A Probabilistic Public Key Encryption Scheme Based on Quartic Reciprocity (Draft 1.41)

Robert A. Threlfall
955 Broadway Drive, Sun Prairie, Wisconsin 53590, United States
rthrelfall@secure-systems.org

July 9, 2022

Abstract

Using a novel class of single bit one-way trapdoor functions we construct a theoretical probabilistic public key encryption scheme that has many interesting properties. These functions are constructed from binary quadratic forms and rational quartic reciprocity laws. They are not based on class group operations nor on universal one-way hash functions. Inverting these functions appears to be as difficult as factoring, and other than factoring, we know of no reductions between this new number theory problem and the standard number theoretic problems used cryptographically.

We are unable to find away to construct a ciphertext without knowing the plaintext, hence this encryption scheme appears to be plaintext aware ($PA1$). By using quartic reciprocity properties there is less information leakage than with quadratic reciprocity based schemes and consequently this encryption scheme appears to be completely non-malleable as defined by M. Fischlin (2005), and strongly plaintext aware (SPA) and secret-key aware (SKA) as well, as defined by M. Barbosa and P. Farshim (2009). Assuming plaintext awareness ($PA1$), the difficulty of inverting our one-way trapdoor function and the hardness of certain standard number theoretic problems, then this scheme is provably secure against adaptive chosen ciphertext attacks ($IND - CCA2$).

The public key is a product of two secret primes. Decryption is fast, requiring just one modular multiplication and one Jacobi symbol evaluation. The encryption step is polynomial time, but slow, and there is a great deal of message expansion. However, the encryption step is amenable to parallelization, both across bits, as well as at the level of encrypting a single bit. The encryption step is also amenable to asynchronous pre-computation. After the pre-computation step, for a t bit public key, encryption only requires three multiplications (with $t + 2c + 5$ bit length numbers) per encrypted bit, where $100 \leq c \leq 150$ is an adjustable security parameter. The computational cost to break an encrypted bit can be optionally adjusted down on a per bit basis.

With no additional keys, multiple senders can individually join secret information to each encrypted bit without changing the parity of the encrypted bit. (Recovering this secret information is harder than recovering the private key.) Each sender can separately and publicly reveal their secret information without revealing the plaintext bit. The senders of the encrypted message bit can also individually authenticate they are senders without the use of a message authentication code and without revealing the plaintext bit.

We are not aware of any hardware faults or other adverse events that might occur during decryption that could be exploited to break the secret key. Encryption faults can occur that could be exploited to reveal plaintext bits, however, these faults can be detected with high probability and with low computational cost.

Keywords. Probabilistic public-key encryption, Adaptive chosen-ciphertext attack, Binary quadratic forms, Class number, Quadratic residues, Quadratic nonresidues, Quartic reciprocity

1. Introduction and Preliminaries

We construct a probabilistic public key cryptographic encryption scheme based on a new class of one-way trap door single bit functions. The proof of security relies on the assumed difficulty of factoring and a new number theoretic problem that appears in practice to be as difficult as factoring. Based on some reasonable heuristics on prime number densities for numbers represented by binary positive definite quadratic forms, this cryptosystem system has an expected polynomial run time for key construction, encryption and decryption. However, the encryption step is very slow and there is considerable message expansion. Despite these limitations, this cryptographic system has some interesting properties that we hope will stimulate further research. Since these one-way trapdoor functions are based on rational quartic reciprocity and quadratic forms, this paper may be of interest to number theorists as well as cryptologists. Accordingly, we have tried to make this article less technical and more broadly accessible than would normally be the case.

The run time to multiply two m-bit numbers is $O(m^u)$ bit operations, where u equals 2 for classical multiplication algorithms. More efficient algorithms exist where u is as small as $1 + \varepsilon$ asymptotically [8, page 111]. (If these more efficient algorithms are used, then the exponents in the run times for both decryption and encryption will be lowered by about 1.) The focus of this paper is theory, not implementation, and whether the public key for this encryption scheme would actually be large enough that these asymptotic improvements could be achieved is outside the scope of this paper. Using these more efficient algorithms may well speed up the encryption step sufficiently that this encryption scheme could be used in the real world for certain applications as the number of bit operations for encryption would be reduced from $O(m^4)$ to $O(m^{3+\varepsilon})$. However, for the sake of consistency and to more easily compare this encryption scheme with other encryption schemes, such as RSA and Goldwasser-Micali, we will only give run times assuming classical algorithms.

This paper is organized as follows. First, we give some preliminary definitions and theorems on binary quadratic forms. Next, we describe the encryption scheme and variations. Third, the essential number theory theorems that make the one-way trapdoor function possible are given. (The proofs of these theorems are provided in the appendix.) Fourth, we analyze the run time of the system. Lastly, we discuss the security and complexity of the system.

The following definitions are somewhat informal. However, they are sufficient for our purposes and encapsulate the essential concepts.

Definition 1 (Trapdoor one-way function)**.** [29, Definitions 1.12 and 1.16] A *one-way function* $f : X \longrightarrow Y$, means evaluation in one direction is easy for all $x \in X$, but evaluation in the inverse direction is very difficult and infeasible for most elements of Y. For the one-way function to be a *trapdoor function* there is the additional essential condition that there is supplementary information (called the trap door information), that when made available changes the computation in the inverse direction from very difficult to feasible. In other words it becomes viable to compute for any given $y \in \operatorname{Im}(f)$, an $x \in X$ such that $f(x) = y$.

Definition 2 (Polynomial time)**.** [1, page 45] For a string x, signify the length of x by $\lg x$. A function g is *computable in polynomial time* if there is an algorithm to compute $g(x)$, that takes at most $P(\lg x)$ bit operations, where P is some polynomial.

In our case, since we are using binary representation, $\lg x$ indicates the bit length of x, where x is a positive integer. Further, the information content of the two values $\{-1, 1\}$ is represented by 1 bit as -1 can be coded instead by 0.

Definition 3. Let $\langle a, b, c\rangle$ represent the *binary quadratic form* $f(x, y) = ax^2 + bxy + cy^2$ with *discriminant* $D = b^2 - 4ac$. A binary quadratic form is *positive definite* when $f(x, y)$ is always positive for real $(x, y) \neq (0, 0)$, and this is equivalent to the discriminant D being negative and a being positive [11, §1.2.4]. We restrict x, y, a, b and c to integers only.

A form is *primitive* if a, b and c have no common factor. A *reduced positive definite quadratic form* is one with $|b| \leq a \leq c$, where a, b and c are relatively prime [12, Chapter 2].

From here on out: "binary quadratic forms" specifically indicate positive definite binary quadratic forms only. While many of the results for positive definite binary quadratic forms can be generalized to indefinite binary quadratic forms ($D > 0$) as well, we make no attempt to give these generalizations, nor their history.

Lagrange proved there are only finitely many reduced binary quadratic forms for a given discriminant and constructed a composition operation on binary quadratic forms. Gauss refined Lagrange's composition operation and discovered that the set of reduced forms for a fixed discriminant using this more refined operation form a finite group, termed the *(form) class group*. (Note, Gauss's group composition law is admissible only for reduced forms.)

Definition 4. The *elements of the (form) class group* with discriminant $D < 0$ are the set of $\mathrm{SL}_2(\mathbb{Z})$-equivalence classes of primitive positive definite binary quadratic forms of discriminant D. Each equivalence class has a unique reduced quadratic form.

Fact 1. *A positive definite binary quadratic form with discriminant $D < 0$ can be transformed into a unique reduced quadratic form in polynomial time in* $\log |D|$.

This class group can be used to construct Diffie-Hellman key exchange type cryptographic systems [11, Chapter 12]. Here we use binary quadratic forms in a completely different way, with no reliance on the class group, and construct a bit level probabilistic public key cryptographic system. However, we do need certain theorems concerning the order of the class group to analyze the run time and security of this new cryptosystem.

Definition 5. For $D < 0$, the *class number* $h(D)$ is the number of reduced (positive definite) binary quadratic forms with discriminant D.

The class number is equal to the order of the class group. The *(ideal) class group* can also be defined as the set of equivalence classes of fractional ideals in the number field $\mathbb{Q}\left(\sqrt{D}\right)$. The ideal class group and the form class group are isomorphic. The ideal class group can be defined for any number field K, not just imaginary quadratic forms. In this paper we will just be using the form class group.

Definition 6. [13, Definition 5.1.2] A discriminant $D = b^2 - 4ac \in \mathbb{Z}$ is a *fundamental discriminant* if $D \equiv 1 \pmod 4$ and square free, or $D \equiv 0 \pmod 4$, $D/4$ is square free and $D/4 \equiv 2$ or $3 \pmod 4$.

We will only be dealing with negative discriminants.

Theorem 2. *[13, Propositions 5.3.1, 5.3.12 and page 233] We have $D = D_0 f^2$, where D_0 is a fundamental discriminant. Then*

$$\frac{h(D)}{w(D)} = \frac{h(D_0)}{w(D_0)} f \prod_{p\,|\,f} \left(1 - \frac{\left(\frac{D_0}{p}\right)}{p}\right),$$

$$\textit{where } w(D) = \begin{cases} 2, & \textit{if } D < -4 \\ 4, & \textit{if } D = -4 \\ 6, & \textit{if } D = -3, \end{cases}$$

and $\left(\frac{D_0}{p}\right)$ *is the Jacobi symbol.*[1]

The above theorem tells us that the class number of a non fundamental discriminant is a known multiple of the class number of a fundamental discriminant. Consequently, for $n \equiv 1 \pmod 4$, $n > 1$, and n square free, the number $D_0 = -8n$ is a fundamental discriminant and

$$h(D) = \begin{cases} 2h(8n), & \text{for } x^2 + 8ny^2,\ D = -4 \cdot 8n = 2^2 D_0 \\ 4h(8n), & \text{for } x^2 + 32ny^2,\ D = -4 \cdot 32n = 4^2 D_0. \end{cases} \tag{1}$$

The class numbers $h(D)$ for $D = -4 \cdot 8n$ and $D = -4 \cdot 32n$ will be used later to analyze the security of this cryptosystem.

Theorem 3 (Dirichlet's Analytic Class Number Formula)**.** *For* $D_0 < 0$*,* D_0 *a fundamental discriminant,*

$$h(D_0) = \frac{w(D_0)\sqrt{|D_0|}}{2\pi} L(1, \chi_{D_0}),$$

where $\chi_{D_0}(p) = \left(\frac{D_0}{p}\right)$*, and*

$$L(s, \chi_{D_0}) = \sum_{n=1}^{\infty} \frac{\chi_{D_0}(n)}{n^s} = \prod_{p} \left(\frac{p^s}{p^s - \chi_{D_0}(p)} \right),$$

is the Dirichlet L-function, $s \in \mathbb{C}$*, and* $Re(s) > 0$*.*

Thus, for $D_0 < -4,\ h(D_0) = \dfrac{\sqrt{|D_0|}}{\pi} L(1, \chi_{D_0})$.

Theorem 4 (Littlewood [28])**.** *For* $D < 0$*, assuming the Generalized Riemann Hypothesis (GRH) for the L-functions, (meaning all the critical strip zeroes of* $L(1, \chi_D)$ *are on the vertical line* $t = 1/2$*), as* $|D| \to \infty$

$$\frac{1 + o(1)}{(12/\pi^2) \log\log |D|} < L(1, \chi_D) < (1 + o(1)) 2e^{\lambda} \log\log |D|,$$

where $\lambda = 0.5772\cdots$ *is Euler's constant.*[2]

The $o(1)$ term, which by definition means a function that converges to zero, is not specified, so we really don't know what values of $o(1)$ go with what values of D. However by applying Abel's summation formula, for $D_0 < -4$, D_0 a fundamental discriminant, and $\chi_{D_0}(n) = \left(\frac{D_0}{n}\right)$, one can show

$$L(1, \chi_{D_0}) = \sum_{n=1}^{\infty} \frac{\chi_{D_0}(n)}{n} < \log |D_0|. \tag{2}$$

This bound is considerably less sharp as $|D_0| \to \infty$ than Littlewood's, but has the advantage of being unconditional [13, exercise 5.27].

[1] Note, $\left(\frac{D}{p}\right) = 0$, when $\gcd(D, p) > 0$.

[2] These bounds also hold true more generally when χ is a non-principal Dirichlet character to modulus $|D|$.

Corollary 5. *For $n \equiv 1 \ (mod\ 4)$, $n > 1$, and square free, for discriminants $D = -4 \cdot 8n$ and $D = -4 \cdot 32n$, corresponding to the two quadratic forms: $x^2 + 4^t \cdot 2ny^2$, with $t = 1$ and $t = 2$, respectively, then*

$$h(D) < \frac{4 \cdot 2^t \sqrt{2n}}{\pi}(\log 8n).$$

Proof. Apply Equations 1 and 2. □

Definition 7. $\pi(z)$ counts all primes $\leq z$.

Theorem 6 (Prime Number Theorem (PNT))**.**

$$\pi(z) \sim \int_2^z \frac{dt}{\log t} \sim \frac{z}{\log z}.$$

Theorem 7 (Prime Number Theorem Assuming the Riemann Hypothesis [PNT+RH])**.** [3]

$$\pi(z) = \int_2^z \frac{dt}{\log t} + O\left(\sqrt{z}\log z\right).$$

2. Description of Quadratic Form Cryptosystem

2.1 Key Generation

The private key is composed of (p_1, p_2, a, b), where p_1 and p_2 are large random primes congruent to 1 modulo 4 and $a^2 + b^2 = p_1 p_2 = n$. It does not matter which of the four possible values of a and b are used. (By convention b is chosen to be the even value.)

The public key is n. For maximum security p_1 and p_2 should have close to the same number of bits. The security of the system depends on $\lg n$, the number of bits in n, and is comparable to RSA, as both cryptosystems, practically speaking, depend on the difficulty of factoring the composite public key n.

2.1.1 Public Key Generation

For n with a fixed bit length, we observe that by Dirichlet's analytic class number formula (Theorem 3), the high order bits of the class number $h(-8n)$ are controlled by the L-function. As we will show shortly in §3.3.1 and §3.3.2, the probability of finding suitable primes for encryption is faster when the class number $h(-8n)$ is smaller. We can get a smaller class number by choosing n such that $\chi_D(p) = -1$, for the first few primes p greater than 2. (Since $D = -8n, \chi_D(2) = 0$.) Thus,

p	QNR's (mod p)	n (mod p)	$-8n$ (mod p)
3	2	2	2
5	2, 3	1, 4	2, 3
7	3, 5, 6	1, 2, 4	6, 5, 3.

[3] Theorem 8.3.3 in [1]; this reference also gives the history of this theorem.

So, for example, when $n \equiv 11, 29, 44, 71, 74, 86 \pmod{3 \cdot 5 \cdot 7}$, we will get a smaller than average (for fixed bit length n) class number. Table 1 below gives some computational data on estimating the size reduction.

We want an n such that $\left(\frac{-8n}{p}\right) = -1$ for the first t odd primes, we can ignore 2 as $\left(\frac{-8n}{2}\right) = 0$. We see by Table 1 that when choosing an n that is optimal for the first 32 odd primes, the estimated worst case (high) and best (low) case do not differ by much, and the average (0.28) is very close to the best case (0.24). Optimizing for the first 32 odd primes is likely too time consuming, one would have to try 2^{32} different n on average to get an n such that for the first 32 odd primes p, $\left(\frac{-8n}{p}\right) = -1$.

Likely optimizing for the first $t = 16$ to $t = 20$ odd primes using a hybrid approach is enough. For the first $k \approx \lfloor t/2 \rfloor$ odd primes use the Chinese remainder theorem and generate the qualifying set C of congruences for $-8n$, (in the same fashion as in the above example with primes 3, 5 and 7). Next generate one random prime p_1, then generate a candidate second number c_2 that meets the congruence conditions of C: meaning $-8p_1c_2 \equiv C \pmod{3 \cdot 5 \cdot 7 \cdot \ldots \cdot p_k}$. Next, test that $-8n = -8p_1c_2$ has a Jacobi symbol value of -1 for each of the remaining primes p_j, for $k + 1 \leq j \leq t$. If $\left(\frac{-8n}{p_j}\right) = -1$, then check if c_2 is prime, and if not, repeat the process until c_2 is prime.

The payoff is significant, an optimal key for encryption only needs to generated once, and thereafter the encryption is many times faster. Looking at the average value in Table 1 for $t = 16$, we see that it is close to $1/3$ the size of the average value when $t = 1$, the case with no attempt at optimizing. Thus for $t = 16$, on average, encryption would be $1.11/.34 \approx 3\frac{1}{4}$ times faster. Significantly, the worse case performance is $4.25/.43 \approx 9.9$ or almost 10 times better, which is close to the maximum possible of ≈ 10.98.

