Cryptography based on Finite Field Isomorphisms

Jasper Liu

April 30, 2022

Abstract

Given a prime number q and a positive integer n, all finite fields of order q^n are isomorphic, and can be viewed as $F_q[x]/f(x)$, where f is some degree n irreducible polynomial mod q. As the elements of finite fields can be viewed as polynomials, they can further be considered as vectors with entries mod q, and the isomorphisms between finite fields can be treated as matrices with entries mod q. The goal of this thesis is to construct an efficient digital signature scheme based on the hardness to recover the secret isomorphism between two finite fields. The final objective is to minimize the size of public keys and signatures, as well as preserving security against combinatorial or lattice-based attacks.

Keywords: Finite Field Isomorphisms, lattice based cryptography, digital signature scheme

1 Introduction

1.1 Digital Signature

Digital signature is a crucial part in cryptography. When we talk about digital signatures, we usually use the following setup:

- Alice publishes some public keys, and keeps secret some private keys.
- Bob sends a document for Alice to sign.
- Alice uses her private keys and the document to produce a signed document (the signature).
- Bob then runs a verification function using the signature and the public keys, verifying that it's indeed Alice who produced the signature.

In this process, there will be a third person, Eve, who will try to attack the system. There will be two paths for Eve to work on. The first is to recover the secret keys given a list of signatures and the public key, and the second is to forge signatures that will pass the verification process. So the most important feature of a digital signature scheme is to make sure that the signature and public

keys don't leak information about the private key, and to ensure that it's hard to forge a new signature from previous signatures.

A digital signature scheme is said to have bit-security k if it takes the attacker (Eve) at least 2^k tries when she attacks on the signature scheme by brute force. In other words, the probability that a random try by Eve will pass the verification process is less than 2^{-k}. The signature scheme that I will introduce in this thesis is expected to have bit security of 256.

1.2 Setups

Let q be a prime and n be an integer, then any two finite fields of order q^n are isomorphic, and any finite field of order q^n can be represented as $F_q[x]/(f(x))$, where $f(x)$ is an irreducible polynomial mod q of degree n.

Let $\mathbf{X} = F_q[x]/(f(x))$ and $\mathbf{Y} = F_q[y]/(F(y))$, where $f(x)$ and $F(y)$ are two monic irreducible polynomials mod q. Elements in $\mathbf{X}, \mathbf{Y}$ are polynomials of degree less than n. For clarity we assume that the coefficients are all between $-q/2$ and $q/2$. These polynomials can also be viewed as vectors, where the entries of the vectors are just the coefficients of the polynomials (But note that the vector multiplication is defined as polynomial product mod f or F). By Finite Field Theory, $\mathbf{X}$ and $\mathbf{Y}$ are isomorphic. There exists isopmorphisms $\phi : \mathbf{X} \to \mathbf{Y}$ and $\Phi : \mathbf{Y} \to \mathbf{X}$. Note that ϕ and Φ can be represented by n by n matrices. For clarity, polynomials represented by capital letters will represent polynomials in $\mathbf{Y}$, and lowercase letters represent polynomials in $\mathbf{X}$

Further, we define a matrix γ, with dimension m by n, where m is smaller than n (For example, $m = 163$ is a good choice to ensure combinatorial security). One example of the matrix γ is a permutation of the $m \times m$ identity matrix adjoined by a block of 0's on the right.

For a vector a in $\mathbf{X}$, we define it to be short if it satisfies one of the following criteria

- the L^∞ norm of a is smaller than some β, where $(\frac{2\beta}{q})^n < 2^{-256}$
- the L^1 norm is smaller than some β, where the probability that a random polynomial in X has a smaller L^1 norm than β is smaller than 2^{-256}.

The first criteria is easy to calculate, while the second one is harder. We refer to the Irwin-Hall Distribution, that is, the cumulative density function of the sum of n independent variables uniformly distributed in $[0, 1]$ is

$$g_n(x) = \frac{1}{n!} \sum_{k=0}^{\lfloor x \rfloor} (-1)^k \binom{n}{k} (x-k)^n$$

We will manipulate this equation, and the function *UniformSumDistribution* in Mathematica will be used for the calculation. These two criteria are set to ensure combinatorial security of the signature scheme, making the probability that a random polynomial in $\mathbf{X}$ is short to be smaller than 2^{256}.

1.3 Finite Field Isomorphism Problem

The Finite Field Isomorphism Problem can be treated in two perspectives: the decisional FFI problem and the computational FFI problem. The statements of the problem are listed below

- **Computational FFI Problem:** Let $a_1, a_2, ..., a_k$ be a list of short polynomials in $\mathbf{X}$, and let $A_1, A_2, ...A_k$ be their images under isopmorphism ϕ in $\mathbf{Y}$. The computational FFI problem asks if it's possible to recover $\mathbf{X} = F_q[x]/(f(x))$ and/or $a_1, a_2...a_k$.
- **Decisional FFI Problem:** Let $a_1, a_2, ...a_k$ and $A_1, A_2, ...A_k$ be as above, and let B_1, B_2 be in $\mathbf{Y}$, with one of them being the image of a short polynomial in $\mathbf{X}$. Given the list $A_1, A_2, ...A_k$, identify which is the image with a probability larger than $1/2$.

In the digital signature scheme that I worked on, security is mainly based on the computational FFI problem. In the next section, I will introduce attacks on the Finite Field Isomorphism Problem.

2 Attacks on the FFI

In this section, I will introduce two possible attacks. The first one is building a field $\mathbf{X}'$ "similar to $\mathbf{X}$, and the second one is based on the lattice reduction problem.

2.1 Uniqueness of X

First, I will give a proof that the attacker cannot build a new field, $\mathbf{X}' = F_q[x]/f'(x)$, such that all the images in $\mathbf{Y}$ of short polynomials in $\mathbf{X}$ are also images of short polynomials in $\mathbf{X}'$. In addition, in the signature scheme that I will mention later, all signatures will be images in $\mathbf{Y}$ of polynomials with L^∞ at most 1. This means, I will prove there cannot be $\mathbf{X}$ and $\mathbf{X}'$, with isomorphism $\Phi : \mathbf{X} \to \mathbf{X}'$, such that all polynomials with entries in $-1, 0, 1$, will be mapped to polynomials with L^∞ norm at most C for some $0 < C < q/4$.

