CARLO TRAVERSO

Groebner Bases

in Public Key Cryptography:

is there still hope?

Journes 2007 de l'ANR GECKO
Sophia Antipolis.
19 Novembre 2007

Everybody tried that:

Find an ideal I with a simple Gröbner basis G. Find a few polynomials $\{f_i\} = F$ in I, such that the Gröbner basis of (f_i) is too hard to find. The public key is F and a finite set M of canonical monomials (coinciding with their normal form). The private key is G. A message is a linear combination of M, and is encoded adding to it a polynomial combination of F. Decoding is done through normal form.

This does not work

as explained in the paper*:

Boo Barkee, Deh Cac Can, Julia Ecks, Theo Moriarty, R. F. Ree: "*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.*" Journal of Symbolic Computation (18)6, 497 - 501 (1994)

Some of the reasons are good, some are bad, many have tried since, all have failed.

*Partially supported by SPECTRE

Barkee's reasons

There is no need to use Buchberger algorithm, or even to compute a Gröbner basis: since the encoding is done with a short process, a bounded computation of a truncated Gröbner basis through linear algebra is enough to grant decoding.

The only possibility is to use very sparse polynomial 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 basis 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.”

Barkee was right!

In a recent work, a new cooperation between

- Franziska Löw ben Bezalel
- Miss M.G. (Mary Grace) Marple
- Theo Moriarty,
- Ludovic Poirot
- C.T. (Cabdulqadir Tariiq) Garweyne

revised all the recent (and even unpublished) research, and the conclusion remains the same.

To try to get more consideration in the academic community, they have chosen pseudonyms matching well-known researchers:

Françoise Levy-dit-Vehel, Maria Grazia Marinari, Teo Mora, Ludovic Perret, Carlo Traverso,
A Survey on Polly Cracker Systems,
to appear in in the Linz workshop volume.

Polly Cracker

Fellows, M. Koblitz, N.
"*Combinatorial cryptosystems galore!*"
in "Finite Fields: Theory, Applications, and Algorithms", Contemporary Mathematics, VOL 168, (1994).

3-colouring of graphs
perfect code in graphs

3-SAT (Levy-dit-Vehel, Perret)

EnRoot (Grant, Krastev, Lieman, Shparlinski)
Polly2 (Le Van Ly)
NC-Polly (Tapan Rai)
Monoid rings (Ackermann, Kreutzer)

3-colouring of a graph

Let $X = \{x_i\}$ be the set of vertices of a graph, and $V \subseteq X \times X$ the set of his edges. Let $C = \{R, G, B\}$ a set of colours; $X \times C = \{x_{i,c}\}$ a set of indeterminates, a 3-colouring of X is the assignement to each $x_{i,c}$ of 1 or 0, being 1 iff x_i has colour c, under the condition that each vertex has a colour, and adjacent vertices have different colours. In equations,

$$\begin{gathered} x_{i,c}^2 = x_{i,c} \\ x_{i,R}x_{i,G} = x_{i,R}x_{i,B} = x_{i,G}x_{i,B} = 0 \\ x_{i,R} + x_{i,G} + x_{i,B} = 1 \\ (i,j) \in V, r \in C \Rightarrow x_{i,r}x_{j,r} = 0 \end{gathered}$$

Perfect code in a graph

Given a graph $\{x_i\} = X$ with edges $V \subseteq X \times X$ a subset $C \subseteq X$ is a perfect code if the minimum distance of C is 3, and any sphere $S(x_i)$ of radius 1 has exactly one element of C. In equations.

$$X_i^2 = X_i$$
$$\sum_{x_j \in S(x_i)} X_j = 1$$

A weakness of these graph systems is that it is not known how to produce hard solved instances of 3-colourable graphs, or graphs with perfect codes. Random graphs seem to give origin to easily solvable systems.

