The Security of all RSA and Discrete Log Bits

JOHAN HÅSTAD
Royal Institute of Technology
and
MATS NÄSLUND
Ericsson Research

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. Moreover, we prove that blocks of $O(\log \log N)$ bits of x are computationally indistinguishable from random bits. The results carry over to the Rabin encryption scheme.

Considering the discrete exponentiation function g^x modulo p, with probability $1 - o(1)$ over random choices of the prime p, the analog results are demonstrated. The results do not rely on group representation, and therefore applies to general cyclic groups as well. Finally, we prove that the bits of $ax + b$ modulo p give hard core predicates for any one-way function f.

All our results follow from a general result on the *chosen multiplier hidden number problem:* given an integer N, and access to an algorithm $\mathcal{P}_x$ that on input a random $a \in \mathbb{Z}_N$, returns a guess of the ith bit of $ax \bmod N$, recover x. We show that for any i, if $\mathcal{P}_x$ has at least a non-negligible advantage in predicting the ith bit, we either recover x, or, obtain a non-trivial factor of N in polynomial time. The result also extends to prove the results about simultaneous security of blocks of $O(\log \log N)$ bits.

Categories and Subject Descriptors: E.3 [**Data Encryption**]: Public Key Cryptosystems; F.2.1 [**Numerical Algorithms and Problems**]: Number-theoretic computations

General Terms: Security, Theory

Additional Key Words and Phrases: Cryptography, complexity, RSA-encryption, bit-security, discrete logarithms

1. INTRODUCTION

What is to be meant by a secure cryptosystem? There are rigorously defined notions, such as semantic security defined by Goldwasser and Micali [Goldwasser and Micali 1984], which informally says that "whatever can be computed efficiently from the crypto-text 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. In fact, any deterministic, public-key crypto system *must* leak some information. 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 pre-

Author addresses. Johan Håstad, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, SE-100 44 Stockholm, Sweden. Mats Näslund. Communicatons Security Lab, Ericsson Research, SE-164 80 Stockholm, Sweden.

dict 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 [Rivest et al. 1978]. 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 modest security result, in a sequence of papers, [Goldwasser et al. 1982; Ben-Or et al. 1983; Vazirani and Vazirani 1984b; Goldreich 1985; Schnorr and Alexi 1985; Chor and Goldreich 1985], the bit-security was strengthened, ending with the final proof of "complete" security by Alexi, Chor, Goldreich, and Schnorr in [Alexi et al. 1988]. There are also other known security results for certain predicates that are related to the individual bits of x, e.g. $\text{half}_N(x) \triangleq 1$ if $x \geq (N+1)/2$, 0 otherwise, see [Goldwasser et al. 1982] 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 $\text{half}_N(x)$ and the individual bits of x, Ben-Or, Chor, and Shamir proved in [Ben-Or et al. 1983], that the internal bits cannot be computed with probability of success exceeding 15/16. By a reduction to this proof, the result in [Alexi et al. 1988] for the least significant bit, then improved the result to 3/4, still leaving a large gap to the desired 1/2-result.

Stated slightly informally we prove the following main theorem.

THEOREM. *For all sufficiently large n, unless RSA can be inverted with non-negligible probability in random polynomial time, no single bit of $E_N^{-1}(x)$ (where $\lceil \log N \rceil = n$) can be predicted in polynomial time with non-negligible advantage.*

Moreover, distinguishing blocks of $O(\log n)$ bits of x from random bits is polynomial-time related to inverting RSA.

The proof uses very little about the structure of RSA and the essential property is that given $E_N(x)$ we can construct $E_N(ax)$ for any known integer a. Using this we can from a presumed predictor get predictions for the ith bit of ax, and we use this for carefully chosen values of a. A more curious property of RSA that we may need to make use of is the fact that N is a product of two primes, see the discussion below.

We phrase this as an abstract problem called the *chosen multiplier hidden number problem*. It is simply the problem of given a black box that on input a random a predicts the ith bit of $ax \bmod N$, extract x. This is exactly the problem we solve and by using the abstract formulation we are able to apply our method to other situations. It is curious to note that the method is not universal and the extractor can fail but in doing so it discovers a factor in the modulus N. We describe a counterexample that shows that the hidden multiplier problem can not be solved for general moduli. For numbers N of a special form it is possible to construct

a predictor that is correct with probability $1 - o(1)$ and such that the predictor behaves the same on exponentially many numbers x and hence it cannot be used to extract the correct x.

One implication of these problems is that for RSA variations where the modulus is allowed to have more than 2 factors we cannot obtain the same result. The predictor might give us a non-trivial factor of the modulus but not a complete factorization. It is unknown how to invert RSA without the complete factorization, and the predictor does not help us to extract all of x either. In other words, the (seemingly unlikely) existence of a bit-predictor for multi-prime RSA would not immediately contradict the one-wayness of this RSA variant.

We do get a number of corollaries of our main result and let us describe them briefly.

Näslund claimed in [Näslund 1996] that all bits in affine functions modulo a (not too small) prime, $x \mapsto ax + b$ modulo p, are secure given the information a, b, p, and $f(x)$ for *any* one-way function f. His proof has been found to contain a gap but we can here prove his result fully by applying our general techniques. Using the results on efficient noisy Chinese remaindering first established by Goldreich, Ron and Sudan [Goldreich et al. 1999] we also extend this results to the case when p is quite small,

We also study the Rabin encryption function, $x \mapsto x^2$ modulo N. This function is not one-to-one and this makes it difficult even to define the notion of a predictor. The function can be made one-to-one in several ways and we choose to study the case when the modulus is the product of two primes congruent to 3 mod 4. We output, on top of the standard output, the Jacobi symbol of the input as well as the half_N predicate defined above. This function is one-to-one, it is closely related to the Rabin function, and it maintains the property that inversion is polynomially related to the factorization problem.

Finally, we also study the discrete logarithm problem and for a randomly chosen prime p, with high probability, the results also hold with respect to the discrete exponentiation function $x \mapsto g^x$ modulo p. That is, for almost all p, predicting a single bit (or distinguishing blocks of $O(\log \log p)$ bits from random bits) is as hard as computing discrete logarithms. In fact, we do not use specific details about the group representation, and the results therefore apply to general cyclic groups.

The paper is organized as follows. In Section 2 we start with some preliminaries giving basic definitions used in the paper. In Section 3 we describe previous work on the security of the RSA-function. In Section 4 we define our main abstract problem and state our main theorems. We then turn to the proofs and warm up by introducing some basic techniques in Section 5. In Section 6 we prove the main theorem in the basic case when the predictor gives the value of one bit that is not biased. We extend this result to obtain simultaneous security of any window of $O(\log n)$ bits in Section 7. We then give our applications to RSA bits (Section 8), Rabin bits (Section 9), discrete logarithms bits (Section 10) and bit security of the mod p hash functions (Section 11). We end by briefly discussing some open problems in Section 12.

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. Slightly abusing notation we do not have different symbols for an integer and the binary string representing it. If S is a set, $\#S$ is the cardinality of S and by $x \in_{\mathcal{D}} S$ we mean an x chosen at random according to the distribution $\mathcal{D}$ on S, $\mathcal{U}$ denoting the uniform distribution. If a random x is chosen with the uniform distribution we sometimes, for readability reasons, write $x \in S$ instead of the more cumbersome but accurate $x \in_{\mathcal{U}} S$.

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 $\lambda(T)$.) For two sets S, T, $S \nabla T$ is the *symmetric difference*: $(S \setminus T) \cup (T \setminus S)$.

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 a random $x \in_{\mathcal{U}} \{0,1\}^n$ and the random coin flips of M.

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

For $m, z \in \mathbb{Z}$, $m > 0$, we write $[z]_m \triangleq z$ modulo m where we use $\{0, 1 \ldots, m-1\}$ as representatives and this is the ring $\mathbb{Z}_m$. We put $\mathrm{abs}_m(z) \triangleq \min\{[z]_m, m - [z]_m\}$ and if for some $\delta \in [0,1]$, $\mathrm{abs}_m(z) \leq \delta m$, z is said to be δ-*small* (modulo m). A number x is δ-*determined* modulo m if it can be written on the form $y + z$ where y is known and z is δ-small. As in the introduction, we define the predicate $\mathrm{half}_N(x)$ to be true iff $N/2 < x < N$. We use (a, b) to denote the greatest common divisor of $a, b \in \mathbb{Z}$.

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 $z \in \mathbb{Z}$, $0 \leq i < \|z\|$, $\mathrm{bit}_i(z)$ denotes the ith bit in the binary representation of z, $\mathrm{bit}_i(z) \triangleq \lfloor z/2^i \rfloor$ modulo 2. This means that the bits are numbered $0, 1, \ldots, \|z\| - 1$, "right-to-left". In particular $\mathrm{lsb}(z) \triangleq \mathrm{bit}_0(z)$. 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 ith bit we mean the value $\beta_i(N)$ such that $\Pr_{z \in_{\mathcal{U}} \mathbb{Z}_N}[\mathrm{bit}_i(z) = 0] = \frac{1+\beta_i(N)}{2}$. 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. A notion of $\epsilon(n)$-security of biased bits is given in Section 4.

Finally, let $\mathcal{D}, \mathcal{D}'$ be distributions on the same space S. We call $\mathcal{D}, \mathcal{D}'$ (polyno-

mially) *distinguishable* if there is a pptm D such that

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

is non-negligible.

A warning about convention. In many places we define integers by an expression that gives a real number. If the number is not integral we round it to one of the two closest integers. Sometimes we round explicitly i.e. by writing $\lfloor x \rfloor$ but at other times, for readability reasons, we do not.

3. PREVIOUS WORK ON RSA BIT-SECURITY

The security of the least significant bit in an RSA encrypted message has gained a lot of attention and let us first describe the results that apply to this bit before we continue the discussion for general bit-positions. The first result by Goldwasser, Micali, and Tong, [Goldwasser et al. 1982], proved that $\mathrm{lsb}(x)$ is $1 - o(1)$ secure for RSA. They used the relation $\mathrm{half}_N(x) = \mathrm{lsb}([2x]_N)$ (half_N as in the introduction), enabling a binary search to find x. By introducing a gcd computation technique a $\frac{1}{2} + o(1)$ result was given for $\mathrm{lsb}(x)$ in [Ben-Or et al. 1983] by Ben-Or, Chor, and Shamir. Further progress (still using the gcd technique) was accomplished by a more intricate sampling technique, and then by an improved combinatorial analysis of this technique. More precisely, Vazirani and Vazirani, [Vazirani and Vazirani 1984b], and then Goldreich, [Goldreich 1985], respectively, showed 0.464- and 0.45-security. The main drawback of the method in [Ben-Or et al. 1983] is that queries to the predictor are made in pairs, causing so called error-doubling.

By improving the sampling techniques once again, Schnorr and Alexi, [Schnorr and Alexi 1985], proved ϵ-security for any constant ϵ. They removed the error-doubling phenomenon by using "preprocessing". The cost of this preprocessing was, however, exponential in ϵ^{-1}.

To show $\epsilon(n)$-security for any non-negligible $\epsilon(\cdot)$, Chor and Goldreich managed in [Chor and Goldreich 1985] (see also [Alexi et al. 1988]) to reduce the cost of preprocessing to $\mathrm{poly}(\epsilon^{-1})$ by introducing the so called two-point based sampling. Recently, a simpler proof of $\epsilon(n)$-security was given in [Fischlin and Schnorr 1997] by Fischlin and Schnorr. This last method does not use a gcd computation. Instead, the main idea is to use lsb-information to iteratively improve an approximation for the rational number $\frac{x}{N}$.

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 result appeared in the paper [Goldwasser et al. 1982], where it was shown that for each i, there are N of very special form, for which the ith bit of x cannot be computed without errors. In [Ben-Or et al. 1983], it was proved that a predictor for the ith bit of RSA can be converted into an lsb-predictor, increasing the error probability by $\frac{1}{4}$ in the worst case. However, they could also prove that for every second bit-position i, the error introduced could 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. The best result so far is the $\frac{1}{2} + o(1)$-security that follows from

the work in [Alexi et al. 1988], still leaving a large gap to the desired $o(1)$ result. The provable security obtainable by these reductions depends on N and i (the bit-position considered), but for worst case N and i, results better than $\frac{1}{2} + o(1)$ are impossible by this "standard" reduction. If the predictor for the ith bit we start with is correct with probability $\frac{1+\epsilon'}{2}$, then after the conversion to an lsb-predictor, a success probability non-negligibly greater than $\frac{1}{2}$ must remain. The extra $\frac{1}{4}$ error that the reduction may add to the error probability is a tight bound, so we certainly need $\frac{1+\epsilon'}{2} - \frac{1}{4} > \frac{1}{2}$, i.e. $\epsilon' > \frac{1}{2}$.

4. THE CHOSEN MULTIPLIER HIDDEN NUMBER PROBLEM

In 1996, Boneh and Venkatesan, [Boneh and Venkatesan 1996], introduced the hidden number problem (HNP): given an algorithm that when queried selects random $a \in \mathbb{Z}_N$ and returns a and some partial information (e.g. the most significant bits) of $[ax]_N$, retrieve x. Following the initial paper, this problem has seen many applications in studying the security of various schemes, e.g. [Li et al. 2002] to mention one. We here study a variant of HNP where we allow ourselves to choose the multiplier, but on the other hand, we only demand a predictor that is correct with non-negligible advantage. In this section we formally describe the type of predictor that we work with in this paper and state our main results.

Definition 4.1. An $(N, i, \epsilon(n))$ *chosen multiplier hidden number predictor*, $||N|| = n$, for a number x is a pptm $\mathcal{P}_x$ such that

$$\Pr[\mathcal{P}_x(a) = \mathrm{bit}_i([ax]_N)] \geq \frac{1+\epsilon(n)}{2},$$

probability taken over the choice of $a \in_{\mathcal{U}} \mathbb{Z}_N$ and the internal coinflips of the predictor.

From now on we reserve i, N and x to be used only as the index of the predicted bit, the modulus and the unknown number respectively. We usually do not explicitly specify them and in particular we call the above specified predictor an $\epsilon(n)$-*CMHNP*. We sometimes even suppress, for the sake of readability the parameter ϵ, which in fact is only used as the advantage of the predictor. Similarly, as x is a fixed element, we usually write $\mathcal{P}$ instead of $\mathcal{P}_x$.

Note that for RSA, a predictor for the ith bit of x given $E_N(x)$ can be used to get a CMHNP. To get a prediction of the ith bit of ax we supply the predictor with

$$E_N(ax) \equiv [a^e E_N(x)]_N.$$

Our first version of the main result can now be stated as follows.

THEOREM 4.2. *There are constants c_1 and c_2 such that given an $(N, i, \epsilon(n))$-CMHNP where $||N|| = n$, $\epsilon(n)$ is non-negligible and $0 \leq i \leq n - c_1 \log \epsilon(n)^{-1} - c_2 \log n$, we can in polynomial time, with probability at least $1/2$, construct a list of elements in $\mathbb{Z}_N$ containing x, or, find a nontrivial factor of N.*

The reason for the need to produce a set of candidates for x, rather than a single value, will soon be obvious, we just note that for applications such as the bit-security of RSA, we are also given the value $E_N(x)$, and can easily find the right x in the list, applying $E_N(\cdot)$ to each candidate.

The proof of this theorem is the main technical contribution of this paper and is given in Section 6. In the next section we show that the possibility of not being able to extract x is real in that if N is of special form we can construct a $(1-o(1))$-CMHNP for which it is still impossible to extract all of x.

The extension of the result to the most significant bits is at the same time straightforward and problematic. It is straightforward in the sense that no new ideas are needed. It is problematic in the sense that even new definitions are needed. The problem being that we run into technicalities which are due to the fact that the most significant bits can be biased and thus even a trivial predictor can predict the bit with probability significantly greater than $1/2$ of being correct. We have the below definition due to Schrift and Shamir [Schrift and Shamir 1991]. There are equivalent definitions and for those we refer to [Schrift and Shamir 1991].

Definition 4.3. Let p be a non-constant predicate. An $(N, \epsilon(n))$-*CMHNP for* p is a pptm $\mathcal{P}_x$ such that

$$|\Pr[\mathcal{P}_x(a) = 1 \mid p([ax]_N) = 1] - \Pr[\mathcal{P}_x(a) = 1]| \geq \epsilon(n). \tag{4.1}$$

A predicate p is $\epsilon(n)$-*secure* if no pptm predictor exists with advantage $\epsilon(n)$ and it is *secure* if it is $\epsilon(n)$-secure for all non-negligible $\epsilon(n)$.

Using this definition we get a notion of an $\epsilon(n)$-CMHNP also for biased bits and we prove the following theorem.

THEOREM 4.4. *Given an* $(N, i, \epsilon(n))$-*CMHNP where* $||N|| = n$ *and* $\epsilon(n)$ *is non-negligible, we can in polynomial time, with probability at least* $1/2$*, output a list of elements containing* x*, or find a nontrivial factor of* N*.*

Apart from single bits we are also interested in predictors that can predict collections of bits. A group of bits is, informally speaking, simultaneously secure if they cannot be distinguished from random bits. Another way to say this is that given a guess for the group of bits we cannot tell whether these are the correct bits or independent random bits. We have the following definition.

Definition 4.5. An $(N, i, \epsilon(n))$-*simultaneous chosen multiplier hidden number predictor of width* $d(n)$ is a pptm $\mathcal{P}_x$ such that

$$\left|\Pr[\mathcal{P}_x(a, \mathrm{B}_i^{i+d(n)-1}([ax]_N)) = 1] - \Pr[\mathcal{P}_x(a, R) = 1]\right| \geq \epsilon(n),$$

where the probabilities are taken over random $a \in_{\mathcal{U}} \mathbb{Z}_N$, $R \in_{\mathcal{U}} \mathbb{Z}_{d(n)}$, and the internal coinflips of the algorithm. We assume here that i and $d(n)$ are such that $\mathrm{B}_i^{i+d(n)-1}([ax]_N)$ for a random a is at most $\epsilon(n)/2$ away from the uniform distribution in L_1-norm.

This definition only works for the non-biased bits, and for general positions we have the following definition.