Suppose there exists two finite fields $\mathbf{X}$ and $\mathbf{X}'$, with a constant C and isomorphism $\Phi : \mathbf{X} \to \mathbf{X}'$, satisfying for any polynomial $f(x)$ with coefficients in $\{-1, 0, 1\}$ in $\mathbf{X}$, $\Phi(f(x))$ has L^∞ norm smaller than C. We denote $\mathbf{X} = F_q[x]/F(x)$ and $\mathbf{X}' = F_q[x']/G(x')$, where F and G are irreducible degree n polynomials.

By assumption, for any positive integer i, $\Phi(x^i) = \Phi(x)^i$ has L^∞ norm smaller than C. So we examine $\Phi(x)$ in $\mathbf{X}'$.

Suppose $\Phi(x)$ has degree larger than 1 in $\mathbf{X}'$. Then there exists integer k such that $deg(\Phi(x)^k) > n$ (Here let the calculation take place in $F_q[x']$). WLOG we can let k be even, or we can just take $k+1$. Then, by assumption, the L^∞ norm of $\Phi(x^{k/2})$ is smaller than C, but when we square it, modulus $G(x')$ will happen since $deg(\Phi(x^k)) > n$. This will, with a high probability, lead to a polynomial with a L^∞ norm larger than C, as C is small compared to q. Note that, more than one such k exists, so the probability that $deg(\Phi(x)) < 1$ will be further lowered.

Thus $deg(Phi(x)) = 1$. So let $\Phi(x) = ax' + b$. By assumption, $|a|, |b| < C$. If $|a| \neq 1$, then $\exists k < n$ such that $|a|^k > C$, which indicates that the L^∞ norm of $\Phi(x^k) = \Phi(x)^k = (ax' + b)^k$ will be larger than C, contradiction. The same arguments applies to b. If $|a| = |b| = 1$, we can also reach a contradiction by binomial expansion. Since $|a| \neq 0$, we have $|b| = 0$, which indicates that the only possible Φ will be $\Phi(x) = \pm x'$. If $\Phi(x) = x'$, then Φ is the identity, $F = G$. If $Phi(x) = -x'$, we can see that $F(x) = G(-x)$, which indicates that finding such an X' is as hard as solving X. This completes the proof.

2.2 Lattice Attack

This attack is is formulated as a conjecture in a paper by Hoffstein, Silverman, Doroz and Sunar[2]. The tools to calculate the difficulty of it is formulated in the paper by Chen and Nguyen [1].

Suppose we have a list of signatures $\mathbf{A}_1, \mathbf{A}_2, ...\mathbf{A}_N$, and $n \times n$ matrix M, with coefficients mod q, with the property that

$$M\mathbf{A}_i \text{ mod } q \text{ are small for all } i = 1, 2, ..., N \quad .$$

In this attack, we will try to recover some row of M. Let $\mathbf{m}$ be a row of M, and we denote

$$b_i = \mathbf{m} \cdot \mathbf{A}_i \text{ for } i = 1, 2, ..., N \quad .$$

If we have that $\mathbf{m}$ is the jth row of M, then define

$$A = (\mathbf{A}_1 | \mathbf{A}_2 | ... | \mathbf{A}_N)$$
$$\mathbf{b}_j = (b_1, b_2, ...b_k)$$

Then we can construct the matrix D

$$D = D(q, N, c, n) = \begin{pmatrix} A_1 & \cdots & A_N \\ \hline & & \\ & qI_N & \\ & & \end{pmatrix},$$

where $N \geq n$.

Consider the lattice $\mathcal{L}(D)$, which is spanned by the rows of D (i.e. the space of integer combinations of all row vectors of D). As $b_i = \mathbf{m} \cdot \mathbf{A}_i$ for $i = 1, 2, ..., N$, we have that the vector $\mathbf{b}_j$ will be a linear combination of the rows of D. Also, with a high probability, $\mathbf{b}_j$ will be significantly shorter (in L^2 norm) than any other vectors in the lattice. Thus, if we have a lattice reduction algorithm (an algorithm which will recover the shortest vector in a lattice), then we will recover $\mathbf{b}_j$. Given the construction of the matrix A, this will automatically give us $\mathbf{m}$, which is a row of the secret isomorphism.

3 Digital Signature Scheme

In this section I will first give a precise way of constructing an isomorphism between two finite fields. This method is first formulated in the paper by Hoffstein, Silverman, Doroz and Sunar[2].

Then, I will introduce the prototype of the digital signature scheme. Later in the thesis, I will go over the experiments that we have run to optimize the parameters, and to improve the signature

scheme.

3.1 Construction of the isomorphism

Recall that in the setup section, we had two finite fields, $\mathbf{X} = F_q[x]/(f(x))$, $\mathbf{Y} = F_q[y]/(F(y))$, where f and F are degree n monic irreducible polynomials mod q. Now I will describe how to find an isomorphosm between the two fields. Note that to obtain the two way isomorphisms, we need the following:

- $\phi(y) \in \mathbf{Y}$ which is the image of polynomial x in $\mathbf{X}$
- $\Phi(x) \in \mathbf{X}$ which is the image of polynomial y in $\mathbf{Y}$
- $F(y) \mid f(\phi(y))$, or in other words, $\phi(y)$ is a root of polynomial f in the finite field $\mathbf{Y}$
- Similarly, $f(x) \mid F(\Phi(x))$
- $\phi(\Phi(x)) \equiv x \pmod{f}$

In order to achieve these objectives, the following algorithm is devised in the paper by Hoffstein, Silverman, Doroz and Sunar[2].

