# Lattice Attacks in Cryptography: A Partial Overview

M. Jason Hinek
School of Computer Science, University of Waterloo
Waterloo, Ontario, N2L-3G1, Canada
mjhinek@alumni.uwaterloo.ca

October 22, 2004

**Abstract**

In this work, we give a partial overview of lattice attacks in cryptography. While different kinds of attacks are considered, the emphasis of this work is given to attacks that are based on Coppersmith's results for solving low degree multivariate modular equations and bivariate integer equations.

## Contents

# 1 Lattice Preliminaries

The information in this Chapter is mostly taken from the survey, *The Two Faces of Lattices in Cryptology*, by Nguyen & Stern [64]. For more information about the geometry of numbers see [16, 37, 84]. For more information about lattices and lattice basis reduction see [18, 36, 55, 56].

## 1.1 Definitions & Basic Facts

A ***lattice*** is a discrete (additive) subgroup of $\mathbb{R}^n$. Equivalently, given $m \leq n$ linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m \in \mathbb{R}^n$, the set

$$\mathcal{L} = \mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m) = \left\{ \sum_{i=1}^{m} \alpha_i \boldsymbol{b}_i \;\middle|\; \alpha_i \in \mathbb{Z} \right\}, \tag{1}$$

is a lattice. The $\boldsymbol{b}_i$ are called ***basis vectors*** of $\mathcal{L}$ and $\mathcal{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\}$ is called a ***lattice basis*** for $\mathcal{L}$. Thus, the lattice generated by a basis $\mathcal{B}$ is the set of all integer linear combinations of the basis vectors in $\mathcal{B}$.

The ***dimension*** (or ***rank***) of the a lattice, denoted $\dim(\mathcal{L})$, is equal to the number of vectors making up the basis. The dimension of a lattice is equal to the dimension of the vector subspace spanned by $\mathcal{B}$. A lattice is said to be ***full dimensional*** (or ***full rank***) when $\dim(\mathcal{L}) = n$.

It is often useful to represent a lattice $\mathcal{L}$ by a so-called ***basis matrix***. Given a basis $\mathcal{B}$, a basis matrix $\mathcal{M}$ for the lattice generated by $\mathcal{B}$ is simply the $m \times n$ matrix whose $\ell^{th}$ row is $\boldsymbol{b}_\ell$. The lattice can then be given by $\mathcal{L} = \{\boldsymbol{v} \mid \boldsymbol{v} = \boldsymbol{y}\mathcal{M}, \boldsymbol{y} \in \mathbb{Z}^n\}$. Similarly, a lattice can be generated by the columns of a $n \times m$ basis matrix whose $\ell^{th}$ column is $\boldsymbol{b}_\ell$ (i.e., $\mathcal{M}^T$).

Lattices with dimension $m \geq 2$ have infinitely many bases. Given a lattice $\mathcal{L}$ with basis matrix $\mathcal{M}$ and any $m \times m$ unimodular matrix $\mathcal{U}$ (i.e., $\mathcal{U}$ is an integral matrix with $\det(\mathcal{U}) = \pm 1$) then $\mathcal{M}' = \mathcal{U}\mathcal{M}$ is also a basis matrix for $\mathcal{L}$.

The ***volume*** (or ***determinant***) of a lattice, denoted by $\mathrm{vol}(\mathcal{L})$ is, by definition, the square root of the Gramian determinant $\det_{1 \leq i,j \leq m} \langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle$, which is independent of the particular choice of basis. This definition corresponds to the actual $m$-dimensional volume of the parallelepiped spanned by the $\boldsymbol{b}_i$'s and leads to the following result, called ***Hadamard's inequality***, which relates the volume of a lattice to any of its bases:

$$\mathrm{vol}(\mathcal{L}) \leq \prod_{i=1}^{m} |\boldsymbol{b}_i|, \tag{2}$$

where equality holds if and only if the basis vectors are mutually orthogonal. When a lattice is full dimensional its volume is given by $\mathrm{vol}(\mathcal{L}) = |\det(\mathcal{M})|$. In this case, it is more clear that the volume is independent of the choice of basis since any two basis matrices are related by a unimodular matrix.

If $\boldsymbol{b}_i \in \mathbb{Q}^n$ for all $1 \leq i \leq d$ (i.e., $\mathcal{L}$ is a subgroup of $\mathbb{Q}^n$) then the lattice $\mathcal{L}$ is called a ***rational lattice***. If $\boldsymbol{b}_i \in \mathbb{Z}^n$ for all $i \leq i \leq d$ (i.e., $\mathcal{L}$ is a subgroup of $\mathbb{Z}^n$) then the lattice $\mathcal{L}$ is called an ***integer lattice***. The volume of a full dimensional integer lattice is also equal to the index $[\mathbb{Z}^n : \mathcal{L}]$ of $\mathcal{L}$ in $\mathbb{Z}^n$.

Given any lattice of dimension $m$, for $1 \leq i \leq m$, the $i^{th}$ ***successive minima*** of $\mathcal{L}$, denoted by $\lambda_i(L)$ is defined to be radius of the smallest ball centred about the origin (of $\mathbb{R}^n$) such that there exists $i$ linearly independent

lattice vectors contained in this ball. That is,

$$\lambda_i(L) = \min_{\substack{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_i \in L \\ \text{lin. ind.}}} \max_{1\le j\le i} \|\boldsymbol{v}_j\|, \tag{3}$$

where $\|\cdot\|$ denotes the Euclidean norm. When the lattice is understood we will sometimes use $\lambda_i$ to denote the successive minima to simplify the notation. Other norms can be used in (3) to give different notions of successive minima. When the infinity-norm is used, we will denote the minima by $\lambda_i^\infty$. It can be shown that the first minimum of a lattice in the infinity and Euclidean norms satisfy $\lambda_1^\infty \le \mathrm{vol}(\mathcal{L})^{1/m}$ and $\lambda_1 \le \sqrt{m}\,\mathrm{vol}(\mathcal{L})^{1/m}$, respectively. More generally, Minkowski has shown the following result.

**Theorem 1.1.** *[Minkowski's Second Theorem] For any $m$-dimensional lattice $\mathcal{L}$ and all $r \le m$*

$$\lambda_1\lambda_2\cdots\lambda_r \le \sqrt{\gamma_m^r}\ \mathrm{vol}(\mathcal{L})^{r/m}, \tag{4}$$

*where $\gamma_m$ is Hermite's constant of dimension $m$.*

Hermite's constant of dimension $m$ is the supremum of $\lambda_1(\mathcal{L})^2/\,\mathrm{vol}(\mathcal{L})^{2/m}$ taken over all $m$ dimensional lattices $\mathcal{L}$. The first eight values of $\gamma_m$ are given in the following table (see Gruber & Lekkerkerker [37]):

| $m$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $(\gamma_m)^m$ | 1 | 4/3 | 2 | 4 | 8 | 64/3 | 64 | 256 |

These are the only known values of $\gamma_m$. The best known asymptotic bounds for Hermite's constant (see Milnor & Husemoller [63] for the lower bound and Conway & Sloane [19] for the upper bound) are given by

$$\frac{m}{2\pi\mathbf{e}} + \frac{\log(\pi m)}{2\pi\mathbf{e}} + o(1) \le \gamma_m \le \frac{1.744\,m}{2\pi\mathbf{e}}(1+o(1)). \tag{5}$$

The number of lattice points in a full dimensional lattice in *nice* sets of $\mathbb{R}^n$ is often estimated to be the volume of the set divided by the volume of the lattice up to a small additive error. This estimation, which dates back to Gauss, can be proven when the lattice dimension $m$ is fixed and the *nice* set is the ball centred at the origin with radius growing to infinity. Using this estimation, the first minimum of a lattice is often approximated by

$$\lambda_1 \approx \sqrt{\tfrac{m}{2\pi\mathbf{e}}}\,\mathrm{vol}(\mathcal{L})^{1/m}. \tag{6}$$

Of course, for some specific lattices, this approximation can be quite bad.

For any lattice $\mathcal{L}$, the ***dual lattice*** (also called the ***polar lattice***) of $\mathcal{L}$, denoted by $\mathcal{L}^\star$ is defined as:

$$\mathcal{L}^\star = \{\boldsymbol{x} \in span(\mathcal{L}) : \forall \boldsymbol{y} \in \mathcal{L}, \langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathbb{Z}\}, \tag{7}$$

where $span(\mathcal{L})$ is the linear span of the $\boldsymbol{b}_i$. Equivalently, one can define $\mathcal{L}^\star$ by the dual family $(\boldsymbol{b}_1^\star, \ldots, \boldsymbol{b}_m^\star)$. The ***dual family*** is the unique linearly independent family of $span(\mathcal{L})$ such that

$$\langle \boldsymbol{b}_i^\star, \boldsymbol{b}_j \rangle = \delta_{i,j}, \tag{8}$$

where $\delta_{i,j}$ is the Kronecker delta function (i.e., $\delta_{i,i} = 1$ and $\delta_{i,j\neq i} = 0$). The dual family is a basis for the dual lattice.

For any integer lattice $\mathcal{L} \subseteq \mathbb{Z}^n$, the ***orthogonal lattice*** $\mathcal{L}^\perp$ is defined to be the set of integral vectors in $\mathbb{Z}^n$ that are orthogonal to $\mathcal{L}$. That is,

$$\mathcal{L}^\perp = \{\boldsymbol{x} \in \mathbb{Z}^n : \forall \boldsymbol{y} \in \mathcal{L}, \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0\}. \tag{9}$$

The dimension of $\mathcal{L}$ and $\mathcal{L}^\perp$ are related by

$$\dim(\mathcal{L}) + \dim(\mathcal{L}^\perp) = n. \tag{10}$$

Further, the volume of $\mathcal{L}^\perp$ is equal to the volume of the lattice $\overline{\mathcal{L}}$ given by $\overline{\mathcal{L}} = \text{span}(\mathcal{L}) \cap \mathbb{Z}^n$ and so $\text{vol}(\mathcal{L}^\perp) \leq \text{vol}(\mathcal{L})$. Thus, when $\mathcal{L}$ has low dimension (relative to $n$) $\mathcal{L}^\perp$ has high dimension and so it is expected that the successive minima of $\mathcal{L}^\perp$ will be much smaller than the successive minima of $\mathcal{L}$.

## 1.2 Lattice Basis Reduction

For a given lattice $\mathcal{L}$ with dimension $m \geq 2$ some bases are *better* than others. Here, *better* depends on the actual application. Usually, we are interested in so-called reduced bases of a lattice. There are several notions of a ***reduced basis***, but in essence, a reduced basis is simply a basis made up of short vectors. Lattice basis reduction is the process in which a reduced basis is found from a given basis.

