Gröbner bases, lattices and cryptography

Carlo Traverso

Sage days 10 in Nancy
October 12 2008

Gröbner bases

$A = k[X]$ a polynomial ring in n variables. I an ideal. M the monoid of the power products of A.

Choose a total ordering on M, monoid compatible, and noetherian (i.e. 1 is the minimum). Examples: Lex, DevRevLex, block orderings.

Define the map $Lpp : A \setminus \{0\} \to M$ associating to each polynomial its *leading power-product*. Its image is the *initial ideal*, and is a monoid ideal. Its complementary is the *staircase*. A *(reduced) Gröbner basis* is a set of polynomials whose Lpp minimally generate the initial ideal, (and whose *tails* are in the staircase). The reduced Gröbner basis is unique, once the ordering is fixed.

The staircase is a basis in the quotient A/I, through division by the Gröbner basis one can algorithmically represent every element of the quotient as linear combination of the basis elements.

Integer lattices

An integer lattice is a subgroup $L \subseteq \mathbf{Z}^n$, with the metric induced by the euclidean metric.

Every lattice is isomorphic (as group) to a $\mathbf{Z}^r$, r being its rank; it has hence a (non-unique) basis, that can be seen as an integer matrix (traditionally, the basis elements are the matrix rows); full-rank lattices have a determinant, that is the cardinality of $\mathbf{Z}^n/L$.

The Shortest Vector Problem (SVP) and Closest Vector Problem (CVP) are NP-hard.

The LLL reduction algorithm, and its BKZ variant allow to find in polynomial time approximate SVP and CVP, up to an exponential factor. In practice, the algorithms perform much better than the theoretical result, both in the running time and the approximation of the result.

Increasing the dimension however the problems become unfeasible for sufficiently high dimension.

Public key cryptography

In public key cryptography a one has a message space M and a cryptogram space C, two key spaces K and H, a randomizing space R, and algorithms for encryption $\epsilon : M \times R \times K \to C$, and decryption $\delta : C \times K' \to R$; one has also an algorithm to construct key pairs $(k, k' \in K \times K'$ such that $\forall r\ \delta(\epsilon(m, k, r), k') = m$ (probabilistically in some cases).

Finding a matching key k' from k (key attack), or a message m from the cryptogram c and the public key k (message attack) should be unfeasible.

Integer lattices and Public key cryptography

Integer lattices are used in public key cryptography: as tools to break codes (Chor-Rivest, Merkle-Hellmann, RSA with small private exponent) but also in the definition of cryptosystems.

In most cryptosystems with lattices, the set M is a set of "small" elements of $\mathbf{Z}^n$, the set K' is a set of lattices, that have a "secret" structure; K is the same, with the secret structure hidden, usually giving a different lattice basis.

Let $k \in K$, then $R = k$ sends (m, r, k) into of $m + r$, and the secret structure of k is used in the decryption algorithm.

Integer lattices and Public key cryptography

Integer lattices are used in public key cryptography: as tools to break codes (Chor-Rivest, Merkle-Hellmann, RSA with small private exponent) but also in the definition of cryptosystems.

In most cryptosystems with lattices, the set M is a set of "small" elements of $\mathbf{Z}^n$, the set K' is a set of lattices, that have a "secret" structure; K is the same, with the secret structure hidden, usually giving a different lattice basis.

Let $k \in K$, then $R = k$ sends (m, r, k) into of $m + r$, and the secret structure of k is used in the decryption algorithm.

We examine now three such cryptosystems:
GGH, McEliece-Niederreiter, NTRU.

GGH (Goldreich-Goldwasser-Halevi)

In GGH, (the security-equivalent Micciancio variant) a lattice M has a secret basis that is almost orthogonal; recovering the basis (or an equivalent one) is hard; a message is a vector with small components, and is encrypted adding to it a lattice element.

Decryption is made finding the equivalent point in the fundamental parallelogram with respect to the secret basis. It works correctly if the message is sufficiently small and the basis is sufficiently orthogonal (if the Hadamard bound is nearly attained).

