# The Security of Individual RSA Bits

Johan Håstad Mats Näslund

NADA, Royal Institute of Technology
Stockholm, Sweden
E-mail: {johanh,matsn}@nada.kth.se

## Abstract

*We study the security of individual bits in an RSA encrypted message $E_N(x)$. We show that given $E_N(x)$, predicting any single bit in $x$ with only a non-negligible advantage over the trivial guessing strategy, is (through a polynomial time reduction) as hard as breaking RSA. We briefly discuss a related result for bit security of the discrete logarithm.*

## 1. Introduction

What is to be meant by a *secure cryptosystem*? There are rigorously defined notions such as *semantic security*; "whatever can be computed efficiently from the cryptotext should also be computable without it". Obtaining semantic security requires rather elaborate constructions, and we cannot in general hope to achieve this by simply applying a natural one-way function. It is therefore important also to analyze the security of *specific* information concerning the plaintext. We here study the question of given the encrypted message $E(x)$, is it feasible to predict even a single bit of $x$? Now, "feasible" refers to the existence of probabilistic, polynomial time algorithms, and we cannot exclude the possibility of "guessing" a bit of $x$. What we *can* hope for is that this is essentially all you can do. With this in mind, as a successful adversary, we consider one who on average has a small advantage over the trivial guessing strategy.

We study the particular case when $E(x) = E_N(x)$ is RSA encryption. Here $N$ is the product of two large primes, see [15]. RSA has been investigated from many different angles over the last 20 years, but still relatively little is known about the security. It is known that certain information such as $(x/N)$, the Jacobi symbol of $x$, leaks through $E_N(x)$. For the specific issue of security for individual bits in $x$, this has so far only been proven to be true for the $O(\log\log N)$ least significant bits. Starting from a relatively weak result, in a sequence of papers, [9, 3, 19, 8, 17, 6], this was improved, ending with the final proof of "complete" security by Alexi, Chor, Goldreich, and Schnorr in [1]. There are also other known security results for certain predicates that are related to the individual bits of $x$, e.g. $\mathrm{half}_N(x) = 1$ if $x \geq (N+1)/2$, 0 otherwise, see [9] for instance.

For the other, internal bits, however, the best known result up until now states that they can cannot be computed with probability greater than $3/4$. By using relations between $\mathrm{half}_N(x)$ and the individual bits of $x$, Ben-Or, Chor, and Shamir proved in [3], that the internal bits cannot be computed with probability of success exceeding $15/16$. By a reduction to this proof, the result in [1] for the least significant bit, then improved the result to $3/4$, still leaving a large gap to the desired $1/2$-result.

In this paper we show the following:

**Theorem.** For any constant $c$ and all sufficiently large $n$, unless RSA can be broken[1] in random polynomial time, no single bit of $E_N^{-1}(x)$ (where $\lceil \log N \rceil = n$) can be predicted with advantage[2] exceeding $n^{-c}$.

For a given function $E(x)$, the concept of bit security is of course only meaningful when computing $E^{-1}(x)$ is assumed (or known) to be hard. Under such assumptions, there are a few cases where all individual bits are known to be secure. Assuming that factoring Blum-integers is hard, Håstad, Schrift, and Shamir proved in [10] that given $g^x \bmod N$, where $N$ is a Blum-integer, all bits of $x$ are secure. Näslund showed in [13] that all bits in affine functions modulo a (not too small) prime, $x \mapsto ax + b \bmod p$, are secure given the information $a, b, p$, and $f(x)$ for *any* one-way function $f$. Our results here are achieved by extending and combining the work in [1, 3, 13].

Besides the obvious usefulness in proving that certain bits of the plaintext are secure, results of this type also have applications for *pseudo-random number generation*. Following a general construction by Blum and Micali [4], one can in the RSA case do as follows. Pick a random $x_0$. Set $x_{j+1} = E_N(x_j)$ and output the sequence

[1] Here, "breaking" simply means retrieving the message $x$. In particular, our result is not connected to issues such as the relationship between RSA and factoring, recently investigated in [5].

[2] We do not give credit to "trivial" advantage due to bias.

$b(x_0), b(x_1), \ldots, b(x_m)$ where $b(\cdot)$ is some $0/1$-function that is secure with respect to $E_N(\cdot)$, in our case $b(x_j) =$ "$i$th bit of $x_j$".

The paper is organized as follows. After first giving some notation in Section 2, we, in Section 3, review some techniques used in previous results. The rest of the paper then proves the security results for all individual RSA bits. Section 4 generalizes some well-known sampling techniques. In Section 5 we treat the case of the internal bits which is the essentially new case. We omit many technical discussions due to the space constraints. Finally, we briefly discuss an extension to prove security for the bits of the discrete logarithm.

## 2. Preliminaries

The model of computation used is that of probabilistic Turing machines running in time poly$(n)$ where $n$ is the length of the input, pptm for short. In general, $|y|$ denotes the length of the binary string $y$. If $S$ is a set, $\#S$ is the cardinality of $S$ and by $x \in_D S$ we mean an $x$ chosen at random according to the distribution $D$ on $S$, $U$ denotes the uniform distribution. If $T \subset S$, then $\lambda_S(T) \triangleq \#T/\#S$ is the standard uniform measure. (When $S$ is obvious from the context, we write $\lambda(T)$.)

We call a function $g(n)$ *negligible* if for every constant $c > 0$ and all sufficiently large $n$, $g(n) < n^{-c}$. A *one-way function* is a poly-time computable function $f$ such that for every pptm, $M$, the probability that $M(f(x)) \in f^{-1}(f(x))$ is negligible. The probability is taken over $x \in_U \{0,1\}^n$ and $M$'s random coin flips.

Let $f$ be a one-way function and let $b$ be a poly-time computable boolean function. An $\varepsilon(n)$-*oracle* for $b$ is a pptm $\mathfrak{O}$ for which $\Pr[\mathfrak{O}(f(x)) = b(x)] \geq \frac{1+\varepsilon(n)}{2}$, the probability taken over $x \in_U \{0,1\}^n$, and $\mathfrak{O}$'s random choices. The only interesting case is when $\varepsilon(n) > 0$. If no $\varepsilon(n)$-oracle exists, we call $b$ $\varepsilon(n)$-*secure for* $f$, and if $b$ is $\varepsilon(n)$-secure for all non-negligible $\varepsilon(n)$, we say that $b$ *is secure for* $f$.

For $m, z \in \mathbb{Z}$, $m > 0$, we write $[z]_m \triangleq z \bmod m$ and put $\text{abs}_m(z) \triangleq \min\{[z]_m, m - [z]_m\}$. If for some $\delta \in [0,1]$, $\text{abs}_m(z) \leq \delta m$, $z$ is said to be $\delta$-*small* (modulo $m$).

We use $E_N(x)$ to denote the RSA encryption function: $E_N(x) \triangleq [x^e]_N$ for $|N| = n$, $N = pq$, the product of two primes, and $e$, an integer relatively prime to $(p-1)(q-1)$.

For $z \in \mathbb{Z}$, $0 \leq i < |z|$, $\text{bit}_i(z)$ denotes the $i$th bit in the binary representation of $z$, $\text{bit}_i(z) \triangleq \lfloor z/2^i \rfloor \bmod 2$. To avoid confusion we stress that this means that the bits are numbered $0, 1, \ldots, |z| - 1$, "right-to-left". In particular $\text{lsb}(z) \triangleq \text{bit}_0(z)$. For $0 \leq i \leq j < |z|$, let $\mathrm{B}_i^j(z)$ denote bits $i, i+1, \ldots, j$ in the binary representation of $z$.

For a given $N$, and random $z$, the bits in $[z]_N$ are not uniformly distributed since the uniform distribution on $\mathbb{Z}_N$ is not the same as the uniform distribution on $\{0,1\}^{|N|}$. By the *bias* of the $i$th bit we mean the value $\beta_i(N)$ such that $\Pr_{z \in_U \mathbb{Z}_N}[\text{bit}_i(z) = 0] = \frac{1+\beta_i(N)}{2}$. It is an easy exercise to verify that always, $\beta_i(N) \leq \frac{2^i}{N}$. The bias is therefore only of significance for the $O(\log \log N)$ most significant bits.