Table 1: L-function estimates. Column two is the maximum ratio $\approx \text{High}_1 / \text{High}_t$. The right most column is based on 100,000 trials of $\prod_{i=t+1}^{2048} \left(\frac{p_i}{p_i + r}\right)$, where r is randomly ±1.

t	$\prod_{i=2}^{t} \left(\frac{p_i + 1}{p_i - 1}\right)$	$\prod_{i=2}^{t} \left(\frac{p_i}{p_i + 1}\right)$	Random from p_{t+1} to p_{2048} [Low, High, Avg]
1		1.00	[0.31, 4.25, 1.11]
2	= 2	0.75	[0.30, 2.06, 0.78]
4	= 4	0.55	[0.30, 1.04, 0.56]
8	= 7	0.42	[0.27, 0.62, 0.42]
16	≈ 10.98	0.33	[0.26, 0.43, 0.34]
20	≈ 12.44	0.31	[0.25, 0.39, 0.31]
24	≈ 13.74	0.30	[0.25, 0.37, 0.30]
28	≈ 14.86	0.29	[0.24, 0.34, 0.29]
32	≈ 15.89	0.28	[0.24, 0.33, 0.28]

2.1.2 Private Key Construction

To compute (a, b) from (p_1, p_2) is a multi-step process. First, $r_i = \sqrt{-1}$ modulo p_1 and p_2 are computed using Tonelli's or Cipolla's algorithm. (For an odd prime p, Cipolla's algorithm has run time of $O((\lg p)^3)$ bit operations compared to $O((\lg p)^4)$ for Tonelli [1, §7.1-7.2].) Next, using the Chinese remainder theorem,

$\sqrt{-1}$ modulo n is computed as

$$w \equiv \sqrt{-1} \equiv \pm r_1 p_2 v_2 \pm r_2 p_1 v_1 \pmod{n}, \tag{3}$$

where $v_2 = p_2^{-1} \pmod{p_1}$, and $v_1 = p_1^{-1} \pmod{p_2}$. To verify Equation 3 is correct we observe that

$$w \equiv r_1 p_2 \cdot p_2^{-1} \equiv r_1 \equiv \sqrt{-1} \pmod{p_1},$$
$$w \equiv r_2 p_1 \cdot p_1^{-1} \equiv r_2 \equiv \sqrt{-1} \pmod{p_2}.$$

Any of the four values of w can be used. One can efficiently compute a and b by exploiting the fact that the ring of Gaussian integers $\mathbb{Z}[i]$ is an Euclidean domain and Euclidean division can be performed in this ring. We observe that $a + bi = \gcd(r + i, n)$, for $r \equiv \sqrt{-1} \pmod{n}$. For example, for $n = 377 = 13 \cdot 29$, we have $r = \sqrt{-1} \equiv -70 \pmod{377}$ and $\gcd(70 + i, 377) = 16 + 11i$ and $16^2 + 11^2 = 256 + 121 = 377$.

If one wishes to only operate in $\mathbb{Z}$, the ring of rational integers, one can instead use the 1908 algorithm due to the Italian mathematician G. Cornacchia[4] [15, pages 61-66], [20, 673–674], [13, pages 34–36], [34] and [3].[5] We include a brief description here of Cornacchia's algorithm as we will also be referring to it later in §4.1.3.

Algorithm 1: Cornacchia's Algorithm.

```
Input  : A number $n$, an integer $m$, where $0 < m < n$, $\gcd(n, m) = 1$, $m \neq \square$, and $-m$ is a quadratic
         residue of $n$, and $r \equiv \sqrt{-m} \pmod{n}$.
Output : A solution $(x, y)$ to $n = x^2 + my^2$, if there is one.
$y \longleftarrow \sqrt{n/m}$
if $y \in \mathbb{Z}$ then return $(0, y)$
$x_0 \longleftarrow r$
if $x_0 \leq (n-1)/2$ then $x_0 = n - x_0$
$i \longleftarrow 0$
$x_1 \longleftarrow n \pmod{x_0}$
while $x_i > \sqrt{n}$ do
    $i \longleftarrow i + 1$
    $x_{i+1} \longleftarrow x_{i-1} \pmod{x_i}$
$x \longleftarrow x_i$
$y \longleftarrow \sqrt{(n - x^2)/m}$
if $y \in \mathbb{Z}$ then return $(x, y)$ else return Fail
```

Example 1. Cornacchia gave two examples of finding (x, y). In the first example: $x^2 + y^2 = 89$, and $x_0 = 55$, $x_1 = 34$, $x_2 = 21$, $x_3 = 13$ and $x_4 = 8$, to give $8^2 + 5^2 = 89$. In the second example: $x^2 + 13y^2 = 3221$, and $x_0 = 2723$, $x_1 = 498$, $x_2 = 233$, and $x_3 = 32$, for a solution of $32^2 + 13 \cdot 13^2 = 3221$.

[4] Specfically, Giuseppe Antonio Cornacchia. I am unable to determine Cornacchia's birth and death years. Possibly Henry John Stephen Smith (1826–1883) discovered this algorithm prior to Cornacchia in 1855 [37].

[5] This algorithm as given here is slightly different than in Cornacchia's 1908 paper.

2.2 Encryption

Let s be a single bit of information represented by $\{-1, 1\}$. To encrypt s, Alice constructs two primes q_1, q_2 represented by two quadratic forms, to be described shortly, such that

$$\left(\frac{(a+bi)(1+i)}{q_1q_2}\right) = s,$$

where $i \equiv \sqrt{-1} \pmod{q_1q_2}$.

We will show this can be done without Alice knowing a or b, it is only necessary that Alice know n. Next, Alice transmits to Bob: $e = q_1q_2$ and $r \equiv \sqrt{-1} \pmod{e}$.

$$\begin{tikzcd}
Alice \arrow[rr] & & Bob \\
 & e,\ r & \\
q_1q_2 \arrow[ur, mapsto, bend right] & & \\
\sqrt{-1} \pmod{e} \arrow[uur, mapsto, bend right] & &
\end{tikzcd}$$

Transmission of encrypted message bit.

The primes q_1 and q_2 are constructed by Alice as follows. Let x_1, x_2, y_1 and y_2 be odd integers less than bound B, where B will be determined later. (Basically, the bound B must be large enough that $(x_1x_2)^2 \gg (q_1q_2 \pmod{n})$ with high probability.) When

$$s = \begin{cases} -1, & \text{then } q_1 = x_1^2 + 8ny_1^2 \text{ and } q_2 = x_2^2 + 32ny_2^2, \\ +1, & \text{then } q_1 = x_1^2 + 8ny_1^2 \text{ and } q_2 = x_2^2 + 8ny_2^2. \end{cases}$$

Alternatively, the following scenario also works. When

$$s = \begin{cases} -1, & \text{then } q_1 = x_1^2 + 8ny_1^2 \text{ and } q_2 = x_2^2 + 32ny_2^2, \\ +1, & \text{then } q_1 = x_1^2 + 32ny_1^2 \text{ and } q_2 = x_2^2 + 32ny_2^2. \end{cases}$$

In either scenario, anytime $x_2^2 + 32ny_2^2$ is used, then the bound on y is set to 1/2 of the bound on y when using $x_2^2 + 8ny_2^2$. This ensures that the distribution of the magnitudes of the primes generated by $x_2^2 + 8ny_2^2$ and $x_2^2 + 32ny_2^2$ are the same. Consider encryption using the first scenario. Without adjustment those e that have both prime factors represented by $x_2^2 + 8ny_2^2$ would on average be smaller than those e that have one prime factor represented by $x_2^2 + 8ny_2^2$, thus the more bits encrypted the more likely some of those e that contain a prime represented by $x_2^2 + 32ny_2^2$ would be large enough to be identified with high probability and those message bits would be compromised. If the bound B is public, then even with just one message bit the expected probability of this happening is 1/4 and the plaintext message bit could definitively be determined to equal 1. Likewise for the alternate encryption scenario.

Theorems 8 and 9 makes it possible for Alice to construct q_1 and q_2 such that

$$\left(\frac{(a+bi)(1+i)}{e}\right) = s.$$

These theorems can be proven using rational quartic reciprocity.[6]

Theorem 8. *For a prime $q = x^2 + 8ny^2$, with y and n both odd, $n = a^2 + b^2$, b even, and $i \equiv \sqrt{-1}$ (mod q), then*

$$\left(\frac{a+bi}{q}\right)\left(\frac{1 \pm i}{q}\right) = -1.$$

Theorem 9. *For a prime $q = x^2 + 32ny^2$, with y and n both odd, $n = a^2 + b^2$, b even, and $i \equiv \sqrt{-1}$ (mod q), then*

$$\left(\frac{a+bi}{q}\right)\left(\frac{1 \pm i}{q}\right) = 1.$$

We observe that for either encryption scenario $i \equiv \sqrt{-1}$ (mod $q_1 q_2$), so $i \equiv \sqrt{-1}$ (mod q_i), and when

$$\begin{aligned} &s = -1, \text{ then} \\ &\qquad \left(\frac{(a+bi)(1+i)}{q_1}\right) = -1, \\ &\qquad \left(\frac{(a+bi)(1+i)}{q_2}\right) = +1, \text{ and} \\ &\qquad \left(\frac{(a+bi)(1+i)}{q_1 q_2}\right) = -1; \end{aligned}$$

and when

$$\begin{aligned} &s = +1, \text{ then} \\ &\qquad \left(\frac{(a+bi)(1+i)}{q_1}\right) = \left(\frac{(a+bi)(1+i)}{q_2}\right) = \pm 1, \text{ and} \\ &\qquad \left(\frac{(a+bi)(1+i)}{q_1 q_2}\right) = +1. \end{aligned}$$

[6] These theorems can also be proven using a theorem of Z. H. Sun in [39, page 1298 and §4] deduced from the law of quadratic reciprocity. Instead of $d = a^2 + b^2$, we use $d' \leftarrow 2d$, with $d' = a'^2 + b'^2$, $a' \leftarrow a + b$ and $b' \leftarrow b - a$, and we have $q = x^2 + d'y^2$, with y even. From Sun's theorem: $\left(\frac{t \pm \sqrt{d'}}{q}\right) = (-1)^{y/2}$, where $t = a'$ or $t = b'$, and the identity $(a' + \sqrt{d'} + b'i)^2 = 2(a' + \sqrt{d'})(a' + b'i)$ we get Theorems 8 and 9. Our theorems and proofs using quartic reciprocity were discovered independently and are not based on Z. H. Sun's work.

The value $i = \sqrt{-1} \pmod{q_1 q_2}$ is computed in the same manner as in the private key construction step, by first computing $r_1 \equiv \sqrt{-1} \pmod{q_1}$ and $r_2 \equiv \sqrt{-1} \pmod{q_1}$ and then using the Chinese remainder theorem to compute

$$r \equiv i \equiv \sqrt{-1} \equiv \pm r_1 q_2 v_2 \pm r_2 q_1 v_1 \pmod{e}, \tag{4}$$

where $v_2 = q_2^{-1} \pmod{q_1}$, and $v_1 = q_1^{-1} \pmod{q_2}$.

2.2.1 Variable Strength Encryption

For any given bit by adjusting $\lg B$ down or otherwise reducing the entropy of the random (x_1, y_1) used in the construction of q_1, then the sender Alice can encrypt a bit that is less secure than the overall key n. This might be useful in those situations where the sender wants to ensure that the information can be decrypted and made public by an approximate future date. Another application might be for bitcoin type cryptocurrencies where the computational cost to mine a bit needs to be adjustable.

2.2.2 Additional Senders

Additional senders: Alicia, Alisha and so on, can be added. For example, for any bit, Alicia can generate a prime q_3 of the form $x^2 + 32ny^2$, compute $r_a \equiv \sqrt{-1} \pmod{q_3}$ and then using the Chinese remainder theorem, compute $r_b \equiv \sqrt{-1} \pmod{q_3 e}$ from r_a and r, and finally send r_b and $q_3 e$ to Bob.

It is also possible to embed user specific information into the middle order bits of x_3 used to generate q_3. The specific bit positions used would need to be predefined and known to Bob. Since $\lg B$ in a certain sense indicates the amount of entropy present in (x_i, y_i), the Bound B should be increased to $2^u B$, where u is the number of bits used to encode the user specific information as $\lg(2^u B) = u + \lg B$.

There is considerable message expansion with this process, but the decryption time is still pretty good as it is only a constant times the square of the combined bit length of e and any additional primes by the extra senders. The maximum total ciphertext bit length in this example is $t = 4(1 + 4 + \lg n + 2\lg B)$ bits with only one sender. With k senders total, the ciphertext bit length would be $t = (2(k-1) + 4)(1 + 4 + \lg n + 2\lg B) = (2k + 2)(5 + \lg n + 2\lg B)$.

Finally, Alicia or Alisha (and so on) can verify they are the senders by either revealing q_3 or q_4, respectively. They can, if they wish, keep their primes secret and instead use the challenge response verification process described in §4.3.

2.3 Decryption

Since $r \equiv \sqrt{-1} \pmod{e}$ is transmitted to Bob along with e, all Bob has to do to decrypt is to compute the Jacobi symbol

$$\left(\frac{(a + br)(1 + r)}{e}\right) = \left(\frac{(a + rb)(1 + r)}{e}\right) = \left(\frac{(a - b + r(a + b)}{e}\right) = s.$$

The decryption is very fast as only one modular multiplication and one Jacobi symbol computation is required, no power exponentiation is required as in the case with RSA. The run time for decryption per bit using standard algorithms is $O((\lg n^2))$ bit operations.

3. Run Time

The run times given here are asymptotic and are for comparison purposes. The actual run times will depend on the implementation and how optimized the software and hardware is. The unspecified constant implied in the big O notation, can make all the difference in practice. It can be the case that two algorithms with the same big O running time actually perform radically differently in practice.

3.1 Key Generation Run Time

The run time to generate the public key n is comparable to RSA, as in both systems two large primes are generated. In our system there is the additional requirement that both primes p_1 and p_2 are congruent to 1 modulo 4. By the prime number theorem the expected number of random trials to find a prime is $O(\log(2^b))$, where b is the bit length of the prime. Thus number of bit operations to generate the public key is $O((\lg n)^4)$ as standard prime testing algorithms such as Miller-Rabin require $O((\lg n)^3)$ bit operations.

To generate the private key requires computing $r \equiv \sqrt{-1} \pmod{n}$. This requires computing $\sqrt{-1}$ over $\mathbb{F}_{p_i}$, for primes p_1 and p_2 and then using the Chinese remainder theorem. The number of bit operations to compute the square root is $O((\lg n)^3)$. The final step is to compute (a, b), where $a^2 + b^2 = n$. The run time for Cornacchia's algorithm is $O((\lg n)^2)$, and consequently the run time for key generation is dominated by the search for the two large random primes.

3.2 Decryption Run Time

The decryption run time is dominated by one multiplication modulo e and one Jacobi symbol computation modulo e. Since $B \ll n$, the run time to decrypt one encrypted plaintext bit is $O((\lg n)^2)$ bit operations [1, pages 113-114], [29, Table 2.5].

3.3 Encryption Run Time

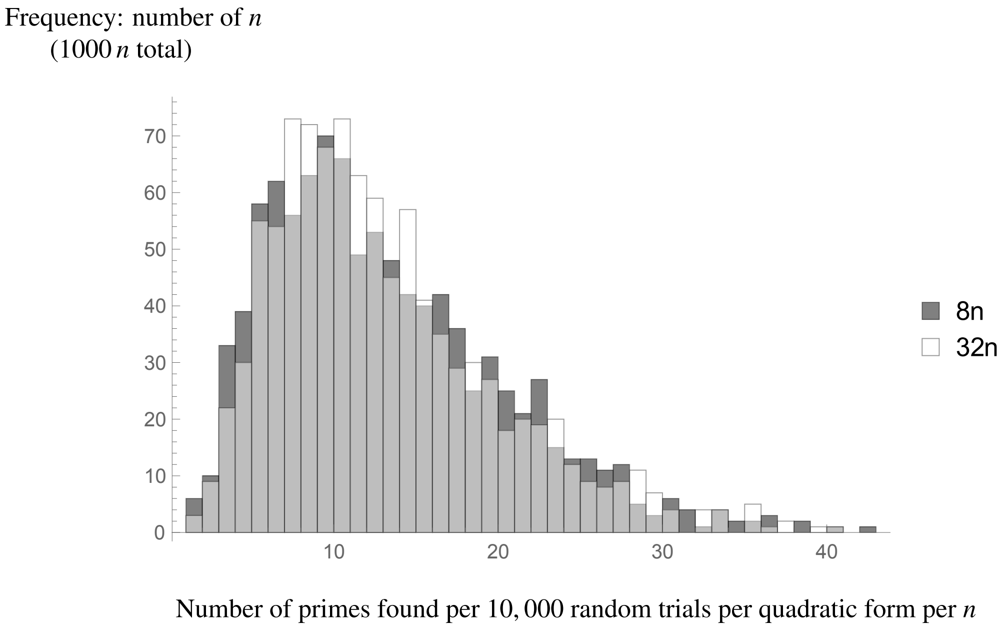
The expected run time to encrypt one message bit is $O((\lg n^4))$. Namely, $2(\log m)/2 = \log m$ primality tests, where $m = x^2 + 8ny^2$ or $m = x^2 + 32ny^2$.