McEliece-Niederreiter

In McEliece, (the security-equivalent Niederreiter variant) the private key E is a Goppa error-correcting code in $\mathbf{Z}/p$; the public key is E concealed by a change of variables. A message is a vector of small Hamming weight; to encrypt, a random element of E is added to it. Decryption is made recovering the error using the key equation, that needs the secret change of variables.

Not knowing the Goppa structure, an attacker can only proceed with generic algorithms, inefficient since the problem is NP-complete.

Remark that a $\mathbf{Z}_p$ vector space lifted back to a $\mathbf{Z}$-module is a lattice. Hence this is a lattice cryptosystem too. The difference wiith other lattice cryptosystems is the use of the Hamming distance instead of the Euclidean distance.

NTRU

In NTRU, in $A = \mathbf{Z}_q[X]/(X^n - 1)$ the public key is $h = pg/f$, where g, f are "small" polynomials (with small coefficients and small support, i.e. small Hamming weight; p is a public small element, e.g. $p = 2$); a message m is a small polynomial, and is encrypted as $c = rh + m$. r being a random small polynomial.

Everything can be seen in the submodule $M \subseteq A^2$ generated by $((q, 0)(h, 1))$: (g, f) is a small element of M (recovering it breaks the key), and (m, r) is short and equivalent mod M to $(c, 0)$. Both are hard to find, for suitable choice of the parameters.

Lattice cryptography is fashionable!

A few years ago, quantum computing has been introduced. A quantum computer can use quantum phenomena to perform exponentially many computations in polynomial time, and algorithms have been found that can factor and solve discrete logarithms in polynomial time. In a breakthrough success, the largest existing quantum computer has succeeded in factoring this big composite number:

Lattice cryptography is fashionable!

A few years ago, quantum computing has been introduced. A quantum computer can use quantum phenomena to perform exponentially many computations in polynomial time, and algorithms have been found that can factor and solve discrete logarithms in polynomial time. In a breakthrough success, the largest existing quantum computer has succeeded in factoring this big composite number:

$$15$$

Lattice cryptography is fashionable!

A few years ago, quantum computing has been introduced. A quantum computer can use quantum phenomena to perform exponentially many computations in polynomial time, and algorithms have been found that can factor and solve discrete logarithms in polynomial time. In a breakthrough success, the largest existing quantum computer has succeeded in factoring this big composite number:

$$15$$

Hence alternatives to RSA and Diffie-Hellman have become urgent. Lattice cryptography is one for the front-runners in this race.

Gröbner bases and Public key cryptography

Of course, Gröbner bases can be used in cryptoanalysis, and have been used successfully. We are instead interested mainly in the use of Gröbner bases in the design of public key cryptosystems.

The basic idea that many people had is the following:

Let $I \subseteq A = k[X]$ be an ideal, for which we know a Gröbner basis, and publish a set $F = \{f_i\}$ of elements of I (a special case is when I is maximal, and a Gröbner basis is a solution of $\{f_i = 0\}$).

Encryption of an element $m \in A$ is done as $m + \sum g_i f_i$, the g_i being chosen adequately, and m can be recovered through normal form computation if it is a linear combination of staircase elements.