Definition 4.6. Let p be a non-constant function defined on $\mathbb{Z}_N$. A pptm $\mathcal{P}_x$ is an $(N, \epsilon(n))$-CMHNP for p if there is a value b in the range of p that is output with non-negligible probability such that

$$|\Pr[\mathcal{P}_x(a) = b \mid p([ax]_N) = b] - \Pr[\mathcal{P}_x(a) = b]| \geq \epsilon(n). \tag{4.2}$$

A function p is $\epsilon(n)$-*secure* if no polynomial time predictor exists with advantage $\epsilon(n)$ and it is *secure* if it is $\epsilon(n)$-secure for all non-negligible $\epsilon(n)$.

We summarize all our theorems into the following.

THEOREM 4.7. *For any $d(n) \in O(\log n)$ and any i such that $\mathrm{B}_i^{i+d(n)-1}([x]_N)$ is non-constant the following is true. Given an $(N, i, \epsilon(n))$-simultaneous CMHNP of width $d(n)$ for $\mathrm{B}_i^{i+d(n)-1}([ax]_N)$ we can in polynomial time, with probability at least $1/2$, produce a list containing x, or, find a nontrivial factor of N.*

It turns out that after proving the basic result of Theorem 4.2 it is convenient to first prove the simultaneous security of Theorem 4.7 for the unbiased positions and then by a reduction we can establish both Theorem 4.4 and Theorem 4.7 for the biased positions.

4.1 A Predictor that Does Not Allow Extraction

Let $N = q_1(2^{i+1} + q_2)$ where $(q_1, 2^{i+1} + q_2) = 1$. The reader should think of q_1 as "large but much smaller than 2^i" and q_2 as "much smaller than $2^i/q_1$". Now consider all x of the form $x_0 + x_1(2^{i+1} + q_2)$. We want to construct an $(1 - o(1))$-CMHNP whose output values are independent of x_1 which implies that extraction is impossible as we have the same predictor for q_1 different values of the x and if q_1 is large, as it might be, we cannot output them all in polynomial time. We have the following lemma

LEMMA 4.8. *For at least a fraction $1 - (q_1 + 1)q_2 2^{-i}$ of all x_0 the ith bit of $[x_0 + j(2^{i+1} + q_2)]_N$ is independent of j.*

PROOF. Since the property is invariant under adding multiples of $2^{i+1} + q_2$ to x_0 we can assume that $0 \leq x_0 < 2^{i+1} + q_2$. Now if $x_0 < 2^i - q_1 q_2$ or $2^i \leq x_0 < 2^{i+1} - q_1 q_2$ the ith bit of all the numbers are the same. Hence the probability of having a constant ith bit is at least

$$\frac{2^{i+1} - 2q_1 q_2}{2^{i+1} + q_2} \geq \frac{2^{i+1} - (2q_1 + 1)q_2}{2^{i+1}} \geq 1 - (q_1 + 1)q_2 2^{-i}.$$

□

We can now define the predictor $\mathcal{P}_x(a)$, for any $x = x_0 + x_1(2^{i+1} + q_2)$, to output the correct value of the ith bit of ax for each number a such that $ax_0 \bmod (2^{i+1} + q_2)$ is such that ith bit is independent of x_1. For other values of a the predictor returns a random coinflip. By the lemma this predictor is correct for a suitable choice of q_1 and q_2 with probability $1 - o(1)$ and that it is, by construction, independent of x_1. From this example we conclude that a CMHNP is not sufficient to extract x for all values of N and i.

4.2 On the Abstract Framework

In our original presentation [Håstad and Näslund 1998] of our work we did not use the formalism of the chosen multiplier hidden number predictor. It is, however, implicit in that we used the same techniques in many situations. The current formalism was suggested by an anonymous referee of this submission. We agree

that the current formalism is good and for this reason we have chosen the current form of the presentation.

Meanwhile, also Kiltz [Kiltz 2001] have studied the hidden number problem and investigated the potential for further application. For this reason we here only state the applications mentioned in the original conference presentation [Håstad and Näslund 1998], RSA and discrete logarithms, and the improvement of the result from [Näslund 1996].

5. THE BASIC TECHNIQUES

In this section we remind the reader of some previous, useful techniques and introduce some tools that we use for our main proof.

5.1 The Method of Fischlin and Schnorr

To recover x using an lsb-predictor [Fischlin and Schnorr 1997] proceeds as follows. We wish to find a number $x \in \mathbb{Z}_N$ given access to a (very accurate) lsb-predictor. Suppose we are given an initial guess y with $|y - x| < N/n^k$ for some k. Then by using the predictor to calculate $\mathrm{lsb}(x)$ we get a guess, $z \triangleq (y - \mathrm{lsb}(x))/2 + \mathrm{lsb}(x)(N+1)/2$, of $[x/2]_N$ with half the uncertainty, namely $|z - [x/2]_N| \leq N/(2n^k)$. Repeating this about n times gives an exact value for a number of the form $[x2^{-l}]_N$ and from this we can retrieve x. Finally note that we can in advance specify a polynomial number of initial values of y, one of which will be accurate enough.

It turns out that it is not necessary to have a very accurate lsb-predictor to start with to make this procedure work. Let an interval $J \subset [0, \ldots, N-1]$ denote a set of consecutive integers in $\mathbb{Z}_N$ and for $z \in \mathbb{Z}$, $J + z$ is the interval J translated by z, allowing reductions modulo N. Suppose that for some not too short interval J, we have a predictor that, when given a, is somewhat more likely to answer "1" for $ax \in J$ than for $ax \in J + (N+1)/2$. Now ask this predictor about $a = [r_j + \frac{1}{2}]_N$ for some cleverly chosen number r_j. We have that

$$[2^{-1}x]_N = \frac{x - \mathrm{lsb}(x)}{2} + \mathrm{lsb}(x)[2^{-1}]_N = \frac{x - \mathrm{lsb}(x)}{2} + \mathrm{lsb}(x)\frac{N+1}{2}, \qquad (5.1)$$

see also Figure 1. Hence, if $r_j x + \frac{x - \mathrm{lsb}(x)}{2} \in J$, then $[(r_j + 2^{-1})x]_N \in J + \mathrm{lsb}(x)(N+1)/2$. Since the predictor behaves differently on J, $J + (N+1)/2$, we get information about $\mathrm{lsb}(x)$. Going over many r_j this information can be refined to a guess with high confidence.

There are technical details to be addressed. For instance, how to generate the numbers r_j in such a way that we can determine when $r_j x + \frac{x - \mathrm{lsb}(x)}{2}$ lie in J. This is achieved by generating pairwise independent numbers r_j where we have some partial information of $r_j x$. This is done formally in Lemma 5.2. A more serious concern is the existence of an interval J with the given properties.

5.2 The Method of Näslund

This method attacks the case when the predictor predicts an arbitrary bit location and was originally designed to extract x from predictions of the ith bit of the

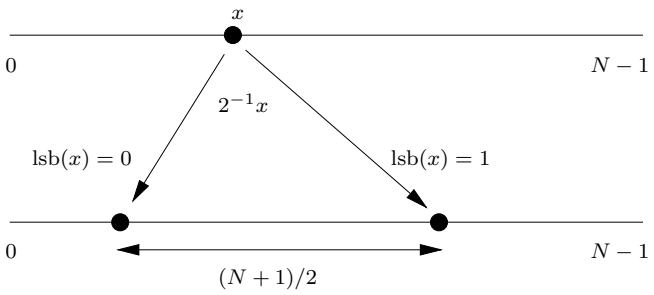

Fig. 1. Division by 2 in $\mathbb{Z}_N$. Values that only differ in their lsb's are mapped to points $\frac{N+1}{2}$ apart.

function $x \mapsto [ax+b]_p$, where p is an $\Omega(\|x\|)$-bit prime and a, b are random elements in $\mathbb{Z}_p$.

To handle the internal bits, the main idea in [Näslund 1996] is to convert the predictor for the ith bit into a predictor 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 Figure 2.

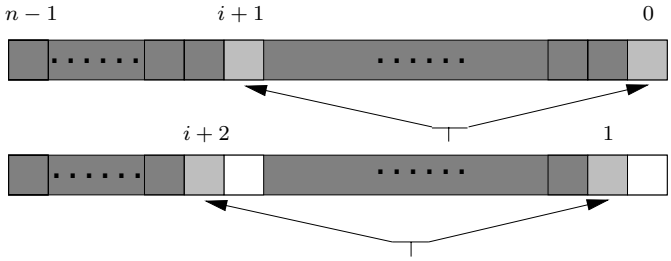

Fig. 2. Deciding bits two-by-two.

As mentioned, a closer study of this work reveals that the methods in fact do not apply for some "highly structured" predictors that behave in a certain way. On the other hand, the predictors for which the methods fail are indeed of a very special nature that we can exploit. The plan is now: (a) Investigate how, and when, the methods in [Näslund 1996] are applicable to provide an extractor from a CMHNP. (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 predictor), and furthermore, the predictor must then have a certain structure. (c) Prove that for bad N, i, and predictors as specified by (b), it is possible either to find a factor in N or to construct an algorithm, i.e. a new predictor, $\mathcal{P}'$, using the original predictor $\mathcal{P}$ as a black box, such that $\mathcal{P}'$ is a lsb-predictor which can then be used as in [Fischlin and Schnorr 1997].

We start by giving some generalizations of well-known sampling techniques and then formalize how the method by Fischlin and Schnorr is used as a "warm-up". We then follow (a), (b), (c) as above.

5.3 Sampling Techniques

We assume that we have a predictor $\mathcal{P}$ that given a, predicts the ith bit of $[ax]_N$ with probability at least $\frac{1+\epsilon(n)}{2}$ where $\epsilon(n)$ is non-negligible.

Definition 5.1. By an *interval*, J, we mean a set of consecutive values $J = \{[u]_N, [u+1]_N, \ldots, [v]_N\}$ in $\mathbb{Z}_N$. 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 an interval $J \subset \mathbb{Z}_N$, let $P^{\mathcal{P}}(J)$ be the fraction of 1-answers the predictor gives when a is picked such that ax is uniformly random in J,

$$P^{\mathcal{P}}(J) \triangleq \mathrm{E}_{ax \in J}[\mathcal{P}(a)] = \Pr_{ax \in_{\mathcal{U}} J}[\mathcal{P}(a) = 1].$$

For $J_1, J_2 \subset \mathbb{Z}_N$ we define

$$\Delta^{\mathcal{P}}(J_1, J_2) \triangleq \left| P^{\mathcal{P}}(J_1) - P^{\mathcal{P}}(J_2) \right|.$$

As discussed before it is important to generate random numbers r_j with partial information about $[r_j x]_N$ and our key tool is the following lemma.

Lemma 5.2. *Let $m(n) \in \mathrm{poly}(n)$. Suppose for some $d_I(n), d_Y(n)$, we are given $r, s \in_{\mathcal{U}} \mathbb{Z}_N$, together with*

$$\mathrm{B}^i_{i-d_I(n)}([rx]_N), \qquad \mathrm{B}^i_{i-(d_I(n)+\log m(n))}([sx]_N), \tag{5.2}$$

and

$$\left\lfloor \frac{2^{1+d_Y(n)}[rx]_N}{N} \right\rfloor, \left\lfloor \frac{2^{(1+d_Y(n)+\log m(n))}[sx]_N}{N} \right\rfloor. \tag{5.3}$$

Then we can in polynomial time generate a list of $m(n)$ values $\{r_j\}$ so that each $[r_j x]_N$ is uniformly distributed and the values in $\{[r_j x]_N\}$ are pairwise independent. Furthermore, we find numbers $\{z_j^I\}$ and $\{z_j^Y\}$ such that for some z_j with $[z_j]_N = [r_j x]_N$, we have

$$\left| z_j - z_j^Y \right| \leq \frac{N}{2^{d_Y(n)}} \tag{5.4}$$

and

$$\mathrm{abs}_{2^{i+1}}(z_j - z_j^I) \leq 2^{i+1-d_I(n)}. \tag{5.5}$$

Similar techniques were used already in [Alexi et al. 1988], where, however, it was only necessary to know the lsb of each point. The construction is by now standard.

Proof. We let $r_j = (r + js)$ for $0 \leq j \leq m(n)-1$. From the given information on $[rx]_N$ and $[sx]_N$ it is straightforward to see that $\{[r_j x]_N\}$ has the desired properties and how to obtain the numbers z_j^I, z_j^Y. □

The reason for the numbers z_j is that when $abs_N(r_j x)$ is small then we are uncertain on the number of modular reductions to perform and we have two guesses on the bits around position i. One in the case when $[r_j x]_N$ is almost 0 and one when it is almost N.

Note in particular that (5.4) implies that each $[r_j x]_N$ is $2^{-d_Y(n)}$-determined.

An important convention: Point sets as specified by Lemma 5.2 are used many times by our extractor with various choices of the parameters $m(n), d_Y(n)$, and $d_I(n)$. Each time we use the same values of r and s. These number are originally chosen randomly but remains fixed throughout the execution. We try all possibilities for the information needed (as specified by (5.2), (5.3)) to compute the numbers z_j. It will be the case that we always have $\log(m(n)), d_Y(n), d_I(n) \in O(\log n)$. For all the incorrect possibilities we do not care what happens, the extractor might output a number and it might not, but we note that there are only a polynomial number of possibilities to try. Our only concern is that we are likely to extract x when we have the correct parameters and hence from now on we only analyze what happens when indeed we have been given the correct information and hence we can assume we have pairwise independent numbers r_j together with numbers z_j^I and z_j^Y that satisfy (5.4) and (5.5).

We have the following lemma that is extremely useful for us.

LEMMA 5.3. *Let $\{r_j\}$ be a set of $m(n)$ values such that $\{[r_j x]_N\}$ are uniformly distributed and pairwise independent. Let $J \subset \mathbb{Z}_N$ with $\lambda(J)$ non-negligible and such that for each j, whether $[r_j x]_N \in J$ can be decided except with probability δ.*

Then there is an absolute constant c such that for any non-negligible $\epsilon'(n)$, and $K(n) \in \mathrm{poly}(n)$, if $m(n) = c\lambda(J)^{-1}\epsilon'(n)^{-2}K(n)$ and $\delta \leq c^{-1}\lambda(J)\epsilon'(n)/K(n)$ it is, in probabilistic polynomial time, possible to compute a value $\tilde{p}$ such that

$$\Pr[\left|P^{\mathcal{P}}(J) - \tilde{p}\right| \geq \epsilon'(n)] \leq \frac{1}{K(n)}.$$

PROOF. The estimate $\tilde{p}$ is simply the fraction of the points that is classified to lie in J and that get the answer one. There are two error sources to this number: that points are incorrectly classified and that the fraction of points getting the answer one does not agree with the expected value. We expect at most $\delta m(n) \leq \epsilon^{-1}$ points to be misclassified and with probability at most $1/2K(n)$ this number is bounded by $2K(N)\epsilon^{-1}$. Since the points that are in J are pairwise independent the standard deviation on the number of points that give the answer one is bounded by $(cK(n))^{1/2}\epsilon^{-1}$. The result now follows by a standard application of Chebychev's inequality. □

Lemma 5.3 is used in two different ways and let us explain how. In our applications the set r_j is either going to be a set produced by Lemma 5.2, or, such a set offset by an adding a fixed number a to each r_j. In both cases the pairwise independence is clear from construction. The information $\{z_j^I\}$ and $\{x_j^Y\}$ is used to determine whether the points belong to the set J and the error δ comes from the fact that some point are close to the border of J and there the given information might not be sufficient to always make a correct decision.

The first use of the lemma is to approximate $P^{\mathcal{P}}(J)$ for a given J. Once this is done, the lemma is applied as follows. Our determination of whether the points lie in J can be correct only subject to certain guessed information about x being correct (usually the values of some yet unknown bits in ax when an additive a is used in the points). Thus, in this case if the obtained estimate $\tilde{p}$ is not close to

the known (previously approximated) value of $P^{\mathcal{P}}(J)$, we can conclude that this guess is incorrect and discard this possibility for the unknown bits in x. We use this repeatedly to trim a list of candidates for bits of x, in the end leaving us with the correct choice.

6. PROOF OF THE MAIN THEOREM

We first present two methods for extracting x. The first is that of Fischlin and Schnorr while the second is due to Näslund. By the result of [Alexi et al. 1988] we can assume that $i \geq c \log n$ for any suitable constant c.

6.1 Extraction, Method 1

The main technical lemma of this section is given below. It generalizes slightly lemmas from [Ben-Or et al. 1983; Alexi et al. 1988; Fischlin and Schnorr 1997].

LEMMA 6.1. *If $\mathcal{P}$ is such that for some interval J we have $\Delta^{\mathcal{P}}(J, J+(N+1)/2) \geq \epsilon'(n)$, where $\lambda(J), \epsilon'(n)$ are non-negligible, then we can in random polynomial time construct a predictor, $\mathcal{P}'$ such that for all $\frac{\lambda(J)\epsilon'(n)}{96cn}$-determined $[ax]_N$, $\mathcal{P}'$ determines $\mathrm{lsb}([ax]_N)$ with probability at least $1 - \frac{1}{2n}$. Here c is the constant from Lemma 5.3.*

We later see how to use such an oracle to find x in a straightforward way using the methods of [Fischlin and Schnorr 1997].

PROOF. By Lemma 5.3 applied with a set of points generated as described in Lemma 5.2 we can assume that we have $\tilde{p}_0, \tilde{p}_1$, approximations to $P^{\mathcal{P}}(J)$, $P^{\mathcal{P}}(J + (N+1)/2)$ respectively, within $\epsilon'(n)/6$. This can be made to hold with probability at least $1 - 1/(4n)$, and we assume for concreteness that $\tilde{p}_1 > \tilde{p}_0$. Our uncertainty in determining whether these points lie in J, i.e. the δ in Lemma 5.3 comes from the fact that the numbers z_j^Y only give the approximate location of each point. By choosing $d_Y(n)$ sufficiently large (but remaining $O(\log n)$) we can make this probability smaller than $\lambda(J)\epsilon'(n)/(48cn)$.