Finally, let $D, D'$ be distributions on the same space $S$. We call $D, D'$ (polynomially) *distinguishable* if there is a pptm $D$ such that

$$\left| \Pr_{y \in_D S}[D(y) = 1] - \Pr_{y' \in_{D'} S}[D(y') = 1] \right|$$

is non-negligible.

## 3. Previous Work and Proof Outline

The security of the least significant bit in an RSA encrypted message has gained a lot of attention. The first result by Goldwasser, Micali, and Tong, was to prove a $1 - o(1)$-security result, [9]. This was improved to $\frac{1}{2} + o(1)$ by Ben-Or, Chor, and Shamir, see [3]. Further progress was accomplished by a more intricate sampling technique, and by an improved analysis of this technique, Vazirani and Vazirani, [19], and then Goldreich, [8], respectively, showed 0.464- and then 0.45-security. By improving the sampling techniques once again, Schnorr and Alexi, [17], proved $\varepsilon$-security for any constant $\varepsilon$ and the $\varepsilon(n)$-security for any non-negligible $\varepsilon(\cdot)$ was then proven by Chor and Goldreich, [6]. A simpler proof of $\varepsilon(n)$-security was recently given by Fischlin and Schnorr [7].

The results for the least significant bit generalizes in a straightforward way to any of the $O(\log n)$ least significant bits. For the internal bits of RSA however, the results so far are not very strong. The first appeared in [9], where it was shown that for each $i$, there *are* $N$ of very special form, for which the $i$th bit of $x$ cannot be computed without errors. In [3], it was proved that an oracle for the $i$th bit of RSA can be converted into an lsb-oracle, increasing the error probability by $\frac{1}{4}$ in the worst case. For every second bit position however, the error can be bounded by $\frac{3}{16}$. Hence, from their own result for the lsb, a $\frac{7}{8}$-security for half of the individual bits followed. All later progress in proving security for the lsb has then, via the reduction by Ben-Or et al, strengthened the provable security for the internal bits. Hence, the best result so far is the $\frac{1}{2} + o(1)$-security that follows from [1], still leaving a large gap to the desired $o(1)$ result. The provable security obtainable by these reductions depends on $N$ and the bit-position, $i$, considered, but for worst case $N$ and $i$, results better than $\frac{1}{2} + o(1)$ are impossible by such "standard" reductions. The extra $\frac{1}{4}$ that the reduction may add to the error probability is a tight bound.

As mentioned, Näslund showed in [13] that all bits in functions of the form $x \mapsto [ax+b]_p$, $p$ an $\Omega(n)$-bit prime, are $\varepsilon(n)$-secure with respect to *any* one-way function. However, one of the proofs in this paper contains a gap, and in fact, it was when trying to fill this gap that we realized that the methods could be applied to the RSA bits as well. There is a common property shared between RSA and $h(x) = [ax+b]_p$, namely the multiplicative properties; $E_N(cx) = [E_N(c)E_N(x)]_N$ and $[ch(x)]_p = [(ca)x+cb]_p$. That is, even if $x$ is unknown, given $E_N(x)$ one can compute $E_N(cx)$, and given $h(x)$, $ch(x)$ can be found as $h'(x)$, another function of the same type. This property is used extensively to obtain the RSA results and also in [13].

In general, how does one prove that some function $b(x)$ is $\varepsilon(n)$-secure for some one-way function $f(x)$? The most natural way is to do the proof by contradiction, proving that if an $\varepsilon(n)$-oracle for $b(x)$ exists, then this oracle can be used in a black-box fashion to retrieve $x$, i.e. to invert $f(x)$. In our specific case, $f(x) = E_N(x)$, RSA encryption. Since RSA is not known to be one-way, the best we can achieve is therefore to show that computing $b(x)$ (here $b(x) = \text{bit}_i(x)$) is as hard as inverting RSA. Let us first review the methods used in [1].

### 3.1. The Method by Alexi et al. [1]

Alexi, Chor, Goldreich, and Schnorr used the following procedure. (In fact, already in [3], the same basic principle was used.) There is a well-known algorithm for computing integer gcd (greatest common divisor) using only parity tests, i.e. lsb-tests. Using RSA's multiplicative properties, it is not hard to see how a reliable oracle for the least significant bit can be used to invert RSA by picking random $a, b$, computing $\gcd([ax]_N, [bx]_N)$, giving us a value of the form $[cx]_N$ with $c$ known. For details, and how to improve the reliability of the $\varepsilon(n)$-oracle we refer the reader to [1]. It is there also shown that as "reliable", an error probability of at most $\frac{1}{2(6\log N+3)}$ is sufficient. For technical reasons, we note that it is also necessary that $ax, bx$ are "small" modulo $N$.

It turns out that it is not necessary to have an lsb-oracle to make the above procedure work. This was noted in [3]. Let an *interval* $J \subset [0,N)$ denote a set of consecutive integers in $\mathbb{Z}_N$ and for $z \in \mathbb{Z}$, we translate $J$ by $J+z \triangleq \{[y+z]_N \mid y \in J\}$. Suppose that for some not too short interval $J$, we have an oracle that, when given $E_N(z)$, is somewhat more likely to answer "1" when $z \in J$ than when $z \in J+(N+1)/2$. Now ask this oracle about $E_N([2^{-1}x]_N)$. We see that

$$\begin{aligned} [2^{-1}x]_N &= \frac{x-\text{lsb}(x)}{2} + \text{lsb}(x)[2^{-1}]_N \\ &= \frac{x-\text{lsb}(x)}{2} + \text{lsb}(x)\frac{N+1}{2}, \qquad (3.1) \end{aligned}$$

Hence, if $\frac{x-\text{lsb}(x)}{2} \in J$, then $[2^{-1}x]_N \in J + \text{lsb}(x)(N+1)/2$. Since the oracle "behaves" differently on $J$, $J+(N+1)/2$, there is some hope to determine the lsb by querying the oracle. (See also Figure 1.)

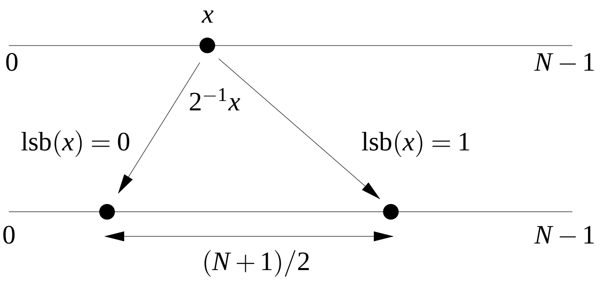

**Figure 1. Division by $2$ in $\mathbb{Z}_N$.**

### 3.2. Näslund's Method, [13]

Here the objective was to use an oracle for the $i$th bit in the function $x \mapsto [ax+b]_p$, $p$ an $\Omega(n)$-bit prime and $a, b$ random elements in $\mathbb{Z}_p$, to retrieve $x$. This was done by first finding $y = [ax+b]_p$ and then $x$ as $x = [a^{-1}(y-b)]_p$.

To handle the internal bits, the main idea in [13] was to convert the oracle for the $i$th bit into an oracle that computed *both* the lsb *and* the $i+1$st bit, creating a two-bit window that by manipulating $a, b$ through multiplications can be made to "slide" over all the bits in $[ax+b]_p$, see Fig. 2.

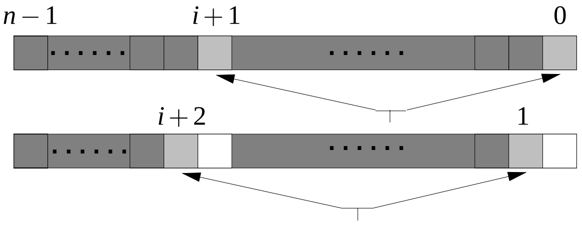

**Figure 2. Deciding bits two-by-two.**

As mentioned, a closer study of [13] reveals that the methods in fact do not apply for some "highly structured" oracles that behave in a certain way. On the other hand, the oracles for which the methods fail are of a *very* special nature that we can exploit. We mentioned above that the tools in [3] can not be used to prove stronger security than $\frac{1}{2}$ for *general* $N$ and general oracles. The plan is now: (a) Investigate how, and when, the methods in [13] are applicable to prove bit security for RSA. (b) Show that when those methods fail, we can deduce that a certain relation between $N$ and $2^{i+1}$ holds ($i$ is the bit position predicted by the oracle), and furthermore, the oracle must then have a certain structure. (c) Prove that when Näslund's method fails, (so that we are in case (b)), this makes it possible to construct an algorithm, i.e. a new oracle, $\mathfrak{O}'$, using the original oracle $\mathfrak{O}$ as a black box, such that $\mathfrak{O}'$ is an lsb-oracle. That is, *either* the methods from [13] work *or* the methods in [3] can be refined to prove the desired result.

