The security of all bits using list decoding

Paz Morillo and Carla Ràfols

Dept. Matemàtica Aplicada IV
Universitat Politècnica de Catalunya
C. Jordi Girona 1-3, E-08034 Barcelona, Spain
e-mail: {paz,crafols}@ma4.upc.edu

Abstract. The relation between list decoding and hard-core predicates has provided a clean and easy methodology to prove the hardness of certain predicates. So far this methodology has only been used to prove that the $O(\log \log N)$ least and most significant bits of any function with multiplicative access —which include the most common number theoretic trapdoor permutations— are secure. In this paper we show that the method applies to all bits of any function defined on a cyclic group of order N with multiplicative access for cryptographically interesting N. As a result, in this paper we reprove the security of all bits of RSA, the discrete logarithm in a group of prime order or the Paillier encryption scheme.

Keywords: bit security, list decoding, one-way function

1 Introduction

One-way functions are one of the most fundamental cryptographic primitives and it is not an overstatement to say that they are behind most of modern cryptography. If some reasonable computational assumptions hold, a one-way function is easy to compute but hard to invert. In some cases, this security requirement may not be enough: in particular, the definition of one-way function does not say anything about how much information it can leak. A predicate of the preimage, P, is a *hard-core* of f if f does not give away any information about P, that is, if there exists a polynomial time reduction from guessing P to inverting f.

The study of hard-core predicates is of interest for various reasons, not only because it strengthens our understanding of the real hardness of the considered one-way function, but also because of its applications, which include the construction of secure bit commitment schemes or cryptographically strong pseudorandom generators. Further, the study of bit security has led to important techniques and insights which have found other applications. For instance, the study of the security of the least significant bit of RSA led to the two-point based sampling technique introduced in [2], later used to prove the well known result of the Goldreich and Levin bit — GL from now on — which states that every one-way function has a hard-core bit. We emphasize that the importance

of the GL result reaches far beyond the domain of bit security, and many works in other lines of research are in some way indebted to it, for instance in learning theory [4],[7].

Many bit security results have very technical and sophisticated proofs. Although many proofs for different one-way functions have a similar structure, they have to be adapted to each particular case. In contrast, Akavia, Goldwasser and Safra [1] give a very elegant and general methodology to prove bit security results. In particular, they show how this methodology applies to prove the security of $O(\log \log N)$ least and most significant bits of any function with *multiplicative access* - such as RSA and DL, for instance.

Akavia *et al.* raised the question whether this methodology applies to prove the security of internal bits, a question which we answer in the affirmative in this paper. Since the existing security proofs for the hardness of internal bits of RSA and DL are particularly technical and cumbersome to follow in all detail — we refer the reader to [8] for an example— , we feel that a more readable proof should contribute much to the public discussion of the results and thus also to their credit and diffusion.

1.1 Previous work

The GL result, which gives a hard-core predicate for any one-way function, can be reinterpreted as a list decoding algorithm for the Hadamard code. This suggested the possibility of a general methodology to prove bit security results. This methodology was formalized by Akavia *et al.* in 2003, where it was used to prove (or often *reprove* known results) the hardness of certain predicates for one way functions defined on a cyclic group $\mathbb{G}$ of order N and having the property of multiplicative access, that is, functions f for which given $f(x)$, $f(x \cdot y)$ can be computed for any known y in almost all of the cases.

The most common number theoretic trapdoor permutations in cryptography, such as $RSA_{N,e}(x) = x^e \mod N$, $Rabin(x) = x^2 \mod N$, $EXP_{p,g}(x) = g^x \mod p$ and $ECL_{p,a,b,Q}(x) = xQ$ — exponentiation in the group of $\mathbb{F}_p$- rational points of an elliptic curve $E_{p,a,b}(\mathbb{F}_p)$ — have this property.

A part from the formalization of the list decoding methodology, one of the key contributions of Akavia *et al.* is to give a learning algorithm for functions $f : \mathbb{Z}_N \to \mathbb{C}$, which is a necessary piece to provide the aforementioned results.

The security of the internal bits had already been proved for some one-way functions with multiplicative access such as RSA and the discrete logarithm in [8] and the Paillier encryption scheme in [3].

1.2 Organization

Sections 2, 3 and 4 are introductory: in section 2 we define the concept of hard-core predicate and give some of the results of Akavia *et al.* Sections 3 and 4 are devoted to basics of Fourier analysis and list decodable codes and to the relation between hard-core predicates and list decoding. We stress that many

of the results and definitions given in these sections are taken from the work of Akavia *et al.* but are necessary to introduce our contributions. In section 5 we prove one of our main results concerning the security of all bits of any one-way function with multiplicative access for special N, while in section 6 we prove it for all N of cryptographic interest. Next, in section 7 we prove the security of all bits of the Paillier encryption scheme. Possible extensions of these results are discussed in section 8. In section 9 we summarize our contribution.

2 Parts that are as hard as the whole

Informally, a hard-core bit for a one-way function $f : \mathcal{D} \to R$ is a boolean predicate $P : \mathcal{D} \to \{\pm 1\}$ which does not leak from f. Obviously, we cannot prevent an adversary from taking a random guess, but the point is that there should not be any strategy to predict P which works *significantly better* than the random one. Define

Definition 1. $maj_P \stackrel{\textbf{def}}{=} max_{b\in\{\pm 1\}} Pr\big(P(x) = b : x \leftarrow D\big)$ *and* $minor_P \stackrel{\textbf{def}}{=} 1 - maj_P$.

We write $x \leftarrow D$ to indicate that we choose an element x in D according to the uniform distribution.

Definition 2. *A function $\nu(\cdot)$ is negligible if for any constant $c \geq 0$ there exists $n_0 \in \mathbb{Z}$, s.t $\nu(n) < n^{-c}$ for all integers $n \geq n_0$.*

This definition of hard-core predicate is taken from [1].

Definition 3. *For each $n \in \mathbb{N}$, let I_n be a countable index set, and set $I = (I_n)_{n\in\mathbb{N}}$. Let $F = (f_i : D_i \to R_i)_{i\in I}$ be a family of one-way functions and $\mathcal{P} = (P_i : D_i \to \{\pm 1\})_{i\in I}$ a family of Boolean predicates, where w.l.o.g. if $i \in I_n$ $D_i \subset \{0,1\}^n$. We will say that $\mathcal{P}$ is a family of* hard-core *predicates for F if and only if, for all $n \in \mathbb{N}$ and $i \in I_n$:*

- *P_i can be computed by means of a Monte-Carlo algorithm $\mathcal{A}_1(i, x)$.*
- *$P_i(x)$ is not computable by means of any efficient algorithm from $f_i(x)$; that is, for any PPT algorithm $\mathcal{A}_2$,*

$$Pr(\mathcal{A}_2(i, f_i(x)) = P_i(x) : x \leftarrow D_i) \leq maj_{P_i} + \nu(n),$$

where $\nu(\cdot)$ is a negligible function.

While there are predicates which are a hard-core of any one-way function, like the GL bit [6] or all the bits of $ax + b \mod p$ [9], there are also many results concerning the security of a certain bit of the binary representation of the preimage for a specific one-way function (e.g. the least significant bit of RSA or Rabin, see for instance [2]).

Given an element $x \in \mathbb{Z}_N$, define $[x]$ as the representative of the class of x in $[0, N)$ and $abs_N(x) = min\{[x], N - [x]\}$. The i-th bit of an element $x \in \mathbb{Z}_N$

is defined as $B_i(x) = 1$ if the i-th bit of the binary representation of $[x]$ is 0 and as -1 otherwise.

Akavia *et al.* prove the security of any basic t-segment predicate, $t \in poly(n)$, for any one-way function with multiplicative access having domain in $\mathbb{Z}_N$, where $n \stackrel{\text{def}}{=} \lfloor \log N \rfloor$.

