ANALYSIS OF RABIN'S IRREDUCIBILITY TEST FOR POLYNOMIALS OVER FINITE FIELDS

DANIEL PANARIO, BORIS PITTEL, BRUCE RICHMOND AND ALFREDO VIOLA

ABSTRACT. We give a precise average-case analysis of Rabin's algorithm for testing the irreducibility of polynomials over finite fields. The main technical contribution of the paper is the study of the probability that a random polynomial of degree n contains an irreducible factor of degree dividing several maximal divisors of the degree n. We then study the expected value and the variance of the number of operations performed by the algorithm. We present an exact analysis when $n = p_1$ and $n = p_1 p_2$ for p_1, p_2 prime numbers, and an asymptotic analysis for the general case. Our method generalizes to other algorithms that deal with similar divisor conditions. In particular, we analyze the average-case number of operations for two variants of Rabin's algorithm, and determine the ordering of prime divisors of n that minimizes the leading factor.

1. INTRODUCTION

Let $\mathbb{F}_q$ be the finite field with q elements, for q a prime power, and let $f \in \mathbb{F}_q[x]$ be an irreducible polynomial of degree n. In this case, the ring of polynomials modulo f, $\mathbb{F}_q[x]/(f)$, is a finite field with q^n elements. The theorem of existence and uniqueness of finite fields ensures that $\mathbb{F}_{q^n} \cong \mathbb{F}_q[x]/(f)$. This isomorphism allows the construction of arithmetic in extensions fields via polynomial operations. We are only required to find irreducible polynomials of any degree n over any finite field $\mathbb{F}_q$.

This paper deals with a probabilistic algorithm for finding irreducible polynomials due to Rabin [23]. The central idea is to use trial and error, i.e., to take polynomials at random and test them for irreducibility. Remarkably enough, this idea was already noted by Galois ([9], p. 119). Let I_n be the number of irreducible polynomials of degree n over a finite field $\mathbb{F}_q$. In Gauss' posthumous book ([13], p. 611-612), see also Mignotte [18] Chapter 6, Section 2), it is proved that $I_n = \frac{1}{n}\sum_{k|n} \mu(k) q^{n/k}$, for μ the Möbius function and in the case of finite prime fields. This identity holds for any prime power q. It follows

([18], p. 238) that

$$q^n - 2q^{n/2} \leq nI_n \leq q^n. \tag{1}$$

Therefore, a fraction very close to $1/n$ of the polynomials of degree n over any finite field $\mathbb{F}_q$ is irreducible. Thus, on average, we find one irreducible polynomial of degree n after about n tries. This justifies the trial and error method.

It remains to choose an irreducibility test. Let $f \in \mathbb{F}_q[x]$ with $\deg f = n$ be a polynomial to be tested for irreducibility. Assume that $n = \prod_{i=1}^{k} p_i^{m_i}$ with $p_1, \ldots, p_k$ the distinct prime divisors of n, and denote $n_i = n/p_i$, for $1 \leq i \leq k$. Rabin's test is based on the following result: f is irreducible if and only if $\gcd(f, x^{q^{n_i}} - x) = 1$ for all $1 \leq i \leq k$, and $x^{q^n} - x \equiv 0 \bmod f$.

Most of the analyses done in algorithms for polynomials over finite fields are based on the worst-case behavior. Very little work has been done in the average-case analysis for these problems, and most of them are done with techniques based on generating functions and asymptotic analysis. This paper is another step towards this direction.

In Section 2, we revisit Rabin's polynomial irreducibility test. The analysis of Rabin's algorithm involves the study of the number of polynomials of degree n that have irreducible factors of degree dividing a maximal divisor of n. In Sections 3 and 4, we give the main technical contributions of this paper. That is, we study the probability that a polynomial contains an irreducible factor of degree dividing some $n_1, \ldots, n_j$, for $1 \leq j \leq k$. We first give an exact expression of this probability when $n = p_1$ or $n = p_1 p_2$ for p_1, p_2 prime numbers. This explicit formula is complicated for general n. Thus, we provide the asymptotic behavior when n tends to infinity. The average-case analysis of Rabin's algorithm follows from these results. We study the expected value and the variance of the number of operations performed by the algorithm. These results are expressed as an asymptotic formula in n, the degree of the polynomial to be tested for irreducibility.

Similar analyses can also be used to study other algorithms with certain divisor conditions. In Section 5, we analyze some variants of Rabin's algorithms found in [10] and [11]. Finally in Section 6 we conclude by comparing several irreducibility test algorithms.

We assume that arithmetic in $\mathbb{F}_q$ is given. The cost measure of an algorithm will be the number of operations in $\mathbb{F}_q$. The algorithms in this paper use basic polynomial operations like products and gcd tests. We distinguish two approaches for the polynomial arithmetic: the "school" method, and the "fast" method based on the Fast Fourier Transform (FFT). Let $M(n) = n \log n \log\log n$ when considering fast methods,

and $M(n) = n^2$ otherwise. The cost of multiplying two polynomials of degree at most n can be taken as $\tau_1 M(n)$, for a constant τ_1 (for fast arithmetic, see [5, 24, 25]). The cost of a gcd between two polynomials of degree at most n can be taken as $\hat{\tau_2}(n)M(n)$, where

$$\hat{\tau_2}(n) = \begin{cases} \tau_2 & \text{classical arithmetic} \\ \tau_2 \log n & \text{fast arithmetic} \end{cases}$$

(for fast arithmetic see [2], §8.9). For a constant τ_3, a division with remainder can be computed with $\tau_3 M(n)$ operations in $\mathbb{F}_q$.

Finally, we need the computation of $h^q \bmod f$ for polynomials h and f of degree at most n. This exponentiation can be done by means of the classical *repeated squaring* method (see [17], p. 461–462). In this case, the number of products needed is $C_q = \lfloor \log_2 q \rfloor + \nu(q) - 1$, with $\nu(q)$ the number of ones in the binary representation of q. Therefore, the cost of computing $h^q \bmod f$ by this method is $\tau_1 C_q M(n)$ operations in $\mathbb{F}_q$ for both arithmetics. For an excellent reference book in the area we refer to [12].

2. Rabin's irreducibility test

The main goal of this paper is to provide a complete analysis of Rabin's polynomial irreducibility test and several associated variants. In this section, we revisit Rabin's test. Its correctness is based on the following theorem due to Rabin ([23], p. 275, Lemma 1), that leads to an immediate algorithm.

Theorem 2.1. *Let $p_1, \ldots, p_k$ be all the prime divisors of n, and denote $n_i = n/p_i$, for $1 \le i \le k$. A polynomial $f \in \mathbb{F}_q[x]$ of degree n is irreducible in $\mathbb{F}_q[x]$ if and only if gcd $(f, x^{q^{n_i}} - x \bmod f) = 1$ for $1 \le i \le k$, and f divides $x^{q^n} - x$.*

Algorithm: Rabin irreducibility test
Input: A monic polynomial $f \in \mathbb{F}_q[x]$ of degree n, and $p_1, \ldots, p_k$ all the distinct prime divisors of n.
Output: Either "f is irreducible" or "f is reducible".

```
      for i := 1 to k do
          n_i := n/p_i;
      for i := 1 to k do
[*]       g := gcd(f, x^{q^{n_i}} - x mod f);
          if g ≠ 1, then 'f is reducible' and STOP;
      endfor;
      g := x^{q^n} - x mod f;
      if g = 0, then ''f is irreducible''
```

else ``f is reducible''.

A well-known result due to Gauss establishes that, for $i \geq 1$, the polynomial $x^{q^i} - x \in \mathbb{F}_q[x]$ is the product of all monic irreducible polynomials in $\mathbb{F}_q[x]$ whose degree divides i. As a consequence, one would be tempted to infer that f is irreducible if and only if f divides $x^{q^n} - x$. This fact is not true since f could be the product of irreducible polynomials all of them with degree dividing n. This situation justifies the need of computing gcd $(f, x^{q^{n_i}} - x \bmod f)$ for every n_i maximal divisor of n.

The computation of $x^{q^{n_i}} \bmod f$ in Rabin's algorithm is done by repeated squaring independently for each value $n_1, \ldots, n_k$. The corresponding gcd is also taken separate ly. In Section 5, we analyze some variants presented in [10] and [11] on the way of computing these powers.

For later comparison with the average-case result, we now give the worst-case cost of Rabin's algorithm. The cost of Rabin's algorithm is dominated by the cost of computing the exponentiations. It is easy to show using the prime number theorem that in the worst-case the number of operations in $\mathbb{F}_q$ for performing the exponentiations is $\tau_1 C_q n M(n) \log\log\log n$.

3. Exact analysis of two important cases

In practice, it is of interest to have extensions of $\mathbb{F}_2$. Blake *et al.* [4] and Coppersmith [6] show how to compute discrete logarithms fast in $\mathbb{F}_{2^{127}}$. Odlyzko's excellent survey [20] analyzes values of n such that discrete logarithms in $\mathbb{F}_{2^n}$ are probably secure. There was some interest in producing data encryption processor chips for the case $n = 593$ (see [19], p. 69). As we can see, some cases when the degree n of the extension is a prime number p_1 are of practical interest. On the other hand, the case $n = p_1 p_2$ for primes p_1, p_2, has also received some attention. For instance, Agnew *et al.* [1] presented an implementation of elliptic curve cryptosystem over $\mathbb{F}_{2^{155}}$.

In this section we present an exact average-case analysis of Rabin's algorithm, when the degree n of the polynomial being tested for irreducibility is a prime p_1 or the product of two primes $p_1 p_2$. The analysis follows from several propositions of independent interest. These propositions hold for any degree n. However, we present them here for the particular cases $n = p_1$ and $n = p_1 p_2$, for p_1 and p_2 primes.

The proofs on this paper are based on two steps: the symbolic method to establish counting generating functions of interest, and the

extraction of coefficients either directly or via asymptotic analysis. The use of asymptotic techniques is restricted to the Section 4. Next, we briefly introduce the symbolic method for the polynomial expressions of interest here. The reader is pointed out to [8] for an introduction to this methodology.

