A history of the development of NTRU

Jeff Hoffstein
Brown University

EUROCRYPT 2014, Copenhagen

A one way function from number theory

- Let D be a large square free integer, and let $p_1, p_2, p_3, \ldots$ be a sequence of primes with $p_i \nmid D$. Define

$$\left(\frac{D}{p}\right) = \begin{cases} 1 & \text{if} \quad x^2 \equiv D \pmod{p} \text{ has a solution,} \\ -1 & \text{if} \quad x^2 \equiv D \pmod{p} \text{ doesn't have a solution.} \end{cases}$$

A one way function from number theory

- Let D be a large square free integer, and let $p_1, p_2, p_3, \ldots$ be a sequence of primes with $p_i \nmid D$. Define

$$\left(\frac{D}{p}\right) = \begin{cases} 1 & \text{if } \ x^2 \equiv D \pmod{p} \text{ has a solution,} \\ -1 & \text{if } \ x^2 \equiv D \pmod{p} \text{ doesn't have a solution.} \end{cases}$$

- Think of D as the key to a bitstream

$$D \to \left(\frac{D}{p_1}\right), \left(\frac{D}{p_2}\right), \ldots, \left(\frac{D}{p_t}\right).$$

A one way function from number theory, (cont.)

- One can also think of D as corresponding to a one way function, from sequences of primes to sequences of bits.

A one way function from number theory, (cont.)

- One can also think of D as corresponding to a one way function, from sequences of primes to sequences of bits.
- This is a very strong one way function in the following sense: Given a sequence such as

$$\{\epsilon_1, \epsilon_2, \ldots, \epsilon_{100{,}000}\},$$

with each $\epsilon_i = \pm 1$, there is with high probability at most one $D < 2^{80}$ with the property that $\left(\frac{D}{p}_i\right) = \epsilon_i$ for every i.

A one way function from number theory, (cont.)

- One can also think of D as corresponding to a one way function, from sequences of primes to sequences of bits.
- This is a very strong one way function in the following sense: Given a sequence such as

$$\{\epsilon_1, \epsilon_2, \ldots, \epsilon_{100{,}000}\},$$

  with each $\epsilon_i = \pm 1$, there is with high probability at most one $D < 2^{80}$ with the property that $\left(\frac{D}{p}_i\right) = \epsilon_i$ for every i.
- However, there is no known way to locate such a D without the knowledge of at least on the order of 2^{40} such $\left(\frac{D}{p}_i\right)$.

A one way function from number theory, (cont.)

- In fact, the $\left(\frac{D}{p}\right)_i$ can be thought of as the coefficients of something called a Dirichlet L-series:

$$L_D(s) = \prod_p \left(1 - \left(\frac{D}{p}\right)_i p^{-s}\right)^{-1}.$$

A one way function from number theory, (cont.)

- In fact, the $\left(\frac{D}{p}\right)_i$ can be thought of as the coefficients of something called a Dirichlet L-series:

$$L_D(s) = \prod_p \left(1 - \left(\frac{D}{p}\right)_i p^{-s}\right)^{-1}.$$

- This is an analog of the Riemann zeta function:

$$\zeta(s) = \sum_{n \geq 1} \frac{1}{n^s} = \prod_p \left(1 - p^{-s}\right)^{-1}.$$

Both are believed to satisfy the Riemann Hypothesis. In the case of $L_D(s)$ this says the the symbols $\left(\frac{D}{p}\right)_i$ are distributed randomly and uniformly.

A one way function from number theory, (cont.)

- In fact, the $\left(\frac{D}{p}\right)_i$ can be thought of as the coefficients of something called a Dirichlet L-series:

$$L_D(s) = \prod_p \left(1 - \left(\frac{D}{p}\right)_i p^{-s}\right)^{-1}.$$

- This is an analog of the Riemann zeta function:

$$\zeta(s) = \sum_{n \geq 1} \frac{1}{n^s} = \prod_p \left(1 - p^{-s}\right)^{-1}.$$

  Both are believed to satisfy the Riemann Hypothesis. In the case of $L_D(s)$ this says the the symbols $\left(\frac{D}{p}\right)_i$ are distributed randomly and uniformly.

- It was first suggested by Damgård (Crypto '88) that this mapping could be used as a one way function to construct a cryptographically strong bit generator.

Elliptic curves as one way functions

An elliptic curve

$$y^2 = x^3 + ax + b$$

has a sequence of coefficients associated to it. For every prime p we have

$$c_E(p) = p + 1 - \#E(F_p)$$

and $\#E(F_p)$ is one plus the number of solutions to
$y^2 \equiv x^3 + ax + b \pmod{p}$.
In 1994 Goldfeld and Anshel proposed that for each E, the mapping

$$E \to c_E(p_1), c_E(p_2), \ldots, c_E(p_t)$$

could be thought of as a one way function.

Elliptic curves as one way functions (cont.)

- Attached to each E there is also a corresponding $D = 4a^3 + 27b^2$.

Elliptic curves as one way functions (cont.)

- Attached to each E there is also a corresponding $D = 4a^3 + 27b^2$.
- There is also a corresponding L series, $L_E(s)$, and the Generalized Riemann Hypothesis implies that the $c_E(p)$ appear random and are well distributed.

Elliptic curves as one way functions (cont.)

- Attached to each E there is also a corresponding $D = 4a^3 + 27b^2$.
- There is also a corresponding L series, $L_E(s)$, and the Generalized Riemann Hypothesis implies that the $c_E(p)$ appear random and are well distributed.
- Goldfeld and I had shown that, assuming the GRH, $(\log D)^2 \log\log D$ coefficients determine the series,

Elliptic curves as one way functions (cont.)