Now we again apply Lemma 5.3 to points $[2^{-1}a + r_j]_N$ where r_j is again generated as in Lemma 5.2. If we assume that $\mathrm{lsb}(ax) = 0$ we can, since $[ax]_N$ is $\frac{\lambda(J)\epsilon'(n)}{96cn}$-determined, based on this assumption determine a new approximation of $P^{\mathcal{P}}(J)$ which should be close to $\tilde{p_0}$. If on the other hand the assumption is incorrect and $\mathrm{lsb}(ax) = 1$ then the computed number instead is an approximation of $P^{\mathcal{P}}(J + (N+1)/2)$ and should be close to $\tilde{p_1}$. Thus, we guess that the lsb is one if the computed estimate is at least $(\tilde{p_0} + \tilde{p_1})/2$ and otherwise we guess 0. By the choice of parameters the probability of making an error is bounded by $\frac{1}{2n}$. □

Let us see how to use Lemma 6.1 to recover x.

LEMMA 6.2. *If $\mathcal{P}$ is such that for some interval J we have $\Delta^{\mathcal{P}}(J, J+(N+1)/2) \geq \epsilon'(n)$, where $\lambda(J), \epsilon'(n)$ are non-negligible, then we can, in random polynomial time, recover x with probability at least 1/2.*

PROOF. We use Lemma 6.1 to construct $\mathcal{P}'$ which takes as inputs the multiplier a and the approximation y of ax showing that it is well determined. In fact, as Lemma 6.1 holds for all well-determined ax, we can simply set $a = 1$, guessing the magnitude of x itself.

ALGORITHM 6.3.

Output: x
(1) *"guess" y so that* $\mathrm{abs}_N(x-y) \leq N\lambda(J)\epsilon'(n)/6$
(2) **for** $j := 0$ **to** $n-1$ **do**
(3) $\quad b \leftarrow \mathcal{P}'(2^{-j}, y)$
(4) $\quad y \leftarrow b(N+1)/2 + (y-b)/2$*;*
(5) **return** $[y2^n]_N$

A sufficiently dense set of possible values of y can be tried in polynomial time and thus "guessing" can in fact replaced by a polynomially bounded loop. By induction, provided that all the predictor calls are answered correctly, y is at the call to $\mathcal{P}'$ for a particular value of the loop variable j, an approximation of $2^{-j}x$ within $2^{-j}N\lambda(J)\epsilon'(n)/6$. This implies that the preconditions of the parameters sent to the predictor remains correct and with probability at least $1 - n \cdot \frac{1}{2n} = 1/2$ we get n correct answers from the predictor. This implies that at the end of the algorithm y is in fact exactly $2^{-n}x$ and the algorithm is correct. □

We next proceed to describe an alternate way to use a predictor to extract x. It applies to an arbitrary bit position and is the main extractor used.

6.2 Extraction, Method 2

This second method is much more technical than the previous, and we start by outlining the ideas. This method follows the principles used in [Näslund 1996].

First we introduce some parameters, let

$$\tau(n) \triangleq 34 + 5\log\epsilon(n)^{-1} + \log n.$$

We first prove Theorem 4.2 assuming $i \leq n - 2\tau(n) - 1$. This assumption bounds the bias of the ith bit and it also allow us to control wrap-around mod N when sampling.

The idea is to use the predictor for the ith bit to decide both the lsb and the $i+1$st bit. To this end, we aim to measure the effect these two bits have on the ith bit when "shifting" x. Suppose that we already know the value of $\mathrm{B}_{i-d+1}^{i}(x)$, the value of the d bits to the right of, and including bit i. Initially we assume we are given such an approximation. The most intuitive approach would again be to ask the predictor on $a = [2^{-1}]_N$. For technical reasons (explained in Section 6.3) we, however, use $a = [2^{-\tau}]_N$ where as defined above $\tau \in \Theta(\log n)$. We now 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 $\mathrm{B}_{i+1}^{i+\tau}(x)$ and $\mathrm{B}_{0}^{\tau-1}(x)$. Hence, an entry in this list looks like (u_j, v_j), $0 \leq u_j, v_j \leq 2^{\tau}-1$, u_j corresponding to a possibility for $\mathrm{B}_{i+1}^{i+\tau}(x)$ and v_j to a possibility for $\mathrm{B}_{0}^{\tau-1}(x)$. The two bits we are after, $\mathrm{bit}_{i+1}(x)$ and $\mathrm{lsb}(x)$, then corresponds to $\mathrm{lsb}(u_j)$ and $\mathrm{lsb}(v_j)$, respectively. Our goal is to "trim" this list, leaving us the correct choice.

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 shall try to exclude one of them (the incorrect one if one is correct). Furthermore, since we only aim to determine the two bits $\mathrm{bit}_{i+1}(x), \mathrm{lsb}(x)$, 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 $\mathrm{lsb}(v_1) \neq \mathrm{lsb}(v_2)$.

Now consider $[2^{-\tau}x]_N$.

$$[2^{-\tau}x]_N = \frac{x - \mathrm{B}_{i+1}^{i+\tau}(x)2^{i+1} - \mathrm{B}_{i-d+1}^{i}(x)2^{i-d+1} - \mathrm{B}_0^{\tau-1}(x)}{2^\tau} + \mathrm{B}_{i+1}^{i+\tau}(x)2^{i+1-\tau} + \mathrm{B}_{i-d+1}^{i}(x)2^{i-d+1-\tau} + \mathrm{B}_0^{\tau-1}(x)[2^{-\tau}]_N. \quad (6.1)$$

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

To decide if $(\mathrm{B}_{i+1}^{i+\tau}(x), \mathrm{B}_0^{\tau-1}(x))$ equals (u_1, v_1) or (u_2, 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 $z = z' + u_1 2^{i+1-\tau} + v_1[2^{-\tau}]_N$, $u = u_2 - u_1$, and $v = v_2 - v_1$. Since are only interested in the differences, we may interchange (u_1, v_1) and (u_2, v_2) to ensure that $v \geq 0$. Because at least one of the pairs u_1, u_2 and v_1, v_2 differs in their least significant bit, we know that 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 6.4. For given $N, \tau(n)$ and $0 \leq v \leq 2^{\tau(n)} - 1$, $|u| \leq 2^{\tau(n)} - 1$, define

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

Clearly this number depends on N, i and $\tau(n)$ but as they remain constant throughout the argument we suppress this dependence.

Note that $\alpha(u, v)$ is computed modulo N, not modulo 2^{i+1}. Again, we emphasize that we are only interested in $\alpha(u, v)$ where at least one of u, v is odd.

Just like we in the previous section wanted to find sets J, $J + (N+1)/2 = J + [2^{-1}]_N$, where the predictor behaved differently, we can now ask if there are similar sets $S, S + \alpha(u, v)$ where the predictor behaves differently.

Consider a particular (u, v) and fix $S \subset \mathbb{Z}_N$ so that all $z \in S$ have the same value for their ith bit. We cannot let S be an interval as before, since the length of 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. The basis for this approach is formalized in the following definitions.

Definition 6.5. We write N as $N \triangleq N_1 2^{i+1} + N_0$ where $N_0 < 2^{i+1}$, and further $N_1 \triangleq N_3 2^{\tau(n)} + N_2$ where $N_2 < 2^{\tau(n)}$.

Definition 6.6. Let $I \triangleq \mathbb{Z}_{2^{i+1}} = \{0, 1, \ldots, 2^{i+1} - 1\}$ and $Y \triangleq \mathbb{Z}_{N_1+1} = \{0, 1, \ldots, N_1\}$. 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 π is surjective, except for some values of the form (j, N_1) with $j \geq N_0$. We would like to draw the readers attention to the fact that since we are really working modulo N, the value z that $\pi(\cdot)$ is applied to should, when necessary, first be reduced modulo N. Such modular reductions could cause problems. For this reason, we mostly, but not always, arrange things so that the argument z (even when z is the sum of elements in $\mathbb{Z}_N$) can be considered as an integer in the range $[0..N-1]$.

Definition 6.7. We define the *plane* $\Pi(N, i) = (I \times Y) \cap \pi(\mathbb{Z}_N)$. For $b \in \{0, 1\}$ we set

$$S^{(b)} \triangleq \{z \in \mathbb{Z}_N \mid \mathrm{bit}_i(z) = b\}.$$

For all non-negative integers we define a *box*, S, of *width* w and *height* h as the following rectilinear subset of $I \times Y$:

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

The *measure* of such a box is $\lambda(S) \triangleq \frac{\#S}{N} = \frac{wh}{N}$ provided that $h < N_1$ and $w \leq 2^{i+1}$. Furthermore, for a box S and $z \in \mathbb{Z}_N$ we define the *z-translation* of S as

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

A *level* is a subset of $\Pi(N, i)$ consisting of the set of values having a fixed π_Y-value. All levels except possibly the N_1th level are of size 2^{i+1}.

Finally, if S is a box, we define as before

$$P^{\mathcal{P}}(S) \triangleq \Pr_{ax \in_{\mathcal{U}} S}[\mathcal{P}(a) = 1],$$

and

$$\Delta^{\mathcal{P}}(S, S') \triangleq \left|P^{\mathcal{P}}(S) - P^{\mathcal{P}}(S')\right|.$$

Figure 3 illustrates the plane.

We now state the main lemma of this section.

Lemma 6.8. *Suppose that for all* $0 \leq v \leq 2^{\tau(n)} - 1$, $|u| \leq 2^{\tau(n)} - 1$, u *or* v *odd, there is a box* $S_{u,v}$ *of width at least* $w(n)2^{i+1}$, *height at least* $h(n)N_1$, *and with* $\Delta^{\mathcal{P}}(S_{u,v}, S_{u,v} + \alpha(u, v)) \geq \epsilon'(n)$, *where* $h(n), w(n), \epsilon'(n)$ *are all non-negligible. Define*

$$d(n) \triangleq \min(i, \log \epsilon'(n)^{-1} + \log(w(n)h(n))^{-1} + 9 + 2\tau(n) + \log n).$$

Then it is possible to construct a predictor, $\mathcal{P}'$, *that given* j, $\mathrm{B}_{i-d(n)}^{i+j}(x)$, $\mathrm{B}_0^{j-1}(x)$, *and* y *so that* $\mathrm{abs}_N(x - y) \leq 2^{-d(n)}N$, *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* $1 - \frac{1}{2n}$.

Proof. We assume that $i > d(n)$ and that $j \leq \min\{i - d(n), n - (i + 1)\}$. Otherwise, only one of the two bits $\mathrm{bit}_{i+j+1}(x), \mathrm{bit}_j(x)$ is unknown, and the situation gets less complicated, but for notational simplicity we only consider the general case.

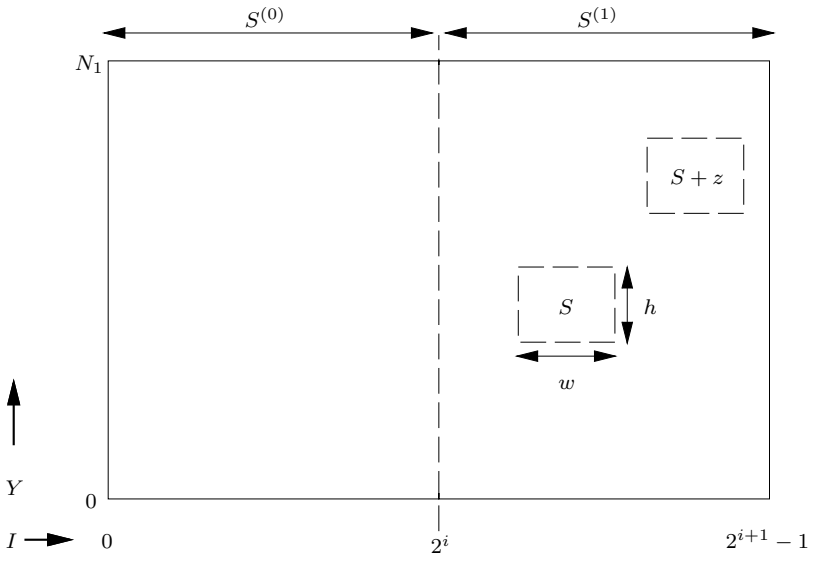

Fig. 3. The $\Pi(N,i)$-plane. Shown is a typical box, S, and a translation, $S+z$.

The proof is very similar to the proof of Lemma 6.1. We need only observe that the given information z_j^I in Lemma 5.2 makes it possible to determine whether a point $[r_j x]_N$ belongs to $S_{u,v}$ except with a probability, δ, that is smaller than a desired inverse polynomial.

We define $\lambda_{u,v} \triangleq \lambda(S_{u,v})$ and let $\tilde{p}_{u,v}$ and $\tilde{p}'_{u,v}$ be estimates for $P^{\mathcal{P}}(S_{u,v})$ and $P^{\mathcal{P}}(S_{u,v}+\alpha(u,v))$ respectively such that $|\tilde{p}_{u,v} - P^{\mathcal{P}}(S_{u,v})| \leq \epsilon(n)'/8$ and $|\tilde{p}'_{u,v} - P^{\mathcal{P}}(S_{u,v}+\alpha(u,v))| \leq \epsilon(n)'/8$ is true for all (u,v) with probability $1-1/(8n)$. Assume for notational simplicity that we always have $\tilde{p}'_{u,v} > \tilde{p}_{u,v}$ and note that such numbers can be found by Lemma 5.3.

By Lemma 5.2, we can generate polynomially many, $m(n)$, sample points of the form $r_k x$ where for some z_k, $[r_k x]_N = [z_k]_N$, z_k is known within $2^{-d(n)}N$ and with $[z_k]_{2^{i+1}}$ known with a relative error of at most $2^{-d(n)}$. Exactly which polynomial number is needed can be computed but since we do not wish to make an exact analysis we leave this at an unspecified polynomial $m(n)$.

The procedure to decide two new bits in x is:

Algorithm 6.9.

Output: $(\mathrm{bit}_{i+j+1}(x), \mathrm{bit}_j(x))$
(1) $T \leftarrow \{0,1\}^{\tau(n)} \times \{0,1\}^{\tau(n)}$
(2) **while** $\exists\,(u_1,v_1),(u_2,v_2) \in T$ *s.t.* $\mathrm{lsb}(u_1) \neq \mathrm{lsb}(u_2)$ **OR** $\mathrm{lsb}(v_1) \neq \mathrm{lsb}(v_2)$ **do**
(3) *possibly exchange* (u_1,v_1), (u_2,v_2) *to ensure* $v_2 \geq v_1$
(4) $(u,v) \leftarrow (u_2-u_1, v_2-v_1)$; $\alpha \leftarrow \alpha(u,v)$
(5) $R = \{\}$
(6) **for** $k := 1$ **to** $m(n)$ **do**
(7) $\pi' \leftarrow$ *approximation to* $\pi([(r_k + 2^{-(j+\tau(n))})x]_N)$

based on $j, \tau(n)$ and (u_1, v_1) being the correct choice
(8) **if** $\pi' \in S_{u,v}$ **then**
(9) $R \leftarrow R \cup \{r_k + 2^{-(j+\tau(n))}\}$
(10) $p \leftarrow$ *number of 1-answers of* $\mathcal{P}$ *on* R
(11) **if** $p \leq \lambda_{u,v} m(n)(\tilde{p}_{u,v} + \tilde{p}'_{u,v})/2$ **then**
(12) *delete* (u_2, v_2) *from* T
(13) **else**
(14) *delete* (u_1, v_1) *from* T
(15) *pick any* $(u, v) \in T$; **return** $(\mathrm{lsb}(u), \mathrm{lsb}(v))$

Some comments may be in place. The while-loop runs over pairs of candidates for $\mathrm{B}_{i+1}^{i+j+\tau(n)}(x)$, $\mathrm{B}_j^{j+\tau(n)-1}(x)$, and terminates when all remaining pairs have the same value both for $\mathrm{lsb}(\mathrm{B}_{i+1}^{i+j+\tau(n)}(x))$ (corresponding to $\mathrm{bit}_{i+j+1}(x)$) and $\mathrm{lsb}(\mathrm{B}_j^{j+\tau(n)-1}(x))$ (i.e. $\mathrm{bit}_j(x)$), meaning that we hopefully have decided two new bits in x.

The computations in line 7 are made as if (u_1, v_1) is correct and if so the π'-values computed are good approximations to the true π-values. Therefore, the distribution on the set R is close to uniform over $S_{u,v}$ and pairwise independent. If instead, (u_2, v_2) is correct, then R consists of values close to the uniform distribution on $S_{u,v} + \alpha(u, v)$. The lemma now follows by Lemma 5.3. □

Given the predictor $\mathcal{P}'$ of Lemma 6.8 it is not difficult to extract x.

LEMMA 6.10. *Given the same assumptions as Lemma 6.8, we can extract x in random polynomial time with probability of success at least $\frac{1}{2}$.*

PROOF. Apply Lemma 6.8 and get the resulting predictor $\mathcal{P}'$. The inversion algorithm is now as follows.

ALGORITHM 6.11.

Output: x
(1) *"guess"* y *so that* $\mathrm{abs}_N(x - y) \leq 2^{-d(n)}N$
(2) *"guess"* $z' = \mathrm{B}_{i-d(n)+1}^{i}(x)$; $z \leftarrow 0$ /* $z = \mathrm{B}_0^{j-1}(x)$ */
(3) **for** $j := 0$ **to** $\max(n - (i+1), i - d(n))$ **do**
(4) $(b', b) \leftarrow \mathcal{P}'(j, z', z, y)$ /* $\mathrm{bit}_{i+1+j}(x), \mathrm{bit}_j(x)$ */
(5) **if** $i + j < n$ **then** $z' \leftarrow 2^{j+d(n)}b' + z'$ /* $\mathrm{B}_{i-d(n)+1}^{i+j}(x)$ */
(6) **if** $j + d(n) < i$ **then** $z \leftarrow 2^j b + z$ /* $\mathrm{B}_0^j(x)$ */
(7) $y \leftarrow b(N+1)/2 + (y - b)/2$
(8) **return** $z'2^{i+1-d(n)} + z$

We repeat the process for all the polynomially many choices for y, z', so we may assume that we have a correct guess. If the predictor does not err, the final $z'2^{i+1-d(n)} + z$ is the correct binary representation of x. Since $\mathcal{P}'$ is used at most n times, the total error probability is at most $n\frac{1}{2n} = \frac{1}{2}$. □

The key to the overall proof is thus to establish the existence of the boxes needed for Lemma 6.10 or the interval needed for Lemma 6.2. This is the topic of the next section.