We start by giving some generalizations of well-known sampling techniques and then formalize how the gcd method is used as a “warm-up”. We then treat (a), (b), and (c).

## 4. Sampling Techniques

Throughout the paper, $i$ is reserved to denote the bit-position predicted by the oracle. We assume that we have an oracle $\mathfrak{O}$ that given $E_N(x)$, $|N| = n$, predicts the $i$th bit of $x$ with probability at least $\frac{1+\varepsilon(n)}{2}$ where $\varepsilon(n)$ is non-negligible.

This section gives results on how to generate lists of pairwise independent random points for which some properties are known. These lists are later used to derive oracles with strong distinguishing capabilities. The basic techniques are well known and we therefore omit the proofs.

**Definition 4.1.** By an *interval*, $J$, we mean a set of consecutive values $J = [u,v] \triangleq \{u, u+1, \ldots, v\}$ in $\mathbb{Z}_N$. (We allow the intervals to wrap around $N$.) The *length* of $J$ is $\#J$ and the *measure* is $\lambda(J) \triangleq \#J/N$. If $J$ is an interval and $z \in \mathbb{Z}_N$, denote by $J + z \triangleq \{[y+z]_N \mid y \in J\}$.

For a distribution $D$ on an arbitrary subset $J \subset \mathbb{Z}_N$, let $P_D^{\mathfrak{O}}(J)$ be the fraction of 1-answers the oracle gives on $D$:

$$P_D^{\mathfrak{O}}(J) \triangleq \mathrm{E}_{z \in_D J}[\mathfrak{O}(E_N(z))] = \Pr_{z \in_D J}[\mathfrak{O}(E_N(z)) = 1].$$

If $D$ is the uniform distribution on $J$, we omit it from the notation, and furthermore, we then also define for $J_1, J_2 \subset \mathbb{Z}_N$: $\Delta^{\mathfrak{O}}(J_1, J_2) \triangleq \left|P^{\mathfrak{O}}(J_1) - P^{\mathfrak{O}}(J_2)\right|$.

We use the oracle to distinguish between two distributions on $\mathbb{Z}_N$ which have support on two intervals, and when some bit (or bits) of $x$ is 0, we sample according to the first distribution, and when the bit is 1 we have the other. By sampling the oracle, we can distinguish these two distributions, i.e. decide one (or more) bits of $x$. To make sure that we hit one of these two subsets when sampling, we can actually arrange things so that we know in advance the approximate locations in $\mathbb{Z}_N$ of the sample points.

**Lemma 4.2.** Let $m(n) \in \mathrm{poly}(n)$, $d_I(n), d_Y(n) \in O(\log n)$. Then, given $r, s \in_U \mathbb{Z}_N$, it is in deterministic polynomial time possible to generate a list of $m(n)$ values of the form $E_N(r_j x)$ so that each $[r_j x]_N$ is uniformly distributed and the values in $\{[r_j x]_N\}$ are pairwise independent. Furthermore, we generate a set consisting of $2^{2(d_I(n)+d_Y(n))} m(n)^2$ pairs of lists, $\{(L^I, L^Y)\}$, each $L^I$ consisting of $m(n)$ values in $\mathbb{Z}_{2^{i+1}}$ and each $L^Y$ of $m(n)$ values in $\mathbb{Z}_N$.

For at least one $(L', L'') \in \{(L^I, L^Y)\}$, for each $j = 1, \ldots, m(n)$, we have

$$\mathrm{abs}_N([r_j x]_N - L''_j) \le \frac{N}{2^{d_Y(n)}} \qquad (4.1)$$

and except with probability $2^{-(d_Y(n)-1)}$,

$$\mathrm{abs}_{2^{i+1}}([r_j x]_N - L'_j) \le 2^{i+1-d_I(n)}. \qquad (4.2)$$

The reader is encouraged to compare this lemma to [1, §4.4]. There, it was only necessary to know the lsb of each $[r_j x]_N$.

## 5. Security of Non Leftmost Bits

In this section, we consider $i$ such that

$$\tau(n) + \log\varepsilon(n)^{-4} + \log n + 33 \le i \le n - 3\tau(n) - \log\varepsilon(n)^{-1} - 7$$

where $\tau(n) \triangleq 32 + \log\varepsilon(n)^{-4} + \log n$. We impose these restrictions on $i$ for technical reasons. The case $i < \tau(n) + \log\varepsilon(n)^{-4} + \log n + 33$ is treated in [1] and the result for $i > n - 3\tau(n) - \log\varepsilon(n)^{-1} - 7$ follows from a simple reduction from the same result.

### 5.1. RSA inversion, Method 1 (gcd)

The main result here is the following lemma, slightly generalizing results in [1, 3].

**Lemma 5.1.** If $\mathfrak{O}$ is such that for some interval $J$ we have $\Delta^{\mathfrak{O}}(J, J + (N+1)/2) \ge \varepsilon'(n)$, where $\lambda(J), \varepsilon'(n)$ are non-negligible, then we can in random polynomial time construct an oracle, $\mathfrak{O}'$ such that for all $\frac{\lambda(J)\varepsilon'(n)}{48}$-small $[ax]_N$, $\mathfrak{O}'$ determines $\mathrm{lsb}([ax]_N)$ with probability at least $1 - \frac{1}{12n+6}$.

The proof is straightforward and uses only well-known sampling techniques. In particular, (4.1) of Lemma 4.2 is used to generate $\Theta(\lambda(J)^{-1} n \varepsilon'(n)^{-2})$ sample points, $\{E_N(r_j x)\}$. With high accuracy we can decide if, depending on $\mathrm{lsb}(ax)$, $E_N([(r_j + 2^{-1}a)x]_N)$ falls in $J$, in $J + (N+1)/2$, or in none of them. This is sufficient to distinguish between the intervals, thereby determining the lsb. We omit further details. Notice that such an oracle, $\mathfrak{O}'$, is all we need to make the gcd inversion algorithm from [1] work.

**Discussion.** We are not able to prove that for any oracle $\mathfrak{O}$, an interval $J$ as above exists. Nevertheless, let us investigate some circumstances, under which we could hope for such an oracle.

If there are no good intervals at distance $(N+1)/2$, then the oracle describes a function $\mathfrak{O} : \mathbb{Z}_N \to [0,1]$ that is almost perfectly $(N+1)/2$-periodic. By assumption, $\mathfrak{O}$ is correlated with the $i$th bit, i.e. it is (at least to some extent) a $2^{i+1}$-periodic function. Intuitively, the only way that a function can be both (almost) $(N+1)/2$-periodic *and* $2^{i+1}$-periodic, is if $(N+1)/2$ and $2^{i+1}$ have integer multiples that are "close", $s(N+1)/2 \approx r2^{i+1}$. This observation suggests a closer study of such relations.

Let us now try to make use of what we know, i.e. that the function $\mathfrak{O} : \mathbb{Z}_N \to [0,1]$ has a structure on its domain modulo $2^{i+1}$. Rather than just looking at how the oracle behaves for values at distance $(N+1)/2$ (which is huge in comparison to $2^{i+1}$), we look closer at the "modulo $2^{i+1}$-behavior". To simplify the presentation, let us for the moment assume that $[x]_{2^{i+1}}$ is known, up to some "small" error. We can extend Equation (3.1) as follows:

$$\begin{aligned}[2^{-1}x]_N \quad = \quad & \frac{x - 2^{i+1}\mathrm{bit}_{i+1}(x) - \mathrm{lsb}(x)}{2} \\ & + 2^i \mathrm{bit}_{i+1}(x) + \mathrm{lsb}(x)\frac{N+1}{2}. \quad (5.1)\end{aligned}$$

Since $x$ is "known" modulo $2^{i+1}$, so is the term $\frac{x-2^{i+1}\mathrm{bit}_{i+1}(x)-\mathrm{lsb}(x)}{2}$. Besides the influence of the lsb, we can expect the oracle's behavior on $[2^{-1}x]_N$ to also depend on the $i+1$st bit of $x$, since this bit "falls into" position $i$ which we know is of significance to the oracle. This suggests that we could try to decide both the lsb *and* the $i+1$st bit of $x$ by querying the oracle on $[2^{-1}x]_N$. To do this, two effects must be considered: On the small scale, modulo $2^{i+1}$, the important term is $2^i\mathrm{bit}_{i+1}(x) + \mathrm{lsb}(x)[\frac{N+1}{2}]_{2^{i+1}}$, and on the large scale, modulo $N$, the term $\mathrm{lsb}(x)\frac{N+1}{2}$ may be of importance.