- Attached to each E there is also a corresponding $D = 4a^3 + 27b^2$.
- There is also a corresponding L series, $L_E(s)$, and the Generalized Riemann Hypothesis implies that the $c_E(p)$ appear random and are well distributed.
- Goldfeld and I had shown that, assuming the GRH, $(\log D)^2 \log\log D$ coefficients determine the series,
- However, no known algorithm for reconstructing the curve with less than $\sqrt{D}$ coefficients exists.

Some irresistible questions

- Could this one way function (curve $\rightarrow$ coefficients) be used to construct a public key cryptosystem?

Some irresistible questions

- Could this one way function (curve $\rightarrow$ coefficients) be used to construct a public key cryptosystem?
- Or maybe a key exchange protocol?

Some irresistible questions

- Could this one way function (curve $\rightarrow$ coefficients) be used to construct a public key cryptosystem?
- Or maybe a key exchange protocol?
- A simpler question: Given a list of coefficients $c(p_1), c(p_2), \ldots, c(p_t)$ is there some way to prove that one has knowledge of the elliptic curve E that generates these coefficients, without revealing E?

Some irresistible questions

- Could this one way function (curve $\rightarrow$ coefficients) be used to construct a public key cryptosystem?
- Or maybe a key exchange protocol?
- A simpler question: Given a list of coefficients $c(p_1), c(p_2), \ldots, c(p_t)$ is there some way to prove that one has knowledge of the elliptic curve E that generates these coefficients, without revealing E?
- Twenty years later I still don't know the answers to these questions.

Enter function fields

- A very simple class of L-series: Fix a prime q and consider monic polynomials with coefficients chosen mod q. Can define a Legendre symbol:

$$\left(\frac{f}{g}\right) = \begin{cases} 1 & \text{if } \; x^2 \equiv f \pmod{g} \text{ has a solution,} \\ -1 & \text{if } \; x^2 \equiv f \pmod{g} \text{ doesn't have a solution,} \\ 0 & \text{if } (f, g) \neq 1. \end{cases}$$

Enter function fields

- A very simple class of L-series: Fix a prime q and consider monic polynomials with coefficients chosen mod q. Can define a Legendre symbol:

$$\left(\frac{f}{g}\right) = \begin{cases} 1 & \text{if } \; x^2 \equiv f \pmod{g} \text{ has a solution,} \\ -1 & \text{if } \; x^2 \equiv f \pmod{g} \text{ doesn't have a solution,} \\ 0 & \text{if } (f, g) \neq 1. \end{cases}$$

- For such symbols an analogous RH is known to be true, proved by A. Weil, and consequently the values of $\left(\frac{f}{g}\right)$ as g varies over irreducible monic polynomials (primes) are random and well distributed.

function fields, cont.

- The very simplest primes are monic of degree one: $g = x - \alpha$, for some $\alpha \pmod{q}$.

- The very simplest primes are monic of degree one: $g = x - \alpha$, for some $\alpha \pmod{q}$.
- In this case
$$\left(\frac{f}{x-\alpha}\right) = \left(\frac{f(\alpha)}{q}\right).$$

function fields, cont.

- The very simplest primes are monic of degree one: $g = x - \alpha$, for some $\alpha \pmod{q}$.
- In this case
$$\left(\frac{f}{x-\alpha}\right) = \left(\frac{f(\alpha)}{q}\right).$$
- So in particular, the values of the usual quadratic symbol $\left(\frac{f(\alpha)}{q}\right)$ are random and uniformly distributed mod q.

- The very simplest primes are monic of degree one: $g = x - \alpha$, for some $\alpha \pmod q$.
- In this case
$$\left(\frac{f}{x-\alpha}\right) = \left(\frac{f(\alpha)}{q}\right).$$
- So in particular, the values of the usual quadratic symbol $\left(\frac{f(\alpha)}{q}\right)$ are random and uniformly distributed mod q.

New Question

Given a public q and a secret polynomial f, prove knowledge of f, given a public collection of values:

$$\left(\frac{f(\alpha_1)}{q}\right), \left(\frac{f(\alpha_2)}{q}\right), \ldots, \left(\frac{f(\alpha_t)}{q}\right).$$

In fact, why not consider the actual values $f(\alpha) \pmod{q}$?

New Question 2

Given a public q and a secret polynomial f, prove knowledge of f, given a public collection of values:

$$f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_t) \pmod{q}.$$

The problem: If $t \approx \deg(f')/2$, there are lots of f' such that $f'(\alpha_i) \equiv f(\alpha_i) \pmod{q}$ for $1 \leq i \leq t$.
Possible solution: Require that f also belong to some restricted class determined by its coefficients.

Lattices enter the picture

Definition

A polynomial $f(x) = a_0 + a_1 x + \cdots + a_{N-1}x^{N-1}$ with coefficients in $\mathbb{Z}$ is called *short* if there exists $1 \leq c \ll q$ such that for each i, $|a_i| \leq c$. A polynomial $f \in \mathbb{Z}/q\mathbb{Z}[x]$ is called short if there is a lift back to $\mathbb{Z}[x]$ that is short.