Before continuing let us, however, explain one point. We do not only need the existence of the given boxes/intervals but also that they can be found efficiently.

Most of our proofs are in fact efficient in this sense, but this is really not needed. If S is a good box of non-negligible size then so is any other box sufficiently close to S. It is not hard to see that once we have non-negligible lower bounds for the size and the advantage then we can in fact specify a polynomial number of candidates $\{S_j\}$ such that if a good box exists then in fact one of the S_j is also good, and only of slightly inferior quality. This S_j can then be located by Lemma 5.3. This implies that existence is equivalent to efficiently being able to find a desired object and hence we can safely ignore this point.

6.3 Proving Existence of Good Boxes/Intervals

For "most" predictors it turns out that the boxes needed for Lemma 6.10 do exist. Unfortunately, they do not exist for all predictors and there are a number of cases to consider to understand in which situations we fail to have all the necessary good boxes. Before we state and prove the technical lemmas let us give a short overview of the argument.

The case when the two considered alternatives for x are such that v is even and u is odd is simple and we can argue that there must be good boxes at distance $\alpha(u, v)$. In this case we have to choose between two alternatives that differ in the $(i + 1)$st bit but have the same least significant bit. In this case, having $\tau(n) = 1$ would have been sufficient but in the now general situation a translation of $2^{\tau(n)-1}\alpha(u, v)$ maps inputs that have ith bit equal to zero to those with ith bit equal to one (and vice versa) and it is not difficult to find the desired box. This is handled in Lemma 6.12.

The situation when the two alternatives for x differ in the least significant bit and hence v is odd is more complicated. Take any box S and look at its translates $S + k\alpha(u, v)$ for $k = 1, 2 \ldots, 2^{\tau(n)} - 1$. If there are two translates (by the triangle inequality, not necessarily adjacent) on which the predictor behaves differently we are done so let us assume that this is not the case. Call the union of all these translated the boxes an *orbit*. Since by assumption the predictor has an advantage in predicting the ith bit there must be some orbit on which it has an advantage. Since the predictor behaves (almost) the same on all boxes within the orbit, the only way this can happen is when the orbit is not equally divided among strings with ith bit equal to one and those with ith bit zero. For this to happen it turns out that a certain rational number $\tilde{\alpha}(u, v)/2^{i+1}$ (with $\tilde{\alpha}(u, v)$ being is closely related to $\alpha(u, v)$) must have a good rational approximation, r/s, with small odd denominator. When proving this, our formal lemma is phrased as "if there is not a good rational approximation of $\tilde{\alpha}(u, v)/2^{i+1}$ then there must be two boxes at distance $\tilde{\alpha}(u, v)$ on which the predictor behaves noticeably differently".

To prove this we first prove, in Lemma 6.14, that if there is no good rational approximation then there must be some boxes within an orbit on which the predictor behaves differently. The reason for this is that if there is no good rational approximation (to $\tilde{\alpha}(u, v)/2^{i+1}$) then the the projection of the orbit when looking mod 2^{i+1} is very uniform, and absence of said boxes would contradict our assumption on the predictor.

Finally, when there is a good rational approximation, r/s, to $\tilde{\alpha}(u, v)/2^{i+1}$, there are two cases to consider. The case of a good approximation with even denominator, s, is handled in Lemma 6.21 and turns out to be simple. Here, a translation by $s\tilde{\alpha}(u, v)$ takes us essentially back where we started mod 2^{i+1}. Thus, when s is even

we can look at what happens after an $s/2$ translation and it turns out that this translation gives almost a bijection between strings with ith bit zero and those with ith bit one. We are then back essentially to the same argument as in the case above when v was even and we must have good boxes at such a distance.

In the final case when we have a good rational approximation with odd denominator s we cannot prove the existence of the good boxes in all cases. For a "troublesome" ith bit predictor we are, however, then able to turn it into a predictor of the least significant bit. This is done by predividing all multipliers input to the predictor by a suitable constant derived from the good rational approximation. A potential problem is that this constant might not be invertible mod N. In this case, however, we have discovered a nontrivial factor of N and by the theorem statement we are allowed to terminate the algorithm at this point.

We now proceed to supply the technical lemmas and proofs following the above outline. We start with the case when v is even. Recall that $\epsilon(n)$ is the assumed advantage of the predictor.

LEMMA 6.12. *If v is even and u is odd there is a $k \leq 2^{\tau(n)} - 1$ such that*

$$\Delta^{\mathcal{P}}(S^{(0)} + k\alpha(u,v), S^{(0)} + (k+1)\alpha(u,v)) \geq \epsilon(n)2^{-\tau(n)}.$$

PROOF. Setting $v = 2v'$ we have

$$2^{\tau(n)-1}\alpha(u,v) \equiv u2^i + v' \mod N.$$

Since u is odd, this implies that

$$\lambda((S^{(0)} + 2^{\tau(n)-1}\alpha(u,v))\triangledown S^{(1)}) \leq 2^{\tau(n)}\frac{2^i}{N} + 2^{\tau(n)-i} \leq \epsilon(n)/3.$$

The two error terms comes from the probability of $u2^i$ causing a reduction modulo N and of v' causing a shift modulo 2^{i+1} respectively. The last inequality is due to the definition of $\tau(n)$ and the assumption made on i.

By assumption on $\mathcal{P}$,

$$\Delta^{\mathcal{P}}(S^{(0)}, S^{(1)}) \geq \epsilon(n) - \beta_i(N),$$

where $\beta_i(N)$ is the bias of the ith bit. Since the bias is bounded by $\epsilon(n)/6$ for the range of i we are considering we conclude that

$$\Delta^{\mathcal{P}}\left(\left(S^{(0)} + 2^{\tau(n)-1}\alpha(u,v)\right), S^{(0)}\right) \geq \epsilon(n)/2.$$

The existence of the k in the lemma now follows by the triangle inequality. □

In general, the magnitude of $\alpha(u,v)$ is given by the "v-component". Since we have no reason to expect that v is significantly smaller than $2^{\tau(n)}$ the size of $\alpha(u,v)$ is comparable to that of N, and during our translations we need to constantly do reductions mod N. This creates complications and hence we work with a number $\tilde{\alpha}(u,v)$ which is essentially $k\alpha(u,v)$ where k is chosen to make this number of minimal size ($k = [-v^{-1}N]_{2^{\tau(n)}}$). Specifically, the magnitude of $\tilde{\alpha}(u,v)$ will be about $N/2^{\tau(n)}$. The translates $j\tilde{\alpha}(u,v), j = 1, 2 \ldots, 2^{\tau(n)} - 1$ are then essentially the same as the translates $j\alpha(u,v), j = 1, 2 \ldots, 2^{\tau(n)} - 1$, but in a permuted order.

We now formally define $\tilde{\alpha}(u,v)$ and prove its relation to $\alpha(u,v)$.

Definition 6.13. For $0 < v \leq 2^{\tau(n)} - 1$, v odd, and $|u| \leq 2^{\tau(n)} - 1$, define $u' = [-uv^{-1}N]_{2^{\tau(n)}}$ and let

$$\tilde{\alpha}(u,v) \triangleq u'2^{i+1-\tau(n)} + \left\lceil \frac{N}{2^{\tau(n)}} \right\rceil.$$

The consequence of the relationship between $\alpha(u,v)$ and $\tilde{\alpha}(u,v)$ outlined above that we need is given by the lemma below. Here we aim for simplicity of proofs rather than getting the best bounds.

LEMMA 6.14. *Let v be odd. If there is a box S' of height h and width w such that $\Delta^{\mathcal{P}}(S', S' + \tilde{\alpha}(u,v)) \geq \epsilon'(n)$, then there is a box S of the same dimensions and with*

$$\Delta^{\mathcal{P}}(S, S + \alpha(u,v)) \geq \frac{\epsilon'(n)}{2^{\tau(n)}} - \frac{2}{h} - \frac{2}{w}.$$

PROOF. Let $k = [-v^{-1}N]_{2^{\tau(n)}}$. Then

$$\begin{aligned} k\alpha(u,v) &\equiv [-v^{-1}N]_{2^{\tau(n)}}(u2^{i+1-\tau(n)} + v[2^{-\tau(n)}]_N) \equiv \\ &\equiv (u' + c_1 2^{\tau(n)})2^{i+1-\tau(n)} + (-N + c_2 2^{\tau(n)})[2^{-\tau(n)}]_N \equiv \\ &\equiv u'2^{i+1-\tau(n)} + c_1 2^{i+1} + c_2 \bmod N, \end{aligned}$$

where $0 \leq c_1 < 2^{\tau(n)}$ and $0 \leq -N + c_2 2^{\tau(n)} \leq 2^{2\tau(n)}$. This implies that

$$k\alpha(u,v) - \tilde{\alpha}(u,v) = c_1 2^{i+1} + c_2' \bmod N,$$

where $c_2' = c_2 - \lceil \frac{N}{2^{\tau(n)}} \rceil$ and hence $0 \leq c_2' < 2^{\tau(n)}$. We conclude that

$$\#\left((S' + \tilde{\alpha}(u,v)) \bigtriangledown (S' + k\alpha(u,v))\right) \leq 2c_1 w + 2c_2' h.$$

Hence

$$\Delta^{\mathcal{P}}(S', S' + k\alpha(u,v)) \geq \epsilon'(n) - \frac{2c_1}{h} - \frac{2c_2'}{w},$$

and the existence of two neighboring translates where the predictor behaves differently follows by the triangle inequality. □

Lemma 6.14 allows us to study sequences/orbits of the form

$$\{j\tilde{\alpha}(u,v)\}_{j\geq 0} = \{j(u'2^{i+1-\tau(n)} + \lceil N/2^{\tau(n)} \rceil)\}_{j\geq 0},$$

rather than $\{j\alpha(u,v)\}_{j\geq 0}$. The key benefit of this is that the former sequence is strictly increasing with respect to $\pi_Y(\cdot)$. Also, since $u' < 2^{\tau(n)}$ and $2^{i+1} < N/2^{2\tau(n)}$ (from the upper bound on i), we never need to perform any modular reductions modulo N, i.e.

$$[j(u'2^{i+1-\tau(n)} + \lceil N/2^{\tau(n)} \rceil)]_N \equiv j(u'2^{i+1-\tau(n)} + \lceil N/2^{\tau(n)} \rceil), \quad 0 \leq j \leq 2^{\tau(n)} - 1,$$

and this simplifies the analysis. The central point point of the rest of the proof is to study how the sequence $\{j\tilde{\alpha}(u,v)\}_{j\geq 0}$ behaves modulo 2^{i+1}. One key property is whether $\tilde{\alpha}(u,v)2^{-(i+1)}$ can be well approximated by a rational number with small denominator. This property is closely related to how uniformly the multiples of $\tilde{\alpha}(u,v)$ are distributed modulo 2^{i+1} and this connection is essential for us. We need some definitions.

Definition 6.15. The number $\zeta \in \mathbb{Q}$ is said to be of (Q, ψ)-*type* if for all integers r, s, $0 < s \leq Q$ and $(r, s) = 1$:

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

Definition 6.16. Define $Q(n) \triangleq 2^{10}\epsilon(n)^{-1}$, $\psi(n) \triangleq \frac{\epsilon(n)2^{\tau(n)}}{2^{12}\log^2 Q(n)}$.

Just for intuition note that the total measure of rationals that are not of $(Q(n), \psi(n))$-type is at most

$$\sum_{s=1}^{Q(n)} \frac{1}{s\psi(n)} \approx \frac{\ln Q(n)}{\psi(n)}$$

which in our case is very small and thus a typical number is of $(Q(n), \psi(n))$-type. However, since we have $2^{2\tau(n)}$ numbers (of form $\zeta_{u,v} = \tilde{\alpha}(u, v)/2^{i+1}$) to consider, we cannot exclude that some of them are "bad".

We are now ready to state the three main lemmas needed to prove the main theorem.

LEMMA 6.17. *Let v be odd. If the rational number $\tilde{\alpha}(u, v)/2^{i+1}$ is of $(Q(n), \psi(n))$-type, then there is a box S of width $2^{i+1}\epsilon(n)/8$, height at least $N_3 - 1$, and with $\Delta^{\mathcal{P}}(S, S + \tilde{\alpha}(u, v)) \geq \frac{\epsilon(n)}{2^{\tau(n)+3}}$.*

The idea behind the proof is to use the famous Weyl equidistribution theorem, which states that if ζ is irrational, the fractional parts of the sequence $\{j\zeta\}_{j=0}^{K-1}$ are uniformly distributed in $[0, 1]$ in the sense that as $K \to \infty$, each $[a, b] \subset [0, 1]$, gets about the expected number of points from the sequence, i.e. a $b - a$ fraction. The rate of convergence to the uniform distribution depends on the extent to which ζ is approximable by rational numbers. Our assumption on $\tilde{\alpha}(u, v)$ implies, through a quantitative version of the Weyl theorem, that $\{j\tilde{\alpha}(u, v)\}_{j=0}^{2^{\tau(n)}-1}$ is nicely distributed modulo 2^{i+1}, see Theorem 6.20. To prove Lemma 6.17, let us start by defining a set of boxes to study.

Definition 6.18. Let $w(n) = 2^{i+1}\epsilon(n)/8$, $m(n) = \left\lfloor 2^{i+1}/w(n) \right\rfloor$, $h(n) = \pi_Y(\tilde{\alpha}(u, v))$ and let $S_{0,0}$ be the box $[0..w(n) - 1] \times [0..h(n) - 2]$. Define

$$S_{j,k} = S_{0,0} + jw(n) + k\tilde{\alpha}(u, v)$$

for $0 \leq j \leq m(n) - 1$ and $0 \leq k \leq 2^{\tau(n)} - 2$. A box is said to be *split* if it intersects both $S^{(0)}$ and $S^{(1)}$. Define the *orbit* o_j by

$$o_j = \bigcup_k S_{j,k},$$

where the union is taken only over non-split boxes.

Figure 4 shows the boxes $S_{j,k}$ in a picture.

We establish the basic properties of our set of boxes.

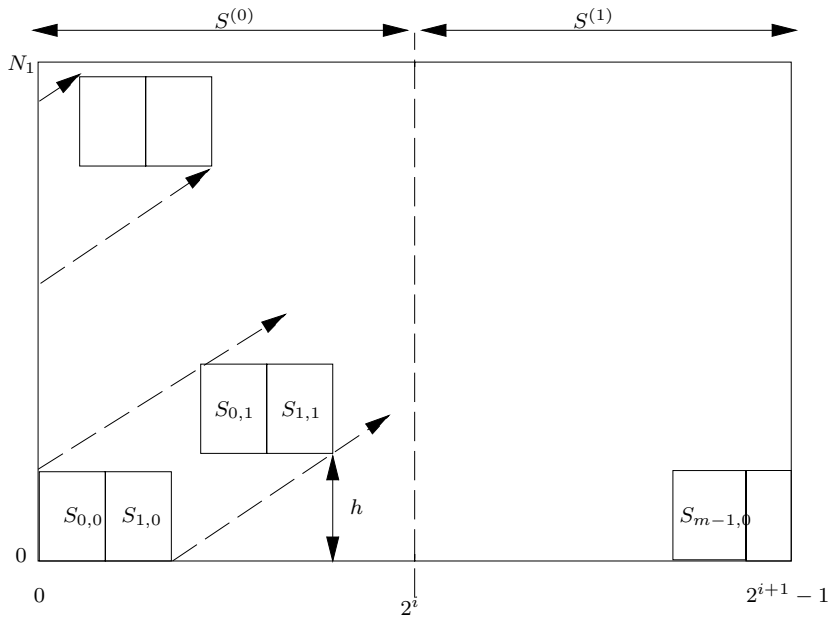

Fig. 4. The basic boxes.

LEMMA 6.19. *The boxes* $\{S_{j,k}\}$ *are pairwise disjoint and cover* $\Pi(N,i)$ *except for at most an* $\epsilon(n)/2$*-fraction. The total measure of the split boxes is at most* $\epsilon(n)/4$.

PROOF. First of all, notice that since the largest point in any box is

$$w(n)-1+(h(n)-2)2^{i+1}+(\lfloor 2^{i+1}/w(n)\rfloor-1)w(n)+(2^{\tau(n)}-2)\tilde{\alpha}(u,v)\leq$$
$$(h(n)-1)2^{i+1}+(2^{\tau(n)}-2)(2^i+\frac{N}{2^{\tau(n)}})<N$$

we need not perform any reductions mod N when studying the boxes $S_{j,k}$. The boxes are disjoint since boxes with different k-values have disjoint projections on the Y-axis and boxes with the same k-value and different j-values have disjoint projections on the I-axis. The total size of all the boxes is

$$(h(n)-1)w(n)m(n)(2^{\tau(n)}-1)\geq(2^{i+1}-w(n))(1-2^{1-\tau(n)})N_1\geq(1-\epsilon(n)/4)N$$

and thus they cover all but an $\epsilon(n)/4$ fraction of the plane. Finally note that for each k at most two $S_{j,k}$ are split and thus we have at most $2^{1+\tau(n)}$ split boxes and the total size of these split boxes is bounded by $2^{1+\tau(n)}(h(n)-1)w(n)\leq\epsilon(n)N/4$. □

As another preliminary consider the below theorem, the proof of which we postpone to the appendix.

THEOREM 6.20. *Let* $0\leq v\leq 2^{\tau(n)}-1$, v *odd,* $|u|\leq 2^{\tau(n)}-1$. *If* $\tilde{\alpha}(u,v)/2^{i+1}\in\mathbb{Q}$ *is of* $(Q(n),\psi(n))$*-type, then for all* $0\leq a<b<2^{i+1}$,

$$\left|\Pr_j[a\leq[j\tilde{\alpha}(u,v)]_{2^{i+1}}\leq b]-\frac{b-a}{2^{i+1}}\right|\leq 14\left(\frac{1}{Q(n)}+\frac{4\psi(n)\log^2 Q(n)}{2^{\tau(n)}}\right),$$

the probability taken over j, chosen uniformly at random in $\{0, 1, \ldots, 2^{\tau(n)} - 2\}$.