To encrypt one bit of information requires up to $4(\lg n + 2\lg B + 5)$ bits. Using a $B \approx 2^{128}$, we have at most $4(\lg n + 261)$ bits of ciphertext per plaintext bit, not including transmission protocol overhead.

To encrypt the message bit s requires finding primes q_1 and q_2 that are represented by the quadratic forms $x^2 + 8ny^2$ or $x^2 + 32ny^2$. Empirical tests show that the expected probability of $z = x^2 + 2ny^2$ being prime, where x is odd and $2 \| y$ or $4 \| y$, is approximately $2/\log z$, but there is considerable variance. This is what one would expect by the prime number theorem (PNT). This probability is quite different from the probability that a random prime p can be represented by the binary quadratic form $x^2 + 2ny^2$.

We performed 1000 trials of n, with $n = p_1 p_2$, and p_i random primes both congruent to 1 modulo 4, each with a bit length of 1024. Thus, each n was a $2047 - 2048$ bit number. For each individual n we randomly selected $10,000$ (x, y) ordered pairs, with x and y both odd, $0 < x < \lfloor y\sqrt{8n} \rfloor$, and $0 < y < B$, where $B = 2^{128}$. (See §3.4 on choosing the bound B.) For each (x, y) we computed $z_8 = f_8(x, y) = x^2 + 8ny^2$ and counted the number of prime hits on z_8. We computed a histogram of the 1000 trials of n to see the frequency of primes encountered per $10,000$ random z_8 per random n. We repeated this for the $x^2 + 32ny^2$ case using the same n, but different random ordered (x, y) pairs. In the case of $z_{32} = x^2 + 32ny^2$, the bound on y was reduced from B to $B/2$ so that z_8 and z_{32} would be comparable in size. The histograms are given in Figure 1.

Figure 1: Histogram for primes represented by quadratic forms $x^2 + 8ny^2$ and $x^2 + 32ny^2$.



3.3.1 A Brief Class Field Tangent

To optimize the encryption running time and avoid worst case situations we will need to briefly segue to class field theory. Specifically, we will need to understand the computational run time required to find primes represented by each of the two different quadratic form types used. In addition, by using class field theory we can improve on the above $2/\log z$ probability estimate to find an encryption prime.[7]

Definition 8 (Dirchichlet analytic density [38])**.** A set S of prime numbers has Dirichlet or analytic density $\delta(S)$, if for primes p

$$\frac{\sum\limits_{p \in S} p^{-s}}{\sum\limits_{p \in \mathbb{Z}} p^{-s}} \to \delta \qquad \text{for } s \downarrow 1.$$

Definition 9 (Natural Density [38])**.** A set S of prime numbers has natural density $\delta_N(S)$, if

$$\frac{\#\{p \leq x : p \in S\}}{\#\{p \leq x : p \in \mathbb{Z}\}} \to \delta \qquad \text{for } x \to \infty.$$

[7] Meaning the number of trials to find an encryption prime is heuristically estimated to be asymptotically equal to $(\log z)/2$.

If a set of primes S has a natural density δ, then the Dirichlet analytic density is the same. However, the converse is false [38], [36, page 76], [35, page 126].

Theorems 10 and 12 below are usually stated using Dirichlet (analytic) density instead of Natural density. Natural density is easier to work with for our purposes. While the Natural density version of Theorem 12 is a known result, and it is not new that natural density can be used, it is not stated explicitly in commonly used class field references such as [16] and so we will state the main theorems needed to show the truth of Theorem 13, the Natural density version of Theorem 12.

Theorem 10 (Frobenius (1880)). [8] *Let $(f(x)$ be a monic degree n polynomial with integer coefficients, with discriminant $\Delta \neq 0$, with Galois group G. The Dirichlet (natural) density* [9] *δ of the set of primes p for which the polynomial f has a given decomposition (i.e., partition of n) into irreducible factors of degree $n_1, n_2, \cdots, n_t$ modulo p exists and*

$$\delta = \frac{|\sigma|}{|G|},$$

where $|\sigma|$ equals the number of $\sigma \in G$ with permutation cycle pattern $(n_1)(n_2)\cdots(n_t)$.

The decomposition pattern we need is when f splits completely into linear factors modulo p. In this case $t = n$ and the partition consists of n_i all identically equal to 1, which corresponds to the identity element in G, namely, the permutation cycle $(1)(2)\cdots(n)$. The Dirichlet and natural density are both equal to $1/|G|$ in this case.

We are interested in f splitting completely, as when f defines a number field K then the complete splitting of f modulo p means that p splits completely in K. (There are only a finite number of prime p exceptions.)

Theorem 11. *(Dedekind factorization theorem [14, page 35, Theorem 3.3.5].) Let $K = \mathbb{Q}(\alpha)$ and $f(x)$ be the unique irreducible monic polynomial of lowest degree with integer coefficients such that $f(\alpha) = 0$. Then for any rational prime not dividing $[O_K : \mathbb{Z}[\alpha]]$ the factorization pattern of $f(x)$ modulo p is the same as the factorization pattern of the ideal (p) into prime ideals of O_K. We have, specifically,*

$$f(x) \equiv f_1(x)^{e_1} \cdots f_g(x)^{e_g} \ (mod\ p), \ \ and$$

$$(p) = \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_g^{e_g}.$$

Assuming the General Riemann Hypothesis, then Lagarias and Odylzko's Theorem 1.1 [26] implies that the number of integer primes below z for which f splits completely is

$$\frac{1}{|G|}\left(\int_2^z \frac{dt}{\log t} + O\left(\sqrt{z}(n\log z + \log|\Delta|)\right)\right).$$

However, we would like an unconditional bound as well.

Theorem 12. *[16, Simplified version of Theorem 9.12] Let S be the set of primes represented by $x^2 + my^2$, $m \in \mathbb{Z}_{>0}$, then the Dirichlet density exists for S and equals $\dfrac{1}{2h(-4m)}$.*

[8] Published in 1896. See the expository paper by Lenstra and Stevenhagen [38] for an accessible account of this theorem, as well as the more general density theorem of Chebotarëv.

[9] According to [38], Frobenius proved this theorem using the concept of Dirichlet density. However, the Frobenius density theorem is also true using natural density, but this is harder to prove and was done later with techniques developed by E. Hecke (1917).

This theorem is stated using Dirichlet density. However, it is also holds true using Natural density, because Theorem 10 is not only true for Dirichlet density, but for Natural density as well. For a prime p to be represented by $x^2 + my^2$ is equivalent to p splitting completely in the ring class field L. The order of G, the Galois group of L/K, where $K = \mathbb{Q}(\sqrt{-m})$ is equal to the class number $h(-4m) = O(\sqrt{m})$ and by Theorem 10 the Natural density of primes that completely split in L is $1/(2|G|)$. The 2 in the denominator is due to $[L : \mathbb{Q}] = [L : K] \cdot [K : \mathbb{Q}] = 2|G|$.

Theorem 13. *[16, Natural density version of Theorem 9.12] Let S be the set of primes represented by $x^2 + my^2$, $m \in \mathbb{Z}_{>0}$, then the Natural density exists for S and equals $\dfrac{1}{2h(-4m)}$.*

This theorem along with Corollary 5 shows that the probability that a random prime can be represented by the quadratic form $x^2 + my^2$, for m sufficiently large, is $O(1/\sqrt{m})$, which means it is exponentially small. In our case we interested in the cases where $m = 8n$ and $m = 32n$, with y, odd. This means we have to subtract out the cases where y is even, meaning we subtract out the cases where $m = 32n$ or $128n$, respectively. Thus,

$$\text{Natural density } = \begin{cases} \dfrac{1}{2h(-32n)} - \dfrac{1}{2h(-128n)} = \dfrac{1}{8h(-8n)}, & \text{for } 8n \text{ case,} \\ \dfrac{1}{2h(-128n)} - \dfrac{1}{2h(-512n)} = \dfrac{1}{16h(-8n)}, & \text{for } 32n \text{ case.} \end{cases}$$

Theorem 14. *The probability r that a random prime can be represented by the quadratic form $x^2 + my^2$, for $m \gg 1$, is $O(1/\sqrt{m})$.*

Theorem 14 follows from Theorem 10, Corollary 5, and a classic result from class field theory first proved by the German mathematician Heinrich Martin Weber (1843–1913).[10] For the sake of simplicity we are only giving the theorem for the case where $d \equiv 0 \pmod 4$.

Definition 10 (Klein's j-invariant)**.** Klein's j-invariant or j-function[11] is the most basic modular invariant function in that every modular function of weight $k = 0$ is a rational function of j. The j-invariant is a complex (analytic) function defined on the upper half-plane $\mathcal{H} = \{\tau \in \mathbb{C}, \operatorname{Im} \tau > 0\}$, , and is periodic of period 1, where $j(\tau) = j(\tau + 1)$ and $j(\tau) = j(-1/\tau)$; in other words, a meromorphic function $\mathcal{H} \to \mathbb{C}$ invariant under the action of $\mathrm{SL}_2(\mathbb{Z})$.

Theorem 15 (Weber [1908])**.** *For p a prime, then $p = x^2 + dy^2, d > 0, d \equiv 0 \ (mod\ 4) \Leftrightarrow p$ completely splits in $\mathbb{Q}\left(\sqrt{-d}, j(\sqrt{-d})\right)$, where $j(\sqrt{-d})$ is an algebraic number of degree exactly equal to $h(-4d)$. The minimal polynomial $f(x)$ of $j(\sqrt{-d})$ over $\mathbb{Q}$ is the Hilbert ring class field (RCF) polynomial for $D = -4d$ and*

$$H_D(x) = \prod_{i=1}^{h(D)} (x - j(\tau_i)),$$

where $\tau_i = (-b_i + \sqrt{D})/2a_i$, for all the reduced binary quadratic forms $\langle a_, b_i, c_i \rangle$ of discriminant D.

[10] See volume 3 (1908) of Weber's *Lehrbuch der Algebra* [41], [14, page 4].

[11] Note, Klein's little j-invariant = $1728 \cdot J$-invariant.

The Galois group of $H_D(x)$ over $K = \mathbb{Q}(\sqrt{-d})$ is isomorphic to the class group for the imaginary quadratic order of discriminant $D = -4d$. Since p splits in $\mathbb{Q}\left(\sqrt{-d}, j(\sqrt{-d})\right)$, then by Frobenius's density theorem [12] both the Dirichlet and natural density of primes in S is $1/|G|$, where G is the Galois group of $f(x)$ over $\mathbb{Q}$ which has order $2h(-4d)$. In other words, the ring class field (RCF{-4d}) with discriminant $D = -4d$ is the splitting field of $f(x)$, and the prime p splits completely in this RCF $= \mathbb{Q}(\sqrt{-d}, j(\sqrt{-d}))$ and $[RCF\{-4d\} : \mathbb{Q}] = 2h(-4d)$.

Diagram 2 shows the subfield relationships for $D = 4 \cdot -32 \cdot 65$. The Galois group over $K = \mathbb{Q}\left(\sqrt{-d}\right)$ for the Hilbert ring class field polynomial is cyclic and isomorphic to the class group, whereas the Galois group over $\mathbb{Q}$ is a dihedral group and is only abelian when the order is less then or equal to 4.

Diagram 2: Illustrative subfields of RCF$\{-128 \cdot 65\}$, (corresponding to quadratic form: $x^2 + 32 \cdot 65y^2$), and their relative Galois groups [extension degree, group order].

$$\begin{tikzcd}
& \text{RCF}\{-128 \cdot 65\} = \mathbb{Q}\left(\sqrt{-65}, j(\sqrt{-32 \cdot 65})\right) \arrow[dr, dash, "\simeq C_2\ [2,\ 2]"] \arrow[ddl, dash, "\simeq \text{Class group (for } D = -128 \cdot 65)\ [16,\ 16]"'] & \\
& & \mathbb{Q}\left(j(\sqrt{-32 \cdot 65})\right) \arrow[ddl, dash, "\simeq \text{Dihedral group}\ [16,\ 32]"] \\
K = \mathbb{Q}\left(\sqrt{-65}\right) \arrow[dr, dash, "\simeq C_2\ [2,\ 2]"'] & & \\
& \mathbb{Q} &
\end{tikzcd}$$

3.3.2 Estimating the probability that $x^2 + 4^t \cdot 8ny^2$ is prime

The probability that $x^2 + 4^t \cdot 2ny^2$, with $t \in \{1, 2\}$ is prime is a function of the class number of $h(-8n)$. Consider the plot in Figure 4a below where we computed the class number for $D = -32n$ and plotted it against the number of primes found with 10,000 random $f(x, y) = x^2 + 32ny^2$, using the same constraints on (x, y) as above. (Note, $h(-4 \cdot 8n) = h(-4 \cdot 32n)/2$, so only the case $-32n$ was considered.)

A simple Bayesian argument can be made showing the above probability is dependent on the class number of $h(-8n)$. Consider

$$\begin{aligned}
P(z \in f(x,y) \wedge z \in \text{ PRIME}) &= P(z \in f(x,y) \,|\, z \in \text{ PRIME}) \cdot P(z \in \text{ PRIME}) \\
&= P(z \in \text{ PRIME} \,|\, z \in f(x,y)) \cdot P(z \in f(x,y)),
\end{aligned}$$

which implies

$$P(z \in \text{ PRIME} \,|\, z \in f(x,y)) = \frac{P(z \in f(x,y) \,|\, z \in PRIME) \cdot P(z \in \text{ PRIME})}{P(z \in f(x,y))}.$$

[12] One can use the more general and stronger density theorem of Chebotarëv as well.

Now in § 3.3.1 we showed that the natural density of primes that are equal to $f(x, y)$ is $\dfrac{1}{4 \cdot 2^t h(-8n)}$, where $t = 1$ in the case of $f(x, y) = f_8 = x^2 + 8ny^2$ and $t = 2$ in the case of $f(x, y) = f_{32} = x^2 + 32ny^2$, so

$$P(z \in \text{ PRIME } | \, z \in f(x, y)) \approx \frac{1}{4 \cdot 2^t h(-8n)} \cdot \frac{P(z \in \text{ PRIME })}{P(z \in f(x, y))}. \tag{5}$$

We can reformulate $f(x, y)$ as $x^2 + 2ny^2$, keep x odd and require y to be divisible by 2 or 4, but not 8. Now z is a function of x and y, with $y \in (0, 4B)$, and y randomly chosen before x, and $x \in (0, \sqrt{2ny^2})$, is also chosen randomly. Therefore, in the case of f_{32} only half as many y can be chosen compared to f_8 and we have

$$P(z \in f_{32}(x, y)) \sim \frac{P(z \in f_8(x, y))}{2}$$

and

$$P(z \in \text{ PRIME } | \, z \in f_8(x, y)) \sim P(z \in \text{ PRIME } | \, z \in f_{32}(x, y)). \tag{6}$$

We have not determined how large n has to be for the error terms (with or without assuming the GRH) to be small enough to insure that the two probabilities are statistically indistinguishable. Though empirical calculations suggest that for n the product of two 1024 bit primes the probabilities are likely indistinguishable. (See Figure 3.)

The results of the two nonparametric statistical tests: Anderson Darling (AD) and(Cramer Von Mises (CVM) for comparing two distributions, using the same data used to create Figure 3, yielded p values of 0.25 and 0.35, respectively, meaning we cannot reject the hypothesis that the two distributions are same. (Both these statistical tests assume the samples come from continuous distributions which is not exactly true here, but with large samples it is very reasonable to accept this assumption.)