We now have a very specific hard problem to work with:

Hard Problem

Given $N > t > 1$, and two collections of values mod q:

$$\{\alpha_1, \alpha_2, \ldots, \alpha_t\} \text{ and } \{\beta_1, \beta_2, \ldots, \beta_t\},$$

find a polynomial f with $\deg f < N$ such that f is short, and

$$f(\alpha_i) \equiv \beta_i \pmod{p} \text{ for } i = 1, 2, \ldots, t.$$

Translation into a closest vector problem, or CVP

For any polynomial p with $\deg p \le N-1$, identify $p(x) = a_0 + a_1 x + \cdots + a_{N-1}x^{N-1}$ with the vector

$$(a_0, a_1, \ldots, a_{N-1}) \in \mathbb{Z}^N.$$

Let L denote the lattice of all vectors p such that

$$p(\alpha_i) \equiv 0 \pmod{q}, \text{ for all } 1 \le i \le t.$$

Let F correspond to any, not necessarily short, polynomial satisfying

$$F(\alpha_i) \equiv b_i \pmod{q}, \text{ for all } 1 \le i \le t.$$

Then if F_0 is the lattice point of L that is closest to F, with high probability $F - F_0$ will be a short polynomial with the correct evaluations.

How hard is it to solve a CVP?

- This had to be left unsolved for now.

How hard is it to solve a CVP?

- This had to be left unsolved for now.
- The LLL algorithm performed better than expected in dimensions less than 100. What would happen if the degree was 200 or 300?

How hard is it to solve a CVP?

- This had to be left unsolved for now.
- The LLL algorithm performed better than expected in dimensions less than 100. What would happen if the degree was 200 or 300?
- It was generally believed in 1995 that it would continue to perform well enough to be a significant danger to lattice based cryptosystems.

How hard is it to solve a CVP?

- This had to be left unsolved for now.
- The LLL algorithm performed better than expected in dimensions less than 100. What would happen if the degree was 200 or 300?
- It was generally believed in 1995 that it would continue to perform well enough to be a significant danger to lattice based cryptosystems.
- I asked H. Lenstra how effective LLL was as the dimension increased?

- This had to be left unsolved for now.
- The LLL algorithm performed better than expected in dimensions less than 100. What would happen if the degree was 200 or 300?
- It was generally believed in 1995 that it would continue to perform well enough to be a significant danger to lattice based cryptosystems.
- I asked H. Lenstra how effective LLL was as the dimension increased?

The Question Remained

Assuming it is hard to find a short polynomial with specific evaluations, how to prove knowledge of one?

Introducing a more compact ring structure

- Rather than taking $f(x) \in \mathbb{Z}/q\mathbb{Z}[x]$, take $f(x) \in \mathbb{Z}/q\mathbb{Z}[x]/(x^N - 1)$.

Introducing a more compact ring structure

- Rather than taking $f(x) \in \mathbb{Z}/q\mathbb{Z}[x]$, take $f(x) \in \mathbb{Z}/q\mathbb{Z}[x]/(x^N - 1)$.
- If for each i, $\alpha_i^N \equiv 1 \pmod{q}$, then the map

$$f \to (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_t)) \pmod{q, x^N - 1}$$

is a ring homomorphism.

Introducing a more compact ring structure

- Rather than taking $f(x) \in \mathbb{Z}/q\mathbb{Z}[x]$, take $f(x) \in \mathbb{Z}/q\mathbb{Z}[x]/(x^N - 1)$.
- If for each i, $\alpha_i^N \equiv 1 \pmod{q}$, then the map

$$f \to (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_t)) \pmod{q, x^N - 1}$$

is a ring homomorphism.
- On the left, multiplication is given by a convolution operation:

$$\left(\sum_{i=0}^{N-1} a_i x^i\right) * \left(\sum_{j=0}^{N-1} b_j x^j\right) = \sum_{k=0}^{N-1} c_k x^k,$$

where

$$c_k = \sum_{i+j\equiv k} a_i b_j.$$

Introducing a more compact ring structure, cont.

- This ring homomorphism is actually the mapping of a function to its Fourier transform.

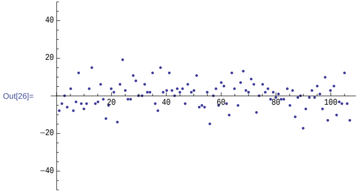
Introducing a more compact ring structure, cont.

- This ring homomorphism is actually the mapping of a function to its Fourier transform.
- A short polynomial is one with its Fourier coefficients concentrated within a bounded distance from 0.

Introducing a more compact ring structure, cont.

- This ring homomorphism is actually the mapping of a function to its Fourier transform.
- A short polynomial is one with its Fourier coefficients concentrated within a bounded distance from 0.
- The uncertainty principle tells us that the tighter the distribution of the Fourier coefficients, the more dispersed the Fourier transform will be.

Suppose, $N = 107$, and f, g have coefficients from $\{-1, 0, 1\}$.
Then the distribution of coefficients of $f * g$ looks like



Proof of knowledge of a short polynomial

The hard problem of finding a short polynomial with a specified collection of values was turned into a digital signature scheme (as opposed to a public key cryptosystem) during the year 1994-95, with Burt Kaliski, Daniel Lieman, Matt Robshaw and Yiqun Lisa Yin.

Proof of knowledge of a short polynomial, (cont.)

- One is given a short f with valuations

$$(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_t)) \pmod{q}.$$

Proof of knowledge of a short polynomial, (cont.)

- One is given a short f with valuations

$$(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_t)) \pmod{q}.$$

- Generate a random short g and publish

$$(g(\alpha_1), g(\alpha_2), \ldots, g(\alpha_t)) \pmod{q}.$$

Proof of knowledge of a short polynomial, (cont.)

- One is given a short f with valuations

$$(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_t)) \pmod{q}.$$

- Generate a random short g and publish

$$(g(\alpha_1), g(\alpha_2), \ldots, g(\alpha_t)) \pmod{q}.$$

- Receive challenge, a short polynomial c.