Definition 4. *A predicate $P_N : \mathbb{Z}_N \to \{\pm 1\}$ is said to be a basic t-segment predicate if there are at most t values of $x \in \mathbb{Z}_N$ for which $P_N(x+1) \neq P_N(x)$.*

In particular, their result implies that the predicate B_{n-i}, where $i \in poly(\log n)$, is a hard-core of any one-way function with multiplicative access, since trivially B_{n-i} is a basic t-segment predicate, where $t = 2^{i+1}$.

Further, there is a correspondence between B_i, where $i \in O(\log n)$ and some t-basic segment predicate with $t \in poly(n)$. For instance, it is easy to verify that $lsb_N(x) = half_N(\frac{x}{2})$, where $half_N(x)$ is a basic 2-segment predicate which is equal to 1 if $[x] \leq N/2$ and is -1 otherwise. This correspondence allows to prove that the predicates B_i, where $i \in O(\log n)$ are also hard-core of any one-way function with multiplicative access when N is odd (see [1] for details).

3 Preliminaries

Before sketching the list decoding methodology of [1], we begin with some basic concepts.

3.1 Fourier Analysis in $\mathbb{Z}_N$

In the space of functions from $\mathbb{Z}_N$ to $\mathbb{C}$ it is possible to define the inner product

$$\langle g, h \rangle \stackrel{\text{def}}{=} \frac{1}{N} \sum_{x \in \mathbb{Z}_N} g(x)\overline{h(x)}.$$

For each $\alpha \in \mathbb{Z}_N$, the α-character is defined as a function $\chi_\alpha : \mathbb{Z}_N \to \mathbb{C}$ such that $\chi_\alpha(x) = w_N^{\alpha x}$, where $w_N \stackrel{\text{def}}{=} e^{\frac{2\pi j}{N}}$. It is easy to check that $B_\alpha \stackrel{\text{def}}{=} \{\chi_\alpha : \alpha \in \mathbb{Z}_N\}$ is an orthonormal basis of the space of functions going from $\mathbb{Z}_N$ to $\mathbb{C}$.

If Γ is a subset of $\mathbb{Z}_N$, it is natural to consider the projection of g in the set of Γ characters, that is,

$$g_{|\Gamma} = \sum_{\alpha \in \Gamma} \widehat{g(\alpha)}\chi_\alpha,$$

where $\widehat{g(\alpha)} = \langle g, \chi_\alpha \rangle$ are the *Fourier coefficients*. Observe that, if $h(y) = g(ay)$ for some $a \in \mathbb{Z}_N^*$, then $\widehat{h(\alpha)} = \widehat{g(\alpha/a)}$.

Because B_α is an orthonormal basis,

$$||g||_2^2 = \sum_{\alpha \in \mathbb{Z}_N} |\widehat{g(\alpha)}|^2 \text{ and } \; ||g_{|\Gamma}||_2^2 = \sum_{\alpha \in \Gamma} |\widehat{g(\alpha)}|^2.$$

Finally, define

Definition 5. *(Fourier Concentrated) A function $g : \mathbb{Z}_N \to \mathbb{C}$ is Fourier concentrated if for every $\epsilon > 0$ there exists a set Γ consisting of poly(n/ϵ) characters, so that*

$$||g - g_{|\Gamma}||_2^2 = \sum_{\alpha \notin \Gamma} |\widehat{g(\alpha)}|^2 \leq \epsilon.$$

In the following, this condition will be referred to as g is ϵ-concentrated on the set Γ.

The *heavy characters* of g are the characters for which the projection of g has a greater modulus, that is, given $\tau > 0$ and $g : \mathbb{Z}_N \to \mathbb{C}$, define

$$Heavy_\tau(g) \stackrel{\text{def}}{=} \{\chi_\alpha : |\widehat{g(\alpha)}|^2 \ \geq \tau\}.$$

3.2 Codes

A binary code is a subset of $\{\pm 1\}^*$. To encode the elements of $\mathbb{Z}_N$ we will limit ourselves to codewords of length N, in this case the code is a subset $\mathcal{C} \subset \{\pm 1\}^N$. Each codeword C_x can be seen as a function $C_x : \mathbb{Z}_N \to \{\pm 1\}$, expressed as $(C_x(0), C_x(1), \ldots, C_x(N-1))$.

Definition 6. *The normalized Hamming distance between two functions $g, h : \mathbb{Z}_N \to \{\pm 1\}$ is $\Delta(g,h) = Pr\big(g(x) \neq h(x) : x \leftarrow \mathbb{Z}_N\big)$.*

The next definition is a natural extension of the concept of error correcting codes.

Definition 7. *A code $\mathcal{C} = \{C_x : \mathbb{Z}_N \to \{\pm 1\}\}$ is list decodable if there exists a PPT algorithm which given access to a corrupted codeword w and inputs $\delta, \epsilon, 1^n$ returns a list $L \supseteq \{x : \Delta(C_x, w) \leq minor_{C_x} - \epsilon\}$ with probability $1 - \delta$.*

Remark 1. In this definition it says "given access to w" because in our examples it will be computationally infeasible to read the whole word w due to its size.

3.3 List decodable codes

In this section we give sufficient conditions for a code to be list decodable, for a detailed explanation we refer the reader to [1].

Definition 8. *A code $\mathcal{C}$ is concentrated if each of its codewords C_x is Fourier concentrated.*

Definition 9. *A code $\mathcal{C}$ is recoverable, if there exists a recovery algorithm, namely, a polynomial time algorithm that, given a character χ_α (for $\alpha \neq 0$), a threshold τ and 1^n, where $n = \lfloor \log N \rfloor$ returns a list L_α containing*

$$\{x \in \mathbb{Z}_N : \chi_\alpha \in Heavy_\tau(C_x)\}.$$

One of the main contributions of Akavia *et al.* is to prove that on input a threshold τ and given access to any function $g : \mathbb{G} \to \mathbb{C}$ where $\mathbb{G}$ is any abelian group with known generators of known orders, it is computationally feasible to obtain a list of all the Fourier coefficients in $Heavy_\tau(g)$. In particular, in the $\mathbb{Z}_N$ case — which is enough for our purposes — they prove that

Theorem 1. *There is an algorithm which, given query access to $g : \mathbb{Z}_N \to \{\pm 1\}$, $0 < \tau$ and $0 < \delta < 1$, outputs a list L, of $O(1/\tau)$ characters s.t. $Heavy_\tau(g) \subset L$ with probability at least $1 - \delta$; and the running time of the algorithm is $\widetilde{O}(n \cdot \ln^2(1/\delta)/\tau^{5.5})$, where the $\widetilde{O}()$ notation indicates that terms with complexity polynomial in $\log(1/\tau), \log n$ or $\ln\ln(1/\delta)$ have been eliminated.*

Another algorithm to the same purpose was given by Strauss and Mutukrishnan [5], resulting in a running time with improved dependence in $1/\tau$.

This theorem is used in [1] to prove the following

Theorem 2. *Let $\mathcal{C} = \{C_x : \mathbb{Z}_N \to \{\pm 1\}\}$ be a concentrated and recoverable code, then $\mathcal{C}$ is list decodable.*

The intuition behind the theorem is the following. Suppose that we have access to a corrupted word w which is close enough to a codeword C_x, then:

- Because of the concentration of the code and the closeness of w and C_x, there exists an explicit threshold τ — non-negligible in n— such that χ_β is a τ-heavy coefficient of both w and C_x, that is, there exists a $\beta \in \mathbb{Z}_N$, $\beta \neq 0$, such that
$$\chi_\beta \in Heavy_\tau(w) \cap Heavy_\tau(C_x).$$
This is proven in the *Concentration and agreement lemma* of [1].
- Because of theorem 1, on input this threshold τ, we can recover a list L with all the Fourier coefficients in $Heavy_\tau(w)$ with probability $1-\delta$. We emphasize that if δ is non-negligible in n, both the running time of the algorithm and the length of the list — which is $1/\tau$— is polynomial in n.
- For each of these coefficients χ_β the recovery algorithm will output a list of codewords and C_x will be in at least one of those lists.

