Lattice Polly Cracker cryptosystems

Massimo Caboara, Fabrizio Caruso, Carlo Traverso

Dipartimento di Matematica, Università di Pisa, Pisa, Italy

ARTICLE INFO

Article history:
Received 31 January 2009
Accepted 11 February 2010
Available online 13 October 2010

Keywords:
Gröbner basis
Hermite normal form
Polly Cracker
Cryptosystem
Lattice

ABSTRACT

Using Gröbner bases for the construction of public key cryptosystems has been often attempted, but has always failed. We review the reason for these failures, and show that only ideals generated by binomials may give a successful cryptosystem.

As a consequence, we concentrate on binomial ideals that correspond to Euclidean lattices. We show how to build a cryptosystem based on lattice ideals and their Gröbner bases, and, after breaking a simple variant, we construct a more elaborate one. In this variant the trapdoor information consists in a “small” change of coordinates that allows one to recover a “fat” Gröbner basis. While finding a change of coordinates giving a fat Gröbner basis is a relatively easy problem, finding a small one seems to be a hard optimization problem.

This paper develops the details and proofs related to computer algebra, the cryptographic details related to security, the comparison with other lattice cryptosystems and discusses the implementation.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Cryptology basically has two aspects: cryptography, i.e. the design of ciphers or other cryptographic protocols, and cryptanalysis, i.e. attempts to recover the message without the knowledge of the secret key, or to recover the secret key itself.

Since every cryptosystem can be expressed as a set of polynomial equations, Gröbner bases have been increasingly used as a cryptanalytic tool, and after some striking success on long standing...
challenges, see e.g. Faugère and Joux (2003), are now regarded as a standard tool in algebraic cryptanalysis, see Cid and Weinmann (2009).

Using Gröbner bases to build cryptosystems has been attempted, but has not been successful. The reason has been explained, accounting for the failures of the authors themselves, in Barkee et al. (1994), (the hiding authors are indeed the “scientist who failed”) and their prophecy has proved to be true up to now. In Levy-dit-Vehel et al. (2009) these attempts, collectively known under the name of Polly Cracker (Fellows and Koblitz, 1993; Levy-dit-Vehel and Perret, 2004; Ackermann and Kreuzer, 2006; Ly, 2006), have been analyzed, together with the attacks that have broken them. The conclusion has been that there is still no hope of creating a satisfying cipher on these lines with the possible exception of binomial ideals. This possibility has been partially developed in Caboara et al. (2008), which contains moreover a first prototype of a cryptosystem based on Gröbner bases of lattices. We called this cryptosystem Lattice Polly Cracker (LPC).

In this paper we elaborate the cryptanalytic tool for Polly Cracker cryptosystems extending the differential attack of Hofheinz and Steinwandt (2002) already sketched in Levy-dit-Vehel et al. (2009) and Caboara et al. (2008). We build a cryptosystem scheme using Gröbner bases of lattices, extending and reinforcing the one sketched in Caboara et al. (2008), and discuss the lattice attacks that can recover the private key or an equivalent one in the simple version and fail because of the algorithmic complexity beyond the simplest cases in the extended version.

It turns out that the fundamental element on which the security of LPC relies is not, as one expected, the difficulty of computing a Gröbner basis. This is an easy computation for zero-dimensional lattice ideals (binomial ideals associated to a lattice of maximal rank) if one is allowed to choose the ordering. And the task of finding a Gröbner basis suitable for the decryption is as difficult for the designer of an instance of the cryptosystem as for the attacker, hence not crucial for the security; showing that, after all, Barkee et al. (1994) was right also in this case. The trapdoor information is a change of variables that transforms the private lattice into the public lattice (the public key is a Lex Gröbner basis of the public lattice).

Retrieving such a change of variables from the public key means finding an integer solution of a non-linear system of equations and inequalities, which seems to be very hard both in theory and in practice.

Note that the resulting cryptosystem, relying on hard lattice problems, seems to be secure against quantum computing attacks, hence it is an example of a Post Quantum Cryptography protocol, see Bernstein et al. (2008).

2. Polly Cracker, and why it fails

Polly Cracker is the common name for a series of cryptosystems using Gröbner bases as trapdoor information (Ackermann and Kreuzer, 2006).

The key pair is a pair of ideals in a multivariate polynomial ring \( J \subseteq I \subseteq k[X] \), in which a Gröbner basis \( G \) of \( I \) is secretly known, and a Gröbner basis of \( J \) is hard to compute, for example hard in space or even infinite (in the non-commutative case). The set of messages \( \mathcal{M} \) is a vector space of polynomials generated by a subset of the staircase of \( I \), \( R \) is a subset of \( J \) and a message \( m \in \mathcal{M} \) is encrypted as \( m + r \), with \( r \in R \). Decryption is made computing the normal form modulo \( G \).

The special case that was initially considered in Fellows and Koblitz (1993) is the case that \( I \) is a maximal ideal corresponding to a solution of a set of polynomial equations generating \( J \). Choosing a system of equations modelling an NP-complete problem was assumed to guarantee the security of the cryptosystem. Note that Fellows and Koblitz (1993) is simultaneous to Barkee et al. (1994), and is not directly broken by their analysis, so the authors can be considered, at the time of the paper, rather as “scientists that have not yet failed”, and indeed some interaction between the two can be discovered in the formulation of the “challenge” on one side, and on some considerations in Koblitz (1998) on the other side.

The assumption hides a double fallacy. First, NP-completeness (under the \( P \neq NP \) hypothesis) guarantees security only asymptotically and in worst case, while cryptographic security requires intractability in a random, finite case (but some proposals have been able to overcome this difficulty, see e.g. Levy-dit-Vehel and Perret (2004)). Second, and more importantly, message attacks do not
require the solution of a generic normal form problem, but only of a normal form of a message that has been built in bounded time with a predefined procedure. Side attacks to the encryption procedure, i.e. attacks that recover how the random $r$ has been built from the information available in the public key, are possible.

In particular, if the polynomials generating $J$ are sufficiently dense, either a truncated Gröbner basis can be computed (the Fantômas attack of Barkee et al. (1994)) or linear algebra in polynomials of bounded degree (Moriarty attack) can be used. If the polynomial ring has many variables or the polynomials are of high degree then the direct application of linear algebra is infeasible, but some form of the differential attack of Hofheinz and Steinwandt (2002) can recover a small support in which sparse linear algebra will then be successful. We will detail this last method in the next section.

These attacks, or similar ones, have defeated all the Polly Cracker variants proposed up to now. See Levy-dit-Vehel et al. (2009) for an update.

3. The differential and the 2-nomial attacks

The differential attack of Hofheinz and Steinwandt (2002) has been used successfully in several variants of Polly Cracker, exploiting the sparsity of the generators $f_i$ of the public ideal $J$.

The cryptogram is $c = m + r = m + \sum g_i f_i$ (where $m$ and the $g_i$ are not known) and we can expand it formally as $\sum a_\alpha m_\alpha + \sum b_{\beta\gamma} g_i\delta f_i,\gamma$ where each term is a monomial. Because of the sparsity, one expects that cancellations are sporadic, hence in the result at least two monomials of one of the $f_i$, multiplied by one of the monomials of one of the $g_i$, will appear. We can then subtract from $c$ a suitable monomial multiple of $f_i$ and proceed inductively. If everything goes well at the end we are left with $m$.