Proof of knowledge of a short polynomial, (cont.)

- One is given a short f with valuations

$$(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_t)) \pmod{q}.$$

- Generate a random short g and publish

$$(g(\alpha_1), g(\alpha_2), \ldots, g(\alpha_t)) \pmod{q}.$$

- Receive challenge, a short polynomial c.
- Compute and publish the polynomial

$$h = g * (f + c).$$

Proof of knowledge of a short polynomial, (cont.)

- One is given a short f with valuations

$$(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_t)) \pmod{q}.$$

- Generate a random short g and publish

$$(g(\alpha_1), g(\alpha_2), \ldots, g(\alpha_t)) \pmod{q}.$$

- Receive challenge, a short polynomial c.
- Compute and publish the polynomial

$$h = g * (f + c).$$

- Verify that h is short, and $h(\alpha_i) \equiv g(\alpha_i)(f(\alpha_i) + c(\alpha_i)) \pmod{q}$ for all i.

The catch....

- It appeared at first that it would be hard to recover the secret f from a long list of h, that is, a long transcript.

The catch....

- It appeared at first that it would be hard to recover the secret f from a long list of h, that is, a long transcript.
- Burt Kaliski noticed that if you introduce the notion of a reversal operation

$$\tilde{h}(x) = h(x^{-1}),$$

then this was a ring homomorphism and an average of $h * \tilde{h}$ would converge to a constant multiple of $f * \tilde{f}$.

The catch....

- It appeared at first that it would be hard to recover the secret f from a long list of h, that is, a long transcript.
- Burt Kaliski noticed that if you introduce the notion of a reversal operation
$$\tilde{h}(x) = h(x^{-1}),$$
then this was a ring homomorphism and an average of $h * \tilde{h}$ would converge to a constant multiple of $f * \tilde{f}$.
- We never found a clean way of reducing or eliminating this information leakage.

A fix, many years later....

- In 2009 V. Lyubashevsky introduced the notion of rejection sampling.

A fix, many years later....

- In 2009 V. Lyubashevsky introduced the notion of rejection sampling.
- It turns out that replacing $g * (f + c)$ by $g + f * c$, and eliminating g, c pairs when $g + f * c$ has too large an infinity norm, can produce an information free transcript.

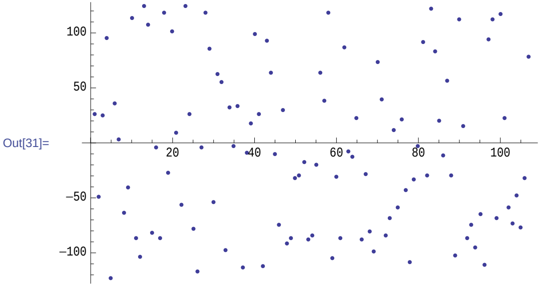
Fall 1995 - enter NTRU

- For most f there exists an inverse f^{-1} with the property that $f * f^{-1} \equiv 1 \pmod{q, x^N - 1}$

Fall 1995 - enter NTRU

- For most f there exists an inverse f^{-1} with the property that $f * f^{-1} \equiv 1 \pmod{q, x^N - 1}$
- If the coefficients of f are chosen from $\{-1, 0, 1\}$, the coefficients of f^{-1} look completely random mod q.

- For most f there exists an inverse f^{-1} with the property that $f * f^{-1} \equiv 1 \pmod{q, x^N - 1}$
- If the coefficients of f are chosen from $\{-1, 0, 1\}$, the coefficients of f^{-1} look completely random mod q.



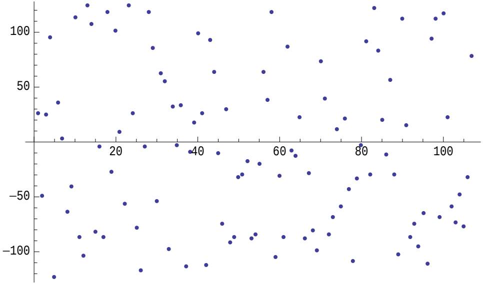
-

Fall 1995 - enter NTRU, cont.

- In fact, if g is another such short polynomial, the coefficients of $h = 3f^{-1} * g$ also look random

- In fact, if g is another such short polynomial, the coefficients of $h = 3f^{-1} * g$ also look random

Out[35]=



-

Fall 1995 - enter NTRU, cont.

- This suggested that if m, r were two additional short polynomials, m could be concealed by writing

$$e = r * h + m.$$

- This suggested that if m, r were two additional short polynomials, m could be concealed by writing

$$e = r * h + m.$$

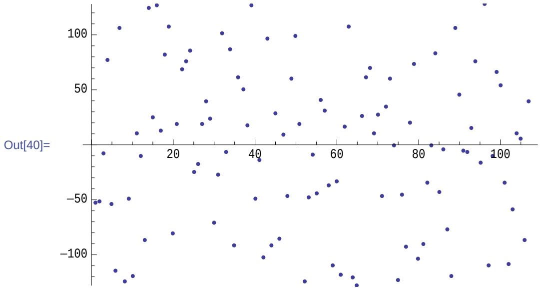
- This looks random:

Fall 1995 - enter NTRU, cont.

- This suggested that if m, r were two additional short polynomials, m could be concealed by writing

$$e = r * h + m.$$

- This looks random:



-

Fall 1995 - enter NTRU, cont.