Let us now turn to the proof of Lemma 6.17.

PROOF PROOF OF LEMMA 6.17. In view of Lemma 6.19, $\mathcal{P}$ must have advantage $\epsilon(n)/4$ of determining the ith bit on o_{j_0} for some j_0. Each individual box that is part of o_{j_0} is not split and hence it is either contained completely in $S^{(0)}$ or completely in $S^{(1)}$. Define $o_{j_0,k} = o_{j_0} \cap S^{(k)}$ and assume that $o_{j_0,k}$ contains n_k boxes. Since being contained in $S^{(0)}$ is equivalent to the lower left hand corner being in an interval of length $2^i - w(n)$ modulo 2^{i+1}, and the same is true for being contained in $S^{(1)}$, two applications of Theorem 6.20 yield

$$|n_1 - n_0| \leq 28(2^{\tau(n)} - 1)\left(\frac{1}{Q(n)} + \frac{4\psi(n)\log^2 Q(n)}{2^{\tau(n)}}\right) \leq 2^{\tau(n)}\epsilon(n)/16 \qquad (6.2)$$

and an additional application (using very blunt estimates) of the same theorem yields

$$n_1 + n_0 \geq 2^{\tau(n)}/2 \qquad (6.3)$$

Assume for concreteness that $n_1 \geq n_0$. Now pair each box in $o_{j_0,0}$ in some arbitrary way with a unique box in $o_{j_0,1}$. By (6.2) at most a fraction $\epsilon(n)/16$ of the boxes remain single. Thus by the assumption on the predictor there must be ℓ_k, $k = 0, 1$ such that $S_{j_0,\ell_k} \in o_{j_0,k}$ and such that $\mathcal{P}$ has advantage at least $\epsilon(n)/8$ over $S_{j_0,\ell_0} \cup S_{j_0,\ell_1}$. Now, since $S_{j_0,\ell_k} \subset S^{(k)}$ we can conclude that

$$\Delta^{\mathcal{P}}(S_{j_0,\ell_0}, S_{j_0,\ell_1}) \geq \epsilon(n)/8.$$

The lemma now follows by the triangle inequality. □

We now address the case when we do have very good rational approximations of $\tilde{\alpha}(u, v)/2^{i+1}$. The analysis is divided into two cases depending on whether the denominator of this strong rational approximation is odd or even.

LEMMA 6.21. *Suppose v is odd and that there are relatively prime integers r, s, $0 < s \leq Q(n)$ and s even, so that*

$$\left|\frac{\tilde{\alpha}(u, v)}{2^{i+1}} - \frac{r}{s}\right| \leq \frac{1}{s^2\psi(n)}, \qquad (6.4)$$

then there is a $k \leq s$ such that

$$\Delta^{\mathcal{P}}(S^{(0)} + k\tilde{\alpha}(u, v), S^{(0)} + (k+1)\tilde{\alpha}(u, v)) \geq \frac{\epsilon(n)}{2s}.$$

PROOF. Set $s = 2s'$ and consider $s'\tilde{\alpha}(u, v)$. By the assumption on $\tilde{\alpha}(u, v)$ and using that r is odd we have

$$|\pi_I(s'\tilde{\alpha}(u, v)) - 2^i| \leq \frac{2^{i+1}}{s\psi(n)}.$$

Furthermore $|s'\tilde{\alpha}(u, v)| \leq Q(n)N2^{-\tau(n)}$. This implies that

$$\lambda\left(\left(S^{(0)} + s'\tilde{\alpha}(u, v)\right) \triangledown S^{(1)}\right) \leq \frac{2Q(n)}{2^{\tau(n)}} + \frac{2}{s\psi(n)},$$

the error terms coming from "drift" in the Y- and I-directions of the plane. By the choice of $Q(n)$ and $\tau(n)$ this latter quantity is bounded from above by $\epsilon(n)/3$. Now,

$$\Delta^{\mathcal{P}}(S^{(0)}, S^{(1)}) \geq \epsilon(n) - \beta_i(N)$$

where $\beta_i(N)$ is the bias of the ith bit. Since this is, by the assumption on i, smaller than $\epsilon(n)/6$ we conclude that

$$\Delta^{\mathcal{P}}\left(\left(S^{(0)} + s'\tilde{\alpha}(u,v)\right) \bigtriangledown S^{(0)}\right) \geq \epsilon(n)/3.$$

The existence of k now follows by the triangle inequality. □

In the case of a good approximation with an odd denominator we cannot prove that there exists a good box and as indicated by the example given in Section 4.1 there might be no such boxes. We show however, that in this case we can find a factor in N or we can find a related predictor which distinguishes intervals at distance $(N+1)/2$.

LEMMA 6.22. *Suppose there are integers u, v, r, s, $0 < v \leq 2^{\tau(n)} - 1$, v odd, $|u| \leq 2^{\tau(n)} - 1$, $0 < s \leq Q(n)$, $(r,s) = 1$ and s odd, such that*

$$\left|\frac{\tilde{\alpha}(u,v)}{2^{i+1}} - \frac{r}{s}\right| \leq \frac{1}{s^2\psi(n)}, \tag{6.5}$$

and for all boxes S of height at least $sN_1 2^{-\tau(n)}$ and width at least $2^{i+1}\epsilon(n)/(30s)$, we have that $\Delta^{\mathcal{P}}(S, S + \tilde{\alpha}(u,v)) \leq \epsilon(n)2^{-(\tau(n)+3)}$. Then, unless we find a factor in N using $\mathcal{P}$, we can in random polynomial time construct a predictor $\mathcal{P}'$ and find an interval J of length at least $N\epsilon(n)/32$ such that $\Delta^{\mathcal{P}'}(J, J + (N+1)/2) \geq \epsilon(n)/8$.

Before we prove this final lemma let us finish the proof of the main theorem.

PROOF PROOF OF THEOREM 4.2. If the hypothesis of Lemma 6.22 is true we can use the constructed $\mathcal{P}'$ together with Lemma 6.2.

If the hypothesis of Lemma 6.22 is false then Lemma 6.12, Lemma 6.14, Lemma 6.17, and Lemma 6.21 establishes the existence of all boxes needed to apply Lemma 6.10. □

Thus all that remain is to prove Lemma 6.22. To see how the proof will go, we remind the reader of the work by Ben-Or et al. in [Ben-Or et al. 1983]. Recall that we write $N = N_1 2^{i+1} + N_0$, $N_1 = N_3 2^{\tau(n)} + N_2$. Ben-Or et al. showed that if $\mathcal{P}$ is an $\epsilon(n)$-CMHNP for the ith bit and we define a new predictor, $\mathcal{P}_2$, by

$$\mathcal{P}_2(a) = \mathcal{P}([N_1^{-1}a]_N)), \tag{6.6}$$

then $\mathcal{P}_2(a)$ distinguishes between some sets $J, J + (N+1)/2$, increasing the error probability of $\mathcal{P}$ by a quantity depending on $[N]_{2^{i+1}}$ and this quantity in turn is $\frac{1}{4}$ in the worst case. The reason that this works is that 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.

The assumptions of Lemma 6.22 enables us to find another transformation (similar to (6.6)) of the original predictor that maps certain sets at distance 2^i to sets also almost at distance $(N+1)/2$ and where the predictor has a significant advantage. We start by a preliminary lemma.

LEMMA 6.23. *If there are integers u, v, r, s, $0 < v \leq 2^{\tau(n)} - 1$, v odd, $|u| \leq 2^{\tau(n)} - 1$, $0 < s \leq Q(n)$, $(r, s) = 1$ and s odd, such that $\left|\frac{\tilde{\alpha}(u,v)}{2^{i+1}} - \frac{r}{s}\right| \leq \frac{1}{s^2\psi(n)}$, then there is $r' \in \mathbb{Z}$, $r' \leq 2Q(n)$ so that for all sufficiently large n,*

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

PROOF. Set $r' = r - sN_3$. Unfolding the definition of $\tilde{\alpha}(u, v)$, for some $\delta \leq 2^{\tau(n)}$ (so that $2^{\tau(n)}|(N + \delta)$) we have

$$\begin{aligned}\left|\frac{\tilde{\alpha}(u,v)}{2^{i+1}} - \frac{r}{s}\right| &= \left|\frac{u'2^{i+1} + N_3 2^{i+1+\tau(n)} + N_2 2^{i+1} + N_0 + \delta}{2^{i+1+\tau(n)}} - \frac{r}{s}\right| \\ &= \left|N_3 + \frac{u'2^{i+1} + N_2 2^{i+1} + N_0 + \delta}{2^{i+1+\tau(n)}} - \frac{r'}{s} - N_3\right| \\ &= \left|\frac{(u' + N_2)2^{i+1} + N_0 + \delta}{2^{i+1+\tau(n)}} - \frac{r'}{s}\right|.\end{aligned}$$

Multiplying by $2^{\tau(n)}s$ and using the assumption we get:

$$\left|s(u' + N_2) + \frac{s(N_0 + \delta)}{2^{i+1}} - 2^{\tau(n)}r'\right| \leq 2^{\tau(n)}\frac{1}{s\psi(n)}.$$

But, in fact, we must also have $N_0 + \delta \leq 2^{i+1}$, so

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

Using $s \leq Q(n)$, $u' < 2^{\tau(n)}$, $N_2 < 2^{\tau(n)}$ and substituting the definition of $Q(n), \psi(n)$, and $\tau(n)$ now establishes the results. □

The integer $s(u'+N_2)-2^{\tau(n)}r'$ plays a special role in our argument and we introduce the symbol κ for it.

Definition 6.24. Define the integer

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

In the remainder of this section we now concentrate on r', s, u', κ as above. We can at this point write down the predictor that (under our assumption on $\mathcal{P}$) distinguishes between some J and $J + (N + 1)/2$.

Definition 6.25. 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 predictor

$$\mathcal{P}'(a) \triangleq \mathcal{P}(\varphi^{-1}(a)).$$

We see that when $s = 1, \kappa = 0$, we get precisely the same predictor construction as in [Ben-Or et al. 1983].

It may be the case that φ^{-1} does not exist, i.e. that $sN_1 - \kappa$ does not have a multiplicative inverse. Since $0 < sN_1 - \kappa < N$ we have in such a case found a

nontrivial factor of N and by the statement in Lemma 6.22 we can then terminate the algorithm. Hence, we may assume that φ^{-1} exists.

It is now convenient for us to work with a slightly different subdivision of the plane.

Definition 6.26. Let

$$w'(n) \triangleq \left\lfloor 2^{i+1}\left(\frac{1}{2s} - \frac{1}{s\psi(n)}\right)\right\rfloor$$

and $w(n) \triangleq \lfloor w'(n)\epsilon(n)/10\rfloor$. Define the base box

$$S_{0,0} \triangleq \{0,\ldots,w(n)-1\} \times \{0,\ldots,\pi_Y(s\tilde{\alpha}(u,v))-1\}$$

and then translated boxes

$$S_{j,k} \triangleq S_{0,0} + k\tilde{\alpha}(u,v) + jw(n), \qquad 0 < k < 2^{\tau(n)} - s, \quad 0 \le j < \lfloor w'(n)/w(n)\rfloor.$$

Also, define the orbit

$$o_j \triangleq \bigcup_k S_{j,k}.$$

For each $S_{j,k}, o_j$ we define $S'_{j,k} \triangleq S_{j,k} + 2^i$, $o'_j \triangleq o_j + 2^i$.

As before, we call a box S *split* if both $S \cap S^{(0)}$, and $S \cap S^{(1)}$ are non-empty.

The proof will now proceed as follows. By assumption, $\mathcal{P}$ behaves almost the same on all boxes within any fixed orbit, o_j. We will shortly see (in Lemmas 6.28 and 6.29), that under the mapping $\varphi(\cdot)$, o_j gets mapped into what is (almost) an interval J_j, and that o'_j (almost) maps to $J_j + (N+1)/2$. We prove that if $\mathcal{P}$ has a significant advantage in guessing the ith bit on at least a single pair of boxes $S_{j,k} \cup S'_{j,k}$ for some j,k (as it by assumption should), then $\mathcal{P}'$ non-negligibly distinguishes J_j from $J_j + (N+1)/2$. We establish that the boxes cover most of the plane and hence there must be such a j and this completes the argument. We start by investigating how well the boxes $S_{j,k}$ and $S'_{j,k}$ cover the $\Pi(N,i)$-plane.

Lemma 6.27. *The collection of boxes given by all $S_{j,k}$, and $S'_{j,k}$ for $0 \le j < \lfloor w'(n)/w(n)\rfloor$ and $0 \le k < 2^{\tau(n)} - s$ are disjoint and cover the plane except for a fraction at most $\epsilon(n)/4$. The total measure of all split boxes is at most $\epsilon(n)/10$.*

Proof. First we claim that no modular reductions are needed in the definition of the boxes. This follows since the maximal value of any element in any of the boxes is bounded by

$$(2^{\tau(n)} - (s+1))\tilde{\alpha}(u,v) + s\tilde{\alpha}(u,v) + 2^{i+1} = (2^{\tau(n)} - 1)\tilde{\alpha}(u,v) + 2^{i+1} < N,$$

for the range of i considered.

Next note that $S_{j,k}$ are disjoint for different j and a fixed value of k and thus we can study the "superboxes"

$$B_k \triangleq \bigcup_j S_{j,k}$$

together with their similarly defined counterparts B'_k at distance 2^i. The width of such a superbox is bounded by $w'(n)$. By symmetry and translation we need

only prove that for any k, neither B_k nor B'_k intersect B_0. Since $w'(n) < 2^i$, B'_0 clearly does not intersect B_0, and by studying Y-coordinates it follows that we need only consider $0 < k < s$. Now, the lower left corner of B_k and B'_k has I-coordinates $\pi_I(k\tilde{\alpha}(u,v))$ and $\pi_I(k\tilde{\alpha}(u,v)) + 2^i$, respectively. For a box to intersect with B_0 this coordinate should be at least $2^{i+1} - w'(n)$. By (6.5) on page 25, setting $\ell = kr$ modulo s, we see that $k\tilde{\alpha}(u,v)$ modulo 2^{i+1} is within distance at most $2^{i+1}(s\psi(n))^{-1}$ of $\ell 2^{i+1}/s$. Since ℓ is not 0, this number attains its maximal value when $\ell = s-1$. To have an intersection of B_k with B_0 we would need

$$\frac{s-1}{s}2^{i+1} + 2^{i+1}\frac{1}{s\psi(n)} \geq 2^{i+1} - w'(n)$$

but

$$2^{i+1}\frac{1}{s\psi(n)} + w'(n) < \frac{2^{i+1}}{2s} \tag{6.7}$$

and thus we can have no intersection. The largest possible value of the lower left corner of B'_k is obtained when when $\ell = (s-1)/2$ and in this case the condition of intersection is

$$\frac{2s-1}{2s}2^{i+1} + 2^{i+1}\frac{1}{s\psi(n)} \geq 2^{i+1} - w'(n),$$

which again is false by (6.7). Thus the boxes are disjoint.

The size of each $S_{j,k}$ is $w(n)\pi_Y(s\tilde{\alpha}(u,v))$ and the number of boxes of each of the two types is at least $(2^{\tau(n)} - s)(w'(n)/w(n) - 1)$. Thus the total size of all the boxes is

$$\begin{aligned} 2(2^{\tau(n)} - s)(w(n)'/w(n) - 1)w(n)\pi_Y(s\tilde{\alpha}(u,v)) &\geq \\ \left\lfloor \frac{N}{2^{\tau(n)+i+1}} \right\rfloor (2^{\tau(n)} - s)2s(w'(n) - w(n)) &\geq \\ \frac{N}{2^{i+1}}(1 - \frac{2s}{2^{\tau(n)}})2^{i+1}(1 - \frac{2}{\psi(n)})(1 - \epsilon(n)/10) &\geq N(1 - \epsilon(n)/4). \end{aligned}$$

Finally let us study the size of the split boxes. Any split box intersects the middle vertical lines (i.e. $\pi_I(z) = 0$ or 2^i) for $\pi_Y(s\tilde{\alpha}(u,v))$ levels. Since there are only $N2^{-(i+1)}$ levels we have at most $2N2^{-(i+1)}/\pi_Y(s\tilde{\alpha}(u,v))$ split boxes. The number of points per box is $w(n)\pi_Y(s\tilde{\alpha}(u,v))$, so the total measure of all split boxes is at most $2w(n)2^{-(i+1)} \leq \epsilon(n)/10$. The proof is complete. □

We proceed by investigating how φ acts on the $\Pi(N,i)$-plane. The key two properties we need is that $\varphi(\tilde{\alpha}(u,v))$ is small and that $\varphi(2^i)$ is close to $N/2$. The former property makes sure that an orbit is (essentially) mapped into an interval and the latter property that o_j and o'_j (at distance 2^i) are mapped to intervals at distance about $N/2$. More formally we state our first lemma.