It now becomes natural to view $\mathbb{Z}_N$, not as a 1-dimensional interval of length $N$, but rather as a 2-dimensional plane, one dimension corresponding to projections modulo $2^{i+1}$, the other corresponding to the rough size modulo $N$ (as before). We represent each $z \in \mathbb{Z}_N$ uniquely by $(z \bmod 2^{i+1}, \lfloor z/2^{i+1} \rfloor)$. When sampling the oracle, we now need to know in advance, not only the sample points approximate location in $\mathbb{Z}_N$, but also the approximate location modulo $2^{i+1}$. This explains the need for (4.2) of Lemma 4.2.

In the next section, we will see that at least in cases where a generalized (and quantified) version of the constraint $s(N+1)/2 \approx r2^{i+1}$ does not hold, we are able to carry out this program. In the cases where we cannot determine the bits two by two, this relation between $N$ and $2^{i+1}$ is shown to hold, and in addition, the oracle has a special property. For such oracles and such $N$, the gcd method outlined above can be made to work.

### 5.2. RSA inversion, Method 2 (Näslund)

This second method is much more technical than the previous, and we start by outlining the ideas. This method follows the principles used in [13].

As mentioned, the idea is to use the oracle for the $i$th bit to decide *both* the lsb *and* the $i+1$st bit. Suppose that we have set things up so that we already know the value of $\mathrm{B}^i_{i-d+1}(x)$, the value of the $d$ bits to the right of, and including bit $i$. (If $d$ is small enough we can initially simply guess this value.) Instead of asking the oracle on $E_N([2^{-1}x]_N)$, we use $E_N([2^{-\tau}x]_N)$ where $1 < \tau \ll i$. (Why $\tau > 1$ is a good idea is explained shortly.) We make a list of all $2^{2\tau}$ possibilities for bits $i+1,\ldots,i+\tau$, and bits $0,\ldots,\tau-1$ in $x$, i.e, for $\mathrm{B}^{i+\tau}_{i+1}(x)$ and $\mathrm{B}^{\tau-1}_0(x)$. Hence, an entry in this list looks like $(u_1, v_1)$, $0 \le u_1, v_1 \le 2^\tau - 1$, $u_1$ corresponding to a possibility for $\mathrm{B}^{i+\tau}_{i+1}(x)$ and $v_1$ to a possibility for $\mathrm{B}^{\tau-1}_0(x)$. The two bits we are after, $\mathrm{bit}_{i+1}(x)$ and $\mathrm{lsb}(x)$ then corresponds to $\mathrm{lsb}(u_1)$ and $\mathrm{lsb}(v_1)$, respectively.

Take any two distinct candidates from the list $(u_1, v_1)$ and $(u_2, v_2)$. Surely, they cannot both be correct, so we try to exclude one of them (the incorrect one if one *is* correct). Furthermore, since we only try to determine the two bits $\mathrm{bit}_{i+1}(x)$ (corresponding to $\mathrm{lsb}(u_1), \mathrm{lsb}(u_2)$) and $\mathrm{lsb}(x)$ (corresponding to $\mathrm{lsb}(v_1), \mathrm{lsb}(v_2)$), we are only interested in pairs $(u_1, v_1)$, $(u_2, v_2)$ for which $\mathrm{lsb}(u_1) \neq \mathrm{lsb}(u_2)$ *or* $\mathrm{lsb}(v_1) \neq \mathrm{lsb}(v_2)$.

We now attempt to distinguish between the two choices by "shifting" $x$ by $\tau$ steps to the right, computing $[2^{-\tau}x]_N$. What happens? Because we know the value of $\mathrm{B}^i_{i-d+1}(x)$, the interesting bits that have effect on the $i$th bit in $[2^{-\tau}x]_N$ are the $\tau$ bits to the left of bit $i$, bits $i+1,\ldots,i+\tau$, since they are shifted to the right, passing position $i$, and the $\tau$ least significant bits, $0,1,\ldots,\tau-1$, since these cause wraparound modulo $N$ when shifted:

$$\begin{aligned}[2^{-\tau}x]_N \;=\; & \frac{x - \mathrm{B}^{i+\tau}_{i+1}(x)2^{i+1} - \mathrm{B}^i_{i-d+1}(x)2^{i-d+1} - \mathrm{B}^{\tau-1}_0(x)}{2^\tau} \\ & + \mathrm{B}^{i+\tau}_{i+1}(x)2^{i+1-\tau} + \mathrm{B}^i_{i-d+1}(x)2^{i-d+1-\tau} \\ & + \mathrm{B}^{\tau-1}_0(x)[2^{-\tau}]_N. \qquad (5.2)\end{aligned}$$

(C.f. equation (5.1).) The term $x - \mathrm{B}^{i+\tau}_{i+1}(x)2^{i+1} - \mathrm{B}^i_{i-d+1}(x)2^{i-d+1} - \mathrm{B}^{\tau-1}_0(x)$ is divisible (as an integer) by $2^\tau$, and it has $d$ consecutive zeros to the right of bit $i$, so it is very small mod $2^{i+1}$. Hence, $\mathrm{B}^{i+\tau}_{i+1}(x)2^{i+1-\tau} + \mathrm{B}^{\tau-1}_0(x)[2^{-\tau}]_N$ is essentially the only *unknown* term that influences the $i$th bit in $[2^{-\tau}x]_N$.

Now let us try to decide if $\mathrm{B}^{i+\tau}_{i+1}(x) = u_1$ or $u_2$ and if $\mathrm{B}^{\tau-1}_0(x) = v_1$ or $v_2$. We would like to tell if $[2^{-\tau}x]_N$ is of the form $z' + u_1 2^{i+1-\tau} + v_1[2^{-\tau}]_N$ or, of the form $z' + u_2 2^{i+1-\tau} + v_2[2^{-\tau}]_N$, and this is the same as distinguishing between values of the form $z$ and $z + u2^{i+1-\tau} + v[2^{-\tau}]_N$, where $u = u_2 - u_1$, $v = v_2 - v_1$. We may assume that $v > 0$ (otherwise swap $(u_1, v_1)$ and $(u_2, v_2)$). Still, we may have $u < 0$. If this happens, replace $u$ by $u + 2^\tau$. Note that

$$(2^\tau + u)2^{i+1-\tau} = 2^{i+1} + u2^{i+1-\tau},$$

so the only difference by doing this replacement is a single multiple of $2^{i+1}$ which should not matter to the oracle (this can be shown formally). Hence, we have $0 \le u, v \le 2^\tau - 1$ and because at least one of the pairs $(u_1, u_2)$, $(v_1, v_2)$ differs in their least significant bit, at least one of $u, v$ is odd. If we assume that $z$ belongs to some subset $S \subset \mathbb{Z}_N$, then $[2^{-\tau}x]_N \in S$ if $(u_1, v_1)$ is correct and $[2^{-\tau}x]_N \in S + u2^{i+1-\tau} + v[2^{-\tau}]_N$ if $(u_2, v_2)$ is correct. We now make the following definition:

**Definition 5.2.** For given $N, \tau$ and $0 \le u, v \le 2^\tau - 1$, define

$$\alpha_N^\tau(u,v) \triangleq u2^{i+1-\tau} + v[2^{-\tau}]_N.$$

Note that $\alpha_N^\tau(u,v)$ is computed modulo $N$, not modulo $2^{i+1}$. Again, we are only interested in $\alpha_N^\tau(u,v)$ where at least one of $u, v$ is odd, so there are $\frac{3}{4}2^{2\tau}$ such values.