Gröbner bases and Public key cryptography

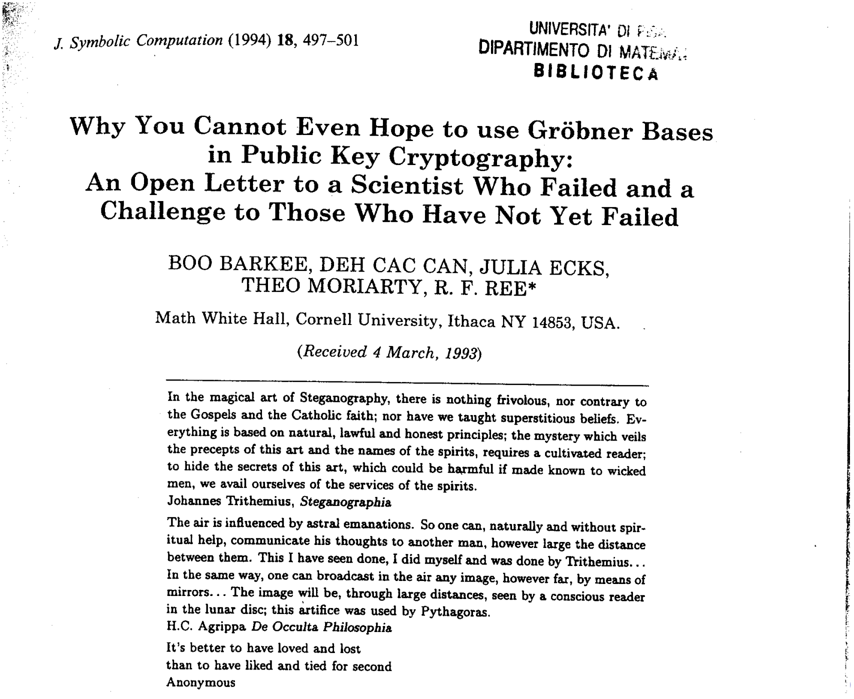
Of course, Gröbner bases can be used in cryptoanalysis, and have been used successfully. We are instead interested mainly in the use of Gröbner bases in the design of public key cryptosystems.

The basic idea that many people had is the following:

Let $I \subseteq A = k[X]$ be an ideal, for which we know a Gröbner basis, and publish a set $F = \{f_i\}$ of elements of I (a special case is when I is maximal, and a Gröbner basis is a solution of $\{f_i = 0\}$).

Encryption of an element $m \in A$ is done as $m + \sum g_i f_i$, the g_i being chosen adequately, and m can be recovered through normal form computation if it is a linear combination of staircase elements.

This turned out to be a very bad idea:



J. Symbolic Computation (1994) **18**, 497–501

UNIVERSITA' DI [illegible]
DIPARTIMENTO DI MATEM[illegible]
BIBLIOTECA

Why You Cannot Even Hope to use Gröbner Bases in Public Key Cryptography: An Open Letter to a Scientist Who Failed and a Challenge to Those Who Have Not Yet Failed

BOO BARKEE, DEH CAC CAN, JULIA ECKS, THEO MORIARTY, R. F. REE*

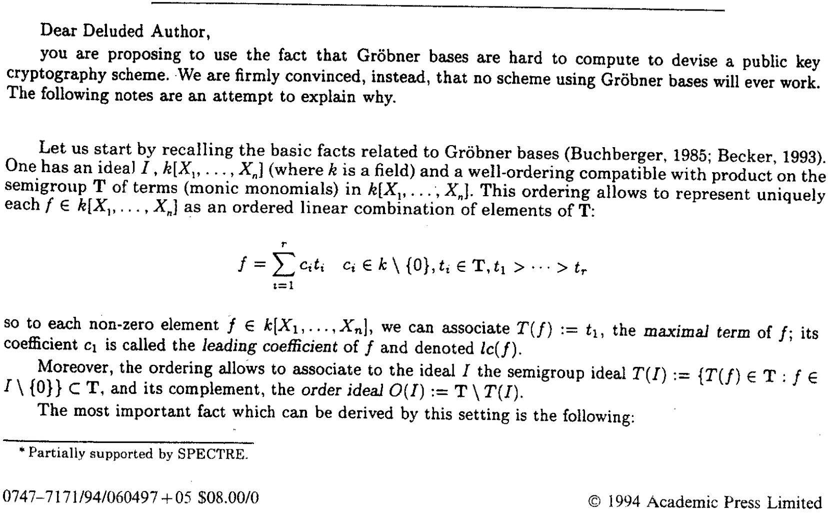
Math White Hall, Cornell University, Ithaca NY 14853, USA.