This does not work if some of the $f_i$ have just one or two monomials (such polynomials are called 2-nomials). In the case that one can find several 2-nomials in $J$, one can find more 2-nomials computing the Gröbner basis of the ideal generated by them, and this has been sometimes sufficient to crack some of the proposed cryptosystems, see Steinwandt et al. (2002).

The two attacks seem to apply to two completely different kinds of systems, but they can be combined into one more powerful attack.

Indeed, when one has an ideal $I$ of $k[X]$ containing another ideal $J$, and one has a a Gröbner basis of $J$, one can consider $I$ as an ideal of $k[X]/J$ that one can represent via linear algebra. In this case, if $J$ is generated by 2-nomials, $k[X]/J$ is a vector space generated by the staircase of $J$, and multiplication is performed as in the polynomial ring, but identifying some monomials either with 0 or with another monomial being a multiple of a staircase element; hence a linear algebra attack, far from being more difficult, is made easier.

One might try to conceal binomial elements of $I$ inside of the generators, but these are easily discovered, for example with a modified Buchberger algorithm that considers only those $S$-polynomials where at least two monomials cancel. This algorithm, in the very sparse case, will usually stop quite soon, and possibly recover a few 2-nomials. Computing the Gröbner basis of the 2-nomial part $J$ one might either perform a linear algebra attack or restart the modified Buchberger algorithm in $k[X]/J$.

The remaining hope is reduced to the difficulty of computing a Gröbner basis of a 2-nomial ideal; so we turn now to the analysis of binomial ideals, which are a simplified form of 2-nomials independent of the ground field, and are a better understood class, linking to different algebraic structures of cryptographic interest.

4. Binomials and lattices

A 2-nomial of the form $X^\alpha - X^\beta$ is called a binomial.\(^1\) An ideal generated by binomials is called a binomial ideal, and its Gröbner basis, in whatever ordering, is composed of binomials. Here we report

\(^1\) As usual, if $\alpha = (a_1, \ldots, a_n) \in \mathbb{N}^n$, $X^\alpha$ denotes $\prod x_i^{a_i}$. 
results that are well known, refer to Bigatti et al. (1999). In the next section we prove some less known or even new results.

An integer lattice, lattice for short, is a subgroup \( L \) of \( \mathbb{Z}^n \) with the Euclidean distance inherited by \( \mathbb{Z}^n \). A lattice has a basis, which is usually represented as a matrix, whose rows are the elements of the basis. The number of rows is the rank of the lattice; when the rank is \( n \) (the lattice is of full rank) this matrix is square, and its determinant is the cardinality of \( \mathbb{Z}^n/L \). A lattice described giving as basis the rows of a matrix \( A \) is denoted by \( L_A \).

Lattices represented by matrices \( A, B \) are isomorphic (an isometric isomorphism exists) if and only if integer square matrices \( X, Y \) exist such that \( A = XBY \), where \( X \) has determinant \( \pm 1 \) and \( Y \) is a permutation matrix. \( X \) corresponds to a change of basis and \( Y \) corresponds to an orthogonal transform of \( \mathbb{Z}^n \).

Given a binomial ideal \( I \), the set \( \{ \alpha - \beta \mid X^\alpha - X^\beta \in I \} \subseteq \mathbb{Z}^n \) is a lattice \( L_I \). Given a lattice \( L \), the ideal \( L_I \) generated by \( \{ X^{v_+} - X^{v_-} \} \), where \( v_+, v_- \) are non-negative with disjoint support and \( v_+ - v_- \in L \), is a binomial ideal. Moreover, \( L_{L_I} = L \) and \( L_{I^J} = L :^* X \), the saturation of \( L \) with respect to the indeterminates.

Given a lattice \( L \) and a basis \( (v_i) \) of \( L \), it is not true that the ideal \( I_{(v_i)} \) generated by \( b_i = X^{v_i+} - X^{v_i-} \) is equal to \( L_I \), but it is true that \( I_{(v_i)} :^* X = L_I \).

A Gröbner basis of \( L_I \) is composed of binomials; these binomials can be represented by vectors that form a set of generators called the Gröbner basis of the lattice.

The staircase of a lattice \( L \) is defined as the staircase of the lattice ideal \( L_I \), and the normal form of a non-negative vector that corresponds to a monomial is the unique element of the staircase equivalent to it, modulo the lattice. If the lattice \( L \) is of full rank we can define the normal form for every vector, since any vector is equivalent to a non-negative vector.

5. The lattice ring

Several properties of the lattice \( L \) correspond to algebraic properties of the lattice ideal \( L_I \) and of the lattice ring \( k[X]/L \). This object is not invariant under lattice isomorphism, so we study a slightly different object, and prove that the two are isomorphic in the case of our interest, i.e. when the lattice is of full rank.

The polynomial ring \( k[X] \) is a monoid algebra \( k[\mathbb{N}^n] \), but a lattice \( L \) is not a submonoid of \( \mathbb{N}^n \), it is a subgroup of \( \mathbb{Z}^n \), hence the natural object to study \( L \) is the group algebra \( k[\mathbb{Z}^n] \). We use the following lemma:

\textbf{Lemma 1.} Let \( G \) be a group, \( H \) an invariant subgroup; consider the canonical map \( \phi : k[G] \to k[G/H] \); then the kernel of \( \phi \) is the ideal generated by the \( (g - g') \), where \( gH = g'H \).

\textbf{Proof.} It is indeed the \( k \)-vector space generated by the \( (g - g') \), since \( G \) and \( G/H \) are bases of \( k[G] \) and \( k[G/H] \) respectively. \( \square \)

Denote \( J(H) = \ker \phi \). The following is immediate:

\textbf{Lemma 2.} If \( L \) is a lattice, then \( L_I = J_L \cap k[X] \).

\textbf{Proof.} Multiplying any generator of \( J_L \) by a suitable power product we obtain an element of \( I_L \). Recall that \( I_L \) is variable-saturated. \( \square \)

If \( L \) is a lattice, \( L^+ \) denotes the elements of \( L \) having positive coordinates.

\textbf{Lemma 3.} Let \( L \) be a full-rank lattice; then \( L^+ \) is non-empty.

\textbf{Proof.} Note that \( L \otimes \mathbb{Q} \subseteq \mathbb{Z}^n \otimes \mathbb{Q} \) is equal to \( \mathbb{Q}^n \); any totally positive element of \( \mathbb{Q}^n \) has hence an integer multiple in \( L \). \( \square \)

\textbf{Corollary 4.} If \( v_1, v_2 \in \mathbb{Z}^n \) then there exists \( v \in L \) such that \( v_1 + v, v_2 + v \) have positive coordinates. Any element of \( L \) can be represented as \( v_1 - v_2, v_1 \in L^+ \).
**Corollary 5.** The homomorphism $k[X]/I_L \rightarrow k[\mathbb{Z}^n/L]$ is injective, and if $L$ is a full-rank lattice, it is an isomorphism.