- However, multiplying by f would create

$$a = f * e = f * (3r * f^{-1} * g + m) = 3r * g + f * m,$$

- However, multiplying by f would create

$$a = f * e = f * (3r * f^{-1} * g + m) = 3r * g + f * m,$$

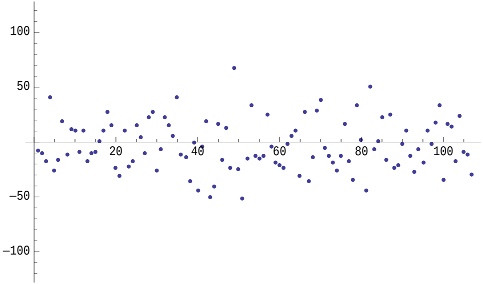
- which is *short*, and has a coefficient distribution

- However, multiplying by f would create

$$a = f * e = f * (3r * f^{-1} * g + m) = 3r * g + f * m,$$

- which is *short*, and has a coefficient distribution

Out[43]=



-

Fall 1995 - enter NTRU, cont.

- The original $3r * g + f * m$, not reduced mod q would then be recovered.

Fall 1995 - enter NTRU, cont.

- The original $3r * g + f * m$, not reduced mod q would then be recovered.
- Reducing mod 3 $\rightarrow f * m \pmod{3}$,

Fall 1995 - enter NTRU, cont.

- The original $3r * g + f * m$, not reduced mod q would then be recovered.
- Reducing mod 3 $\rightarrow f * m \pmod 3$,
- and multiplying by $f^{-1} \pmod 3$ would reveal m

Fall 1995 - Spring 1996

- Teamed up with Jill Pipher and Joe Silverman to analyze security, determine parameters, and to generally try to transform a vague idea into a concrete cryptosystem.

Fall 1995 - Spring 1996

- Teamed up with Jill Pipher and Joe Silverman to analyze security, determine parameters, and to generally try to transform a vague idea into a concrete cryptosystem.
- The fundamental hard problem was: Given a polynomial h of degree $N - 1$ with coefficients mod q, find a short polynomial f with the property that after reduction mod q, $f * h$ was also short.

Fall 1995 - Spring 1996

- Teamed up with Jill Pipher and Joe Silverman to analyze security, determine parameters, and to generally try to transform a vague idea into a concrete cryptosystem.
- The fundamental hard problem was: Given a polynomial h of degree $N-1$ with coefficients mod q, find a short polynomial f with the property that after reduction mod q, $f * h$ was also short.
- This was immediately translatable into the problem of finding a very short vector (f, g) in a certain $2N$-dimensional lattice.

Fall 1995 - Spring 1996

- Teamed up with Jill Pipher and Joe Silverman to analyze security, determine parameters, and to generally try to transform a vague idea into a concrete cryptosystem.
- The fundamental hard problem was: Given a polynomial h of degree $N-1$ with coefficients mod q, find a short polynomial f with the property that after reduction mod q, $f * h$ was also short.
- This was immediately translatable into the problem of finding a very short vector (f, g) in a certain $2N$-dimensional lattice.
- We believed this problem should be hard, but we had no idea how to quantify the hardness.

Fall 1995 - Spring 1996

- Teamed up with Jill Pipher and Joe Silverman to analyze security, determine parameters, and to generally try to transform a vague idea into a concrete cryptosystem.
- The fundamental hard problem was: Given a polynomial h of degree $N-1$ with coefficients mod q, find a short polynomial f with the property that after reduction mod q, $f * h$ was also short.
- This was immediately translatable into the problem of finding a very short vector (f, g) in a certain $2N$-dimensional lattice.
- We believed this problem should be hard, but we had no idea how to quantify the hardness.
- We calculated the combinatorial difficulty of searching for f via brute force, and A. Odlyzko showed us how a meet in the middle attack could cut the combinatorial security exponent in half.

The rump session of Crypto '96

- I presented the ideas in a several minute slot in the rump session of Crypto '96.

- I presented the ideas in a several minute slot in the rump session of Crypto ’96.
- I presented the ideas as I would have at any math conference: i.e., I hoped that they would be thought interesting and that people who knew more about this stuff than I did would be able to make helpful suggestions.

The rump session of Crypto ’96

- I presented the ideas in a several minute slot in the rump session of Crypto ’96.
- I presented the ideas as I would have at any math conference: i.e., I hoped that they would be thought interesting and that people who knew more about this stuff than I did would be able to make helpful suggestions.
- People were, in fact, interested, but they also seemed irritated that I had not done a complete security analysis before presenting it, and had not circulated it to experts in cryptography first.

The response from D. Coppersmith and A. Shamir at EuroCrypt '97

- Their position was that LLL would easily solve the shortest vector problem for any remotely practical parameters.

The response from D. Coppersmith and A. Shamir at EuroCrypt '97

- Their position was that LLL would easily solve the shortest vector problem for any remotely practical parameters.
- They made one important observation that we had missed:

The response from D. Coppersmith and A. Shamir at EuroCrypt '97

- Their position was that LLL would easily solve the shortest vector problem for any remotely practical parameters.
- They made one important observation that we had missed:
- If there was another vector in the lattice (f', g') of a similar length to (f, g), or shorter, then f' would probably act as a moderately good decryption key. Here's what they said:

The response from D. Coppersmith and A. Shamir at EuroCrypt '97

- To summarize: if there are many vectors f' with $n_{f'} \leq n_f$ then we are likely to stumble across one and be able to decrypt. If f is much shorter than all other vectors then we are likely to find f. The only hope for the scheme to remain secure is for many vectors to satisfy, say, $n_{f'} = 10 \times n_f$ and hope that the lattice basis reduction methods fail to find f among the sea of f'. With any improvements in the technology of lattice basis reductions, this temporary security would vanish.