The normal distribution curve fit to Figure 3 is a heuristic approximation. By the Central Limit Theorem (CLT) we know the distribution of the difference of two proportions p_1 and p_2 is approximately normally distributed if the sample sizes are large enough. The mean is $\rho_1 - \rho_2$. We are assuming the proportions ρ_1 and ρ_2 are the same and are close to $\rho = 12.5/10000$ by the Prime Number Theorem (PNT). The variance equals $\rho_1(1-\rho_1)/10000 + \rho_2(1-\rho_2)/10000 = 2\rho(\rho - 1)/10000$, with $\rho = 12.5/10000$, this works out to $24.97/10000$ or a variance of $24.97 \approx 25.0$ primes per $10,000$ trials. It is perhaps a bit surprising that $\mathcal{N}(0, 25.0)$ matches the empirical data of Figure 3 so well, as by Figure 1 we see the histograms of ρ_1 and ρ_2 are both positively skewed and neither approximates a binomial distribution. In conclusion, $\mathcal{N}(0.0, 25.0)$ is a surprisingly good approximation, given that we are computing our estimate using two asymptotic theorems without the use of any error bounds. (Unpublished data using much smaller primes, but more trials, strongly suggest the two distributions generated from the quadratic forms $8n$ and $32n$ are virtually identical.)

Table 2: Empirical frequency of primes of the form $f(x, y) = x^2 + 2 \cdot 4^t \cdot ny^2$, $t \in \{1, 2\}$, $n = pq$, $1,000$ random n. Each n, $10,000$ random ordered pairs (x, y) were generated.

Form Type	Bit Length of p and q	Standard Deviation	Min.	Max.	Mean	PNT Estimate of Mean
$8n$	1024	7.1	1	42	12.7	12.5
$32n$	1024	6.9	1	40	12.6	12.5

Figure 3: Histogram for #(primes equal to $x^2 + 8ny^2$) - #(primes equal to $x^2 + 32ny^2$). A total of 1000 n, and 10,000 random trials of (x, y) per n. Each n is a product of two random 1024 bit primes congruent to 1 modulo 4. The variance of $25.0 = 2 \cdot 12.5$ for the normal curve, with mean 0.0, that was used to fit the count difference was computed from the prime number theorem estimate of 12.5 primes per run of $10,000$ random (x, y). This normal curve fits the empirical data quite well, as the empirical sample mean and variance are 0.048 and 25.75, respectively. (The bounds on x and y were computed per §3.3.)

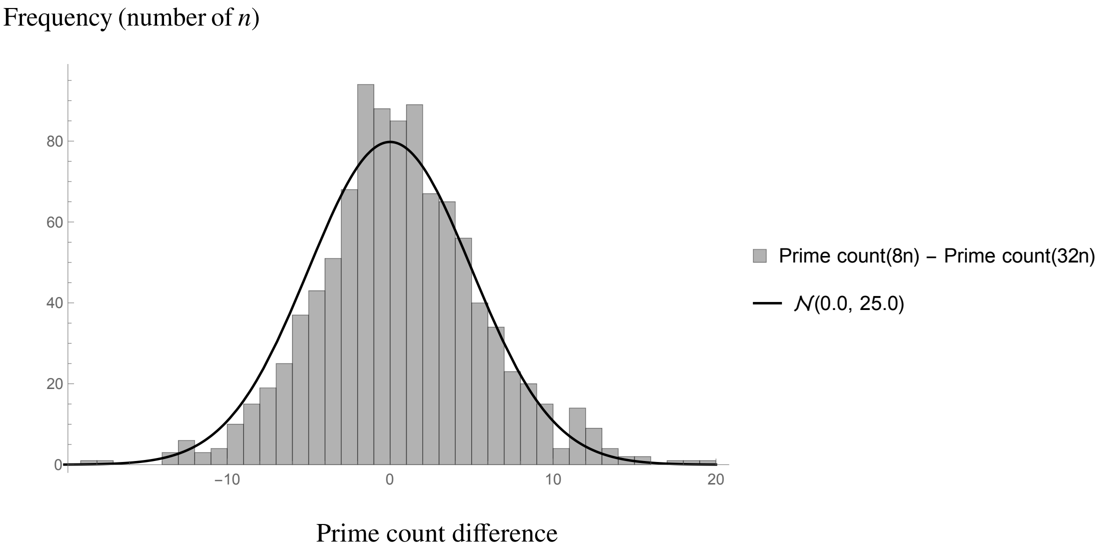


Figure 4a: Scatter plot of class number $h(-4 \cdot 32n)$ versus number of primes represented by quadratic form $x^2 + 32ny^2$, $n = pq$, p and q random primes $\equiv 1 \pmod 4$, 16 bits in length. Same data used for Figure 4b.

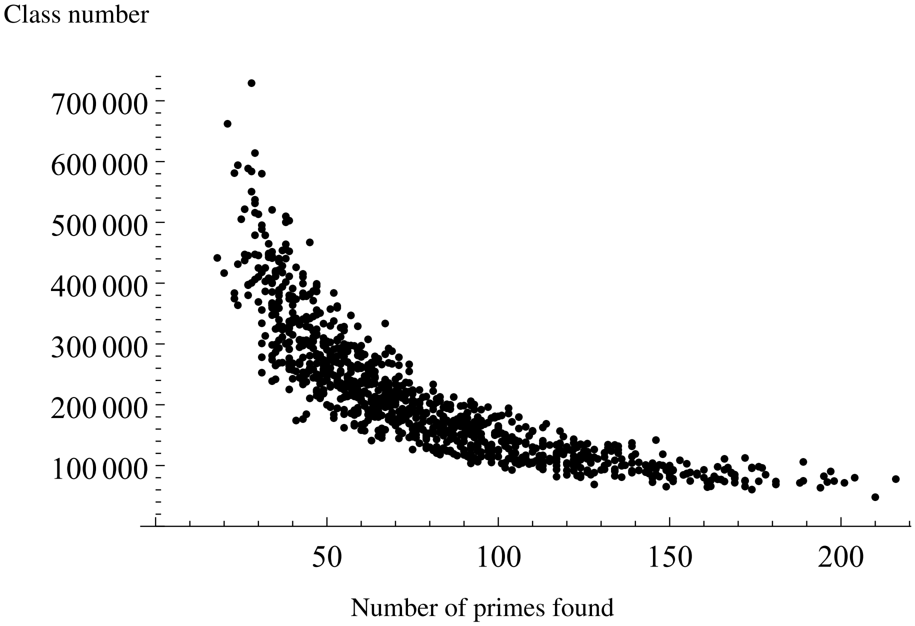
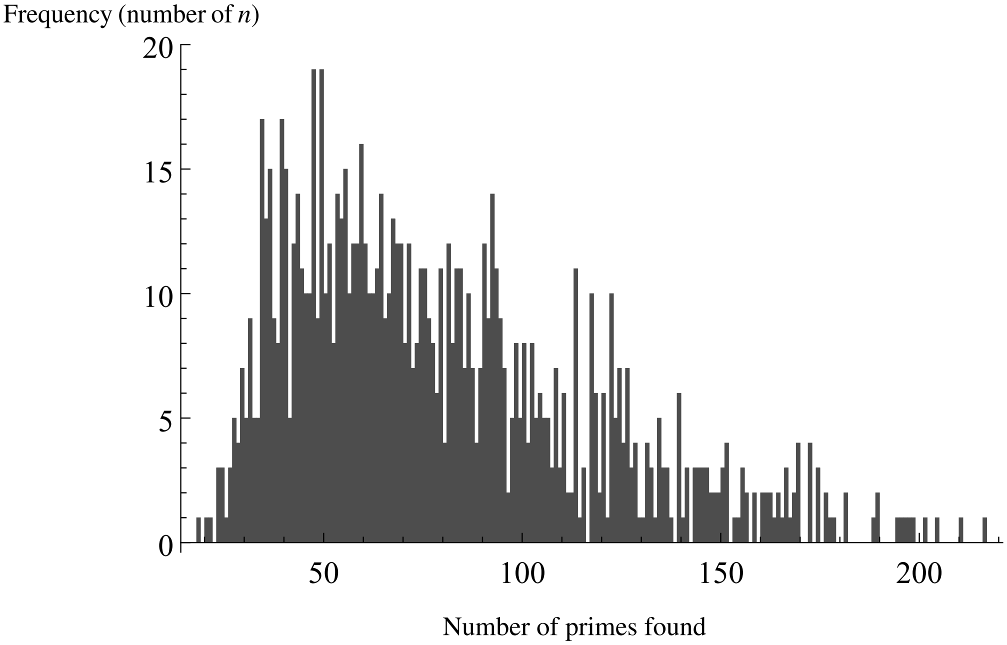


Figure 4b: Histogram for primes represented by quadratic form $x^2 + 32ny^2$.



3.4 Bound B

We also need to choose a suitable bound B on x and y. The most critical concern is to insure $x_1x_2 \gg n$, as for each encrypted bit we have

$$e = (x_1^2 + 8ny_1^2)(x_2^2 + 8mny_2^2) = (x_1x_2)^2 + 8n\big(m(x_1y_2)^2 + (x_2y_1)^2\big) + 64mn^2(y_1y_2)^2, \tag{7}$$

where $m = 1$ or 4. Thus, when $x_1x_2 < n$, $x_1x_2 = e \pmod n$, and in the situation when x_1x_2 could completely be factored, then x_1 and x_2 could in many cases be determined quickly. For example, if x_1x_2 had 12 prime factors all roughly the same number of digits, then x_1x_2 could be factored by the elliptic curve factoring algorithm due to each factor being roughly $n^{1/12}$, or about 51 digits in size if n was a 2048 bit number. In this example the number of factor combinations to check to reconstruct x_i would be very modest. It would be enough to reconstruct either x_1 or x_2 directly and then determine x_2 or x_1 by factoring x_1x_2. The various combinations of factors could tried methodically. Since x_1 and x_2 are roughly the same size, it likely would be sufficient to initially choose 6 factors out of 12 and only have to test $\binom{12}{6} = 924$ combinations.

We recommend making $\lg x_i \approx \lg \sqrt{8n}y_i$. A very safe bound B on y_i would $B = 2^{128}$. In the case of the quadratic form $z_{32} = x^2 + 32ny^2$, the bound on y should be $\frac{1}{2}$ the bound on y used for the quadratic form $z_8 = x^2 + 8ny^2$, this insures z_8 and z_{32} have the same range. The recommended bound B is either 2^{128} or 2^{127} depending on the quadratic form. Putting this all together we get

$$B/8 < y_i < B,\ y_i \text{ odd, and}$$
$$\left\lfloor \frac{y_i\sqrt{8n}}{2} \right\rfloor \le x_i \le \left\lfloor \frac{3y_i\sqrt{8n}}{2} \right\rfloor,\ x_i \text{ odd.}$$

4. Security

Since encryption is done using a probabilistic method, all message bits are expected with extremely high probability to always produce different ciphertexts.[13] So this encryption scheme is not vulnerable to attacks using a dictionary of known ciphertexts. This scheme also has the property that its security against each individual encrypted bit is adaptable by varying the size of B.

The bound B also needs to be large enough ensure that no prime used to encrypt a message bit is ever inadvertently used a second time. When two ciphertexts share a common prime this common prime can be extracted out quickly with a single GCD operation.

Theorem 16. *[Birthday paradox (Fact 2.27), 29] The expected number of draws before a collision from randomly selecting a number in the range* $[1, r]$*, inclusive, with replacement, as r* $\to \infty$ *is* $\sim \sqrt{\pi r/2}$.

When constructing each encryption prime, the odd y value of the quadratic form is chosen randomly first and the odd x value is then randomly chosen from

$$\left\lfloor \frac{y\sqrt{8n}}{2} \right\rfloor \le x \le \left\lfloor \frac{3y\sqrt{8n}}{2} \right\rfloor,$$

[13] Historical note: a classical substitution cipher is termed a *homophonic substitution cipher* when a plaintext symbol $a \in \mathcal{A}$ is replaced with a randomly chosen string from a fixed set $H(a)$, where for all $a \in \mathcal{A}$, $H(a)$ and a are pairwise disjoint and $|H(a)|$ is the same [29, Definition 1.28].

so the number count of possible quadratic form values z is $O\left(\sqrt{8n}\right)$ and the average probability of of z being prime is $2\log z$. We can approximate the number count of possible prime z values by $O\left(\sqrt{n}/\log n\right)$, and by the above birthday paradox theorem we observe a collision will realistically not occur until $O\left(n^{1/4}/\sqrt{\log n}\right)$ primes have been generated. The $n^{1/4}$ term dominates and clearly there is no earthly possibility of a collision.

One might think the the primes q_1 represented by $x_1^2+8ny_1^2$ are somehow distinguishable from the primes q_2 represented by $x_2^2+32ny_2^2$. However, once q_1 and q_2 are multiplied together there is no known way to determine x_i and y_i for q_i without factoring q_1q_2. Since $x_1^2 + 8ny_1^2 = x_1^2 + 2n(2y_1)^2$ and $x_2^2 + 32ny_2^2 = x_2^2 + 2n(4y_2)^2$, y_i odd, it is enough to determine y_2 modulo 4. Currently, we know of no way to do this without knowing q_1 and q_2 or knowing the private key (a, b).

Definition 11 (Quadratic Form Problem)**.** Let n be the product of two primes each congruent to 1 modulo 4. Let p, q and r be primes, and let x_i and y_i be odd numbers. Let $p = x_1^2 + 8ny_1^2$, let $q = x_2^2 + 8ny_2^2$, and let $r = x_3^2 + 32ny_3^2$, where y_1, and y_2 are randomly and uniformly drawn from $[1, B]$, with B large enough that $\min\left(x_1^2x_2^2, x_2^2x_3^2\right) \gg n$ with high probability. The bound on y_3 is $B/2$. The values x_1, x_2 and x_3 are randomly drawn from $\left[1, \lfloor\sqrt{T}\rfloor\right]$, where $T = \sqrt{8ny_i^2}$ for x_i, $i \in \{1, 2\}$, and $T = \sqrt{32ny_3^2}$ for x_3. The *quadratic form problem* is to determine whether $w = pq$ or $w = pr$ without factoring w or n, or utilizing $\sqrt{-1} \pmod{n}$ or some efficiently computable equivalent. Only w, n and B are given, the x_i and y_i values are unknown.

The bound on y_3 is set to half of the identical bound on y_1 and y_2, so that the two cases: $w = pq$ and $w = pr$, are computationally indistinguishable in polynomial time in $\log w$. (See §2.2 and 3.3.2.)
Currently, not only does this problem require either knowing (a, b) or the factorization of n or w, it appears an oracle that can solve this problem does not help with factoring n or w, or even help to determine (a, b). Clearly, there could be many variants of this problem. Some quadratic form patterns can be easily distinguished as the primes they represent have a unique distribution of residuals modulo certain prime powers. For example, the following is an unusable set of quadratic forms for cryptographic purposes:

$$s = \begin{cases} +1, & q_1 = x^2 + 4ny^2 \text{ and } q_2 = x^2 + x^2 + 4ny^2, \\ -1, & q_1 = x^2 + 4ny^2 \text{ and } q_2 = x^2 + x^2 + 16ny^2. \end{cases}$$

It is easily broken by computing q_1q_2 modulo 8. When $s = 1$, $q_1q_2 \equiv 1 \pmod{8}$, and when $s = -1$, $q_1q_2 \equiv 5 \pmod{8}$.

Theorem 17. *Let n be an odd integer equal to $a^2 + b^2$, B be a positive integer bound, P be the set of all primes of the form $x^2 + 8ny^2$, with x and y both odd positive integers $< B$, and Q be the set of all primes of the form $x^2 + 32ny^2$, with x and y also both odd positive integers $< B$. Let m be any integer > 1 such that $\gcd(m, n) = 1$. Then for sufficiently large B the distribution of the residues of the sets P and Q modulo m are statistically indistinguishable. In other words, for all m, and $1 \leq r < m$, a random element of P modulo m and a random element of Q modulo m are both equally likely to equal r. Further, for a prime $p \in P$, where p is constructed by randomly choosing x and y independently until $x^2 + 8ny^2$ is prime, and likewise for q using $x^2 + 32ny^2$, the residues of p and q modulo m have the same probability distribution.*

Proof. By the Chinese remainder theorem there are two cases to consider: when m is an odd prime power and when m is a power of 2. We first consider the case when m is the power of an odd prime. We observe that since $8ny^2 = 2n(2y)^2$, and $32ny^2 = 2n(4y)^2$, then for a random odd y there is no difference in the probability distribution of the residues of $8ny^2$ and $32ny^2$ modulo m as the quadratic characters of these residues modulo m are the same.

