Chapter 18
The Quadratic Sieve Algorithm for Integer Factoring

Kannan Balasubramanian
Mepco Schlenk Engineering College, India

M. Rajakani
Mepco Schlenk Engineering College, India

ABSTRACT
At the time when RSA was invented in 1977, factoring integers with as few as 80 decimal digits was intractable. The first major breakthrough was quadratic sieve, a relatively simple factoring algorithm invented by Carl Pomerance in 1981, which can factor numbers up to 100 digits and more. It’s still the best-known method for numbers under 110 digits or so; for larger numbers, the general number field sieve (GNFS) is now used. However, the general number field sieve is extremely complicated, for even the most basic implementation. However, GNFS is based on the same fundamental ideas as quadratic sieve. The fundamentals of the Quadratic Sieve algorithm are discussed in this chapter.

INTRODUCTION
At the time when RSA was invented in 1977, factoring integers with as few as 80 decimal digits was intractable; all known algorithms were either too slow or required the number to have a special form. This made even small, 256-bit keys relatively secure. The first major breakthrough was quadratic sieve, a relatively simple factoring algorithm invented by Carl Pomerance in 1981, which can factor numbers up to 100 digits and more. It’s still the best-known method for numbers under 110 digits or so; for larger numbers, the general number field sieve (GNFS) is now used. However, the general number field sieve is extremely complicated, and requires extensive explanation and background for even the most basic implementation. However, GNFS is based on the same fundamental ideas as quadratic sieve. The chapter discusses the fundamentals of the Quadratic Sieve algorithm.
FINDING A SUBSET OF INTEGERS WHOSE PRODUCT IS A SQUARE

Suppose I give you a set of integers and I ask you to find a subset of those integers whose product is a square, if one exists. For example, given the set \{10, 24, 35, 52, 54, 78\}, the product 24×52×78 is 97344 = 312². The brute-force solution, trying every subset, is too expensive because there are an exponential number of subsets.

Another approach is based on prime factorizations and linear algebra. First, we factor each of the input numbers into prime factors; for now, we will assume that these numbers are easy to factor. For the above example set, we get:

\[
10 = 2 \times 5 \\
24 = 2^3 \times 3 \\
35 = 5 \times 7 \\
52 = 2^2 \times 13 \\
54 = 2 \times 3^3 \\
78 = 2 \times 3 \times 13
\]

When you multiply two numbers written as prime factorizations, you simply add the exponents of the primes used. For example, the exponent of 2 in 24×52×78 is 6, because it’s 3 in 24, 2 in 52, and 1 in 78. A number is a square if and only if all the exponents in its prime factorization are even. Suppose we write the above factorizations as vectors, where the \(k\)th entry corresponds to the exponent of the \(k\)th prime number. We get:

\[
\begin{align*}
\[1 & 0 & 1 & 0 & 0 & 0\] \\
\[3 & 1 & 0 & 0 & 0 & 0\] \\
\[0 & 0 & 1 & 1 & 0 & 0\] \\
\[2 & 0 & 0 & 0 & 0 & 1\] \\
\[1 & 3 & 0 & 0 & 0 & 0\] \\
\[1 & 1 & 0 & 0 & 0 & 1\]
\end{align*}
\]

Now, multiplying numbers is as simple as adding vectors. If we add rows 2, 4, and 6, we get \[6 2 0 0 0 2\], which has all even exponents and so must be a square. In more familiar terms, we want the last