The first basis reduction algorithm, due to Gauss, is for 2-dimensional lattices. Let $\mathcal{L}$ be a lattice with dimension $m = 2$. Gauss's basis reduction algorithm transforms any basis of $\mathcal{L}$ into a basis $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ so that $\boldsymbol{b}_1$ is a shortest vector in the lattice and the component of $\boldsymbol{b}_2$ parallel to $\boldsymbol{b}_1$ has length at most $1/2$. The new basis $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is said to be ***Gaussian-reduced***. The Gaussian algorithm, which has runtime quadratic in the input size, is

given in Algorithm 1.1. This algorithm has been generalized by Nguyen & Stehlé [71] to lattices of any dimension. The generalized algorithm, called the greedy algorithm, is only optimal for lattices of dimension $m \leq 4$ though. By optimal, we mean that $\boldsymbol{b}_i = \Lambda_i(\mathcal{L})$ for each $i = 1, \ldots, m$. The greedy algorithm is also quadratic in the size of the input.

---

**Algorithm 1.1:** GAUSSIANREDUCTION$(\boldsymbol{b}_1, \boldsymbol{b}_2)$

---

**comment:** Computes a Gaussian-reduced basis of $(\boldsymbol{b}_1, \boldsymbol{b}_2)$

**repeat**
 **if** $\|\boldsymbol{b}_1\| > \|\boldsymbol{b}_2\|$
  **then** swap $\boldsymbol{b}_1, \boldsymbol{b}_2$
 $\mu \leftarrow \langle \boldsymbol{b}_1, \boldsymbol{b}_2 \rangle / \|\boldsymbol{b}_1\|^2$
 $\boldsymbol{b}_2 \leftarrow \boldsymbol{b}_2 - \lceil \mu \rfloor \, \boldsymbol{b}_1$ where $\lceil \alpha \rfloor = \lfloor \alpha + 0.5 \rfloor$
**until** $\|\boldsymbol{b}_1\| < \|\boldsymbol{b}_2\|$
**output** $(\boldsymbol{b}_1, \boldsymbol{b}_2)$

---

Before describing the next important class of reduced bases we first recall the ***Gram-Schmidt Orthogonalization*** process. Given $m$ linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m \in \mathbb{R}^n$, define the vectors $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_m^* \in \mathbb{R}^n$ by the recurrence

$$\begin{aligned} \boldsymbol{b}_1^* &= \boldsymbol{b}_1, \\ \boldsymbol{b}_i^* &= \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \boldsymbol{b}_j^* \quad \text{for} \quad 2 \leq i \leq m, \end{aligned}$$

where $\mu_{i,j} = (\boldsymbol{b}_i \cdot \boldsymbol{b}_j^*)/\|\boldsymbol{b}_j^*\|^2$ are called the Gram-Schmidt coefficients. We will call $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_m^*$ the Gram-Schmidt orthogonalization of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$. The Gram-Schmidt orthogonalization process creates an orthogonal basis of the span of $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$. Unfortunately, the $\mu_{i,j}$ are usually not integers so $\mathcal{L}(\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_m^*)$ is not, in general, the same lattice as $\mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$. Letting $\mathcal{L} = \mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$, the Gram-Schmidt orthogonalization of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ does, however, satisfy

$$\mathrm{vol}(\mathcal{L}) = \|\boldsymbol{b}_1^*\| \times \cdots \times \|\boldsymbol{b}_m^*\| \quad \text{and} \quad \lambda_1(\mathcal{L}) \geq \min_{1 \leq i \leq m} \{ \|\boldsymbol{b}_1^*\|, \ldots, \|\boldsymbol{b}_m^*\| \}.$$

Now, a basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ of a lattice $\mathcal{L}$ is said to be ***Lovász-reduced*** or ***LLL-reduced*** if

$$|\mu_{i,j}| \leq \frac{1}{2} \quad \text{for } 1 \leq j < i \leq n \tag{11}$$

and

$$|\boldsymbol{b}_i^* + \mu_{i,i-1}\boldsymbol{b}_{i-1}^*|^2 \geq \frac{3}{4}|\boldsymbol{b}_{i-1}^*|^2 \quad \text{for } 1 < i \leq n, \tag{12}$$

or equivalently

$$|\boldsymbol{b}_i^*|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right)|\boldsymbol{b}_{i-1}^*|^2 \quad \text{for } 1 < i \leq n, \tag{13}$$

where the $\boldsymbol{b}_i^*$ and $\mu_{i,j}$ are defined by the Gram-Schmidt orthogonalization process acting on the $\boldsymbol{b}_i$. Notice that the vectors $\boldsymbol{b}_i^* + \mu_{i,i-1}\boldsymbol{b}_{i-1}^*$ and $\boldsymbol{b}_{i-1}^*$ are the projections of $\boldsymbol{b}_i$ and $\boldsymbol{b}_{i-1}$, respectively, on the orthogonal complement of $\sum_{j=1}^{i-2}\mathbb{R}\boldsymbol{b}_j$. Some useful properties of LLL-reduced bases are given in the following theorem (see Cohen [18]).

**Theorem 1.2.** *Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ be an LLL-reduced basis of a rational lattice $\mathcal{L} \in \mathbb{Q}^n$ and $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_m^*$ be its Gram-Schmidt orthogonalization. Then*

*1.*

$$\operatorname{vol}(\mathcal{L}) \leq \prod_{i=1}^{m} |\boldsymbol{b}_i| \leq 2^{m(m-1)/4}\operatorname{vol}(\mathcal{L}), \tag{14}$$

*2.*

$$|\boldsymbol{b}_j| \leq 2^{(i-1)/2}|\boldsymbol{b}_i^*| \quad \textit{for } 1 \leq j \leq i \leq m, \tag{15}$$

*3.*

$$|\boldsymbol{b}_1| \leq 2^{(m-1)/4}\operatorname{vol}(\mathcal{L})^{1/m}, \tag{16}$$

*4. For every $\boldsymbol{x} \in \mathcal{L}$ with $\boldsymbol{x} \neq 0$ we have*

$$|\boldsymbol{b}_1| \leq 2^{(m-1)/2}|\boldsymbol{x}|. \tag{17}$$

*5. More generally, for any $t \leq m$ linearly independent vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t \in \mathcal{L}$ we have*

$$|\boldsymbol{b}_j| \leq 2^{(m-1)/2}\max(|\boldsymbol{x}_1|, \ldots, |\boldsymbol{x}_t|) \quad \textit{for } 1 \leq j \leq t. \tag{18}$$

The results of Theorem 1.2 lead directly to the following bounds on each of the LLL-reduced basis vectors.

**Corollary 1.1.** *Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ be an LLL-reduced basis of an integral lattice $\mathcal{L} \in \mathbb{Z}^n$. Then*

1. *Blömer & May [7]: For* $1 \leq \ell \leq m$

$$|\boldsymbol{b}_\ell| \leq 2^{\frac{m(m-1)}{4(m-\ell+1)}} \operatorname{vol}(\mathcal{L})^{\frac{1}{m-\ell+1}} \tag{19}$$

2. *Proos [77]: For* $\ell = 1$ *or* $1 < \ell \leq m$ *and* $|\boldsymbol{b}_1| \geq 2^{(\ell-2)/2}$

$$|\boldsymbol{b}_\ell| \leq 2^{\frac{m+l-2}{4}} \operatorname{vol}(\mathcal{L})^{\frac{1}{m-\ell+1}} \tag{20}$$

LLL-reduced bases are an important class of reduced bases because there exists a polynomial time algorithm to compute them. The first such algorithm, due to Lovász, is called the ***Lovász reduction algorithm***, or more commonly the ***LLL-algorithm*** [54] (see Appendix A.1 for the algorithm). For an $m$-dimensional lattice $\mathcal{L} \in \mathbb{Q}^n$ it has runtime $O(nm^5B^3)$ where $B$ is a bound on bitsize of the components of the basis vectors (i.e., $B \geq \max_i \log \|\boldsymbol{b}_i\|_\infty$).

Some other notions of reduced bases include Minkowski-reduced and (Korkine-Zolotareff) KZ-reduced. They are defined as follows. Let $\mathcal{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$ be a basis for the lattice $\mathcal{L}$. The basis $\mathcal{B}$ is said to be ***Minkowski-reduced*** if $\boldsymbol{b}_1$ is a shortest vector of $\mathcal{L}$ and for each $i = 2, \ldots, m$ the vector $\boldsymbol{b}_i$ is a shortest vector independent from $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$ such that $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_i$ can be extended to a basis of $\mathcal{L}$. The basis $\mathcal{B}$ is said to be ***KZ-reduced*** if $\boldsymbol{b}_1$ is a shortest vector of $\mathcal{L}$ and for each $i = 2, \ldots, m$ the vector $\boldsymbol{b}_i$ is a shortest vector of the lattice $\mathcal{L}_i$ which is the projection of $\mathcal{L}$ onto the subspace of $\mathbb{R}^n$ perpendicular to $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$. For 2-dimensional lattices, KZ-reduction and Gaussian-reduction are equivalent. Also, for each $i = 1, \ldots, m$ the $\boldsymbol{b}_i$ of a KZ-reduced basis satisfy

$$\sqrt{\frac{4}{i+3}}\, \lambda_i(\mathcal{L}) \leq \|\boldsymbol{b}_i\| \leq \sqrt{\frac{i+3}{4}}\, \lambda_i(\mathcal{L}).$$

Thus, each $\boldsymbol{b}_i$ is at most a factor of $\sqrt{n}$ away from $\lambda_i(\mathcal{L})$.

In [80], Schnorr introduced a hierarchy of polynomial-time lattice basis reduction algorithms. These algorithms, called ***blockwise Korkine-Zolotareff reductions*** or ***BKZ-reductions***, can compute a reduced basis ranging from LLL-reduced to KZ-reduced depending on a parameter called the blocksize. The algorithms are super-exponential in the blocksize (doing an exhaustive search on sets defined by the blocksize).

## 1.3 Algorithmic Problems

There are three main algorithmic problems dealing with lattice basis reduction: the shortest vector problem, the closest vector problem, and the smallest basis problem. We will be concerned with the first two problems.

In the rest of this section, we will assume that all lattices are rational lattices ($\mathcal{L} \subseteq \mathbb{Q}^n$) with dimension $m \leq n$, unless otherwise stated.

**Shortest Vector Problem** Given a basis for a lattice $\mathcal{L}$, the ***shortest vector problem (SVP)*** is to find $\boldsymbol{v} \in \mathcal{L}$ such that $\|\boldsymbol{v}\| = \lambda_1(\mathcal{L})$. The approximate shortest vector problem is to find a vector $\boldsymbol{v} \in \mathcal{L}$ such that $\|\boldsymbol{v}\| = f(m)\,\lambda_1(\mathcal{L})$ for some approximation factor $f(m)$.

It has been shown by Ajtai [1], that SVP is NP-hard under randomized reductions. Micciancio [61] has further shown that approximating SVP to within a factor less than $\sqrt{2}$ is also NP-hard under randomized reductions. The NP-hardness of SVP under deterministic reductions remains an open problem.

The best known theoretical result for exact SVP, by Ajtai, Kumar & Sivakumar [2], requires randomized $2^{O(m)}$-time.