Let $\mathcal{I}$ be the collection of all monic irreducible polynomials in $\mathbb{F}_q$, and denote by $|\omega|$ the degree of an element $\omega \in \mathcal{I}$. Formally, all monic polynomials with at least one irreducible factor of degree belonging to a set $\mathcal{S}$, and no irreducible factor of degree belonging to a set $\mathcal{T}$ can be written as

$$\prod_{\omega\in\mathcal{I},\ |\omega|\notin\mathcal{S}\cup\mathcal{T}} (1+\omega+\omega^2+\cdots)\left(\prod_{\omega\in\mathcal{I},\ |\omega|\in\mathcal{S}} (1+\omega+\omega^2+\cdots)-1\right)$$
$$= \prod_{\omega\in\mathcal{I},\ |\omega|\notin\mathcal{T}} (1-\omega)^{-1} - \prod_{\omega\in\mathcal{I},\ |\omega|\notin\mathcal{S}\cup\mathcal{T}} (1-\omega)^{-1}.$$

As usual, we consider a formal variable z, and the substitution $\omega \mapsto z^{|\omega|}$. This transformation produces the following generating function:

$$\prod_{\omega\in\mathcal{I},\ |\omega|\notin\mathcal{T}} \left(1-z^{|\omega|}\right)^{-1} - \prod_{\omega\in\mathcal{I},\ |\omega|\notin\mathcal{S}\cup\mathcal{T}} \left(1-z^{|\omega|}\right)^{-1}$$
$$= \prod_{j\notin\mathcal{T}} \left(1-z^j\right)^{-I_j} - \prod_{j\notin\mathcal{S}\cup\mathcal{T}} \left(1-z^j\right)^{-I_j}$$
$$= \frac{\prod_{j\in\mathcal{T}}(1-z^j)^{I_j}}{1-qz} - \frac{\prod_{j\in\mathcal{S}\cup\mathcal{T}}(1-z^j)^{I_j}}{1-qz}, \qquad (2)$$

where we have used the well-known identity (see [8])

$$\frac{1}{1-qz} = \prod_{j\geq 1}\left(\frac{1}{1-z^j}\right)^{I_j}.$$

In a similar way, we obtain the generating function of polynomials with no irreducible factors of degree belonging to a set $\mathcal{T}$:

$$\prod_{\omega\in\mathcal{I},\ |\omega|\notin\mathcal{T}} \left(1-z^{|\omega|}\right)^{-1} = \prod_{j\notin\mathcal{T}} \left(1-z^j\right)^{-I_j} = \frac{\prod_{j\in\mathcal{T}}(1-z^j)^{I_j}}{1-qz}. \qquad (3)$$

3.1. **Basic probabilities.** We first study the situation when $n = p_1$ with p_1 a prime number. In this case, given a polynomial f of degree p_1, the algorithm executes once the gcd [$*$] for $n_1 = 1$. Then, if f does not have an irreducible factor of degree 1, executes its last step, otherwise it terminates. The following easy proposition studies the probability that a random polynomial of degree p_1 does not contain linear irreducible

factors. We give it here for completeness and as an introduction to the methodology we extensively use in this paper. It should be pointed out that Knopfmacher and Knopfmacher ([15], Theorem 1) have studied the probability of a random polynomial having k roots, a result that implies ours.

Proposition 3.1. *Let $P(q, p_1)$ be the probability that a random monic polynomial of degree p_1 over $\mathbb{F}_q$ contains an irreducible factor of degree 1. Then*

$$P(q, p_1) = 1 - \sum_{i=0}^{p_1} \binom{q}{i} \left(\frac{-1}{q}\right)^i.$$

Moreover, when $q \leq p_1$

$$P(q, p_1) = 1 - \left(1 - \frac{1}{q}\right)^q.$$

Proof. We consider Equation (2) when $\mathcal{S} = \{1\}$ and $\mathcal{T} = \emptyset$. Then, since $I_1 = q$, we find after doing the normalization $z \leftarrow z/q$ that the desired probability is

$$P(q, p_1) \quad = \quad [z^{p_1}] \left(\frac{1}{1-z} - \frac{\left(1 - \frac{z}{q}\right)^q}{1-z} \right).$$

The second term is a product of generating functions and thus, a convolution argument gives

$$\begin{aligned} P(q, p_1) &= [z^{p_1}] \left(\frac{1}{1-z} - \sum_{k \geq 0} z^k \sum_{i=0}^{k} \binom{q}{i} \left(\frac{-1}{q}\right)^i \right) \\ &= 1 - \sum_{i=0}^{p_1} \binom{q}{i} \left(\frac{-1}{q}\right)^i. \end{aligned}$$

The second assertion follows immediately from the binomial theorem. ∎

For the most important applications $p_1 \geq q$. In these cases we have

$$0.63212\ldots = 1 - e^{-1} \leq P(q, p_1) \leq 0.75.$$

Thus, as expected, there is a large probability that the polynomial is rejected by the first gcd.

We now consider the case when $n = p_1 p_2$ with p_1, p_2 prime numbers. In this case, given a polynomial f of degree $p_1 p_2$, the algorithm executes the gcd in line [$*$] for $n_1 = p_2$. Then, if f does not have an irreducible factor of degree 1 or p_2, it executes again the gcd for $n_2 = p_1$. Since we

already know that this polynomial does not have an irreducible factor of degree 1, this second gcd only discards polynomials that have an irreducible factor of degree p_1. Finally, if the polynomial does not have an irreducible factor of degree 1, p_1 or p_2, it executes the last step.

The following propositions study the probabilities that a random polynomial of degree p_1p_2 is discarded by the first or second gcd test, respectively.

Proposition 3.2. *Let $P_1(q,n)$ be the probability that a random monic polynomial of degree $n = p_1p_2$ over $\mathbb{F}_q$ contains an irreducible factor of degree 1 or p_2. Then*

$$P_1(q,n) = 1 - \sum_{k=0}^{p_1} \binom{I_{p_2}}{k} \left(\frac{-1}{q^{p_2}}\right)^k \left(\sum_{i=0}^{n-kp_2} \binom{q}{i} \left(\frac{-1}{q}\right)^i \right).$$

Moreover, when $p_2 \geq q$

$$P_1(q,n) = 1 - \binom{I_{p_2}}{p_1} \left(\frac{-1}{q^{p_2}}\right)^{p_1} - \left(1-\frac{1}{q}\right)^q \left(\sum_{k=0}^{p_1-1} \binom{I_{p_2}}{k} \left(\frac{-1}{q^{p_2}}\right)^k \right),$$

and, if in addition $I_{p_2} \leq p_1 - 1$,

$$P_1(q,n) = 1 - \left(1-\frac{1}{q}\right)^q \left(1-\frac{1}{q^{p_2}}\right)^{I_{p_2}}.$$

Proof. The proof is based on Equation (2) with $\mathcal{T} = \emptyset$ and $\mathcal{S} = \{1, p_2\}$. Then, after normalization, we have

$$P_1(q,n) \;=\; [z^{p_1p_2}] \left(\frac{1}{1-z} - \frac{\left(1-\frac{z}{q}\right)^q}{1-z} \left(1-\frac{z^{p_2}}{q^{p_2}}\right)^{I_{p_2}} \right).$$

We study the second generating function as a convolution of $(1 - z/q)^q/(1-z)$ and $(1-(z/q)^{p_2})^{I_{p_2}}$. Thus, we are interested in

$$\begin{aligned}
&[z^{p_1p_2}] \left(\sum_{j\geq 0} z^j \sum_{i=0}^{j} \binom{q}{i} \left(\frac{-1}{q}\right)^i \right) \left(\sum_{k\geq 0} (-1)^k \binom{I_{p_2}}{k} \left(\frac{z^{p_2}}{q^{p_2}}\right)^k \right) \\
&\quad = \; [z^{p_1p_2}] \sum_{j\geq 0} z^j \sum_{k=0}^{\lfloor j/p_2 \rfloor} \binom{I_{p_2}}{k} \left(\frac{-1}{q^{p_2}}\right)^k \left(\sum_{i=0}^{j-kp_2} \binom{q}{i} \left(\frac{-1}{q}\right)^i \right) \\
&\quad = \; \sum_{k=0}^{p_1} \binom{I_{p_2}}{k} \left(\frac{-1}{q^{p_2}}\right)^k \left(\sum_{i=0}^{p_2(p_1-k)} \binom{q}{i} \left(\frac{-1}{q}\right)^i \right).
\end{aligned}$$

When $p_2 \geq q$, the sum indexed by i can be simplified. Indeed, this sum is 1 when $p_1 = k$, and the binomial $(1-1/q)^q$, otherwise. This proves our second assertion, that is,

$$P_1(q,n) = 1 - \binom{I_{p_2}}{p_1}\left(\frac{-1}{q^{p_2}}\right)^{p_1} - \left(1-\frac{1}{q}\right)^q \left(\sum_{k=0}^{p_1-1}\binom{I_{p_2}}{k}\left(\frac{-1}{q^{p_2}}\right)^k\right).$$

Finally, if $I_{p_2} \leq p_1 - 1$ the internal sum and the second term simplify, and we obtain

$$P_1(q,n) = 1 - \left(1-\frac{1}{q}\right)^q \left(1-\frac{1}{q^{p_2}}\right)^{I_{p_2}}.$$

■

In the following, we study the probability that a polynomial is discarded by the second gcd test when it is not discarded by the first gcd test.