(*Received 4 March, 1993*)

In the magical art of Steganography, there is nothing frivolous, nor contrary to the Gospels and the Catholic faith; nor have we taught superstitious beliefs. Everything is based on natural, lawful and honest principles; the mystery which veils the precepts of this art and the names of the spirits, requires a cultivated reader; to hide the secrets of this art, which could be harmful if made known to wicked men, we avail ourselves of the services of the spirits.
Johannes Trithemius, *Steganographia*

The air is influenced by astral emanations. So one can, naturally and without spiritual help, communicate his thoughts to another man, however large the distance between them. This I have seen done, I did myself and was done by Trithemius... In the same way, one can broadcast in the air any image, however far, by means of mirrors... The image will be, through large distances, seen by a conscious reader in the lunar disc; this artifice was used by Pythagoras.
H.C. Agrippa *De Occulta Philosophia*

It's better to have loved and lost
than to have liked and tied for second
Anonymous



Dear Deluded Author,

you are proposing to use the fact that Gröbner bases are hard to compute to devise a public key cryptography scheme. We are firmly convinced, instead, that no scheme using Gröbner bases will ever work. The following notes are an attempt to explain why.

Let us start by recalling the basic facts related to Gröbner bases (Buchberger, 1985; Becker, 1993). One has an ideal I, $k[X_1, \dots, X_n]$ (where k is a field) and a well-ordering compatible with product on the semigroup $\mathbf{T}$ of terms (monic monomials) in $k[X_1, \dots, X_n]$. This ordering allows to represent uniquely each $f \in k[X_1, \dots, X_n]$ as an ordered linear combination of elements of $\mathbf{T}$:

$$f = \sum_{i=1}^{r} c_i t_i \quad c_i \in k \setminus \{0\}, t_i \in \mathbf{T}, t_1 > \cdots > t_r$$

so to each non-zero element $f \in k[X_1, \dots, X_n]$, we can associate $T(f) := t_1$, the *maximal term* of f; its coefficient c_1 is called the *leading coefficient* of f and denoted $lc(f)$.

Moreover, the ordering allows to associate to the ideal I the semigroup ideal $T(I) := \{T(f) \in \mathbf{T} : f \in I \setminus \{0\}\} \subset \mathbf{T}$, and its complement, the *order ideal* $O(I) := \mathbf{T} \setminus T(I)$.

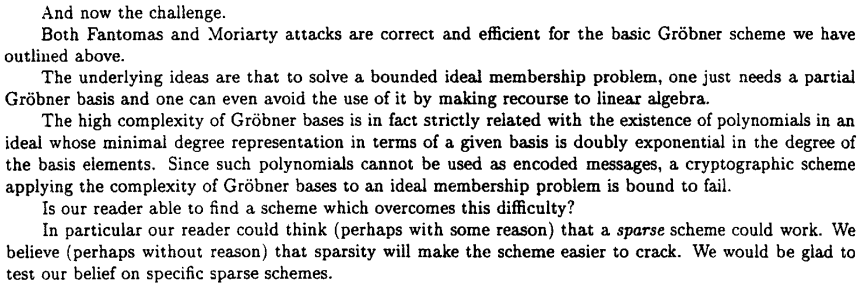
The most important fact which can be derived by this setting is the following:

* Partially supported by SPECTRE.

© 1994 Academic Press Limited

The argument

The main point of the paper is that the hardness of computing the Gröbner basis (that is doubly exponential in the worst case) cannot be used, since you don't need a Gröbner basis to decrypt: you only need to guess the g_i that have been used in the encryption, and this can be made by linear algebra. Cracking the message is in the same complexity class of encrypting it.



And now the challenge.

Both Fantomas and Moriarty attacks are correct and efficient for the basic Gröbner scheme we have outlined above.

The underlying ideas are that to solve a bounded ideal membership problem, one just needs a partial Gröbner basis and one can even avoid the use of it by making recourse to linear algebra.

The high complexity of Gröbner bases is in fact strictly related with the existence of polynomials in an ideal whose minimal degree representation in terms of a given basis is doubly exponential in the degree of the basis elements. Since such polynomials cannot be used as encoded messages, a cryptographic scheme applying the complexity of Gröbner bases to an ideal membership problem is bound to fail.

Is our reader able to find a scheme which overcomes this difficulty?

In particular our reader could think (perhaps with some reason) that a *sparse* scheme could work. We believe (perhaps without reason) that sparsity will make the scheme easier to crack. We would be glad to test our belief on specific sparse schemes.