**Proof.** The canonical homomorphism $k[X]/I_L \rightarrow k[\mathbb{Z}^n]/J_L$ is injective by **Lemma 2** and surjective by **Corollary 4.**

To test ideal membership in $I_L$, it is hence sufficient to test zero equality in $k[\mathbb{Z}^n/L]$. This can be computed using the following theorem:

**Corollary 6.** Let $f = \sum a_i X^{\alpha_i} \in k[X]$. Then $f \in I_L$ if and only if $\forall i \sum_{\alpha_j - \alpha_i \in L} a_j = 0$.

**Proof.** Note that $\alpha_j - \alpha_i \in L$ means equality in $\mathbb{Z}^n/L$, and can be checked through linear algebra. Then use **Corollary 5.**

**Theorem 7.** Let $L$ be a full-rank lattice, of determinant $d$; then the multiplicity of $I_L$ is equal to $d$.

**Proof.** The multiplicity for a zero-dimensional ideal is the $k$-dimension of $k[X]/I$.

**Theorem 8.** Let $A = (a_{ij})$ be an integer square matrix, upper triangular, $a_{ii} > 0$ and assume that $-a_{ij} < a_{ij} \leq 0$ if $i < j$. Then the rows of $A$ are the reduced Lex Gröbner basis of the lattice $L$ defined by $A$.

**Proof.** Since the diagonal elements are positive, and the elements above the diagonal are non-positive, the $i$-th row corresponds to the binomial $x_i^{a_{ii}} - \prod_{j > i} x_j^{a_{ij}}$. Since the ordering is Lex, the head is $x_i^{a_{ii}}$. These elements are a Gröbner basis $G$, since the heads are pairwise coprime. We have to show that $G$ generates the ideal of the lattice.

The multiplicity of the ideal generated by $G$ is $\prod a_{ii}$, which is also the determinant of the lattice, i.e. the cardinality of $\mathbb{Z}^n/L$. This proves that the ideal generated by $G$, which is contained in the ideal of the lattice, coincides with it.

The inequalities $-a_{ij} < a_{ij} \leq 0$ prove that the tail of every element of $G$ cannot be reduced by the head of another element of $G$, hence $G$ is a reduced Gröbner basis.

If $L$ is of full rank, then a Lex Gröbner basis of $I_L$ is easily computed from the Hermite Normal Form (HNF) of $L$. Indeed, in this case the HNF is the unique upper-triangular matrix generating $L$ such that $0 \leq a_{ij} < a_i$ if $i < j$, and the Lex basis is the unique upper-triangular matrix generating $L$ such that $0 \geq a_{ij} > -a_{ij}$ if $i < j$. Passing from the one to the other form is immediate through Gaussian reduction with straightforward row operations.

Note that the Hermite normal form can be computed in polynomial time, see Kannan and Bachem (1979) and subsequent extensive literature on the subject, and efficient implementations are widely available, e.g. Shoup (2009). Hence the computation of a Lex Gröbner basis of a full-rank lattice is trivial up to several hundred variables.

To compute a Gröbner basis with other orderings may be much harder: experimentally the DegRevLex Gröbner basis of an $n \times n$ random matrix with entries in $[-20, 20]$ has approximately $3^n$ elements, and the computation time required grows exponentially with respect to $n$. This of course holds as far as experiments can go, i.e. for $n \leq 11$.

### 6. Lattice Polly Cracker (basic version)

Polly Cracker for lattice ideals can be defined as in the general case, but some special properties of lattice ideals require specific handling. Let $L$ be a lattice and $I = I_L$.

As in any binomial ideal, the normal form of a monomial is a monomial. Let $f$ be a polynomial; **Corollary 6** allows one to find an $f'$ equivalent to $f$ modulo $I_L$ with the smallest support, aggregating in one all the equivalent monomials. Then finding the normal form is equivalent to finding the normal form of all the monomials.
This implies that encryption and decryption in a Polly Cracker based on a binomial ideal can be made one monomial at a time, encrypting a monomial with another monomial, since encrypting one monomial with many monomials does not increase the security.

This motivates the following preliminary definition of a cryptosystem that we call Lattice Polly Cracker, or LPC:

- The set of messages is a subset \( \mathcal{M} = [0, s]^n \subseteq \mathbb{Z}^n \).
- The public key is a lattice \( L \subseteq \mathbb{Z}^n \) such that two different elements of \( \mathcal{M} \) are not equivalent modulo \( L \).
- The encryption procedure replaces \( m \in \mathcal{M} \) with \( m + l, l \in L \).
- The private key is composed of a term ordering and a corresponding Gröbner basis \( G \) of \( L \); the staircase is assumed to contain \( \mathcal{M} \).
- The decryption of a cryptogram \( c = m + l \) is done by computing its normal form \( NF(c) \) modulo \( G \).

The cryptosystem will be generalized later, since the initial version will be insecure; this definition will be only used to show that it is insecure, and to motivate the generalization, which will inherit the name.

Some comments: it is possible to choose the set of messages differently: however, adapting \( \mathcal{M} \) to the staircase might disclose details on the term ordering or on the Gröbner basis, hence \( \mathcal{M} \) should be invariant under the symmetric group. A subset \( \mathcal{M}' \) of \( \mathcal{M} \) might be chosen to restrict messages for security reasons; this will be discussed later.

The choice of \( L \) in the encryption procedure is largely irrelevant, since it is always possible to find a normal form of a message from its encrypted form, for example through the round-off algorithm (Babai, 1986) or through the Hermite normal form (Micciancio, 2001); this shows that for a practical cryptosystem an additional layer should be added; this is however a standard cryptographic tool, and will not be discussed here.

The key property that allows decryption is the fact that the staircase contains the message space. The message space being \([0, s]^n\), this is equivalent to requiring that \( (s, s, \ldots, s) \) is in the staircase. The condition that the messages are not equivalent mod \( L \) entails that the determinant \( d \) of \( L \) has to be larger than the cardinality of \( \mathcal{M} \), i.e. \( (s + 1)^n \); security considerations require that \( d \) is not much larger than \( (s + 1)^n \), e.g. not much more than \( 2^n(s + 1)^n \). These conditions make it difficult to provide examples of an LPC. This however makes recovering a Gröbner basis suitable to decrypt messages difficult too.

We will use informally the concept of fat and slim staircases. A staircase is fat (\( s \)-fat) if it contains \((s, s, \ldots, s)\), and is hence suitable to be used in the decryption of an LPC, it is slim otherwise. Breaking an LPC instance means finding a Gröbner basis of the public lattice whose staircase is fat.

7. Block Lattices and their Gröbner Bases

To instantiate a Polly Cracker system, it is necessary to have a lattice \( L \), a basis \( A \) of the lattice, and an ordering \( \prec_L \), such that the Gröbner basis of \( L \) can be computed from \( A \), and such that the staircase is “fat” enough.

The first problem is the choice of a class of suitable term ordering. Lex orderings are inadequate, since the staircase is usually slim for the larger variables, and forcing the lattice to have a fat Lex staircase reduces its genericity to much. A DegRevLex ordering is usually impossible to compute for a sufficiently large lattice. Our choice is to use block orderings, and we show that it is possible to efficiently compute in practice a Gröbner basis of a lattice given through a block-triangular basis with respect to a block ordering. A lattice with a block generating matrix and a block ordering will be called a block lattice. Every lattice can be represented as a block lattice, since the Hermite normal form is a block matrix.