1. Find a root of f in the finite field $\mathbf{Y} = F_q[y]/(F(y)) \simeq F_{q^n}$, and lift this to a polynomial $\phi(y) \in F_q[y]$ with degree less than n.
2. Construct a unique polynomial $\Phi(x) \in F_q[x]$ with degree less than n such that $\Phi(\phi(y)) \equiv y \pmod{F}$
3. Return $\Phi(x)$ and $\phi(y)$

In step 2, we need to find a root of f in the finite field $\mathbf{Y}$. Note that in our setup, f is monic irreducible mod q of degree n. Thus, any of its roots generates F_{q^n}. Also, given that $F_(q^n)/F_q$ is a Galois extension, any irreducible polynomial mod q with 1 root in F_{q^n} must split completely. And, as $\mathbf{X}$ and $\mathbf{Y}$ are isomorphic, we know that f must also split completely in $\mathbf{Y}$. We can set $\phi(y)$ to be any of these roots, and, there is a polynomial time algorithm to find $\phi(y)$.

In step 3, we need to construct $\Phi(x)$. In the paper by Hoffstein, Silverman, Doroz and Sunar[2], 3 methods are included. Here I will list two of them. Although both methods are efficient, the second turns out to work faster than the first one.

- Compute the roots of F in the finite field $\mathbf{X}$. Similar to step 2, there will be n distinct roots, and one of these n roots will be the desired $\Phi(x)$.
- Compute a root of $\phi(y) - x$ in the field $\mathbf{X}$.

A note on the second method: here y is simply a notation of variable. The solution to this equation will be a unique polynomial in $\mathbf{X}$.

3.2 Signature Scheme Prototype

I will use the traditional Alice-Bob-Eve notations in this paper, i.e., Bob produces a document for Alice to sign, and Eve will try to forge Alice's signature.

3.2.1 Public Key

Alice publishes the field $\mathbf{Y} = F_q[y]/(F(y))$ and the matrix $\gamma\Phi$. Note that this is the product of the two matrices in the setup, and this matrix will not give information about the isomorphism.

3.2.2 Private Key

Alice keeps secret $\mathbf{X} = F_q[x]/(f(x)), \phi$, and Φ.

3.2.3 Signing

For a document $message$ to sign we apply the $Hash$ function to it such that $Hash(message)$ is a m dimensional vector of 0's and ± 1's. Recall that m is the row-dimension of the matrix γ. Then, Alice finds a vector a in **X**, satisfying:

- $\gamma(a) \equiv Hash(message) \pmod 3$
- a has 163 coefficients taking values from $-1, 0, 1$. All other coefficients of a are 0.

Alice then publishes $A = \phi(a)$ in **Y** as the signature.

3.2.4 Verification

For Bob to verify the signature, he checks:

- $\gamma\Phi(A) \equiv Hash(message) \pmod 3$
- $\gamma\Phi(A)$ is short in L^∞ norm
- $\gamma\Phi(A^2)$ is also short.in L^∞ norm.

4 Experiments and Optimization of the Prototype

First of all, we discuss the necessity of checking $\gamma\Phi(A^2)$ is short. As Eve can always solve the linear equation $\gamma\Phi(A) = Hash(m)$, the solution to this equation will automatically satisfy the first two conditions. However, according to our experiments using Mathematica, the square of a polynomial in **X** is short if and only if the polynomial itself is very short. This fact guarantees the security of the scheme. However, this is an observation confirmed by extensive experiments, but proving this remains to be an open question.

In this section, I will discuss the experiments I have done to optimize various factors, such that the size of signatures and public keys can be optimized (as small as possible).

In the following subsections, we will forget about q first, and take a look at the behavior of norms of the square of a short polynomial mod f in $Z[x]$. The norms will be significantly smaller when we consider then mod q, but the general pattern will remain the same.

4.1 Verification and Public Keys

As mentioned earlier, we have the conjecture that the square of a polynomial (mod f) is short (in either L^1 or L^∞ norm) if and only if the polynomial itself is very short. In our case, it must be that the polynomial's coefficient list consists only with $-1, 0, 1$ such that its square can be short. Here, we make a stronger argument:
Given a short and sparse polynomial c (the coefficients consists only of $-1, 0, 1$ and there are way more $0's$ than ± 1's), then for a polynomial a, the product ac is short if and only if a is very short itself, consisting of 0's and ± 1's.

Similar to the original conjecture, this is verified by mass experiments. However, we do not know approaches to rigorously prove it. Using this conjecture, we can add in several "verification polynomials" in the public key, and in the verification process, instead of testing whether the square of the signature is short, Bob would calculate the products of the signature and the verification polynomials. Suppose there are k verification polynomials, then we need the probability that a random polynomial would pass the test of each verification polynomial to be less than $2^{-256/k}$. In this way, the prime q needed will be further reduced for each n.

There is also a requirement on the verification polynomials. We set a verification polynomial to have d 1's and d (-1)'s in its coefficient list, where d is the smallest integer such that

$$\binom{n}{d}\binom{n-d}{d} \geq 2^{256}$$

With this criteria, the probability that someone would randomly discover a verification polynomial is less than 2^{-256}, which can be considered minimal.

4.2 Setting up Criteria for Short Polynomials

Let β_0 and β_1 be the criteria for short polynomials with respect to L^∞ and ^{1} norms, i.e., we classify a polynomial to be short in the verification process if its product with all the verification polynomials has L^∞ norms smaller than β_0, or smaller L^1 norms smaller than β_1. The most trivial way to set up the β's is to find a value such that the products of short polynomials and verification polynomials will always have L^∞ and L^1 norm smaller than the corresponding β's, while the probability that a random polynomial has norm smaller than this value is close to 0. However, we can improve this, by setting β's to be values such that about 50 percent of short polynomials, when multiplied with each of the verification polynomials, will have a small norm, and the probability that a random polynomial has smaller L^1 norm is smaller than 2^{-256}. In this way, Alice would have to verify if the signature she

produces is indeed short in the signing process, but it wouldn't be too hard as there will be $n - m$ solutions to the equation $\gamma x = Hash(message)$, and approximately half of them would satisfy the criteria. In this way, we can further reduce q needed for each specific n.

4.3 Choice of f

In this section we will discuss how the choice of f may affect the L^∞ norm results. According to the paper by Hoffstein, Silverman, Doroz and Sunar[2], polynomials with complex roots of smaller norm will be more suitable choices for f, i.e. they will lead to smaller norms for products of short polynomials.