Just like we in the previous section wanted to find sets $J$, $J + (N+1)/2 = J + [2^{-1}]_N$, where the oracle behaved differently, we can now ask if there are similar sets $S, S + \alpha_N^\tau(u,v)$ where the oracle behaves differently. Consider the case when $v$ is odd. (The case when $v$ is even is “easy”, and we return to this later.) There are then $2^\tau$ distinct values of the form $k\alpha_N^\tau(u,v)$, $k = 0, 1, \ldots, 2^\tau - 1$, and one can hope that for at least one of these $k$’s, the oracle distinguishes between some $S + k\alpha_N^\tau(u,v)$ and $S + (k+1)\alpha_N^\tau(u,v)$. (When $k = 2^\tau$, $[k\alpha_N^\tau(u,v)]_N = u2^{i+1} + v$, which in turn is $v$ modulo $2^{i+1}$. Since $v$ is small and the oracle predicts the $i$th bit, as far as the oracle is concerned, we are then “back where we started”. When $\tau = 1$ there are therefore essentially only two possible multiples of $\alpha_N^1(u,v)$, so this explains why we use $\tau > 1$.) Now, *if* we can find good interval pairs for *all* these $\alpha_N^\tau$-values, we seem to be in good shape. This seems unlikely at first—we could not even argue that there must be a single good interval pair $J, J + (N+1)/2$. On one hand, we are better off now, because we are free (within some range) to choose $k, \tau$ as we wish. Then again, we now need an exponential number (in $\tau$) of such interval pairs.

Consider a particular $(u,v)$ and fix $S \subset \mathbb{Z}_N$ so that all $z \in S$ have the same value for their $i$th bit. We can thus not let $S$ be an interval as before, since the length $S$ would then be bounded by $2^i$, which is negligible compared to $N$. Instead, we take $S$ as a union of short intervals, each at distance $2^{i+1}$, i.e. $S = \bigcup_l (J' + l2^{i+1})$ where $J'$ is a “traditional” interval of length at most $2^i$ and the range of $l$ is chosen suitably so that the measure of the set $S$ is non-negligible. We will shortly see that we can view these $S$ as two-dimensional intervals, or “boxes”, in a nice way.

For such an $S$, consider now the sets

$$S_k = S + k\alpha_N^\tau(u,v), \qquad 0 \le k \le 2^\tau - 1.$$

If for some $k$, $\Delta^{\mathfrak{O}}(S_k, S_{k+1})$ is non-negligible we are done, so suppose this is not the case. Consider next all translations of $\{S_k\}$:

$$S_{j,k} = S + k\alpha_N^\tau(u,v) + j(\#J'), \qquad \begin{array}{l} 0 \le k \le 2^\tau - 1, \\ 0 \le j \le \lfloor 2^{i+1}/\#J' \rfloor. \end{array}$$

If $S$ is chosen suitably, these sets can be made to cover $\mathbb{Z}_N$ very nicely. If, in addition, $\{k\alpha_N^\tau(u,v)\}_{k=0}^{2^\tau-1}$ is nicely distributed modulo $2^{i+1}$ then there must be a fixed $j$ such that the oracle does not behave the same on some $S_{j,k}$, $S_{j,k+1}$, otherwise the oracle cannot be correlated with the $i$th bit. Hence the key property is that of the distribution of the sequence $\{k\alpha_N^\tau(u,v)\}_{k=0}^{2^\tau-1}$ modulo $2^{i+1}$. For odd $v$, we define a new quantity that is later shown to be intimately related to $\alpha_N^\tau(u,v)$.

**Definition 5.3.** For $0 \le u, v \le 2^\tau - 1$, $v$ odd, define

$$\tilde{\alpha}_N^\tau(u,v) \triangleq [-uv^{-1}N]_{2^\tau} 2^{i+1-\tau} + \left\lceil \frac{N}{2^\tau} \right\rceil.$$

We show that unless a small integer multiple of $\tilde{\alpha}_N^\tau(u,v)$ is close to a multiple of $2^{i+1}$, then the sequence $\{k\tilde{\alpha}_N^\tau(u,v)\}_{k=0}^{2^\tau-1}$ is indeed “equidistributed” mod $2^{i+1}$, and this can be shown to imply the existence of good boxes at distance $\alpha_N^\tau(u,v)$. The only problem is now when for some $u, v, r$ ($v$ odd) and “small” $s \in \mathbb{Z}$,

$$\left| \frac{\tilde{\alpha}_N^\tau(u,v)}{2^{i+1}} - \frac{r}{s} \right|$$

is “small”.

We cannot rule out the possibility that such $u, v, s, r$ exist, but the good news is that when they do, *and* the original oracle behaves the same on all $S, S + \alpha_N^\tau(u,v)$, then we can construct an oracle and find an interval $J$ so that this new oracle behaves differently on $J$ and $J + (N+1)/2$.

**Definition 5.4.** In the remainder of the paper we write $N$ as $N \triangleq N_1 2^{i+1} + N_0$ where $0 \le N_0 < 2^{i+1}$. We sometimes also study $N_1$ closer, and it is convenient to write $N_1$ as $N_1 \triangleq N_3 2^{\tau(n)} + N_2$ where $0 \le N_2 < 2^{\tau(n)}$.

**Definition 5.5.** Let $I \triangleq \mathbb{Z}_{2^{i+1}}$ and $Y \triangleq \mathbb{Z}_{N_1+1}$. We can view $\mathbb{Z}_N$ as a subset of $I \times Y$ by defining the natural projection $\pi : \mathbb{Z}_N \to I \times Y$ by

$$\pi(z) = (\pi_I(z), \pi_Y(z)) \triangleq (z \bmod 2^{i+1}, \lfloor z/2^{i+1} \rfloor).$$

Note that $\pi$ is surjective, except possibly for some values of the form $(j, N_1)$ with $j \ge N_0$. We define the *plane* $\Pi(N,i) = (I \times Y) \cap \pi(\mathbb{Z}_N)$.

For all non-negative integers $z_0, y_0, w, h$ that satisfy $2^{i+1}(y_0 + h - 1) + z_0 + w - 1 < N$ we define a *box*, $S$, of *width* $w$ and *height* $h$ as the following rectilinear subset of $I \times Y$:

$$\{\pi(z + 2^{i+1}y) \mid z_0 \le z < z_0 + w, y_0 \le y < y_0 + h\}.$$

That is, should $z_0 + w \geq 2^{i+1}$, we allow wrap around the edges of $I$ in the natural way, but wrap in the $Y$-direction is not allowed for technical reasons. The *measure* of such a box is simply $\lambda(S) \triangleq \frac{\#S}{N} = \frac{wh}{N}$. Furthermore, for a box $S$ and $z \in \mathbb{Z}_N$ such that $\max_{(z',y')\in S}(2^{i+1}y' + z' + z) < N$, we define the *z-translation* of $S$ as

$$S + z \triangleq \{(\pi_I(z'+z), \pi_Y(y'+z)) \mid (z',y') \in S\}.$$

That is, the operation is defined unless the box is forced to wrap in the $Y$-direction. A *level* is a subset of $\Pi(N,i)$ consisting of the set of values having a fixed $\pi_Y$-value. Finally, we remind the reader of Definition 4.1 for the definitions of $P_D^{\mathfrak{O}}(S)$, $\Delta^{\mathfrak{O}}(S,S')$ for boxes $S,S'$.

Figure 3 below illustrates the plane.

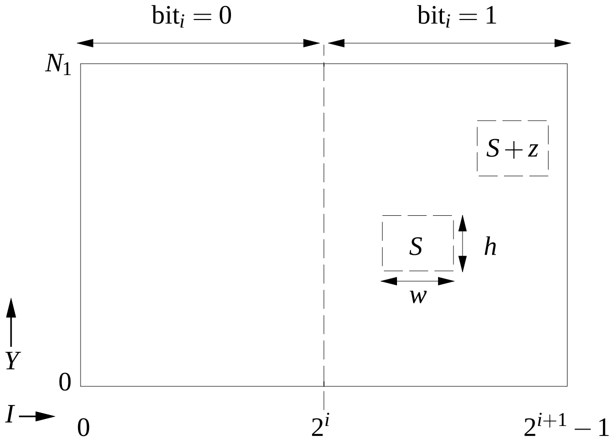

**Figure 3. The $\Pi(N,i)$-plane. Shown is a typical box, $S$, and a translation, $S+z$. Half-planes correspond to fixed values for the $i$th bit.**

Before discussing the existence of "good" boxes, let us assume that we have all the boxes we need, and see how we use them to invert RSA.

