Finding Small Roots of Bivariate Integer Polynomial Equations Revisited

Jean-Sébastien Coron

Gemplus Card International

Issy-les-Moulineaux, France

GEMPLUS

Solving polynomial equations

- Let $p(x)$ be a polynomial and N an RSA modulus. Solving $p(x) = 0 \mod N$: hard problem :
  - For $p(x) = x^2 - a$, equivalent to factoring N.
  - For $p(x) = x^e - a$, equivalent to inverting RSA.
- Let $f(x, y)$ be a polynomial with integer coefficients. Finding $(x_0, y_0) \in \mathbb{Z}^2$, $f(x_0, z_0) = 0$: hard problem.
  - Take $f(x, y) = N - x \cdot y$, equivalent to factoring N.
- Coppersmith showed (E96) that finding small roots is easy:
  - Univariate modular case: $p(x) = 0 \mod N$.
  - Bivariate integer case: $f(x, y) = 0$ over $\mathbb{Z}$.

Summary

- Two distinct algorithms by Coppersmith:
  - The univariate modular case: $p(x) = 0 \mod N$.
    - ✓ Simplified by Howgrave-Graham in 1997.
  - The bivariate integer case: $p(x, y) = 0$ over $\mathbb{Z}$.
    - ✓ Algorithm still difficult to understand.
- New algorithm to solve the bivariate integer case:
  - Simplification analogous to [HG97] for the univariate case.
  - Easy to understand and implement.
- Application :
  - Factoring $n = pq$ knowing the high-order bits of p.

Summary

- Summary of Coppersmith's algorithms:

Problem	Solution [Cop96]	Simplification
$f(x) = 0 \mod N$	Proven	[HG97]
$f(x, y) = 0 \mod N$	Heuristic	[HG97]
$f(x, y) = 0$ over $\mathbb{Z}$	Proven	this talk

- Finding a proof for $f(x, y) = 0 \mod N$ is still an open problem.

Solving $p(x) = 0 \mod N$

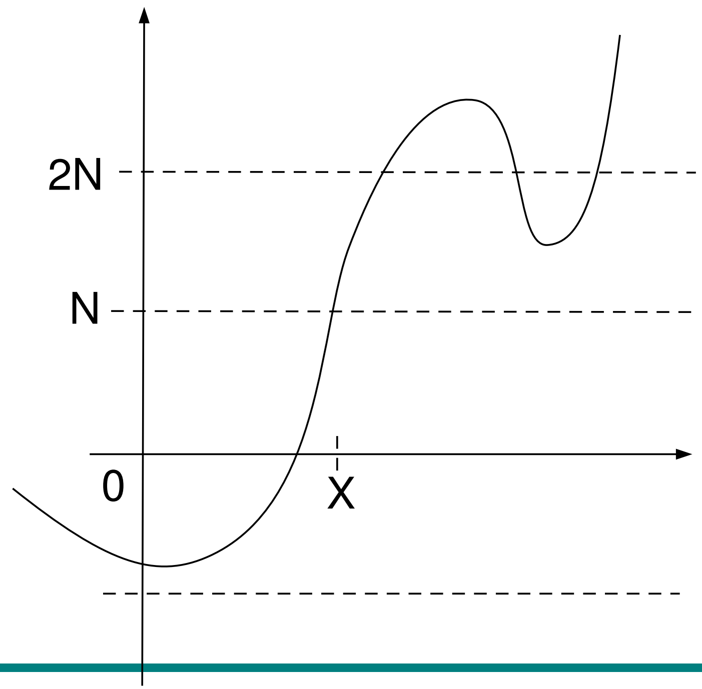
- Coppersmith's theorem:
  - Given an integer N and a polynomial $p(x)$ such that $\deg p = \delta$, one can find in polynomial time all integer x_0 such that $p(x_0) = 0 \mod N$ and $|x_0| \leq N^{1/\delta}$.
  - Based on LLL lattice reduction algorithm.
- Numerous applications in cryptography:
  - Cryptanalysis of plain RSA encryption when some part of the message is known :
    - ✓ If $c = (B + x_0)^3 \mod N$, let $p(x) = (B + x)^3 - c$ and recover x_0 if $x_0 < N^{1/3}$.