SAT-3

The satifiability problem consists in the following: given a logical formula (composed of propositional variables P_i, negation $\neg$ and connectives $\wedge, \vee$ find an assignement of truth values to the variables making the formula to evaluate to TRUE. SAT-3 means that the formula is $\bigwedge(X \vee Y \vee Z)$, each X, Y, Z being either P_i or $\neg P_i$; it is known how to produce formulas that are hard to solve with every known method.

Fix a field K, two elements $T, F \neq 0$ and a variable x_i for each propositional variable P_i. A clause P_i is represented by $x_i - T$, $\neg P_i$ by $x_i - F$, $X \vee Y \vee Z$ by the equation equating the product to 0, and the whole formula by a conjunction of the equation. Each equation is a cubic with 8 terms. Solutions of the system correspond to solutions of the SAT problem.

$$(P_1 \vee \neg P_2 \vee P_3) \Leftrightarrow (x_1 - T)(x_2 - F)(x_3 - T)$$

EnRoot, Polly Two

EnRoot is a scheme that relies on polynomials with few monomials of high degree, in few variables, constructed to vanish at a root defined a priori as the private key. It is subject to standard attacks to the message.

PollyTwo adds a map and the kernel of such a map to polynomials of high degree in few variables with few monomials. Although cleverly constructed to resist some linear algebra attacks, still succumbs to standard attacks to the message.

Non-commutative Polly Cracker

This is an attempt to use non-commutative polynomials, since Gröbner bases are usually infinite in non-commutative polynomials.

While this property considerably limits the choice of the private Gröbner basis (that is hence limited to principal ideals) does not succeed to avoid the standard attacks, that can be performed with truncated Gröbner bases.

Moreover, finitely determined infinite Gröbner bases are also possible (and software exists to construct them), and additional attacks based on non-commutative factorisation are possible too due to the fact that the private key is a single polynomial.

A non-commuative factorisation algorithm is present in a long-lost and forgot manuscript of James H. Davenport, that has been recently retrieved in the secret archives of T. Moriarty.

Monoid rings polly

P. Ackermann, M. Kreuzer *Gröbner basis cyptosystems*, J. Appl. Alg. **17** (2006) 173–194 propose to define generalizations of Gröbner bases and use them to define cryptosystems.

There is however no concrete proposal, only examples that show that any cryptosystem can be interprted in this framework, just showing that the proposal is a pure illusion.

The weakness of Polly Cracker: differential attack

Even if the underlying problem is NP-hard, it is possible that the Gröbner basis of I for a random item is easy to compute; but the attacks can be made to the message.

The preparation of a PollyCracker cryptogram consists in preparing an obscuring element $h \in I$ and if m is the message, $c = h + m$ is the cryptogram.

We have $h = \sum \phi_i f_i$, the f_i being the public key, and if we can guess the monomials involved in the computation then linear algebra can be used (like in the F4 algorithm).

Usually one can easily find some polynomials with 1 or 2 monomials in the ideal I (others may exist, but harder to find). Assume that the Gröbner basis of the ideal J generated by these 2-nomials (the 2-nomial sub-ideal) is easy to compute: the quotient $K[X]/J$ as vector space is our playground. This leaves polynomials with at least 3 monomials.

Because of the extreme sparsity, usually adding a monomial multiple of some polynomial just cancels one monomial and introduces at least two. Hence the chain of reductions cannot be long, and the last one performed has left at least two monomials. It is hence easy to identify the element used, and find a monomial that has been removed with the last reduction, or a few candidates. Backtracking the reduction may be exponential (if multiple guesses are possible), but never substantially worse than what the encoder has done.

The basic setting is described in D. Hofheinz, R. Steinwandt *A "Differential" Attack on Polly Cracker.* Int. J. Inf. Secur. **1** (2002) 143–148. This is a more elaborate version:

Let c be the cryptogram to decode, $F = \{f_i\}$ the public key, T the set of monomials that can compose a message.

