North-Holland, 1987.
17. V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Trans. on Information Theory*, 45(6):1757–1767, 1999. An extended abstract appeared in the Proc. of IEEE FOCS '98.
18. A. Joux and J. Stern. Lattice reduction: A toolbox for the cryptanalyst. *J. of Cryptology*, 11:161–185, 1998.
19. A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Proc. of Crypto '99*, volume 1666 of *LNCS*, pages 19–30. Springer-Verlag, 1999.
20. J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *Journal of the Association for Computing Machinery*, January 1985.
21. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.
22. R. Lercier and F. Morain. Counting the number of points on elliptic curves over finite fields: strategies and performances. In *Proc. of Eurocrypt '95*, volume 921 of *LNCS*, pages 79–94. Springer-Verlag, 1995.
23. J. E. Mazo and A. M. Odlyzko. Lattice points in high-dimensional spheres. *Monatsh. Math.*, 110:47–61, 1990.
24. D. Micciancio. The shortest vector problem is NP-hard to approximate within some constant. In *Proc. 39th IEEE FOCS*, 1998. Available at [11] as TR98-016.
25. F. Monrose, M. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. In *Proc. of 6th Conf. on Computer and Communications Security.* ACM, 1999.
26. M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In *Proc. of 31st STOC*, pages 245–254. ACM, 1999.
27. 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.
28. R. Peralta and E. Okamoto. Faster factoring of integers of a special form. *IEICE Trans. Fund. of Electronics, Communications, and Computer Sciences*, 79(4), 1996.
29. B. Pinkas, 1999. Private communication.
30. C.-P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
31. C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Programming*, 66:181–199, 1994.
32. 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.
33. R. Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7:219–254, 1995.
34. V. Shoup. Number Theory Library (NTL) version 3.7a. Can be obtained at `http://www.shoup.net/ntl/`.
35. C. L. Siegel. *Lectures on the Geometry of Numbers.* Springer-Verlag, 1989.

Appendix A: Proof of lemma 6

To simplify the notations of the proof, we assume without loss of generality that $g(i) = 1$ for all i. Since $\mathbf{d}$ satisfies (1), there exists $\lambda \in \mathbb{Z}$ such that, for all i :

$$\sum_{j=1}^{m} d_{i,j} = \lambda.$$

Let $\tilde{y}_{i,j} = y_{i,j} - f(x_i)$ for $2 \leq j \leq m$, and $h(X) = \sum_{i=1}^{n} \sum_{j=2}^{m} d_{ij} \tilde{y}_{ij} L_i(X)$. Since the $y_{i,j}$'s, for $j \geq 2$, are independent and uniformly distributed over $\mathbb{F}$, the same holds for the $\tilde{y}_{i,j}$'s, for all i and j. We have :

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} y_{i,j} L_i(X) &= \sum_{i=1}^{n} \sum_{j=2}^{m} d_{i,j} y_{i,j} L_i(X) + \sum_{i=1}^{n} d_{i,1} P(x_i) L_i(X) \\
&= \sum_{i=1}^{n} \sum_{j=2}^{m} d_{i,j} y_{i,j} L_i(X) + \sum_{i=1}^{n} \left(\lambda - \sum_{j=2}^{m} d_{i,j} \right) P(x_i) L_i(X) \\
&= \sum_{i=1}^{n} \sum_{j=2}^{m} d_{i,j} \left(y_{i,j} - P(x_i) \right) L_i(X) + \lambda \sum_{i=1}^{n} P(x_i) L_i(X) \\
&= h(X) + \lambda P(X)
\end{aligned}$$

Since $P(X)$ has degree k, it follows that:

$$p(\mathbf{d}) = \Pr \left(\deg \left(\sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} y_{i,j} L_i(X) \right) \leq k \right) = \Pr \left(\deg h(X) \leq k \right).$$

This new expression of $p(\mathbf{d})$ will be useful because it is independent of the polynomial $P(X)$. Note that $h(x_i) = \sum_{j=2}^{m} d_{ij} \tilde{y}_{ij}$. We will use two basic remarks :

1. Any nonzero polynomial of degree k has at most k roots over $\mathbb{F}$.
2. Any polynomial of degree at most $n-1$ can be uniquely expressed as a linear combination of the $L_i(X)$, since the L_i's are linearly independent.