Next we consider the case when $m = 2^k, k \in \mathbb{Z}_{>0}$. Since x and y are odd we have that x^2 and y^2 are both $\equiv 1 \pmod 8$. For $q_1 = x_1^2 + 8ny_1^2$ and $q_2 = x_2^2 + 32ny_2^2$, we have, respectively, $x_1^2 = 8\alpha_1 + 1$, $y_1^2 = 8\beta_1 + 1$, $x_2^2 = 8\alpha_2 + 1$, and $y_2^2 = 8\beta_2 + 1$. Let $n = 4\gamma + 1$, since $n \equiv 1 \pmod 4$. Putting this all together we get

$$\begin{aligned} q_1 =& 8\alpha_1 + 32\gamma + 64\beta_1(4\gamma + 1) + 1, \\ q_2 =& 8\alpha_2 + 128\gamma + 32 + 256\beta_2(4\gamma + 1) + 1, \end{aligned}$$

where α_i,β_i and γ are unrestricted non negative integers. For $k \leq 3$, q_1 and q_2 are both always $\equiv 1 \pmod{2^k}$. For $k > 3$, the term $8\alpha_i$ controls the value of $q_i \pmod{2^k}$. Since α_1 and α_2 are both unrestricted random non negative integers, both quadratic forms, representing q_1 and q_2 respectively, encompass the same set of residues modulo 2^k and have the same probability distribution. □

4.1 Encryption Scheme Attacks

With current knowledge, assuming B is large enough to be secure, it appears that the only known way to break this encryption scheme requires factoring n or e. Since $n \ll e$, the security appears to depend on the bit length of n.

4.1.1 Polynomial and Semantic Security

Definition 12 (Goldwasser and Micali [23], also see [29, Definition 8.46])**.** A public-key encryption scheme is *polynomially secure* if no passive adversary in expected polynomial time can select two plaintext messages m_1 and m_2 and correctly distinguish between encrypted(m_1) and encrypted(m_2) with probability greater then $\frac{1}{2} + \varepsilon$, where $\varepsilon < 1/P(\lg k)$, for some polynomial P, where $\lg k$ is the length of the public key.

Definition 13 ([29, Definition 8.47] and [23])**.** A public key-encryption scheme is *semantically secure* if, over all probability distributions of the message space, whatever a passive adversary can determine in expected polynomial time about the plaintext message given the ciphertext, it can also determine in expected polynomial time without the use of the ciphertext.

Basically the above definitions are saying no passive adversary can select m_1 and m_2 and then distinguish encrypted(m_1)and encrypted(m_2) with polynomial time bounded computational resources. In other words, the adversary can learn nothing as the ciphertext leaks no information that can be computed in expected polynomial time [29, page 306]. In essence semantic security is a polynomial bounded version of Shannon's perfect secrecy [29, Remark 8.48].

Example 2. [29, §8.7] RSA is not semantically secure. With public key (n, e), for plaintext m, we can recover some slight information about m from the ciphertext $c = m^e \pmod n$, namely, the Jacobi symbol value

$$\left(\frac{m}{n}\right) = \left(\frac{m}{n}\right)^e = \left(\frac{m^e}{n}\right) = \left(\frac{c}{n}\right).$$

However, RSA with Optimal Asymmetric Encryption Padding (OAEP) is secure against adaptive chosen-ciphertext attacks using the random oracle model under certain assumptions [22], [6]. Pure RSA is deterministic and m always encrypts the same way and consequently one can easily detect when a plaintext message is reused.

Theorem 18. *Semantic security is equivalent to polynomial security [23].*

Our encryption scheme based on the Quadratic Form Problem (QFP) is semantically secure assuming the difficulty of factoring and the difficulty of the QFP. There is no known way to determine anything about the plaintext bit from the ciphertext in expected polynomial time.

4.1.2 Adaptive Chosen-Ciphertext Security

Definition 14 (IND)**.** *Indistinguishability* (IND) is defined by a game with a challenger and an adversary $\mathcal{A}$. The adversary is allowed to choose any two plaintext messages m_1 and m_2. A encryption scheme is probabilistic polynomial time (PPT) indistinguishable when every PPT adversary $\mathcal{A}$, given the encryption of m_i by a challenger with $i \in \{1, 2\}$, cannot identify the message choice with probability negligibly better than 1/2.

Basically, no PPT adversary $\mathcal{A}$ can do better than random guessing. Next, we will give $\mathcal{A}$ greater and greater powers through the use of oracles and see if $\mathcal{A}$ can do better than chance. Any encryption scheme that stands up to these escalating attacks by $\mathcal{A}$ is progressively more secure, all other things being equal.

Theorem 19. *Assuming the quadratic form problem is difficult, then our quadratic form encryption scheme has the property of indistinguishability.*

Definition 15 (Adaptive chosen-ciphertext attack [29, pages 41–42], [17])**.** A *chosen-ciphertext attack* is a type of cryptographic attack where the adversary has the means to select ciphertext and this chosen ciphertext is then decrypted by a decryption oracle [31]. The chosen ciphertext is selected prior to attaining the target ciphertext(s).

An *adaptive chosen-ciphertext attack* ($ACCA$) is a type of chosen ciphertext attack where the choice of ciphertext is adaptive and may depend on the plaintext received from prior decryption queries. The goal is to deduce information about the plaintext of the target ciphertext(s).

Definition 16 (Indistinguishable against adaptive chosen ciphertext attack [*IND-CCA2*])**.** *Indistinguishable against adaptive chosen ciphertex*t attack means indistinguishability holds even when adversary $\mathcal{A}$ can decrypt arbitrary ciphertext messages before obtaining the target ciphertexts C. Next, $\mathcal{A}$ is given access to the decryption oracle after receiving C, (with the restriction $\mathcal{A}$ cannot query the decryption oracle with any ciphertexts in C), but ciphertexts related to C are permitted to be sent to the decryption oracle.

An encryption scheme that is $IND - CCA2$ secure has strong security as $\mathcal{A}$ has access to the decryption oracle after obtaining C, and these queries to the decryption oracle can be customized by the knowledge gained prior to obtaining C.

One way an adaptive chosen-ciphertext attack can be done is if the adversary gets access to the equipment used to decrypt, but without access to the actual decryption key. Also, in the case of pure trap door signature schemes such as RSA, an attacker can theoretically mount a adaptive chosen-ciphertext attack by submitting messages to be signed by the key holder.

Assuming the intractability of the quadratic form problem, this encryption scheme appears to be plaintext aware and hence resistant to chosen-ciphertext attack. This is of interest as the efficient Blum-Goldwasser encryption scheme, which is also probabilistic, (but can operate on many bits at a time), is not secure against chosen-ciphertext attacks.

Systems secure against adaptive chosen-ciphertext have been around for quite some time. For example, Goldwasser, Micali and Rivest (1988)[24] proposed a signature scheme both secure and practical against

$ACCA$ based on the difficulty of factoring. Cramer and Shoup (1998) [17] developed a practical cryptosystem and secure against $ACCA$ based on the difficulty of the Diffie-Hellmen decision problem.

Despite $r \equiv \sqrt{-1} \pmod{e}$ being sent to Bob, and thus public, r does not appear to help factor e. What is needed is t, a second square root of -1 modulo e, where $|t| \not\equiv |r| \pmod{e}$. Then we would have $r^2 \equiv t^2 \equiv -1 \pmod{e}$, with $(r+t)(r-t) \equiv 0 \pmod{e}$ and $\gcd(r \pm t, e)$ would yield the two prime factors e.

We point out that for Fermat numbers $F_n = 2^{2^n}+1$, the square root of -1 is known, as $\sqrt{-1} \equiv 2^{2^{n-1}} \pmod{F_n}$. Despite this additional information there currently is no known way to exploit this additional information to help factor Fermat numbers.

4.1.3 Plaintext Aware

Definition 17 (Plaintext Aware [6])**.** A public-key encryption scheme is *plaintext-aware* if it is computationally unfeasible for an adversary to generate a legitimate ciphertext without knowing the corresponding plaintext.

While the intuitive concept of plaintext aware is straight forward. Formally defining it is more difficult. The definition varies depending if one is using the random oracle model or not, and on the purpose. There are progressively stronger definitions of plaintext aware (such as $PA0$, $PA1$ and $PA2$), in the sense that those encryptions schemes that meet these plaintext aware definitions are also progressively more secure [5].

The idea that knowing the plaintext of a ciphertext renders a decryption oracle useless dates back to Blum, Feldman and Micali from 1988 [9], [10]. Later, in 1994, Bellare and Rogaway formalized the notion of plaintext aware (PA) using the random oracle ($\mathcal{RO}$) model [6]. The definition of plaintext aware using the random oracle model was motivated by the idea that $PA + IND - CPA$ (indistinguishability under chosen-plaintext attack) should entail $IND - CCA2$ (indistinguishability under adaptive chosen-ciphertext attack).

Plaintext awareness is a desirable feature for an encryption scheme to have. A plaintext aware encryption scheme with semantic security is secure against a chosen ciphertext attack as the attacker already knows the plaintext of any chosen ciphertext that they would query for decryption. The first provably secure and practical plaintext aware encryption scheme was developed by Bellare and Rogaway in 1994 [6]. Their proof was done using the $\mathcal{RO}$ model, meaning the hash function is assumed to be “ideal” and completely random. While collision resistant and one-way hash functions exist, it is an open and controversial question whether real world instantiations of ideal hash functions are sufficiently random such that these types of theoretical proofs can be trusted. An encryption scheme (or protocol) that fails to be provably secure using the $\mathcal{RO}$ model is clearly weaker in some sense than one that does not fail, but how much extra confidence to give an encryption scheme that is provably secure using the $\mathcal{RO}$ model is currently, perhaps, more a matter of personal “theology” than mathematics.

Definition 18 (Bellare’s and Rogaway’s 1994 definition of plaintext aware [6])**.** Using the random oracle model, an encryption scheme is *plaintext aware* (PA_{BR}) when for every adversary $\mathcal{A}$ outputting ciphertext from the public key $\mathcal{K}$, there exists an extractor $\mathcal{E}$ that with $\mathcal{K}$ and a reproduction of the interaction of $\mathcal{A}$ with its random oracle ($\mathcal{RO}$) is able to decrypt the ciphertext outputted by $\mathcal{A}$ [5].

Later it was determined that Bellare’s and Rogaway’s 1994 definition of PA_{BR} was too weak to support the inference that $PA_{BR} + IND - CPA \Rightarrow IND - CCA2$. Bellare, Desai, Pointcheval and Rogaway extended the formal definition of PA using the $\mathcal{RO}$ model to include the ability for the adversary to be able to eavesdrop on communications intended for the receiver of ciphertexts, thus the adversary can acquire ciphertexts with unknown plaintexts. Using this enhanced plaintext definition ($PA2$), Bellare et al. was able to show that $PA2 + IND - CPA \Rightarrow IND - CCA2$.[14]

[14] The history of plaintext awareness given here (starting from 1988) very closely follows [5].

Theorem 20 ([4], [5])**.** *Any public key encryption scheme that is semantically secure and plaintext-aware with eavesdropping ($PA2$) is secure against adaptive chosen-ciphertext attacks ($IND-CCA2$). This theorem is true both in the random oracle model and the standard model.*

Bellare and Palacio in 2004 proposed a definition for plaintext awareness ($PA1$) without using the random oracle model [5]. We will just give a non technical definition of $PA1$ that captures the applicable notions. This definition closely follows Birkett's and Dent's description [7].

Definition 19 (Plaintext awareness ($PA1$) without $\mathcal{RO}$ model)**.** For any polynomial-time algorithm $\mathcal{A}$ (aka the *ciphertext creator*), which outputs ciphertexts, there exists a second polynomial-time algorithm $\mathcal{A}^*$ (aka the *plaintext extractor*), which is given all the inputs, including any random coins, that $\mathcal{A}$ has, and by this $\mathcal{A}^*$ will output the correct plaintext messages corresponding to the ciphertexts outputted by $\mathcal{A}$.

In essence, $\mathcal{A}^*$, the *plaintext extractor*, is like a polynomial time "spy" that watches $\mathcal{A}$, the adaptive chosen-cipher text *attacker*, as the ciphertext is created, and from this information the "spy" is able to reconstruct the underlying plaintext. The "spy," then, gives this back to $\mathcal{A}$, the attacker, when $\mathcal{A}$ makes a query to the decryption oracle [7].

In 2013 Birkett and Dent [7] formally defined a new notion of plaintext awareness: $PA1+$, which is stronger than $PA1$ but weaker than $PA2$, and formally distinct from both. (In essence, an encryption scheme is *PA1+* if it continues to be plaintext aware even when an attacker has access to new fixed length random strings at will. In *PA1*, the plaintext extractor $\mathcal{A}^*$ has access to all the random coins of $\mathcal{A}$ and knows what it is going to do. In *PA1+*, $\mathcal{A}$ has access to random bits after it has received the response of $\mathcal{A}$, the extractor, and this randomness can alter $\mathcal{A}$ response [19].) They showed how $PA1+$ can be used to prove that the Cramer-Shoup public-key encryption scheme is $PA2$, which informally, is plaintext awareness with eavesdropping.

Also, the notion of PA appears to not be very helpful in proving an encryption scheme is $ACCA$ secure, all known techniques for proving plaintext awareness can only be applied to encryption schemes which are already known to be $IND-CCA2$ secure [7, Introduction and §4].

We propose an extended definition of plaintext aware that is stronger than *PA1*. In our new definition, the adversary is able to eavesdrop both the ciphertext and the corresponding plaintext. We believe that our quadratic form encryption scheme meets this stronger definition.

Definition 20. A public-key encryption scheme is *dual plaintext-aware* if it is computationally infeasible for an adversary to generate a legitimate ciphertext without knowing the corresponding plaintext, and further, given any set of ciphertexts C, each of fixed bit length t, and access to an oracle that randomly generates both a ciphertext c, with the restriction that $c \notin C$, and its corresponding plaintext, then it is computationally infeasible for the adversary to gain information about the plaintexts in C.

We forego from giving a formal, technical definition for *dual plaintext-aware* — akin to putting a rough unpolished stone atop a velvet black cushion. It seems premature to build a rigid, formal definition a top a foundation that barely has had time to set. The one-way trapdoor encryption scheme that inspired this definition has yet to pass the test of time.

Our scheme appears to be dual plaintext-aware as there is no known way to generate two primes q_1 and q_2, or their product $e = q_1q_2$, so that each of q_1 and q_2 conform to the above binary quadratic form requirements without knowing (x_1, y_1) and (x_2, y_2) respectively. Given (x_i, y_i), then with Cornacchia's algorithm one can quickly determine whether $q_i = x_i^2 + 8ny_i^2$ or $q_i = x_i^2 + 32ny_i^2$. In other words, it is not computationally feasible to generate legitimate primes q_1 and q_2 or their product e without knowing the corresponding plaintext that goes with q_1, q_2 or n.

By Theorem 14 the probability of randomly guessing a prime that would have the correct quadratic form: $x^2 + 4^t \cdot 8ny^2$, x, y both odd, and $t \in \{0, 1\}$, and with the correct t, is exponentially low. Factoring the public key n would in general be more efficient.

Knowing full information about other plaintext and ciphertext pairs regardless of which public key n is used, appears to be of no help in decrypting any particular ciphertext. Unless ciphertexts share a common prime factor, which can be made astronomically improbable, we know of no way that ciphertexts can help decrypt each other, even when their underlying plaintext is known.

We refrain from rigorously conditionally proving our encryption scheme is plaintext aware as we would need to assume the hardness of certain number theory problems to do so. Some of the assumptions required, though virtually certain to be computationally hard, are unusual and not well studied, and any rigorous proof using these assumptions would hardly engender any additional sense of security than a simpler intuitive examination of the situation. We believe in this situation it is more transparent to conditionally assume plaintext awareness as opposed to assuming the computational hardness of certain non standard number theory problems and then proving plaintext awareness based on these hardness assumptions.