4 The relation between list decoding and hard-core predicates

In this section we summarize the connection between list decoding and hard-core predicates from [1].

Suppose we want to prove that $P : \mathbb{Z}_N \to \{\pm 1\}$ is a hard-core of $f : \mathbb{Z}_N \to R$. As it is standard in cryptography, the security of P is proved by a *reduction argument*. The proof consists in trying to invert f (recover x) given a challenge $f(x)$ and assuming we have access to an oracle predicting $P(y)$ from $f(y)$ with non-negligible advantage over a random guess.

When f has multiplicative access, the connection between list decoding and hard-core predicates comes from encoding each element $x \in \mathbb{Z}_N$ as $C_x^P = (C_x^P(0), C_x^P(1), \ldots, C_x^P(N-1))$, where $C_x^P(j) = P(jx)$. This is the so-called *multiplication code*. An oracle predicting $P(y)$ from $f(y)$ without errors would give us access to C_x^P, but since the oracle gives incorrect answers we have access to a corrupted codeword w instead. If the code is list decodable we can find a list of codewords containing C_x^P, thus inverting f.

Now, in general, for any function f — not necessarily with multiplicative access— to prove that P is a hard-core predicate of f following the list decoding methodology, it would suffice to somehow encode the elements of $\mathbb{Z}_N$ in such a way that,

- The code is concentrated and recoverable (that is, list decodable).
- Given the challenge $f(x)$ and an oracle predicting P we can devise access to a corrupted codeword w close enough to the encoding of x.

This is formalized in Theorem 2 (List Decoding Approach) of [1],

Theorem 3. *Assume a collection of codes $\mathcal{C}^P = \{\mathcal{C}^{P_i}\}_{i \in I}$ s.t. $\forall i \in I$, (1) $\mathcal{C}^{P_i}$ is list decodable, and (2) $\mathcal{C}^{P_i}$ accessible with respect to f_i. Then $\mathcal{P}$ is hard-core of F.*

The definition of *accessible code* is:

Definition 10. *Let $\mathcal{P}$ be a collection of predicates and F a family of one-way functions. The code $\mathcal{C}$ is accessible with respect to F if there exists a PPT access algorithm $\mathcal{A}$, such that for all $i \in I_n$, $\mathcal{C}^{P_i}$ is accessible with respect to f_i, namely*

1. ***Code access:*** *$\forall x, j \in \mathcal{D}_i$, $\mathcal{A}(i, f_i(x), j)$ returns $f_i(x')$ such that $C_x^{P_i}(j) = P_i(x')$*
2. ***Well spread:*** *For uniformly distributed $C_x^{P_i} \in \mathcal{C}^{P_i}$ and $j \in D_i$, the distribution of x' satisfying $f_i(x') = \mathcal{A}(i, f_i(x), j)$ is statistically close to the uniform distribution on D_i*
3. ***Bias preserving:*** *For every codeword $C_x^{P_i} \in \mathcal{C}^{P_i}$,*

$$|Pr\big(C_x^{P_i}(j) = 1 : j \leftarrow \mathcal{D}_i\big) - Pr\big(P_i(z) = 1 : z \leftarrow \mathcal{D}_i\big)| \leq \nu(n),$$

where ν is a negligible function.

Lemma 3 of [1] proves that if $\mathcal{C}^P$ is accessible with respect to F and an algorithm $\mathcal{B}$ that predicts P from F with probability at least $maj_P + \epsilon$ is given, then, for a non-negligible fraction of the codewords $C_x^P \in \mathcal{C}^P$, given $f(x)$ we have access to a corrupted codeword w_x close enough to C_x^P.

Akavia *et al.* prove that the multiplication code $\mathcal{C}^P$ is accessible with respect to RSA and $EXP_{p,g}$ and they state that it also holds for *Rabin* and ECL. In section 7 we prove that $\mathcal{C}^P$ is accessible with respect to the Paillier one-way function.

Once the accessibility of the code with respect to a one-way function f is stablished, to prove that P is a hard-core of f it suffices to see that the multiplication code C_x^P is concentrated and recoverable. Concerning the concentration,

observe that if $x \in \mathbb{Z}_N^*$, from the definition of multiplication code, there is a simple relation between the Fourier coefficients of C_x^P and P, $\widehat{C_x^P(\beta)} = \widehat{P\big(\beta/x\big)}$. As a consequence,

Lemma 1. *For all $\epsilon > 0$, if P is ϵ-concentrated in Γ then C_x^P is ϵ-concentrated in $\Gamma' = \{\chi_\beta : \beta = \alpha x \mod N, \chi_\alpha \in \Gamma\}$.*

5 The security of all bits for special N

The purpose of this section is to prove that the predicate $P(x) = B_i(x)$, defined in section 2, is a hard-core predicate of any one-way function defined over $\mathbb{Z}_N$ for which the multiplication code is accessible, for N of special form. Because of theorem 3 it suffices to prove that the multiplication code $\mathcal{C}^{B_i}$ is concentrated and recoverable.

The organization of this section is the following: to prove that P is concentrated, we begin giving an explicit formula for the Fourier coefficients of the ith bit in subsection 5.1. This formula is used in subsection 5.2 to study the asymptotic behavior of $|\widehat{P(\alpha)}|^2$.

In subsection 5.3 we prove that P is concentrated for all N of a special form. Theorem 6 of subsection 5.4 proves one of the main results of the paper namely that the predicate ith bit is hard-core of any one-way function defined over $\mathbb{Z}_N$ for which the multiplication code is accessible, for N of special form. To do this we prove the recoverability of the code $\mathcal{C}^{B_i}$ in theorem 5. It turns out that these partial results are enough to reprove the hardness of $O(\log n)$ most and least significant bits.

5.1 The Fourier representation of the ith bit

Let $P(x) = B_i(x)$ be the ith bit as defined in section 2 and $N = r2^{i+1} \pm m$, where $0 < m < 2^i$. Define the function $g(x) = \dfrac{P(x + 2^i) + P(x)}{2}$.

Recall that $w_N = e^{\frac{2\pi j}{N}}$. From the definitions given in section 3.1 and using some properties of the Fourier coefficients, we have the following relation

$$\widehat{g(\alpha)} = \frac{(w_N^{2^i\alpha} + 1)}{2}\widehat{P(\alpha)}.$$

We consider two different situations:

- **Case 1** $N = r2^{i+1} - m$. In this case $g(x) = 1$ if and only if $x \in I_1 \stackrel{\text{def}}{=} [(r-1)2^{i+1} + 2^i - m, (r-1)2^{i+1} + 2^i - 1]$, else $g(x) = 0$.
- **Case 2** $N = r2^{i+1} + m$. In this case $g(x) = 1$ if and only if $x \in I_2 \stackrel{\text{def}}{=} [r2^{i+1}, r2^{i+1} + m - 1]$, else $g(x) = 0$.

In either of the two cases it is easy to compute the Fourier coefficients of P explicitly. Indeed, it suffices to find $\widehat{g(\alpha)}$ since $w_N^{2^i\alpha}+1 \neq 0$ because $m \neq 0$. Note that in both Case 1 and Case 2, $g(x)$ is only different from 0 in an interval of length m. As a result the non-zero summands in the expression of $\widehat{g(\alpha)}$ form a geometric progression with exactly m terms and $\widehat{g(\alpha)}$ can be computed explicitly. If $\alpha \neq 0$, in Case 1:

$$\widehat{g(\alpha)} = \frac{1}{N}\sum_{y\in I_1}\overline{\chi_\alpha(y)} = \frac{1}{N}\; w_N^{-\alpha((r-1)2^{i+1}+2^i-m)}\; \frac{(w_N^{-\alpha m}-1)}{(w_N^{-\alpha}-1)}.$$

Analogously, in Case 2, $\widehat{g(\alpha)} = \frac{1}{N}\; w_N^{-\alpha(r2^{i+1})}\; \frac{(w_N^{-\alpha m}-1)}{(w_N^{-\alpha}-1)}$. Moreover, in both cases, $\widehat{g(0)} = \frac{m}{N}$.