Solving $x^2 + ax + b = 0 \mod N$.

- Illustration with a polynomial of degree 2 :
  - Let $p(x) = x^2 + ax + b \mod N$.
  - We must find x_0 such that $p(x_0) = 0 \mod N$ and $|x_0| \leq X$.
- We generate a linear integer combination $h(x)$ of the polynomials :
  - $p(x)$, Nx and N.
  - Then $h(x_0) = 0 \mod N$.
- If the coefficients of $h(x)$ are small enough :
  - Then $|h(x_0)| < N$ and $h(x_0) = 0$ must hold over $\mathbb{Z}$.
  - This enables to recover x_0.

Howgrave-Graham lemma

- Given $h(x) = \sum h_i x^i$, let $\|h\|^2 = \sum h_i^2$.
- Howgrave-Graham lemma :
  - Let $h \in \mathbb{Z}[x]$ be a sum of at most ω monomials. If $h(x_0) = 0 \mod N$ with $|x_0| \leq X$ and $\|h(xX)\| < N/\sqrt{\omega}$, then $h(x_0) = 0$ holds over $\mathbb{Z}$.



Building the lattice

- The coefficients of $h(xX)$ must be small:
  - $h(xX)$ is a linear integer combination of the polynomials

$$\begin{array}{rcl} p(xX) & = & X^2 \cdot x^2 + aX \cdot x + b \\ q_1(xX) & = & NX \cdot x \\ q_2(xX) & = & N \end{array}$$

- We must find a small integer linear combination of the vectors:
  - $[X^2, aX, b]$, $[0, NX, 0]$ and $[0, 0, N]$
- Tool: LLL algorithm.

Building the lattice

- We must find a small linear integer combination $h(xX)$ of the polynomials $p(xX)$, xXN and N.
  - Let L be the corresponding lattice, with a basis of row vectors :

$$\begin{bmatrix} X^2 & aX & b \\ & NX & \\ & & N \end{bmatrix}$$

  - Using LLL, one can find a lattice vector b of norm : $\|b\| \leq 2(\det L)^{1/3} \leq 2N^{2/3}X$
- Then if $X < N^{1/3}/4$, then $\|h(xX)\| = \|b\| < N/2$
  - Howgrave-Graham lemma applies and $h(x_0) = 0$.

Solving $p(x) = 0 \mod N$

- The previous bound gives $|x_0| \leq N^{1/3}/4$.
  - But Coppersmith's bound gives $|x_0| \leq N^{1/2}$.
- One obtains Coppersmith's bound by using more multiples of $p(x)$ and working modulo N^ℓ:
  - Let $q_{ik}(x) = x^i \cdot N^{\ell-k} p^k(x) \mod N^\ell$
  - $p(x_0) = 0 \mod N \Rightarrow p^k(x_0) = 0 \mod N^k$ $\Rightarrow q_{ik}(x_0) = 0 \mod N^\ell$.
  - Then $h(x_0) = 0 \mod N^\ell$.
  - If the coefficients of $h(x)$ are small enough, then $h(x_0) = 0$ and one can recover x_0 using any standard root-finding algorithm.

The bivariate integer case

- Solving $p(x, y) = 0$ seems to be hard.
  - Integer factorization is a special case: take $p(x, y) = N - x \cdot y$.
- Coppersmith's showed (E96) that finding small roots is easy :
  - Let $p(x, y) \in \mathbb{Z}[x, y]$ has a maximum degree δ independently in x, y, and let $W = \max |p_{ij}| X^i Y^j$.
  - If $XY < W^{2/(3\delta)}$ one can find in polynomial time all integer pairs (x_0, y_0) such that $p(x_0, y_0) = 0$, $|x_0| \le X$ and $|y_0| \le Y$.
  - Based on the LLL algorithm.

The bivariate integer case

- But Coppersmith's algorithm is difficult to understand.
  - It uses non full-rank lattices, which makes determinant computation tedious.
- Our contribution : a new algorithm for solving $p(x, y) = 0$.
  - Simplification analogous to Howgrave-Graham for the univariate case.
  - Easy to understand and implement.
  - But asymptotically less efficient than Coppersmith's algorithm.