There is no known algorithm to approximate SVP to within a polynomial factor of the dimension of the lattice. There are some polynomial time algorithms that can approximate it to within a slightly exponential factor though. From (17), we see that the LLL-algorithm approximates SVP to with a factor of $2^{(m-1)/2}$. This was improved to $2^{O(m(\log\log m)^2/\log m)}$ by Schnorr [80] and in randomized polynomial time further lowered by Ajtai, Kumar & Sivakumar [2] to $2^{O(m\log\log m/\log m)}$.

**Closest Vector Problem** Given a basis for a lattice $\mathcal{L} \subset \mathbb{Q}^n$ and a vector $\boldsymbol{u} \in \mathbb{R}^n$, the ***closest vector problem (CVP)*** is to find a vector $\boldsymbol{v} \in \mathcal{L}$ that minimizes $\|\boldsymbol{u} - \boldsymbol{v}\|$. Notice that the closest vector problem is simply a non-homogeneous version of the shortest vector problem.

It has been shown by van Emde Boas [89] that CVP is NP-hard. Arora, Babai, Stern & Sweedyk [3] have shown that approximating CVP to within a factor $2^{\log^{1-\epsilon} m}$ is also NP-hard.

The best known exact CVP algorithm, due to Kannan [50], has runtime $2^{O(m\log m)}$.

There is no known algorithm to approximate SVP to within a polynomial factor of the dimension of the lattice. There are some polynomial time algorithms that can approximate it to within a slightly exponential factor though. Using the LLL-algorithm, Babai [4] showed how to approximate CVP to within a factor of $2^{m/2}$. It has been shown by Kannan [51] that any algorithm approximating SVP to within a factor $f(m)$ (a non-decreasing function) can be used to approximate CVP to within a factor $m^{3/2}f(m)^2$. Combining this with the approximate SVP results from above,

we see that Schnorr's algorithm [80] can approximate CVP to within a factor $2^{O(m(\log\log m)^2/\log m)}$ and Ajtai, Kumar & Sivakumar's algorithm [2] to within a factor $2^{O(m\log\log m/\log m)}$ in randomized polynomial time.

A simple heuristic reduction from CVP to SVP, referred to as the ***embedding method*** [35, 65], seems to be the most used approximation technique though. Given a basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ for a lattice $\mathcal{L}$ and a vector $\boldsymbol{u} \notin \mathcal{L}$, the embedding method uses the $(m+1)$-dimensional lattice $\mathcal{L}'$ generated by the vectors $(\boldsymbol{b}_1, 0), \ldots, (\boldsymbol{b}_m, 0)$, and $(\boldsymbol{u}, \alpha)$ where $\alpha \in \mathbb{R}$ is typically chosen to be $\alpha = 1$ but may be different depending on $\mathcal{L}$. Let $\boldsymbol{v} \in \mathcal{L}$ be a closest vector to $\boldsymbol{u}$. When $\|\boldsymbol{u} - \boldsymbol{v}\| < \lambda_1(\mathcal{L})$, it is hoped that a shortest vector in $\mathcal{L}'$ is of the form $(\boldsymbol{u} - \boldsymbol{v}, \alpha)$, which discloses the desired closest vector $\boldsymbol{v}$. The approximate SVP algorithms from above are used to (attempt to) find a shortest vector in $\mathcal{L}'$.

**Smallest Basis Problem** Given a basis for a lattice $\mathcal{L}$, the ***smallest basis problem (SBP)*** is to find another basis for $\mathcal{L}$ which is *small* in some way. Two common notions of small include finding a basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ so that $\max_{1\leq i\leq m}\{\, \|\boldsymbol{b}_i\| \,\}$ is minimized, or so that the product $\|\boldsymbol{b}_1\| \cdots \|\boldsymbol{b}_m\|$ is minimized. This second notion of small corresponds to finding a basis that is close to orthogonal (from Hadamard's inequality).

There is no known algorithm to approximate SBP to within a polynomial factor in the dimension of the lattice, but the LLL-algorithm can be used to approximate it.

# 2 Linear Problems

## 2.1 Knapsacks

One of the first applications of lattice basis reduction to cryptography was attacking knapsack cryptosystems. In this section we will outline an attack of the original Merkle-Hellman knapsack cryptosystem as well as a method to solve almost any knapsack problem with low density.

We define an instance of the ***knapsack*** problem as follows: given a list of positive integers $a_1, \ldots, a_n$ and a positive integer $s$ find $x_1 \ldots, x_n \in \{0, 1\}$ such that $s = \sum_{i=1}^{n} x_i a_i$. We will refer to the integers $a_1, \ldots, a_n$ as the knapsack weights and to $s$ as the target. The problem might also be stated using vector notation. That is, given knapsack weights $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$ and a target $s \in \mathbb{N}$ find $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0, 1\}^n$ so that $s = \boldsymbol{x} \cdot \boldsymbol{a}$. The knapsack problem, as defined above, is actually an instance of the ***subset sum*** search problem which is NP-hard. There are some instances of

knapsack problems that are easy to solve though. For example, a knapsack problem with superincreasing[1] weights is very simple to solve since the $x_i$ can be computed from $x_n$ down to $x_1$ using the condition

$$x_i = 1 \quad \iff \quad s - \sum_{j=i+1}^{n} x_j\, a_j \geq a_i.$$

A simple algorithm to compute the $x_i$ can be constructed that is very similar to converting a decimal number into its binary representation.

### 2.1.1 Merkle-Hellman

In 1978, Merkle & Hellman [60] proposed one of first candidates for a public key cryptosystem. Their candidate, the ***Merkle-Hellman Knapsack Cryptosystem*** involved transforming an easy knapsack problem (with superincreasing weights) to a, hopefully, hard knapsack problem in such a way that the inverse transformation was hard to perform without knowledge of the private key. For pseudocode of the Merkle-Hellman knapsack cryptosystem see Appendix A.2.

Let $\boldsymbol{s} = (s_1, \ldots, s_n)$ be a superincreasing list of knapsack weights and let $M$ and $W$ be integers satisfying $M > \sum_{i=1}^{n} s_i$ and $\gcd(M, W) = 1$. Merkle & Hellman convert the knapsack weights $\boldsymbol{s}$ into another set of knapsack weights $\boldsymbol{a}$ by the transformation

$$a_i = W s_i \bmod M \qquad 1 \leq i \leq n.$$

The public key is the new set of weights $\boldsymbol{a}$. The private key consists of $M$, $W$, and the easy knapsack weights $\boldsymbol{s}$. To encrypt a message $\boldsymbol{x} \in \{0,1\}^n$ one simply computes $c = \boldsymbol{x} \cdot \boldsymbol{a}$. To recover $\boldsymbol{x}$ from $c$ and $\boldsymbol{a}$ one, seemingly, needs to solve the knapsack instance with weights $\boldsymbol{a}$ and target $c$. Knowledge of the private key, however, allows $\boldsymbol{x}$ to be recovered by solving the knapsack instance with weights $\boldsymbol{s}$ and target $c' = cW^{-1} \bmod M$. Letting $U = W^{-1} \bmod M$ we see that

$$c' = cU \equiv (\boldsymbol{x} \cdot \boldsymbol{a})U \equiv \boldsymbol{x} \cdot (\boldsymbol{a}U) \equiv \boldsymbol{x} \cdot \boldsymbol{s} \pmod{M},$$

and since $\sum_{i=1}^{n} x_i s_i < M$ we have $c' = \boldsymbol{x} \cdot \boldsymbol{s}$. So, we can recover $\boldsymbol{x}$ by solving the easy knapsack $\boldsymbol{s}$ with target $c'$.

[1] A list of numbers $a_1, \ldots, a_n$ is said to be ***superincreasing*** if each element in the list is strictly greater than the sum of all the elements preceding it. That is, $a_j > \sum_{i=1}^{j-1} a_i$ for all $j = 2, \ldots, n$.

We will outline an attack on the Merkle-Hellman knapsack that is based on lattice basis reduction and simultaneous Diophantine approximations. Starting with the equivalence relations $a_i \equiv s_i W \pmod{M}$ we define integers $k_i$ for $i = 1, \ldots, n$ so that

$$a_i U - k_i M = s_i. \tag{21}$$

The first step in the attack is noticing that the given knapsack problem (with weights $\boldsymbol{a}$ and target $c$) can be transformed into infinitely many different easy knapsack problems (with superincreasing weights $\boldsymbol{a'}$ and target $c'$). This was independently observed by Eier & Lagger [29] and Desmedt, Vanderwalle & Govaerts [27]. Their result can be summarized in the following lemma.

**Lemma 2.1.** *Let $M$, $U$, $\boldsymbol{a}$, $k_i$ for $i = 1, \ldots, n$ be defined as above. There exists an $\epsilon > 0$ such that if $\frac{U'}{M'}$ is rational with $\left|\frac{U'}{M'} - \frac{U}{M}\right| \leq \epsilon$, then the weights $\boldsymbol{s'} = (s'_1, \ldots, s'_n)$ where $s'_i = a_i U' - k_i M'$ for $i = 1, \ldots, n$ are superincreasing.*

Now, from equation (21) see that for $i = 1, \ldots, n$

$$\frac{U}{M} - \frac{k_i}{a_i} = \frac{s_i}{a_i M},$$

so each $\frac{k_i}{a_i}$ is a good approximation to $\frac{U}{M}$. In fact, if any $k_i$ was known then $\frac{k_i}{a_i}$ could be used to find a $U'$ and $M'$ satisfying the properties in Lemma 2.1. In order to recover the $k_i$ values, Shamir [81] noticed that

$$\left|\frac{a_i}{a_1} - \frac{k_i}{k_1}\right| \leq \frac{M}{2^{n-i}|a_1 k_1|} \quad \text{for} \quad i = 1, \ldots, n, \tag{22}$$

thus each $\frac{k_i}{k_1}$ is a good approximation to $\frac{a_i}{a_1}$. This leads to the simultaneous Diophantine approximation problem of finding integers $k_1, \ldots, k_n$ so that $\left(\frac{k_2}{k_1}, \ldots, \frac{k_n}{k_1}\right)$ is a good approximation of $\left(\frac{a_2}{a_1}, \ldots, \frac{a_n}{a_1}\right)$. From Lagarias [52], this approximation is said to be an ***unusually good simultaneous Diophantine approximation (UGSDA)*** if

$$\max_{2 \leq i \leq n} \left\{ \frac{a_i}{a_1} k_1 - k_i \right\} \leq a_i^{-1/n}.$$

The approximation is called *unusually good* because the likelihood of such an approximation existing is quite small over all choices of the $a_i$. For some $t \leq n$ we try to find a UGSDA of $\left(\frac{a_2}{a_1}, \ldots, \frac{a_t}{a_1}\right)$ from which we can extract