Taking the modulus, we obtain that in both cases, for any $\alpha \neq 0$

$$|\widehat{g(\alpha)}|^2 = \frac{1}{N^2}\frac{\sin^2(\frac{m\alpha\pi}{N})}{\sin^2(\frac{\alpha\pi}{N})}.$$

Using the fact that $|w_N^{2^i\alpha}+1|^2 = 4\cos^2(\frac{2^i\alpha\pi}{N})$, we obtain

$$|\widehat{P(\alpha)}|^2 = |\widehat{g(\alpha)}|^2\frac{1}{\cos^2(\frac{2^i\alpha\pi}{N})} = \frac{1}{N^2}\frac{\sin^2(\frac{m\alpha\pi}{N})}{\sin^2(\frac{\alpha\pi}{N})\cos^2(\frac{2^i\alpha\pi}{N})}. \tag{1}$$

Remark There is an alternative trick to compute $|\widehat{P(\alpha)}|^2$, it suffices to consider the function $G(x) = \frac{P(x+1)-P(x)}{2}$. In this case,

$$|\widehat{P(\alpha)}|^2 = |\widehat{G(\alpha)}|^2\frac{1}{\sin^2(\frac{\alpha\pi}{N})}. \tag{2}$$

Define Case 1 and Case 2 as above. Note that in either one of the cases the function takes values in $\{\pm 1\}$ whenever $x = k2^i - 1$, for $k \in \mathbb{Z}$. Additionally, in Case 1, $G(N-1) = 1$. As a consequence, in both cases the function takes exactly $2r$ non-zero values. This remark will be useful in subsections 5.2 and 5.3.

5.2 Asymptotic behaviour of the Fourier Coefficients of the ith bit

We have just seen how to compute the coefficients $\widehat{P(\alpha)}$. In this section we use basic calculus techniques to study its asymptotic behavior.

Proposition 1.

$$|\widehat{P(\alpha)}|^2 = \Theta\left(\frac{abs_N(m\alpha)^2}{abs_N(\alpha)^2 abs_N(2^i\alpha - N/2)^2}\right)$$

The proof essentially follows from the following lemma.

Lemma 2. $\pi^2(1-\frac{\pi^2}{12})(abs_N(y))^2 \leq N^2\sin^2(\frac{y\pi}{N}) \leq \pi^2(abs_N(y))^2$

Proof. We use the fact that $x^2 - \frac{x^4}{3} \leq \sin^2 x \leq x^2$, for any $x \in [-\pi, \pi]$, then:

- Left inequality: Let j be the only integer such that $|y - jN| \leq \frac{N}{2}$, then

$$N^2\sin^2(\tfrac{y\pi}{N}) = N^2\sin^2(\tfrac{y\pi}{N} - j\pi) \geq N^2(\tfrac{y\pi}{N} - j\pi)^2(1 - \tfrac{1}{3}(\tfrac{y\pi}{N} - j\pi)^2) \geq \\ \geq \pi^2(1-\tfrac{\pi^2}{12})(abs_N(y))^2$$

- Right inequality: Similarly, $N^2\sin^2(\frac{y\pi}{N}) \leq N^2(\frac{y\pi}{N} - j\pi)^2 = \pi^2(abs_N(y))^2$.

□

To prove proposition 1, we first define j as the unique odd integer in $[-2^i, 2^i]$ such that $|2^i\alpha - j\frac{N}{2}| \leq \frac{N}{2}$. Since $\cos^2(\frac{2^i\alpha\pi}{N}) = \sin^2(\frac{2^i\alpha\pi}{N} - j\frac{\pi}{2})$, proposition 1 is derived from expression (1) of $|\widehat{P(\alpha)}|^2$ and lemma 2 for $y = \alpha$, $y = m\alpha$ and $y = 2^i\alpha - N/2$.

Summarizing, we have proven proposition 1 and the constants implied in the symbol $\Theta()$ can be found explicitly, indeed,

$$K_1 \cdot \frac{abs_N(m\alpha)^2}{abs_N(\alpha)^2 abs_N(2^i\alpha - \frac{N}{2})^2} \leq |\widehat{P(\alpha)}|^2 \leq K_2 \cdot \frac{abs_N(m\alpha)^2}{abs_N(\alpha)^2 abs_N(2^i\alpha - \frac{N}{2})^2},$$

where $K_1 \stackrel{\textbf{def}}{=} (\frac{1}{\pi^2} - \frac{1}{12})$ and $K_2 \stackrel{\textbf{def}}{=} \dfrac{1}{\pi^2(1-\frac{\pi^2}{12})^2}$.

Finally, if $K_3 \stackrel{\textbf{def}}{=} \dfrac{1}{\pi^2(1-\frac{\pi^2}{12})}$, we prove

Lemma 3. $|\widehat{P(\alpha)}|^2 \leq K_3 \cdot \ min\left\{\dfrac{m^2}{abs_N(2^i\alpha - \frac{N}{2})^2}, \dfrac{4r^2}{abs_N(\alpha)^2}\right\}$

Proof. The function g is equal to 0 in all but m elements of the domain and therefore $|\widehat{g(\alpha)}|^2 \leq \dfrac{m^2}{N^2}$, since each coefficient is the sum of m terms in the unit circle. The inequalities of lemma 2 and expression (1) imply that $|\widehat{P(\alpha)}|^2 \leq K_3 \cdot \dfrac{m^2}{abs_N(2^i\alpha - \frac{N}{2})^2}$. To prove the other bound observe that $|\widehat{G(\alpha)}|^2 \leq \dfrac{4r^2}{N^2}$ and use expression (2) and lemma 2.

□

5.3 The concentration of the ith bit for certain N

In the previous section we found an expression of the asymptotic behavior of the coefficients of P which is hard to interpret. It is clear that the heavy coefficients of P will be around the points that annihilate the denominator, but otherwise it is not trivial to show that there exists a set Γ of size $poly(n/\epsilon)$ such that $||P - P_{|\Gamma}||_2^2 \leq \epsilon$.

In this section we prove that if $N = r2^{i+1} \pm m$ and either $r \in poly(n)$ or $m \in poly(n)$ then P is concentrated. The result is a consequence of the two following lemmas.

Lemma 4. *For any $\epsilon > 0$, $||P - P_{|\Gamma_{2^i}}||_2^2 \leq \epsilon$, where*

$$\Gamma_{2^i} \stackrel{\textbf{def}}{=} \{\chi_\alpha : abs_N(2^i\alpha - N/2) \leq O(\frac{m^2}{\epsilon})\}.$$

Proof. Let $\Gamma_k^c \stackrel{\textbf{def}}{=} \{\chi_\alpha : abs_N(2^i\alpha - N/2) > k\}$ then, using one of the bounds of lemma 3

$$\sum_{\chi_\alpha \in \Gamma_k^c} |\widehat{P(\alpha)}|^2 \leq O(m^2) \sum_{\chi_\alpha \in \Gamma_k^c} \frac{1}{abs_N(2^i\alpha - N/2)^2} < O(\frac{m^2}{k}).$$

Taking $k \in O(\frac{m^2}{\epsilon})$, $||P - P_{|\Gamma_{2^i}}||_2^2 \leq \epsilon$. □

Similarly, using the other bound of lemma 3

Lemma 5. *For any $\epsilon > 0$, $||P - P_{|\Gamma_0}||_2^2 \leq \epsilon$, where*

$$\Gamma_0 \stackrel{\textbf{def}}{=} \{\chi_\alpha : abs_N(\alpha) \leq O(\frac{r^2}{\epsilon})\}.$$

That is, P is concentrated in $\Gamma \stackrel{\textbf{def}}{=} \Gamma_0 \cap \Gamma_{2^i}$ and in case either $r \in poly(n)$ or $m \in poly(n)$, the cardinal of this set is $poly(n/\epsilon)$. Because of lemma 1, we have proven the following theorem.