Proposition 3.3. *Let $P_2(q,n)$ be the probability that a random monic polynomial of degree $n = p_1 p_2$ over $\mathbb{F}_q$ contains an irreducible factor of degree p_1 but does not contain an irreducible factor of degree 1 or p_2. Then*

$$P_2(q,n) = \sum_{j=1}^{p_2}(-1)^{j+1}\binom{I_{p_1}}{j}\left(\frac{1}{q^{p_1}}\right)^j \sum_{k=0}^{\left\lfloor\frac{n-jp_1}{p_2}\right\rfloor}\binom{I_{p_2}}{k}\left(\frac{-1}{q^{p_2}}\right)^k \left(\sum_{i=0}^{n-jp_1-kp_2}\binom{q}{i}\left(\frac{-1}{q}\right)^i\right).$$

Moreover, when $q \leq p_2 \leq \frac{p_1}{I_{p_2}}$

$$P_2(q,n) = \left(1-\frac{1}{q}\right)^q \left(1-\frac{1}{q^{p_2}}\right)^{I_{p_2}} \left(\sum_{j=1}^{p_2-1}(-1)^{j+1}\binom{I_{p_1}}{j}\left(\frac{1}{q^{p_1}}\right)^j\right) - \binom{I_{p_1}}{p_2}\left(\frac{-1}{q^{p_1}}\right)^{p_2}.$$

Proof. The proof is similar to that of Proposition 3.2 for $\mathcal{T} = \{1, p_2\}$ and $\mathcal{S} = \{p_1\}$. Then, by Equation (2) and after normalization, we have

$$P_1(q,n) = [z^{p_1 p_2}]\left(\frac{\left(1-\frac{z}{q}\right)^q\left(1-\left(\frac{z}{q}\right)^{p_2}\right)^{I_{p_2}}}{1-z} - \frac{\left(1-\frac{z}{q}\right)^q\left(1-\left(\frac{z}{q}\right)^{p_1}\right)^{I_{p_1}}\left(1-\left(\frac{z}{q}\right)^{p_2}\right)^{I_{p_2}}}{1-z}\right).$$

The first generating function was studied in Proposition 3.2. For the second one, we consider the convolution of $(1-z/q)^q\,(1-(z/q)^{p_2})^{I_{p_2}}/(1-$

z) and $(1-(z/q)^{p_1})^{I_{p_1}}$. This convolution gives

$$[z^{p_1p_2}]\left(\frac{\left(1-\frac{z}{q}\right)^q\left(1-\left(\frac{z}{q}\right)^{p_2}\right)^{I_{p_2}}}{1-z}\right)\left(1-\left(\frac{z}{q}\right)^{p_1}\right)^{I_{p_1}} = [z^{p_1p_2}]$$

$$\left(\sum_{s\geq 0} z^s \sum_{k=0}^{\lfloor s/p_2\rfloor}\binom{I_{p_2}}{k}\left(\frac{-1}{q^{p_2}}\right)^k \sum_{i=0}^{s-kp_2}\binom{q}{i}\left(\frac{-1}{q}\right)^i\right)\left(\sum_{j\geq 0}(-1)^j\binom{I_{p_1}}{j}\left(\frac{z^{p_1}}{q^{p_1}}\right)^j\right).$$

The theorem then follows after extracting coefficients and noting that the case $j=0$ cancels with the coefficient given by the generating function in the first term. For the second assertion, we use the binomial theorem as in Proposition 3.2. ■

Table 1 shows the probabilities given by Propositions 3.2 and 3.3 for several values of p_1, p_2 and q. We give the proportion of polynomials rejected in the second step with respect to the total number of polynomials rejected in both steps. It can be seen that most of the polynomials are rejected by the first gcd test, since with probability greater than $1-e^{-1} = 0.63212\ldots$ a random polynomial of degree n has an irreducible factor of degree 1. Moreover, a polynomial is rejected in step 2 only if it has an irreducible factor of degree p_1 but not one of degree 1 or p_2. Thus, the proportion of polynomials being rejected in step 2 is maximized when $p_1 = 2$ and p_2 is large. As it is seen in Table 1, this ratio is upper bounded by $0.1863\ldots$.

The limit estimations of Table 1 are done using the simple asymptotics given by the next proposition.

Proposition 3.4. *Let $n = p_1p_2$.*

(1) *When $p_1 \to \infty$ with p_2, q fixed we have*

$$\begin{aligned} P_1(q,n) &\sim 1-\left(1-\frac{1}{q}\right)^q\left(1-\frac{1}{q^{p_2}}\right)^{I_{p_2}}, \\ P_2(q,n) &\sim \left(1-\frac{1}{q}\right)^q\left(1-\frac{1}{q^{p_2}}\right)^{I_{p_2}}\left(1-e^{-1/p_1}\right). \end{aligned}$$

Moreover, if $q \to \infty$ then

$$\begin{aligned} P_1(q,n) &\sim 1-e^{-1-1/p_2}, \\ P_2(q,n) &\sim e^{-1-1/p_2}\left(1-e^{-1/p_1}\right). \end{aligned}$$

(2) *When $p_2 \to \infty$ with p_1, q fixed we have*

$$\begin{aligned} P_1(q,n) &\sim 1 - \left(1 - \frac{1}{q}\right)^q, \\ P_2(q,n) &\sim \left(1 - \frac{1}{q}\right)^q \left(1 - \left(1 - \frac{1}{q^{p_1}}\right)^{I_{p_1}}\right). \end{aligned}$$

Moreover, if $q \to \infty$ then

$$\begin{aligned} P_1(q,n) &\sim 1 - e^{-1}, \\ P_2(q,n) &\sim e^{-1}\left(1 - e^{-1/p_1}\right). \end{aligned}$$

(3) *When $q \to \infty$ with p_1, p_2 fixed we have*

$$\begin{aligned} P_1(q,n) &\sim 1 - \sum_{k=0}^{p_1} \frac{(-1)^k}{p_2^k k!} \left(\sum_{j=0}^{n-kp_1} \frac{(-1)^j}{j!}\right), \\ P_2(q,n) &\sim \sum_{i=1}^{p_2} \frac{(-1)^{i+1}}{p_1^i i!} \left(\sum_{k=0}^{\lfloor \frac{n-ip_1}{p_2} \rfloor} \frac{(-1)^k}{p_2^k k!}\right) \left(\sum_{j=0}^{n-ip_1-kp_2} \frac{(-1)^j}{j!}\right). \end{aligned}$$

PROOF. (Sketch.) The asymptotic relations (1) and (2) are two special cases covered by the general Proposition 4.1 proved in Section 4. For the case (3), it is enough to extract coefficients from the generating functions

$$\begin{aligned} P_1^{\infty,p_1p_2}(z) &= [z^{p_1p_2}] \left(\frac{1}{1-z} - \frac{e^{-z}e^{-z^{p_2}/p_2}}{1-z}\right), \\ P_2^{\infty,p_1p_2}(z) &= [z^{p_1p_2}] \frac{e^{-z}e^{-z^{p_2}/p_2}}{1-z}\left(1 - e^{-z^{p_1}/p_1}\right). \end{aligned}$$

■

3.2. **Cost of Rabin's algorithm.** To estimate the average-case cost of Rabin's algorithm we need to evaluate both the cost of each step and the probability that the work of the algorithm will require this step. The average-case analysis of Rabin's algorithm is computed combining the cost of each step with the respective probability in Section 3.1. Given the degree of the polynomial $n = \prod_{i=1}^{k} p_i^{m_i}$ with $p_1, \ldots, p_k$ its distinct prime divisors, the cost of executing for the ith time the gcd test in line $[*]$ of the algorithm is

$$C_i(q,n) = (n_i \tau_1 C_q + \hat{\tau}_2(n)) M(n) \tag{4}$$

p_1	p_2	q	rej. both steps	rej. step 2	rej2/rej
2	3	2	.8593750000	.0625000000	.0727272727
2	47	2	.8162630149	.0611702128	.0749393414
2	∞	2	.8125000000	.0625000000	.0769230769
2	3	243	.8333333449	.1450627767	.1740753296
2	47	243	.7813931006	.1409295453	.1803567823
2	∞	243	.7768721241	.1439933111	.1853500810
2	3	∞	.8333333332	.1458333333	.1750000000
2	47	∞	.7813909432	.1416695091	.1813042630
2	∞	∞	.7768698398	.1447492810	.1863237232
5	3	2	.8341484070	.0255546570	.0306356240
5	31	2	.7898071046	.0318712739	.0403532377
5	∞	2	.7933619467	.0433619467	.0546559447
5	3	∞	.7869961636	.0505520833	.0642342182
5	31	∞	.7083666933	.0645684185	.0911511215
5	∞	∞	.6988057881	.0666852292	.0954274140
31	5	2	.7898071046	.0065593786	.0083050387

TABLE 1. Probabilities that a polynomial of degree $n = p_1 p_2$ be rejected.

for $i = 1 \ldots k$, since we have to compute an exponentiation and then a gcd test. The cost of performing the last step is

$$C_{k+1}(q, n) = (n\tau_1 C_q + \tau_3) M(n) \tag{5}$$

since we have to compute an exponentiation and a division with remainder.

The cost moments of Rabin's algorithm can be derived from Equation (4) and (5) and Propositions 3.1, 3.2 and 3.3. Let $E(q, n)$ and $Var(q, n)$ denote the expected value and the variance of the cost of Rabin's algorithm. Consider first the case when $n = p$ is a prime number.

Theorem 3.1. *Let $n = p_1$, a prime number. Then*

$$E(q, n) = \begin{cases} ((1 + s_n n) C_q \tau_1 + \tau_2 + s_n \tau_3) M(n), & \textit{c-arithmetic}, \\ ((1 + s_n n) C_q \tau_1 + \tau_2 \log n + s_n \tau_3) M(n), & \textit{f-arithmetic}; \end{cases}$$

here

$$s_n := \sum_{i=0}^{n} \binom{q}{i} \left(\frac{-1}{q}\right)^i,$$

and c-arithmetic, f-arithmetic stand for classical and fast arithmetic respectively. Furthermore, for both arithmetics,

$$Var(q,n) = s_n(1-s_n)(nC_q\tau_1+\tau_3)^2 M(n)^2.$$

PROOF. If we denote by $C_1(q,n)$ and $C_2(q,n)$ the costs of executing each step of Rabin's algorithm, then we have

$$E(q,n) \;=\; P(q,n)C_1(q,n) + (1-P(q,n))(C_1(q,n)+C_2(q,n)),$$

where $P(q,n)$ is that studied in Proposition 3.1, and $n = p_1$. The result follows after considering $n_1 = 1$ and using formulas (4) and (5).

For the variance, we use the well-known formula $Var[X] = E[X^2] - E[X]^2$, and the previous result. ■

We now study the case when n is a product of two prime numbers.