If S is a set of monomials, let
$F_S = \{X^\alpha f_i \mid \text{Supp}(X^\alpha f_i) \subseteq S \cup T\}$

```
Let $S = \text{Supp}(c)$
LOOP:
  IF $\exists$ a linear combination $L$ of $F_S$
      such that Supp($c - L$) $\subseteq T$
    THEN return $c - L$;
  ADD to $S$ the support of all the $X^\alpha f_i$
      that meet $S$ in at least two monomials;
  IF $S$ has not increased,
    THEN return FAIL
```

"Monomials" here can be "standard monomials for the 2-nomial sub-ideal J" and the computations made mod J.

Toric Polly Cracker

The reasoning breaks for ideal generated only by binomials. Here every reduction replaces one monomial with another, the encoder can perform long chains of reductions without exponential growth. The decoder for backtracking has always multiple choices, hence an exponential growth.

Hence binomials ideals are the remaining hope for a successfull Polly Cracker.

The step from binomial ideals to toric ideals, and from these to lattices is short.

Lattice cryptosystems have been studied, and are the last resort to meet the Barkee challenge. Unfortunalely, most of them have been broken.

Toric ideals and lattices

A toric ideal is an ideal generated by binomials, and saturated by the variables.

The correspondence $X^\alpha - X^\beta \Leftrightarrow \alpha - \beta$ transforms toric ideals into lattices. Differences/sums correspond to S-Poly/tail reductions. One can compute a Gröbner basis of a lattice, via lattice operations.

Open problem: how can lattice tools (lattice reductions, LLL) and ideal tools (Gröbner bases) interact?

There are current investigations in course (M. Caboara, F. Caruso, C.T.) on this and in the rest of the talk.

Toric Polly

Reduction through binomials always transforms monomials into monomials; hence one has to encode monomials.

This means that a message is a vector, and encoding means adding to it a random lattice element.

The set of allowed messages is a set of vectors in $\mathbf{Z}^n$ (uniquely represented modulo the lattice).

The public key is a set of lattice vectors (not necessarily generating the lattice).

The private key is a set of vectors of the lattice, possibly a Gröbner basis, but a subset of a Gröbner basis, sufficient to decode (most of) the messages might be enough.

GGH

Consider the Goldreich-Goldwasser-Halevi cryptosystem, as described in the paper "*Public-Key Cryptosystems from Lattice Reduction Problems*", Advances in Cryptology - CRYPTO '97.

The private key is a reduced (= almost orthogonal) basis of a lattice. It allows to solve the CVP (closest vector problem) for the lattice.

The public key is a different basis of the lattice. Recovering a reduced basis is hard, and the CVP is hard.

Encoding is done through the sum of a lattice vector V and a small vector E (the message can be either V or E, the other being random). This is like McEliece cryptosystem, that allows using either the codeword or the error as message.

GGH fits in our definition of toric Polly Cracker. Unfortunately, the system is considerd broken, due to improvements of lattice reduction techniques. It is however much more robust than any other Polly Cracker ever designed.

However the decoding is not done through polynomial reduction.

NTRU

Following Nguyen, P. Q. Stern, J. *Lattice Reduction in Cryptology: An Update* ANTS-IV LNCS 1838 (2000) the only lattice-based cryptosystem still resisting is NTRU,

J. Hoffstein, J. Pipher, J. H. Silverman, *NTRU: A Ring-Based Public Key Cryptosystem*, ANTS III (1998),

and looking at the recent LLL+25 conference:

Nick Howgrave-Graham, *Practical lattice-based cryptography: NTRUEncrypt and NTRUSign*

(and `http://www.ntru.com`) it is still resisting. So it is worth considering.

NTRU (cont.)

Strictly speaking, NTRU is not a lattice cryptosystem, it is a polynomial algebra cryptosystem, but it can be attacked by lattice algorithms (and resists to these attacks).