$k_1$. We turn out attention to finding such a UGSDA now. Consider the $t$-dimensional full rank lattice $\mathcal{L}$ generated by the rows of the basis matrix

$$\mathcal{M} = \begin{bmatrix} a_2 & a_3 & \cdots & a_t & \lfloor a_1^{1/t} \rfloor \\ -a_1 & & & & \\ & -a_1 & & & \\ & & \ddots & & \\ & & & -a_1 & \end{bmatrix}.$$

Every element of $\mathcal{L}$ is of the form

$$\boldsymbol{h} = \left( h_1 a_2 - h_2 a_1, \ldots, h_1 a_t - h_t a_1, h_1 \lfloor a_1^{1/t} \rfloor \right),$$

where $h_1, \ldots, h_t \in \mathbb{Z}$. If $\boldsymbol{h}$ is the smallest vector found by the LLL-algorithm, then

$$\|\boldsymbol{h}\| \leq 2^{(t-1)/4} d(L)^{1/t} = 2^{(t-1)/4} a_1^{(t-1)/t} \lfloor a_1^{1/t} \rfloor^{1/t} \leq 2^{(t-1)/4} a_1^{(t^2-t-1)/t^2}.$$

Looking at the first $t-1$ components of $\boldsymbol{h}$ we see that for $i = 1, \ldots, t-1$

$$\left| \frac{a_i}{a_1} h_1 - h_i \right| \leq 2^{(t-1)/4} a_1^{-(t+1)/t^2} \leq a_1^{-1/(t-1)},$$

where the last inequality holds whenever $t \geq 2\sqrt{\log_2 a_1}$. Thus, using a large enough $t$ the *LLL*-algorithm will find a vector that discloses a UGSDA of $\left( \frac{a_2}{a_1}, \ldots, \frac{a_t}{a_1} \right)$. Since the UGSDA's are so rare it is expected that $h_i = k_i$ for $i = 1, \ldots, n$. Using any of these found $k_i$ ($k_1$ say), values of $U'$ and $M'$ with $\gcd(U', M') = 1$ can be computed by approximating $k_1/a_1$. The knapsack problem with weights $\boldsymbol{a}$ and target $c$ can then be transformed into an easy knapsack problem with superincreasing weights $\boldsymbol{a'}$ and target $c'$ where $a'_i = a_i U' \bmod M'$ for $i = 1 \ldots, n$ and $c' = cU' \bmod M'$.

### 2.1.2 Low Density Knapsacks

In this section we consider another class of knapsack problems that can be solved efficiently; the low density knapsack problems. Let $\boldsymbol{a} = (a_1, \ldots, a_n)$ be a set of knapsack weights with maximum weight $A$ (i.e., $A = \max_{1 \leq i \leq n} a_i$). The ***density*** of the knapsack weights $a_1, \ldots, a_n$, denoted by $d$, is defined as

$$d = \frac{n}{\log_2 A}.$$

It has been independently shown by Brickell [14] and Lagarias & Odlyzko [53] that almost all low density knapsack problems with large $n$ can be solved provided there exists an SVP oracle. In particular, the attack by Lagarias & Odlyzko can be shown to work for almost all knapsack instances with density $d < 0.6463\ldots$. Their attack involves looking for a shortest vector in the lattice $\mathcal{L}_0$ generated by the rows of the matrix

$$\mathcal{M}_0 = \begin{bmatrix} 1 & & & Na_1 \\ & 1 & & Na_2 \\ & & \ddots & \vdots \\ & & 1 & Na_n \\ & & & Ns \end{bmatrix}, \tag{23}$$

where $N > \sqrt{n}$ is some integer. Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ be the solution to the knapsack problem with weights $\boldsymbol{a}$ and target $s$. Notice that the vector $\boldsymbol{c_0} = (x_1, \ldots, x_n, -1) \in \mathbb{Z}^{n+1}$ generates the lattice vector $\boldsymbol{x_0} = \boldsymbol{c_0}\mathcal{M}_0$ given by

$$\boldsymbol{x_0} = \boldsymbol{c_0}\mathcal{M}_0 = \Big(x_1, \ldots, x_n, N\left(x_1a_1 + \cdots + x_na_n - s\right)\Big) = (\boldsymbol{x}, 0).$$

Thus, finding $\boldsymbol{x_0}$ in $\mathcal{L}_0$ recovers $\boldsymbol{x}$. Assuming that at most $\frac{1}{2}$ of the $x_i$ are non-zero this vector is small ($\|\boldsymbol{x_0}\|^2 \leq \frac{1}{2}n$). In fact, for large $n$ the authors show that the probability that $\boldsymbol{x_0}$ is not the unique smallest vector in $\mathcal{L}_0$ is given by

$$\Pr \leq n\left(2n\sqrt{\tfrac{1}{2}n} + 1\right)\frac{2^{c_0 n}}{A},$$

where $c_0 = 1.54724\ldots$. This bound on the probability is obtained by counting lattice points in high dimensional spheres. Letting $A = 2^{cn}$ this probability tends to zero as $n$ gets large whenever $c > c_0$. Thus, almost all knapsack problems with large $n$ and density $d = n/\log_2 A = 1/c < 0.6463\ldots$ can be solved, provided there exists an SVP oracle. The original presentation can be found in Lagarias & Odlyzko [53]. For a simpler presentation of the probability bound see Frieze [31].

This bound on the density was independently increased to $d < 0.9408\ldots$ by Coster, LaMacchia, Odlyzko & Schnorr [26] and Joux & Stern [47]. These improvements are summarized in [25]. Coster, LaMacchia, Odlyzko & Schnorr

consider the lattice $\mathcal{L}_1$ generated by the rows of the basis matrix

$$\mathcal{M}_1 = \begin{bmatrix} 1 & & & & Na_1 \\ & 1 & & & Na_2 \\ & & \ddots & & \vdots \\ & & & 1 & Na_n \\ \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & Ns \end{bmatrix}. \tag{24}$$

In this case, the vector $\boldsymbol{c_1} = (x_1, \ldots, x_n, -1) \in \mathbb{Z}^{n+1}$ generates the lattice element $\boldsymbol{x_1} = \boldsymbol{c_1}\mathcal{M}_1$ given by

$$\boldsymbol{x_1} = (x_1 - \tfrac{1}{2}, \ldots, x_n - \tfrac{1}{2}, 0),$$

where each component of $\boldsymbol{x_1}$ is an element of $\{-\frac{1}{2}, 0, \frac{1}{2}\}$ since $x_i \in \{0, 1\}$ for each $1 \leq i \leq n$. In this case, regardless of the actual values of the $x_i$, the vector $\boldsymbol{x_1}$ must satisfy

$$\|\boldsymbol{x_1}\|^2 = \tfrac{1}{4}n.$$

Thus, $\boldsymbol{x_1}$ is a small vector in $L_1$. The authors go on to show that the probability that $\boldsymbol{x_1}$ is not the unique smallest vector in the lattice is bound above by

$$\Pr \leq n(4n\sqrt{n} + 1)\frac{2^{c_1 n}}{A},$$

where $c_1 = 1.0628\ldots$. When $A = 2^{cn}$ this bound tends to zero as $n$ gets large whenever $c > c_1$. Thus, almost all knapsack problems with large $n$ and density $d < 1/c = 0.9408\ldots$ can be solved with an SVP oracle.

Of course, there does not exist an SVP oracle for lattices with dimension $n > 4$ but in practise algorithms such as LLL will usually suffice.

## 2.2 Orthogonal Lattices

The notion of the orthogonal lattice can also be used to, heuristically, attack low density knapsack problems. We outline such an attack below.

Let $\boldsymbol{a} \in \mathbb{N}^n$ be a set of knapsack weights, $s$ be the target, and $\boldsymbol{x} \in \{0, 1\}^n$ be the solution (i.e., $s = \boldsymbol{x} \cdot \boldsymbol{a}$). Consider the vector of knapsack weights $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{Z}^n$. Let $\mathcal{L} = \boldsymbol{a}^\perp$ be the orthogonal lattice of the lattice generated by $\boldsymbol{a}$. That is,

$$\mathcal{L} = \{\boldsymbol{z} \mid \boldsymbol{z} \cdot \boldsymbol{a} = 0\,,\, \boldsymbol{z} \in \mathbb{Z}^n\}$$

is the lattice of all solutions to the homogeneous equation $\boldsymbol{z} \cdot \boldsymbol{a} = 0$. Let $\boldsymbol{y} \in \mathbb{Z}^n$ be any vector satisfying $s = \boldsymbol{y} \cdot \boldsymbol{a}$. The vector $\boldsymbol{\ell} = (y_1 - x_1, \ldots, y_n - x_n)$

is then an element of $\mathcal{L}$. Since $x_i \in \{0,1\}$ for each $i = 1, \ldots, n$, the vector

$$\widehat{\boldsymbol{y}} = (y_1 - \tfrac{1}{2}, \ldots, y_n - \tfrac{1}{2}) \notin \mathcal{L}$$

is close to $\boldsymbol{\ell}$. In fact, $\|\boldsymbol{\ell} - \widehat{\boldsymbol{y}}\| = \sqrt{n/4}$ and $\boldsymbol{\ell}$ is the closest vector in $\mathcal{L}$ to $\widehat{\boldsymbol{y}}$. Thus, solving the CVP with $\mathcal{L}$ and $\widehat{\boldsymbol{y}}$ will disclose $\boldsymbol{x}$.

Now, if $\|\boldsymbol{\ell} - \widehat{\boldsymbol{y}}\| = \sqrt{n/4} < 2^{-(n-1)/2-1}\lambda_1$, where $\lambda_1$ is the first minimum of $\mathcal{L}$, then Babai's CVP approximation algorithm will find a vector $\boldsymbol{w} \in \mathcal{L}$ such that $\|\boldsymbol{w} - \widehat{\boldsymbol{y}}\| < 2^{n/2}\|\boldsymbol{\ell} - \widehat{\boldsymbol{y}}\| < \lambda_1/2$ and so $\|\boldsymbol{w} - \boldsymbol{\ell}\| < \lambda_1$. Since $\boldsymbol{w}, \boldsymbol{\ell} \in \mathcal{L}$ this implies that $\boldsymbol{w} = \boldsymbol{\ell}$ and knowing $\boldsymbol{\ell}$ and $\boldsymbol{y}$ reveals $\boldsymbol{x}$. Let $d$ be the density of the knapsack weights $\boldsymbol{a}$. The volume of the lattice will satisfy

$$d(\mathcal{L}) = \left(\frac{\sum_{i=1}^{n} a_i^2}{\gcd(a_1, \ldots, a_n)}\right)^{1/2} \approx 2^{n/d}\sqrt{n},$$

so using Gauss' heuristic, it is expected that $\lambda_1 \approx 2^{1/d}\sqrt{\frac{n}{2\boldsymbol{\pi}\mathbf{e}}}$. Thus, the method should work provided that

$$\sqrt{\frac{n}{4}} < 2^{-(n-1)/2-1} 2^{1/d} \sqrt{\frac{n}{2\boldsymbol{\pi}\mathbf{e}}}.$$

This is roughly equivalent to $d \leq 2/n$. It is expected, then, that knapsacks with density less than $2/n$ should be solvable with this method. Using the embedding method, described in Section 1.3, to reduce CVP to SVP we expect that we can solve this problem whenever the distance from $\widehat{\boldsymbol{y}}$ to the lattice is smaller than the first minimum. Thus, heuristically, we can solve the problem when

$$\sqrt{\frac{n}{4}} < 2^{1/d} \sqrt{\frac{n}{2\boldsymbol{\pi}\mathbf{e}}},$$

which is equivalent to

$$d \leq \frac{2}{\log_2(\boldsymbol{\pi}\mathbf{e}/2)} \approx 0.955\ldots.$$

Notice that this heuristic bound is quite close to the provable bound from the previous section. Also, finding a closest vector to $\boldsymbol{y}$ in $\mathcal{L}$ instead of $\widehat{\boldsymbol{y}}$ leads to the heuristic bound of $d \leq 0.64637\ldots$, matching the results of the previous section as well.