Given a list of positive integers \( B = (b_1, \ldots, b_m) \) and \( n = \sum b_i \), let \( a_j = \sum_{i=1}^j b_i \) (hence \( a_0 = 0 \)). An \( n \times n \) matrix \( L = (l_{i,j}) \) is called a B-block triangular matrix (block matrix for short) if \( s \leq a_i < r \) implies \( l_{i,j} = 0 \). The submatrix \((l_{i,j})\) with \( a_{i-1} < r, s \leq a_i \) is called the \( i \)-th diagonal block, and \((l_{i,j})\) with \( a_{i-1} < r \leq a_i \leq s \) is called the \( i \)-th tail. The determinant \( d \) of the matrix is the product of the
A block structure induces a filtration. Theorem 11. We prove that vectors $v_i$ of the diagonal blocks. The list $B$ is the block structure, and is usually (but not always) taken as constant, i.e. $b_i = b_j$.

**Remark 9.** A block structure induces a filtration $\mathcal{M}'$ of $\mathbb{Z}^n$, $M'_i = 0^n \oplus \mathbb{Z}^{n-a_i}$, and consequently a block decomposition of a lattice $L$ induces a filtration $L_i = L \cap M_i$ and corresponding subgroups $M_i/L_i$ of $\mathbb{Z}^n/L$, where the $M_i$ correspond to the $M'_i$ with variables shuffled. The order of $(M_i/L_i)/(M_{i+1}/L_{i+1})$ is the determinant $d_i$ of the $i$-th diagonal block.

The $M_i/M_{i+1}$ are torsion-free; a torsion-free filtration of $\mathbb{Z}^n$ induces a block decomposition up to a (non-isometric) automorphism of $\mathbb{Z}^n$.

Given a block structure $B$ and a vector $v \neq 0$, we can split the entries of $v$ in blocks according to $B$. If $i$ is the first non-zero block of $v$ (from left to right) then the head of $v$, $H(v)$, is a vector with as many components as $v$, all zero except for the $i$th, which is equal to the $i$th component of $v$. The tail of $v$ is $T(v) = v - H(v)$.

Associated with a block structure, we have a block ordering in the polynomial ring $k[X]$: power products are compared considering the head of the difference. The comparison can be made via any predefined term ordering for the individual blocks, but we will usually adopt the DegRevLex ordering.

**Remark 10.** Given a block structure, a vector $v$ and its corresponding polynomial $p_v = X^{v_T} - X^{v_H}$, if the head of the vector has no negative component and the tail has no positive component, then the leading monomial of $p_v$ corresponds to the head of $v$.

**Theorem 11.** Let $B$ be a block structure with an associated block ordering, and $A$ a B-block triangular matrix. Let $G_i = (g_{i,j})$ be the Gröbner basis of the $i$-th diagonal block $A_i$ of $A$; then the reduced Gröbner basis $G$ of $A$ is composed of vectors $v_{i,j}$ whose head corresponds to $g_{i,j}$, and whose tail is a vector with non-positive components.

**Proof.** We prove that vectors $v_{i,j}$ as detailed in the hypothesis are a Gröbner basis, and show how to find them from $A$ and the $G_i$ via linear algebra. The uniqueness of a reduced Gröbner basis concludes the proof.

Let $A_i$ be the $i$-th diagonal block of $A$, let $d_i = \det A_i$; note that $\prod d_i = \det A$.

A set $(f_j)$ of polynomials of $I$ is a Gröbner basis if and only if the monomial ideal generated by the heads of the $f_j$ is equal to the initial ideal of $I$; if $I$ is zero-dimensional it is enough to test that the multiplicities are equal, see Traverso (1996). Since $G_i$ is a Gröbner basis, the multiplicity of the corresponding initial ideal is $d_i$, for Theorem 7; since the tails of the $v_{i,j}$ have non-positive components, the tail variables do not appear in the leading term, hence the multiplicity of $\text{In}(v_{i,j})$, the initial ideal of $\{v_{i,j}\}_{i,j}$, is $\prod d_j = d$, which is the determinant of $A$; hence $v_{i,j}$ is a Gröbner basis, again for Theorem 7.

It remains to show that such a set exists, and the theorem follows from the uniqueness of a reduced Gröbner basis. We show how to obtain such a set via linear algebra.

Since the $G_i$ is composed of vectors of the lattice generated by $A_i$, a matrix $N_i$ exists such that $N_i A_i = G_i$. Consider the matrix $N$ composed of the diagonal blocks $N_i$; $NA$ does not satisfy the non-positivity assumption of the thesis, but with further elementary row operations this can be achieved without modifying the blocks $G_i$. □

As a consequence, the staircase of a block lattice with respect to a block ordering is just the product of the staircases of the diagonal blocks.

Although this theorem allows one to compute a Gröbner basis, it is seldom useful to compute one. To compute the normal form of a vector modulo a block lattice it is enough to have the Gröbner bases of the diagonal blocks.
Let \( \nu \) be a vector, \( \nu = H(\nu) + T(\nu) \). First, find the normal form \( \tilde{\nu} \) of \( H(\nu) \) modulo \( A_i \); this can be computed via \( G_i \); note that it is not a lattice element, (since the rows of \( A_i \) are not rows of \( A \)) but we can find a row vector \( m \) such that \( \nu - mA_i = \tilde{\nu} \). Consider now \( A_i \), the submatrix of \( A \) composed of the rows corresponding to the \( i \)-th block; the rows of \( A_i \) are lattice elements, and \( \nu' = \nu - mA_i \) is a vector equivalent modulo \( A \) to \( \nu \) and having head \( \tilde{\nu} \). Since \( \tilde{\nu} \) cannot be reduced by any Gröbner basis element, the normal form of \( \nu \) is \( \tilde{\nu} + NF(\nu' - \tilde{\nu}) \). Since \( \nu' - \tilde{\nu} \) has its head in a smaller block, inductively we find the normal form of \( \nu \).

We note, moreover, that to find the normal form of a vector \( \nu \) modulo a Gröbner basis, even one that does not have a block structure, it is not necessary to proceed with Buchberger division from the start; it is enough to find first a lattice vector \( w \) “sufficiently close” to \( \nu \), and perform Buchberger division on \( \nu - w \); for this, it is useful that \( \nu - w \) has non-negative components, hence one can add a positive vector \( u \) to \( \nu \), find an approximate closest vector \( w \) to \( u + \nu \), and look at \( \nu - w \); if it still has negative components, increase \( u \). This heuristic procedure usually performs quite well.

8. Breaking the basic block LPC; Hermite and Smith normal form

Computing a Gröbner basis of a lattice is easy for certain (block) orderings: for example a Lex Gröbner basis can be computed in polynomial time, and Lex is a block ordering for every block structure. But not all such bases are useful: with very high probability the staircase of a randomly chosen block ordering will not be fat enough to be used in an LPC.

Reordering the basis of \( \mathbb{Z}^n \), or changing the basis of \( L \), destroys the original block structure.

However, retrieving the block decomposition (which variables are in which block) of the private key, although not enough to find the private key, is a big step forward. One still has to find the ordering to be used inside the blocks, but this is a problem limited in few variables, and should hence be considered feasible; when one can guess the original block decomposition, the private key should be considered broken.