Theorem 3.2. *Let $n = p_1p_2$ with p_1 and p_2 prime numbers. Then*

$$E(q,n) = \begin{cases} ((p_2 + s_n p_1 + S_n n)C_q\tau_1 \\ \quad + (1+s_n)\tau_2 + S_n\tau_3)M(n), & \textit{c-arithmetic}, \\ ((p_2 + s_n p_1 + S_n n)C_q\tau_1 \\ (1+s_n)\tau_2 \log n + S_n\tau_3)M(n), & \textit{f-arithmetic}; \end{cases}$$

here

$$s_n := 1 - P_1(q,n) = \sum_{k=0}^{p_1} \binom{I_{p_2}}{k} \left(\frac{-1}{q^{p_2}}\right)^k \left(\sum_{i=0}^{n-kp_2} \binom{q}{i} \left(\frac{-1}{q}\right)^i \right),$$

and

$$S_n \;:=\; 1 - P_1(q,n) - P_2(q,n) =$$

$$\sum_{j=0}^{p_2} (-1)^j \binom{I_{p_1}}{j} \left(\frac{1}{q^{p_1}}\right)^j \sum_{k=0}^{\left\lfloor \frac{n-jp_1}{p_2} \right\rfloor} \binom{I_{p_2}}{k} \left(\frac{-1}{q^{p_2}}\right)^k \left(\sum_{i=0}^{n-jp_1-kp_2} \binom{q}{i} \left(\frac{-1}{q}\right)^i \right),$$

and c-arithmetic, f-arithmetic stand for classical and fast arithmetic respectively.

PROOF. If $C_i(q,n)$ stands for the cost of executing an $i-$th step of the algorithm, then we have

$$\begin{aligned} E(q,n) \;&=\; P_1(q,n)C_1(q,n) + P_2(q,n)(C_1(q,n)+C_2(q,n)) \\ &\quad + (1-P_1(q,n)-P_2(q,n))(C_1(q,n)+C_2(q,n)+C_3(q,n)), \end{aligned}$$

with $P_1(q,n)$ and $P_2(q,n)$, $n = p_1p_2$, studied in Propositions 3.2 and 3.3. The result follows after using formulas (4) and (5) with $n_1 = p_2$ and $n_2 = p_1$. ■

Analogously to Theorem 3.1, we have the following.

Theorem 3.3. *Let $n = p_1 p_2$ with p_1 and p_2 prime numbers. Then*

$$Var(q,n) = \begin{cases} (s_n(1-s_n)(p_1\tau_1 C_q + \tau_2)^2 + S_n(1-S_n)(n\tau_1 C_q + \tau_3)^2 \\ \quad + 2S_n(1-s_n)(p_1\tau_1 C_q + \tau_2)(n\tau_1 C_q + \tau_3))M^2(n), & \text{c-ar,} \\ (s_n(1-s_n)(p_1\tau_1 C_q + \tau_2 \log n)^2 + S_n(1-S_n)(n\tau_1 C_q + \tau_3)^2 \\ \quad + 2S_n(1-s_n)(p_1\tau_1 C_q + \tau_2 \log n)(n\tau_1 C_q + \tau_3))M^2(n) & \text{f-ar;} \end{cases}$$

(Here f-ar, c-ar stand for two types of arithmetic, classical and fast.) It is important to notice that the standard deviation is very high. This is clearly seen if we use the asymptotic formulas presented in Proposition 3.4, since in these cases we have for $p_1 \to \infty$

$$\begin{aligned} E(q,n) &\sim s_n(1+p_2)\tau_1 C_q p_1 M(n), \\ Var(q,n) &\sim s_n(1-s_n)(1+p_2)^2\tau_1^2 C_q^2 p_1^2 M^2(n), \end{aligned}$$

where we used that $s_n \sim S_n$. For $p_2 \to \infty$, we have

$$\begin{aligned} E(q,n) &\sim (1+S_n p_1)\tau_1 C_q p_2 M(n), \\ Var(q,n) &\sim S_n(1-S_n)\tau_1^2 C_q^2 n^2 M^2(n). \end{aligned}$$

This high standard deviation is a clear consequence of the fact that even though the second and third steps are executed with very low probability, their costs (especially in the third step) are very high with respect to that of the first step. It can also be seen, as expected, that the important contribution to the cost of Rabin's algorithm is due to the exponentiations performed.

4. Asymptotic analysis of the algorithm

We now turn to the asymptotic study of the cost of Rabin's algorithm for a general case of large n, not subject—unless stipulated otherwise—to any restrictions on its prime factorization. The analysis of Rabin's irreducibility test is done in several stages. Let us denote by "step i", $1 \le i \le k$, the gcd computation in line [$*$] of the algorithm, and "step $k+1$", the final division with remainder. We begin with a sharp asymptotic estimate of the probability that the random polynomial is not rejected before the step i, Proposition 4.1. Then, we give the expected value and the variance for the number of operations performed by Rabin's algorithm. Finally, we present some other asymptotic results.

We fix the notation for the rest of the section. The degree n of the polynomial being tested for irreducibility satisfies $n = \prod_{i=1}^{k} p_i^{m_i}$, with $p_1, \ldots, p_k$ its distinct prime divisors. Given the structure of the algorithm, there is a step for every prime divisor of n.

We denote by $\mathcal{P}_i$ the set of divisors of $n_1, \ldots, n_{i-1}$, where $n_j = n/p_j, j = 1 \ldots, i-1$. In other words, $\mathcal{P}_i$ contains the set of all degrees checked when we start the ith step. We have the initial condition

$$\mathcal{P}_1 = \emptyset.$$

For any $j \geq 1$, we denote by $\mathcal{Q}_j$ the degrees considered on step j that were not considered in the previous steps, that is,

$$\begin{aligned} \mathcal{Q}_j &= \{p_1^{m_1} \cdots p_{j-1}^{m_{j-1}} p_j^{e_j} \cdots p_k^{e_k} : \\ &\qquad 0 \leq e_j \leq m_j - 1;\ 0 \leq e_s \leq m_s,\ j < s \leq k\}. \end{aligned} \tag{6}$$

Then, for $i \geq 2$ we have

$$\mathcal{P}_i = \cup_{j=1}^{i-1} \mathcal{Q}_j.$$

According to the definition (6) of $\mathcal{Q}_j$, the new degrees considered at any step j are relatively larger than those considered in the previous steps, since they must be multiples of $p_1^{m_1} \cdots p_{j-1}^{m_{j-1}}$. In particular, the first step searches for irreducible factors of small degree, most notably linear factors. This will turn out to be an important remark for our analysis. It will imply that a big proportion of the polynomials that pass the first gcd pass all the gcds. In other words, in most of the cases, the first gcd will be the only one that effectively discards polynomials. For instance, we have seen that when $n = p_1 p_2$, the proportion of polynomials rejected during the first step is at least $0.63212\ldots$. We quantify these comments in a precise sense in this section. Other conclusions are drawn in the last section of the paper.

The coming proposition deals with a technical estimation required in the computation of the probability that a polynomial is rejected at certain step of the execution of Rabin's algorithm. Let us denote by $\overline{P}_i(q, n)$ the probability that a random polynomial is not rejected before step i. Using Equation (3) with $\mathcal{T} = \mathcal{P}_i$, we obtain

$$\sum_{n \geq 0} \overline{P}_i(q, n)(qz)^n = \frac{1}{1 - qz} \prod_{\ell \in \mathcal{P}_i} (1 - z^\ell)^{I_\ell}.$$

We observe that $\overline{P}_i(q, n) - \overline{P}_{i+1}(q, n)$ is the probability that a random polynomial of degree n is rejected in step i. In this case, we use Equation (2).

Proposition 4.1. *For $n \to \infty$, uniformly for $q \geq 2$ and for $i \leq k = k(n)$, we have*

$$[z^n] \frac{1}{1 - qz} \prod_{\ell \in \mathcal{P}_i} (1 - z^\ell)^{I_\ell} = q^n \prod_{\ell \in \mathcal{P}_i} \left(1 - q^{-\ell}\right)^{I_\ell} \cdot \left(1 + O(n^{-1+\varepsilon_n})\right),$$

where

$$\varepsilon_n = \frac{\log 2 + a}{\log\log n}, \quad \forall\, a > 0,$$

and the product factor is at least $\log^{-b} n, \forall\, b > e^\gamma$, γ *being the Euler constant* ($\sum_{j=1}^{k} 1/k = \log k + \gamma + o(1)$, $k \to \infty$).

PROOF. Let $\delta = \delta_n \downarrow 0$ and $r = e^\delta/q$. Then the circle $C = \{z : |z| = r\}$ encloses $z = 1/q$, and Cauchy's formula implies

$$\begin{aligned}[z^n]\frac{1}{1-qz}\prod_{\ell\in\mathcal{P}_i}(1-z^\ell)^{I_\ell} &= \mathrm{Res}_{z=1/q}\left(\frac{z^{-n-1}}{qz-1}\prod_{\ell\in\mathcal{P}_i}(1-z^\ell)^{I_\ell}\right)\\ &+ \frac{1}{2\pi i}\int_{|z|=r}\frac{1}{1-qz}\prod_{\ell\in\mathcal{P}_i}(1-z^\ell)^{I_\ell}\frac{dz}{z^{n+1}}.\end{aligned}$$

Here

$$\mathrm{Res}_{z=1/q}\left(\frac{z^{-n-1}}{qz-1}\prod_{\ell\in\mathcal{P}_i}(1-z^\ell)^{I_\ell}\right) = q^n\prod_{\ell\in\mathcal{P}_i}(1-q^{-\ell})^{I_\ell},$$

and we need to show that the contribution of the contour integral is negligible compared to the residue term. First of all, we obtain, uniformly for $|z| \le r$,

$$\prod_{\ell\in\mathcal{P}_i}(1-z^\ell)^{I_\ell} = \exp\left[-\sum_{\ell\in\mathcal{P}_i} I_\ell z^\ell + O\left(\sum_{\ell\in\mathcal{P}_i} I_\ell r^{2\ell}\right)\right],$$

since, for n large, $r^\ell \le (q^{-1}1.5)^\ell \le 0.75$ and

$$|\log(1-\eta) + \eta| \le \mathrm{const}|\eta|^2, \text{ if } |\eta| \le 0.75.$$

(We use the main branch of logarithm here.) Furthermore, using Equation (1), we see that the remainder term in the above exponent is of order

$$\sum_{\ell=0}^{\infty}(qr^2)^\ell = \frac{1}{1-q^{-1}e^{2\delta}} = O(1).$$

Therefore

$$\prod_{\ell\in\mathcal{P}_i}(1-z^\ell)^{I_\ell} = \exp\left(-\sum_{\ell\in\mathcal{P}_i} I_\ell z^\ell + O(1)\right), \tag{7}$$

uniformly for $|z| \le r$. In particular, applying this formula for $z = 1/q$ and using the full power of Equation (1), we see that (for n large) the

residue term equals

$$q^n \prod_{\ell \in \mathcal{P}_i} \left(1 - q^{-\ell}\right)^{I_\ell} = q^n \exp\left(- \sum_{\ell \in \mathcal{P}_i} \frac{1}{\ell} + O(1) \right),$$

uniformly for all i. How large may the last sum be? It is certainly bounded above by

$$D(n) = \sum_{d|n} \frac{1}{d},$$

since each element of $\mathcal{P}_i$ is a divisor of n. It is known (for instance, Theorem 323 in Hardy-Wright [14]) that

$$\limsup_{n \to \infty} \frac{D(n)}{\log \log n} = e^\gamma,$$

where γ is the Euler constant. With this estimate at hand, we see that the residue term is at least of order $q^n \log^{-b} n$, $\forall\, b > e^\gamma$.