A list of failures, one main attack

Several attempts have been made to find very sparse systems, always based on the hardness of an underlying problem. And all the attempts have been attacked through the message. A list of attempts contains

- Fellows and Koblitz "Polly Cracker" aka "CA crypto" (CA stands both for California and Combinatorial-Algebraic). Two examples, based on graph 3-coloring and graph perfect code. Many variables, polynomials with 2 to 4 monomials of degree 2.
  They fail both on the key (it is not known how to obtain hard solved instances of the combinatorial problems) and on the message (it can be recovered by a general attack described later).

"Polly Cracker" has become the generic names of the cryptosystems challenged by Barkee.

- SAT-3, based on satisfiability, for which heuristic criteria for unfeasibility exist. The key is secure, but the message is easy to find. Many variables, polynomials with 8 monomials of degree 3.
- EnRoot, based on roots of extremely sparse polynomial systems, with polynomials with 3 monomials and very high degree.
- PollyTwo, in which the ideal is a sum of two ideals, one defined by generators, the other by a map, the ideal being its kernel. Separating the two in a message is easy, and after the separation the parts are recovered easily.
- Non-commutative Polly Cracker, in which the non-commutativity adds further key attacks, and disturbs the encoder while easing the task of the attacker.

Enhanced differential attack

The attack, mainly due to Hofheinz and Steinwandt, reconstructs the support of the computation using the characteristic differences of the input polynomials.

Enhanced differential attack

The attack, mainly due to Hofheinz and Steinwandt, reconstructs the support of the computation using the characteristic differences of the input polynomials.

1. Find as many 2-nomials as possible in the ideal (a 2-nomial is a polynomial with 1 or 2 monomials, $f = aX^\alpha - bX^\beta$, a many-nomial has at least 3).
2. Compute the staircase of the ideal generated by the 2-nomials.

Enhanced differential attack

The attack, mainly due to Hofheinz and Steinwandt, reconstructs the support of the computation using the characteristic differences of the input polynomials.

1. Find as many 2-nomials as possible in the ideal (a 2-nomial is a polynomial with 1 or 2 monomials, $f = aX^\alpha - bX^\beta$, a many-nomial has at least 3).
2. Compute the staircase of the ideal generated by the 2-nomials.
3. Working inside of the staircase found in point 2, start with the support of the message, increase it adding all the "concealed monomials", those that can complete the support of a monomial multiple of a basis polynomial that matches at least two points of the support.

Enhanced differential attack

Phase 1 can be done through a modified Buchberger algorithm, in which a polynomial pair is considered critical if two cancellations happen when suitably multiplied. If the polynomials are sparse, only a few pairs appear, and after a few steps the completion is concluded.

Phase 2 might fail if phase 1 has discovered many 2-nomials. But a partial Gröbner bases is sufficient to attempt phase 3.

The support completion (phase 3) usually stops very soon too, and gives a reasonably small set on which linear algebra can be performed. But even if it does not stop, it is possible that with the partial support the g_i (in $c = m + \sum g_i f_i$) can be recovered. Step 2 however may fail. And of course 3 cannot work for binomial ideals.

More on binomials now.

Gröbner bases and Integer lattices

A binomial is a difference of power products, $X^\alpha - X^\beta$. Binomial ideals are ideals generated by binomials. The reduced Gröbner basis of a binomial ideal is composed of binomials.

To a vector of $\alpha \in \mathbf{Z}^n$ one can associate a binomial $X^{\alpha^+} - X^{-\alpha^-}$, where α^+ is $\sup(\alpha, 0)$ and $\alpha = \alpha^+ + \alpha^-$. Conversely, to $X^\alpha - X^\beta$ one associates $\alpha - \beta$, and one can define an associated lattice.

A lattice L has an associated ideal I_L, generated by its associated binomials. Ideal binomials are variable-saturated (if $x_i f \in I_L$ then $f \in I_L$, and the ideal associated to the associated lattice is its saturation.