We divide our analysis into two cases:

- Case 1: $t(\mathbf{d}) \leq n - k - 1$. From the expression of $h(x_i)$, it follows that $h(x_i) = 0$ for at least $k+1$ values x_i. Therefore, by remark 1, $\deg h(X) \leq k$ if and only if $h(X) = 0$. From remark 2, $h(X) = 0$ holds if and only if $\sum_{j=2}^{m} d_{ij} \tilde{y}_{ij} = 0$ for all $1 \leq i \leq n$. Among these n independent equations, only $t(\mathbf{d})$ are non trivial, each of which being satisfied with probability $1/q$. Hence, we proved that $p(\mathbf{d}) = q^{-t(\mathbf{d})}$.
- Case 2: $t(\mathbf{d}) \geq n-k$. There exists a set $S = \{(i_1, j_1), \ldots, (i_{n-k-1}, j_{n-k-1})\}$, such that the i_r's are pairwise distinct and $d_{i_r, j_r} \neq 0$ for $1 \leq r \leq n - k - 1$. Let the $\tilde{y}_{i,j}$'s be fixed for all $(i,j) \notin S$. Now consider the function ϕ that maps any $(\tilde{y}_{i,j})_{(i,j) \in S} \in \mathbb{F}^{n-k-1}$ to the $(n-k-1)$-tuple formed by the coefficients of degree $k+1, k+2, \ldots, n-1$ of the polynomial $h(X) = \sum_{i=1}^{n} \sum_{j=2}^{m} d_{ij} \tilde{y}_{ij} L_i(X)$. We notice that ϕ is one-to-one, by showing that ϕ is injective. Suppose indeed there at two tuples $(\tilde{y}_{i,j})_{(i,j) \in S}$ and $(\hat{y}_{i,j})_{(i,j) \in S}$ leading to two polynomials $\tilde{h}(X)$ and $\hat{h}(X)$ sharing the same $n-k-1$ high-order coefficients. Then $\deg \Delta(X) \leq k$, where:

$$\Delta(X) = \tilde{h}(X) - \hat{h}(X) = \sum_{(i,j) \in S} d_{i,j} (\tilde{y}_{i,j} - \hat{y}_{i,j}) L_i(X).$$

The polynomial $\Delta(X)$ is a linear combination of $n-k-1$ polynomials L_i's: there are $k+1$ distinct x_i's such that $\Delta(x_i)=0$. By remark 1, this implies $\Delta(X)=0$, and thus, by remark 2, $\tilde{y}_{i,j}-\hat{y}_{i,j}=0$ for all $(i,j)\in S$, since $d_{i,j}\not\equiv 0 \pmod q$ for $(i,j)\in S$. Hence, we proved that ϕ is one-to-one. In particular, there is a unique assignment of $(\hat{y}_{i,j})_{(i,j)\in S}$ such that $\deg h(X)\le k$. We therefore proved $p(\mathbf{d})=q^{-(n-k-1)}$.

Appendix B: Estimating the number of short vectors

In order to compute the expected number of short vectors, we have to find the number of vectors $\mathbf{d}$ for a given $t(\mathbf{d})$ and given maximal norm. We do not have an asymptotic formula for this number, but we have found recurrence relations that allowed us to find the exact number for the necessary cases.

Let w, r, λ, n, m be non-negative integers. Then we denote by $T(r,\lambda,m)$ the number of vectors $\mathbf{d}=(d_1,\ldots,d_m)\in\mathbb{Z}^m$ satisfying $\sum_{j=1}^m d_j=\lambda$ and $\|\mathbf{d}\|=\sqrt{r}$. Denote by $R(w,r,\lambda,n,m)$ the number of vectors $\mathbf{d}=(d_{1,1},\ldots,d_{n,m})\in\mathbb{Z}^{mn}$ with $t(\mathbf{d})=w$ and norm $\|\mathbf{d}\|\le\sqrt{r}$ and $\sum_{j=1}^m d_{i,j}=\lambda$.