At first sight, retrieving the block decomposition is hard, since it looks like guessing a permutation of \( n \) variables, \( n \) large, but it is not. We will show that the block decomposition of the variables is easily guessed assuming that the block structure and the determinants \( d_i \) are known. This is plausible, since the product \( d_i \) is the determinant \( d \) of the lattice and is known, the fatness \( s \) of the staircase is known since it has to fit the messages, the blocks should be small (to allow the computation of the Gröbner basis), each \( d_i \) must be larger than \( (s + 1)^{b_i} \), and \( d \) has to be not much larger than \( (s + 1)^n \) (to ensure the security against message attacks).

**Lemma 12.** Let \( L \) be a block lattice, and let \( d_m \) be the determinant of the smallest block. Then for every variable \( x_i \) of the smallest block the polynomial \( x_i^{d_m} - 1 \) is in the lattice ideal.

**Proof.** Consider the lattice \( L_m \) defined by the last block \( A_m \) (a square matrix of dimension \( b_m \), hence \( L_m \subseteq \mathbb{Z}^{b_m} \)); then \( \mathbb{Z}^{b_m} / L \) is a group of order \( d_m \), hence every element has order dividing \( d_m \). **Corollary 5** allows one to conclude. \( \square \)

One can find for every variable a univariate polynomial, (e.g. by computing a Lex Gröbner basis) whose head term is necessarily of the form \( x_i^{d_m} \), and this usually reveals the block decomposition, since variables contained in the \( i \)-th block are only forced to have order dividing \( d_i \cdot \ldots \cdot d_m \), hence sorting the variables in decreasing order usually gives the block decomposition. (It is possible to design block matrices avoiding this kind of attack, but introducing this extra structure will most likely allow other attacks).

**Remark 13.** To compute the order of all the variables, one can compute one Lex basis, i.e. one Hermite normal form, for every variable. This however is an overkill. We can compute the order of any element in \( \mathbb{Z}^n / L \) by computing once the Smith normal form. Indeed, let \( A \) be a square matrix corresponding to a basis of the lattice, and let \( X, Y \) be invertible integer matrices such that \( XAY = \Delta \) is a diagonal matrix. \( \Delta \) defines a lattice \( L' \), and \( Y \) defines an automorphism of \( \mathbb{Z}^n \) bringing \( L \) into \( L' \). This automorphism does not preserve the Euclidean metric, since \( Y \) is not in general orthogonal; but computing the order of an element in \( \mathbb{Z}^n / L' \), which is explicitly a direct sum of cyclic groups, is immediate, and this is enough to compute the order of any element.
The same method allows one to compute \( L : c = \{ v \in \mathbb{Z}^n \mid cv \in L \} \) for any \( c \in \mathbb{Z} \). It is just sufficient to note that if \( L \subseteq \mathbb{Z} \) then \( L = m\mathbb{Z} \), and \( m\mathbb{Z} : c = d\mathbb{Z} \), \( d = m/\text{GCD}(d, m) \).

9. Lattice Polly Cracker

We have shown that to define a Lattice Polly Cracker we can use block matrices, but we have to conceal the lattice block structure.

We tried first to conceal it by publishing only a sublattice, but this does not work, since the sublattice inherits the block structure. The solution that we found is to conceal the block structure through a change of coordinates, i.e., a (non-isometric) automorphism of \( \mathbb{Z}^n \). The concept of a block structure and block decomposition of a lattice is not invariant under group isomorphism.

We have hence a lattice \( L \subseteq \mathbb{Z}^n \) to be used for decryption and a lattice \( L_{\text{pub}} \subseteq \mathbb{Z}^n \) that is used for encryption. They are connected through an automorphism \( \phi \) of \( \mathbb{Z}^n \) sending \( L_{\text{pub}} \) into \( L \). Since \( \phi \) is not an isometry, the lattices are not isomorphic (hence for example a closest vector problem might give different results).

This modification entails deeper modifications than just going back and forth from \( L \) to \( L_{\text{pub}} \) (or, better, from \( \mathbb{Z}^n / L \) to \( \mathbb{Z}^n / L_{\text{pub}} \)) since we have to modify the decryption procedure.

We now give the final definition of a Lattice Polly Cracker cryptosystem:

- The set of messages is a subset \( \mathcal{M} = [0, s]^n \subseteq \mathbb{Z}^n \).
- The public key is a lattice \( L_{\text{pub}} \subseteq \mathbb{Z}^n \) such that two different elements of \( \mathcal{M} \) are not equivalent modulo \( L_{\text{pub}} \).
- The encryption procedure replaces \( m \in \mathcal{M} \) with \( m + l, l \in L_{\text{pub}} \).
- The private key is composed of
  - an automorphism \( \phi : \mathbb{Z}^n \to \mathbb{Z}^n \); let \( L = \phi(L_{\text{pub}}) \).
  - a term ordering and a Gröbner basis \( G \) of \( L \).
  - a shift vector \( \tau \) such that \( \phi(\mathcal{M}) + \tau \) is contained in the staircase of \( L \).
- To decrypt a cryptogram \( c = m + \tau \) we compute \( \phi^{-1}(-\tau + \text{NF}(\phi(c) + \tau)) \).

The basic LPC defined previously corresponds to the case when \( \phi \) is orthogonal. The fundamental difference is that now the set of messages \( \mathcal{M} \) in the private coordinates is no longer a cube \( [0, s]^n \), and it is not even a set of vectors with non-negative coordinates, unless we apply a shift vector.

To define an LPC instance, one starts defining the set \( \mathcal{M} \) and the map \( \phi \); this defines the subset \( \phi(\mathcal{M}) \) and a shift vector \( \tau \) such that \( \tau + \phi^{-1}(\mathcal{M}) \) is contained in the cone of vectors with non-negative components. Then we have to define a lattice \( L \) such that its staircase contains \( \tau + \phi^{-1}(\mathcal{M}) \) and \( L_{\text{pub}} = \phi^{-1}(L) \). The map \( \phi \), being the trapdoor information, has to be chosen in a set that is sufficiently large; the staircase of \( L \) has to contain a \( n \)-dimensional rectangle whose sides are the \( l_i \) norm of the columns of the matrix \( X \) associated to \( \phi \), and for security reasons one wants to have the determinant of \( L \) as small as possible with respect to the volume of \( \mathcal{M} \). This poses some challenges, indeed \( \phi \), \( L \) and the term ordering have to be chosen jointly in such a way that \( L \) has a staircase that can fit \( \phi(\mathcal{M}) \), and that \( L \) has nevertheless a determinant that is sufficiently “tight” to ensure message security. This is true both for the construction, where we can decide beforehand \( \phi \) and build the lattice accordingly, and for the cryptanalysis, in which we have the lattice \( L_{\text{pub}} \) and we try to find \( \phi \).

10. Attacks to LPC and key security

With an automorphism of \( \mathbb{Z}^n \) involved, the problem has changed. To recover a private key (or an equivalent one) looking for a block decomposition of the public key \( L \) is useless, since the existing ones will not give a fat staircase.