The notion of the orthogonal lattice was first introduced by Nguyen & Stern [72] to attack the Qu-Vanstone cryptosystem [78]. It has since been used to attack various cryptographic systems such as in [73, 74] and in particular [75] to solve the ***hidden subset sum problem***.

## 2.3 Hidden Number Problem

The hidden number problem, introduced by Boneh & Venkatesan in 1996 [13], was used in the first positive application of lattice basis reduction in cryptography (showing the security of Diffie-Hellman bits). It has since been used, in a negative sense, to attack certain signature (and identification) schemes when some partial information about the secret values used in the signature generation is known.

We first need to define the notion of the most significant bits of a number $x \in \mathbb{Z}_p$ (as opposed to the most significant bits in a binary representation). For integers $s$ and $m \geq 1$ let the remainder of $s$ divided by $m$ be denoted by $\lfloor s \rfloor_m$. This is simply $s \bmod m$ when considering the positive representation. Given a prime $p$ and $\ell > 0$, let $\mathrm{MSB}_{\ell,p}(x)$ be any integer $u$ that satisfies

$$\left| \lfloor x \rfloor_p - u \right| \leq \frac{p}{2^{\ell+1}}.$$

When $\ell$ is an integer, $\mathrm{MSB}_{\ell,p}(x)$ corresponds to the $\ell$ most significant bits of $x$ in $\mathbb{Z}_p$. This definition is somewhat more flexible though, as $\ell$ does not need to be an integer.

An instance of the ***hidden number problem (HNP)*** is the problem of recovering $\alpha \in \mathbb{Z}_p$ such that for $k$ elements $t_1, \ldots, t_k \in \mathbb{Z}_p^*$, chosen independently and uniformly at random, we are given $k$ pairs

$$(t_i, \mathrm{MSB}_{\ell,p}(\alpha t_i)), \quad i = 1, \ldots, k,$$

for some $\ell > 0$.

In order to solve the HNP, consider the $(k+1)$-dimensional full rank lattice $\mathcal{L}(p, \ell, t_1, \ldots, t_k)$ spanned by the rows of the basis matrix

$$\mathcal{M} = \begin{bmatrix} p & & & & & \\ & p & & & & \\ & & \ddots & & & \\ & & & p & & \\ t_1 & t_2 & \cdots & t_k & 1/2^{\ell+1} \end{bmatrix}.$$

Letting $a_i = \mathrm{MSB}_{\ell,p}(\alpha t_i)$ for $i = 1 \ldots, k$ we see that the vector $\boldsymbol{u} = (a_1, \ldots, a_k, 0)$ is very close to the vector

$$\boldsymbol{w} = \left( \lfloor \alpha t_1 \rfloor_p, \ldots, \lfloor \alpha t_k \rfloor_p, \tfrac{\alpha}{2^{\ell+1}} \right) \in \mathcal{L}(p, \ell, t_1, \ldots, t_k).$$

In fact, the distance is of the order $p2^{-\ell}$. If $\boldsymbol{w}$ is the only lattice vector close to $\boldsymbol{u}$ it can be recovered using CVP approximation algorithms.

Using current approximate CVP algorithms, one can find a vector $\boldsymbol{v} \in \mathcal{L}$ such that

$$\|\boldsymbol{v} - \boldsymbol{u}\| \leq \min_{\boldsymbol{z} \in \mathcal{L}} \|\boldsymbol{z} - \boldsymbol{u}\| \ \exp\left(O\left(\frac{k \log^2 \log k}{\log k}\right)\right).$$

Assuming $\min_{\boldsymbol{z} \in \mathcal{L}} \|\boldsymbol{z} - \boldsymbol{u}\| \leq p2^{-\ell}$ we wish to show that there are a negligible number of $k$-tuples $(t_i, \ldots, t_k) \in \mathbb{Z}_p^k$ for which the lattice $\mathcal{L}(p, \ell, t_1, \ldots, t_k)$ has a vector $\boldsymbol{v} \neq \boldsymbol{w}$ satisfying

$$\|\boldsymbol{v} - \boldsymbol{w}\| \leq p2^{-\ell} \exp\left(O\left(\frac{k \log^2 \log k}{\log k}\right)\right).$$

That is, we want to show that in almost all instances the vector $\boldsymbol{w}$ is the only vector in $\mathcal{L}$ that is close to $\boldsymbol{u}$. Looking at $\mathcal{M}$ we see that $\boldsymbol{v}$ must be of the form

$$\boldsymbol{v} = \left(\beta t_1 - \lambda_1 p, \ldots, \beta t_k - \lambda_k p, \beta / 2^{\ell+1}\right),$$

for some integers $\lambda_1, \ldots, \lambda_k$ and $\beta$. In order for $\boldsymbol{v}$ to satisfy the above inequality the first $k$ components of $\boldsymbol{v} - \boldsymbol{w}$ must satisfy

$$(\alpha - \beta) t_i \equiv y_i \pmod{p}, \tag{25}$$

for some $y_i \in [-h, h]$ where $h$ is given by

$$h = p2^{-\ell} \exp\left(O\left(\frac{k \log^2 \log k}{\log k}\right)\right).$$

Now, for any $\gamma \neq 0$

$$\Pr_{y \in \mathbb{Z}_p}\left(\gamma t \equiv y \pmod{p} \mid y \in [-h, h]\right) \leq \frac{2h+1}{p},$$

so the probability $\mathcal{P}$ that each of the first $k$ components of $\boldsymbol{v} - \boldsymbol{w}$ satisfy (25) for at least one $\beta \neq \alpha$ is bounded above by

$$\mathcal{P} \leq (p-1)\left(\frac{2h+1}{p}\right)^k \leq p\left(\frac{3h}{p}\right)^k = p2^{-\ell k} \exp\left(O\left(\frac{k^2 \log^2 \log k}{\log k}\right)\right).$$

Choosing the parameters $\ell$ and $k$ so that

$$\ell = \left\lceil C \frac{\log^{1/2} p \log\log\log p}{\log\log p} \right\rceil \quad \text{and} \quad k = 2\left\lceil \frac{\log p}{\ell} \right\rceil,$$

for some constant $C > 0$ we see that the probability that $\boldsymbol{w}$ is the only lattice vector close to $\boldsymbol{u}$ is exponentially close to 1. Therefore, the approximate CVP algorithms will almost always return the desired $\boldsymbol{w}$. Of course, once $\boldsymbol{w}$ is known the value of $\alpha$ is revealed since the last component of $\boldsymbol{w}$ is equal to $\alpha/2^{\ell+1}$.

In some practical applications the condition that the $t_i$ be chosen independently and uniformly at random from $\mathbb{Z}_p$ is too restrictive. To accommodate some of these instances, we consider an extended version of the hidden number problem in some finite field $\mathbb{F}_p$ (see Shparlinski [82] for example). An instance of the ***extended hidden number problem (EHNP)*** is the problem of recovering $\alpha \in \mathbb{F}_p$ such that for $k$ elements $t_1, \ldots, t_k \in \mathcal{T}$, chosen independently and uniformly at random from some given subset $\mathcal{T} \subseteq \mathbb{F}_p$, we are given $k$ pairs

$$(t_i, \mathrm{MSB}_{\ell,p}(\alpha t_i)), \quad i = 1, \ldots, k,$$

for some $\ell > 0$. In order to prove that these problems can be solved (using the same method as for the HNP) some results on the uniformity of the distribution of $\mathcal{T}$ must be known.

When $\mathcal{T} \neq \mathbb{F}_p$ the uniformity of $\mathcal{T}$ is obtained using discrepancy theory. We sketch the main details below. For more information see Shparlinski [82, 83]. The ***discrepancy*** of an $n$-element sequence $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ where each $\gamma_i \in [0, 1]$ is defined as

$$\mathcal{D}(\Gamma) = \sup_{J \subseteq [0,1]} \left| \frac{A(J, n)}{n} - |J| \right|,$$

where the supremum is over all subintervals $J$ of $[0, 1]$, $|J|$ is the length of $J$, and $A(J, n)$ is the number of elements in the intersection $\Gamma \cap J$. Now, a finite sequence of integers $\mathcal{T}$ is ***$\Delta$-homogeneously distributed modulo $p$*** if for any integer $a$ with $\gcd(a, p) = 1$ the discrepancy of the sequence $\{\lfloor at \rfloor_p / p\}_{t \in \mathcal{T}}$ is at most $\Delta$. In this case, for any $\gamma \neq 0$ we have

$$\Pr_{y \in \mathbb{Z}_p} \left( \gamma t \equiv y \pmod{p} \mid y \in [-h, h] \right) \leq \frac{2h + 1}{p} + \Delta.$$

Choosing the parameters $\ell$ and $k$ so that

$$\ell = \lceil \log^{1/2} p \rceil + \lceil \log \log p \rceil \quad \text{and} \quad k = 2 \lceil \log^{1/2} p \rceil,$$

if $\mathcal{T}$ is $2^{-\log^{1/2} p}$-homogeneously distributed modulo $p$ then there exists an algorithm that can recover $\alpha$ with probability greater than $1 - 2^{-\log^{1/2} p \log \log p}$.

In general, it turns out that $\mathcal{T}$ is $\Delta$-homogeneously distributed modulo $p$ with $\Delta$ given by

$$\Delta = O\left(\frac{\log p}{|\mathcal{T}|}\max_{c\in\mathbb{Z}_p^*}\left|\sum_{t\in\mathcal{T}}\exp\left(\frac{2\pi ict}{p}\right)\right|\right).$$

Thus, the theory of exponential sums plays an important role in the EHNP.