Let us turn our attention to the integral along the circular contour $\{z = re^{i\theta} : \theta \in (-\pi, \pi]\}$. Because of the denominator $1 - qz$, we need to consider separately "small" θ and the remaining θ. To determine a separation threshold, we compute

$$\begin{aligned} |1 - qz| &= |1 - e^{\delta + i\theta}| = (1 - 2e^\delta \cos\theta + e^{2\delta})^{1/2} \\ &= \left(2(1 - \cos\theta) + (e^\delta - 1)^2 + 2(e^\delta - 1)(1 - \cos\theta)\right)^{1/2}. \end{aligned}$$

Recalling that $\delta = \delta_n \downarrow 0$, we see that

$$\begin{aligned} |1 - qz| &\geq c_1 \delta, \quad |\theta| \leq \delta, \\ |1 - qz| &\geq c_2 |\theta|, \quad |\theta| \geq \delta, \end{aligned}$$

where c_1, c_2 are absolute constants. We call θ *small* if $|\theta| \leq \delta$. Consider the small θ's. With the eye on Equation (7),

$$\begin{aligned} \mathrm{Re}\left(- \sum_{\ell \in \mathcal{P}_i} I_\ell z^\ell \right) &\leq \sum_{\ell \in \mathcal{P}_i} I_\ell r^\ell (1 - \cos(\theta \ell)) \\ &\leq \frac{\theta^2}{2} \sum_{\ell \in \mathcal{P}_i} I_\ell \ell^2 r^\ell. \end{aligned}$$

Since $I_\ell = O(\ell^{-1} q^\ell)$, $r = q^{-1} e^\delta$ and $|\mathcal{P}_i| \leq d(n)$, the total number of divisors of n, the last expression is of order

$$\delta^2 \sum_{\ell \in \mathcal{P}_i} \ell e^{\delta \ell} \leq n e^{\delta n} \delta^2 d(n).$$

Now, according to Theorem 317 in Hardy-Wright [14], for every $\varepsilon > 0$,

$$d(n) \leq 2^{(1+\varepsilon)\log n/\log\log n}, \quad n > n_0(\varepsilon).$$

So let us finally define

$$\delta = n^{-1}(\log n - (1+2\varepsilon)\log 2 \log n/\log\log n).$$

Then, for $n \geq n_0(\varepsilon)$,

$$\begin{aligned} ne^{\delta n}\delta^2 d(n) &= \frac{e^{\delta n}(\delta n)^2 d(n)}{n} \\ &= O\big(\log^2 n \exp(-(1+2\varepsilon)\log 2\log n/\log\log n) d(n)\big) \\ &= O\big(\log^2 n \exp(-\varepsilon \log 2 \log n/\log\log n)\big) \\ &= o(1). \end{aligned}$$

Therefore

$$\left|\prod_{\ell\in\mathcal{P}_i}(1-z^\ell)^{I_\ell}\right| = O(1),$$

uniformly for $|\theta| \leq \delta$. So the contribution of these θ's to the circular contour integral is of order

$$\delta\frac{1}{\delta r^n} = \frac{q^n}{e^{n\delta}} = \frac{q^n}{n^{1-(1+2\varepsilon)\log 2/\log\log n}}.$$

(The attentive reader certainly noticed that we could have multiplied the last bound by $\exp(-\sum_\ell r^\ell I_\ell) < 1$.) Let us consider $|\theta| > \delta$. Using $I_\ell = O(\ell^{-1}q^\ell)$, and $e^x \leq 1 + xe^x$, we bound

$$\begin{aligned} \left|\sum_{\ell\in\mathcal{P}_i} z^\ell I_\ell\right| &= O\left(\sum_{\ell\in\mathcal{P}_i}\ell^{-1}(1+\delta\ell e^{\delta\ell})\right) \\ &= O\left(\sum_{\ell\in\mathcal{P}_i}\frac{1}{\ell} + \delta e^{\delta n} d(n)\right), \end{aligned}$$

using again $|\mathcal{P}_i| \leq d(n)$. The definition of δ provides

$$\begin{aligned} \delta e^{\delta n} d(n) &= O\left(\log n \exp(-(1+2\varepsilon)\log 2\log n/\log\log n) d(n)\right) \\ &= O\left(\log n \exp(-\varepsilon\log 2\log n/\log\log n)\right) \\ &= o(1). \end{aligned}$$

So this time the contribution to the contour integral is of order

$$\begin{aligned} q^n \frac{\exp(O(\sum_\ell 1/\ell))}{e^{n\delta}} \int_{|\theta|\geq\delta} \frac{d\theta}{|\theta|} \\ &= O\left(q^n \frac{\log(1/\delta)\exp(O(\sum_\ell 1/\ell))}{\exp(\log n - (1+2\varepsilon)\log 2 \log n/\log\log n)}\right) \\ &= O\left(q^n \frac{\exp(O(\sum_\ell 1/\ell))}{n^{1-(1+3\varepsilon)\log 2/\log\log n}}\right). \end{aligned}$$

In summary, the contour integral is of order

$$q^n n^{-1+(\log 2+a)/\log\log n} \exp\left(O\left(\sum_{\ell\in\mathcal{P}_i} 1/\ell\right)\right), \quad \forall\, a>0.$$

Recalling the asymptotic expression for the residue term and the fact that $\sum_{\ell\in\mathcal{P}_i} 1/\ell$ is of order $\log\log n$, we arrive at the statement. ■

Remark. We observe that the previous proposition holds for any set $\mathcal{S}$ whose elements are divisors of n, not necessarily the set $\mathcal{P}_i$.

Proposition 4.1 implies that the probability that Rabin's algorithm takes $k+1$ steps (k being the number of distinct primes of n) is asymptotic to:

$$F(n) := \prod_{\ell\in\mathcal{P}_{k+1}} \left(1-\frac{1}{q^\ell}\right)^{I_\ell},$$

where $\mathcal{P}_{k+1}$ is the set of all divisors of n except n itself. Of course, $F(n)$ is not additive but if we define $G(n)$ by

$$G(n) := \begin{cases} 0 & n=1, \\ \log F(n) & n\geq 2, \end{cases}$$

then $G(n)$ is easily seen to be additive. Using

$$\log F(n) = \sum_{\ell\leq N} \mathbf{1}_{\ell|n} I_\ell \log\left(1-\frac{1}{q^\ell}\right), \quad E\mathbf{1}_{\ell|n} = N^{-1}\left[\frac{N}{\ell}\right],$$

and the linearity of expectation, we easily get

$$\lim_{N\to\infty} E[\log F(n)] = \sum_{\ell=1}^{\infty} \frac{I_\ell}{\ell} \log\left(1-\frac{1}{q^\ell}\right).$$

Notice that in the limit $q\to\infty$ the sum reduces to

$$-\sum_{\ell=1}^{\infty} \frac{1}{\ell^2} = -\frac{\pi^2}{6}.$$

So for $E(F(n))$, the *unconditional* probability that Rabin's algorithm takes all $k+1$ steps, we have

$$\liminf_{N\to\infty,q\to\infty} E(F(n)) \geq \exp\left(\liminf_{N\to\infty,q\to\infty} E(\log F(n))\right) = \exp\left(-\frac{\pi^2}{6}\right).$$

(Since n is uniform on $[1,N]$, $n\to\infty$ in probability as $N\to\infty$. So we can use $F(n)$ as an asymptotic estimate for the (conditional on n) probability that the algorithm takes $k+1$ steps, and averaging it over n uniform on $[1,N]$, we get the limit (as $N\to\infty$) of the corresponding unconditional probability.)

Moreover, if p is a prime, then

$$G(p) = \left|I_p \log\left(1-\frac{1}{q^p}\right)\right| \sim \left|\frac{-1}{p}\right| = \frac{1}{p}.$$

We also have $G^2(p)\sim 1/p^2$, so

$$\sum_{|G(p)|>1}\frac{1}{p}, \quad \sum_{|G(p)|\leq 1}\frac{G(p)}{p}, \quad \sum_{|G(p)|\leq 1}\frac{G^2(p)}{p},$$

converge. Thus, by Theorem 5.1 in Elliot [7], $G(n)$ possesses a limit distribution and the characteristic function $v(t)$ of the limit distribution has the representation

$$v(t) = \prod_p \left(1-\frac{1}{p}\right)\left(1+\sum_{m=1}^{\infty} p^{-m}\exp(itG(p^m))\right)$$

where the product is taken over all prime numbers. This limit distribution is of pure type and is continuous since

$$\sum_{G(p)\neq 0}\frac{1}{p} = \sum_p \frac{1}{p}$$

diverges.

We have now the tools to evaluate sharply the expected value $E(q,n)$ and the variance $Var(q,n)$ of the cost of Rabin's algorithm.

