# Math 571-01, Cryptography Project 02
# Quadratic Sieve
# University of Massachusetts, Amherst


Matthew Gramigna
Wei Xie
Barry Greengus


April 4, 2017

# 1 Introduction

The quadratic sieve is an efficient means of finding many numbers greater than the square root of a given $N$ whos squares modulus $N$ are $B - smooth$ for a given positive integer $B$. i.e

Let $N, B \in Z$
find $a^2 \mod (N)$ s.t. $\forall p_i$ in $a^2 = p_1^{e_1} * p_2^{e_2} * ... * p_k^{e_k}$, $p_i \leq B$

## 1.1 Difference by Squares Factoring

Why is this useful? Consider the problem of factoring, specifically the method of factoring using difference of squares. If a number $N$ is know to be the difference of two squares, say $X = Y^2 - Z^2$, then $X = (Y+Z)(Y-Z)$. So all we have to do to factor $N$ is to find a number $b$ such that $N + b^2$ is a perfect square. Then $N + b^2 = a^2$, so

$$N = a^2 - b^2 = (a+b)(a-b)$$

and we have just factored N.

A random value of $b$ is unlikely to produce a perfect square, but it is fairly likely for a multiple $k$ of $N$ to equal the difference of two squares

$$kN = a^2 - b^2 = (a+b)(a-b)$$

such that $(a+b)$ or $(a-b)$, besides for being a factor of $kN$, is also a non-trivial factor of $N$. This means we only need to find a difference of two squares that equals a mutiple of $N$, which is the equivalent of finding $a$ and $b$ such that $a^2 \equiv b^2 \pmod{N}$.
This fact enables a three step factoring algorithm comprised of:
**Step1**:
**Step2**:
**Step3**:

Quadratic sieve solves the first step of this algorithm.

# 2 Overview of Quadratic Sieve

Explain how quadratic sieve works TODO more info +equations + fix ”=” to congruent
**Setup Step**: Given number N and set of primes P, where all elemenet in P ¿= B, set a=floor of the sqrt(N), set a quadratic polynomial. we will use F(T) = T2̂ - 221.

**Step 1**: build a list of F(a) to F(L(a)). TODO define L(), explain why we use it. explain why we start at a.

**Step 2**: For i=2 to B, where i=some p in P or is prime factor of some p in P:

**Step 3**: Predict where division of elem in list by i CAN happen.
if p — F(T), then T$\hat{2}$ = N mod p has a solution, else no solution so you cant divide by p.
So, if p odd and T$\hat{2}$ = N mod p has two solutions, a and b. all mulitples of those solutions can also be divided by the p

**Step 4**: Divide all multiples of the solutions a and b in the list by p

**Step 5**: Whenever the quotient of a list element is 1, it's prime factors are clearly only primes ¡= B and is thus B-smooth

# 3 Implementation

We used GP, TODO more info

## 3.1 Initial Approach

## 3.2 Final Implementation

## 3.3 Interesting Details/etc?

## 3.4 Testing

# 4 Efficiency

# 5 Source Code

# 6 Group Organization/Administrative

## 6.1 Git

TODO, we used git bc useful for XYZ

## 6.2 Meetings

met how often? helpful bc why?