The response from D. Coppersmith and A. Shamir at EuroCrypt '97

- To summarize: if there are many vectors f' with $n_{f'} \leq n_f$ then we are likely to stumble across one and be able to decrypt. If f is much shorter than all other vectors then we are likely to find f. The only hope for the scheme to remain secure is for many vectors to satisfy, say, $n_{f'} = 10 \times n_f$ and hope that the lattice basis reduction methods fail to find f among the sea of f'. With any improvements in the technology of lattice basis reductions, this temporary security would vanish.
- They also said:

The response from D. Coppersmith and A. Shamir at EuroCrypt '97

- To summarize: if there are many vectors f' with $n_{f'} \leq n_f$ then we are likely to stumble across one and be able to decrypt. If f is much shorter than all other vectors then we are likely to find f. The only hope for the scheme to remain secure is for many vectors to satisfy, say, $n_{f'} = 10 \times n_f$ and hope that the lattice basis reduction methods fail to find f among the sea of f'. With any improvements in the technology of lattice basis reductions, this temporary security would vanish.
- They also said:
- ... We believe that for the recommended parameters of the NTRU cryptosystem the LLL algorithm will be able to find the original secret key f ...

- I was told by someone who was there (a leading figure in the field) that by the end of the talk NTRU lay in shreds and tatters on the floor.

- I was told by someone who was there (a leading figure in the field) that by the end of the talk NTRU lay in shreds and tatters on the floor.
- The original NTRU paper was rejected by the Crypto '97 organizing committee.

What next

- We were sure that the surprising successes of LLL were an artifact of low dimensions.

What next

- We were sure that the surprising successes of LLL were an artifact of low dimensions.
- We set up a (cheap) computer lab and began running months of testing of the BKZ algorithm, using Victor Shoup's NTL implementation.

What next

- We were sure that the surprising successes of LLL were an artifact of low dimensions.
- We set up a (cheap) computer lab and began running months of testing of the BKZ algorithm, using Victor Shoup's NTL implementation.
- We solved the key recovery problem for sequences $(N_1, q_1), (N_2, q_2), \ldots$ with the N_i increasing, and the ratio N_i/q_i constant.

What next

- We were sure that the surprising successes of LLL were an artifact of low dimensions.
- We set up a (cheap) computer lab and began running months of testing of the BKZ algorithm, using Victor Shoup's NTL implementation.
- We solved the key recovery problem for sequences $(N_1, q_1), (N_2, q_2), \ldots$ with the N_i increasing, and the ratio N_i/q_i constant.
- We found that block size 2 ($=$ LLL) worked for initial N up to about 50, (Corresponding lattice dimension $= 100$).

What next

- We were sure that the surprising successes of LLL were an artifact of low dimensions.
- We set up a (cheap) computer lab and began running months of testing of the BKZ algorithm, using Victor Shoup's NTL implementation.
- We solved the key recovery problem for sequences $(N_1, q_1), (N_2, q_2), \ldots$ with the N_i increasing, and the ratio N_i/q_i constant.
- We found that block size 2 ($=$ LLL) worked for initial N up to about 50, (Corresponding lattice dimension $= 100$).
- Afterwords, the necessary block size increased linearly with N, with a slope depending on the N/q ratio.

What next

- We were sure that the surprising successes of LLL were an artifact of low dimensions.
- We set up a (cheap) computer lab and began running months of testing of the BKZ algorithm, using Victor Shoup's NTL implementation.
- We solved the key recovery problem for sequences $(N_1, q_1), (N_2, q_2), \ldots$ with the N_i increasing, and the ratio N_i/q_i constant.
- We found that block size 2 ($=$ LLL) worked for initial N up to about 50, (Corresponding lattice dimension $=$ 100).
- Afterwords, the necessary block size increased linearly with N, with a slope depending on the N/q ratio.
- Computation time went up slightly super exponentially with block size, and also with N.

What we found

- There is a radius that comes from the Gaussian heuristic:

$$r = \sqrt{\frac{Nq}{\pi e}}.$$

What we found

- There is a radius that comes from the Gaussian heuristic:

$$r = \sqrt{\frac{Nq}{\pi e}}.$$

- We set things so that

$$\text{target} = ||(f, g)|| \approx r/\sqrt{N}.$$

What we found

- There is a radius that comes from the Gaussian heuristic:

$$r = \sqrt{\frac{Nq}{\pi e}}.$$

- We set things so that

$$\text{target} = ||(f, g)|| \approx r/\sqrt{N}.$$

- With high probability there were no lattice elements other than rotations of (f, g) inside a sphere of radius r.

What we found

- There is a radius that comes from the Gaussian heuristic:

$$r = \sqrt{\frac{Nq}{\pi e}}.$$

- We set things so that

$$\text{target} = ||(f, g)|| \approx r/\sqrt{N}.$$

- With high probability there were no lattice elements other than rotations of (f, g) inside a sphere of radius r.
- We found that finding a lattice element with norm close to r was a little like trying to approach the speed of light.

What we found

- There is a radius that comes from the Gaussian heuristic:

$$r = \sqrt{\frac{Nq}{\pi e}}.$$

- We set things so that

$$\text{target} = ||(f, g)|| \approx r/\sqrt{N}.$$

- With high probability there were no lattice elements other than rotations of (f, g) inside a sphere of radius r.
- We found that finding a lattice element with norm close to r was a little like trying to approach the speed of light.
- BKZ would find only trivial solutions until the block size was big enough, then break through directly to the key.