Consider the following. Suppose this encryption scheme was not plaintext aware. If we know the factors q_1 and q_2 of e we can use Cornacchia's algorithm to determine $q_1 = x_1^2 + 8ny_1^2$ and $x_2^2 + 8ny_2^2$ and hence we would know the plaintext bit value by examining $y_i \pmod 4$. So we can exclude knowing the factors of e. Instead, we have the following problem to solve. The ciphertext component e must be constructed in a way that not all the underlying prime factors are known, and at the same we must be able to compute $\sqrt{-1} \pmod e$. This mean we have $\alpha e = x^2 + 1$, for some known $x, \alpha \in \mathbb{Z}_{>0}$, and then we have to determine that each unknown prime factor of e is of the form $x^2 + 2ny^2$, with x odd and $2||y$ or $4||y$. Even if we knew $\sqrt{-2n} \pmod e$, so we could compute $e = x^2 + 8ny^2$ using Cornacchia's algorithm, it would not tell us which prime factors of e could be represented by f_8 or f_{32}. This appears to be a hard problem that doesn't reduce to any other standard number theory problem other than factoring. With our current knowledge, to construct a rigorous proof of plaintext awareness would require assuming the above problem or variants of it are computationally hard.

We have the situation that by assuming the hardness of the QFP, the hardness of some standard number theoretic problems and plaintext awareness ($PA1$), then we can show this encryption scheme is secure against an $ACCA$, but to prove plaintext awareness ($PA1$) requires nonstandard number theoretic hardness assumptions which are more convoluted to state and no more convincing than just assuming $PA1$. (Note, Bellare and Palacio [5] showed that $IND - CCA2$ does not imply $PA1$.)

Theorem 21. *Assuming the difficulty of both the integer factoring problem (IFP) and the modular square root problem ($SQRTP$), there is an equivalence between this encryption scheme being secure against an adaptive chosen ciphertext (ACC) attack and the hardness of the quadratic form problem (QFP).*

Proof. To show equivalence we will prove the negative of each of the two implications.

($\Rightarrow$), clearly if the quadratic form problem is not hard, then this encryption scheme is broken on a bit by bit basis, and not secure against an adaptive chosen ciphertext (ACC) attack.

($\Leftarrow$), consider the following scenario with Alice, Bob, Eve and Eva. If Eve can mount a successful attack on her own, then she can solve the QFP for ciphertexts C. So we can assume Eve has to acquire some ciphertext she has not created to mount a successful ACC attack. Let Eve eavesdrop on Eva's ciphertexts. If Eve subsequently can do a successful ACC attack, then Eve and Eva working together can solve the quadratic form problem for target ciphertexts C of Eve's choosing. Eva constructs her ciphertexts, and when Eve and Eva work together there are no external ciphertexts with unknown plaintexts other than C, and Eve's and Eva's ciphertexts are indistinguishable; however, this successful ACC attack yielded a solution of the quadratic form problem with ciphertexts C, and this violates the hardness assumption of the quadratic form problem. □

4.1.4 Non-Malleability

Definition 21 (Non-malleability [18])**.** Given a challenge ciphertext c, an encryption scheme is *non-malleable* if an adversary is unable to construct a ciphertext $\acute{c}$ such that the plaintexts m, $\acute{m}$ of c and $\acute{c}$, respectively, are "meaningfully" related, such as $m = -\acute{m}$, for example.

This encryption scheme is non-malleable for fixed bit length e. Since we are operating on a bit level, malleability only makes sense if for some plaintext bit(s) $m_i \in \{-1, 1\}$ we can construct a ciphertext $\acute{c}_i$ such that $\acute{c}_i$ encrypts $-m_i$. Since $\lg e$ is fixed in length we cannot multiple e by another prime of the form $x^2 + 8ny^2$, y odd. Substituting a prime $\acute{q}_j$ of the form $x^2 + 8ny^2$ for one of the factors of e_i requires knowing the factors of e_i which is computationally hard.

In 2005 M. Fischlin [21] introduced the notion of *complete non-malleability* by extending the definition to include the possibility of substituting the real public key K_P with a "fake" public key $\acute{K}_P$ (without the attacker necessarily knowing the secret key associated with this fake public key) and thereby making it possible to create a ciphertext with a plaintext with a known relationship with the target plaintext under the new "fake" public key. Fischlin showed that while the Cramer-Shoup DDH encryption scheme is *non-malleable* in the classical sense, it is not *completely non-malleable*. Briefly, Cramer-Shoup DDH consists of: a collision-intractable hash function H, a group $\mathcal{G}$ of prime order q, two random generators g_1, g_2 and elements c, d and h where $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$ and $h = g_1^{z_1} g_2^{z_2}$, for random $x_1, x_2, y_1, y_2, z_1, z_2 \in \mathbb{Z}_q$. For a plaintext $m \in \mathcal{G}$, with ciphertext $c = (u_1, u_2, e, v)$ one can create a ciphertext $\acute{c} = ({u_1}^2, {u_2}^2, e^2, \acute{v})$ with plaintext $\acute{m} = m^2 \in \mathcal{G}$ using the the public key $\acute{K}_P = (\mathcal{G}, g_1, g_2, H, \acute{c}, d, h)$. Full details are given in Fischlin's paper.

In our encryption scheme, an encrypted bit encodes as the ordered pair $\left(e = q_1 q_2, r \equiv \sqrt{-1} \pmod{e}\right)$, and if we require that $\lg e$ has a fixed size for all encryptions with a given public key n, then we know of no way to create a known relationship between a target message bit m and a new message bit $\acute{m}$ under any scenario short of decrypting the ciphertext (e, r) associated with m. Creating a "fake" public key $\acute{n}$ is unlikely to help as then one is trying to create a relationship between quadratic forms with different discriminants. We believe this encryption scheme is completely non-malleable.

4.1.5 Strong Plaintext Awareness & Secret-Key Awareness

Informally Barbosa and Farshim in 2010 defined *strongly plaintext aware* (SPA) as the inability to construct a ciphertext under any public key without knowing the plaintext. This means there exists a strong plaintext extractor (see Definition 18) that decrypts ciphertexts, no matter the public key, including public keys generated by an adversary [2].[15]

Theorem 22. *[2] An encryption scheme that is strongly plaintext aware and* $IND - CPA$ *secure is secure against strong chosen-ciphertext attacks.*

We believe our encryption scheme is strongly plaintext aware, but we have no proof of this. It is an open problem if this could be proven assuming the hardness of standard number theory problems.

Definition 22. [2] A public key encryption scheme is *secret-key aware* when it is infeasible to generate a public key in probabilistic polynomial time without also being able to determine in probabilistic polynomial time the secret-key linked with the public key.

[15] Roughly, the formal definition is based on two games, with an adversary $\mathcal{A}$ that has access to two different oracles: an inversion oracle and a polynomial-time secret key extractor $\mathcal{E}$ oracle. The adversary $\mathcal{A}$ also has access to extra auxiliary information for producing public keys, as well as a decryption and encryption oracle. An encryption scheme is secret-key aware if the behavior of $\mathcal{A}$ is computationally indistinguishable between the two oracles.

Theorem 23. *[2] An encryption scheme that is secret key aware (SKA) and $IND-CCA2$ secure is probabilistic polynomial time secure against strong adaptive chosen-ciphertext attacks ($IND-SCCA2$) and therefore completely non-malleable.*

Barbosa and Farshim also showed using the random oracle model that RSA-OAEP can be modified to be secret-key aware ($SKA2$). They then showed that using what they termed *knowledge of factoring assumptions* an RSA-based scheme could be constructed using a public key of the form p^2q that was provably weakly secret key secure (and hence $IND-SCCA1$ secure) using the standard model, not the random oracle model.

Our encryption scheme is not completely secret-key aware, but we show how it can be modified to be weakly secret key aware, meaning secret key aware, but with some *knowledge of factoring assumptions*. It is possible to generate a functional public key without knowing the secret key. An adversary such as Mallory may possibly succeed in creating a functional public key by randomly choosing an $n \equiv 1 \pmod 4$ and making sure n is not divisible by small primes. Mallory can only give a probability that n is valid. For n to be valid it must be expressible as the sum of two integer squares. There appears to be no way to know this is the case without knowing the prime factors of n or knowing $\sqrt{-1} \pmod n$, in which case the secret private key is efficiently computable.[16] For n to be secure as well, it must be the product of two primes, each of comparable bit length, which currently requires knowing the factorization of n. The density of all k-bit numbers which can be represented as the sum of two integer squares is non negligible. The density of numbers that are the product of two primes, with each prime very close to $\lfloor k/2 \rfloor$ bits in length is non negligible as well.

Barbosa and Farshim [2] in the pursuit of secret-key aware encryption schemes that are strongly plaintext aware without random oracles defined the notion of *weakly secret-key aware* by using public keys of the form $n = pq^2$. (They defined formal *knowledge of factorization assumptions* (KFA's) of varying strength concomitant with their definition of weak secret-key awareness. Roughly, the KFA, in its weakest form, means a number n of the form pq^2 cannot be constructed in probabilistic polynomial time without starting with p and q, and in addition, having an auxiliary number m of this form is of no help.) Numbers of the form pq^2 have a negligible density in the set of all k-bit integers.

For this encryption scheme there should be no obstacle to using a public key n of form pq^2 instead of the form pq. If at a later time it is determined that pq^2 type numbers are not secure enough, one can also use numbers of the type pqr^2, which also have negligible density. Thus, these variants of the quadratic form encryption scheme likely achieve $IND-SCCA2$ security via secret key awareness with knowledge of factorization assumptions.[17]

4.1.6 Side Channel Attacks

Based on Equation (6) we would not expect the computer resources needed to find primes represented by each quadratic form type to differ significantly; however, unless we assume the GRH, the error terms in the L-function are sufficiently large that we are not confident that there are no weak n. By weak n we mean the computer time needed to find each type of prime is different enough to cause some plaintext information leakage, such as, for example, the high order bits of the Hamming weight of the complete message.

Even assuming the GRH we have not made the effort to determine how large n has to be for the error terms to be small enough to insure that the computation time needed to find each type of primes is only negligibly different, no matter the value of n. Rather we examined several blinding methods and by far the simplest method is to change the encryption scheme slightly.

[16] It is an open problem if there is an algorithm to determine if a whole number n can be expressed as the sum of two squares that does not require factoring n.

[17] Barbosa and Farshim state: that likewise, Hofheinz's and Kiltz's factorization based encryption scheme [25] is also a candidate for achieving *IND-SCCA2* security.

The following encryption scheme modification insures an equal number of primes of type f_8 and f_{32} are used. Let m_p the number of message bits that equal $+1$, and let $m_{ph} = \lfloor m_p/2 \rfloor + (m_p \pmod 2)$. For those message bits that equal $+1$ take m_{ph} number of these bits and use quadratic forms of type $f_8(x, y)$ for all the primes needed to encrypt these message bits. For the remaining bits that equal $+1$ use the quadratic form of type $f_{32}(x, y)$. If $m_p \pmod 2 > 0$ encrypt an extra $+1$ bit using $f_{32}(x, y)$ and then discard both primes. Since those message bits that equal -1 use primes of both types equally, this method of encryption insures that primes of both types are used in equal proportion for the message as a whole.

To further blind the difference between the two types of primes constructed, Alice can also generate the ciphertext in a random order. For a short message the whole message could be encrypted in a random order. For a long message blocks of bits could be encrypted randomly.

4.1.7 Fault Injection Attacks

This encryption scheme is vulnerable to fault injection attacks. The encryption of each plaintext bit requires computing two primes q_1 and q_2, along with $r_i \equiv \sqrt{-1}$ modulo each prime q_i. Next, using the Chinese remainder theorem, $r \equiv \sqrt{-1} \pmod{q_1 q_2}$ is computed as

$$r \equiv i \equiv \sqrt{-1} \equiv \pm r_1 q_2 v_2 \pm r_2 q_1 v_1 \pmod e,$$

where $v_2 = q_2^{-1} \pmod{q_1}$, and $v_1 = q_1^{-1} \pmod{q_2}$.

If the computation of $\sqrt{-1}$ was in error for one prime, say q_1, but not the other, then $e = q_1 q_2$ could be factored and the plaintext bit could easily be computed by using Cornacchia's algorithm. Let $\hat{r}$ be the faulty value of $\sqrt{-1} \pmod e$, then $\hat{r}^2 \not\equiv -1 \pmod{q_1}$, but $\hat{r}^2 \equiv -1 \pmod{q_2}$, and consequently $\gcd(\hat{r}^2 + 1, e) = q_2$.

The same fault vulnerability exists for the computation of the private key (a, b), where $a^2 + b^2 = n$, as $\sqrt{-1} \pmod n$ is used to compute a and b. Since the private key is kept secret all that is needed is to check that $\sqrt{-1} \pmod n$ was computed correctly or alternatively that $a^2 + b^2$ does indeed equal n.

In essence there are two basic approaches to dealing with fault injection. Either validate the calculation or inject some randomness that causes a faulty computation to reveal no useful information [32]. In our case, one can easily verify that $r \equiv \sqrt{-1} \pmod e$ was computed correctly. The running time for encryption is dominated by the search for the two primes q_1 and q_2, so the relative time penalty for verifying $r^2 \equiv -1 \pmod e$ is minor.

Randomness can be injected into the process, but verification is still required. Instead of computing $r \equiv \sqrt{-1} \pmod e$, one computes $\acute{r} \equiv \sqrt{-x^2} \pmod e$ using the Chinese remainder theorem, where x is a secret random number. Next, one verifies that $\acute{r}^2 \equiv -x^2 \pmod e$ and then one computes $r \equiv \sqrt{-1} \equiv \acute{r} \cdot x^{-1} \pmod e$.[18]

A fault could be introduced in the search for the two primes q_1 and q_2. Two obvious types of faults can occur. First, one or both of q_1 and q_2 may be composite. If this type of fault occurs it will with high probability show up as an error in the subsequent verification of $\sqrt{-1} \pmod e$. Second, one or both of the primes are not of the form $x^2 + 8 \cdot 4^t n y^2$, $t \in \{0, 1\}$. This would be very unlikely to occur as a fault would have to occur during the very narrow time window when the number to be tested for primality is being constructed, and then either there would be no more faults, or any subsequent faults would have to not interfere with the actual prime test. This latter fault would not compromise the plaintext message as this type of fault would create an invalid ciphertext bit.

An important feature of this encryption scheme is that we know of no hardware faults occurring during decryption that an attacker could exploit to determine the secret key. Decryption does not require using the Chinese remainder theorem. The worse that can happen is during encryption some plaintext bits are encrypted incorrectly, and thus either their decryption will be incorrect, or the plaintext bits can be determined without the secret key. With high probability these incorrect encryptions can be detected with low computation cost.

[18] We thank Eric Bach for suggesting using $\sqrt{-x^2} \pmod e$.

4.2 Non Repudiation

It is not possible to prove conclusively that Alice is the sender. Alice can simply self repudiate on a per bit basis, claim she does not know the individual factors q_i of e and there is no way to prove otherwise. In other words, this encryption scheme without further enhancements is not immune to non-repudiation.

4.3 Authentication

Alice can authenticate that she is the sender by revealing q_1 or q_2 for any encrypted bit. Assuming the difficulty of factoring, only Alice would know the factorization of $e = q_1 q_2$. If message secrecy is required, Alice can authenticate herself with high probability in three ways.

First, with a simple challenge response protocol. Suppose Bob wishes to verify that Alice truly is the sender. Bob first chooses a parameter t, where the probability of a successful authentication occurring equals $1 - 1/2^t$. Bob randomly chooses an encrypted bit (r_i, e_i) and then randomly chooses t numbers c_j, with $1 < c_j < 2e_i^{1/2}$, $c_j \notin \mathbb{Z}^2$, and $\left(\frac{c_j}{e_i}\right) = 1$, $1 \leq j \leq t$. Bob then sends all the c_j's along with e_i (or just i) to Alice. Alice then searches through the c_j values until she finds a c_k that is a quadratic residue of e_i, she then returns (c_k, t_k), where $t_k \equiv \sqrt{c_k} \pmod{e_i}$ to Bob. Bob verifies that $t_k^2 \equiv c_k \pmod{e_i}$. The process is repeated if no quadratic residue c_k is found by Alice.

The above protocol insures that someone cannot impersonate Bob and trick Alice into revealing the factors of e_i. Suppose Paul is an imposter of Bob and randomly selects an integer $\sqrt{e_i} < z_j < e_j - 1$ and sends $z_j^2 \pmod{e_i}$ in place one of the c_j's. Now Alice will with probability $\frac{1}{t}$ return a (c_k, t_k), where $\gcd(t_k \pm z_j, e_i)$ equals q_1 and q_2, and thus Paul can determine the plaintext message corresponding to (r_i, e_i) by computing $\sqrt{-2n}$ for q_1 and q_2 and determining x_1, x_2, y_1 and y_2 where $q_1 = x_1^2 + 2ny_1$ and $q_2 = x_2^2 + 2ny^2$ using Cornacchia's algorithm in §2.1.2. If $y_1 \equiv y_2 \pmod 4$ then the plaintext bit is $+1$, otherwise it is -1.