Theorem 4.1. *If we let $p_{k+1}=1$, for $n\to\infty$, we have*

$$E(q,n) \quad \sim \quad \sum_{i=1}^{k+1} \frac{\prod_{\ell\in\mathcal{P}_i}\left(1-\frac{1}{q^\ell}\right)^{I_\ell}}{p_i}\ \tau_1 C_q n M(n). \tag{8}$$

PROOF. The proof is an extension of that of Theorem 3.2. First we note that

$$\begin{aligned} E(q,n) &= \sum_{i=1}^{k} P_i(q,n) \sum_{j=1}^{i} C_j(q,n) + \left(1 - \sum_{i=1}^{k} P_i(q,n)\right) \sum_{j=1}^{k+1} C_j(q,n) \\ &= \sum_{i=1}^{k} (a_i - a_{i+1}) \sum_{j=1}^{i} C_j(q,n) + a_{k+1} \sum_{j=1}^{k+1} C_j(q,n), \end{aligned}$$

where we let

$$a_i = [(z/q)^n] \frac{1}{1-qz} \prod_{\ell \in \mathcal{P}_i} (1 - z^\ell)^{I_\ell}.$$

Thus, we have

$$E(q,n) = \sum_{i=1}^{k+1} a_i C_i(q,n).$$

Proposition 4.1 gives uniformly for $1 \leq i \leq k$

$$a_i = \prod_{\ell \in \mathcal{P}_i} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} \left(1 + O(n^{-1+\varepsilon_n})\right), \tag{9}$$

and if we use the formulas (4) and (5) with $n_i = n/p_i$, we get

$$\begin{aligned} E(q,n) \sim & \left(\sum_{i=1}^{k} \frac{\prod_{\ell \in \mathcal{P}_i} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell}}{p_i} + \prod_{\ell \in \mathcal{P}_{k+1}} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} \right) \tau_1 C_q n M(n) \\ + & \left(\sum_{i=1}^{k} \prod_{\ell \in \mathcal{P}_i} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} \right) \hat{\tau_2}(n) M(n) + \prod_{\ell \in \mathcal{P}_{k+1}} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} \tau_3 M(n). \end{aligned}$$

Since τ_3 is constant, $\hat{\tau_2}(n) = O(\log n)$ and k (the number of prime factors of n) is at most of order $\log n / \log\log n$ ([26], p. 83), then the factors containing $\hat{\tau_2}(n)$ and τ_3 do not contribute to the main term of the asymptotics, and the theorem follows.

■

The p-dependent factor is the asymptotic estimate of the expected number of most expensive operation, i.e. exponentiation. It is natural to ask which ordering of prime divisors minimizes this number. The next theorem answers this question for a subset of integers n that includes squarefree numbers.

Theorem 4.2. *The ordering of prime divisors that minimizes the p-dependent factor in Equation (8) is* $p_1 > p_2 > \cdots > p_k$, *provided that* $m_1 \geq m_2 \geq \cdots \geq m_k$.

PROOF. Suppose on the contrary that, for some i, $p_i < p_{i+1}$ in the optimal ordering $\mathbf{p} = (p_1, p_2, \ldots, p_k)$. Then, according to the condition on multiplicities, we have $m_i \leq m_{i+1}$. As usual, let $\mathcal{P}_i$ denote the set of divisors of $n_1, n_2, \ldots, n_{i-1}$, that is, the set of divisors of $n/p_1, n/p_2, \ldots, n/p_{i-1}$. Alternatively, $\mathcal{P}_i$ is the set of divisors of n such that multiplicity of p_j is strictly less than m_j for at least one $j < i$. Let us introduce

$$\mathbf{p}^{'} = (p_1, \ldots, p_{i-1}, p_{i+1}, p_i, \ldots, p_k),$$

and the corresponding sets $\mathcal{P}_j^{'}$. Clearly, $\mathcal{P}_j^{'} = \mathcal{P}_j$ for $j \leq i$ and $j > i+1$, and $\mathcal{P}_{i+1}^{'}$ is the set of divisors of n multiplicity of p_j is strictly less than m_j for at least one $j \in \{1, \ldots, i-1, i+1\}$. Notice that $|\mathcal{P}_{i+1}| \leq |\mathcal{P}_{i+1}^{'}|$, since $m_i \leq m_{i+1}$. Let us denote

$$\Pi_j = \prod_{\ell \in \mathcal{P}_j} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} \quad \text{and} \quad \Pi_j^{'} = \prod_{\ell \in \mathcal{P}_j^{'}} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell},$$

so that $\Pi_j = \Pi_j^{'}$ for $j \leq i$ and $j > i+1$. Since $\mathbf{p}$ is optimal we must have that

$$\frac{\Pi_i}{p_i} + \frac{\Pi_{i+1}}{p_{i+1}} \leq \frac{\Pi_i}{p_{i+1}} + \frac{\Pi_{i+1}^{'}}{p_i},$$

that is,

$$\frac{1}{p_i}\left(\Pi_i - \Pi_{i+1}^{'}\right) \leq \frac{1}{p_{i+1}}\left(\Pi_i - \Pi_{i+1}\right).$$

Since $p_i < p_{i+1}$ and $\Pi_i - \Pi_{i+1} > 0$, we must have then

$$\Pi_i - \Pi_{i+1}^{'} \leq \Pi_i - \Pi_{i+1}.$$

Therefore, we have

$$\prod_{\ell \in \mathcal{P}_{i+1}} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} \leq \prod_{\ell \in \mathcal{P}_{i+1}^{'}} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell}. \tag{10}$$

Furthermore, let R be the intersection of $\mathcal{P}_{i+1}$ and $\mathcal{P}_{i+1}^{'}$. Then, denoting $e_j(\ell)$ the multiplicity of p_j in a divisor ℓ of n,

$$\mathcal{P}_{i+1} - R = \{\ell | n : \; e_j(\ell) = m_j, (j < i); \; e_i(\ell) < m_i\},$$

and

$$\mathcal{P}_{i+1}^{'} - R = \{\ell | n : \; e_j(\ell) = m_j, (j < i); \; e_{i+1}(\ell) < m_{i+1}\}.$$

Since $m_i \leq m_{i+1}$, to each $\ell = \cdots p_i^{e_i} \cdots \in \mathcal{P}_{i+1}$ we can associate $\ell' = f(\ell) = \cdots p_{i+1}^{e_i} \cdots \in \mathcal{P}'_{i+1}$, with $\ell' > \ell$ if $e_i > 0$, and $\ell' = \ell$ if $e_i = 0$. Setting $f(\ell) = \ell$ for $\ell \in R$, we obtain an *injective* mapping f of $\mathcal{P}_{i+1}$ into $\mathcal{P}'_{i+1}$ such that if $\ell' = f(\ell)$, then $\ell' \leq \ell$. Besides, $\{\ell \in \mathcal{P}_{i+1} : \ell > f(\ell)\} \neq \emptyset$. Then, using Lemma 4.1 below, we must have

$$\prod_{\ell \in \mathcal{P}_{i+1}} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} > \prod_{\ell \in \mathcal{P}_{i+1}} \left(1 - \frac{1}{q^{f(\ell)}}\right)^{I_{f(\ell)}} \geq \prod_{\ell \in \mathcal{P}'_{i+1}} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell},$$

(as $|\mathcal{P}_{i+1}| \leq |\mathcal{P}'_{i+1}|$), which contradicts Equation (10). ■

Lemma 4.1. *The numbers*

$$\left(1 - \frac{1}{q^\ell}\right)^{I_\ell}$$

strictly increase with ℓ.

Proof. We need to show that

$$I_\ell \log \left(1 - \frac{1}{q^\ell}\right)^{-1} > I_{\ell+1} \log \left(1 - \frac{1}{q^{\ell+1}}\right)^{-1},$$

that is,

$$\frac{I_\ell}{q^\ell} > \frac{I_{\ell+1}}{q^{\ell+1}} \frac{q^{\ell+1} \log \left(1 - \frac{1}{q^{\ell+1}}\right)^{-1}}{q^\ell \log \left(1 - \frac{1}{q^\ell}\right)^{-1}}.$$

For $x > 1$,

$$x \log \left(1 - \frac{1}{x}\right)^{-1} = x \left(\frac{1}{x} + \frac{1}{2x^2} + \cdots\right) = 1 + \frac{1}{2x} + \frac{1}{3x^2} + \cdots$$

decreases with x. Therefore, it is enough to show that

$$\frac{I_\ell}{q^\ell} > \frac{I_{\ell+1}}{q^{\ell+1}}. \tag{11}$$

Using Equation (1), the inequality (11) holds if

$$\frac{q^\ell - 2q^{\ell/2}}{\ell q^\ell} > \frac{1}{\ell + 1},$$

or equivalently, if

$$q^{\ell/2} > 2(\ell + 1).$$

The last inequality holds for all $q \geq 7$ and $\ell \geq 2$, for $q = 5, 4$ and $\ell \geq 3$, for $q = 3$ and $\ell \geq 5$, and for $q = 2$ and $\ell \geq 9$. Maple helps to verify Equation (11) for all (finitely many) remaining values of q and ℓ. ■

Remark. For a variation of Rabin's algorithm, Gao & Panario ([10], § 2) had suggested that the above ordering of prime factors might optimize, asymptotically, the average cost of running the algorithm. We prove that this is so indeed in Section 5, without any conditions on the multiplicities m_j. Does the ordering $p_1 > \cdots > p_k$ remain optimal for Rabin's algorithm when the condition $m_1 \geq \cdots \geq m_k$ is not meet? Finding an answer appears to be an interesting open problem.