As this result is already verified in the paper by Hoffstein, Silverman, Doroz and Sunar[2], I only did a simple verification for $n = 300$, in the following way:

- Construct 10 verification polynomials satisfying the criteria mentioned in Section 4.1. Then, construct 100 testing signature polynomials, each with 163 random coefficients chosen randomly in $-1, 0, 1$ and all other coefficients 0.
- Set up 50 different irreducible f's of the form $x^n + \sum_{i=0}^{162} a^i x^i$.
- for each f, calculate the maximum of the norms of all its complex roots
- Calculate the product of verification polynomials and testing signature polynomials mod f. Record the coefficient lists as vectors, and calculate the L^∞ and L^1 norms of these vectors. The mean and median of these norms are recorded.
- Plot the mean and median of the norms with respect to the maximum of the the norms of the polynomials' complex roots.

The results are recorded in Figures 1-4.

It can be seen from the results that, as the maximum norm of the complex roots of f decreases, the resulting L^1 and L^∞ norms are smaller. This relation is not linear. However, if we choose an f with maximum norm of complex roots smaller than 1.016, the norms of products of short polynomials mod f will, with a high probability, be smaller than the results from an f with maximum roots larger than 1.017. In the following sections of this paper, we refer the f's with roots of small maximum norms as "carefully chosen".

4.4 Choice of Norm

In this section, I will discuss the experiment used to determine whether L^1 or L^∞ norm would be a better measure in determining whether a polynomial is short. Recall that the bit-length of a signature is $n \log_2 q$, so given any degree number n, the measure that requires a smaller prime q should be adopted.