Challenge problems.

In 1997/98 we published four challenge problems:

- N=107, q=64 (a warmup),
- N = 167, q=128,
- N = 251, q = 256,
- N= 503, q = 256.

The $N = 107$ problem was solved by A. May and P. Nguyen. (And possibly others that didn't communicate with us.) To this day I am not aware of any solutions to even the $N = 167$ problem.

Time for a signature scheme and a disaster.

- While these experiments were progressing there were a number of papers published on NTRU. Some proposed methods of speeding up lattice reduction, such as zero forcing (A. May). Others focused on attacks due to potential decryption failures.

Time for a signature scheme and a disaster.

- While these experiments were progressing there were a number of papers published on NTRU. Some proposed methods of speeding up lattice reduction, such as zero forcing (A. May). Others focused on attacks due to potential decryption failures.
- In the meantime, we figured we would try to find a signature scheme based on this circle of ideas.

Time for a signature scheme and a disaster.

- While these experiments were progressing there were a number of papers published on NTRU. Some proposed methods of speeding up lattice reduction, such as zero forcing (A. May). Others focused on attacks due to potential decryption failures.
- In the meantime, we figured we would try to find a signature scheme based on this circle of ideas.
- The hope was to find something based on the following hard problem: Given the product $f * g$, and the knowledge that f, g are short, recover f, g.

Time for a signature scheme and a disaster.

- Skipping over some other mistakes we made, what we came up with unfortunately produced a transcript reducible to: $f * g_1, f * g_2, \ldots f * g_t$, and this turned out to be a lot easier than the original problem.

Time for a signature scheme and a disaster.

- Skipping over some other mistakes we made, what we came up with unfortunately produced a transcript reducible to: $f * g_1, f * g_2, \ldots f * g_t$, and this turned out to be a lot easier than the original problem.
- It could be treated as a lattice problem, finding a gcd, but C. Gentry and M. Szydlo found a much more powerful way to attack it.

Time for a signature scheme and a disaster.

- Skipping over some other mistakes we made, what we came up with unfortunately produced a transcript reducible to: $f * g_1, f * g_2, \ldots f * g_t$, and this turned out to be a lot easier than the original problem.
- It could be treated as a lattice problem, finding a gcd, but C. Gentry and M. Szydlo found a much more powerful way to attack it.
- $$(f * g_i) * \mathrm{rev}(f * g_i) \to f * \tilde{f} * g_i * \tilde{g}_i \to (\mathrm{constant})(f * \tilde{f}).$$

Time for a signature scheme and a disaster.

- Skipping over some other mistakes we made, what we came up with unfortunately produced a transcript reducible to: $f * g_1, f * g_2, \ldots f * g_t$, and this turned out to be a lot easier than the original problem.
- It could be treated as a lattice problem, finding a gcd, but C. Gentry and M. Szydlo found a much more powerful way to attack it.
- $$(f * g_i) * \mathrm{rev}(f * g_i) \to f * \tilde{f} * g_i * \tilde{g}_i \to (\mathrm{constant})(f * \tilde{f}).$$
- They found a very clever way to recover f from $f * \tilde{f}$.

Time for a signature scheme and a disaster.

- Luckily the crypto community was pretty forgiving about this mishap.

Time for a signature scheme and a disaster.

- Luckily the crypto community was pretty forgiving about this mishap.

Signature scheme, take 2.

- The alternative was to base a signature scheme directly on the NTRU lattice, following the model of GGH,

Signature scheme, take 2.

- The alternative was to base a signature scheme directly on the NTRU lattice, following the model of GGH,
- Nick Howgrave-Graham helped us find a way to construct a complete basis for the NTRU lattice, out of the half basis consisting of rotations of (f, g).

Signature scheme, take 2.

- The alternative was to base a signature scheme directly on the NTRU lattice, following the model of GGH,
- Nick Howgrave-Graham helped us find a way to construct a complete basis for the NTRU lattice, out of the half basis consisting of rotations of (f, g).
- The scheme was then simply the traditional one of using the better private basis to find non-trivial solutions to CVP.

- It was still vulnerable to the derivation of a 2 by 2 Gram matrix from a long transcript. This matrix had four entries similar to, but somewhat more complicated than, the $f\tilde{f}$ object that caused the vulnerability of NSS.

New vulnerabilities.

- It was still vulnerable to the derivation of a 2 by 2 Gram matrix from a long transcript. This matrix had four entries similar to, but somewhat more complicated than, the $f\tilde{f}$ object that caused the vulnerability of NSS.
- I still don't know if this higher dimensional object is attackable by the same strategy.

New vulnerabilities.

- It was still vulnerable to the derivation of a 2 by 2 Gram matrix from a long transcript. This matrix had four entries similar to, but somewhat more complicated than, the $f\tilde{f}$ object that caused the vulnerability of NSS.
- I still don't know if this higher dimensional object is attackable by the same strategy.
- Just this summer I asked H. Lenstra....

- A long transcript of NTRUSign signatures revealed a fuzzy image of a private $2n$-dimensional fundamental parallelepiped.

New vulnerabilities, cont.

- A long transcript of NTRUSign signatures revealed a fuzzy image of a private $2n$-dimensional fundamental parallelepiped.
- In 2006 P. Nguyen and O. Regev found a very clever way of using fourth moments and independent component analysis to recover the secret key from such an image.

New vulnerabilities, cont.