Approach: solving $p(x,y)=0$

- Let $q(x,y) = p_{00}^{-1} p(x,y) \mod n$ for some integer n.
- Find a small integer linear combination $h(x,y)$ of the polynomials $x^i y^j q(x,y)$ and $x^i y^j n$.
  - $q(x_0,y_0) = 0 \mod n \Rightarrow h(x_0,y_0) = 0 \mod n$.
- If the coefficients of $h(x,y)$ are sufficiently small :
  - 1) $h(x_0,y_0)=0$ using Howgrave-Graham lemma.
  - 2) $h(x,y)$ cannot be a multiple of $p(x,y)$.
- Then since $p(x,y)$ is irreducible :
  - $Q(x) = \text{Resultant}_y(h(x,y), p(x,y))$ is such that $Q \neq 0$ and $Q(x_0)=0$.
  - This gives x_0 and finally y_0 by solving $p(x_0,y)=0$.

An illustration

- Example with $p(x,y) = a + bx + cy + dxy$.
  - Assume that $a \neq 0$ and $d \neq 0$.
  - Find (x_0, y_0) such that $p(x_0, y_0) = 0$.
  - $W = \|p(xX, yY)\|_\infty = \max\{|a|, |b|X, |c|Y, |d|XY\}$, where $|x_0| \leq X$ and $|y_0| \leq Y$.
- Generate n such that $W \leq n < 2W$ and $\gcd(n, a) = 1$
  - Define $q_{00}(x,y) = a^{-1} p(x,y) \mod n$, $q_{00}(x,y) = 1 + b'x + c'y + d'xy \mod n$
  - Define $q_{10}(x,y) = nx$, $q_{01}(x,y) = ny$ and $q_{11}(x,y) = n$.
  - We have $q_{ij}(x_0, y_0) = 0 \mod n$.

Lattice of polynomials

- Let $h(x,y)$ be a linear combination of the $q_{ij}(x,y)$.
  - Then $h(x_0,y_0)=0 \mod n$

$$L = \begin{bmatrix} 1 & b'X & c'Y & d'XY \\ & nX & & \\ & & nY & \\ & & & nXY \end{bmatrix}$$

- Using LLL, one obtains $h(x,y)$ such that:
  - $\|h(xX,yY)\| \leq 2 \cdot (\det L)^{1/4} \leq 2n^{3/4}(XY)^{1/2}$
  - If $XY < n^{1/2}/16$, then $\|h(xX,yY)\| < n/2$.
  - HG lemma applies, and $h(x_0,y_0)=0$.

Solving $p(x, y) = 0$

- $\|h(xX, yY)\| < n/2 \leq \|p(xX, yY)\|_\infty \leq \|p(xX, yY)\|$
- If $h(x, y)$ was a multiple of $p(x, y)$.
  - Then $h(x, y) = \lambda \cdot p(x, y)$ with $\lambda \in \mathbb{Z}^*$
  - We would have $\|h(xX, yY)\| \geq \|p(xX, yY)\|$.
  - $\Rightarrow h(x, y)$ cannot be a multiple of $p(x, y)$.
- $p(x_0, y_0) = h(x_0, y_0) = 0$ and $p(x, y)$ is irreducible.
  - One can recover (x_0, y_0) by taking the resultant.
- This works if $XY < W^{1/2}/16 < W^{2/3}$.
  - By adding more multiples of $q(x, y)$ in the lattice, one recovers Coppersmith's bound.

Solving $p(x, y) = 0$

- Theorem :
  - Let $p(x, y) \in \mathbb{Z}[x, y]$ has a maximum degree δ independently in x, y, and let $W = \max |p_{ij}| X^i Y^j = \|p(xX, yY)\|_\infty$.
  - If $XY < W^{2/(3\delta)-\varepsilon}$ for some $\varepsilon > 0$, one can find in polynomial time all integer pairs (x_0, y_0) such that $p(x_0, y_0) = 0$, $|x_0| \leq X$ and $|y_0| \leq Y$.
- Asymptotically weaker than Coppersmith's theorem
  - which only assumes $XY < W^{2/(3\delta)}$.
  - Our algorithm is not polynomial for this weaker bound.