Theorem 4. *The code $\mathcal{C}^{B_i}$ is concentrated for all $N = r2^{i+1} \pm m$, $0 < m < 2^i$, with either $r \in poly(n)$ or $m \in poly(n)$.*

5.4 The hardness of the ith bit for certain N

We study the recoverability of $\mathcal{C}^P$. The recovery algorithm is adapted from lemma 5 of [1] which proved that if B is a t-segment predicate, $\mathcal{C}^B$ is recoverable. Combined with the concentration proven in theorem 4, the recoverability of $\mathcal{C}^P$ will prove the main result about the hardness of the ith bit for certain N.

Theorem 5. *The code $\mathcal{C}^{B_i}$ is recoverable for all $N = r2^{i+1} \pm m$ with either $r \in poly(n)$ or $m \in poly(n)$ for N a prime or an RSA modulus.*

Proof. We first consider the case $r \in poly(n)$. In this case, because of lemma 1 and lemma 5, C_x^P is τ-concentrated in $\Gamma_0' \stackrel{\mathbf{def}}{=} \{\chi_\beta : \beta = \alpha x \mod N, abs_N(\alpha) \leq O(\frac{r^2}{\tau})\}$.

The inputs of the recovery algorithm are a character χ_β and a threshold parameter τ (where $1/\tau \in poly(n)$). The output is a list containing $x \in \mathbb{Z}_N$ such that $\chi_\beta \in Heavy_\tau(C_x^P)$.

Since, C_x^P is τ-concentrated in Γ_0', $\chi_\beta \in Heavy_\tau(C_x^P)$ implies $\chi_\beta \in \Gamma_0'$ and thus $\beta = \alpha x \mod N$ for $abs_N(\alpha) \leq poly(n/\tau)$. The algorithm outputs the union of the lists L_α such that L_α contains all x so that $x = \beta/\alpha \mod N$. If $\alpha \in \mathbb{Z}_N^*$ there is a single solution to this equation. If $gcd(N, \alpha) = d \neq 1$, the solution of $\frac{\alpha}{d}x = \beta \mod \frac{N}{d}$ is either empty or a list

$$L_\alpha = \left\{x + i \cdot \frac{N}{d} \mod N\right\}_{i=0,\ldots,d-1}.$$

The union of the lists L_α (over all α such that $abs(\alpha) \leq O(\frac{r^2}{\tau})$) contains all x such that $Heavy_\tau(C_x^P) \ni \chi_\beta$. For the length of the lists and the time of constructing them be $poly(n/\tau)$, it must be that $d \in poly(n)$, since L_α has length d. This condition is trivially satisfied if N is a prime. If N is an RSA modulus, $gcd(N, \alpha) \neq 1$ implies factoring N. So, for an RSA modulus $N = r2^{i+1} \pm m$ with $r \in poly(n)$, either the code $\mathcal{C}^{B_i}$ is recoverable or P is concentrated in a known set Γ_0 of polynomial size which contains some element which allows to factorize N, contradicting the unfeasibility of factoring RSA modulus.

The case $m \in poly(n)$ is proven in a similar way but now taking into account C_x^P is τ-concentrated in $\Gamma_{2^i}' \stackrel{\mathbf{def}}{=} \{\chi_\beta : \beta = \alpha x \mod N, abs_N(2^i\alpha - N/2) \leq O(\frac{m^2}{\tau})\}$. □

Theorem 2 states that, as a consequence of theorems 4 and 5, the code $\mathcal{C}^{B_i}$ is list decodable for all $N = r2^{i+1} \pm m$ with either $r \in poly(n)$ or $m \in poly(n)$ and N a prime or an RSA modulus.

Finally, we conclude

Theorem 6. *The predicate B_i, ith bit, is a hard-core predicate for any one-way function defined over $\mathbb{Z}_N$ for which the multiplication code $\mathcal{C}^{B_i}$ is accessible, for all $N = r2^{i+1} \pm m$ prime or RSA modulus such that either $r \in poly(n)$ or $m \in poly(n)$.*

This theorem is a consequence of the list decodability of the code $\mathcal{C}^{B_i}$ and theorem 3.

This proves the hardness of all bits for N with a special binary representation, but it also reproves the hardness of the $O(\log n)$ most and least significant bits for all N of cryptographic interest.

– **Most significant bits** If $n - i \in O(\log n)$, then $r \in poly(n)$. Theorem 6 proves the security of the first $O(\log n)$ most significant bits for all N prime

or RSA modulus for any one-way function defined over $\mathbb{Z}_N$ for which the multiplication code $\mathcal{C}^{B_i}$ is accessible.

– **Least significant bits** If $i \in O(\log n)$ then $m \in poly(n)$. Theorem 6 proves the security of the first $O(\log n)$ most significant bits for all N prime or RSA modulus for any one-way function defined over $\mathbb{Z}_N$ for which the multiplication code $\mathcal{C}^{B_i}$ is accessible.

6 The security of all bits for all N

This section is devoted to prove the hardness of all bits for all cryptographically relevant N. First of all in subsection 6.1 we study the bounds given in section 5.2 more accurately. In subsection 6.2 we proceed to prove that P is concentrated for N prime or RSA modulus. This result, together with the recovery algorithm given in subsection 6.3 and the accessibility of $\mathcal{C}^P$, implies the security of the internal bits for all N of cryptographic interest. This is summarized in theorem 9.

6.1 A closer look at the asymptotic behavior of $|\widehat{P(\alpha)}|^2$

The bounds of section 5.2 are not enough to prove the concentration in the general case. Therefore, in this section we will study the asymptotic behavior of $|\widehat{P(\alpha)}|^2$ in more detail.

As it was proven in proposition 1

$$|\widehat{P(\alpha)}|^2 = \Theta\left(\frac{abs_N(m\alpha)^2}{abs_N(\alpha)^2 abs_N(2^i\alpha - N/2)^2}\right).$$

We introduce some notation to express the elements $\alpha \in \mathbb{Z}_N$ as a function of some parameters useful to describe $abs_N(\alpha)$ and $abs_N(2^i\alpha - N/2)$. Recall that $N = r2^{i+1} \pm m$, where $0 < m < 2^i$.

Some parameters. We define some parameters depending on $\alpha \geq 0$ or $\alpha < 0$. First consider the case $\alpha \in [0, \frac{N-1}{2}]$. Denote $\delta_\alpha \equiv \alpha 2^i - \frac{N-1}{2} \mod N$, where $\delta_\alpha \in [-\frac{N-1}{2}, \frac{N-1}{2}]$ and let λ_α be the integer in $[0, 2^{i-1} - 1]$ such that

$$\alpha 2^i = (N-1)/2 + \delta_\alpha + \lambda_\alpha N. \tag{3}$$

Let α be an integer in $[-\frac{N-1}{2}, 0)$. Denote $\delta_\alpha \equiv \alpha 2^i + \frac{N+1}{2} \mod N$, where $\delta_\alpha \in [-\frac{N-1}{2}, \frac{N-1}{2}]$ and let λ_α be the integer in $[0, 2^{i-1} - 1]$ such that

$$\alpha 2^i = -(N+1)/2 + \delta_\alpha - \lambda_\alpha N. \tag{4}$$

Finally, for any $\alpha \in [-\frac{N-1}{2}, \frac{N-1}{2}]$ we define $\mu_\alpha \in [0, r]$ as the only integer such that $abs_N(2^i\alpha - (N-1)/2) = \mu_\alpha 2^i + \widetilde{\delta}_\alpha$ with $\widetilde{\delta}_\alpha \in [0, 2^i - 1]$.

From equations 3 and 4, if $\alpha \geq 0$,

$$\alpha = ((N-1)/2 + \delta_\alpha + \lambda_\alpha N)/2^i, \tag{5}$$

and if $\alpha < 0$,

$$\alpha = (-(N+1)/2 + \delta_\alpha - \lambda_\alpha N)/2^i. \tag{6}$$

We emphasize that equations 3, 4, 5 and 6 are integer equalities.