**Lemma 5.6.** Suppose that for all $0 \leq u,v \leq 2^{\tau(n)} - 1$, $u$ or $v$ odd, there is a box $S_{u,v}$ of width $w(n)2^{i+1}$, height $h(n)N_1$, and with $\Delta^{\mathfrak{O}}(S_{u,v}, S_{u,v} + \alpha_N^{\tau(n)}(u,v)) \geq \varepsilon'(n)$, where $h(n), w(n), \varepsilon'(n)$ are all non-negligible. Define $d_Y(n) \triangleq \log \varepsilon'(n)^{-1} + \log(w(n)h(n))^{-1} + 9$, $d_I(n) \triangleq \log \varepsilon'(n)^{-1} + \log w(n)^{-1} + 8$. Then it is possible to construct an oracle, $\mathfrak{O}'$, that given $E_N(x)$, $j$, $\mathrm{B}_{i-d_I(n)+1}^{i+j}(x)$, $\mathrm{B}_0^{j-1}(x)$, and $y$ so that $\mathrm{abs}_N(x-y) \leq 2^{-d_Y(n)}N$, for any $0 \leq j \leq \max(n-i-2, i)$, determines $\mathrm{bit}_{i+j+1}(x)$ *and* $\mathrm{bit}_j(x)$ with probability at least $1 - \frac{1}{2n}$.

**Lemma 5.7.** If there is an oracle $\mathfrak{O}$ satisfying the conclusion of Lemma 5.6, then we can invert RSA in random polynomial time with probability of success at least $\frac{1}{2}$.

*Proof.* By trying all possibilities for $\mathrm{B}_{i-d_I(n)+1}^{i}(x)$ and $y$ so that $\mathrm{abs}_N(x-y) \leq 2^{-d_Y(n)}N$ we can now by repeated application of Lemma 5.6, first decide $\mathrm{bit}_0(x), \mathrm{bit}_{i+1}(x)$, then $\mathrm{bit}_1(x), \mathrm{bit}_{i+2}(x)$ etc. Since $\mathfrak{O}$ is used at most $n$ times, the total error probability is at most $n\frac{1}{2n} = \frac{1}{2}$. □

In the remainder of this section, we find sufficient conditions for the existence of the good boxes.

**The Existence of Good Boxes Needed in Lemma 5.6.** This section deals with how certain relations between $\tilde{\alpha}_N^{\tau(n)}(u,v)$ and $2^{i+1}$ are of significance for the existence of good boxes as distance $\alpha_N^{\tau(n)}(u,v)$.

**Definition 5.8.** The number $\zeta \in \mathbb{Q}$ is said to be of $(Q,\psi)$-*type* if for all integers $r,s$, $0 < s \leq Q$, $(r,s) = 1$:

$$\left|\zeta - \frac{r}{s}\right| > \frac{1}{s^2\psi}.$$

(That is, the distance from $s\zeta$ to the nearest integer is at least $\frac{1}{s\psi}$.)

It is well known that there are no $\zeta$ of $(Q,1)$-type (even if we allow $\zeta$ to be irrational), and if $\zeta$ is of the form $\alpha/2^{i+1}$ (which is interesting for us) and $\alpha$ is *random*, then it is easy to show that the probability that $\zeta$ is of $(Q,\psi)$-type is $1 - O(\frac{\log Q}{\psi})$, provided $2^{i+1} \geq Q^2\psi$.

Why is type a useful concept? The famous *Weyl equidistribution theorem* states that if $\zeta$ is *irrational*, the fractional parts of the sequence $\{j\zeta\}_{j=0}^{K-1}$ is uniformly distributed in $[0,1]$ in the sense that as $K \to \infty$, each $[a,b] \subset [0,1]$, gets about the "correct" number of points from the sequence, i.e. a $b-a$ fraction. The rate of convergence to the uniform distribution depends on a similar notion of type for irrational $\zeta$. We want to show that (at least for some $u,v$), $\{j\alpha_N^{\tau(n)}(u,v)\}_{j=0}^{2^{\tau(n)}-1}$ is nicely distributed modulo $2^{i+1}$. There should be a close correspondence between this sequence modulo $2^{i+1}$ and the sequence $\{j\zeta\}_{j=0}^{2^{\tau(n)}-1}$ modulo 1, where $\zeta = \alpha_N^{\tau(n)}(u,v)/2^{i+1}$, so we look for a quantitative version of the Weyl-theorem for rational $\zeta$.

However, notice that the sequence $\{j\alpha_N^{\tau(n)}(u,v)\}_{j=0}^{2^{\tau(n)}-1}$ is really a sequence modulo $N$. So, we cannot consider it as an arbitrary integer sequence modulo $2^{i+1}$ as reductions modulo $N$ occur first. It is here that we switch to studying $\tilde{\alpha}_N^{\tau(n)}(u,v)$ instead. We encourage the reader to verify that by definition, the sequence $\{j\tilde{\alpha}_N^{\tau(n)}(u,v)\}_{j=0}^{2^{\tau(n)}-1}$ does not wrap modulo $N$ and can therefore be treated as an integer sequence. It is not too hard to show (we omit the proof) that $\{j\tilde{\alpha}_N^{\tau(n)}(u,v)\}$ is, except for small integer multiples of $2^{i+1}$, just a permutation of the set $\{j\alpha_N^{\tau(n)}(u,v)\}$.

So, we can now consider the rational $\zeta_{u,v} = \tilde{\alpha}_N^{\tau(n)}(u,v)/2^{i+1}$, and there are three cases in which we can show that there are good boxes at distance $\alpha_N^{\tau(n)}(u,v)$.

1. $v = 2v'$ is even. Then $\tilde{\alpha}_N^{\tau(n)}(u,v)$ is actually not defined. But we can study $\alpha_N^{\tau(n)}(u,v)$ directly, and since now, $u$ must be odd, we see that

   $$\begin{aligned} 2^{\tau(n)-1}\alpha_N^{\tau(n)}(u,v) &= 2^{\tau(n)-1}u2^{i+1-\tau(n)} \\ &\quad +2^{\tau(n)-1}2v'[2^{-\tau(n)}]_N \\ &= u2^i + v' \bmod N \approx u2^i, \end{aligned}$$

   since $v'$ is "small". We know that the oracle is correlated with the $i$th bit, so since $u$ is odd there should now be good boxes at distance $2^{\tau(n)-1}\alpha_N^{\tau(n)}(u,v) \approx u2^i$, and $2^{\tau(n)-1} \in O(\text{poly}(n))$, so there should also be good boxes at distance $\alpha_N^{\tau(n)}(u,v)$.

2. $v$ is odd and $\zeta_{u,v}$ is of bad type, but $\zeta_{u,v}$ is close to some $r/s$, $(r,s) = 1$, and where $s$ is *even*. This case can be treated very similarly to case 1.

3. $v$ is odd and $\zeta_{u,v}$ is of good type, i.e. inapproximable by small rationals. Then we win because we can prove that the sequence $\{j\tilde{\alpha}_N^{\tau(n)}(u,v)\}_{j=0}^{2^{\tau(n)}-1}$ is nicely distributed modulo $2^{i+1}$.

The three cases are considered in Propositions 5.9, 5.11, and 5.12, respectively. (The proofs are omitted.)

**Proposition 5.9.** For $0 \le u,v \le 2^{\tau(n)}-1$ with $u$ odd, $v$ even, there is a box $S$ of width $2^i$, height at least $N_1 - 2^{\tau(n)-1}$, so that

$$\Delta^{\mathfrak{O}}(S, S+\alpha_N^{\tau(n)}(u,v)) \ge \frac{\varepsilon(n)}{2^{\tau(n)+1}}.$$

We now investigate how the type of the number $\frac{\tilde{\alpha}_N^{\tau(n)}(u,v)}{2^{i+1}}$ is related to the existence of good boxes.

**Definition 5.10.** Define

$$Q(n) \triangleq 2^{10}\varepsilon(n)^{-1}, \qquad \psi(n) \triangleq \frac{\varepsilon(n)2^{\tau(n)}}{2^{12}\log^2 Q(n)}.$$

We first have the case when the type is "bad", but we have a good approximation with an even denominator.