To find a private key (either the original or an equivalent one) we need an automorphism \( \phi \) of \( \mathbb{Z}^n \) and a block decomposition of \( \phi(L) \); using Remark 9 we just need to find a filtration \( (M_i) \) of \( \mathbb{Z}^n \) that gives a block decomposition up to an automorphism \( \phi \) of \( \mathbb{Z}^n \). This can be made, but we have an additional constraint: not only do we need a block decomposition, we also need the image of \( \mathcal{M} = [0, s]^n \) to be contained in the staircase of the image of \( L \) under the automorphism. This implies a bound on \( \phi \).
The first problem is an integer linear algebra problem, hence easy to solve, but the second adds constraints, hence the problem becomes a hard integer programming problem. For specific (tight) values of the parameters one might have that \( \phi \) is not expected to exist for generic lattices, and hence for an LPC public lattice the solution is expected to be (substantially) unique and hence to coincide with the private key. Unfortunately we have not been able to prove such a result for the parameters that we suggest, but we have heuristic evidence supporting this conjecture.

Some of the information useful to find the private filtration might be considered known, since they can be restricted to a small, or at least not prohibitively large, number of cases; these include the size of the blocks, and the determinant of the diagonal blocks. In particular, the group structure of \( G = \mathbb{Z}/L \) can be easily computed, and if it is cyclic or is the sum of a few cyclic groups one might consider that the subgroups \( G_i = M_i/M_i \cap L \subseteq G \) are known. This does not mean that in this case the block structure is known, since recovering the \( M_i \) from the \( M_i/M_i \cap L \) in such a way that the map \( \phi \) is small still requires solving an SVP. But it could be a good idea to make the reconstruction of the subgroup chain \( (G_i) \) hard; this happens if \( \mathbb{Z}/L \) has many cyclic factors, and to maximize it, it is useful to use lattices with determinant power of 2. Experimentally, a large number of cyclic groups in \( \mathbb{Z}^n/L \) can be achieved with a block matrix having the determinant power of 2, and the blocks near to the diagonal a multiple of a power of 2. Hence we may assume that we know the determinants of the blocks that we have to reconstruct, but not the subgroups \( G_i \), and that an exhaustive search is impossible.

We do not have a formal reduction to a provably hard problem, but this is in common with most current cryptosystems. Lattice cryptosystems usually enjoy provable security, in the sense that breaking them means breaking all instances of a certain hard problem, but not all of them do, and the most handy GGH (Goldreich et al., 1997) and NTRU (Hoffstein et al., 1998) do not have this property.

11. Attacks to the block structure

Structure and attacks to the private key. We have shown that it is easy to compute a Gröbner basis of a block lattice with respect to a block ordering. Such computation is needed for key creation but it will be easy also for the attacker. It is hence essential for the block structure to be well hidden. To recover the block structure is not enough to recover the private key, since one has to choose the term ordering such that the staircase is sufficiently fat to allow decryption, and not every term-ordering is suitable. But relying on the secrecy of the term-ordering seems risky, hence we assume that recovering the block structure is enough for a total break of the key.

Because of the inductive character of the block structure, we consider that recovering the smallest block lattice might be a serious blow to the private key. On the other hand it seems unlikely that an equivalent private key might be built without recovering the smallest block. We will consider mainly attacks of this kind and try to design the structure in such a way to parry them.

Recover the smallest block. Assume that we have an instance of an LPC, through its public key (a basis of the public lattice) and message set; we want to reconstruct the matrix \( X \) (or an equivalent one) that gives the private change of variables \( \phi \) from the public lattice \( L_{pub} \). We know that \( X \) is small, since the product of its columns times the size of the message space is bounded by the determinant. We also assume we know the size and determinant of the smallest block, since in any case they are chosen in a small set (the determinant of the block has to be a factor of the determinant of the lattice).

The rows of the matrix \( Y = X^{-1} \) are the coordinates of the private lattice. Hence abandoning temporarily the condition on the staircase, and relaxing the condition on the columns of \( X \), the conditions are:

1. \( Y \) has as rows vectors \( v_i \) and we know \( n_i \) such that \( 2^{n_i} v_i \in L \).
2. \( Y^{-1} \) is small (its columns have small norm).

We relax the first condition, considering it for the last rows only, i.e. we aim at finding the last block.

We consider the second condition as an optimization problem on the set of solutions of the first. It is a non-linear problem (because of the inverse). But since \( X = X_c X_r \) and \( X_r \) is the sum of the identity matrix and a small nilpotent matrix, it can be approximated linearly if we require that \( Y \) is small.
The problem has hence become a lattice problem: find $m$ small linearly independent vectors in $L : d'$, where $m$ is the size of the last block and $d'$ is its determinant. This can be attacked with standard lattice reduction methods.

Of course, the simplified problem is only a heuristic solution, but we will show experimentally that it often gives a solution, at least a partial one, in low dimension. We consider it a full solution if the span of the $m$ smallest vectors of the reduced basis is equal to the smallest block, and a partial solution if the two intersect non-trivially. The goodness of a partial solution can be measured by the dimension of the intersection, or by the intersection of the smallest block with the space generated by the $cm$ smallest vectors for small multiples $c$.

So in large dimension, when the simplified problem becomes unfeasible with standard lattice techniques, we may consider breaking the key as unfeasible, at least with this type of attack.

We will discuss the experimental findings in a subsequent section.

12. Suggested parameters and implementation tips

Choosing the blocks. We outline here how we implemented the construction of LPC examples on which we conducted our tests.

Every block will be of dimension 4, each block except the last one will have determinant $2^{17}$ and its staircase will contain $(0, 8)^3 \times (0, 99)$. The last block will have determinant $2^{13}$ and the staircase will contain $(0, 8)^4$. This means that the message space is $(0, 8)^9$ and the matrix $X$ can have one every 4 columns (except the last 4) of $l_1$ norm at most 12. All the others will be of $l_1$ norm 1. We need hence to find many such blocks.

We consider several random matrices with given determinant (these are found filling randomly 3 columns, and choosing the fourth suitably). Then we compute their DegRevLex Gröbner basis.

For the smallest block, with determinant $2^{13}$, we discard all the matrices whose staircase is not 8-fat; experimentally, one block every $10^6$ survives. More are found through small deformations, and we currently have a small collection of several hundred blocks.

For the larger blocks, we first find random matrices with determinant $2^{15}$, compute the DegRevLex Gröbner basis, and discard those that do not have fatness $(8, 8, 8, 24)$ or a permutation. It turns out that on average we find more than one in $10^4$ samples. We have built a collection of several thousand.

Then we multiply the column of each block having fatness 24 by 4; the Gröbner basis with respect to a suitable ordering (in which the variable that has been multiplied has weight $1/4$) is the same, with the component multiplied by 4, and its fatness is $(8, 8, 8, 99)$.

The choice of a power of 2 as determinant is suggested to maximize the number of subgroups of $\mathbb{Z}^n/L$, and the exponent are chosen as small as possible: the staircase of the last block has at least $9^4 = 6561$ elements, that of the other blocks has at least $100 \cdot 9^3 = 72 900$, and the determinants are 8192 and 131 072 respectively. So the choices are tight.

In the construction of the full block matrix we may ensure that $\mathbb{Z}^n/L$ is far from cyclic. To this end we require that the elements of the tails of the blocks that are nearer to the diagonal are multiples of a power of 2. For example, if the first 16 elements of every tail are multiples of 4, one experimentally expects to have, in a lattice of 50 blocks (and dimension 200), $2^{120}$ elements in $\mathbb{Z}^n/L$ of order dividing $2^{13}$, instead of the $2^{13}$ of a cyclic group. Identifying the subgroup corresponding to the smallest block seems hard.