LEMMA 6.28.

$$|\varphi(\tilde{\alpha}(u,v))| \leq 2^{12}ns\epsilon(n)^{-1}\max(2^{i+1}, N/2^{i+1}).$$

PROOF. We need to estimate $(sN_1 - \kappa)\tilde{\alpha}(u,v)$. Let us for the moment ignore the term $\kappa\tilde{\alpha}(u,v)$ and concentrate on $sN_1\tilde{\alpha}(u,v)$. Since $N_1 2^{i+1}$ is close to N it

is useful to write $s\tilde{\alpha}(u,v)$ on the form $a2^{i+1}+b$ for integers a and b. Introducing $\delta < 2^{\tau(n)}$, so that $N+\delta$ is divisible by $2^{\tau(n)}$, we have

$$\begin{aligned} s\tilde{\alpha}(u,v) &= s\frac{u'2^{i+1}+N+\delta}{2^{\tau(n)}} = s\frac{(u'+N_3 2^{\tau(n)}+N_2)2^{i+1}+N_0+\delta}{2^{\tau(n)}} \\ &= sN_3 2^{i+1} + s\frac{(u'+N_2)2^{i+1}+N_0+\delta}{2^{\tau(n)}} = \{ \text{ by Def. 6.24 } \} = \\ &= sN_3 2^{i+1} + \frac{(\kappa+2^{\tau(n)}r')2^{i+1}+s(N_0+\delta)}{2^{\tau(n)}} \\ &= (sN_3+r')2^{i+1} + \kappa 2^{i+1-\tau(n)} + \frac{s(N_0+\delta)}{2^{\tau(n)}}. \end{aligned} \tag{6.8}$$

Now, $N_1 2^{i+1} \equiv -N_0$ modulo N and hence using (6.8)

$$sN_1\tilde{\alpha}(u,v) \equiv -N_0(sN_3+r') + N_1\kappa 2^{i+1-\tau(n)} + \frac{sN_1(N_0+\delta)}{2^{\tau(n)}} \bmod N.$$

Now by Lemma 6.23, $|r'N_0| \le 2^{i+1}2Q(n) \le 2^{11}\epsilon(n)^{-1}2^{i+1}$ and $|s\delta N_1 2^{-\tau(n)}| \le sN2^{-(i+1)}$. Furthermore

$$sN_1N_0 2^{-\tau(n)} - sN_0N_3 = sN_0N_2 2^{-\tau(n)}$$

and this is of absolute value at most $s2^{i+1}$. Remembering the omitted term $\kappa\tilde{\alpha}(u,v)$ we have

$$N_1\kappa 2^{i+1-\tau(n)} - \kappa\tilde{\alpha}(u,v) = \kappa(N_0+\delta+u'2^{i+1})2^{-\tau(n)}$$

which is of absolute value at most $\kappa 2^{i+2}$. Collecting the error terms, and using Lemma 6.23, the lemma follows. □

It may seem that the error term $\sim \max(2^{i+1}, N/2^{i+1})$ is very large. However, since the plan is to find intervals $J, J+(N+1)/2$ where the predictor behaves differently, the error term should be compared to N and for the range of i currently under consideration our error is relatively small compared to N.

LEMMA 6.29. *For sufficiently large n,*

$$\left|\varphi(2^i) - \frac{N+1}{2}\right| \le 2sn\epsilon(n)^{-1}2^{i+1}$$

and

$$\operatorname{abs}_N\left(\varphi(2^{i+1})\right) \le 4sn\epsilon(n)^{-1}2^{i+1}.$$

PROOF. To study $\varphi(2^i) = [(sN_1-\kappa)2^i]_N$ we first note that by Definition 6.24 and Lemma 6.23, $|\kappa 2^i| \le ns\epsilon(n)^{-1}2^i$ and this will be part of the error term. Since s is odd, writing $s = 2s'+1$ for an integer s' we see that

$$sN_1 2^i = s'N_1 2^{i+1} + N_1 2^i.$$

Now $N_1 2^{i+1} \equiv -N_0$ modulo N and $|s'N_0| \le s2^{i+1}$. Noting that $|N_1 2^i - (N+1)/2| \le 2^i$, we establish the first part of the lemma by collecting the error terms. The second part of the lemma follows immediately from the first. □

The first part of the Lemma says that values that differ in their ith bit gets mapped to values essentially $(N+1)/2$ apart.

We now study how orbits, o_j, o'_j can be mapped into intervals.

LEMMA 6.30. *There is an interval J_j of length at least $N\epsilon(n)/32$ such that*

$$\#\left(J_j \nabla \varphi(o_j)\right) \leq \epsilon(n) w(n) s N_1/16$$

and

$$\#\left(\left(J_j + \frac{N+1}{2}\right) \nabla \varphi(o'_j)\right) \leq \epsilon(n) w(n) s N_1/16.$$

PROOF. Define J_j as $[jsN_1w(n), \ldots, (j+1)sN_1w(n) - 1]$. The length of this interval is

$$\#J_j = sN_1w(n) \geq w'(n)\epsilon(n)sN_1/11 \geq \epsilon(n)2^{i+1}N_1/23 \geq \epsilon(n)N/32.$$

The orbit o_j contains $(2^{\tau(n)} - s)\pi_Y(s\tilde{\alpha}(u,v))w(n)$ points. As a first part to establish the claim we prove that the sizes of the two sets (i.e. J_j and $\varphi(o_j)$) are about equal. To see this, note that $\pi_Y(s\tilde{\alpha}(u,v))$ is within 1 of $sN2^{-(i+1+\tau(n))}$ which in its turn is within 1 of $sN_1 2^{-\tau(n)}$. Thus the total number of points in o_j is of the form $(1+\delta(n))sN_1w(n)$ where

$$|\delta(n)| \leq (s+2)2^{-\tau(n)} \leq \epsilon(n)/64.$$

To establish the first part of the lemma we thus just need to prove that at most a fraction $\epsilon(n)/32$ of the points of o_j are mapped outside J_j by φ.

Let us first consider the bottom level of $S_{j,0}$. If it was not for the presence of κ in the definition of φ this bottom level would have been mapped evenly to the entire J_j. However the presence of κ only displaces elements of this bottom level at most $\kappa 2^i$ which is bounded by $|J_j|\epsilon(n)/128$.

Let us next consider the bottom levels of $S_{j,k}$. By Lemma 6.28 these are only shifted a distance at most

$$2^{\tau(n)}2^{12}ns\epsilon(n)^{-1}\max(2^{i+1}, N/2^{i+1})$$

which is again bounded by $|J_j|\epsilon(n)/128$.

Finally let us consider the non-bottom levels. By Lemma 6.29 starting points of adjacent levels get mapped to points at most $4sn\epsilon(n)^{-1}2^{i+1}$ apart. Since we have $sN_1 2^{-\tau(n)}$ levels in one box the top level has been shifted a distance at most $4s^2n\epsilon(n)^{-1}2^{-\tau(n)}N$. This is also, by the choice of $\tau(n)$, bounded by $|J_j|\epsilon(n)/128$. Adding the error terms we get the first part of the lemma.

Note that this part of the argument is the only part that depends on κ being small.

To study the behavior of o'_j, we need only to add the extra error term $2sn\epsilon^{-1}(n)2^{i+1}$ as given by Lemma 6.29 and coming from that fact that 2^i is not mapped exactly to $(N+1)/2$. This small extra term does not disturb the calculations. □

Note that by definition of i and the unterval J_j, at most an $\epsilon(n)/16$-fraction of the points are mapped outside J_j. We get immediately.

COROLLARY 6.31. *If there is a j such that $\Delta^{\mathcal{P}}(o'_j, o_j) \geq \epsilon(n)/4$ then there is an interval J_j, of length at least $\epsilon(n)N/32$ for which the predictor $\mathcal{P}'$ has*

$$\Delta^{\mathcal{P}'}(J_j, J_j + (N+1)/2) \geq \frac{\epsilon(n)}{8}.$$

The last piece in the proof of Lemma 6.22 is given by the following lemma.

LEMMA 6.32. *If $\mathcal{P}$ has advantage $\epsilon(n)$ in deciding the ith bit and for all boxes S of height at least $sN_1 2^{-\tau(n)}$ and width at least $2^{i+1}\epsilon(n)/(30s)$, we have that $\Delta^{\mathcal{P}}(S, S + \tilde{\alpha}(u,v)) \leq \epsilon(n)2^{-(\tau(n)+3)}$, then for some j we have $\Delta^{\mathcal{P}}(o'_j, o_j) \geq \epsilon(n)/4$.*

PROOF. When considering the predictor only on the part of $\mathbb{Z}_N$ covered by non-split boxes of the form $S_{j,k}$ or $S'_{j,k}$ the predictor must, by Lemma 6.27, still have advantage $\epsilon(n)/2$. Since $\mathcal{P}$ must achieve its average somewhere there must be a pair on non-split boxes $(S_{j,k}, S'_{j,k})$ such that $\mathcal{P}$ has advantage at least $\epsilon(n)/2$ in predicting the ith bit on $S_{j,k} \cup S'_{j,k}$. Since the ith bit is constant on both $S_{j,k}$ and $S'_{j,k}$ and different on these two sets we can conclude that $\Delta^{\mathcal{P}}(S_{j,k}, S'_{j,k}) \geq \epsilon(n)/2$. Now, by assumption on $\mathcal{P}$ for any l we have

$$\Delta^{\mathcal{P}}(S_{j,l}, S_{j,k}) \leq |k-l|2^{-(\tau(n)+3)}\epsilon(n) \leq \epsilon(n)/8.$$

This implies that $\Delta^{\mathcal{P}}(o_j, S_{j,k}) \leq \epsilon(n)/8$ and by a similar reasoning $\Delta^{\mathcal{P}}(o'_j, S'_{j,k}) \leq \epsilon(n)/8$. By the triangle inequality we conclude that $\Delta^{\mathcal{P}}(o'_j, o_j) \geq \epsilon(n)/4$. □

We can now draw the final conclusion, proving Lemma 6.22. By Lemma 6.32 we get a pair of orbits on which $\mathcal{P}$ behaves differently. By Corollary 6.31 this gives the desired pairs of intervals. The proof is complete.

6.4 Extensions when a Factor is Found

In this section we prove that if a factor q is found in N by computing $(N, sN_1 - \kappa)$ and $(q, N/q) = 1$ then, in most cases, we can find $[x]_{N/q}$. We have the following theorem.

THEOREM 6.33. *Let $q = (N, sN_1 - \kappa) > 1$ and assume that $(q, N/q) = 1$. If $(x, N) = 1$ then, using an $\epsilon(n)$-CMHNP, we can extract a polynomial number of candidates for $[x]_{N/q}$ in polynomial time.*

PROOF. By the proof of Theorem 4.2 we need only analyze what happens because of the non-invertability of φ. It is still true that for an orbit o_j there is an interval J_j of about the same measure as o_j such that $\varphi(o_j)$ is contained in J_j. It is, however, the case that only a fraction $1/q$ of the points of J_j are in the image of φ and each has about q preimages. If we make sure that the inputs to the predictor is one of the points in the target space we can still apply the argument.

We use our techniques to compute $[qx]_{N/q}$ and we make sure that all numbers which are feed to the predictor are multiples of q.

Let t be the multiplicative inverse of $(sN_1 - \kappa)$ modulo N/q and for z which is divisible by q define the pseudo-inverse of φ as a random preimage of z, in other words

$$\varphi^{-1}(z) \triangleq tz + rN/q$$

where we choose r uniformly at random in $\mathbb{Z}_q$ each time we compute $\varphi^{-1}(z)$.

The advantage of distinguishing a random multiple of q in the interval J_j and a random multiple of q in the interval $J_j + (N+1)/2$ can now be seen to be close to the predictors ability to distinguish a random number from o_j and o'_j and we can apply the old argument.

There are a few minor details to take care of. We can make sure that all the points r_j are also multiples of q and hence all queried points are of the correct form. We also need to be able to add rN/q for a random r to the points we query. Since $(x, N) = 1$, $rN/q = r'xN/q$ for a related and random r' and thus we can add this term in by choosing a random r'. □

7. SIMULTANEOUS SECURITY OF BITS AND SECURITY OF HIGH ORDER BITS

We would like to extend Theorem 4.2 to all bits but the easiest way to deal with the most significant bits is to reduce this case to the simultaneous security of the least significant (non-biased) bits and hence we address this case first. In fact, for this purpose, the simultaneous security of the $O(\log n)$ least significant bits, which follows from [Alexi et al. 1988], would suffice.

7.1 Simultaneous Security of Non-leftmost Bits

To establish simultaneous security of sets of non-biased bits we consider the next bit test, [Yao 1982], and assume that we have a predictor for $\mathrm{bit}_i([ax]_N)$, given $\mathrm{B}_{i-j}^{i-1}([ax]_N)$ for some $j \in O(\log n)$. We want to use our previous argument but a problem is that when trimming the lists as in Lemma 6.8, $\mathrm{B}_{i-j}^{i-1}([r'_k x]_N)$ of the sample point depends on which of the two alternatives of the unknown bits that is correct and hence these cannot be supplied to the predictor without problems. However, only slight modifications are needed.

For any $\beta \in \{0,1\}^j$ consider what happens if we supply these fixed bits as $\mathrm{B}_{i-j}^{i-1}([r'_k x]_N)$, regardless of whether they are correct or not. The fact that this is incorrect most of the time does not prevent us from analyzing the situation. Call the corresponding predictor $\mathcal{P}^\beta$. Fix u and v as in the previous argument.

If any of the predictors $\{\mathcal{P}^\beta\}$ can be used to extract x we are done so let us assume that this is not the case and hence for some u and v there are no boxes, $S, S+\alpha(u,v)$, as needed by Lemma 6.8. This implies that for any β and for any box S of non-negligible size, $\mathcal{P}^\beta$ outputs (almost) the same fraction of 1-answers on S and $S + \tilde{\alpha}(u,v)$ (and $S + \alpha(u,v)$). We want to establish that the original predictor $\mathcal{P}$ then does not have a significant advantage even when it is fed the correct value of the bits $\mathrm{B}_{i-j}^{i-1}([r'_k x]_N)$. As before, we have a number of cases to consider.

When v is even, $2^{\tau(n)-1}\alpha(u,v)$ is a small odd multiple of 2^i and the bits $\mathrm{B}_{i-j}^{i-1}(\cdot)$ are, with high probability, the same for z and $z + 2^{\tau(n)-1}\alpha(u,v)$ and we have no problems in supplying bits that are correct for both the guesses we want to distinguish.

If v is odd and $\tilde{\alpha}(u,v)2^{-(i+1)}$ is not close to a rational number with small denominator we argue as follows. Consider an orbit as in the proof of Lemma 6.17. Call a box split if there are two points within the box with different values for $\mathrm{B}_{i-j}^{i-1}(\cdot)$. By making the size of the box sufficiently small (but still of non-negligible size) we can again make sure that at most a fraction $\epsilon(n)/10$ of the boxes are split. Look at the advantage of $\mathcal{P}^\beta$ in an orbit, over the non-split boxes having β as the correct

value for $\mathrm{B}_{i-j}^{i-1}([r'_k x]_N)$ of all points in the box. That is, consider the predictor over the set

$$o_j^\beta = \{S_{j,k} \in o_j \mid \forall z \in S_{j,k},\, B_{i-j}^{i-1}(z) = \beta\}.$$

Since the predictor is (almost) constant within the orbit this advantage is essentially given by the difference of the number of such boxes with $\mathrm{bit}_i = 0$ and $\mathrm{bit}_i = 1$. But using Theorem 6.20 with a suitable choice of the parameters Q and ψ shows that this difference is small.

Next, consider the case when v is odd and we have a good rational approximation of $\tilde{\alpha}(u,v)2^{-(i+1)}$ with even denominator $s = 2s'$. The argument is in this case similar with v even, using $s'\tilde{\alpha}(u,v)$ instead of $2^{\tau(n)-1}\alpha(u,v)$.

Finally, let us consider the case when v is odd and we have a good rational approximation of $\tilde{\alpha}(u,v)2^{-(i+1)}$ with odd denominator s. We know that $\mathcal{P}^\beta$ is almost constant on each orbit o_j. Furthermore, if it behaves differently on o_j and o'_j we know how to convert this to a predictor that distinguishes the intervals J and $J + (N+1)/2$ for some J and thus we can assume that each $\mathcal{P}^\beta$ behaves the same on each o_j and o'_j. Now take any box S and assume that $\mathrm{B}_{i-j}^{i-1}(\cdot)$ is constant on this box. Then since each $\mathcal{P}^\beta$ behaves (almost) the same on S and S' this is true also for the β that gives the correct value. Thus the main part of the advantage must come from split boxes but by making the size of the boxes small enough we can make an arbitrarily small fraction of the boxes to be split.

Since we have covered all the cases we get a contradiction to the assumption that $\mathcal{P}$ has an advantage in predicting the ith bit. This concludes the proof sketch to Theorem 4.7. To make this formal is straightforward but tedious and we leave the details to the reader.

7.2 Security of the Most Significant Bits

The extension to the most significant bits does not require any new ideas and hence let us only sketch the argument.

Suppose that the leftmost bits are not (simultaneously) secure. This implies that there is an algorithm that given a predicts the most significant bits of $[ax]_N$ with a better probability of being correct than the trivial algorithm, i.e. only using that $[ax]_N$ is a number in $\mathbb{Z}_N$.

Now, the $c \log n$ most significant bits of $[ax]_N$ is with probability $1 - n^{-d}$ given by the $(c+O(d)) \log n$ least significant bits of $[2^{c+O(d)}ax]_N$ and this correspondence is efficiently computable using a straightforward generalization of Eq. (5.1).

This implies that a good guess of the most significant bits of $[ax]_N$ can be used to get an advantage on guessing the $(c + O(d)) \log n$ least significant bits of $[2^{c+O(d)}ax]_N$. This violates the simultaneous security of the $O(\log n)$ least significant bits already established in [Alexi et al. 1988].

We invite the interested reader to write down a detailed proof.

8. SECURITY OF RSA BITS

The framework applies perfectly to RSA bits when N has two factors. We get

THEOREM 8.1. *If $N = pq$ where p and q are prime, each bit of the RSA function is secure and blocks of $O(\log n)$ bits are simultaneously secure.*

PROOF. We apply Theorem 4.7. We only have to turn a predictor of the bits of x given $E_N(x)$ into a CMHNP. This is straightforward since on input a we compute $E_N(ax) = [a^e E_N(x)]_N$ and feed this to the RSA-predictor. This clearly gives a CMHNP with the same advantage. We can hence apply Theorem 4.7 and we need only note that when we get a factor in N we have the complete factorization and hence we can compute the decryption exponent d and retrieve x "the easy way". □

In the case when N is the product of 3 or more primes then the partial factorization we obtain might not be sufficient to compute the complete factorization of N needed to compute the decryption exponent. However all is not lost. Suppose that $q = (sN_1 - \kappa, N)$ is the obtained factor. By Theorem 6.33, if N is square-free then we can compute $[x]_{N/q}$. If q is a prime then we can compute $[x]_q$ using Fermat's little theorem and hence in this case we can compute all of x. Thus the only case we get into trouble is when q has at least two prime factors, and in this case we do not know how to recover all of x.