Gröbner basis of a lattice

Through the ideal association one can define a Gröbner basis of a lattice; it can also be defined directly, S-polynomials of binomials correspond to difference of vectors.

The ideal generated by the binomials associated to a basis of a lattice might be smaller than the ideal associated to the lattice; computing the associated ideal is the harder part of the computation of the Gröbner basis.

Saturating the ideal generated gives the associated ideal, but it is possible to do better, with ad-hoc algorithms; the general purpose algorithms are highly inefficient to compute with binomial ideals, special shortcuts are possible.

F4 and F5 cannot handle binomial ideals efficiently, because of the extreme sparsity and dimension of the matrices involved; a plain Buchberger algorithm is better.

Applications of binomial ideals

Binomial ideals have several applications, to algebraic geometry (toric varieties), to combinatorics, to statistics, to biology, to linear optimization.

In particular, with lattice Gröbner bases one can solve linear optimization problems: given a non-homogeneous system of equations and inequalities one can find if a solution exists, and find the optimal solution with respect to a linear cost. The cost function is used to define the ordering of the polynomial ring.

Computing a Gröbner basis is an intermediate step for computing a Graver basis, an object that was defined independently in the context of integer programming, and contains a Gröbner basis.

Implementations of Gröbner bases for lattices

Specialized Gröbner basis algorithms for binomials and lattices have been implemented, but are not included in any general type computer algebra system.

There is an implementation included in CoCoA-4, but not as efficient as a dedicated implementation called 4ti2, by Hemmeke^2 and Malkin, whose core is apparently a GPL C++ library. The user interface of 4ti2 is extremely inefficient and clumsy.

Implementations of Gröbner bases for lattices

Specialized Gröbner basis algorithms for binomials and lattices have been implemented, but are not included in any general type computer algebra system.

There is an implementation included in CoCoA-4, but not as efficient as a dedicated implementation called 4ti2, by Hemmeke^2 and Malkin, whose core is apparently a GPL C++ library. The user interface of 4ti2 is extremely inefficient and clumsy.
(**Hint!!! Hint!!!**)

Heuristics for binomials

In particular, it has been remarked that the heuristic in the computation of a Gröbner basis of a binomial ideal is drastically different from the standard heuristics. In particular, a Lex Gröbner basis is usually much easier than with a generic ideal, and is by far the easiest ordering (although often useless for applications).

Heuristics for binomials

In particular, it has been remarked that the heuristic in the computation of a Gröbner basis of a binomial ideal is drastically different from the standard heuristics. In particular, a Lex Gröbner basis is usually much easier than with a generic ideal, and is by far the easiest ordering (although often useless for applications).

Recently we proved that for a full-dimensional lattice (corresponding to a zero-dimensional ideal) the Hermite normal form (that can be computed polynomially) gives almost immediately a Lex basis. This, although trivial to prove, has come as a big surprise. Lex basis can be computed for dimension up to a few thousand. In contrast, a DegRevLex basis of a lattice of dimension n in n variables generated by random vectors with components bounded by 20 has, heuristically, about 3^n elements, and the computation seems unfeasible for $n \geq 12$.

Gröbner bases, Integer lattices and Public key cryptography

Time now to bring everything together. Polly Cracker with a lattice Gröbner basis is a natural candidate. Call it LPC.

We have to do a few modifications to the basic Polly Cracker:

- a monomial is rewritten into one monomial. Hence every monomial has his own history, nothing is mixed, and coefficients are never modified; hence as messages we have to take individual monomials. This means that the set of messages will be a subset of the staircase (that has hence to be quite large).
- encrypting one monomial we can have only one monomial: if we have more, then it would be easy o match the pairs that are combined together into one lattice element. Hence the encryption consists in adding to a monomial a lattice element.

This puts LPC in line with the other lattice cryptosystem.

Block lattices

- We need to be able to compute a Gröbner basis, but it has to be difficult to recover for an attacker.
- The lattice should not have otherwise recognizable properties: it would be liable to attacks.
- The messages have to be inside the staircase. But the set of messages has to give out as little information as possible. It should be invariant under permutation of variables. An obvious choice is to allow all the vectors having components in $[0, s]$ for a predefined limit. We chose to experiment mainly with $s = 8$. This turned out to be a good choice. This means that the staircase has to be "fat".