The block structure outlined allows the matrix $X$ to have weight 12 in one out of 4 columns, making exhaustive search attacks on $X$ impossible.

Public key, encryption. We represent the public key as a reduced Lex Gröbner basis, an upper triangular matrix with non-positive entries off the diagonal, substantially equivalent to the Hermite normal form.

Encryption of a message, in this case, can be defined as a Lex normal form. Namely, if $d_1, d_2, \ldots, d_9$ are the values on the diagonal of the matrix in Hermite normal form, a cryptogram is represented by a vector $c_1, c_2, \ldots, c_9$ with $0 \leq c_i < d_i$. In particular the bit length of a cryptogram is the bit length $l_d$ of the determinant $\prod d_i$. The bit length of the public key is $nl_d$. Considering the proposed parameters, each block (except the first) has determinant of bit length 17; while the message has bit length 12.68 per block. This means that we have message expansion 1.34.
Decryption: Normal form in lattices. Normal form modulo a Gröbner basis can be computed through Buchberger division, but for lattices it is inefficient, and should be used as a last resort; in the univariate case this corresponds to computing integer division remainder through repeated subtraction.

Moreover, it can only be used for vectors without negative components, if a vector has mixed signs one has first to find an equivalent vector without negative components.

In lattices, one can use a different procedure: for a vector \( v \), find a lattice vector \( l \) that is sufficiently near to \( v \); the normal form of \( v \) and of \( v \) \( \mod \) \( l \) are the same. Hence we proceed as follows:

- choose a vector \( w \) sufficiently small with positive components.
- choose an approximate closest lattice vector \( l \) for \( u = v - w \).
- if \( u + w \) has no negative component, compute the normal form \( u' \) of \( u + l \) via Buchberger division, otherwise choose a larger \( w \).

This might be a problem if the lattice is large, but we will show now that to find the normal form in block lattices it is enough to compute the normal form for the blocks.

Assume now that we have a block lattice, and that we have computed, for every diagonal block, its Gröbner basis.

Remark that having a basis \( B = (b_i) \) of a lattice, a vector \( v \) and its normal form \( w \), one can compute via linear algebra the coefficients \( a_i \) such that \( w = v - \sum a_i b_i \).

To compute the normal form of a full vector \( v \), we operate inductively on the head. Let \( v = v' + v'' \) being head and tail respectively; let \( B \) be the set of the rows of the block matrix having the head in the same block as \( v \), let \( B' \) be the corresponding diagonal block.

Find the normal form \( w' \) of \( v' \) with respect to the block Gröbner basis of \( B' \), and let \( A = (a_i) \) be the coefficients as above. Then \( w = v - AB \) is a vector whose head \( w' \) can no longer be reduced; find inductively the normal form \( w'' \) of \( w - w' \), then \( w' + w'' \) is the normal form of \( v \).

The result is the normal form, since it is inside of the staircase and equivalent to \( v \mod L \).

13. Implementation and experiments

We have implemented the cryptosystem and the attacks three times. A first prototype has been implemented in CoCoA, (CoCoATeam, 2009). This implementation has been used for experiments with more general types of LPC, and for comparisons. A Lisp and a C++ implementation has been hence developed. The Lisp implementation has been used to test and develop the algorithms, while the C++ implementation has been used for speed.

All rely on Shoup’s NTL (Shoup, 2009) for the LLL, BKZ and standard lattice algorithms, and 4ti2, (4ti2 team, 2009) for the lattice Gröbner basis algorithms.

The C++ implementation has been made by Daniele Trainini in a stage for the degree in Computer Science.

Sizes, complexity and timings. We can estimate quite accurately the size of the public and private keys, and the complexity of encryption and decryption. As parameter we take the number of the blocks.

A public key with \( m \) blocks is an \( n \times n \) matrix \( (n = 4m) \) of determinant \( 2^{17m-4} \) in Hermite normal form. This means in particular that every row of the matrix is a sequence of integers whose global bit length is bounded by \( 17m \). The size of the key is hence \( 68m^2 \). A key with 64 blocks (the minimal size that we consider secure) has size 35 KB, one with 256 blocks has size 0.5 MB.

The size of the private key is not much larger, but has size only marginally larger, with the same asymptotic growth: the private lattice is an \( n \times n \) matrix of small integers, the Gröbner basis of each block has approximately the same size, (estimated at most 50 elements per block), the private coordinate change is another \( n \times n \) matrix of small integers. The decryption time too can only be estimated, but it is cubic in \( m \), and apparently about 6 times the encryption time.

The message expansion rate is 1.34, as remarked in 12, although for security reasons that will be discussed later a subset of messages only might be used, bringing the message expansion rate up to 3.

These are the times needed for the key preparation, encryption and decryption for an LPC as described above, not including the preliminary preparation of the blocks, in the C++ implementation.
The times are in seconds, the largest part of the preparation time is the HNF computation to obtain the public key.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Creation</th>
<th>Encrypt</th>
<th>Decrypt</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>0.79 s</td>
<td>0.00054 s</td>
<td>0.00267 s</td>
</tr>
<tr>
<td>192</td>
<td>3.4 s</td>
<td>0.00120 s</td>
<td>0.00558 s</td>
</tr>
<tr>
<td>256</td>
<td>10.46 s</td>
<td>0.00228 s</td>
<td>0.0123 s</td>
</tr>
<tr>
<td>384</td>
<td>58.09 s</td>
<td>0.00622 s</td>
<td>0.0279 s</td>
</tr>
<tr>
<td>512</td>
<td>192.48 s</td>
<td>0.01070 s</td>
<td>0.0559 s</td>
</tr>
<tr>
<td>768</td>
<td>1195.71 s</td>
<td>0.02370 s</td>
<td>0.130 s</td>
</tr>
<tr>
<td>1024</td>
<td>4658.14 s</td>
<td>0.04160 s</td>
<td>0.345 s</td>
</tr>
</tbody>
</table>

These timings (in seconds) are averages of 1000 runs on an Intel Core 2 Duo E6750 @ 2.66 GHz running GNU/Linux Ubuntu 7.10.

**Message attacks.** Messages can be attacked through a closest vector problem. Since a message \( m \) is a vector in \([0, 8]\), and is encrypted as \( c = m + v, v \in L \) where \( L \) is the lattice whose basis is the public key, the vector \( v \) can be (presumably) retrieved as the closest vector to \((4, 4, \ldots, 4)\).

We have used the nearest plane algorithm (Babai, 1986), using a BKZ-reduced basis of the lattice. As starting basis of the lattice reduction we have tested either the public key, or the block lattice in public coordinates. The latter basis has proved to be more efficient, but is of course unavailable to an attacker: the lattice is always the same, but the reduction starting from the private basis gives a better reduced basis than starting from the HNF (i.e. the public basis).

We have tested several variants of messages, reducing the range of the vector: (1) The set \( \{0, 1, 7, 8\} \); (2) the set \( \{0, 1, 2, 6, 7, 8\} \). (3) the full interval \([0, 8]\); (4) The set \( \{1, 2, 6, 7\} \); (5) The interval \([1, 7]\); (6) The interval \([2, 6]\); (7) The interval \([3, 5]\); varying the number of blocks. We report the number of successes in cracking 100 random messages each for 10 different random LPC.