The parameters μ_α and $\widetilde{\delta}_\alpha$ are determined by δ_α. Indeed,

Lemma 6. *For all $\alpha \in \mathbb{Z}_N$, $\mu_\alpha 2^i + \widetilde{\delta}_\alpha = |\delta_\alpha|$.*

Proof. We will prove that $abs_N(2^i\alpha - (N-1)/2) = |\delta_\alpha|$. This is obvious when $\alpha \geq 0$, since $\alpha 2^i - (N-1)/2 \equiv \delta_\alpha \mod N$ and $\delta_\alpha \in [-\frac{N-1}{2}, \frac{N-1}{2}]$. When $\alpha < 0$, note that

$$\alpha 2^i - (N-1)/2 \equiv \alpha 2^i + (N+1)/2 \equiv \delta_\alpha \mod N.$$

Since $\delta_\alpha \in [-\frac{N-1}{2}, \frac{N-1}{2}]$, $abs_N(\alpha 2^i - (N-1)/2) = |\delta_\alpha|$ by definition of absolute value. □

Note that lemmas 4 and 5 in section 5.3 imply that $||P - P_{|\Gamma_{2^i} \cap \Gamma_0}||_2^2 \leq \epsilon$. That is, if there exists a set Γ where P is concentrated, then $\Gamma \subset \Gamma_0 \cap \Gamma_{2^i}$. The choice of these parameters is motivated by the remark that points $\alpha \in \Gamma_0 \cap \Gamma_{2^i}$ should be close to small odd multiples of $N/2^{i+1}$, that is, observing that $N/2^{i+1} \approx r$, we will have that $abs_N(\alpha) \approx (2\lambda_\alpha + 1)r \pm \mu_\alpha$, with λ_α and μ_α small. Indeed,

Lemma 7. *For all $\alpha \in \mathbb{Z}_N$, $abs_N(\alpha) = (2\lambda_\alpha + 1)r \pm \mu_\alpha + R$, with $|R| \leq \lambda_\alpha$.*

Proof. First of all we consider the case $\alpha \in [0, \frac{N-1}{2}]$. In this case $abs_N(\alpha) = \alpha$.

Suppose $\delta_\alpha \in [0, \frac{N-1}{2}]$ and $N = r2^{i+1} - m$ (i.e., Case 1 of section 5.1). Since $\delta_\alpha \geq 0$, lemma 6 implies $\delta_\alpha = \widetilde{\delta}_\alpha + \mu_\alpha 2^i$. Substituting in equation 5, we get

$$abs_N(\alpha) = (2\lambda_\alpha + 1)r + \mu_\alpha + \frac{-(2\lambda_\alpha + 1)m + 2\widetilde{\delta}_\alpha - 1}{2^{i+1}}.$$

Similarly for the rest of cases, that is: (1) $\delta_\alpha \in [-\frac{N-1}{2}, 0)$ and $N = r2^{i+1} - m$, (2) $\delta_\alpha \in [0, \frac{N-1}{2}]$ and $N = r2^{i+1} + m$ and (3) $\delta_\alpha \in [-\frac{N-1}{2}, 0)$ and $N = r2^{i+1} + m$, we obtain

$$abs_N(\alpha) = (2\lambda_\alpha + 1)r \pm \mu_\alpha + \frac{\pm(2\lambda_\alpha + 1)m \pm 2\widetilde{\delta}_\alpha - 1}{2^{i+1}}.$$

On the other hand, in the case $\alpha < 0$, $abs_N(\alpha) = -\alpha$. Considering all the possible combinations for the sign of δ_α and Case 1 and Case 2, as above, and substituting in equation 6, we obtain

$$abs_N(\alpha) = (2\lambda_\alpha + 1)r \pm \mu_\alpha + \frac{\pm(2\lambda_\alpha + 1)m \pm 2\widetilde{\delta}_\alpha + 1}{2^{i+1}}.$$

If $\lambda_\alpha = 0$ it is easy to see that $R = 0$ and the lemma is true. Indeed, $\lambda_\alpha = 0$ implies $\pm m \pm 2\widetilde{\delta}_\alpha \pm 1 = 0 \mod 2^{i+1}$ due to the fact that equations 3 and 4 were integer equalities. Because of the range of definition of m and $\widetilde{\delta}_\alpha$, this congruence is equivalent to the equality $\pm m \pm 2\widetilde{\delta}_\alpha \pm 1 = 0$ and therefore $R = 0$. Now, to prove the lemma proceed by induction over λ_α. □

Corollary 1. *For all* $\alpha \in \mathbb{Z}_N$, $abs_N(\alpha) \geq \lambda_\alpha(2r-1)$.

Proof. Simply note that $\mu_\alpha \in [0, r]$ and $|R| \leq \lambda_\alpha$. Then,

$$abs_N(\alpha) = (2\lambda_\alpha + 1)r \pm \mu_\alpha + R \geq 2\lambda_\alpha r + R \geq \lambda_\alpha(2r-1).$$

□

From these lemmas we can easily prove the following:

Lemma 8. $abs_N(\alpha)^2 abs_N(2^i\alpha - \frac{N-1}{2})^2 \geq \lambda_\alpha^2\mu_\alpha^2 r^2 2^{2i+2}\frac{1}{4}$.

Proof. As we have seen in corollary 1, $abs_N(\alpha) \geq \lambda_\alpha(2r-1)$, therefore

$abs_N(\alpha)^2 abs_N(2^i\alpha - \frac{N-1}{2})^2 \geq \lambda_\alpha^2(2r-1)^2(\mu_\alpha 2^i + \widetilde{\delta}_\alpha)^2 \geq \lambda_\alpha^2(2r-1)^2(\mu_\alpha 2^i)^2 \geq$
$\geq \lambda_\alpha^2\frac{(2r-1)^2}{(2r)^2}(2r)^2(\mu_\alpha 2^i)^2 \geq \lambda_\alpha^2\mu_\alpha^2 r^2 2^{2i+2}\frac{1}{4}$.

□

Now it is easy to characterize the asymptotic behavior of $|\widehat{P(\alpha)}|^2$ in terms of λ_α and μ_α.

Proposition 2. *For all* $\alpha \in \mathbb{Z}_N$ *such that* $\lambda_\alpha > 0$ *and* $\mu_\alpha > 0$

$$|\widehat{P(\alpha)}|^2 < O\big(\frac{1}{\lambda_\alpha^2\mu_\alpha^2}\big).$$

Proof. Since $abs_N(m\alpha) \leq N/2$, from proposition 1 and the lemma 8, if $\lambda_\alpha > 0$ and $\mu_\alpha > 0$,

$$|\widehat{P(\alpha)}|^2 < O\left(\frac{abs_N(m\alpha)^2}{abs_N(\alpha)^2 abs_N(2^i\alpha - N/2)^2}\right) < O\big(\frac{N^2}{\lambda_\alpha^2\mu_\alpha^2 r^2 2^{2i+2}}\big) < O\big(\frac{1}{\lambda_\alpha^2\mu_\alpha^2}\big).$$

□

Before proceeding to prove our main theorem, we note that elements in $\mathbb{Z}_N$ have a convenient representation in these parameters.

Lemma 9. *The following map is injective*

$$\begin{array}{ccccccccc} \pi : [-\frac{N-1}{2}, \frac{N-1}{2}] & \longrightarrow & [0, 2^{i-1}-1] & \times [0,r] & \times \{\pm 1\} & \times \{\pm 1\} \\ \alpha & \longmapsto & (\ \lambda_\alpha, & \mu_\alpha, & s_\alpha, & s_\delta\ ) \end{array}$$

where $s_\alpha = 1$ *if* $\alpha \geq 0$ *and* -1 *otherwise and* $s_\delta = 1$ *if* $\delta_\alpha \geq 0$ *and* -1 *otherwise.*

Proof. Reducing equations 3 and 4 modulo 2^i it is clear that λ_α and s_δ determine $\widetilde{\delta}_\alpha$ modulo 2^i and therefore $\widetilde{\delta}_\alpha$. Then α can be computed from $\pi(\alpha)$ using equations 5 or 6. □

6.2 The concentration of the ith bit for all N

As a result of lemma 9 of the above section, we can describe the elements of $\mathbb{Z}_N$ as regions in $[0, 2^{i-1}-1] \times [0, r] \times \{\pm 1\} \times \{\pm 1\}$. Now we can present our result about the concentration of the ith bit.