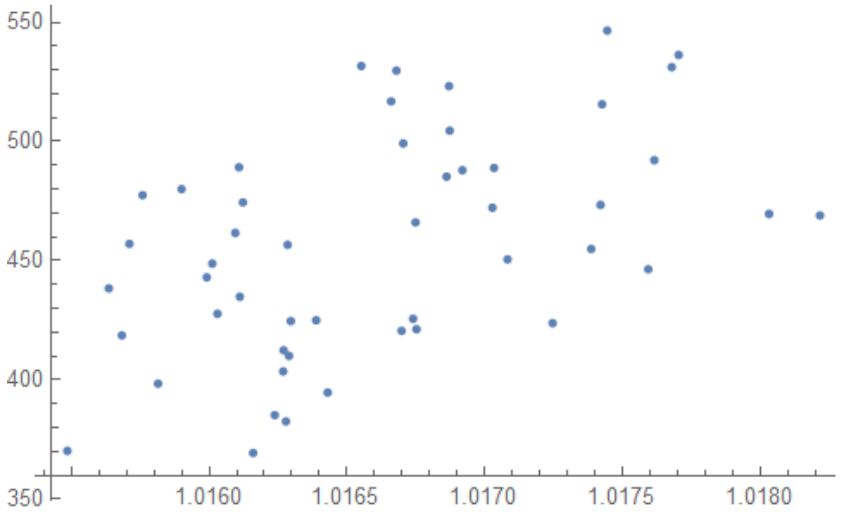

Figure 1: mean of L^∞ norms

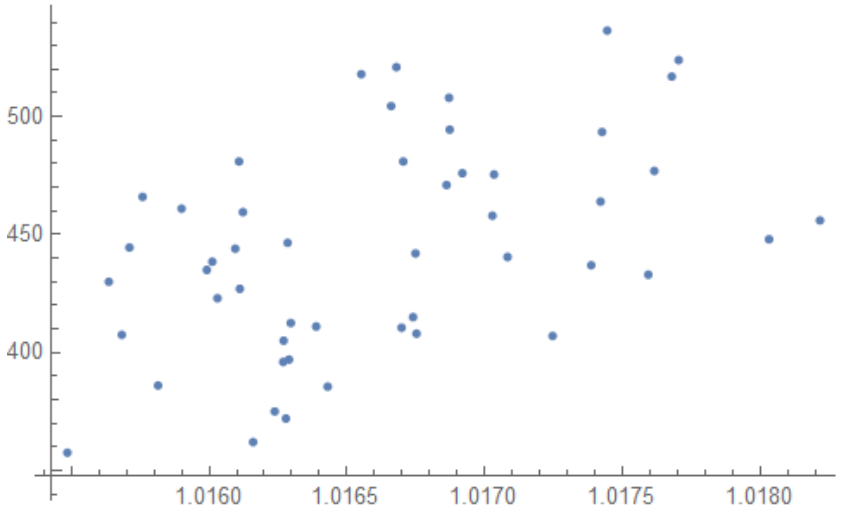

Figure 2: median of L^∞ norms

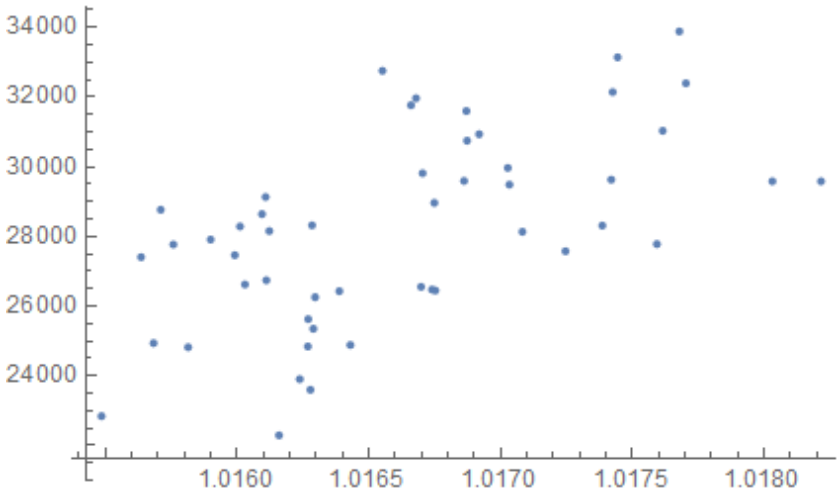

Figure 3: mean of L^1 norms

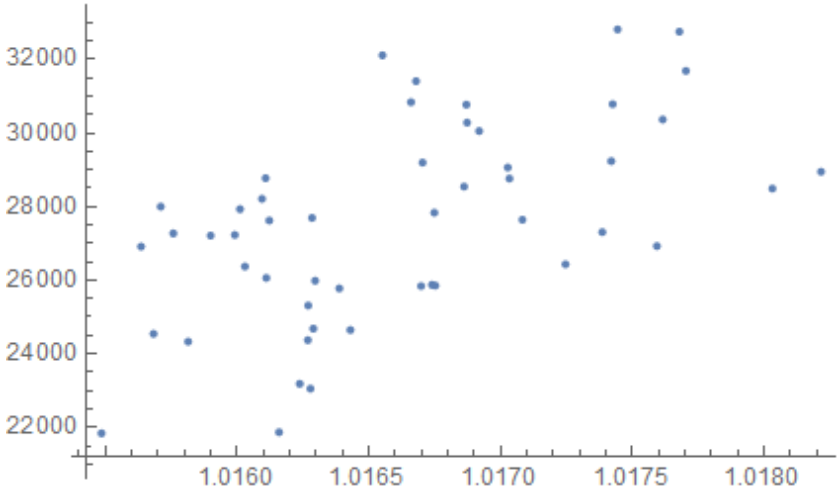

Figure 4: median of L^1 norms

4.4.1 Design of Experiment

The experiment is designed for n's ranging from 280 to 565, in intervals of 10. For each n, we conduct the following:

- Construct 50 monic irreducible polynomials, each of the form $x^n + \sum_{i=0}^{162} a_i x^i$, where a_i are all uniformly distributed in $-1, 0, 1$. Then, find the polynomial f whose complex roots have smallest maximum root. Fix this polynomial f with the corresponding n.
- Construct 8 verification polynomials satisfying the criteria mentioned in Section 4.1. Then, construct 250 testing signature polynomials, each with 163 random coefficients chosen randomly in $-1, 0, 1$ and all other coefficients 0.
- Take the products of testing polynomials and verification polynomials mod f. The coefficient list of each product is then viewed as a n-dimensional vector, and the L^∞ and L^1 norms of these 2000 vectors are recorded.
- Then, take the 160th largest L^∞ norm and L^1 norms, and use them as criteria for short polynomials for the corresponding n's.
- Use the methods discussed below to evaluate the prime q needed for each n using L^∞ and L^1 norms.

Note: There are 8 verification polynomials in our case. Recall that in section 4.1, we stated that the probability that a short polynomial would pass the verification should be at least 50%. As the 160th largest norm is the 92% cutoff of the norms of products, and $0.92^5 \approx 0.5$, it's set as the criteria for short polynomials.

4.4.2 Formulas to calculate q

For any fixed n, let β_0 and β_1 be the criteria obtained from the experiment for L^∞ and L^1 norms, respectively. Then, as there are 8 verification polynomials, we would need the probability that a random polynomial pass the test of each verification polynomial to be $(2^{-256})^{1/8} = 2^{-32}$

We want prime q_0 such that a random polynomial mod q_0 has probability less than 2^{-32} to have a smaller L^∞ norm than β_0. As the coefficient list of a random polynomial mod q_0 can be viewed as a n dimensional vector with each coefficient uniformly distributed in $[-\left\lfloor \frac{q_0}{2} \right\rfloor, \left\lfloor \frac{q_0}{2} \right\rfloor]$. Then, each coefficient has a probability of $\frac{\beta_0}{\left\lfloor \frac{q_0}{2} \right\rfloor}$ to be smaller than β_0, and the probability that the vector has a smaller L^∞ norm than β_0 is then $(\frac{\beta_0}{\left\lfloor \frac{q_0}{2} \right\rfloor})^n = (\frac{2\beta_0}{q_0 - 1})^n$. We want q_0 to be the smallest prime such that the probability is smaller than 2^{-32}, that is,

$$q_0 > 2^{\frac{32}{n}+1}\beta_0$$

For L^1 norms, we want a prime q_1 such that a random polynomial mod q_1 has probability less than 2^{-32} to have a smaller L^1 norm than β_1. Recall that in section 1.2, we introduced the cumulative

density function of the Irwin-Hall distribution with $x \in [0, n]$:

$$g_n(x) = \frac{1}{n!} \sum_{k=0}^{\lfloor x \rfloor} (-1)^k \binom{n}{k} (x-k)^n$$

which is the distribution of sum of n variables uniformly distributed in $[0, 1]$. Let x_0 be the solution to $g_n(x_0) = 2^{-32}$. Then, consider the sum of n independent variables each distributed uniformly in $[o, \lfloor \frac{q_1}{2} \rfloor]$. $x_0 \lfloor \frac{q_1}{2} \rfloor$ would give value of 2^{-32} for the cumulative density function of this distribution. Then, we have that q_1 should be the smallest prime satisfying

$$x_0 \left\lfloor \frac{q_1}{2} \right\rfloor > \beta_1$$

As q_1 is odd, this is equivalent to $x_0 \frac{q-1}{2} > \beta_1$, and thus

$$q_1 \geq \frac{2\beta_1}{x_0} + 1$$

4.4.3 Results and Analysis

The results are shown in Figure 5, 6, and 7. Figure 5 and 6 plots the β_1 and β_0, respectively for different n's. Figure 7 plots the calculated primes q_0's and q_1's for different n's, using the methods mentioned in the previous section. The yellow line represents q_0's while the blue line represents q_1's. It can be seen from this experiment that L^1 norms would allow for much smaller prime q's for a fixed n, which will make $n \log_2 q$ smaller. So we will adopt L^1 norms to measure short polynomials in Section 5 of this paper.

Note that, this experiment is quite coarse and does not determine the final choices of q's for each n. If the coefficient lists of products of polynomials mod f are taken mod q before the norms are calculated, the results may be different. The final choices of (n, q) pairs will be determined in Section 5.

5 Updated Signature Scheme and Optimization

In this section I will write up the new signature scheme after all the optimizations listed above. After that, I will introduce the numerical experiments used to determine (n, q) pairs.