Another variation of the hidden number problem involves working in a ring rather than a field (see Proos [77] for example). Let $N$ be a composite number. An instance of the ***generalized hidden number problem (GHNP)*** is the problem of recovering $\alpha \in \mathbb{Z}_N$ such that for $k$ elements $t_1, \ldots, t_k \in \mathbb{Z}_N$, chosen independently and uniformly at random, we are given $k$ pairs

$$(t_i, \mathrm{MSB}_{\ell,N}(\alpha t_i)), \quad i = 1, \ldots, k,$$

for some $\ell > 0$. Using the same methods as for the HNP, the GNHP can be solved in certain circumstances. Unlike the HNP and EHNP though, these methods are only heuristic. There are no rigorous proofs to show that the methods will recover $\alpha$ in almost all cases as with the HNP and EHNP. This being said, in certain instances the GHNP can be solved in practise.

The above results for the HNP and its variants also hold when a fraction of the least significant bits of $(\alpha t_i \bmod p)$ are known instead of the most significant bits. Results for partial knowledge of the interior of $(\alpha t_i \bmod p)$ can also be derived. In this case, if the known information is contiguous it can be shown that the HNP can be solved using twice as many bits as needed if the most (or least) significant bits are known.

**Applications** The main application of the HNP and its variants is attacking signature (and identification) schemes that use a hidden random integer, often called a ***nonce***, during the signature generation. The attacks can be mounted successfully when some number of bits of these nonces are known for some some number of signatures.

To illustrate the basic application of the HNP we consider the digital signature algorithm (DSA). First we recall the DSA signature generation algorithm (see Appendix A.3 for more detail). Let $p$ and $q \geq 3$ be prime numbers such that $q$ divides $p - 1$. Let $\mathcal{M}$ be the message space and $h : \mathcal{M} \to \mathbb{Z}_q$ be an arbitrary hash function. The signer selects a secret key $\alpha \in \mathbb{Z}_q^*$ and computes the public key $(p, q, g, g^\alpha)$ where $g \in \mathbb{Z}_p$ has order $q$. To sign a message $m \in \mathcal{M}$ the signer chooses a random nonce $k \in \mathbb{Z}_q^*$ and computes

$$r(k) = (g^k \bmod p) \bmod q \quad \text{and} \quad s(k, m) = k^{-1}(h(m) + \alpha r(k)) \bmod q.$$

The pair $(r(k), s(k,m))$ is the DSA signature of the message $m$ with nonce $k$.

We assume that the $\ell$ least significant bits of a nonce $k \in \mathbb{F}_q^*$ is known. That is, we know $k_0$ such that $0 \leq k_0 \leq 2^\ell - 1$ and $k - k_0 = 2^\ell b$ for some integer $0 \leq b \leq q/2^\ell$. Notice that by the definition of $s(k,m)$ we have the following

$$\alpha r(k) \equiv s(k,m)k - h(m) \pmod{q},$$

which can be rewritten for $s(k,m) \neq 0$ as

$$\alpha r(k) 2^{-\ell} s(k,m)^{-1} \equiv \left(k_0 - s(k,m)^{-1} h(m)\right) 2^{-\ell} + b \pmod{q}.$$

Let $t(k,m)$ and $u(k,m)$ be defined by

$$\begin{aligned} t(k,m) &= \left\lfloor 2^{-\ell} r(k) s(k,m)^{-1} \right\rfloor_q, \\ u(k,m) &= \left\lfloor 2^{-\ell} \left(k_0 - s(k,m)^{-1} h(m)\right) \right\rfloor_q. \end{aligned}$$

Notice that these values satisfy

$$0 \leq \lfloor \alpha t(k,m) - u(k,m) \rfloor_q \leq \frac{q}{2^\ell},$$

which leads to the following relation

$$\left| \alpha t(k,m) - u(k,m) - \frac{q}{2^{\ell+1}} \right| \leq \frac{q}{2^{\ell+1}}.$$

Therefore, the most significant bits of $(\alpha t(k,m) \bmod q)$ are known. Computing this quantity for some number of signatures (generated with the same $\alpha$ of course) results in an instance of the EHNP since the distribution of the multiplier $t(k,m)$ for random $m$ and $k$ is not uniform. With a reasonable assumption on the hash function $h(x)$, Nguyen & Shparlinski [69] show that the private key $\alpha$ can be recovered provided

$$\ell = \left\lceil \omega \left( \frac{\log q \log\log\log q}{\log\log q} \right)^{1/2} \right\rceil,$$

given

$$O\left( \left( \frac{\log q \log\log q}{\log\log\log q} \right)^{1/2} \right),$$

signatures, where $\omega > 0$ is some constant. Their analysis involved using the Weil bound for exponential sums with rational functions to handle the non-uniformity of the multipliers.

There are many instances where the HNP or one of its generalizations can be used to mount a successful attack against various cryptographic protocols. We give a brief list of some of these attacks below.

**Signature Schemes** In each of these cases, some partial knowledge of the nonces used in signature generation is needed from some number of signatures in order to construct an instance of the HNP or one of its variants.

• ***DSA*** – The first lattice based attacks on DSA with partially known nonces were by Howgrave-Graham & Smart [46] and Nguyen [66]. It was Nguyen who first related the problem to a variant of the HNP. These first attacks were heuristic in nature as no provable results were obtained. A provable attack, using exponential sums to analyze the distribution of signatures, was presented by Nguyen & Shparlinski [69].

•***ECDSA*** – Following their work with DSA, Nguyen & Shparlinski [70] present a provable attack on ECDSA. The proof of the attack involves estimating exponential sums that differ from those in the DSA case and results in slightly weaker results.

•***Nyberg-Rueppel*** – Nguyen & Shparlinski [68] show that the Nyberg-Rueppel variants of DSA are provably insecure with partially known nonces.

•***ESIGN*** – Howgrave-Graham [45] was the first to observe that breaking the ESIGN signature scheme with partial knowledge of the nonces for some number of signatures could be reduced to a certain GHNP. It was claimed that the results of Nguyen & Shparlinski for DSA carried over to the ESIGN case, but this was incorrect as some important conditions are different (such as the fact that the modulus is no longer prime). Later, Proos [77] presented a heuristic attack with experimental evidence to estimate the practicality of it.

**Identification Schemes** •***NBD*** – Proos [77] showed how the problem of recovering an NBD secret key with partial knowledge of the nonces can be reduced to a GHNP. A heuristic attack was given along with experimental evidence to estimate the practicality of it.

**Key Agreement protocols** •***Arazi*** – Brown & Menezes [15] present an attack on Arazi's key agreement protocol, a protocol that uses both the Diffie-Hellman key agreement protocol and DSA. The attack can be used to obtain a users private DSA key. This attack is unique in that it generates the partial knowledge (DSA nonces) needed to solve the HNP itself.

## 2.4 GnuPG

Recently, Nguyen [67] has shown a vulnerability in the freely distributed email security package GNU Privacy Guard v1.2.3, referred to as GPG hereafter (see [34] for more information about GPG). The vulnerability involves GPG's version of ElGamal signatures. In fact, given only one valid signature/message pair, the secret signing key can be recovered almost immediately. We first give an outline of GPG's ElGamal signature scheme and then show Nguyen's attack (see Appendix A.4 for more detail of the signature scheme).

Let $p$ be a prime such that the factorization of $p-1$ is known and all prime factors of $(p-1)/2$ have bit-length greater than $q_{bit}$, which is a function of $p$. The values of $q_{bit}$ that GPG uses for various sizes of $p$, as given by the so-called Wiener table, is partially shown in Table 1. Notice

| Bit-length of $p$ | 512 | 768 | 1024 | 1280 | 1536 | 2048 | 2560 | 3072 |
|---|---|---|---|---|---|---|---|---|
| $q_{bit}$ | 119 | 145 | 165 | 183 | 198 | 225 | 249 | 269 |
| $q_{bit}/\log p$ | 0.23 | 0.19 | 0.16 | 0.14 | 0.13 | 0.11 | 0.10 | 0.09 |

Table 1: Partial Wiener table for ElGamal primes.

that $q_{bit} < \frac{1}{4}\log_2 p$ for each choice of prime $p$ (which also holds for all values not shown in the table).

Let $g$ be a generator of $\mathbb{Z}_p^*$. The secret signing key $x$ is chosen as a pseudo-random number with bit-length $\frac{3}{2}q_{bit}$. As will be seen, this condition on $x$ is one of the reasons why the system is vulnerable. In the standard ElGamal key generation algorithm the secret key is chosen as a random number in $\mathbb{Z}_p^*$.

The secret key is simply $x$ and the public key is given by $(p, g, y)$, where $y = g^x \pmod p$. The signature of a message $m \in \mathbb{Z}_p$ is the pair $(a, b)$ where

$$a = g^k \bmod p \quad \text{and} \quad b = (m - ax)k^{-1} \bmod (p-1),$$

and $k$ is a number that is relatively prime to $p-1$. In the standard ElGamal signature generation algorithm, $k$ would be chosen to be a cryptographically secure random number modulo $p-1$. In GPG, the random number $k$ is first chosen with $\frac{3}{2}q_{bit}$ pseudo-random bits (so $k$ might have less than $\frac{3}{2}q_{bit}$ bits) and incremented until $\gcd(k, p-1) = 1$. Thus, $k$ will approximately be a $\frac{3}{2}q_{bit}$-bit number. This is the second reason why the system is so vulnerable. The signature $(a, b)$ is verified if $0 < a < p$ and $y^a a^b \equiv g^m \pmod p$.

Now let's look at Nguyen's attack, as described in [67]. Let $(a, b)$ be a valid signature for a message $m$. Since $(a, b)$ is a valid we know that

$0 < a < p$ and $y^a a^b \equiv g^m \pmod{p}$, but this second condition is equivalent to

$$ax + bk \equiv m \pmod{p-1}, \tag{26}$$

since $y^a a^b \equiv g^{xa} g^{kx} \equiv g^m \pmod{p}$. This congruence has two unknowns, $x$ and $k$, which are much smaller than the modulus. In fact, $x$ is a $\frac{3}{2}q_{bit}$-bit number, $k$ is roughly a $\frac{3}{2}q_{bit}$-bit number, and $p$ is at least a $4q_{bit}$-bit number. Nguyen proposed two methods to recover $x$ and $k$ from (26)

The first method uses an orthogonal lattice of the lattice of solutions of the homogeneous version of (26). Consider the 2-dimensional lattice

$$\mathcal{L}_1 = \left\{ (s,t) \in \mathbb{Z}^2 \;\middle|\; as + bt \equiv 0 \pmod{p-1} \right\}. \tag{27}$$

Let $d = \gcd(a, p-1)$ and $e = \gcd(b, p-1)$. Nguyen shows that one basis for $\mathcal{L}_1$ is given by basis matrix

$$\mathcal{M}_1 = \begin{bmatrix} (p-1)/d & \\ u & d/e \end{bmatrix},$$

where $u$ is any integer satisfying $au + (b/e)d \equiv 0 \pmod{p-1}$. Let $x'$ and $k'$ be any integers satisfying (26). Then the vector $\boldsymbol{\ell} = (x' - x, k' - k) \in \mathcal{L}_1$ will be close to the vector $\boldsymbol{z} = (x' - 2^{3q_{bit}/2}, k' - 2^{3q_{bit}/2})$. By the construction of $x$ and $k$ it is then expected that $\|\boldsymbol{\ell} - \boldsymbol{z}\| \approx 2^{(3q_{bit}+1)/2}$. When $e = \gcd(b, p-1)$ is small, the volume of the lattice satisfies

$$d(\mathcal{L}_1) = \frac{p-1}{\gcd(b, p-1)} \approx p.$$

Thus, it is expected that $\|\boldsymbol{\ell} - \boldsymbol{z}\| \ll d(\mathcal{L}_1)^{1/2}$ and hopefully $\boldsymbol{\ell}$ will be the closest vector in $\mathcal{L}_1$ to $\boldsymbol{z}$. The result can be proved when $a$ and $b$ are uniformly distributed modulo $p-1$.