Theorem 4.3. *If we let $p_{k+1} = 1$, for $n \to \infty$,*

$$Var(q,n) \sim \left(2\sum_{i=1}^{k+1} \frac{\prod_{\ell\in\mathcal{P}_i}\left(1-\frac{1}{q^\ell}\right)^{I_\ell}}{p_i} \sum_{j=1}^{i} \frac{1-\prod_{\ell\in\mathcal{P}_j}\left(1-\frac{1}{q^\ell}\right)^{I_\ell}}{p_j} \right.$$
$$\left. - \sum_{i=1}^{k+1} \frac{\prod_{\ell\in\mathcal{P}_i}\left(1-\frac{1}{q^\ell}\right)^{I_\ell}\left(1-\prod_{\ell\in\mathcal{P}_i}\left(1-\frac{1}{q^\ell}\right)^{I_\ell}\right)}{p_i^2} \right) \tau_1^2 C_q^2 n^2 M^2(n).$$

Proof. The proof is similar to that of Theorem 3.3 using the formula $Var[X] = E[X^2] - E[X]^2$. Then we have

$$\begin{aligned} Var(q,n) &= \sum_{i=1}^{k} P_i(q,n)\left(\sum_{j=1}^{i} C_j(q,n)\right)^2 \\ &\quad + \left(1-\sum_{i=1}^{k} P_i(q,n)\right)\left(\sum_{j=1}^{k+1} C_j(q,n)\right)^2 - E^2(q,n) \\ &= \sum_{i=1}^{k}(a_i - a_{i+1})\left(\sum_{j=1}^{i} C_j(q,n)\right)^2 \\ &\quad + a_{k+1}\left(\sum_{j=1}^{k+1} C_j(q,n)\right)^2 - E^2(q,n), \end{aligned}$$

where we let

$$a_i = [(z/q)^n]\frac{1}{1-qz}\prod_{\ell\in\mathcal{P}_i}(1-z^\ell)^{I_\ell}.$$

After expanding the squares and simplifying the obtained expressions we obtain

$$Var(q,n) = \sum_{i=1}^{k+1} a_i C_i(q,n)\left(2\sum_{j=1}^{i}(1-a_j)C_j(q,n) - (1-a_i)C_i(q,n)\right).$$

The result follows after using Proposition 4.1, formulas (4), (5) and (9), and the simplifications introduced in the proof of Theorem 4.1 to discard the terms involving $\hat{\tau_2}(n)$ and τ_3. ■

This last result shows that the cost of Rabin's algorithm is dominated by the cost of computing the exponentiations. Since in the worst case the algorithm performs $O(n \log\log\log n M(n) \log q)$ operations, we see that in average the algorithm is a factor of $\log\log\log n$ faster. Moreover, both the expected value and the standard deviation are $O(nM(n)\log q)$. This large deviation is explained by the fact that even though most of the time only one gcd test is executed, the number of operations performed by the algorithm when more than one gcd is computed is very high.

5. Variants

The analysis of Rabin's algorithm can be adapted to other algorithms with similar divisor conditions. In this section, we briefly show that this is the case for two variants of Rabin's method. As we will see, both variants give different ways of computing the exponentiations inside the algorithm. However, they use the same divisor construction as in Rabin's irreducibility test.

First we comment on an algorithm due to von zur Gathen & Shoup ([11], § 7). Their algorithm initially calculates all the exponentiations $x^{q^n/p_i} \bmod f$ by computing trace maps (Algorithm 5.2 in [11]). Then, it computes the gcd tests as in Rabin's algorithm. Using fast arithmetic, their algorithm uses $(kn\log n + C_q)\tau_1 M(n)$ operations to perform the exponentiations. There are at most $O(\log n/\log\log n)$ prime divisors of n, thus the total worst-case cost of their algorithm is $O(n^2\log^3 n + M(n)\log q)$. The space requirement of the algorithm is $O(n)$ elements in $\mathbb{F}_q$, as in the other algorithms of this paper.

We note that von zur Gathen & Shoup also give another algorithm with cost $O(n^{1.7} + M(n)\log q)$ in time. However, this time is achieved by using fast matrix multiplication and it has a space requirement of $O(n\log n)$ elements in $\mathbb{F}_q$. Thus, it seems to be of mainly theoretical interest.

We now focus on the average-case analysis of von zur Gathen & Shoup's algorithm. The proofs are similar to those of Theorems 4.1 and 4.3. We only give results for fast arithmetic.

Corollary 5.1. *Let $E(q,n)$ and $Var(q,n)$ denote the expected value and the variance of the number of operations used by von zur Gathen*

& Shoup's algorithm. Then, for $n \to \infty$,

$$E(q,n) \sim (kn \log n + C_q)\tau_1 M(n).$$

$$Var(q,n) \sim \left(\sum_{i=1}^{k} (2i-1) \prod_{\ell \in \mathcal{P}_i} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} - \left(\sum_{i=1}^{k} \prod_{\ell \in \mathcal{P}_i} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} \right)^2 \right) \tau_2^2 \log^2 n M^2(n).$$

This variation is interesting in several senses. First, the standard deviation of the number of operations used by this algorithm is much smaller than its expected value. The reason is that this algorithm computes all the exponentiations at the beginning, so the variation is only due to the distribution of degree factors in a random polynomial. Moreover, it may seem at first that its behavior is worse than the original Rabin's algorithm. However, this depends on the relationship between C_q and n. As we have seen, $C_q = \lfloor \log_2 q \rfloor + \nu(q) - 1$, with $\nu(q)$ the number of ones in the binary representation of q, and so $C_q = O(\log q)$. Thus, for example, if $C_q \sim n$, this variation is faster than Rabin's algorithm, since it does only one repeating squaring (plus some trace map computations) to compute the exponentiations. On the other hand, if $C_q = o(\log n)$ then Rabin's algorithm is faster.

Here is a second variant for the computation of $x^{q^{n_i}} \bmod f$, for $1 \leq i \leq k$, mentioned in the remark following Lemma 4.1 (see [10], § 2). The key idea in this algorithm is to sort the exponents n_i in an increasing order (equivalently, to put p_i in the decreasing order), and then to compute, for suitable i,

$$\left(x^{q^{n_i}}\right)^{q^{n_{i+1}-n_i}} \equiv \quad x^{q^{n_{i+1}}} \bmod f.$$

The appearance of the increments $n_{i+1} - n_i$ provides, via a telescopic effect, for some reduction of running time in the worst-case analysis. More precisely, they prove that this variant correctly tests for polynomial irreducibility, and uses $O(nM(n)C_q)$ operations in $\mathbb{F}_q$ in the worst-case, as compared to $O(nM(n)C_q \log\log\log n)$ for Rabin's method. Thus, it behaves better than Rabin's method in the worst-case. Gao & Panario's algorithm uses $\tau_1(n_i - n_{i-1})M(n)C_q$ operations to compute the ith modular exponentiation with $2 \leq i \leq k$, $\tau_1 n_1 M(n)C_q$ operations to compute the first modular exponentiation, whence $\tau_1(n - n_k)M(n)C_q$ operations to compute the last one.

We now give an average-case estimate (for both classical and fast arithmetic), for the cost of this algorithm, based on Theorems 4.1 and 4.3.

Corollary 5.2. *Let $E(q,n)$ and $Var(q,n)$ denote the expected value and the variance of the number of operations used by Gao & Panario's algorithm. Then, if we let $p_{k+1}=1$ and $p_0=\infty$, for $n\to\infty$,*

$$E(q,n) \sim \left(\sum_{i=0}^{k}\prod_{\ell\in\mathcal{P}_{i+1}}\left(1-\frac{1}{q^\ell}\right)^{I_\ell}\left(\frac{1}{p_{i+1}}-\frac{1}{p_i}\right)\right)\tau_1 C_q n M(n) \quad (12)$$

$$\begin{aligned} Var(q,n) \sim \Bigg(& 2\sum_{i=2}^{k+1}\prod_{\ell\in\mathcal{P}_i}\left(1-\frac{1}{q^\ell}\right)^{I_\ell}\left(\frac{1}{p_i}-\frac{1}{p_{i-1}}\right) \\ & \sum_{j=2}^{i}\left(1-\prod_{\ell\in\mathcal{P}_j}\left(1-\frac{1}{q^\ell}\right)^{I_\ell}\right)\left(\frac{1}{p_j}-\frac{1}{p_{j-1}}\right) \\ & -\sum_{i=2}^{k+1}\prod_{\ell\in\mathcal{P}_i}\left(1-\frac{1}{q^\ell}\right)^{I_\ell}\left(1-\prod_{\ell\in\mathcal{P}_i}\left(1-\frac{1}{q^\ell}\right)^{I_\ell}\right)\left(\frac{1}{p_i}-\frac{1}{p_{i-1}}\right)^2\Bigg) \\ & \tau_1^2 C_q^2 n^2 M^2(n). \end{aligned}$$

Although this variation does not change the fact that both the expected value and the standard deviation are $O(C_q n M(n))$, the constants are much smaller than in the original algorithm. Furthermore, it shows again that the main contribution to the cost is originated by the few times the algorithm has to execute the last step.

As in the case of Rabin's algorithm, putting n_i in increasing order, we can expect that the current variant will require, on average, a minimal number of operations. For example, as can be seen in Table 1, for $q=2$ and $n=5\times 31=155$, it is better to choose $p_1=31$ and $p_2=5$ than $p_1=5$ and $p_2=31$, since in the first case over 99% of the polynomials rejected in the first two steps are rejected in the first step, and this number is less than 96% in the second case. The reason is that in the first case, a polynomial is rejected in the first step if it has an irreducible factor of degree 1 or 5, while in the second case if it has an irreducible factor of degree 1 or 31. This variation tests the smaller degrees as early as possible, and so optimizes these probabilities.