5.1 Signature Scheme

5.1.1 Updated Public Key

- the field $\mathbf{Y}$
- the matrix $\gamma\Phi$

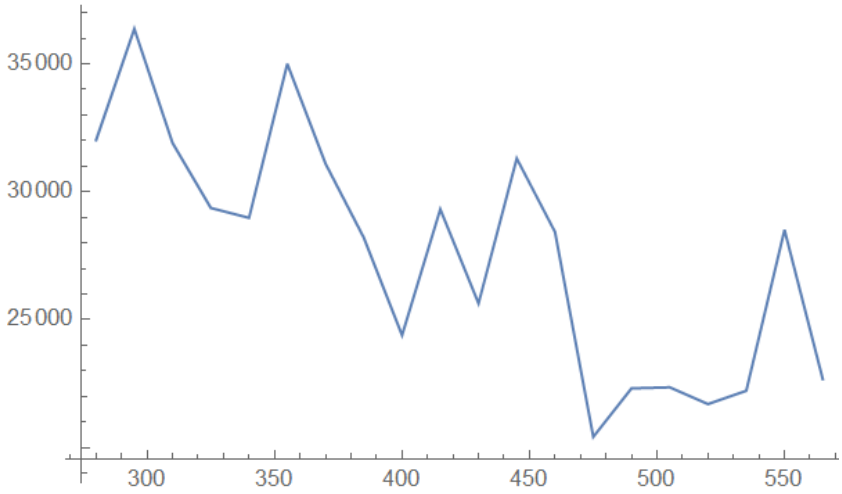

Figure 5: Cutoff of L^1 norms

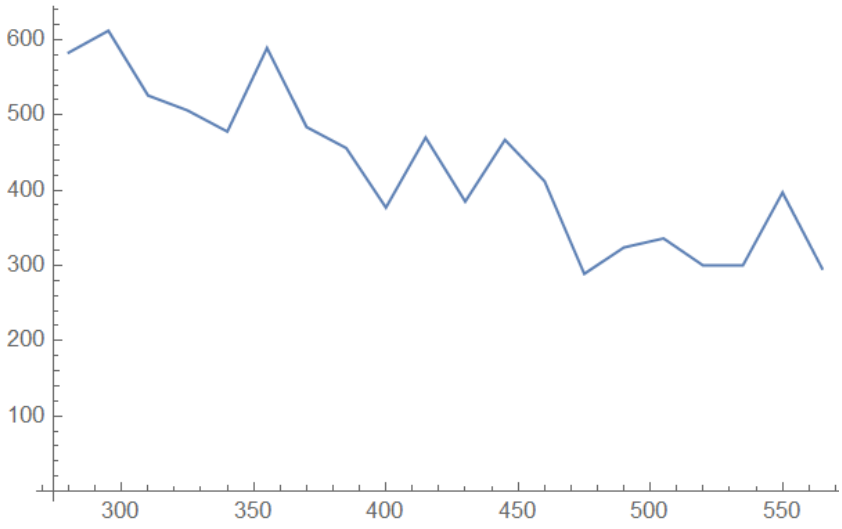

Figure 6: Cutoff of L^∞ norms

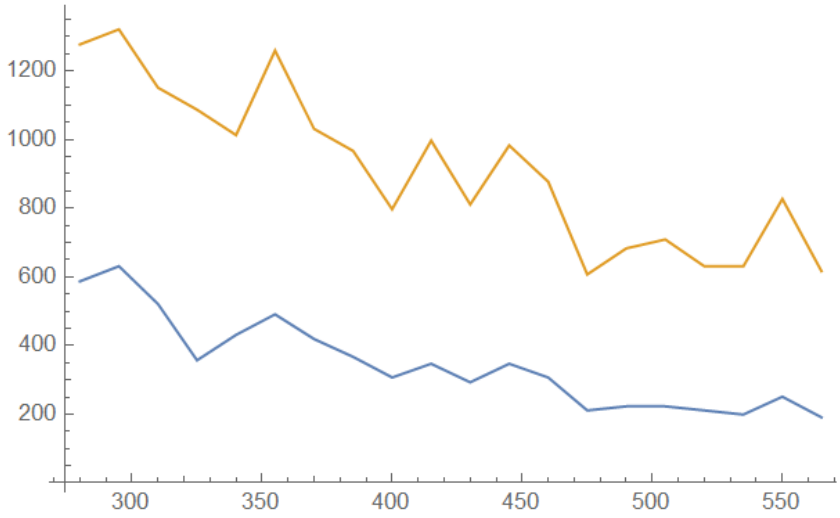

Figure 7: Primes needed for L^1 and $L^\infty norms$

- the list of images of short polynomials C_i's. The preimage of each C_i, denoted by c_i, will be polynomials in X space with d coefficients from 1 and d coefficients -1, and rest 0. Here d is the smallest value such that $\binom{n}{d}\binom{n-d}{d} > 2^{128}$. This is set to make the verification polynomials combinatorically hard to solve.

5.1.2 Private Key

Alice keeps secret $\mathbf{X} = F_q[x]/(f(x)), \phi$, and Φ, together with the c_i's.

5.1.3 Signing

For a document $message$ to sign we apply the $Hash$ function to it such that $Hash(message)$ is a m dimensional vector of 0's and ± 1's. Recall that m is the row-dimension of the matrix γ. Then, Alice finds a vector a in **X**, satisfying:

- $\gamma(a) \equiv Hash(message) \pmod 3$
- $\gamma(a * c_i)$ is indeed short for each i
- a has 163 coefficients taking values from $-1, 0, 1$. All other coefficients of a are 0.

5.1.4 Updated Verification

For a signature A, Bob verifies:

- $\gamma\Phi(A) \equiv Hash(m) \pmod 3$
- $\gamma\Phi(A)$ is short in L^1 norm
- $\gamma\Phi(A * C_i)$ is short for all i, where $*$ is polynomial multiplication.

5.2 Experiment on different f's and Resulting q

Here I will write about the experiments I did for thickening f, and the impact it has on the choice of n and q.