Theorem 7. *P is ϵ-concentrated in $\Gamma \stackrel{\textbf{def}}{=} \{\chi_\alpha : \lambda_\alpha < O(\frac{1}{\epsilon}), \mu_\alpha < O(\frac{1}{\epsilon})\}$.*

Proof. Let $\Gamma_k \stackrel{\textbf{def}}{=} \{\chi_\alpha : \lambda_\alpha \leq k, \mu_\alpha \leq k\}$, we will prove that

$$\sum_{\chi_\alpha \notin \Gamma_k} |\widehat{P(\alpha)}|^2 < O(\frac{1}{k}).$$

Note that $\mathbb{Z}_N \backslash \Gamma_k = \{\chi_\alpha : \lambda_\alpha = 0, \mu_\alpha > k\} \cup \{\chi_\alpha : \lambda_\alpha > k, \mu_\alpha = 0\} \cup \{\chi_\alpha : \lambda_\alpha > k, \mu_\alpha \geq 1\} \cup \{\chi_\alpha : \lambda_\alpha \geq 1, \mu_\alpha > k\}$.

To bound the sum of $|\widehat{P(\alpha)}|^2$ over the two first sets, the bounds of lemma 3 of section 5.2 will suffice, while we will need proposition for the other bounds. Indeed, when $\lambda_\alpha = 0$, using one of the bounds of lemma 3,

$$\begin{aligned}
\sum_{\lambda_\alpha=0,\mu_\alpha>k} |\widehat{P(\alpha)}|^2 &\leq O(m^2) \sum_{\lambda_\alpha=0,\mu_\alpha>k} \frac{1}{abs_N(2^i\alpha - (N-1)/2)^2} < \\
&< O(m^2) \sum_{\lambda_\alpha=0,\mu_\alpha>k} \frac{1}{(\widetilde{\delta}_\alpha + \mu_\alpha 2^i)^2} < \\
&< O(m^2) \sum_{\lambda_\alpha=0,\mu_\alpha>k} \frac{1}{(\mu_\alpha 2^i)^2} < \\
&< O(1) \sum_{\lambda_\alpha=0,\mu_\alpha>k} \frac{1}{\mu_\alpha^2} < O(\frac{1}{k}).
\end{aligned}$$

On the other hand, when $\mu_\alpha = 0$, using the other bound of lemma 3,

$$\begin{aligned}
\sum_{\lambda_\alpha>k,\mu_\alpha=0} |\widehat{P(\alpha)}|^2 &\leq O(r^2) \sum_{\lambda_\alpha>k,\mu_\alpha=0} \frac{1}{abs_N(\alpha)^2} < \\
&< O(r^2) \sum_{\lambda_\alpha>k,\mu_\alpha=0} \frac{1}{\lambda_\alpha^2 (2r-1)^2} < \\
&< O(1) \sum_{\lambda_\alpha>k,\mu_\alpha=0} \frac{1}{\lambda_\alpha^2} < O(\frac{1}{k}).
\end{aligned}$$

To conclude the proof we need to show that

$$\sum_{\lambda_\alpha>k,\mu_\alpha\geq 1} |\widehat{P(\alpha)}|^2 + \sum_{\lambda_\alpha\geq 1,\mu_\alpha>k} |\widehat{P(\alpha)}|^2 < O(\frac{1}{k}).$$

From proposition 6.2,

$$|\widehat{P(\alpha)}|^2 < O(\frac{1}{\lambda_\alpha^2 \mu_\alpha^2}).$$

As a consequence,

$$\sum_{\lambda_\alpha > k, \mu_\alpha \geq 1} |\widehat{P(\alpha)}|^2 + \sum_{\lambda_\alpha \geq 1, \mu_\alpha > k} |\widehat{P(\alpha)}|^2 \leq \sum_{\lambda_\alpha > k, \mu_\alpha \geq 1} \frac{1}{\lambda_\alpha^2 \mu_\alpha^2} + \sum_{\lambda_\alpha \geq 1, \mu_\alpha > k} \frac{1}{\lambda_\alpha^2 \mu_\alpha^2} =$$
$$= \sum_{\mu_\alpha \geq 1} \frac{1}{\mu_\alpha^2} \Big(\sum_{\lambda_\alpha > k} \frac{1}{\lambda_\alpha^2} \Big) + \sum_{\lambda_\alpha \geq 1} \frac{1}{\lambda_\alpha^2} \Big(\sum_{\mu_\alpha > k} \frac{1}{\mu_\alpha^2} \Big) < O(\frac{1}{k}).$$

We conclude that if $k \in O(\frac{1}{\epsilon})$, $\sum_{\chi_\alpha \notin \Gamma} |\widehat{P(\alpha)}|^2 \leq \epsilon$ as stated in theorem 7.

□

6.3 The hardness of the ith bit for all N

In the previous section we proved the concentration of the predicate B_i. To complete the proof of the main theorem concerning its hardness, in this section we prove the recoverability of the code $\mathcal{C}^{B_i}$.

Theorem 8. *The code $\mathcal{C}^{B_i}$ is recoverable for all N prime or RSA modulus.*

Proof. This recovery algorithm is almost identical to the one given in [1].

Because of theorem 7 and lemma 1, C_x^P is τ-concentrated in $\Gamma' \stackrel{\text{def}}{=} \{\chi_\beta : \beta = \alpha x \mod N, \chi_\alpha \in \Gamma\}$, where $\Gamma \stackrel{\text{def}}{=} \{\chi_\alpha : \lambda_\alpha < O(\frac{1}{\tau}), \mu_\alpha < O(\frac{1}{\tau})\}$.

The inputs of the recovery algorithm are a character χ_β and a threshold parameter τ, where $1/\tau \in poly(n)$. The output is a list containing $x \in \mathbb{Z}_N$ such that $\chi_\beta \in Heavy_\tau(C_x^P)$.

Since, C_x^P is τ-concentrated in Γ', $\chi_\beta \in Heavy_\tau(C_x^P)$ implies $\chi_\beta \in \Gamma'$ and thus $\beta = \alpha x \mod N$ for $\lambda_\alpha < poly(n/\tau)$ and $\mu_\alpha < poly(n/\tau)$. The algorithm outputs the list $L \stackrel{\text{def}}{=} \{x : x = \beta/\alpha \mod N, \chi_\alpha \in \Gamma\}$. The length of the list and the time of constructing it is $poly(n/\tau)$.

□

Theorem 2 states that, as a consequence of theorems 7 and 8, the code $\mathcal{C}^{B_i}$ is list decodable for all N prime or RSA modulus. We conclude

Theorem 9. *If N is prime or is an RSA modulus, the predicate B_i is hard-core for any one-way function defined over $\mathbb{Z}_N$ for which the multiplication code is accessible.*

7 All bits of the Paillier encryption scheme are secure

Let $N = p \cdot q$ be an RSA modulus, given an element $g \in \mathbb{Z}_{N^2}^*$ such that N divides the order of g, the Paillier trapdoor permutation, introduced in [10] is the map:

$$\begin{array}{rccc} \mathcal{E}_g : & \mathbb{Z}_N^* \times \mathbb{Z}_N & \longrightarrow & \mathbb{Z}_{N^2}^* \\ & (r, m) & \mapsto & r^N \ g^m \end{array}$$

Taking r to be a random element and m the plaintext, the Paillier probabilistic encryption scheme encrypts m as $\mathcal{E}_g(r, m)$ and is semantically secure under the Decisional Composite N-th residuosity assumption.