The basic connection is: an ideal $I \subseteq \mathbf{Z}[X]/\phi(X)$, where ϕ is a monic polynomial, is a lattice, invariant under multiplication by X. In particular, if $\phi(X) = X^n - 1$ it is a lattice invariant under cyclic permutation of variables. Let $A = \mathbf{Z}[X]/X^N - 1$.

It is fun that and ideal can be seen as a lattice, that in turn can be seen as a toric ideal. We are unable to exploit this remark.

NTRU (cont.)

NTRU requires to fix two modules, q and p (to fix ideas, take q prime and $p = 3$, p, q coprime). Encoding is done in A/q, decoding partly in A/q, partly in A/p.

A polynomial is *small* if its coefficients are in $\{0, 1, -1)\}$ and its support is small (size to be determined). It is *moderate* if its coefficients are smaller than $q/2$

The private key is a pair (f, g) of small polynomials, f, g invertible mod q, f invertible mod p too.

The public key is $h = f_q^{-1}g$
(f_q^{-1} is the inverse mod q).

The public key is $h = f_q^{-1} g$

The message is a small polynomial m, and is encoded as $c = phr + m$, r being a small random polynomial.

To decode, first compute fc.

We have $fc = pgr + fm \bmod q$, see it as an element of A, (not A/q).

Reduce $fc = pgr + fm \bmod p$ and multiply by f_p^{-1}. This is $m \bmod p$, and since m is small, the message is recovered.

The decoding works correctly if $pgr + fm$ is moderate, that is ensured, at least probabilistically, by the bounds on f, g, r, m.

(If $p = 2$, the message is the support of m, the signs are random).

Attacks to NTRU

Since $h = f_q^{-1}g$ is not moderate, one needs f to decode (indeed, any small f' such that $hf' \bmod q$ is small is OK). Finding f' breaks the key.

Or one can solve a CVP (closest vector problem) to recover m without recovering the private key.

There are three conflicting needs:

a) protect the private key
b) protect the message
c) allow decoding with the private key.

Key security requires that the supports of f and g are large, message security requires that the supports of r and c are large, and the size of $pgr + fm$ (that we need to bound) depends on the sizes of g, r, f, m: we can increase r, m only decreasing f, g.

Lattice attacks to the NTRU key

Consider $A \oplus A$. It can be seen as $\mathbf{Z}^N \oplus \mathbf{Z}^N$ Consider the sub-A-module M generated by $(q, 0)$ and $(h, 1)$. It is a lattice that contains (g, f) since $hf = g$ mod q.

(g, f) is a small vector in the lattice, and can be found by LLL or a variant, if it is too small. If it is sufficiently large, it will be difficult to find, if it is even larger it will be impossible to find through SVP, not being a small vector in the lattice.

Lattice attacks to the NTRU message

If the private key is robust, one can try to attack the message. Consider the vector $(c, 0)$; the lattice contains the vector $(prh \bmod q, pr)$ that might be the closest vector to $(c, 0)$, since the difference is $(m, -pr)$ and is small. We need to solve a CVP.

For this, one can try to solve a SVP (shorter vector problem) in the lattice $(M, 0)+(c, 0, 1) \subseteq A \oplus A \oplus Z$.

If $(m, -pr, 1)$ is sufficiently large the answer will be difficult, if it is not the shortest vector it will not be found.

Because of the various constraints it is impossible to unconditionally protect key, message and decoding, but a compromise is possible. And ntru.com asserts that it is safe.

Our plans concerning NTRU

We have three objectives in studying NTRU:

1) develop attacks to NTRU improving mixed Gröbner-LLL methods

2) develop a Toric Polly Cracker using the NTRU lattice

3) improve NTRU protecting the key, while allowing very small f and g (hence allowing to increase r and m, i.e. message security).

While 2) might obviously respond to Barkee's challenge[†], 3) might respond to it too in a more surprising way.

[†] *"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."*