5.2.1 Experiment

For each n and for $c = 1$, $c = 2$, and $c = 3$, I constructed $f = f_0 + x^n$, where $f_0 = \sum_{i=1}^{r} a_i x^i$, with r being the smallest integer satisfying $(2c+1)^r > 2^{256}$, and a_i is chosen randomly from integers in range $[-c, c]$. Then a list of 20 testing polynomials are constructed. Such a polynomial has degree n, and has d 1's and d -1's, with d being the smallest integer such that $\binom{n}{d}\binom{n-d}{d} > 2^{128}$.

100 signature polynomials are then constructed. One such polynomial has 163 coefficients from $\{-1, 0, 1\}$ and 0 for the rest. A prime q around 1000 is chosen. The 2000 products of test polynomials and signature polynomials are calculated. The products are reduced mod f, and then coefficients are reduced mod q and modified to the $[-q/2, q/2]$ interval. The L^1 norms of the 2000 modified vectors are calculated, and I took the 160th largest norm as the L^1 norm cutoff. Call this cutoff B.

The probability that sum of n random polynomials uniformly distributed between $[-q/2, q/2]$ is smaller than B is calculated. We need this probability to be approximately 2^{-32}, and we switch to a larger q if this probability is too large, and smaller q if the probability is too small, until we get the optimal choice of q.

5.2.2 Some Explanations

According to previous experiments, L^1 norm is a better criteria for testing, as it allows for smaller prime q's for a given n.

As explained in the previous sections, there will be 8 test polynomials, such that $2^{-32\times 8} = 2^{-256}$. This is the reason to set the 160th largest norm as B, as for each testing polynomial, 92 percent of signatures will pass the test, and $0.92^8 > 0.5$, which means it won't be too hard for Alice to construct a valid signature.

In the results below, I have included the optimal (n, q) pairs . The L^2 norms of the products mod f are also recorded. However I didn't include those data here, as they are used solely for calculating HRF's.

Another thing to mention is that, in the real signature scheme, the L^1 norms are measured after the product vectors mod f are multiplied by the $m \times n$ matrix γ. Recall that γ can be treated as a permutation of the $m \times m$ identity matrix adjoined by columns of zeroes. So we did numerical experiments, calculating the sum of the absolute value of the first m coefficients in the product vector, and the result turns out to be approximately m/n of the L^1 norm of the vector. This experiment verifies that the (n, q) pair we obtained from this experiment is justified in the signature scheme.

5.2.3 Results

The following chart illustrates how the thickness (c, as mentioned on the previous page) would affect the choice of (n, q) pairs

n	q for $c = 1$	q for $c = 2$	q for $c = 3$
210	3413	907	733
220	1987	733	601
230	1291	733	541
240	829	631	463
250	761	439	409
260	659	349	313
270	541	349	281
280	439	313	281
290	379	257	257
300	313	257	229
310	293	229	197
320	281	197	197
330	257	191	197
340	223	173	199
350	197	163	173
370	181	163	173
390	149	137	173
410	131	113	151
430	113	109	139
450	103	109	131

It can be seen that a larger c behaves better when n is very small, but the corresponding q turns out to be similar for n large enough. To find the optimal choice of q, we will first check the security on all these (n, q) pairs.

6 Analysis of Attacks

Recall that in earlier sections, we talked about the lattice attack on the FFI problem, using the lattice generated by the rows of

$$D = D(q, N, c, n) = \begin{pmatrix} \underline{A_1 \quad \cdots \quad A_N} \\ \\ qI_N \\ \\ \end{pmatrix},$$

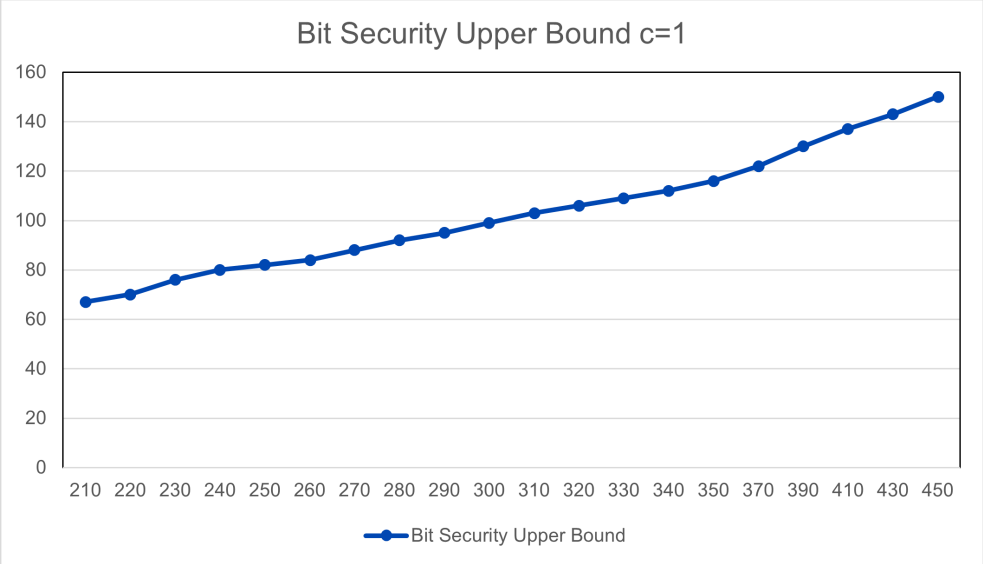

Figure 8:

where $N \geq n$. So each signature $A_i \in Y$ is the image of a polynomial $a_i \in X$, whose coefficients can be viewed as uniformly and randomly distributed between -1 and 1. This means that $\det D \approx q^{N-n}$, and the shortest vector has L^2 norm

$$\lambda_1(D) \approx \left(\frac{N}{3}(0)^2 + \frac{2N}{3}(1)^2\right)^{1/2} = \left(\frac{2N}{3}\right)^{1/2}.$$

This means that the HRF for D is

$$HRF = HRF(q, N, c, n) = \frac{\lambda_1(D)^{1/N}}{(\det D)^{1/N^2}} = \frac{\left(\frac{2N}{3}\right)^{1/2N}}{q^{(N-n)/N^2}}.$$

This calculation is based on the formulae given in the paper by Chen and Nguyen [1].

for each (n, q) pair, we computed the least $N > n$ for which $\lambda_1(D) <$ Gaussian expected SV, because if the attacker wants a good chance of getting $\lambda_1(D)$ (as this would give a slice of ϕ) then they need $\lambda_1(D) <$ Gaussian expected SV.