**Proposition 5.11.** If for $0 \le u,v \le 2^{\tau(n)}-1$, $v$ odd, there are relatively prime integers $r,s$, $0 < s \le Q(n)$ and $s$ even, so that

$$\left|\frac{\tilde{\alpha}_N^{\tau(n)}(u,v)}{2^{i+1}} - \frac{r}{s}\right| \le \frac{1}{s^2\psi(n)},$$

then there is a box $S$ of width $2^i$, height $\ge \frac{N_1}{2}$ and with

$$\Delta^{\mathfrak{O}}(S, S+\alpha_N^{\tau(n)}(u,v)) \ge \frac{\varepsilon(n)}{2^{\tau(n)+3}s}.$$

Next, the case when the type is "good".

**Proposition 5.12.** For $0 \le u,v \le 2^{\tau(n)}-1$, $v$ odd, such that $\tilde{\alpha}_N^{\tau(n)}(u,v)/2^{i+1}$ is of $(Q(n),\psi(n))$-type, there is a box, $S$, of height at least $N_3 - 1$, width $2^{i+1}\varepsilon(n)/80$ and

$$\Delta^{\mathfrak{O}}(S, S+\alpha_N^{\tau(n)}(u,v)) \ge \frac{\varepsilon(n)}{2^{2\tau(n)+2}}.$$

The proof relies on results on uniform distribution of sequences, such as the Erdős-Turán Theorem, see [11], pp. 112–114, and methods similar to the proof of Lemmas 3.2, 3.3 in [11].

The only remaining difficulty is now when $v$ is odd *and* $\tilde{\alpha}_N^{\tau(n)}(u,v)/2^{i+1}$ is of bad type, and approximable by a rational with a small, *odd* denominator. It is in this case (and when the oracle does not distinguish between any $S$ and $S+\alpha_N^{\tau(n)}(u,v)$) that we will be able to use the gcd method described previously. This last case is treated in the next section.

## 5.3. RSA inversion, Combining Method 1 and 2

The main result of this section is the following Proposition.

**Proposition 5.13.** If there are integers $u,v,r,s$, $0 \le u,v \le 2^{\tau(n)}-1$, $v$ odd, $0 < s \le Q(n)$, $(r,s) = 1$ and $s$ odd, such that

$$\left|\frac{\tilde{\alpha}_N^{\tau(n)}(u,v)}{2^{i+1}} - \frac{r}{s}\right| \le \frac{1}{s^2\psi(n)},$$

and for all boxes $S$, $\Delta^{\mathfrak{O}}(S, S+\alpha_N^{\tau(n)}(u,v))$ is negligible, then using $\mathfrak{O}$, we can in random polynomial time construct an oracle $\mathfrak{O}'$ and find an interval $J$ for which we have $\Delta^{\mathfrak{O}'}(J, J+(N+1)/2) \ge \varepsilon'(n)$ where $\lambda(J), \varepsilon'(n)$ are non-negligible.

In other words, the conditions of Lemma 5.1 is fulfilled, and the gcd method can be used.

We sketch the idea behind the proof. Recall that we write $N = N_1 2^{i+1} + N_0$. In [3] it was shown that if $\mathfrak{O}$ is an $\varepsilon(n)$-oracle for the $i$th bit in $E_N^{-1}(x)$, and we (utilizing the multiplicative properties of RSA) define a new oracle, $\mathfrak{O}_2$, by

$$\mathfrak{O}_2(E_N(x)) = \mathfrak{O}(E_N([N_1^{-1}x]_N)), \tag{5.3}$$

then $\mathfrak{O}_2(E_N(x))$ (using the improved sampling techniques in [1]) distinguishes between some sets $J, J+(N+1)/2$ with some advantage. This can be used to give a $\frac{1}{2}$-security result for all RSA bits as follows: The mapping $z \mapsto [N_1 z]_N$ maps intervals at distance $2^i$ to intervals "almost" at distance $(N+1)/2$. This "almost" depends on $[N]_{2^{i+1}}$ and gives rise to the additional error term which is $\frac{1}{4}$ in the worst case.

We have in this paper so far encountered some “bad” $\tilde{\alpha}_N^{\tau(n)}(u,v)$’s (expressing more general relations between $N$ and $i$), but we have gathered a lot of information on these and on the behavior of the oracle. We can use this information to find another transformation (similar to (5.3)) of the original oracle that maps certain sets at distance $2^i$ to sets also almost at distance $(N+1)/2$ and where the oracle’s advantage is concentrated.

**Lemma 5.14.** *If there are integers $u,v,r,s$, $0 \leq u,v \leq 2^{\tau(n)} - 1$, $v$ odd, $0 < s \leq Q(n)$, $(r,s) = 1$ and $s$ odd, such that $\left|\tilde{\alpha}_N^{\tau(n)}(u,v)/2^{i+1} - r/s\right| \leq \frac{1}{s^2\psi(n)}$, then with $u' = [-uv^{-1}N]_{2^{\tau(n)}}$ there is $r' \in \mathbb{Z}$, $r' \leq 2Q(n)$ so that*

$$\left|s(u'+N_2) - r'2^{\tau(n)}\right| \leq ns\varepsilon(n)^{-1}.$$

Notice that $s(u'+N_2) - r'2^{\tau(n)} \in \mathbb{Z}$ so we can make the following definition.

**Definition 5.15.** For $s, u', r'$ as above, define the integer

$$\kappa \triangleq s(u'+N_2) - r'2^{\tau(n)}.$$

We can now write down the oracle that we claim distinguishes between some $J$ and $J + (N+1)/2$. Let $\mathfrak{O}$ be the original oracle for the $i$th bit and $s, \kappa$ as above.

**Definition 5.16.** Define $\varphi : \mathbb{Z}_N \to \mathbb{Z}_N$ by

$$\varphi(z) \triangleq [(sN_1 - \kappa)z]_N.$$

For $S \subset \mathbb{Z}_N$, $\varphi(S)$ is defined in the natural way; $\{\varphi(z) \mid z \in S\}$. We now define the oracle

$$\mathfrak{O}'(E_N(x)) \triangleq \mathfrak{O}(E_N(\varphi^{-1}(x))).$$

It may be the case that $\varphi^{-1}$ does not exist, i.e. that $sN_1 - \kappa$ does not have a multiplicative inverse, but if so, we have factored $N$ and we are done. We see that when $s = 1, \kappa = 0$, we get precisely the same oracle construction as in [3].

To prove that $\mathfrak{O}'$ is good, the plan is the following. Consider a set in the $\Pi(N,i)$-plane of the following form:

$$o_1 = \{S + ks\tilde{\alpha}_N^{\tau(n)}(u,v) \mid 0 \leq k \leq \lfloor (2^{\tau(n)} - 1)/s \rfloor\}$$

where $S$ is a box of height $\pi_Y(s\tilde{\alpha}_N^{\tau(n)}(u,v)) \approx sN_3$ and width $\Theta(\varepsilon(n)2^{i+1})$. We call such a set an *orbit*. The reason for considering translations by $s\tilde{\alpha}_N^{\tau(n)}(u,v)$ rather than by $\tilde{\alpha}_N^{\tau(n)}(u,v)$ is that we know by assumption that $s\tilde{\alpha}_N^{\tau(n)}(u,v)$ is close to a multiple of $2^{i+1}$, and the sequence $ks\tilde{\alpha}_N^{\tau(n)}(u,v)$, $k = 0, 1, \ldots$, has a relatively steep “slope” in the $Y$-direction of the plane $\Pi(N,i)$. We can also assume that the oracle $\mathfrak{O}$ behaves almost the same on all boxes in this orbit. Since $s$ is small, we would otherwise have boxes $S', S' + \tilde{\alpha}_N^{\tau(n)}(u,v)$ where the oracle behaves differently, contradictory to assumption. (This would imply good boxes at distance $\alpha_N^{\tau(n)}(u,v)$.) Consider now also

$$o_1' = \{S + ks\tilde{\alpha}_N^{\tau(n)}(u,v) + 2^i \mid 0 \leq k \leq \lfloor (2^{\tau(n)} - 1)/s \rfloor\}.$$

The same holds here; the oracle behaves the same on all of these or else we are done. Having chosen the size of $S$ appropriately (as above), we can by translations (in the $I$-direction of $\Pi(N,i)$) of these orbits cover almost the entire plane with pairs of orbits, where the oracle’s behavior is “flat” within each orbit. Notice that for each point $z \in o_1$, there is a corresponding point $z' \in o_1'$ so that $\mathrm{bit}_i(z) \neq \mathrm{bit}_i(z')$. It is possible to show that under the mapping $\varphi(\cdot)$, $o_1$ gets mapped into what is (almost) an interval $J_1$, and that $o_1'$ maps to a similar “interval” $J_1'$ such that $J_1' \approx J_1 + (N+1)/2$. If the new oracle, $\mathfrak{O}'(E_N(x)) = \mathfrak{O}(E_N(\varphi^{-1}(x)))$, would behave the same on $J_1$ and $J_1'$, then the original oracle, $\mathfrak{O}$, can have no advantage in deciding the $i$th bit on $o_1 \cup o_1'$. If, on the other hand, $\mathfrak{O}'$ behaves differently we have the desired oracle that distinguishes between intervals at distance $(N+1)/2$. Repeating the argument for all translations of orbits $\{(o_j, o_j')\}$, always finding that $\mathfrak{O}'$ behaves the same on the corresponding $J_j, J_j'$ ($J_j = \varphi(o_j)$, $J_j' = \varphi(o_j')$) would thus give a contradiction to the assumptions on $\mathfrak{O}$, since it would then nowhere have an advantage in deciding the $i$th bit.