We used the following recurrence relations for computing T and R.

$$T(r,\lambda,m)=\begin{cases}1 & \text{if } m=0 \text{ and } r=0 \text{ and } \lambda=0,\\ 0 & \text{if } m=0 \text{ and } (r\neq 0 \text{ or } \lambda\neq 0),\\ \sum_{j=-\lfloor\sqrt{r}\rfloor}^{\lfloor\sqrt{r}\rfloor} T(r-j^2,\lambda-j,m-1) & \text{if } m>0.\end{cases}$$

$$S(w,r,\lambda,m)=\begin{cases}1 & \text{if } w=0 \text{ and } r\ge 0,\\ 0 & \text{if } w=0 \text{ and } r<0,\\ \sum_{j=1}^{r}(T(j,\lambda,m)-\epsilon_{j,\lambda})S(w-1,r-j,\lambda,m) & \text{if } w>0,\end{cases}$$

where $\epsilon_{j,\lambda}=1$ if $j^2=\lambda$ and 0 otherwise. Here, $S(w,r,\lambda,m)$ denotes the number of vectors in $\mathbf{d}=(d_{1,1},\ldots,d_{w,m}),\in\mathbb{Z}^{wm}$, such that $\sum_{j=1}^m d_{i,j}=\lambda$ for all $1\le i\le w$, $\|\mathbf{d}\|\le\sqrt{r}$ and for every $1\le i\le w$ there exists j such that $d_{i,j}\neq 0$. The latest condition ensures that $t(\mathbf{d})=w$, it is also the reason for the correction $\epsilon_{j,\lambda}$ above.

In order to compute $R(w,r,\lambda,n,m)$ we note there must be exactly $(n-w)$ values for i, such that $d_{i,g(i)}=\lambda$ and $d_{i,j}=0$ for $j\in\{1,\ldots,m\}\setminus\{g(i)\}$, where $g(i)$ is a function defined by $y_{i,g(i)}=P(x_i)$. This leads to the equation

$$R(w,r,\lambda,n,m)=\binom{n}{w}S(w,r-(n-w)\lambda^2,\lambda,m).$$

Now, we can compute the expected number of non-target vectors (i.e. a vector with $t(\mathbf{d}) > 0$) with norm $\leq \sqrt{r}$ as

$$\begin{aligned} E(r,n,m) &= \sum_{\mathbf{d}:\|\mathbf{d}\|\leq\sqrt{r}} q^{-\min(t(\mathbf{d}),n-k-1)} \\ &= \sum_{\lambda=-\lfloor r/n\rfloor}^{\lfloor r/n\rfloor} \sum_{w=1}^{n} R(w,r,\lambda,n,m) q^{-\min(w,n-k-1)}. \end{aligned}$$

Similarly the exact number $N(n,r)$ of vectors of dimension n and norm at most $\sqrt{r}$ can, of course, be computed by

$$N(n,r) = \begin{cases} 1 & \text{if } n = 0 \text{ and } r \geq 0, \\ 0 & \text{if } n = 0 \text{ and } r < 0, \\ \sum_{j=-\lfloor\sqrt{r}\rfloor}^{\lfloor\sqrt{r}\rfloor} N(n-1, r-j^2) & \text{if } n > 0. \end{cases}$$

Appendix C: The lattice-based method for noisy Chinese remaindering

In this appendix, we give some conditions for which the lattice-based method for noisy Chinese remaindering must fail, that is, for which the lattice contains at least one nonzero vector shorter than the target vector.

We first consider the simple case of the original lattice L. We note that any short linear relation between $r_{i,1}, r_{i,2}, \ldots, r_{i,m}$ give rise to a short lattice point, by adding zero entries and multiplying by B. To predict the size of those linear relations, we use the notion of orthogonal lattice [27]. For any lattice Λ in $\mathbb{Z}^m$, the lattice $\Lambda^\perp$ is the set of integer row vectors $\mathbf{x}$ such that $\mathbf{x}$ is orthogonal to $\mathbf{y}$ for all $\mathbf{y} \in \Lambda$. It is known that $\det(\Lambda^\perp) \leq \det(\Lambda)$ and $\dim \Lambda + \dim \Lambda^\perp = n$. If we let $\mathbf{r}_i = (r_{i,1}, r_{i,2}, \ldots, r_{i,m})$, then the one-dimensional lattice spanned by $\mathbf{r}_i$ has volume less than $p_i\sqrt{m}$. It follows that:

$$\lambda_1(\mathbf{r_i}^\perp) \leq \sqrt{\gamma_{m-1}} \left(p_i\sqrt{m}\right)^{1/(m-1)}.$$

Therefore, there is at least one nonzero lattice point of L shorter than the target vector if:

$$\sqrt{\gamma_{m-1}} \left(p_i\sqrt{m}\right)^{1/(m-1)} < \sqrt{n+1}.$$

One of these n inequalities is satisfied when one of the p_i's is sufficiently small, or m is sufficiently big (or equivalently for the general noisy Chinese remaindering problem, when one of the $|S_i|$'s is large).