<table>
<thead>
<tr>
<th>Dim</th>
<th>BKZ from the public key</th>
<th>BKZ from the private key</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>40</td>
<td>926</td>
<td>952</td>
</tr>
<tr>
<td>60</td>
<td>889</td>
<td>912</td>
</tr>
<tr>
<td>80</td>
<td>855</td>
<td>897</td>
</tr>
<tr>
<td>100</td>
<td>119</td>
<td>120</td>
</tr>
<tr>
<td>120</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>140</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>160</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>180</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

We see from these data that reducing the range of messages makes them much more vulnerable, and instead using only the extreme range makes them more robust. An important part of the security is using a public key very far from a reduced lattice; starting near to the private key allows one to obtain a better reduced basis, increasing the successful attacks in some marginal situation.

Our conclusion is that for more than 50 blocks, i.e. lattice dimension 200 or more, the messages are secure with the current state of the art. Whether for such lengths the messages have to be restricted to the extreme values has not been concluded.

A further discussion comparing the method with the security estimates of other known cryptosystems will be done in Section 14.

**Key attacks.** We have tried the key attack to the smallest block; the attack has varying success for small block numbers, but always fails for 40 blocks or more, since the lattice reduction algorithm fails to detect vectors sufficiently small, and the space generated by the shortest vectors found is substantially random.
We measure the success reporting, for 100 LPC instances of each length, the dimension of the intersection of the smallest block with the 4, 8, 12 smallest vectors of the reduced basis. The time is in seconds needed to perform one attack.

<table>
<thead>
<tr>
<th>Dim</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5 s</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1.5 s</td>
</tr>
<tr>
<td>80</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3.5 s</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>5.0 s</td>
</tr>
<tr>
<td>120</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>6.5 s</td>
</tr>
<tr>
<td>140</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>8.0 s</td>
</tr>
</tbody>
</table>

Our interpretation of the data is that not only the cost of the attack increases exponentially, but it becomes less and less effective when the dimension increases, since the linear approximation becomes less and less valid. The assumption that breaking the key is equivalent to solving the hard non-linear optimization problem is substantially confirmed.

Our conclusion is that for more than 50 blocks, i.e. lattice dimension 200 or more, the private key is presumably secure with the current state of the art.

See [http://posso.dm.unipi.it/crypto](http://posso.dm.unipi.it/crypto) for more documentation and challenges.

14. Other lattice cryptosystems, security analysis of LPC

Lattice Polly Cracker is similar to members of the Polly Cracker family, but also shares several traits with other lattice cryptosystems.

The encryption method of LPC (adding a lattice element to a message or, dually, encrypting a message that is a lattice element adding some small “noise” to it) is the standard method in lattice and in error correcting code cryptosystems. Their message security lies in the difficulty of solving a closest vector problem (CVP) in lattices, or in linear error-correcting codes, where the Hamming distance is used instead of the Euclidean distance. See [Nguyen and Stern](#2001) for an overview.

The main cryptosystems of possibly practical interest in these families are GGH, (Goldreich et al., 1997), NTRU, (Hoffstein et al., 1998) and McEliece–Niederreiter (McEliece, 1978; Niederreiter, 1986). All of them can be described as cryptosystems in which a small vector is added to an element of the lattice, and the message is retrieved through some trapdoor information. The message is either the small vector or the lattice element, but since finding one determines the other the two are equivalent.

They differ in the mechanism of the trapdoor and in the definition of “small”:

- In GGH, the trapdoor is a quasi-orthogonal lattice basis and for a vector to be “small” means that the absolute value of every component is fixed (so it is small in $l_\infty$ norm);
- in NTRU, the trapdoor is a small vector in the lattice, and “smallness” is measured both in $l_\infty$ norm and in the Hamming weight;
- in McEliece, the trapdoor is a Goppa code structure, i.e. a key equation in a different basis, and “smallness” is defined in the Hamming weight.

In LPC, the trapdoor information is a fat block structure in a homomorphic, non-isometric lattice, and the “smallness” is measured in $l_\infty$ metric. Hence LPC is almost identical to GGH from the public point of view. So far as message security is concerned we may rely on the analysis of GGH.

From the public point of view the two look the same, the only visible difference is that with our choice of parameters the determinant of a LPC lattice is always a power of 2.

At closer inspection, however, one sees that a GGH public key is a lattice with a much larger determinant: for a GGH lattice of dimension $n$ the determinant is approximately $(4n\sqrt{n})^n$, while for LPC the determinant is approximately $19^n$, that is smaller for $n \geq 3$. Representing the public lattice in Hermite normal form, if $d$ is the determinant, and $n$ the dimension, requires $n \log_2(d)$ bits, so for example, for $n = 200$, a GGH public key is more than 3 times larger than an LPC public key (21.25 KB vs. 67 KB). With a smaller key, the encryption is simpler and the message expansion is more moderate.
Moreover, LPC can encode and decode deterministically a message with elements of range 9, (from 0 to 8, or equivalently from −4 to 4), while GGH can encode and decode with estimated failure rate $10^{-5}$ a vector of range 7 (from −3 to 3). This means that for vectors of the same length, LPC not only has a lattice of much smaller determinant, but can transmit vectors of larger range that are more secure.

The GGH vectors can only take values in {−3, 3}, a more reduced range is not considered secure enough. For comparison, the vectors allowed for LPC can be taken with values in {−4, −3, 3, 4}, with better security. This means that the estimates given in Goldreich et al. (1997) on the cost of the attacks to a GGH message can apply unchanged as a lower bound to the same attacks to LPC.

We remark that GGH has been broken with an attack of Nguyen (1999) that exploits the fact that a GGH vector is constant mod 6. The same attack does not apply to LPC, which does not have this flaw.

A comparison with NTRU is more difficult; the NTRU lattice has a cyclic structure and a polynomial representation that produces smaller public keys, moreover all the computations can be performed with machine integers, hence with considerable performance advantages that are difficult to match. We are currently studying the possibility of using machine integers in a large part of the LPC computations to improve the performance.

15. Conclusions and future work

We have shown a cryptosystem, based on lattices and using Gröbner bases for decryption, along the lines of the Polly Cracker cryptosystems. Encryption is similar to other lattice cryptosystems, and shares with them problems and remedies, as well as the feature of being immune (up to now) to quantum computing attacks.

Being exempt from the differential attack, messages appear to be secure. The basic scheme is not secure, but the addition of a change of coordinates seems to guarantee a reasonable level of security with a relatively short key. Computational experiences confirm this belief.

The construction of LPC has started to respond to a challenge of Barkee et al. (1994): 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.

Quoting from that paper:

“And now the challenge. (…)

“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 or belief on specific sparse schemes.”

While LPC has shown that Gröbner bases can be used in the construction of public key cryptosystems, that apparently cannot be broken trivially, and perform quite reasonably compared with other lattice cryptosystems, definitely it is not a cryptosystem applying the complexity of Gröbner bases. Indeed, the simple version of LPC has been cracked, just exploiting the possibility of computing easily a Lex Gröbner basis in some specific classes, i.e. a Hermite normal form.

Future work will be concentrated on tying the hardness of finding a “good” (i.e. manageable and sufficiently fat) Gröbner basis, or a change of coordinates and a good Gröbner basis, to hard lattice problems, and developing a signature protocol for LPC.

References