Solving $p(x, y) = 0$

- Let $p(x,y) = p_{00} + \sum p_{ij} x^i y^j$ of degree δ.
  - Assume first that $p_{00} \neq 0$ and $\gcd(p_{00}, XY) = 1$.
  - Let $k \geq 0$ be a parameter.
  - Generate $n = (XY)^k \cdot u$, where $u \simeq \|p(xX, yY)\|_\infty$
  - Let $q(x,y) = p_{00}^{-1} \cdot p(x,y) \mod n$
  - Then $q(x,y) = 1 + \sum_{(i,j)\neq(0,0)} a_{ij} x^i y^j$
- We form the polynomials $q_{ij}(x,y)$:
  - $q_{ij}(x,y) = x^i y^j X^{k-i} Y^{k-j} q(x,y)$, for $0 \leq i,j \leq k$.
  - $q_{ij}(x,y) = x^i y^j n$, for $(i,j) \in [0, \delta+k]^2 \setminus [0,k]^2$.
  - $q_{ij}(x_0, y_0) = 0 \mod n$ and $(XY)^k | q_{ij}(xX, yY)$.

Lattice of polynomials

- Lattice formed by the coefficient vectors of the polynomials $q_{ij}(xX, yY)$.
  - Full-rank lattice of dimension $\omega = (\delta + k + 1)^2$.
  - Illustration for $q(x,y) = 1 + a_{10}x + a_{01}y + a_{11}xy$ and $k = 1$.

	1	x	y	xy	x^2	x^2y	y^2	xy^2	x^2y^2
XYq	XY	$a_{10}X^2Y$	$a_{01}XY^2$	$a_{11}X^2Y^2$					
Yxq		XY		$a_{01}XY^2$	$a_{10}X^2Y$	$a_{11}X^2Y^2$			
Xyq			XY	$a_{10}X^2Y$			$a_{01}XY^2$	$a_{11}X^2Y^2$	
xyq				XY		$a_{10}X^2Y$		$a_{01}XY^2$	$a_{11}X^2Y^2$
x^2n					X^2n				
x^2yn						X^2Yn			
y^2n							Y^2n		
xy^2n								XY^2n	
x^2y^2n									X^2Y^2n

Size of $h(x, y)$

- We want the coefficients of $h(xX, yY)$ to be small enough, for the following two reasons :
- 1) If the coefficients of $h(xX, yY)$ are small enough :
  - Then $h(x_0, y_0) = 0$ holds not only modulo n, but also over $\mathbb{Z}$ (Howgrave-Graham's lemma).
  - The condition is $\|h(xX, yY)\| < \frac{n}{\sqrt{\omega}}$.
- 2) If the coefficients of $h(xX, yY)$ are small enough :
  - Then $h(x, y)$ cannot be a multiple of $p(x, y)$.
  - The condition is $\|h(xX, yY)\| < 2^{-\omega} \cdot (XY)^k \cdot W$
- From the definition of n, the first condition is satisfied when the second is satisfied.

Size of the factors of polynomials

- Mignotte's bound :
  - Let $f, g \in \mathbb{Z}[x]$ and $\deg f = k$.
  - If f divides g in $\mathbb{Z}[x]$, then $\|g\| \geq 2^{-k} \cdot \|f\|_\infty$.
- Extension to bivariate polynomials :
  - Let $a, b \in \mathbb{Z}[x, y]$ of degree less than d independently in x, y.
  - If a divides b in $\mathbb{Z}[x, y]$, then $\|b\| \geq 2^{-(d+1)^2} \cdot \|a\|_\infty$
- Proof: let $f(x) = a(x, x^{d+1})$ and $g(x) = b(x, x^{d+1})$.
  - Then f divides g and $\deg f \leq (d+1)^2$.
  - $\|f\|_\infty = \|a\|_\infty$ and $\|g\| = \|b\|$.
  - Apply Mignotte's bound to f and g.