Let us now show that the ordering $p_1>p_2>\cdots>p_k$ minimizes the p-dependent factor in Equation (12) for *arbitrary* multiplicities $m_1,m_2,\ldots,m_k$. If not, in the optimal ordering $\mathbf{p}=(p_1,\ldots,p_k)$ we must have $p_i<p_{i+1}$ for some i. Then, $1\le i$, and $i+1\le k$. As before,

we introduce $\mathbf{p}' = (p_1, \ldots, p_{i-1}, p_{i+1}, p_i, \ldots, p_k)$, the corresponding sets $\mathcal{P}'_j$, and the products Π'_j. Since $\mathbf{p}$ is optimal we must have

$$\begin{aligned} & \Pi_i \left(\frac{1}{p_i} - \frac{1}{p_{i-1}}\right) + \Pi_{i+1} \left(\frac{1}{p_{i+1}} - \frac{1}{p_i}\right) + \Pi_{i+2} \left(\frac{1}{p_{i+2}} - \frac{1}{p_{i+1}}\right) \\ \leq \; & \Pi_i \left(\frac{1}{p_{i+1}} - \frac{1}{p_{i-1}}\right) + \Pi'_{i+1} \left(\frac{1}{p_i} - \frac{1}{p_{i+1}}\right) + \Pi_{i+2} \left(\frac{1}{p_{i+2}} - \frac{1}{p_i}\right), \end{aligned}$$

or

$$\frac{\Pi_i}{p_i} + \Pi_{i+1} \left(\frac{1}{p_{i+1}} - \frac{1}{p_i}\right) - \frac{\Pi_{i+2}}{p_{i+1}} \leq \frac{\Pi_i}{p_{i+1}} + \Pi'_{i+1} \left(\frac{1}{p_i} - \frac{1}{p_{i+1}}\right) - \frac{\Pi_{i+2}}{p_i}.$$

Grouping terms containing Π_i and Π_{i+2} respectively, and canceling the positive term $\frac{1}{p_i} - \frac{1}{p_{i+1}}$, we have

$$\Pi_i - \Pi_{i+1} \leq \Pi'_{i+1} - \Pi_{i+2},$$

that is,

$$1 - \frac{\Pi_{i+1}}{\Pi_i} \leq \frac{\Pi'_{i+1}}{\Pi_i} \left(1 - \frac{\Pi_{i+2}}{\Pi'_{i+1}}\right).$$

Here

$$\frac{\prod'_{i+1}}{\prod_i} = \prod_{\substack{p_1 \cdots p_{i-1} \mid \ell \\ p_{i+1} \nmid \ell}} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} \leq 1,$$

so that

$$\frac{\Pi_{i+1}}{\Pi_i} > \frac{\Pi_{i+2}}{\Pi'_{i+1}}.$$

Furthermore, observe that $\mathcal{Q}_{i+1} = \mathcal{P}_{i+1} \setminus \mathcal{P}_i$ consists of the divisors ℓ of n such that $e_j(\ell) = m_j$ for $j < i$ and $e_i(\ell) < m_i$, cf. Equation (6). Likewise, $\mathcal{Q}'_{i+2} := \mathcal{P}_{i+2} \setminus \mathcal{P}'_{i+1}$ consists of the divisors ℓ of n such that $e_j(\ell) = m_j$ for $j < i$ *and* $j = i+1$, and $e_i(\ell) < m_i$. Consequently, $\mathcal{Q}_{i+1} \supset \mathcal{Q}'_{i+2}$. (!) Therefore

$$\frac{\Pi_{i+1}}{\Pi_i} = \prod_{\ell \in \mathcal{Q}_{i+1}} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} \leq \prod_{\ell \in \mathcal{Q}'_{i+2}} \left(1 - \frac{1}{q^\ell}\right)^{I_\ell} = \frac{\Pi_{i+2}}{\Pi'_{i+1}},$$

a contradiction. So the optimal ordering is $p_1 > p_2 > \cdots > p_k$. (We observe that we do not need Lemma 4.1 in this proof.)

6. Conclusions

This paper analyzes Rabin's algorithm for testing the irreducibility of polynomials over finite fields and some of its variants. The correctness of the algorithm is based on Theorem 2.1. However, a direct implementation as suggested by the theorem does not lead to an efficient algorithm. Indeed, first we note the presence of redundancies in the computations. Since in each gcd we are checking the appearance of irreducible factors of degrees dividing a maximal divisor of n, it is possible that two gcds check some common degrees. For instance, all gcds check the linear factors. (Of course, this fact is intrinsic to the Theorem 2.1). On the other hand, in most of the cases only the first gcd test is performed. The polynomials that survive the first gcd test are likely to survive all gcds. Thus, the effect of redundancies is not crucial.

A much more important weakness of these algorithms is the large computation involved in the exponentiations, especially the first one. We indicate that the bottleneck of all algorithms discussed in this paper is the computation of exponentiations modulo a polynomial. The variance of the number of operations performed by most of these algorithms is very high, and for several cases the standard deviation is actually higher than the expected value. It would be very interesting to find an efficient way to perform the exponentiations, since it is a standard ingredient of a broad variety of algebraic algorithms.

A different probabilistic algorithm for testing the irreducibility of polynomials over finite fields is due to Ben-Or [3]. A detailed analysis of Ben-Or's algorithm is given in [21]. It involves the study of the expected smallest factor degree of a random polynomial over $\mathbb{F}_q$. The expected number of operations performed by this algorithm is $O(C_q \log n M(n))$, and this average is by factor $\log n/n$ lower than Rabin's. The main reason is that Ben-Or's algorithm quickly detects irreducible factors of small degrees, thus performing much less computations than Rabin's. For example, to test for linear irreducible factors, Ben-Or's method computes $x^q \bmod f$, while Rabin's executes the much more expensive exponentiation $x^{q^{n/p_1}} \bmod f$, for some prime factor p_1 of n. The efficiency of Ben-Or's algorithm is due to distribution of degree factors in a random polynomial, that leads to expect very often irreducible factors of small degree [16]. However the variance of Ben-Or's algorithm is also very high ($O(C_q n M(n))$), a fact that shows again the impact of the cost of performing the exponentiations.

Acknowledgments. Most of the work of the first author was done while he was with the Department of Computer Science of the University of Toronto. Part of this work was done while the fourth author was visiting the University of Waterloo. For the invitation, support and hospitality, he would like to thank Ian Munro and the Department of Computer Science. The research of Boris Pittel was supported in part by the NSF Grant DMS98-03410. The work of Alfredo Viola was supported in part by Proyecto BID-CONICYT 140/94, and Proyecto BID Fondo Clemente Estable 2078/96. An extended abstract of a preliminary version of this paper appeared in *Latin American Theoretical INformatics* (LATIN'98), Campinas, Brazil, April 20-24, 1998 [22].

References

[1] G.B. Agnew, R.C. Mullin, and S.A. Vanstone. An implementation of elliptic curve cryptosystem over $\mathbb{F}_{2^{155}}$. *IEEE J. Selected Areas Commun.*, 11:804–813, 1993.

[2] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading MA, 1974.

[3] M. Ben-Or. Probabilistic algorithms in finite fields. In *Proc. 22nd IEEE Symp. Foundations Computer Science*, pages 394–398, 1981.

[4] I.F. Blake, R. Fuji-Hara, R.C. Mullin, and S.A. Vanstone. Computing discrete logarithms in finite fields of characteristic two. *SIAM J. Alg. Disc. Meth.*, 5:276–285, 1984.

[5] D.G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta. Inform.*, 28:693–701, 1991.

[6] D. Coppersmith. Fast evaluation of logarithms in fields of characteristic two. *IEEE Trans. Info. Theory*, 30:587–594, 1984.

[7] P.D.T.A. Elliot. Probabilistic Number Theory I. Springer Verlag, 1979.

[8] P. Flajolet, X. Gourdon, and D. Panario. The complete analysis of a polynomial factorization algorithm over finite fields. To appear in *J. of Algorithms*. [Extended abstract in *Proc. 23rd ICALP Symp.*, Lecture Notes in Computer Science, vol. 1099, p. 232–243, 1996.] Full version in technical report 3370, INRIA, March 1998.

[9] É. Galois. Sur la théorie des nombres. In R. Bourgne and J.P. Arza, editors, *Écrits et mémoires d'Évariste Galois*, pages 112–128. Gauthier-Villars, 1830.

[10] S. Gao and D. Panario. Tests and constructions of irreducible polynomials over finite fields. In F. Cucker and M. Shub, editors, *Foundations of Computational Mathematics*, pages 346–361. Springer Verlag, 1997.

[11] J. von zur Gathen and V. Shoup. Computing Frobenius maps and factoring polynomials. *Comput complexity*, 2:187–224, 1992.

[12] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.

[13] C.F. Gauss. *Untersuchungen über Höhere Mathematik*. Chelsea, New York, 1889.

[14] G.H. Hardy and E.M. Wright. *An Introduction to the Theory of Numbers*. Clarendon Press, Oxford, 1962.

[15] A. Knopfmacher and J. Knopfmacher. Counting polynomials with a given number of zeros in a finite field. *Linear and Multilinear Algebra*, 26:287–292, 1990.

[16] J. Knopfmacher and A. Knopfmacher. Counting irreducible factors of polynomials over a finite field. *SIAM Journal on Discrete Mathematics*, 112:103–118, 1993.

[17] D.E. Knuth. *The art of computer programming, vol.2: seminumerical algorithms*. Addison-Wesley, Reading MA, 3 edition, 1997.

[18] M. Mignotte. *Mathematics for Computer Algebra*. Springer-Verlag, New York, 1992.

[19] K.S. McCurley. The discrete logarithm problem. In *Proc. of Symposia in Applied Mathematics*, pages 49–74. American Mathematical Society, 1990.

[20] A. Odlyzko. Discrete logarithms and their cryptographic significance. In *Advances in Cryptology, Proceedings of Eurocrypt 1984*, volume 209 of *Lecture Notes in Computer Science*, pages 224–314. Springer-Verlag, 1985.

[21] D. Panario and B. Richmond. Analysis of Ben-Or's polynomial irreducibility test. *Random Struct. Alg.*, 13:439–456, 1998.

[22] D. Panario and A. Viola. Average-case analysis of Rabin's polynomial irreducibility test. In C. L. Lucchesi and A. V. Moura, editors, *Latin American Theoretical INformatics*, volume 1380 of *Lecture Notes in Computer Science*, pages 1–10. Springer-Verlag, 1998. Proceedings of LATIN'98, Campinas, April 1998.

[23] M.O. Rabin. Probabilistic algorithms in finite fields. *SIAM J. Comp.*, 9:273–280, 1980.

[24] A. Schönhage. Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2. *Acta Inf.*, 7:395–398, 1977.

[25] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.

[26] G. Tenenbaum. Introduction to Analytic and Probabilistic Number Theory. Cambridge Studies in Advanced Mathematics 46, Cambridge University Press, New York, 1995.

School of Mathematics and Statistics, Carleton University, K1S 5B6, Ottawa, Canada, daniel@math.carleton.ca

Department of Mathematics, Ohio State University, 231 W 18th Avenue, Columbus OH 43210-1174, USA, bgp@math.ohio-state.edu

Department of Combinatorics and Optimization, University of Waterloo, N2L 3G1, Waterloo, Canada, lbrichmond@watdragon.uwaterloo.ca

Instituto de Computación, Universidad de la República, Casilla de Correo 16120, Distrito 6, Montevideo, Uruguay, viola@fing.edu.uy