In this section we will sketch the proof of the security of any bit P of the message. We stress that by security of P we mean that we relate the ability of an adversary in predicting $P(m)$ from $\mathcal{E}_g(r, m)$ to the ability of recovering m from $\mathcal{E}_g(r, m)$, and not to the ability of inverting $\mathcal{E}_g$.

The concentration and recoverability of the multiplication code $\mathcal{C}^P$, where $P : \mathbb{Z}_N \to \{\pm 1\}$ is the predicate ith bit of the message, follows from our results of section 6.3. Then, the code $\mathcal{C}^P$ is list decodable.

The one-way function $\mathcal{E}_g$ has domain in $\mathbb{Z}_N^* \times \mathbb{Z}_N$, while the predicate has domain in $\mathbb{Z}_N$, so we need to slightly change the definition of accessibility to fit this situation. We will first give the access algorithm, then we will give the new definition of accessible code and argue that this new definition is enough to apply the list decoding methodology.

The access algorithm $\mathcal{A}$, on input $(N, g, \mathcal{E}_g(r, x), j)$, chooses a random element $\ell \in \mathbb{Z}_N^*$, and outputs $(\mathcal{E}_g(r, x))^j \cdot \ell^N$. Note that

$$(\mathcal{E}_g(r, x))^j \cdot \ell^N \equiv \mathcal{E}_g(r^j \cdot \ell, xj) \mod N^2.$$

It is not hard to see that for this access algorithm $\mathcal{A}$ the code satisfies the following properties:

1. **Code access:** $\forall x, j \in \mathbb{Z}_N$, $\mathcal{A}(N, g, \mathcal{E}_g(r, x), j)$ returns $\mathcal{E}_g(r', x')$ such that $C_x^{P_{N,g}}(j) = P_{N,g}(x')$
2. **Well spread:** For uniformly distributed $C_x^{P_{N,g}} \in \mathcal{C}^{P_{N,g}}$ and $j \in \mathbb{Z}_N$, the distribution of $(r', x') \in \mathbb{Z}_N^* \times \mathbb{Z}_N$ satisfying $\mathcal{E}_g(r', x') = \mathcal{A}(N, g, \mathcal{E}_g(r, x), j)$ is statistically close to the uniform distribution on $\mathbb{Z}_N^* \times \mathbb{Z}_N$
3. **Bias preserving:** For every codeword $C_x^{P_{N,g}} \in \mathcal{C}^{P_{N,g}}$,

$$|\Pr\big(C_x^{P_{N,g}}(j) = 1 : j \leftarrow \mathbb{Z}_N\big) - \Pr\big(P_{N,g}(z) = 1 : z \leftarrow \mathbb{Z}_N\big)| \leq \nu(n),$$

where ν is a negligible function.

Compare these properties with the ones that an accessible code must verify (see definition 10 of section 4). Both definitions are almost identical but now the property that the code is well spread is for $(r', x') \in \mathbb{Z}_N^* \times \mathbb{Z}_N$ and not on the domain of the definition of the code.

Lemmas 2 and 3 of [1] prove that if the code $\mathcal{C}^P$ is accessible with respect to F, an oracle $\mathcal{B}$ predicting P from F with probability exceeding $maj_P + \epsilon$

implies access to a corrupted codeword w such that $\Delta(C_x, w) \leq minor_{C_x} - \epsilon$. The property that the code is well spread is used in the proofs of these lemmas. It is immediate to see that to do the same reasoning in the Paillier case we need to require precisely the second condition above.

Summarizing, the definition we gave above is exactly the one we need to prove that an oracle predicting $P(x)$ from $\mathcal{E}_g(r', x)$ gives access to a corrupted codeword w sufficiently close to x. But to prove theorem 3 which states that list decodability of $\mathcal{C}^P$ plus accessibility implies that P is a hard-core of F, accessibility was only necessary to prove access to a corrupted codeword w. Therefore, the argument we gave in this section with the concentration and recoverability of the multiplication code $\mathcal{C}^{P_{N,g}}$ of section 6.3 implies the security of all bits of the message of the Paillier trapdoor permutation.

We emphasize that the security of all bits of the Paillier encryption scheme was only known based on a non-standard computational assumption [3].

8 Other predicates

We note that theorem 6 is enough to reprove the hardness of segment predicates. Recall from section 2 that a t-segment predicate is a predicate which changes value $t \in poly(n)$ number of times. Define G as

$$G(x) = \frac{P(x+1) - P(x)}{2},$$

and note that $G(x) \neq 0$ for exactly t values of x. Although in this case we cannot compute $\widehat{G(\alpha)}$ explicitly as before, we still have $|\widehat{G(\alpha)}|^2 \leq t^2/N^2$. The same arguments as in lemma 5 prove that P is concentrated up to ϵ in $\Gamma_0 \stackrel{\mathbf{def}}{=} \{\chi_\alpha : abs_N(\alpha) \leq O(\frac{t^2}{\epsilon})\}$ - this corresponds to claim 4.1 of [1].

In the last section we proved the security of all bits in the binary representation of the preimage for any one-way function defined over $\mathbb{Z}_N$ with multiplicative access provided that N is odd. Note that the same proof would do for any other "almost periodic" predicate. Indeed, for any $d \in \mathbb{N}$ define $P_d : \mathbb{Z}_N \to \{\pm 1\}$ as 1 if $[x] \in [kd, (k+1)d - 1]$, k even, and -1 otherwise. Write $N = r2d \pm m$, with $0 < m < d$. Then all the results proven in the last section are also valid for P_d just writing d instead of 2^i.

9 Conclusion

In our opinion the list decoding methodology formalized in [1] has not received enough attention. Because of the elegance and generality of the method and the power of the different tools it uses it should be considered the starting point of any bit security proof. In this paper we have extended the number of predicates to which the list decoding methodology applies. As a result we prove the security of all bits of any of the usual cryptographic one-way functions with multiplicative access defined on a cyclic group of order N.

10 Acknowledgements

The authors would like to thank Eike Kiltz for his comments and suggestions that contributed to make the paper more readable.

References

1. A. Akavia, S. Goldwasser and S. Safra, *Proving Hard-Core Predicates Using List Decoding*, Proc. of the 44th Symposium on Foundations of Computer Science, 2003.
2. W.Alexi, B.Chor, O.Goldreich and C.P. Schnorr. *RSA and Rabin functions: certain parts are as hard as the whole*. SIAM J.Comp. Vol. 17-2, 1988.
3. D.Catalano, R.Gennaro and N.Howgrave-Graham. *Paillier's Trapdoor Function Hides up to $O(n)$ Bits*. J.Cryptology, vol.15-4, 2002.
4. E. Kushilevitz and Y. Mansour. *Learning Decision Trees Using the Fourier Spectrum*. Proc. of the 23rd Annual ACM Symposium on Theory of Computing, 1991.
5. A. C. Gilbert, S. Muthukrishnan and M. Strauss. *Improved time bounds for near-optimal sparse Fourier representation via sampling*. Proc. of SPIE Wavelets XI, 2005.
6. O. Goldreich and L. Levin. *A hard-core predicate for all one-way functions*. Proc. of the 21st Annual ACM Symposium on Theory of Computing, 1989.
7. O. Goldreich, R. Rubinfeld and M. Sudan. *Learning Polynomials with Queries: The Highly Noisy Case*, SIAM J. Discrete Math., vol. 13-4, 2000.
8. J.Håstad and M. Näslund. *The security of all RSA and discrete log bits*. J. ACM, vol. 51-2, 2004.
9. M. Näslund. *All Bits ax+b mod p are Hard*, Proc. of Advances in Cryptology - Crypto'96, LNCS 921, Springer-Verlag, 1996.
10. P.Paillier. *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*. Proc. Advances in Cryptology - Eurocrypt'99, LNCS 1592, Springer-Verlag, 1999.