Nguyen's second method involves finding the shortest vector in the lattice $\mathcal{L}_2$ generated by the rows of the basis matrix

$$\mathcal{M}_2 = \begin{bmatrix} (p-1)K & & & \\ -mK & 2^{3q_{bit}/2} & & \\ bK & & 1 & \\ aK & & & 1 \end{bmatrix},$$

where $K$ is some a large integer. The vector $(0, 2^{3q_{bit}/2}, k, x) \in \mathcal{L}_2$ is expected to be the shortest vector in $\mathcal{L}_2$. In experiments carried by Nguyen, the LLL-algorithm found this vector for all values in the Wiener table (see Table 1).

As a result of Nguyen's attacks, ElGamal signing keys have been removed from GPG.

# 3 Non-Linear Equations I (Theory)

In the discussion that follows, we will say that a polynomial $f(\cdot)$ is ***root equivalent*** to a polynomial $g(\cdot)$ if each root of $g(\cdot)$ is also a root of $f(\cdot)$. When the roots are modulo some integer $N$, we will say that $f(\cdot)$ is root equivalent to $g(\cdot)$ modulo $N$.

To motivate the techniques in this chapter, let $N$ be some large integer of unknown factorization and $f_N(x) \in \mathbb{Z}[x]$ be a polynomial of degree $d$. We are interested in finding solutions of the univariate modular equation

$$f_N(x) = a_d\, x^d + a_{d-1}\, x^{d-1} + \cdots + a_2\, x^2 + a_1\, x + a_0 \equiv 0 \pmod{N}.$$

In some instances, small solutions of the modular equation can be found by simply solving the equation $f_N(x) = 0$. Let $X$ be a bound on the size of the solutions that can be found this way. When $f_N(x) = x^d - a_0$ we have $X = N^{1/d}$, as any $|x_0| < X = N^{1/d}$ can be found by simply computing the $d^{\text{th}}$ roots of $a_0$ over the integers. More generally, if each coefficient of $f_N(x)$ satisfies $|a_i| < N^{(1-i/d)}/(d+1)$ then all solutions $|x_0| < X = N^{1/d}$ can be found by solving $f_N(x) = 0$ over the integers, since $N \,|\, p(x_0)$ and

$$|f_N(x_0)| \leq \sum_{i=0}^{d} |a_i|\, |x_0^i| < \frac{1}{d+1} \sum_{i=0}^{d} N^{(1-i/d)} N^{i/d} = N.$$

Another, more useful, sufficient condition for solutions of $f_N(x) \equiv 0 \pmod{N}$ to be solutions of $f_N(x) = 0$ is the following observation.

**Lemma 3.1.** *Let $h(x) \in \mathbb{Z}[x]$ be the sum of at most $\omega$ monomials. Suppose that $h(x_0) \equiv 0 \pmod{N}$ and $\|h(xX)\| < N/\sqrt{\omega}$, where $|x_0| < X$. Then $h(x_0) = 0$.*

Of course the coefficients of $f_N(x)$ will not, in general, be small enough to satisfy the conditions in Lemma 3.1 or the result preceding it. In order to make use of these results the methods in this chapter aim to find a polynomial $f(x)$ that is root equivalent to $f_N(x)$ modulo $N$ which also satisfies Lemma 3.1 so that $f(x_0) = 0$. To this end lattice basis reduction is used. First a lattice whose every element corresponds to the coefficient vector of a polynomial that is root equivalent to $f_N(x)$ modulo $N$ is constructed. Using lattice basis reduction a polynomial $f(x)$ with small norm is found (SVP). With this polynomial an ***enabling equation*** is derived. The enabling condition, which is actually an inequality, gives a sufficient condition on the bound $X$ so that all $|x_0| < X$ satisfying $f_N(x_0) \equiv 0 \pmod{N}$ will also satisfy $f(x_0) = 0$. Each integer solution of $f(x) = 0$ is then a potential solution

of $f_N(x) \equiv 0 \pmod{N}$. Using known techniques all integer roots of $f(x)$ are found and checked against the original modular equation. The number of such solutions will be bound by the degree of $f(x)$.

The main goal is to find methods that achieve the largest bound $X$ so that all solutions $|x_0| < X$ of the equation $f_N(x) \equiv 0 \pmod{N}$ can be found efficiently.

## 3.1 Modular Equations

We begin with non-linear univariate modular equations. Let $N$ be some large integer of unknown factorization (having no easy factors) and let $f_N(x) \in \mathbb{Z}[x]$ be a monic polynomial of degree $d$. We are interested in finding the largest bound $X$ such that all solutions $|x_0| < X$ of the modular equation

$$f_N(x) = x^d + a_{d-1}\, x^{d-1} + \cdots + a_2\, x^2 + a_1\, x + a_0 \equiv 0 \pmod{N},$$

can be found efficiently.

### 3.1.1 Early Efforts

In the mid to late 1980's, results by Håstad [38, 39] and Vallée, Girault & Toffin [88, 33], show that $X = N^{\frac{2}{d(d+1)} - \epsilon}$ is attainable where $\epsilon > 0$ is a function of the degree $d$.

We briefly outline a method that achieves this bound. Let $X$ be our bound and define the $d+1$ polynomials $f_i(x)$ for $i = 0, \ldots, d$ by

$$f_i(x) = \begin{cases} Nx^i & 0 \le i \le d-1 \\ f_N(x) & i = d \end{cases}.$$

Consider the $(d+1)$-dimensional lattice $L$ generated by the basis matrix $M$ whose rows are the coefficient vectors of $f_i(xX)$ for $i = 0, \ldots, d$. The basis matrix is given by

$$M = \begin{bmatrix} N & & & & & \\ & NX & & & & \\ & & NX^2 & & & \\ & & & \ddots & & \\ & & & & NX^{d-1} & \\ a_0 & a_1X & a_2X^2 & \cdots & a_{d-1}X^{d-1} & X^d \end{bmatrix}.$$

Notice that any element in the $L$ can be written as

$$\Big((ca_0 - c_0N), (ca_1 - c_1N)X, \ldots, (ca_{d-1} - c_{d-1}N)X^{d-1}, cX^d\Big),$$

which corresponds to the coefficient vector of some polynomial $h(x)$ given by

$$h(x) = (ca_0 - c_0N) + (ca_1 - c_1N)x + \cdots + (ca_{d-1} - c_{d-1}N)x^{d-1} + cx^d,$$

evaluated at $xX$. Thus, each element of $L$ corresponds to a polynomial $h(x)$ that is root equivalent to $f_N(x)$ modulo $N$, as $h(x) \equiv c \cdot f_N(x) \pmod{N}$. We now use lattice basis reduction to find a small normed element of $L$. Let $h(x)$ be the polynomial whose coefficient vector, evaluated at $xX$, is the smallest element returned by the LLL-algorithm. Using (16), we know that $h(x)$ satisfies

$$\|h(xX)\| \leq 2^{\frac{d}{4}} d(L)^{\frac{1}{d+1}}.$$

In order to apply Lemma 3.1 on $h(x)$, so that $h(x_0) = 0$, it is sufficient that

$$\|h(xX)\| < N/\sqrt{d+1}.$$

Combining these, we see that a sufficient condition for $h(x_0) = 0$ to hold is given by

$$2^{\frac{d}{4}} d(L)^{\frac{1}{d+1}} < N/\sqrt{d+1}.$$

Substituting $d(L) = N^d X^{\frac{d(d+1)}{2}}$ (which is simply the product of the diagonal elements of $M$) and solving for $X$ we obtain the enabling equation

$$X < N^{\frac{2}{d(d+1)} - \epsilon},$$

where $\epsilon > 0$ is function of $d$ only.

Essentially, this method uses lattice basis reduction to find a polynomial $h(x)$ that is simply a constant multiple of $f_N(x)$ modulo $N$.

### 3.1.2 Coppersmith's Method

The main advancement over the previous results came in 1996 when Coppersmith [21, 22], increased the bound from $N^{2/d(d+1)}$ to $N^{1/d}$. This improvement is a result of considering polynomial combinations of $f_N(x)$ modulo $N^u$ for some integer $u$ instead of just a multiple of $f_N(x)$ modulo $N$ as in the previous section. In the original presentation [21, 22], Coppersmith was working with an unnatural space. The presentation was difficult to follow and was not easily transfered to practical implementations. However, shortly after in 1997, Howgrave-Graham[42] gave an alternate presentation that was more natural and easily implemented. In fact, all current uses of Coppersmith's univariate modular method use Howgrave-Graham's approach.

We present a generalization of Coppersmith's result for univariate modular polynomials as given by May[58] in 2004. This is the best known result for univariate modular polynomial equations to date.

**Theorem 3.1 (Coppersmith).** *Let $N$ be an integer of unknown factorization, which has a divisor $b \geq N^\beta$. Let $f_b(x)$ be a monic univariate polynomial of degree $d$, $c_N$ be a function that is upper-bounded by a polynomial in $\log N$, and $\epsilon > 0$. Then we can find all solutions $x_0$ for the equation $f_b(x) \equiv 0 \pmod{b}$ such that*

$$|x_0| \leq c_N N^{\beta^2/d-\epsilon},$$

*in polynomial time.*

*Proof.* Let $h \geq 2$ and $m \geq hd - 1$ be arbitrary but fixed integers and let $X$ be our bound on the solutions of the equation. For integers $i \geq 0$ and $0 \leq j \leq h$ define the $m$ polynomials $f_{i,j}(x)$ by

$$f_{i,j}(x) = N^{h-j}\, x^i\, (f_b(x))^j.$$

By construction, each $x_0$ that is a root of $f_N(x)$ modulo $N$ is also a root of $f_{i,j}(x)$ modulo $N^h$.

Now consider the $m$-dimensional full rank lattice $L$ generated by a basis matrix $M$ whose rows are the coefficient vectors of $f_{i,j}(xX)$ for $i \geq 0$ and $0 \leq j \leq h$. Each element of $L$ is the coefficient vector of a polynomial that is an integer linear combination of the $f_{i,j}(xX)$.