Also mentioned in the paper by Chen and Nguyen [1], the algorithm called **BKZ** is used in high dimensional lattice attacks. And we have the relation between the block size and Hermite root factor. In the paper, it was mentioned that the block size of **BKZ** multiplied by 0.265 is a well-estimated upper bound on the bit security of the lattice.

Hermite Root Factor	1.01^n	1.009^n	1.008^n	1.007^n	1.006^n	1.005^n
Approximate Blocksize	85	106	133	168	216	286

Note that our lattice is of dimension N, so we can form a relationship between the Nth root of the Hermite Root Factor and the blocksize. It can be seen that for either $c = 1$, $c = 2$, or $c = 3$, the upper bound on the bit security increases as n increases. As we require a bit security of at least 128

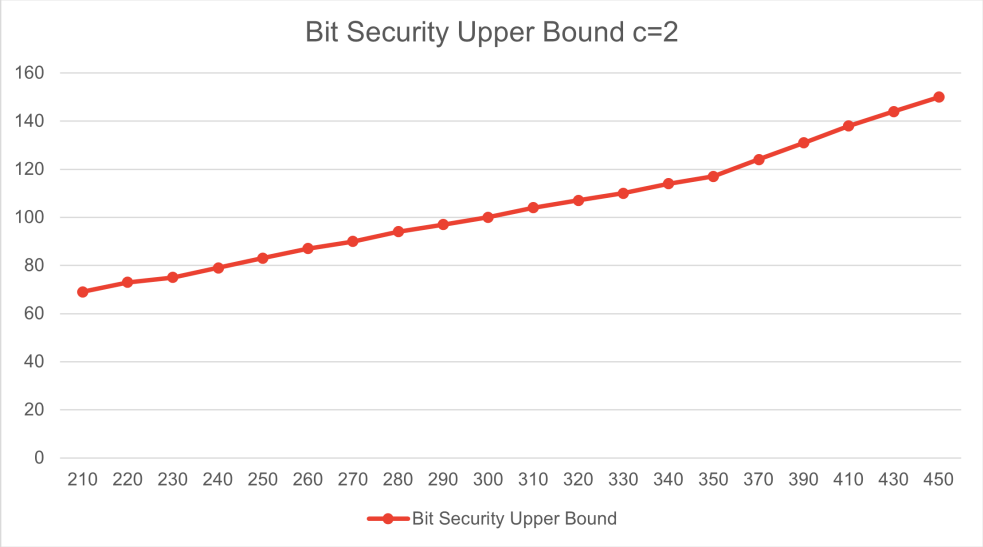

Figure 9:

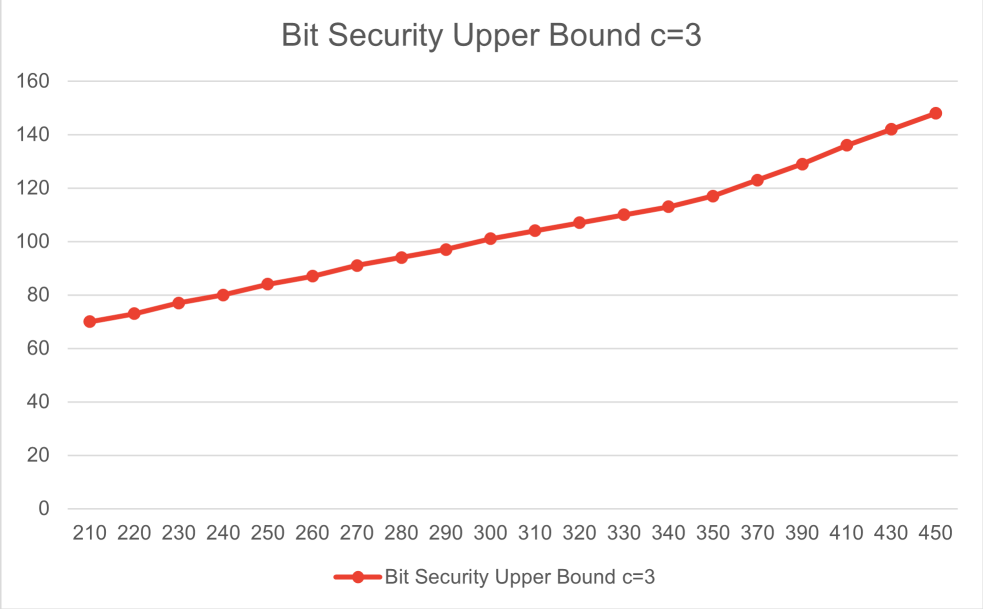

Figure 10:

(We used combinatorial bit security of 256 to avoid an attack based on some collision algorithm) and we are treating with some large n's, it's sufficient to use the case $c = 1$. The (n, q) pairs that would satisfy our criteria are

n	390	410	430	450
q	149	131	113	103

7 Conclusions

So we have devised the digital signature scheme, and found several optimal choices of n and q to serve as parameters. There are still many topics in this scheme that needs further study. For example, it remains unknown if there exists a rigorous proof that in field $\mathbf{F}[x]$, the square of a polynomial has short L^1 or L^∞ norm if and only if the polynomial is itself short. Or, there may be other functional lattice attacks on the FFI problem, and the upper bound on the bit-security may be improved. As many topics related to the FFI problem remain unsolved, this thesis serves as a possible way of utilizing it in cryptography.

References

[1] Yuanmi Chen and Phong Q. Nguyen. "BKZ 2.0: Better Lattice Security Estimates". In: *Advances in Cryptology – ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–20. ISBN: 978-3-642-25385-0.

[2] Yarkın Doröz et al. "Fully Homomorphic Encryption from the Finite Field Isomorphism Problem". In: *Public-Key Cryptography – PKC 2018*. Ed. by Michel Abdalla and Ricardo Dahab. Cham: Springer International Publishing, 2018, pp. 125–155. ISBN: 978-3-319-76578-5.