Size of $h(x, y)$

- If $h(x, y)$ was a multiple of $p(x, y)$:
  - Then $h(xX, yY)$ is a multiple of $(XY)^k \cdot p(xX, yY)$.
  - From the previous lemma, this would give: $\|h(xX, yY)\| \geq 2^{-\omega} \cdot (XY)^k \cdot W$
- Conversely, if $\|h(xX, yY)\| < 2^{-\omega} \cdot (XY)^k \cdot W$:
  - $h(x, y)$ can not be a multiple of $p(x, y)$.
  - One recovers (x_0, y_0) by taking the resultant.
- Using LLL, we obtain a non-zero $h(x, y)$ such that :
  - $\|h(xX, yY)\| \leq 2^{(\omega-1)/4} \cdot \det(L)^{1/\omega}$
  - Make sure that : $2^{(\omega-1)/4} \cdot \det(L)^{1/\omega} < 2^{-\omega} \cdot (XY)^k \cdot W$.

The bound for XY

- We obtain the following condition on XY.
  - $XY < 2^{-\beta} W^{\alpha}$
  - where $\alpha = \frac{2}{3\delta} - \frac{2}{3\cdot(k+1)}$ and $\beta = \frac{4k^2}{\delta} + 13 \cdot \delta$.
- Taking $k = \lfloor 1/\varepsilon \rfloor$, we obtain :
  - $XY < W^{2/(3\delta)-\varepsilon} \cdot 2^{-4/(\delta\cdot\varepsilon^2)-13\delta}$
  - The algorithm is polynomial in $(\log W, \delta, 1/\varepsilon)$.
- If $XY < W^{2/(3\delta)-\varepsilon}$,
  - We exhaustively search the high-order $4/(\delta \cdot \varepsilon^2) + 13\delta$ bits of x_0.
  - For a fixed ε, the running time is polynomial in $(\log W, 2^{\delta})$.

Comparison with Coppersmith

- Difference in lattice dimension :
  - Coppersmith's algorithm works with a d-dimensional lattice over $\mathbb{Z}^\omega$, where $d = \delta^2 + 2(k+1)\delta$ and $\omega = (\delta + k + 1)^2$
  - We work with a full-rank lattice over $\mathbb{Z}^\omega$
- Our algorithm is asymptotically less efficient than Coppersmith's:
  - It is polynomial-time under the condition $XY < W^{2/(3\delta)-\varepsilon}$.
  - Instead of $XY < W^{2/(3\delta)}$.

Application to factoring

- Let $N = p \cdot q$ and assume that we know the half high-order bits of p.
- Write $p = p_0 + x_0$ and $q = q_0 + y_0$.
  - p_0 and q_0 are known.
  - $|x_0| < N^{1/4}$ and $|y_0| < N^{1/4}$
- Define the polynomial:

$$
\begin{aligned}
p(x, y) &= (p_0 + x)(q_0 + y) - N \\
&= (p_0 q_0 - N) + q_0 x + p_0 y + xy
\end{aligned}
$$

  - Then (x_0, y_0) is a small root of $p(x, y)$.
  - Using the previous theorem, one can recover (x_0, y_0) in polynomial time.

Practical experiments

- Using Shoup's NTL library, on a 733MHz PC under Linux :

N	bits of p given	lattice dimension	running time
512 bits	144 bits	25	35 sec
512 bits	141 bits	36	3 min
1024 bits	282 bits	36	20 min

- Using the simplification of Howgrave-Graham for the particular case of factoring with high-bits known :

N	bits of p given	lattice dimension	running time
1024 bits	282 bits	11	1 sec
1024 bits	266 bits	25	1 min
1536 bits	396 bits	33	19 min

  - This simplification does not apply to the general case of finding small roots of $p(x, y) = 0$.

Conclusion

- A new algorithm for finding small roots of $p(x, y) = 0$.
  - Simpler than Coppersmith's algorithm, but asymptotically less efficient.
  - The bivariate integer case is now as simple to analyze and implement as the univariate modular case.
- Experiments show that the algorithm works well in practice.
  - But for the particular case of integer factorization with high-bits known, the Howgrave-Graham simplification appears to be more efficient.