Currently there is no known way to generate for a nontrivial quadratic residue $r \ll \sqrt{2m}$ a known square root of r modulo m without knowing the factors of m. By using the convergents of the continued fraction of $\sqrt{m}$, numbers x such that $x^2 < 2\sqrt{m} \pmod{m}$ can easily be generated, but these x are sufficiently random for large m that it would be computationally infeasible for Paul to find an (x, y) such that $x \equiv y^2 \pmod{e_i}$ and $\sqrt{e_i} < x < 2\sqrt{e_i}$.

Alice can also authenticate by providing $a \equiv \sqrt{n} \pmod{e}$ for each encrypted bit. Only the entity who generated the prime factors of e has the ability to compute $\sqrt{n} \pmod{e}$. All Bob needs to do is verify that $a^2 \equiv n \pmod{e}$. Since $q_i = x^2 + 2ny^2$, with $2 \| y$ or $4 \| y$, the encryption of the bits is not broken as knowing $\sqrt{-1} \pmod{e}$ and $\sqrt{n} \pmod{e}$ is not sufficient, one still needs $\sqrt{2} \pmod{e}$ to determine $y \pmod 4$.

Finally, at the expense of further computation and message expansion, Alice can use three primes q_1, q_2, and q_3 to encrypt a bit, where $q_3 = x^2 + 32ny^2$. To authenticate herself, Alice can reveal q_3. Alice can authenticate any encrypted bit without revealing the actual plaintext bit, as revealing q_3 does not help in factoring $e/q_3 = q_1 q_2$. The Jacobi symbol using q_3 reveals no information as

$$\left(\frac{(a + bi)(1 + i)}{q_3}\right) = 1$$

for all bits and thus no exploitable information is leaked. More than three primes can be used if multiple parties need to authenticate. This manner of authentication might be useful for certain e-commerce transactions and e-voting, as transactions or votes could be authenticated by multiple parties without revealing the actual transactions or votes. The slowness of the encryption step can to some degree be circumvented as random primes represented by both $x^2 + 8ny^2$ and $x^2 + 32ny^2$ can be pre-computed and thus a bounded number of message bits could be encrypted quickly.

If the property of non malleability needs to be preserved, then $\lg e$ needs to be within a narrow fixed range. If e needs to be made longer, then extra prime factors of the form $x^2 + 32ny^2$ can be used as when e is multiplied by primes of this form it does not change the decrypted bit value.

4.4 Complexity

Definition 23. Let $n \in \mathbb{Z}_{>0}$, and $0 < a < n$, $a \in \mathbb{Z}$. We define $|a|$ to be the smaller of a or $n - a \pmod{n}$.

There are some interesting complexity theory aspects to this one-way trapdoor function. First, just knowing (a, b) does not reveal the factors of n, but does break the security of the cryptosystem, so the problem of finding (a, b) given n may possibly be easier in general than factoring. To find (a, b) is polynomial time equivalent to computing $i = \sqrt{-1} \equiv a/b \pmod{n}$. Currently, in general, there is no known way to compute i without knowing the factorization of n. In those exceptional cases where $\pm i$ is known for n, such as when n is a Fermat number $2^{2^k} + 1$ or more generally $n = x^2 + 1$ and $i = \pm x$, there is no known way to use the knowledge of $\pm i$ to factor n, more information is needed, such as another value of $\sqrt{-1} \not\equiv \pm i \pmod{n}$. Given $x^2 \equiv y^2 \pmod{n}$, then $(x - y)(x + y) \equiv \pmod{n}$. Here $x^2 \equiv y^2 \equiv 1 \pmod{n}$ so we can split n by computing $\gcd(x \pm y, n)$, provided $|x| \not\equiv |y| \pmod{n}$.

To factor n with a square root oracle requires the oracle to randomly return any of the possible square roots modulo n. The standard way to factor n with a square root oracle is to randomly pick an $x \in (\mathbb{Z}/(n))^*$ and compute $r \equiv x^2 \pmod{n}$ and then query the oracle and get a square root v of $r \equiv x^2 \pmod{n}$. Given just r there is no way of knowing which of the many possible $\sqrt{r}$ modulo n values was used, the information is forever gone after squaring, so repeated queries to the square root oracle are needed to eventually return a v such that $|v| \not\equiv |x| \pmod{n}$ and thus split n. When $n = pq$, the expected number of oracle queries to get a new nontrivial square root of x^2 modulo n is 2. There is an exponentially low probability that an exponential number of queries would be needed. There is no way to absolutely guarantee a nontrivial v will be returned no matter how many queries are made.

Recalling Definition 11, it is evident that the QUADRATIC FORM PROBLEM (QFP) $<_P$ FACTORING, meaning QFP polynomial time reduces to the problem of finding the prime factorization of n (or w as well, but $\lg n < \lg w$, so factoring n is faster). We are not sure about the converse. However, in the real world, without oracles, we believe the QFP cannot be solved without factoring n, but we have no proof of this.

It is unlikely that such a proof will be found in the near future. Consider, the QFP is related to the QUADRATIC RESIDUE PROBLEM (QRP). Given an integer a and a composite number $n = pq$, where p and q are odd primes, and the Jacobi symbol $(a|n) = 1$, with a not an integer square, the *quadratic residue problem* is to determine if a is a quadratic residue of n or not. In other words, do the Legendre symbols $(a|p) = (a|q) = 1$ or -1.

The QRP also polynomial time reduces to FACTORING. While it is widely believed that QRP is easier than factoring, in truth at present, no one really has any realistic idea how tackle this problem.

5. Summary

This encryption scheme has some interesting properties due to the novel one-way trapdoor function used. Despite the intense message expansion and very slow encryption, this cryptosystem may someday have some limited practical applications when there is ample bandwidth available and the message to be encrypted is short. Further, the application possibilities can be broadened in those situation were the primes needed for encryption can be pre-computed or computed in parallel. This encryption scheme is very amenable to parallel computation. Each bit can be encrypted in parallel, and the search for the two special primes needed to encrypt each message bit can be done in parallel.

For short bit length messages the decryption time for our encryption scheme is asymptotically less than the decryption time for RSA for the same size public key. Each encrypted message bit in our scheme only requires $O(\lg n)^2)$ time to decrypt, but each bit requires two numbers, with each number having twice as many bits compared to the single number, encompassing $\lg n$ messages bits, used in RSA. The decryption time for RSA is $O(\lg n)^3)$. Thus this system may have applications where the decryption device has ample bandwidth, but limited computing power. Also, this system may be useful in those applications where encryption authentication is needed, but with no message bits revealed, such as in financial transactions and electronic voting.

In the long term, increases in computing power (and bandwidth) and decreases in cost favors encryption in general as the cost to encrypt (and decrypt) decreases rapidly in comparison to the cost to break a cryptosystem. The cost to encrypt and decrypt increases in polynomial time, while the cost to break cryptosystems that depend on the difficulty of factoring increase instead sub-exponentially. Computing cost is still decreasing exponentially, so keys can increase in size with no additional cost. Thus this cryptosystem will become more practical over time if computing costs continue to decline rapidly. Since the encryption can easily be parallelized this encryption scheme may someday be practical. Even when basic computer circuits cannot be made significantly faster, it may be possible to keep bringing manufacturing cost down long after this point and thus parallel computing costs would also decrease.

Based on data in [30] we conservatively estimate that in the thirty plus years since RSA was invented in 1977[19] to the year 2010 the cost of computing decreased by at least a million. In this same time period RSA public key sizes as measured by bit-length have typically increased 4 to 8 fold. This works out to a $10^6/8^3 \approx 1950$ to $10^6/4^3 \approx 15600$ fold reduction in the cost to encrypt and decrypt a message. In the case of this encryption scheme, the reduction in the cost to encrypt would be more akin to a 250 to 3900 fold reduction as the encryption takes $O((\lg n)^4)$ time, not $O((\lg n)^3)$. After a reduction in cost of this magnitude this encryption scheme wold be completely viable in terms of computational cost.

It is an open question if quantum factoring is truly a serious threat to public key cryptosystems that depend on the difficulty of factoring. As public keys get larger both the number of physical and logical qubits[20] needed to factor public keys[21] may become greater than what is economically feasible, or technologically or physically possible. [40].

Acknowledgements

We extend our appreciation to Eric Bach and Jin-Yi Cai for their guidance and encouragement. This research paper is an independent outgrowth of an ongoing joint research project with Eric Bach on derandomizing quadratic nonresidue construction.

[19] Note, The English mathematician Clifford Cocks while working for the Government Communications Headquarters also developed a system equivalent to RSA in 1973, but it remained classified until 1997.

[20] Depending on physics of the quantum computer, the amount of decoherence present and the degree of error correction required, a logical qubit, practically speaking, may have anywhere from about a few hundred to ten thousand physical qubits. University of Maryland–College Park physicist and quantum computer researcher Sankar Das Sarma is of the opinion that millions to billions of physical qubits would be needed to break RSA; however, a topological quantum computer theoretically needs very few or even just one physical qubit per logical qubit. It is an open research problem if a topological quantum computer can be scaled up sufficiently to be a threat to public key cryptography [33].

[21] Basically, there are polynomial time quantum algorithms to solve the hidden subgroup problem.

References

[1] E. Bach and J. Shallit, *Algorithmic Number Theory. Volume 1: Efficient Algorithms*, MIT Press, Cambridge, Massachusetts, 1996.

[2] M. Barbosa and P. Farshim, *Strong Knowledge Extractors for Public-Key Encryption Schemes*, in Information Security and Privacy, R. Steinfeld and P. Hawkes, eds., Berlin Heidelberg, 2010, Springer, pp. 164–181.

[3] J. M. Basilla, *On the solution of $x^2 + dy^2 = m$*, Proc. Japan Acad., 80(A) (2004), pp. 40–41.

[4] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, *Relations Among Notions of Security for Public-Key Encryption Schemes*, in Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '98), H. Krawczyk, ed., vol. 1462 of LNCS, Berlin Heidelberg, 1998, Springer, pp. 26–45.

[5] M. Bellare and A. Palacio, *Towards Plaintext-Aware Public-Key Encryption Without Random Oracles*, in Advances in Cryptology - ASIACRYPT 2004, P. J. Lee, ed., Berlin, Heidelberg, 2004, Springer, pp. 48–62.

[6] M. Bellare and P. Rogaway, *Optimal Asymmetric Encryption*, in Advances in Cryptology – EUROCRYPT '94, A. D. Santis, ed., vol. 950 of Lecture Notes in Computer Science, Berlin Heidelberg, 1995, Springer Verlag, pp. 92–111. ISBN: 978-3-540-44717-7.

[7] J. Birkett and A. W. Dent, *Security Models and Proof Strategies for Plaintext-Aware Encryption*, J. of Cryptology, 27 (2013), pp. 139–180.

[8] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, London Mathematical Society Lecture Note Series 265, Cambridge University Press, Cambridge, United Kingdom, 1999.

[9] M. Blum, P. Feldman, and S. Micali, *Non-Interactive Zero-Knowledge and Its Applications*, in Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC '88, New York, NY, 1988, Association for Computing Machinery, pp. 103–112.

[10] ——, *Proving Security Against Chosen Ciphertext Attacks*, in Advances in Cryptology — CRYPTO' 88, S. Goldwasser, ed., New York, NY, 1990, Springer, pp. 256–268.

[11] J. Buchmann and U. Vollmer, *Binary Quadratic Forms: An Algorithmic Approach*, Springer-Verlag, Berlin Heidelberg New York, 2007.

[12] D. A. Buell, *Binary Quadratic Forms: Classical Theory and Modern Computations*, Springer Verlag New York, 1989.

[13] H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer, Berlin Heidelberg New York, 1996.

[14] H. Cohn, *Introduction to the Construction of Class Fields*, Dover Publications, Inc., New York, 1994. Unabridged, corrected republication on 1985 Cambridge University Press.

[15] D. G. Cornacchia, *Su di un metodo per la risoluzione in numeri interi dell'equazione $\sum_{h=0}^{n} c_h x^{n-h} \cdot y^h = P$*, Giornale di matematiche di Battaglini, XLVI (1908), pp. 33–90. (Pages 33–64 in Gennaio e Febbraio issue, pages 65–90 in Marzo e Aprile issue.).

[16] D. A. Cox, *Primes of the Form $x^2 + ny^2$*, John Wiley & Sons, Inc., Hoboken, New Jersey, second ed., 2013.

[17] R. Cramer and V. Shoup, *A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack*, in Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '98), vol. 1462 of LNCS, Berlin Heidelberg, 1998, Springer-Verlag, p. 13–25.

[18] Danny Dolev, Cynthia Dwork, and Moni Naor, *Non-Malleable Cryptography*, in Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing, STOC '91, New York, NY, USA, 1991, Association for Computing Machinery, pp. 542—552.

[19] A. W. Dent, *The Cramer-Shoup Encryption Scheme is Plaintext Aware in the Standard Model.* Cryptology ePrint Archive, Report 2005/261, 2005.

[20] L. Dickson, *History of the Theory of Numbers, Volume II: Diophantine Analysis*, Chelsea Publishing Company (Reprinted by the American Mathematical Society), Providence, Rhode Island, 2000.

[21] M. Fischlin, *Completely Non-Malleable Schemes*, in Proceedings of the 32nd International Conference on Automata, Languages and Programming, ICALP'05, Berlin Heidelberg, 2005, Springer-Verlag, pp. 779—-790.

[22] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern, *RSA-OAEP Is Secure under the RSA Assumption*, J. of Cryptology, 17 (2004), p. 81–104.

[23] S. Goldwasser and S. Micali, *Probabilistic encryption*, J. Comput. Syst. Sci., 28 (1984), pp. 270–299.

[24] S. Goldwasser, S. Micali, and R. L. Rivest, *A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks*, SIAM J. Comput., 17 (1988), pp. 281–308.

[25] D. Hofheinz and E. Kiltz, *Practical Chosen Ciphertext Secure Encryption from Factoring*, in Advances in Cryptology - EUROCRYPT 2009, A. Joux, ed., vol. 5479 of Lecture Notes in Computer Science, Berlin, Heidelberg, 2009, Springer, pp. 313–332.

[26] J. C. Lagarias and A. M. Odlyzko, *Effective versions of the Chebotrev density theorem*, in Algebraic Number Fields, A. Frölich, ed., 1977, pp. 409–464.

[27] F. Lemmermeyer, *Reciprocity Laws: From Euler to Eisenstein*, Springer, Berlin Heidelberg New York, 2000.

[28] J. E. Littlewood, *On the Class-Number of the Corpus $P(\sqrt{-k})$*, Proc. London Math. Soc., s2-27 (1928), pp. 358–372.

[29] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, LLC, a division of Taylor & Francis (a subsidiary of Informa), Boca Raton, Florida, 1997.

[30] W. D. Nordhaus, *Two Centuries of Productivity Growth in Computing*, J. Econ. Hist., 67 (2007), p. 128–159.

[31] C. Rackoff and D. Simon, *Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack*, in CRYPTO 1991, J. Feigenbaum, ed., vol. 576 of LNCS, Berlin Heidelberg, 1992, Springer, p. 433–444.

[32] P. Rauzy and S. Guilley, *Countermeasures against high-order fault-injection attacks on crt-rsa*, in 2014 Workshop on Fault Diagnosis and Tolerance in Cryptography, Busan, South Korea, D. Choi and A. Tria, eds., IEEE Xplore Digital Library, September 23, 2014, pp. 68–82.

[33] Sankar Das Sarma, *Quantum computing has a hype problem*, MIT Technology Review, March 28 (2022).

[34] R. Schoof, *Counting points on elliptic curves over finite fields*, J. Théor. Nombres Bordeaux, 7 (1995), pp. 219–254.

[35] J.-P. Serre, *Cours D'Arithmétique*, Presses Universitaires de France., Paris, 1970.

[36] ——, *A Course in Arithmetic*, Springer, New York Berlin Heidelberg, 1973.

[37] H. J. S. Smith, *De compositione numerorum primorum formae* $4\lambda + 1$ *ex duobus qudratis.*, J. Reine Angew. Math., 50 (1855), pp. 91–92.

[38] P. Stevenhagen and H. Lenstra, Jr., *Chebotarëv and his Density Theorem*, Math. Intelligencer, 18 (1996), pp. 26–37.

[39] Z.-H. Sun, *On the quadratic character of quadratic units*, J. Number Theory, 128 (2008), pp. 1295–1335.