We now consider the sublattice Λ of L. To see that the previous obstruction still holds, it suffices to consider vectors which are orthogonal to both one of the $\mathbf{r}_i$'s, and the vector $(1, 1, \ldots, 1) \in \mathbb{Z}^m$. By adding zero entries to those vectors, and multiplying by B, one obtains a lattice point of Λ. The lattice spanned by

$\mathbf{r}_i$ and $(1,1,\ldots,1)$ has volume less than $p_i m$. It follows that there exists at least one nonzero lattice point of Λ shorter than the target vector if for some i:

$$\sqrt{\gamma_{m-2}}\,(p_i m)^{1/(m-2)} < \sqrt{n+1}.$$

Again, one of these n inequalities is satisfied when one of the p_i's is sufficiently small, or m is sufficiently big.

Appendix D: Proof of Lemma 8.

Proof of Equation (2) The number $N(n,r)$ of n-dimensional vectors of norm $\leq \sqrt{r}$ satisfies the following inequality[4]

$$N(n,r) \leq 2^n \binom{n+r}{r}.$$

Let $\mathbf{d}$ be a vector satisfying $\sum_{j=1}^{m} d_{i,j} = \lambda$ for some integer λ. Then the norm of $\mathbf{d}$ is at least $\sqrt{|\lambda n|}$ and hence $r = n$ implies $\lambda \in \{-1,0,1\}$. We will represent $E(n,n,2)$ as a sum of 4 summands depending on the value of λ and whether $t(\mathbf{d}) \leq n-1-k$. In particular, we let

$$\begin{aligned}
E(n,n,2) &= A + B + C + D, \quad \text{where} \\
A &= \sum_{w=1}^{n-2-k} R(w,n,0,n,2)q^{-w} \\
B &= \sum_{w=n-1-k}^{n} R(w,n,0,n,2)q^{-n+1+k} \\
C &= \sum_{\lambda\in\{1,-1\}} \sum_{w=1}^{n-2-k} R(w,n,\lambda,n,2)q^{-w} \\
D &= \sum_{\lambda\in\{1,-1\}} \sum_{w=n-1-k}^{n} R(w,n,\lambda,n,2)q^{-n+1+k}
\end{aligned}$$

If $\lambda = 0$ then $d_{i,2} = -d_{i,1}$ and

$$\|\mathbf{d}\| = \sqrt{2\sum_{i=1}^{n} d_{i,1}^2}.$$

Hence there is a one-to-one mapping between the $2n$-dimensional vectors $\mathbf{d} = (d_{1,1},\ldots,d_{n,2})$ satisfying $d_{i,1}+d_{i,2} = 0$ of norm $\leq \sqrt{n}$ and the n-dimensional vectors $(d_{1,1},\ldots,d_{n,1})$ of norm $\leq \sqrt{n/2}$. It follows that the number $R(w,n,0,n,2)$

[4] This is quite a bad estimate. I'll prove it if nothing better can be found. The estimate is however, sufficient for this proof.

of vectors $\mathbf{d}$ with $d_{i,1} + d_{i,2} = 0$ and $t(\mathbf{d}) = w$ and norm $\|\mathbf{d}\| \le n$ satisfies

$$R(w,n,0,n,2) \le \binom{n}{w} N(w, \lfloor n/2 \rfloor).$$

And, hence we get

$$\begin{aligned} A = \sum_{w=1}^{n-k-2} R(w,n,0,n,2) q^{-w} &\le \sum_{w=1}^{n-k-2} \binom{n}{w} N(w, \lfloor n/2 \rfloor) q^{-w} \\ &\le \sum_{w=1}^{n-k-2} \binom{n}{w} \binom{\lfloor n/2 \rfloor + w}{w} 2^w q^{-w} \le \sum_{w=1}^{n-k-2} \frac{n^w}{w!} (\lfloor n/2 \rfloor)^w 2^w q^{-w} \\ &\le \sum_{w=1}^{\infty} \frac{(n^2/q)^w}{w!} = e^{n^2/q} - 1 \le 2n^2/q. \end{aligned}$$