- A long transcript of NTRUSign signatures revealed a fuzzy image of a private $2n$-dimensional fundamental parallelepiped.
- In 2006 P. Nguyen and O. Regev found a very clever way of using fourth moments and independent component analysis to recover the secret key from such an image.
- One defense against this was the addition of perturbations to the signatures. Essentially this replaced the $2n$-dimensional fundamental parallelepiped by the sum of several such parallelepipeds.

New vulnerabilities, cont.

- A long transcript of NTRUSign signatures revealed a fuzzy image of a private $2n$-dimensional fundamental parallelepiped.
- In 2006 P. Nguyen and O. Regev found a very clever way of using fourth moments and independent component analysis to recover the secret key from such an image.
- One defense against this was the addition of perturbations to the signatures. Essentially this replaced the $2n$-dimensional fundamental parallelepiped by the sum of several such parallelepipeds.
- Then, around a year and a half ago, P. Nguyen and L. Ducas managed to solve the case of one perturbation, with the possibility of going further.

New vulnerabilities, cont.

- A long transcript of NTRUSign signatures revealed a fuzzy image of a private $2n$-dimensional fundamental parallelepiped.
- In 2006 P. Nguyen and O. Regev found a very clever way of using fourth moments and independent component analysis to recover the secret key from such an image.
- One defense against this was the addition of perturbations to the signatures. Essentially this replaced the $2n$-dimensional fundamental parallelepiped by the sum of several such parallelepipeds.
- Then, around a year and a half ago, P. Nguyen and L. Ducas managed to solve the case of one perturbation, with the possibility of going further.
- So clearly this sort of perturbation was not the answer.

A new signature scheme!

- Finally, inspired by rejection sampling, there is a completely new version of NTRUSign.

A new signature scheme!

- Finally, inspired by rejection sampling, there is a completely new version of NTRUSign.
- It uses only the half basis of rotations of (f, g), and an auxiliary small prime p.

A new signature scheme!

- Finally, inspired by rejection sampling, there is a completely new version of NTRUSign.
- It uses only the half basis of rotations of (f, g), and an auxiliary small prime p.
- It has a provably information-free transcript.

Gaussian sampling and a new era

In 2008 C Gentry, C Peikert, V Vaikuntanathan introduced the notion of generating lattice points according to a Gaussian distribution. This was extended by a number of authors, including C. Peikert, L. Ducas and P. Nguyen. In 2011 D. Stehlé and R. Steinfeld showed how to use such techniques to relate the security of NTRU and NTRUSign to worst case problems over ideal lattices. They showed that if the secret key polynomials are selected by rejection from discrete Gaussians, then the public key, which is their ratio, is statistically indistinguishable from uniform over its domain.

Homomorphic encryption

- In 2009 C. Gentry created the first fully homomorphic encryption scheme.

Homomorphic encryption

- In 2009 C. Gentry created the first fully homomorphic encryption scheme.
- As it was ring based, NTRU was (inadvertently) somewhat homomorphic - if q was large enough.

Homomorphic encryption

- In 2009 C. Gentry created the first fully homomorphic encryption scheme.
- As it was ring based, NTRU was (inadvertently) somewhat homomorphic - if q was large enough.
- This made it a logical direction to look in for exploring more efficient fully homomorphic encryption schemes.

Homomorphic encryption

- In 2009 C. Gentry created the first fully homomorphic encryption scheme.
- As it was ring based, NTRU was (inadvertently) somewhat homomorphic - if q was large enough.
- This made it a logical direction to look in for exploring more efficient fully homomorphic encryption schemes.
- Since then there has been a massive amount of work on this subject, and on the subject of further improving the theoretical foundations of schemes based on ideal lattices. Far too many authors to fit on this slide!

Homomorphic encryption

- In 2009 C. Gentry created the first fully homomorphic encryption scheme.
- As it was ring based, NTRU was (inadvertently) somewhat homomorphic - if q was large enough.
- This made it a logical direction to look in for exploring more efficient fully homomorphic encryption schemes.
- Since then there has been a massive amount of work on this subject, and on the subject of further improving the theoretical foundations of schemes based on ideal lattices. Far too many authors to fit on this slide!
- I could talk about this for hours more....

Homomorphic encryption

- In 2009 C. Gentry created the first fully homomorphic encryption scheme.
- As it was ring based, NTRU was (inadvertently) somewhat homomorphic - if q was large enough.
- This made it a logical direction to look in for exploring more efficient fully homomorphic encryption schemes.
- Since then there has been a massive amount of work on this subject, and on the subject of further improving the theoretical foundations of schemes based on ideal lattices. Far too many authors to fit on this slide!
- I could talk about this for hours more....
- ... or I could stop right here.

Homomorphic encryption

- In 2009 C. Gentry created the first fully homomorphic encryption scheme.
- As it was ring based, NTRU was (inadvertently) somewhat homomorphic - if q was large enough.
- This made it a logical direction to look in for exploring more efficient fully homomorphic encryption schemes.
- Since then there has been a massive amount of work on this subject, and on the subject of further improving the theoretical foundations of schemes based on ideal lattices. Far too many authors to fit on this slide!
- I could talk about this for hours more....
- ... or I could stop right here.
- Thanks!

Homomorphic encryption

- In 2009 C. Gentry created the first fully homomorphic encryption scheme.
- As it was ring based, NTRU was (inadvertently) somewhat homomorphic - if q was large enough.
- This made it a logical direction to look in for exploring more efficient fully homomorphic encryption schemes.
- Since then there has been a massive amount of work on this subject, and on the subject of further improving the theoretical foundations of schemes based on ideal lattices. Far too many authors to fit on this slide!
- I could talk about this for hours more....
- ... or I could stop right here.
- Thanks!
- For the memories....