[40] M. Y. Vardi, *Quantum Hype and Quantum Skepticism*, Comm. ACM, 62 (May 1999), p. 7.

[41] H. Weber, *Lehrbuch der Algebra*, Braunschweig: Vieweg, 1898-1908.

[42] K. Williams, K. Hardy, and C. Friesen, *On the evaluation of the Legendre-symbol* $\left(\frac{a+b\sqrt{m}}{p}\right)$, Acta Arith., 45 (1985), pp. 255–272.

Appendix — Proofs of Theorems 8 and 9

We first need some basic definitions and two rational quartic reciprocity laws: Frölich's reciprocity law and Scholt's reciprocity law. (Interestingly, there is a more general rational reciprocity law that encompasses these two laws, plus Burde's law. In fact all of the quartic reciprocity laws in [27] can be proven from the 1985 quartic reciprocity law discovered by K. S. Williams, K. Hardy and C. Friesen [42].)

Definition 24. The multiplicative group $(\mathbb{Z}/n\mathbb{Z})^* = \{1 \leq a < n \mid \gcd(a,n) = 1\}$.

Definition 25 (Euler *phi* function)**.**

$$\varphi(n) = |(\mathbb{Z}/n\mathbb{Z})^*| = \sum_{\substack{1 \leq k < n \\ \gcd(k,n)=1}} 1.$$

Definition 26 (Rational quartic residue symbol)**.** For q and p odd primes and $q \equiv 1 \pmod 4$, and $\left(\frac{p}{q}\right) = 1$. Then

$$\left(\frac{p}{q}\right)_4 = \begin{cases} +1, \text{ if } \left(\frac{\sqrt{p}}{q}\right) = +1, \\ -1, \text{ if } \left(\frac{\sqrt{p}}{q}\right) = -1. \end{cases}$$

For $m = \prod_{i=1}^{t} p_i$, $p_i \equiv 1 \pmod 4$, prime, and $\left(\frac{p}{p_i}\right) = 1$, then

$$\left(\frac{p}{m}\right)_4 = \prod_{i=1}^{t} \left(\frac{p}{p_i}\right)_4.$$

For $r = 2^k p$, with p and q as above and $\left(\frac{2}{q}\right) = 1$, then

$$\left(\frac{r}{q}\right)_4 = \left(\frac{\sqrt{2}}{q}\right)_2^k \cdot \left(\frac{p}{q}\right)_4.$$

Definition 27 (Fundamental Unit [27, pages 44, 71, 97–98])**.** For $K = \mathbb{Q}(\sqrt{m})$, $m > 0$, the set $E_K = O_K^\times$ of invertible elements in O_K is the unit group. It is generated by $\langle -1, \varepsilon_m \rangle$, where ε_m is the fundamental unit and equals $(x + y\sqrt{m})/2$, where (x, y) is the minimal solution to $X^2 - mY^2 = \pm 4$.

When m is a prime $\equiv 1 \pmod 4$, then the norm of ε_m is -1. When $m \equiv 1 \pmod 4$, then the integers of O_K are $\mathbb{Z} \oplus \mathbb{Z}\frac{1+\sqrt{m}}{2}$.

Theorem 24 (Frölich's Reciprocity Law [27, page 173])**.** *Let $m = a^2 + b^2 = \prod_{i=1}^{t} p_i^{e_i}$, where each prime p_i is congruent to* 1 *modulo* 4*, and b is even. Let q be a prime number equal to $c^2 + d^2$, with d even, and q a quadratic residue for each prime factor of m. Let $i_m = \sqrt{-1}$ (mod m) and $i_q = \sqrt{-1}$ (mod q), then*

$$\left(\frac{m}{q}\right)_4 \left(\frac{q}{m}\right)_4 = \left(\frac{a + bi_q}{q}\right) = \left(\frac{a - bi_q}{q}\right) = \left(\frac{c + di_m}{m}\right) = \left(\frac{c - di_m}{m}\right).$$

Normally this theorem is given in its more basic form with m replaced by a prime number p congruent to 1 modulo 4 with $\left(\frac{p}{q}\right) = 1$.

Theorem 25 (Scholz's Reciprocity Law [27, pages 160–161])**.** *Let p and q be different primes congruent to* 1 *modulo* 4 *such that $\left(\frac{p}{q}\right) = 1$, and ε_p and ε_q the fundamental units of $\mathbb{Q}(\sqrt{p})$ and $\mathbb{Q}(\sqrt{q})$, respectively, then*

$$\left(\frac{\varepsilon_q}{p}\right) = \left(\frac{p}{q}\right)_4 \left(\frac{q}{p}\right)_4 = \left(\frac{\varepsilon_p}{q}\right).$$

Supplement

For notational consistency, we define for a prime $q \equiv 1$ (mod 8),

$$\left(\frac{q}{2}\right)_4 = (-1)^{(q-1)/8}, \text{ thus}$$

$$\left(\frac{q}{2}\right)_4 = \left(\frac{\sqrt{i}}{q}\right) = \left(\frac{i}{q}\right)_4.$$

If $q \equiv 9$ (mod 16), then

$$\left(\frac{2}{q}\right)_4 \left(\frac{q}{2}\right)_4 = -\left(\frac{\sqrt{2}}{q}\right) = \left(\frac{1+\sqrt{2}}{q}\right) = \left(\frac{\varepsilon_2}{q}\right).$$

If $q \equiv 1 \pmod{16}$, then

$$\left(\frac{2}{q}\right)_4 \left(\frac{q}{2}\right)_4 = +\left(\frac{\sqrt{2}}{q}\right) = \left(\frac{1+\sqrt{2}}{q}\right) = \left(\frac{\varepsilon_2}{q}\right).$$

Theorem 26. *For every prime $q = x^2 + 8py^2$, with y odd, p a odd prime $= a^2 + b^2$, with a odd and b even, set*

$$J_1 = \left(\frac{\varepsilon_2}{q}\right), \text{ where } \varepsilon_2 = 1+\sqrt{2}, \text{ and}$$

$$J_2 = \left(\frac{a+\sqrt{p}}{q}\right), \text{ then } J_1 \cdot J_2 = -1.$$

In addition,

$$J_1 = \left(\frac{1-\sqrt{2}}{q}\right) = \left(\frac{1 \pm i}{q}\right), \text{ and}$$

$$J_2 = \left(\frac{a-\sqrt{p}}{q}\right) = \left(\frac{b \pm \sqrt{p}}{q}\right) = \left(\frac{a \pm bi}{q}\right) = \left(\frac{b \pm ai}{q}\right).$$

All of the above ± signs can be chosen independently.

Proof. Since $q \equiv 1 \pmod 8$, we have $\left(\frac{-1}{q}\right) = \left(\frac{2}{q}\right) = 1$. Now $(1+\sqrt{2})(1-\sqrt{2}) = 1-2 = -1$, so $\left(\frac{1+\sqrt{2}}{q}\right) = \left(\frac{1-\sqrt{2}}{q}\right)$. Likewise, $\left(\frac{1+i}{q}\right) = \left(\frac{1-i}{q}\right)$, as $(1+i)(1-i) = 2$. The equivalence of $\left(\frac{1\pm\sqrt{2}}{q}\right)$ and $\left(\frac{1\pm i}{q}\right)$ follows from the identity $(1+\sqrt{2}+i)^2 = 2(1+\sqrt{2})(1+i)$.

Next, $(a-\sqrt{p})(a+\sqrt{p}) = a^2 - p = -b^2$ so $\left(\frac{a+\sqrt{p}}{q}\right) = \left(\frac{a-\sqrt{p}}{q}\right)$. Likewise, $(b+\sqrt{p})(b-\sqrt{p}) = b^2 - p = -a^2$, so $\left(\frac{b+\sqrt{p}}{q}\right) = \left(\frac{b-\sqrt{p}}{q}\right)$. The equivalence of $\left(\frac{a+\sqrt{p}}{q}\right)$ with $\left(\frac{b+\sqrt{p}}{q}\right)$ follows from the the identity $(a+b+\sqrt{p})^2 = 2(a+\sqrt{p})(b+\sqrt{p})$.

The equivalence of $\left(\frac{a+bi}{q}\right)$ with $\left(\frac{a-bi}{q}\right)$ follows from $\left(\frac{a+bi}{q}\right) \cdot \left(\frac{a-bi}{q}\right) = \left(\frac{a^2+b^2}{q}\right) = \left(\frac{p}{q}\right) = 1$. The equivalence of $\left(\frac{a+bi}{q}\right)$ with $\left(\frac{a+\sqrt{p}}{q}\right)$ follows from $(a+\sqrt{p}+bi)^2 = 2(a+\sqrt{p})(a+bi)$. Likewise for $\left(\frac{b+\sqrt{p}}{q}\right)$.

To show that $J_1 \cdot J_2 = -1$, we first reduce $q = x^2 + 8py^2$ modulo q and modulo p.[22] We now have

$$\frac{x^2}{y^2} \equiv -8p \pmod{q} \text{ and } q \equiv x^2 \pmod{p}, \text{ thus}$$

$$\left(\frac{p}{q}\right)_4 = \left(\frac{-8}{q}\right)_4 \left(\frac{xy}{q}\right) = \left(\frac{2i\sqrt{2}xy}{q}\right) = \left(\frac{\sqrt{2}xy}{q}\right), \text{ and} \left(\frac{q}{p}\right)_4 = \left(\frac{x}{p}\right).$$

[22] This part of the proof uses similar techniques as the proof of Proposition 5.12 (E. Lehmer) in [27, page 163]. Also see reference 508 and pages 449, 163 and 173 in [27].

Next, we apply Frölich's reciprocity law, and get

$$\left(\frac{a+bi}{q}\right) = \left(\frac{q}{p}\right)_4 \left(\frac{p}{q}\right)_4 = \left(\frac{\sqrt{2}xy}{q}\right)\left(\frac{x}{p}\right).$$

Now, $8py^2 \equiv q \pmod{x}$ and $q \equiv x^2 \pmod{y}$, and since x and y are both odd, we have

$$\left(\frac{2p}{x}\right) = \left(\frac{q}{x}\right) = \left(\frac{x}{q}\right) = \left(\frac{2}{x}\right)\left(\frac{x}{p}\right), \text{ so}$$
$$\left(\frac{x}{p}\right)\left(\frac{x}{q}\right) = \left(\frac{2}{x}\right), \text{ and}$$

$$\begin{aligned}\left(\frac{a+bi}{q}\right) &= \left(\frac{\sqrt{2}xy}{q}\right)\left(\frac{x}{p}\right) = \left(\frac{2}{x}\right)\left(\frac{\sqrt{2}y}{q}\right)\\ &= \left(\frac{2}{x}\right)\left(\frac{\sqrt{2}}{q}\right)\left(\frac{q}{y}\right) = \left(\frac{2}{x}\right)\left(\frac{\sqrt{2}}{q}\right).\end{aligned} \tag{8}$$

Now $x^2 \equiv 1 \pmod 8$ so we have two cases to consider: when $x^2 \equiv 9 \pmod{16}$ and $x^2 \equiv 1 \pmod{16}$. We have

$x^2 \pmod{16}$		$x \pmod 8$		$\left(\frac{2}{x}\right)$		$q \pmod{16}$
$\equiv 9$	$\Leftrightarrow$	$\equiv 3, 5$	$\Leftrightarrow$	$= -1$	$\Leftrightarrow$	$\equiv 1$
$\equiv 1$	$\Leftrightarrow$	$\equiv 1, 7$	$\Leftrightarrow$	$= +1$	$\Leftrightarrow$	$\equiv 9.$

Putting this together with (8) we have

$$\left(\frac{a+bi}{q}\right) = \begin{cases} -\left(\frac{\sqrt{2}}{q}\right), & \text{when } q \equiv 1 \pmod{16},\\ +\left(\frac{\sqrt{2}}{q}\right), & \text{when } q \equiv 9 \pmod{16}.\end{cases}$$

Now, with the supplement to Scholz's reciprocity law (Theorem 25), namely,

$$\left(\frac{\sqrt{2}}{q}\right) = \begin{cases} \left(\frac{\varepsilon_2}{q}\right), & \text{when } q \equiv 1 \pmod{16}, \text{ and}\\ -\left(\frac{\varepsilon_2}{q}\right), & \text{when } q \equiv 9 \pmod{16},\end{cases}$$

we have

$$J_2 = \left(\frac{a+bi}{q}\right) = -\left(\frac{\varepsilon_2}{q}\right) = -J_1, \text{ and so } J_1 \cdot J_2 = -1.$$

□

Theorem 27. *For every prime $q = x^2 + 4^t \cdot 8py^2$, $t \in \mathbb{Z}_{>0}$, with y odd, p a odd prime $= a^2 + b^2$, with a odd and b even, set $J_1 = \left(\frac{\varepsilon_2}{q}\right)$, where $\varepsilon_2 = 1 + \sqrt{2}$ and $J_2 = \left(\frac{a+\sqrt{p}}{q}\right)$, then $J_1 \cdot J_2 = 1$. In addition, $J_1 = \left(\frac{1-\sqrt{2}}{q}\right) = \left(\frac{1\pm i}{q}\right)$, and $J_2 = \left(\frac{a-\sqrt{p}}{q}\right) = \left(\frac{b\pm\sqrt{p}}{q}\right) = \left(\frac{a\pm bi}{q}\right) = \left(\frac{b\pm ai}{q}\right)$. All of the above ± signs can be chosen independently.*

Proof. The proof is identical to the proof of Theorem 26 up to and including Equation (8) as 4^t is always a quadratic residue. In case of Theorem 27 we have

$x^2 \pmod{16}$		$x \pmod 8$		$\left(\frac{2}{x}\right)$		$q \pmod{16}$
$\equiv 1$	$\Leftrightarrow$	$\equiv 1, 7$	$\Leftrightarrow$	$= +1$	$\Leftrightarrow$	$\equiv 1$
$\equiv 9$	$\Leftrightarrow$	$\equiv 3, 5$	$\Leftrightarrow$	$= -1$	$\Leftrightarrow$	$\equiv 9.$

Thus, there is a sign change compared to Theorem 26 and putting this together with (8) we have

$$\left(\frac{a+bi}{q}\right) = \begin{cases} +\left(\frac{\sqrt{2}}{q}\right), & \text{when } q \equiv 1 \pmod{16}, \\ -\left(\frac{\sqrt{2}}{q}\right), & \text{when } q \equiv 9 \pmod{16}. \end{cases} \tag{9}$$

Next,

$$\left(\frac{\sqrt{2}}{q}\right) = \begin{cases} \left(\frac{\varepsilon_2}{q}\right), & \text{when } q \equiv 1 \pmod{16}, \text{ and} \\ -\left(\frac{\varepsilon_2}{q}\right), & \text{when } q \equiv 9 \pmod{16}, \end{cases} \tag{10}$$

and this time the signs are always identical in Equations (9) and (10) for $q \equiv 1 \pmod{16}$ and $q \equiv 9 \pmod{16}$, and therefore $J_1 \cdot J_2 = 1$.

□

Theorem 28. *For a prime $q = x^2 + 8dy^2$, with y and d both odd, and $d = a^2 + b^2$, b even,(d does not need to be square free and d can be composite), then*

$$\left(\frac{d}{q}\right)_4 \left(\frac{q}{d}\right)_4 = \left(\frac{a+bi}{q}\right) = -\left(\frac{\varepsilon_2}{q}\right) = -\left(\frac{1 \pm \sqrt{2}}{q}\right) = -\left(\frac{1 \pm i}{q}\right).$$

For $q = x^2 + 4^t \cdot 8dy^2$, $t \in \mathbb{Z}_{>0}$, and holding everything else the same as earlier, we have

$$\left(\frac{d}{q}\right)_4 \left(\frac{q}{d}\right)_4 = \left(\frac{a+bi}{q}\right) = \left(\frac{\varepsilon_2}{q}\right) = \left(\frac{1 \pm \sqrt{2}}{q}\right) = \left(\frac{1 \pm i}{q}\right).$$

Proof. This theorem is a generalization of Theorems 26 and 27, but instead of the constant p being an odd prime, the equivalent constant $d = a^2 + b^2$ can be an odd composite number. All that is necessary is that d is such that Frölich's reciprocity law holds, which is the case for $d = a^2 + b^2$.

If d is not square free, Theorem 28 still holds, but in the proof d needs to be square free. However, we can effectively make $d = mw^2$ square free in the proof by viewing w^2, the square term of d, as being part of the y^2 term by setting $y \leftarrow wy$ and $d \leftarrow m$. □