9. SECURITY OF RABIN BITS

The Rabin encryption function is defined by $R_N(x) \triangleq [x^2]_N$ where $N = pq$ as before. Many of the earlier results for RSA (e.g. [Vazirani and Vazirani 1984a; Alexi et al. 1988]), carry over to the Rabin function in a straight-forward manner. The main complication to take care of is of basic nature, namely that R_N is not a 1–1 function since there are four roots to each quadratic residue. Hence, given some r, it is not well-defined what the "ith bit of $\sqrt{r}$" should be. One standard way to handle this problem is to demand $p \equiv q \equiv 3 \bmod 4$ (sometimes such N are called Blum-integers) and restrict the domain of R_N to

$$M_N \triangleq \{x \in \mathbb{Z}_N \mid x < N/2 \text{ and } (x/N) = 1\}$$

(where $(\cdot/N)$ denotes the Jacobi-symbol). Then the function

$$R'_N(x) \triangleq \begin{cases} R_N(x), & \text{if } R_N(x) < N/2; \\ N - R_N(x), & \text{otherwise} \end{cases}$$

induces a permutation on M_N.

This approach runs into technical problems in our situation. When searching for boxes in $\Pi(N, i)$, where the predictor behaves differently on $S, S + \alpha(u, v)$, we need that all of these boxes contain a non-negligible fraction of x with $(x/N) = 1$. This leads to difficult number theoretic complications that we wish to avoid. For a short discussion of those we refer to an earlier version of this paper, appearing in [Näslund 1998].

To avoid these problems we propose another way of converting the Rabin function by outputting enough information to make it one-to-one. Let as before $\text{half}_N(x) = 1$ if $x > N/2$ and 0 otherwise. We define

$$R''_N(x) \triangleq R_N(x), (x/N), \text{half}_N(x)$$

i.e we output the Jacobi symbol and the half predicate as well.

We would like to convert a predictor of the ith bit of this function to a CMHNP and the problem is to specify the input that would give us a prediction of the ith bit of ax. The most natural would be if we could calculate $R''_N(ax)$ given a and $R''_N(x)$.

As $R_N(ax) = [a^2 R_N(x)]_N$ and $(ax/N) = (a/N)(x/N)$ the first two components of $R''_N(ax)$ do not pose any problem.

In our extractor, in some cases it turns out to be easy to supply $\text{half}_N([ax]_N)$ for an a under consideration given current information about x while sometimes it is more difficult. We deal with the problem in a very much similar way when we discussed the simultaneous security in Section 7. Let $\mathcal{P}^k$ for $k \in \{0,1\}$ be the predictor that is obtained by having the bit $\text{half}_N([ax]_N)$ hardwired to k for all a queried, regardless of whether this is correct or not. We need to enter the proof to see that we can extract x from one of these predictors or that we can factor. The proof is very similar to the proof of simultaneous security given in Section 7.

As in that proof we can assume that there is a pair (u, v) such that for no box S of non-negligible size does neither $\mathcal{P}^0$, nor $\mathcal{P}^1$, distinguish S and $S + \alpha(u, v)$ (or S and $S + \tilde{\alpha}(u, v)$). This follows since if such an S existed for all (u, v), Lemma 6.8 would enable us to extract x. We have a number of cases to consider.

When v is even, $2^{\tau(n)-1}\alpha(u, v)$ is a small odd multiple of 2^i and thus with high probability z and $z + \alpha 2^{\tau(n)-1}(u, v)$ have the same value of half_N for both the guesses we want to distinguish. Hence we can supply half_N without problems and the old argument applies.

If v is odd and $\tilde{\alpha}(u, v)2^{-(i+1)}$ is not close to a rational number with small denominator we argue as follows. Each orbit of Lemma 6.17 can be partitioned according to the predicate $\text{half}_N(ax)$. Already the part of the orbit in each half is nicely distributed modulo 2^{i+1} and thus the predictor can in this case have no advantage in predicting the ith bit.

Consider the case when v is odd and we have a good rational approximation of $\tilde{\alpha}(u, v)2^{-(i+1)}$ with even denominator $2s'$. The argument is in this case similar to that for even v, using $s'\tilde{\alpha}(u, v)$ instead of $2^{\tau(n)-1}\alpha(u, v)$. We just need to observe that this number is small so we again get a sufficiently good prediction of $\text{half}_N(ax)$ to apply the old argument.

Finally let us consider the case when v is odd and we have a good rational approximation of $\tilde{\alpha}(u, v)2^{-(i+1)}$ with odd denominator s. We know that $\mathcal{P}^k$ is almost constant on each o_j. Furthermore, if it behaves differently on o_j and o'_j we can convert this to a distinguisher for some intervals J and $J + (N + 1)/2$ and thus we can assume that each $\mathcal{P}^k$ behaves the same on each o_j and o'_j. Thus, for any box S and any k, $\mathcal{P}^k$ behaves almost the same on S and S'. Concentrating on the correct value of k we conclude that $\mathcal{P}$ has no overall significant advantage and this contradiction proves the below theorem for individual bits. The extension to simultaneous security works as in the general case.

THEOREM 9.1. *Let N be a product of two primes which are both equal to 3 modulo 4. For each i, given $R''_N(x)$, $\text{bit}_i(x)$ is secure, unless $R''_N(x)$ can be inverted in random polynomial time. Similarly, blocks of $O(\log n)$ bits of x are simultaneously secure.*

Note in fact that inverting $R''_N(x)$ is equivalent to factoring x. We pick a random x compute $R''_N(x)$ and change the value of (x/N) and ask for an inverse image of this point. If that is found and equals y, $(x - y, N)$ gives a nontrivial factor of N.

10. SECURITY OF DISCRETE LOGARITHM BITS

Let $G = \langle g \rangle$ be a cyclic group (written multiplicatively) of order q, $||q|| = n$, and let $f_g(x) = g^x$, be the function composing the group operation with itself x times. For instance, we can consider $[g^x]_p$ for an n-bit prime p where $q = p - 1$ and is g a generator for $\mathbb{Z}_p^*$. Suppose that $q = p'2^k$, where p' is odd. Given $f_g(x)$, the k least significant bits of x are "easy" since they can be found by the Pohlig-Hellman algorithm, [Pohlig and Hellman 1978], and the $O(\log n)$ following bits are secure, see Peralta [Peralta 1986]. Also, the $O(\log n)$ most significant bits are secure; Long and Wigderson [Long and Wigderson 1988]. Finally, Long, [Long 1983], shows that each bit cannot be predicted without errors.

By a reduction from factoring Blum-integers $N = pq$ (and relaxing that g must generate all of $\mathbb{Z}_N^*$) Håstad, Schrift, and Shamir, [Håstad et al. 1993], shows that all bits of x are individually hard with respect to $[g^x]_N$, and $n/2$ bits are simultaneously secure. Patel and Sundaram, [Patel and Sundaram 1998], adopt the techniques from [Håstad et al. 1993] and prove that if $[g^x]_p$ is a one-way function, even if x is restricted to be "small", then almost all the bits of x are (simultaneously) hard. Using another bit-representation than the standard binary, Schnorr [Schnorr 1998], recently proved security for all bits in this representation under similar assumptions.

Hence, despite the large attention given the (general) problem has remained open and we want to apply our general methods.

One problem is that we cannot query the predictor on $f_g(2^{-\tau}x)$ when the group order, q, is even. By the work of Schnorr in [Schnorr 1998], we can however reduce the problem to a subgroup of odd order, p'. Specifically, by the remark above, $z = [x]_{2^k}$ is easily found. The remaining bits of x can then be found as $\lfloor x/2^k \rfloor$, as the discrete logarithm of g^x/g^z, to the base g^{2^k}, and this value is considered modulo p'. Finally, notice that the ith bit of this number is just the $(i+k)$th bit of x. For convenience, in analogy to the previous discussion, write $p' = P_1 2^w + P_0$, $w = i - k$.

A predictor of the ith bit can now be turned into a CMHNP since the appropriate input to the discrete logarithm predictor can be obtained as $g^{ax} = (g^x)^a$.

The main problem that we need to address is the fact that obtaining a factor in p' does not give significant help in the discrete logarithm problem if the factor is large.

Namely, suppose that φ is not invertible so that $(sP_1 - \kappa, p') = d > 1$. When $d \in O(\text{poly}(n))$ we can compute a factor d' of p' that is such that it contains the same prime factors as d and $(d, p'/d') = 1$. Now x modulo d' can be computed in polynomial time even by exhaustive search and using Theorem 6.33 we can compute x modulo p'/d'. By the Chinese remainder theorem we can compute all of x in this case. When d is large this method does not apply and in fact we do not know how to compute all of x in this case. Nevertheless, we prove that this situation is not likely to occur for a random q and position i. Indeed, even considering that "classical" discrete logs impose the restriction that $q + 1$ be a prime, we can still obtain a result qualitatively as strong.

LEMMA 10.1. *Fix $t, w < t$ and let $p' = P_1 2^w + P_0$, be a randomly chosen t-bit*

integer (not necessarily a prime). Then

$$\Pr_{p'}[\exists s, \kappa \leq M \text{ s.t. } (sP_1 - \kappa, p') \geq D] \in O\left(\frac{M^2}{D} + tM^3 \max\left(2^{-w}, 2^{-(t-w)}\right)\right).$$

The proof is postponed to the Appendix.

THEOREM 10.2. *Unless the discrete logarithm problem in $G = \langle g \rangle$ can be solved in random polynomial time, with probability $1 - O(n^{-2})$ over random choices of $\#G = q = p'2^k$, $\|q\| = n$, bits $k, \ldots, n-1$ of x are individually secure for $f_g(x)$. Blocks of $O(\log n)$ bits are simultaneously secure.*

For $G = \mathbb{Z}_p^*$, $||p|| = n$, $p - 1 = p'2^k$, the same result holds with probability $1 - O(n^{-1})$ over random choice of p.

PROOF. Let $i_0(n) \triangleq 5 \log n + 6 \log \epsilon(n)^{-1}$, where $\epsilon(n)$ is the assumed predictor-advantage. Also, define $M \triangleq cn\epsilon(n)^{-2}$ for a constant c, and $D \triangleq n^5 \epsilon(n)^{-4}$.

Choose a random n-bit number q (not necessarily a prime), and let $k \geq 0$ be the highest power of 2, dividing q. Consider a fixed $i \in [k + i_0(n)..n - 1 - i_0(n)]$ (by the results in [Long and Wigderson 1988; Peralta 1986], these are the interesting bits). Write $q = p'2^k$ as above and call q "bad" for this i if $p' = P_1^{(i)} 2^{i+1-k} + P_0^{(i)}$ is bad in the sense of Lemma 10.1, i.e. if there are $s, \kappa \leq M$ (by Lemma 6.23, M as above suffices) such that $(sP_1^{(i)} - \kappa, p') \geq D$. By Lemma 10.1 with $t = n - k$, $w = i + 1 - k$, q is bad with probability $O(M^2 D^{-1} + tM^3 \max(2^{-w}, 2^{-(t-w)}))$, which by the choices above is $O(n^{-3})$. Moreover, there are less than n different bit positions, i, to consider, so the probability that one of them gives a bad split is $O(n^{-2})$.

What does this tell us about the probability that $q = p - 1$ is bad when p is a prime? The worst case is clearly if all bad q's such that $q + 1$ is a prime numbers. By the prime number theorem, the probability that an n-bit integer is a prime is $\Theta(n^{-1})$. Thus,

$$\Pr_p[p \text{ is a bad prime }] \leq \frac{\Pr_{p \in_U \mathbb{Z}_{2^n}}[\exists i \text{ s.t. } p' \text{ is bad for } i]}{\Pr_{p \in_U \mathbb{Z}_{2^n}}[p \text{ is prime }]}.$$

We may thus lose at most an extra factor of n here, so the probability that p is a bad prime is still bounded by $O(n^{-1})$.

Finally if q is not a bad, the results of the previous sections extend to show that all bits are secure. □

11. SECURITY OF $AX + B$ MODULO P

As described in the introduction, the methods utilized in this paper were first discovered when completing the proof of the results claimed in [Näslund 1996]. We here give the proofs for this original application in a slightly stronger form in that they apply to smaller primes. We are interested in the following family of hash functions.

Definition 11.1. Let H_m be the set of functions of the form $h(x) \triangleq ax + b \bmod p$ with the following probability distribution. The number p is a random prime of m bits while a and b are random numbers modulo p.

We need to define a family of hard core predicates.

Definition 11.2. A family B of predicates is hard core for a one-way function f if given $f(x)$ and a description of a random $b \in B$, $b(x)$ cannot be predicted with a non-negligible advantage. The definition extends to functions outputting more than 1 bit by requiring that the output cannot be distinguished from random bits with non-negligible advantage.

Even though we mostly think of functions f that are (almost) one-to-one this definition also makes sense for general f. In the case when f is severely many-to-one almost any predicate is hard core since $f(x)$ does not contain enough information to determine $b(x)$ even given unbounded resources.

THEOREM 11.3. *Let f be any one-way function and consider it on inputs of length n. Let $m = \omega(\log n)$, then for any i, $0 \leq i < m$ and any constant c, $\mathrm{B}_i^{i+c\log n}(H_m)$ form a family of hard core functions for f.*

PROOF. Assume that we have some $\mathcal{P}$ that predicts the ith bit of $ax+b$ modulo p given $f(x)$, a, b and p with non-negligible advantage $\epsilon(n)$. We that want to recover x. Let us first fix an x such that the advantage over random a, b and p is at least $\epsilon(n)/2$. (Such x have density at least $\epsilon(n)/2$ and it suffices to succeed on these.) Let us say that a pair (p, b') is *good* for this x if the advantage of $\mathcal{P}$ for this fixed p when asking for functions of the type $a(x + b')$ for a random a is at least $\epsilon(n)/4$. It is easy to see that at least a fraction $\epsilon(n)/4$ of all (p, b) are good. For such pairs we can apply our main theorem to compute $x + b'$ and hence x modulo p and this proves the result if $m > n$.

However, if $m < n$ we cannot check the result as the n-bit value x is not uniquely determined when $p < 2^n$. This implies that among the polynomial number of different guesses for x modulo p that are given by the polynomially many different choices in the construction of our pairwise independent sample points, the correct x cannot be immediately distinguished. If $m = \Omega(n)$ one can get around this as in [Näslund 1995], but for smaller primes, that method becomes exponential-time, and we need to be more clever.

Specifically, we apply an error correction result for the Chinese remainder theorem. The below powerful result was obtained by Goldreich, Ron, and Sudan [Goldreich et al. 1999]. Later generalizations have been obtained by Boneh [Boneh 2000] and Guruswami, Sahai and Sudan [Guruswami et al. 2000] but the basic result is sufficient for us.

THEOREM 11.4. *Let $p_1 < p_2 < p_3 \ldots < p_s$ be primes, t and k be integers and $(r_j)_{j=1}^s$ be given numbers. Then, provided*

$$t \geq \Omega\left(\sqrt{ks\frac{\log p_s}{\log p_1}}\right),$$

it is possible in polynomial time to output the list of all numbers z such that $0 \leq z \leq \prod_{j=1}^k p_i$ and such that $z \equiv r_j$ modulo p_j for at least t different values of j.

To apply this theorem we proceed as follows. Let ℓ be a parameter to be specified shortly. Take ℓ different and random p_j each with m bits and apply Theorem 4.2 (or the corresponding theorem for the simultaneous security) to get a list of size

$m^{c_1}\epsilon(n)^{-c_2}$ (for some constants c_1 and c_2 implicit in those proofs) of possible candidates for x modulo p_j for each p_j. Now for each j randomly pick one element r_j in the list and input the list of $(p_j)_{j=1}^{\ell}$ and $(r_j)_{j=1}^{\ell}$ to the algorithm implied by Theorem 11.4. For any element z output by that procedure compute $f(z)$ to see whether z is an acceptable answer.

We need to specify the choice of k, ℓ and t. Since x has n bits we have $x \leq 2^n$ and since each p_j is at least 2^{m-1} we can have $k = \lceil \frac{n}{m-1} \rceil$. Let us estimate the number of modular equations satisfied by x. First, the fraction of p_j that are good is at least $\epsilon(n)/4$ and as stated above for each such p_j we have a list of length $m^{c_1}\epsilon(n)^{-c_2}$ such that with probability at least $1/2$ the value of x modulo p_j appears on it. Thus the expected number of modular equations satisfied by x is at least

$$\ell m^{-c_1}\epsilon(n)^{c_2+1}/8.$$

For sufficiently large ℓ with probability at least $1/2$ the actual number is at least half the expected value i.e.

$$\ell m^{-c_1}\epsilon(n)^{c_2+1}/16$$

and this is the value we choose for t. We need to check the condition of the Theorem 11.4 i.e. that

$$t \geq \Omega\left(\sqrt{ks\frac{\log p_s}{\log p_1}}\right),$$

which in our case is translates to

$$\ell m^{-c_1}\epsilon(n)^{c_2+1}/16 \geq \Omega(\sqrt{\ell n/m})$$

or

$$\ell \geq \Omega\left(m^{2c_1-1}n\epsilon(n)^{-2(c_2+1)}\right).$$

This implies that we can choose an ℓ of polynomial size which satisfies this inequality and in this case the procedure runs in polynomial time and recovers x with probability $1/2$. We conclude that when f is a one-way function such a predictor cannot exist and the ith bit is secure. □

12. DISCUSSION AND OPEN PROBLEMS

Although the reduction from RSA inversion to predicting the individual bits is polynomial time, it is still quite complex and it is hard to give practical implications of the results obtained here. It would therefore be of great interest to find, if possible, a simpler proof, leading to tighter relation between bit security and overall security for RSA.

Hence, 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 [Bellare and Rogaway 1995].

For the simultaneous security, it is in general impossible to go beyond $O(\log n)$ bits. For specific functions (e.g. [Håstad et al. 1993]) it has been done, so we ask if it is possible also for RSA.