"Easy" orderings like Lex cannot be used because they are easy also for an attacker, and they produce slim staircases in most variables. We chose to use block lattices and block orderings.

A block lattice

$$L = \begin{pmatrix} \bullet & \bullet & \bullet & \times & \times & \times & \times & \times & \times \\ \bullet & \bullet & \bullet & \times & \times & \times & \times & \times & \times \\ \bullet & \bullet & \bullet & \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \bullet & \bullet & \bullet & \times & \times & \times \\ 0 & 0 & 0 & \bullet & \bullet & \bullet & \times & \times & \times \\ 0 & 0 & 0 & \bullet & \bullet & \bullet & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \bullet & \bullet & \bullet \\ 0 & 0 & 0 & 0 & 0 & 0 & \bullet & \bullet & \bullet \\ 0 & 0 & 0 & 0 & 0 & 0 & \bullet & \bullet & \bullet \end{pmatrix}$$

The Gröbner basis with respect to a block ordering can be computed easily from the Gröbner bases of blocks on the diagonal. The “fatness” of the staircase only depends on these blocks. The multiplicity is equal to the determinant of the lattice, and this too only depends on the diagonal blocks.

How to choose the blocks

If the diagonal blocks have coprime determinants, it is easy to recover the blocks (just consider $\mathbf{Z}^n/L$ and split it as finite abelian group). It is hence useful to have all the blocks with the same determinant, and possibly each determinant being a power of a small prime.

There is no apparent risk in choosing a power of 2 as determinant of the blocks.

It is useful that $\mathbf{Z}^n/L$ has a rich structure (it should not be cyclic, but sum of many cyclic groups). This can be done easily choosing adequately the entries with $\times$ (the entries near to the diagonal should be multiples of a power of 2).

Tight blocks

It is useful that the messages are a large percentage of all the staircase elements. This means that the determinant has to be small, and the staircase fat. For reasons that will be apparent later, we chose blocks of dimension 4, with staircase containing $[0,8]^3 \times [0,24]$, i.e. one of the variables extended three times more than the others, and determinant 2^{15}. This is tight, since $2^{14} < 9^3 \cdot 25 < 2^{15}$.

Indeed, finding such lattices is difficult: randomly, one in 10000 with the given determinant satisfies the condition.

Concealing the blocks

The initial attempt has been to conceal the block structure changing the basis. But it is impossible to conceal the block structure: one can compute the order in $\mathbf{Z}^n/L$ of the coordinates, (using the isomorphism with a sum of cyclic groups, i.e. the Smith normal form) and the blocks reappear: the variables in the smaller blocks have smaller order.

Concealing the blocks

The initial attempt has been to conceal the block structure changing the basis. But it is impossible to conceal the block structure: one can compute the order in $\mathbf{Z}^n/L$ of the coordinates, (using the isomorphism with a sum of cyclic groups, i.e. the Smith normal form) and the blocks reappear: the variables in the smaller blocks have smaller order.

We have hence to introduce a change of coordinates. We will have public coordinates with a public lattice (used for encryption) and private coordinates with a private block lattice (used for decryption). But the message space that is a hypercube in public coordinates becomes a more general parallelogram in private coordinates. This is the reason of the extra width in some coordinates: the message space can become more elongated in these directions.

Stretching further

The change of coordinates (apart from a coordinate permutation) is taken lower-triangular, with elements off the diagonal corresponding to the variables with extra space: the l_1 norm of each column has to be at most the stretching factor.

This is how the change of variables acts on monomials: we add to the exponent of each "fat" variable a sum of the exponents of the other variables, according to the corresponding columns.

Stretching further

The change of coordinates (apart from a coordinate permutation) is taken lower-triangular, with elements off the diagonal corresponding to the variables with extra space: the l_1 norm of each column has to be at most the stretching factor.

This is how the change of variables acts on monomials: we add to the exponent of each “fat” variable a sum of the exponents of the other variables, according to the corresponding columns.