In summary, the work that needs to be done in order to show that our orbits gets mapped into intervals as above, is to prove that $\varphi(s\tilde{\alpha}_N^{\tau(n)}(u,v))$ is very small and that $\varphi(2^i) \approx (N+1)/2$. This is not hard, but somewhat tedious, and we omit the details.

**Theorem 5.17.** *For any $i$, the $i$th bit in an RSA encrypted message is secure, unless RSA can be broken in random polynomial time.*

*Proof.* We may assume that $i \geq \tau(n) + \log \varepsilon(n)^{-4} + \log n + 33$, since the other case is covered by [1]. For $i > n - 3\tau(n) - \log \varepsilon(n)^{-1} - 7$ the results can be proved by a reduction from the simultaneous bit security of the least significant bits established in the same paper. This follows since for $t \in O(\log n)$, the most significant bits of $[2^{-t}x]_N$ are closely related to the least significant bits of $x$. However, we also need to take into account that the most significant bits may be non-negligibly biased towards 0. To handle this, we use *weighted success ratio* as a definition of security. This is the correct security measure for biased bits, see [18]. We omit the details.

It remains to treat the bits that are far away from either end. If for all $u,v$ we have good boxes at distance $\alpha_N^{\tau(n)}(u,v)$, we are done by Lemmas 5.6, 5.7. If not, the theorem follows from Propositions 5.9, 5.11, 5.12, 5.13, and Lemma 5.1. □

## 6. Security of Discrete Log Bits

Let $f_{p,g}(x) = [g^x]_p$, $p$ an $n$-bit prime and $g$ a generator for $\mathbb{Z}_p^*$. Suppose that $p-1 = p'2^k$, where $p'$ is odd. Then given $f_{p,g}(x)$, the $k$ least significant bits of $x$ are "easy" and the $O(\log n)$ following bits are secure, see [14]. Also, the $O(\log n)$ most significant bits are secure, see [12]. Can our methods developed here be used to prove security for *all* bits of $x$? When trying to extend the current methods, two problems are encountered.

The first problem is that we cannot query the oracle on $f_{p,g}(2^{-\tau}x)$ when the group order, $p-1$, is even. By recent work of Schnorr, [16], we can however reduce the problem to a subgroup of odd order, $p'$.

Writing $p' = P_1 2^{i+1-k} + P_0$, the second problem is that $\varphi^{-1}$ (recall Definition 5.16) may not exist, i.e. that $\gcd(sP_1 - \kappa, p') > 1$. As long as this gcd is not too large (which is likely for a random $p$), we can however take care of this problem also.

**Theorem 6.1.** Unless the discrete log problem can be solved in random polynomial time, with probability $1 - o(1)$ over random choices of $p = p'2^k + 1$, bits $k+1, \ldots, n-1$ of $x$ are individually secure for $f_{p,g}(x)$.

It remains to extend Theorem 6.1 to cover all values of $p$ and in particular to treat the case when the above gcd is large. Although this might sound like a technicality, it seems that such an extension would require new techniques. To see this, consider the following example.

Assume that $p = q(2^{i+1}+2)+1$ where $q$ is a prime of size around $2^{i/2}$. Our bit security proofs compute the discrete logarithm of a number $y$ by querying the $i$th bit of the discrete logarithm of numbers of the form $y^a g^b$. This is equivalent to reconstructing $x$ from information on the $i$th bit of $ax+b$. Now we claim that using this approach, for the above $p$, it is hard to distinguish $x$ and $x' = x + t(2^{i+1}+2)$ for any $t > 0$. The reason is simply that $ax+b$ and $ax'+b$ (mod $p-1$) differ by $at(2^{i+1}+2)$ and since $at$ is only considered mod $q$, except with exponentially small probability, the two numbers have the same value for their $i$th bit.

We will elaborate on the results for the discrete logarithm problem in the journal version of this paper.

## 7. Discussion and Open Problems

Although the reduction from RSA inversion to predicting the individual bits is polynomial time, we ask if there is a simpler proof, strengthening the practical implications of the result. To hide partial information on $x$ in a practical application involving RSA, it is of course still wise to use RSA in a more sophisticated way such as in [2].

## References

[1] W. Alexi, B. Chor, O. Goldreich, and C. Schnorr. RSA and Rabin functions: Certain parts are as hard as the whole. *SIAM Journal on Computing*, 17(2):194–209, 1988.

[2] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *Proceedings of Eurocrypt '94*, pages 92–111. LNCS 950, Springer-Verlag, 1995.

[3] M. Ben-Or, B. Chor, and A. Shamir. On the cryptographic security of single RSA bits. In *Proceedings of the 15th ACM STOC*, pages 421–430, 1983.

[4] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, 1986.

[5] D. Boneh and R. Venkatesan. Breaking RSA may not be equivalent to factoring. In *Proceedings of Eurocrypt '98*, pages 59–71. LNCS 1403, Springer-Verlag, 1998.

[6] B. Chor and O. Goldreich. RSA/Rabin least significant bits are $\frac{1}{2} + \frac{1}{\mathrm{poly}(\log n)}$ secure. In *Proceedings of CRYPTO '84*, pages 303–313. LNCS 196, Springer-Verlag, 1985.

[7] R. Fischlin and C. P. Schnorr. Stronger security proofs for RSA and Rabin bits. In *Proceedings of Eurocrypt '97*, pages 267–279. LNCS 1233, Springer-Verlag, 1997.

[8] O. Goldreich. On the number of close-and-equal pairs of bits in a string (with applications on the security of RSA's L.S.B.). In *Proceedings of Eurocrypt '84*, pages 127–141. LNCS 209, Springer-Verlag, 1985.

[9] S. Goldwasser, S. Micali, and P. Tong. Why and how to establish a private code on a public network (Extended abstract). In *Proceedings of the 23rd IEEE FOCS*, pages 134–144, 1982.

[10] J. Håstad, A. W. Schrift, and A. Shamir. The discrete logarithm modulo a composite hides $O(n)$ bits. *Journal of Computer and System Sciences*, 47:850–864, 1993.

[11] L. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences*. John Wiley & Sons, 1 edition, 1974.

[12] D. L. Long and A. Wigderson. The discrete log hides $O(\log n)$ bits. *SIAM Journal on Computing*, 17(2):413–420, 1988.

[13] M. Näslund. All bits in $ax+b$ mod $p$ are hard. In *Proceedings of CRYPTO '96*, pages 114–128. LNCS 1109, Springer-Verlag, 1996.

[14] R. Peralta. Simultaneous security of bits in the discrete log. In *Proceedings of Eurocrypt '85*, pages 62–72. LNCS 219, Springer-Verlag, 1986.

[15] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[16] C. P. Schnorr. Security of almost all discrete log bits. ECCC report TR98-033, 1998.

[17] C. P. Schnorr and W. Alexi. RSA-bits are $0.5+\varepsilon$ secure. In *Proceedings of Eurocrypt '84*, pages 114–128. LNCS 209, Springer-Verlag, 1985.

[18] A. W. Schrift and A. Shamir. On the universality of the next bit test. In *Proceedings of CRYPTO '90*, pages 394–408. LNCS 537, Springer-Verlag, 1991.

[19] U. V. Vazirani and V. V. Vazirani. RSA bits are $.732+\varepsilon$ secure. In *Proceedings of CRYPTO '83*, pages 369–375. Plenum Press, 1984.