Moreover, we also have

$$\sum_{w=1}^{n} R(w,n,0,n,2) = N(n, \lfloor n/2 \rfloor) - 1.$$

Thus it follows

$$B \le N(n, \lfloor n/2 \rfloor) q^{-n+1+k}.$$

Finally, we also note that

$$A + B \ge \sum_{w=1}^{n} R(w,n,0,n,2) q^{-n+1+k} = (N(n, \lfloor n/2 \rfloor) - 1) q^{-n+1+k}.$$

If $\lambda = \pm 1$ then for all $1 \le i \le n$ either $d_{i,1} = \lambda$ and $d_{i,2} = 0$ or $d_{i,1} = \lambda$ and $d_{i,2} = 0$. Hence, we have

$$R(w,n,\lambda,n,2) = \binom{n}{w}.$$

Therefore,

$$C = 2 \sum_{w=1}^{n-2-k} \binom{n}{w} q^{-w} \le 2\left(1 + q^{-1}\right)^n - 2 \le 2e^{n/q} - 2 \le 4n/q.$$

Also, we have

$$D \le 2 \sum_{w=1}^{n} R(w,n,\lambda,n,2) q^{n-1-k} \le 2^{n+1} q^{n-1-k}.$$

And finally, we get

$$C + D \ge 2 \sum_{w=1}^{n} R(w,n,\lambda,n,2) q^{n-1-k} = (2^{n+1} - 2) q^{n-1-k}.$$

Adding the results for $\lambda = 0$ and $\lambda = \pm 1$ completes the proof.

Proof of Equation (3). According to [23, Lemma 1] (with $x = e^{-s}$ and $\alpha = 1$) we have

$$N(n,r) \le x^{-r} h(x)^n, \text{ for all } 0 < x < 1$$

where $h(x) = 1 + 2\sum_{k=1}^{\infty} x^{k^2}$.

$N(nm,n)q^{-n+1+k}$ is a trivial upper bound for the expected number of short vectors with $t(\mathbf{d}) \ge n-1-k$. Hence, it remains to approximate the expected number of vectors with $t(\mathbf{d}) < n-1-k$. For any $s > 0$ this number is given by

$$\begin{aligned}
&\sum_{w=1}^{n-1-k} \sum_{\lambda \in \{-1,0,1\}} R(w,n,\lambda,n,m) q^{-w} \\
&\le 3 \sum_{w=1}^{n-1-k} \binom{n}{w} N(wm,n) q^{-w} \\
&\le 3 \sum_{w=1}^{n-1-k} \binom{n}{w} e^{sn} h(s)^{wm} q^{-w} \\
&\le 3e^{sn} \left((1 + h(s)^m/q)^n - 1 \right)
\end{aligned}$$

Appendix E: Proof of Lemma 12

The proof is analogous to the one of Lemma 6, with some slight technical differences. To simplify the notations of the proof, we assume without loss of generality that $g(i) = 1$ for all i. Since $\mathbf{d}$ satisfies (1), there exists $\lambda \in \mathbb{Z}$ such that, for all i :

$$\sum_{j=1}^{m} d_{i,j} = \lambda.$$

Let $\tilde{r}_{i,j} = (r_{i,j} - N) \bmod p_i$ for $2 \le j \le m$, and h be the smallest residue (in absolute value) of $\sum_{i=1}^{n} \sum_{j=2}^{m} d_{ij} \tilde{r}_{ij} L_i) \bmod P$. Since the $r_{i,j}$'s, for $j \ge 2$, are independent and uniformly distributed over $\mathbb{Z}_{p_i}$, the same holds for the $\tilde{r}_{i,j}$'s, for all i and j. We have :