To have more space, (allowing a larger choice of changes of coordinates) we multiply these coordinates by a further factor of 4; this brings the determinant of each block to 2^{17}, and the staircase can fit a block $[0, 8]^3 \times [0, 99]$ that means that one can stretch in that direction 12 times.

The keys

Now the public key is a lattice, in addition the private key has a change of coordinates in a special form. In the private coordinates the lattice has block form: one can compute easily a Gröbner basis and use it for decryption.

So the trapdoor information is not the Gröbner basis, it is the change of variables.

Of course, a permutation of variables has to be included in the change of coordinates.

Attacking the private key

Now the public lattice has no further information on the blocks. Every lattice, if we allow changes of coordinates, (that, by definition, change the lattice in a different one) can be brought in block form. It is just enough to remark that every abelian group with 2^{17n} elements can be represented as a chain of extensions of groups with 2^{17} elements.

We need to represent each one of these blocks as a lattice with a staircase of special shape (and this can be done) but we also need to fit the transformed message space into the staircase that we obtained.

This is an optimization problem, that is non-linear (the change of coordinates that we build is the inverse of the one that has to be minimized) and has all the characteristics of a hard optimization problem.

Indeed, we have tried to recover the smallest block with a linearization of the optimization problem, using lattice reduction, and while for dimension sufficiently small (up to 24 blocks of 4 variables) often the smallest block can be retrieved, and up to 32 blocks recovering the smallest vector in that block is sometimes possible, it has never been possible to complete the lattice reduction for 40 blocks (dimension 160).

We conjecture that for dimension 200 (50 blocks) the key attacks are expected to fail with any current computing power; this is also the expectation for the message attacks, that have never succeeded for dimension larger than 100 unless the range of the message has been artificially reduced.

Optimize normal form

A naïve implementation of normal form is completely inefficient. To decrypt, one needs a clever normal form.

First, one can reduce block by block, starting from the largest ones. It is only necessary to reduce the top part, and use linear algebra to reconstruct the tail.

Second, to reduce one block it is better to initially find an approximation with the Babai roundoff algorithm, then complete the normal form with Gröbner normal form. Combining the two one obtains the normal form very fast.

Finally, one can remark that one knows the order of the group, that is the determinant of the lattice, hence any element exceeding that value can be reduced modulo the determinant. This is probably only useful to give a complexity bound through a limit on the coefficient growth.

encryption as normal form

Since the Lex Gröbner basis is easy to find, hence the Lex normal form is easy too, it is always possible to find a canonical form of a cryptogram (this is true for every lattice cryptosystem, and has been already remarked by Micciancio).

It is esthetically pleasant to define encryption as reduction of a message to Lex normal form, and decryption as reduction to the block normal form in the secret coordinates. But it is also space efficient, and can optimize message encryption.

This means that, as public key, we can provide the Lex Gröbner basis.

Cryptographic parameters

The resulting cryptosystem us the following parameters; given n the lattice length:

- Message size $3.17n$, cryptogram size $4.25n$, expansion of a factor of 1.37 (independent of the dimension).
- public key size $4.25n^2$, private key size marginally larger (the estimate is heuristic, depending on the Gröbner bases, but apparently less than a factor 10).
- Encryption complexity $18n^3$ byte multiplication (using standard integer multiplication), decryption heuristically of the same order.

A prototype implementation

These are some timings in seconds for our prototype implementation (excluding the preliminary search for blocks):

Dim.	Prep.	Encr.	Decr.
1024	6906.56	11.51	68.89
768	2073.40	3.63	26.14
512	342.96	1.82	9.29
384	108.25	0.89	4.28
256	23.70	0.35	1.28
192	9.09	0.20	0.54
128	2.93	0.08	0.23

Conclusions

In conclusion, we have shown how to combine Gröbner bases and lattice to define a public key cryptosystem, that seems superior in many respects to existing lattice cryptosystems, answering Barkee's challenge.

It is now Barkee's turn to try and break our proposal.

This is joint work with M. Caboara and F. Caruso.

More material is available at

`http://posso.dm.unipi.it/crypto`