For the general chosen multiplier hidden number problem we have given examples of predictors that have an almost perfect advantage and still cannot be used to

extract all of x. It would be surprising if one of those predictors would be efficiently implementable e.g. in the case of RSA with many factors in N. Proving this fact however would seem to require a new method in that such a predictor does not seem to enable us to invert RSA.

APPENDIX

A. THE DISCREPANCY OF A RATIONAL SEQUENCE

This section follows closely the ideas behind the proof of Theorem 2.5 in [Kuipers and Niederreiter 1974]. The aim is to prove Theorem 6.20.

Definition A.1. Recall that for $\zeta \in \mathbb{Q}$, $[\zeta]_1$ denotes the fractional part, $\zeta \pmod 1$ and $\langle\zeta\rangle$ is the distance to the closest integer $\langle\zeta\rangle \triangleq \min([\zeta]_1, 1-[\zeta]_1)$. By a *rational sequence* we mean a sequence of the form $\{[j\zeta]_1 \mid 0 \le j \le T-1\}$ where $\zeta \in \mathbb{Q}$, $T \in \mathbb{N}$. We denote such a sequence by $(\zeta)_T$.

For any sequence $W_T = w_1, w_2, \ldots, w_T \subset [0,1]$, the *discrepancy* of W is defined to be

$$\mathfrak{D}(W_T) \triangleq \sup_{0\le a<b<1} \left| \frac{\#(W_T \cap [a,b])}{T} - (b-a) \right|.$$

Our objective is first to prove the following theorem, from which the desired result then easily follows.

THEOREM A.2. *If $\zeta \in \mathbb{Q}$ is of (Q,ψ)-type, then the rational sequence $(\zeta)_T$ satisfies*

$$\mathfrak{D}((\zeta)_T) \le 6\left(\frac{2}{Q} + \frac{8\psi\log^2 Q}{T}\right).$$

In order to do so, we first need a few preliminaries.

THEOREM A.3 ERDŐS-TURÁN. *For any finite set $W_T = \{w_1, w_2, \ldots, w_T\}$ of real numbers and any positive integer m:*

$$\mathfrak{D}(W_T) \le 6\left(\frac{1}{m} + \sum_{h=1}^{m}\frac{1}{h}\left|\frac{1}{T}\sum_{j=1}^{T} e^{2\pi i h w_j}\right|\right).$$

A proof can be found in [Kuipers and Niederreiter 1974].

LEMMA A.4. *If $\zeta \in \mathbb{Q}$ is of (Q,ψ)-type, then for any $m \le Q$:*

$$\mathfrak{D}((\zeta)_T) \le 6\left(\frac{1}{m} + \frac{1}{T}\sum_{h=1}^{m}\frac{1}{h\langle h\zeta\rangle}\right).$$

PROOF. By the Erdős-Turán Theorem,

$$\mathfrak{D}((\zeta)_T) \le 6\left(\frac{1}{m} + \sum_{h=1}^{m}\frac{1}{h}\left|\frac{1}{T}\sum_{j=1}^{T} e^{2\pi i h j\zeta}\right|\right)$$

for any m. Now,

$$\left|\sum_{j=1}^{T} e^{2\pi i h j\zeta}\right| \le \frac{2}{|e^{2\pi i h\zeta}-1|} = \frac{1}{|\sin \pi h\zeta|}$$

since $h\zeta$ is never an integer for $h \leq m \leq Q$. This also implies that $|\sin \pi h\zeta| = \sin \pi\langle h\zeta\rangle$. Finally, note that $\sin \pi x \geq 2x$ for $0 \leq x \leq 1/2$ so that

$$\frac{1}{|\sin \pi h\zeta|} = \frac{1}{\sin \pi\langle h\zeta\rangle} \leq \frac{1}{2\langle h\zeta\rangle}.$$

□

LEMMA A.5. *Suppose $\zeta \in \mathbb{Q}$ is of (Q, ψ)-type and $m \leq Q/2$. Then*

$$\sum_{j=1}^{m} \frac{1}{j\langle j\zeta\rangle} \leq 8\psi \log^2 m.$$

PROOF. Define $s_j = \sum_{k=1}^{j} 1/\langle k\zeta\rangle$, $j = 1, 2, \ldots, m$. Then, by induction, it is easy to see that

$$\sum_{j=1}^{m} \frac{1}{j\langle j\zeta\rangle} = \sum_{j=1}^{m} \frac{s_j}{j(j+1)} + \frac{s_m}{m+1}. \tag{A.1}$$

If $0 \leq r < s \leq j \leq m \leq Q/2$,

$$\langle s\zeta \pm r\zeta\rangle = \langle (s \pm r)\zeta\rangle \geq \frac{1}{(s \pm r)\psi} \geq \frac{1}{2j\psi}$$

and hence

$$|\langle s\zeta\rangle - \langle r\zeta\rangle| \geq \frac{1}{2j\psi}. \tag{A.2}$$

Consider the intervals

$$\left[0, \frac{1}{2j\psi}\right), \left[\frac{1}{2j\psi}, \frac{2}{2j\psi}\right), \ldots, \left[\frac{j}{2j\psi}, \frac{j+1}{2j\psi}\right).$$

Each of these can by $(A.2)$ contain at most one rational of the form $\langle k\zeta\rangle$, $1 \leq k \leq j$, with no such in the first interval. Therefore

$$s_j = \sum_{k=1}^{j} \frac{1}{\langle k\zeta\rangle} \leq \sum_{k=1}^{j} \frac{2j\psi}{k} \leq 4j\psi \log j$$

so that from $(A.1)$,

$$\sum_{j=1}^{m} \frac{1}{j\langle j\zeta\rangle} \leq 4\psi \left(\sum_{j=1}^{m} \frac{\log j}{j} + \log m \right) \leq 8\psi \log^2 m.$$

□

We are now ready to prove Theorem A.2.

PROOF PROOF OF THEOREM A.2. By Lemma A.4 and A.5, setting $m = Q/2$:

$$\mathfrak{D}((\zeta)_T) \leq 6 \left(\frac{1}{m} + \frac{1}{T} \sum_{j=1}^{m} \frac{1}{j\langle j\zeta\rangle} \right) \leq 6 \left(\frac{2}{Q} + \frac{1}{T} 8\psi \log^2(Q/2) \right).$$

□

Finally, we prove Theorem 6.20.

PROOF PROOF OF THEOREM 6.20. Let $\zeta = \tilde{\alpha}(u,v)/2^{i+1}$ and

$$p' = \Pr_{j \in_U \mathbb{Z}_{2^{\tau(n)}}} [a \le [j\tilde{\alpha}(u,v)]_{2^{i+1}} \le b.]$$

Then

$$\begin{aligned} p' &= \Pr_j \left[\frac{a}{2^{i+1}} \le j\zeta \bmod 1 \le \frac{b}{2^{i+1}} \right] \\ &= \frac{\#\left(\{[j\zeta]_1 \mid 0 \le j \le 2^{\tau(n)} - 1\} \cap \left[\frac{a}{2^{i+1}}, \frac{b}{2^{i+1}}\right]\right)}{2^{\tau(n)}} \in \frac{b-a}{2^{i+1}} \pm \mathfrak{D}((\zeta)_{2^{\tau(n)}}). \end{aligned}$$

Since ζ is of $(Q(n), \psi(n))$-type, Theorem A.2 tells us that

$$\mathfrak{D}((\zeta)_{2^{\tau(n)}}) \le 6\left(\frac{2}{Q(n)} + \frac{1}{2^{\tau(n)}} 8\psi(n)\log^2(Q(n))\right).$$

However, we are restricted to picking j in $\{0, \ldots, 2^{\tau(n)} - 2\}$ only. But it is easy to see that by omitting the single value $(2^{\tau(n)}-1)\zeta$, this can only make the discrepancy go up by $2^{-\tau(n)}$ so certainly, if we pick j at random in $\{0, \ldots, 2^{\tau(n)} - 2\}$,

$$\left| \Pr_j [a \le [j\tilde{\alpha}(u,v)]_{2^{i+1}} \le b] - \frac{b-a}{2^{i+1}} \right| \le 7\left(\frac{2}{Q(n)} + \frac{8\psi(n)\log^2(Q(n))}{2^{\tau(n)}}\right).$$

□

B. PROOF OF LEMMA 10.1

PROOF. Say that p' is "bad" if there are $s, \kappa \le M$ such that $(sP_1 - \kappa, p') \ge D$. Clearly, $(sP_1 - \kappa, p') \le sP_1 + |\kappa| < 2M2^{t-w} \triangleq D_1$. Then

$$\begin{aligned} \Pr_{p'}[p' \text{ bad }] &\le \sum_d \sum_{s,\kappa} \Pr_{p'}[(sP_1 - \kappa, p') = d] \\ &= \sum_d \sum_{s,\kappa} \underbrace{\sum_{P_1} \Pr_{P_0}[(sP_1 - \kappa, p') = d \mid P_1] \Pr[P_1]}_{(*)}, \end{aligned} \tag{B.1}$$

where the sums range over $D \le d \le D_1$, $s, \kappa \le M$ and $0 \le P_1 < 2^{t-w}$. Next,

$$\begin{aligned} (*) &= \sum_{P_1 : d | sP_1 - \kappa} \Pr_{P_0}[(sP_1 - \kappa, p') = d \mid P_1 \wedge d | sP_1 - \kappa] \Pr[P_1] \\ &\le \sum_{P_1 : d | sP_1 - \kappa} \left(\frac{1}{d} + 2^{-w}\right) \Pr[P_1] = \left(\frac{1}{d} + 2^{-w}\right) \Pr_{P_1}[d | sP_1 - \kappa]. \end{aligned}$$

Now, $d | sP_1 - \kappa$ if and only if $sP_1 \equiv \kappa \bmod d$, and this equation is solvable (in P_1) if and only if (d, s) divides κ, in which case there are precisely (d, s) solutions to $P_1 \bmod d$, each selected with probability at most $\frac{1}{d} + 2^{-(t-w)}$ for random P_1. Moreover, since $\kappa \le M$, for each fixed d, s, there are at most $M/(d, s)$ different κ

possible, so continuing from (B.1),

$$\begin{aligned}
\Pr_{p'}[p' \text{ bad }] &\le \sum_d \left(\frac{1}{d} + 2^{-w}\right) \sum_s (d,s) \sum_{\kappa:(d,s)|\kappa} \left(\frac{1}{d} + 2^{-(t-w)}\right) \\
&\le \sum_d \left(\frac{1}{d} + 2^{-w}\right) \sum_s \left(\frac{M}{d} + M2^{-(t-w)}\right) \\
&= \sum_d \sum_s \left(\frac{M}{d^2} + \frac{M}{d}2^{-(t-w)} + \frac{M}{d}2^{-w} + M2^{-t}\right) \\
&= M^2 \sum_d \left(\frac{1}{d^2} + \frac{1}{d}(2^{-(t-w)} + 2^{-w}) + 2^{-t}\right),
\end{aligned}$$

and this sum is bounded by $O(M^2(D^{-1} + \log D_1 \max(2^{-w}, 2^{-(t-w)}) + D_1 2^{-t}))$. □

REFERENCES

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

Bellare, M. and Rogaway, P. 1995. Optimal asymmetric encryption. In *Advances in Cryptology—Eurocrypt '94*, A. De Santis, Ed. Lecture Notes in Computer Science, vol. 950. Springer-Verlag, May 9–12 1994, Perugia, Italy, 92–111.

Ben-Or, M., Chor, B., and Shamir, A. 1983. On the cryptographic security of single RSA bits. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*. ACM, Springer-Verlag, Apr. 25–27 1983, Boston, Massachusetts, 421–430.

Beth, T., Cot, N., and Ingemarsson, I., Eds. 1985. *Advances in Cryptology: Proceedings of Eurocrypt '84*. Lecture Notes in Computer Science, vol. 209. Springer-Verlag, Apr. 9–11 1984, Paris, France.

Boneh, D. 2000. Finding smooth integers in short intervals using crt decoding. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*. ACM, ACM Press, May 21–23 1999, Portland, Oregon, 265–272.

Boneh, D. and Venkatesan, R. 1996. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. See Koblitz [1996], 129–142.

Chor, B. and Goldreich, O. 1985. RSA/Rabin least significant bits are $\frac{1}{2} + \frac{1}{\text{poly}(\log n)}$ secure. In *Advances in Cryptology: Proceedings of CRYPTO '84*, G. R. Blakley and D. Chaum, Eds. Lecture Notes in Computer Science, vol. 196. Springer-Verlag, Aug. 19–22 1984, University of California, Santa Barbara, 303–313.

Fischlin, R. and Schnorr, C. P. 1997. Stronger security proofs for RSA and Rabin bits. In *Advances in Cryptology—Eurocrypt '97*, W. Fumy, Ed. Lecture Notes in Computer Science, vol. 1233. Springer-Verlag, May 11–15 1997, Konstanz, Germany, 267–279.

Goldreich, O. 1985. On the number of close-and-equal pairs of bits in a string (with applications on the security of RSA's L.S.B.). See Beth et al. [1985], 127–141.

Goldreich, O., Ron, D., and Sudan, M. 1999. Chinese remaindering with errors. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*. ACM, ACM Press, May 1–4 1999, Atlanta, Georgia, 225–234.

Goldwasser, S. and Micali, S. 1984. Probabilistic encryption. *Journal of Computer and System Sciences 28*, 270–299.

Goldwasser, S., Micali, S., and Tong, P. 1982. Why and how to establish a private code on a public network (Extended abstract). See IEEE [1982], 134–144.

Guruswami, V., Sahai, A., and Sudan, M. 2000. Soft-decision decoding of chinese remainder codes. In *41st Annual Symposium on Foundations of Computer Science*. IEEE, IEEE Computer Society, Nov. 12–14 2000, Redondo Beach, California, 159–168.

Håstad, J. and Näslund, M. 1998. The security of individual RSA bits. In *39th Annual Symposium on Foundations of Computer Science*. IEEE, IEEE Computer Society, Nov. 8–11 1998, Palo Alto, California, 510–519.

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

IEEE 1982. *23rd Annual Symposium on Foundations of Computer Science*. IEEE, Nov. 3–5 1982, Chicago, Illinois.

Kiltz, E. 2001. A useful primitive to prove security of every bit and about hard core predicates and universal hash functions. In *Proceedings of FCT01*. Lecture Notes in Computer Science, vol. 2138. Springer-Verlag, 388–392.

Koblitz, N., Ed. 1996. *Advances in Cryptology—CRYPTO '96*. Lecture Notes in Computer Science, vol. 1109. Springer-Verlag, Aug. 18–22 1996, University of California, Santa Barbara.

Kuipers, L. and Niederreiter, H. 1974. *Uniform Distribution of Sequences*, 1 ed. Pure & Applied Mathematics. John Wiley & Sons.

Li, W.-C. W., Näslund, M., and Shparlinski, I. E. 2002. Hidden number problem with the trace and bit security of xtr and luc. In *Advances in Cryptology—CRYPTO 2002*, M. Yung, Ed. Lecture Notes in Computer Science, vol. 2442. Springer-Verlag, Aug. 18–22 2002, University of California, Santa Barbara, 433–448.

Long, D. L. 1983. *The Security of bits in the discrete logarithm*. Ph.D. thesis, Princeton University.

Long, D. L. and Wigderson, A. 1988. The discrete log hides $O(\log n)$ bits. *SIAM Journal on Computing 17,* 2, 363–372.

Näslund, M. 1995. Universal hash functions & hard core bits. In *Advances in Cryptology—Eurocrypt '95*, L. C. Guillou and J.-J. Quisquater, Eds. Lecture Notes in Computer Science, vol. 921. Springer-Verlag, May 21–25 1995, Saint-Malo, France, 356–366.

Näslund, M. 1996. All bits in $ax + b \bmod p$ are hard. See Koblitz [1996], 114–128.

Näslund, M. 1998. *Bit Extraction, Hard-Core Predicates, and the Bit Security of RSA*. Ph.D. Thesis, Royal Institute of Technology.

Patel, S. and Sundaram, G. S. 1998. An efficient discrete log pseudo random generator. In *Advances in Cryptology—CRYPTO '98*, H. Krawczyk, Ed. Lecture Notes in Computer Science, vol. 1462. Springer-Verlag, Aug. 23–27 1998, University of California, Santa Barbara, 304–317.

Peralta, R. 1986. Simultaneous security of bits in the discrete log. In *Advances in Cryptology—Eurocrypt '85*, F. Pichler, Ed. Lecture Notes in Computer Science, vol. 219. Springer-Verlag, Apr. 1985, Linz, Austria, 62–72.

Pohlig, S. C. and Hellman, M. 1978. An improved algorithm for computing logarithms over GF(p). *IEEE Transactions on Information Theory IT-24,* 1, 106–110.

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

Schnorr, C. P. 1998. Security of almost all discrete log bits. Electronic Colloquium on Computational Complexity, report TR98-033. Available on-line from http://www.eccc.uni-trier.de/eccc/.

Schnorr, C. P. and Alexi, W. 1985. RSA-bits are $0.5+\epsilon$ secure. See Beth et al. [1985], 114–128.

Schrift, A. W. and Shamir, A. 1991. On the universality of the next bit test. In *Advances in Cryptology—CRYPTO '90*, A. J. Menezes and S. A. Vanstone, Eds. Lecture Notes in Computer Science, vol. 537. Springer-Verlag, Aug. 11–15 1990, University of California, Santa Barbara, 394–408.

Vazirani, U. V. and Vazirani, V. V. 1984a. Efficient and secure pseudo-random number generation (Extended abstract). In *25th Annual Symposium on Foundations of Computer Science*. IEEE, IEEE Computer Society, Oct. 24–26 1984, Singer Island, Florida, 458–463.

Vazirani, U. V. and Vazirani, V. V. 1984b. RSA bits are $.732 + \epsilon$ secure. In *Advances in Cryptology: Proceedings of CRYPTO '83*, D. Chaum, Ed. Plenum Press, New York and London, Aug. 22–24 1983, University of California, Santa Barbara, 369–375.

Yao, A. C. 1982. Theory and applications of trapdoor functions (Extended abstract). See IEEE [1982], 80–91.

Redeived Month Year; Revised Month Year; accepted Month Year