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} r_{i,j} L_i &= \sum_{i=1}^{n} \sum_{j=2}^{m} d_{i,j} r_{i,j} L_i + \sum_{i=1}^{n} d_{i,1} (N \bmod p_i) L_i \\
&= \sum_{i=1}^{n} \sum_{j=2}^{m} d_{i,j} r_{i,j} L_i + \sum_{i=1}^{n} \left(\lambda - \sum_{j=2}^{m} d_{i,j} \right) (N \bmod p_i) L_i \\
&= \sum_{i=1}^{n} \sum_{j=2}^{m} d_{i,j} \left(r_{i,j} - N \bmod p_i \right) L_i + \lambda \sum_{i=1}^{n} (N \bmod p_i) L_i \\
&\equiv h + \lambda N \pmod{P}
\end{aligned}$$

Since $\mathbf{d}$ is shorter than the target vector: $\|\mathbf{d}\| \leq B\sqrt{n+1}$. Therefore $M \leq B\sqrt{n+1}$ and $|d_{i,j}| \leq \sqrt{n+1}$, so that $|\lambda| \leq m\sqrt{n+1}$. If $\mathbf{d}$ is in the sublattice then $M \equiv \sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} r_{i,j} L_i \pmod{P}$ so that $M \equiv h + \lambda N \pmod{P}$. Notice that $-P/2 < h \leq P/2$ and $-P/2 < M - \lambda N \leq P/2$ as $|M - \lambda N| \leq B(m+1)\sqrt{n+1} < P/2$. Therefore $M = h + \lambda N$ holds over $\mathbb{Z}$, which implies $|h| \leq B(m+1)\sqrt{n+1}$. We therefore proved:

$$p(\mathbf{d}) \leq \Pr\left(|h| \leq B(m+1)\sqrt{n+1}\right).$$

Note that $h \equiv \sum_{j=2}^{m} d_{ij} \tilde{r}_{ij} \pmod{p_i}$. We will use one basic remark : If $0 \leq |u| < q^k$ and $u \equiv 0 \pmod{p_i}$ for at least k values of i, then u is zero. We divide our analysis into two cases:

- Case 1: $t(\mathbf{d}) \leq n - k$. From the expression of $h \bmod p_i$, it follows that $h \equiv 0 \bmod p_i$ for at least k values of i. Since $|h| \leq B(m+1)\sqrt{n+1} < q^k$, we therefore have $h = 0$, thus $\sum_{j=2}^{m} d_{ij} \tilde{r}_{ij} \equiv 0 \pmod{p_i}$ for all $1 \leq i \leq n$. Among these n independent equations, only $t(\mathbf{d})$ are non trivial, each of which being satisfied with probability less than $1/q$. Hence, we proved that $p(\mathbf{d}) \leq q^{-t(\mathbf{d})}$.
- Case 2: $t(\mathbf{d}) > n - k$. There exists a set $S = \{(i_1, j_1), \ldots, (i_{n-k}, j_{n-k})\}$, such that the i_r's are pairwise distinct and $d_{i_r,j_r} \neq 0$ for $1 \leq r \leq n-k$. Let the $\tilde{r}_{i,j}$'s be fixed for all $(i,j) \notin S$. Now suppose there at two tuples $(\tilde{r}_{i,j})_{(i,j)\in S}$ and $(\hat{r}_{i,j})_{(i,j)\in S}$ leading to two numbers $\tilde{h}$ and $\hat{h}$ less than $B(m+1)\sqrt{n+1}$ in absolute value. Then $|\Delta| < q^k$ where:

  $$\Delta = \tilde{h} - \hat{h} \equiv \sum_{(i,j)\in S} d_{i,j}(\tilde{r}_{i,j} - \hat{r}_{i,j}) L_i \pmod{P}.$$

  There are at least k distinct i's such that $\Delta \equiv 0 \pmod{p_i}$, which implies $\Delta = 0$, and thus, $\tilde{r}_{i,j} - \hat{r}_{i,j} \equiv 0 \pmod{p_i}$ for all $(i,j) \in S$, since $d_{i,j} \not\equiv 0 \pmod{p_i}$ for $(i,j) \in S$. Hence, we proved that there is a unique assignment of $(\hat{r}_{i,j})_{(i,j)\in S}$ such that $|h| \leq B(m+1)\sqrt{n+1}$. We therefore proved $p(\mathbf{d}) \leq q^{-(n-k)}$.