Using the LLL-algorithm, we can find a small element in $L$ that corresponds to a polynomial $h(x)$ satisfying (see (16))

$$\|h(xX)\| \leq 2^{(m-1)/4} d(L)^{1/m}.$$

The basis matrix is triangular (with a proper ordering of the $f_{i,j}(xX)$). A simple calculation shows that the volume of $L$ is given by

$$d(L) = N^{dh(h+1)/2} X^{m(m-1)/2}.$$

In order to apply the integer equation property, (Lemma 3.2), on $h(x)$ it is sufficient that $\|h(xX)\| < b^h/\sqrt{m}$ holds. Since $N^\beta \leq b$, a sufficient condition for this inequality to hold is

$$2^{(m-1)/4} N^{dh(h+1)/(2m)} X^{(m-1)/2} < N^{\beta h}/\sqrt{m},$$

as this implies

$$\|f(xX)\| \leq 2^{(m-1)/4} N^{dh(h+1)/(2m)} X^{(m-1)/2} < N^{\beta h}/\sqrt{m} \leq b^h/\sqrt{m}.$$

Rearranging to isolate $X$ we see that this is equivalent to

$$X \leq \frac{1}{\sqrt{2}}\, m^{\frac{-1}{m-1}}\, N^{\frac{2m\beta h - dh(h+1)}{m(m-1)}}.$$

We now look for an optimal $h$ value to maximize $X$. To this end, we consider the exponent of $N$ in the above inequality as a polynomial in $h$:

$$\left(\frac{-d}{m(m-1)}\right) h^2 + \left(\frac{2m\beta - d}{m(m-1)}\right) h.$$

Notice that for any values of $d$ and $m$ this expression attains its maximum when $h$ is chosen to be $h_0 = \frac{2\beta m - d}{2d}$. Substituting this into the bound for $X$, we have

$$X \leq \frac{1}{\sqrt{2}}\, m^{\frac{-1}{m-1}}\, N^{\frac{\beta^2}{d}\frac{m}{m-1} - \frac{\beta}{m} + \frac{d}{4m(m-1)}}.$$

Since $\frac{\beta^2}{d}\frac{m}{m-1} = \frac{\beta^2}{d} + \frac{\beta^2}{d(m-1)}$, we see that this inequality is satisfied whenever

$$X \leq X_0 = \frac{1}{\sqrt{2}}\, N^{\beta^2/d - \epsilon},$$

where $\epsilon = \frac{1}{m-1}\log_N m + \frac{\beta}{m}$. Therefore, any solution $x_0$ to $f_b(x) \equiv 0 \pmod{b}$ such that

$$|x_0| \leq \frac{1}{\sqrt{2}}\, N^{\beta^2/d - \epsilon},$$

is also a solution to the equation $h(x) = 0$. Now, for any $c_N$ (a function of $N$) we can partition the range $(0, c_N N^{\beta^2/d-\epsilon}]$ into intervals

$$I_i = \left(\lfloor iX_0 \rfloor, \lceil (i+1)X_0 \rceil\right],$$

for all integers $0 \leq i \leq \sqrt{2}c_N$. Applying the above method for each interval $I_j$ with the function $f_b(x + \lfloor jX_0 \rfloor)$ for all $0 \leq j \leq \lceil \sqrt{2}c_N \rceil + 2$ such that $3 \mid j$ will find all positive solutions $x_0 \leq c_N N^{\beta^2/d-\epsilon}$. The small negative solutions can be obtained by applying the method to the same intervals using $f_b(x - \lceil jX_0 \rceil)$. Thus, all solutions $|x_0| \leq c_N N^{\beta^2/d-\epsilon}$ can be found in polynomial time, provided that $c_N$ is polynomial in $\log N$. □

Notice that in this case, unlike the results of the previous section, the error term $\epsilon$ can be made arbitrarily small by using an arbitrarily large lattice dimension $m$. That is,

$$\lim_{m\to\infty} \epsilon = 0.$$

So we can make the bound as close to $c_N N^{\beta^2/d}$ as we want at the expense of using larger lattice dimensions.

Coppersmith's original result is the special case of $b = N$, which states that all roots $|x_0| < N^{1/d-\epsilon}$ of a univariate modular polynomial of degree $d$ can be recovered in polynomial time.

### 3.1.3 Multivariate Modular Equations

Let $f_N(x_1, \ldots, x_\ell) \in \mathbb{Z}[x_1, \ldots, x_\ell]$ be a multivariate polynomial in $\ell$ variables with integer coefficients. We are interested in finding solutions $\overline{y} = (y_1, \ldots, y_\ell)$ to the modular equation

$$f_N(\overline{x}) = f_N(x_1, \ldots, x_\ell) = \sum_{i_1,\ldots,i_\ell} a_{i_1,\ldots,i_\ell} x_1^{i_1} \cdots x_\ell^{i_\ell} \equiv 0 \pmod{N}. \qquad (28)$$

The results of the previous two sections are easily extended to the multivariate case. Håstad's result was extended by Takagi & Naito [87] (for the first presentation in [38]) and Joye, Koeunne & Quisquater [48] (for the improved presentation in [39]). Both of these methods show how to construct a single polynomial $h(\overline{x}) \in \mathbb{Z}[x_1, \ldots, x_\ell]$ that satisfies $h(\overline{y}) = 0$. For Coppersmith's method, Jutla [49] was the first to show how to construct many polynomials with root $\overline{y}$ over the integers. Since then, other instances have appeared in the literature (for example see Boneh & Durfee [8] for bivariate polynomials and Durfee & Nguyen [28] for trivariate polynomials). We will outline the general framework of the multivariate case below. As in the univariate case, we follow Howgrave-Graham's presentation.

Before proceeding, we state a generalization of Lemma 3.1 which we will call the ***integer equation property***.

**Lemma 3.2 (Integer Equation Property).** *For any integer $\ell \geq 1$, let $h(x_1, \ldots, x_\ell) \in \mathbb{Z}[x_1, \ldots, x_\ell]$ be the sum of at most $\omega$ monomials and let $u$ be a positive integer. Suppose that*

$$h(y_1, \ldots, y_\ell) \equiv 0 \pmod{N^u} \qquad \textit{and} \qquad \|h(x_1 X_1, \ldots, x_\ell X_\ell)\| < N^u/\sqrt{\omega},$$

*where $|y_i| < X_i$ for $1 \leq i \leq \ell$. Then $h(y_1, \ldots, y_\ell) = 0$.*

We will assume that the equation $f_N(\overline{x}) \equiv 0 \pmod{N}$ has a small solution $\overline{y} = (y_1, \ldots, y_\ell)$. That is, $f_N(\overline{y}) \equiv 0 \bmod N$ such that $|y_1| \leq X_1, \ldots, |y_\ell| \leq X_\ell$ where $X_1, \ldots, X_\ell \in \mathbb{Z}$. We are interested in finding the maximum bounds $X_1, \ldots, X_\ell$ so that all such solutions $\overline{y}$ can be found efficiently.

As in the univariate case, we will construct a lattice whose every element corresponds to a polynomial that is root equivalent to $f_N(\overline{x})$ modulo $N$. Using lattice basis reduction we will look for $\ell$ small vectors that correspond to polynomials that satisfy the conditions in the integer equation property. We then hope to solve the nonlinear system of $\ell$ equations in $\ell$ unknowns to recover the roots of $f_N(\overline{x})$ modulo $N$.

Let $m$ and $d$ be positive integers. Define the polynomial

$$f_{\gamma_1,\ldots,\gamma_\ell,j}(\overline{x}) = f_{\gamma_1,\ldots,\gamma_\ell,j}(x_1,\ldots,x_\ell) \in \mathbb{Z}[x_1,\ldots,x_\ell]$$

by

$$f_{\gamma_1,\ldots,\gamma_\ell,j}(\overline{x}) = N^{m-j} x_1^{\gamma_1} \cdots x_\ell^{\gamma_\ell} (f_N(\overline{x}))^j, \tag{29}$$

where $0 \leq j \leq m$ and $\gamma_i \geq 0$ for $i = 1,\ldots,\ell$ are integers. By construction, $\overline{y}$ is a root of $f_{\gamma_1,\ldots,\gamma_\ell,j}(\overline{x})$ modulo $N^m$ for all valid $j$ and $\gamma_i$. Also, for any fixed $j$, all polynomials of the form (29) with different $(\gamma_1,\ldots,\gamma_\ell)$ values are linearly independent. We construct the $d$-dimensional lattice $L$ whose basis matrix $M$ is made up of the coefficient vectors of $d$ linearly independent polynomials of the form

$$f_{\gamma_1,\ldots,\gamma_\ell,j}(x_1X_1,\ldots,x_\ell X_\ell).$$

With a clever choice of the $(\gamma_1,\ldots,\gamma_\ell,j)$ one can construct $M$ so that it is triangular, which allows easy computation of the lattice volume. The particular choice of the $(\gamma_1,\ldots,\gamma_\ell,j)$ is very dependent on the polynomial $f_N(\overline{x})$.

Using the LLL-algorithm, we find $\ell$ linearly independent vectors in $L$ corresponding to $\ell$ linearly independent polynomials $p_i(\overline{x})$ such that

$$\|p_1(x_1X_1,\ldots,x_\ell X_\ell)\| \leq \cdots \leq \|p_\ell(x_1X_1,\ldots,x_\ell X_\ell)\|,$$

and

$$\|p_\ell(x_1X_1,\ldots,x_\ell X_\ell)\| \leq c(\ell,d) \cdot d(L)^{\frac{1}{d-\ell+1}},$$

Here $c(\ell,d)$ is a function that depends only on $\ell$ and $d$ (see (19) and (20) in Section 1.2). A sufficient condition to apply the integer equation property, (Lemma 3.2), on each of these polynomials is that the right-hand side of the above inequality be bound by $N^m/\sqrt{d}$. That is,

$$\|p_\ell(x_1X_1,\ldots,x_\ell X_\ell)\| \leq c(\ell,d)\, d(L)^{\frac{1}{d-\ell+1}} < N^m/\sqrt{d}.$$

From this, we can derive an enabling equation for the bounds $X_i$. Generally, when deriving the enabling equation the terms $c(\ell,d)$ and $\sqrt{d}$ are assumed

to be negligible as compared to the rest of the terms and are ignored. This greatly simplifies the bounds.

When the enabling equation is satisfied we have $\ell$ linearly independent polynomials $p_i(\overline{x})$ such that $p_i(\overline{y}) = 0$ for $i = 1, \ldots, \ell$. In order to solve for $\overline{y}$ we must solve a system of $\ell$ non-linear equations in $\ell$ variables. In general, there is no known method to do this. However, in the special case when all the polynomials are also algebraically independent we can solve for $\overline{y}$.

In this case, using resultant computations, we construct a family of polynomials $g_{i,j}(x_1, \ldots, x_i)$ such that for each $i = 1, \ldots, \ell - 1$ and $j = 1, \ldots, i$ we have $g_{i,j}(x_1, \ldots, x_i) \in \mathbb{Z}[x_1, \ldots, x_i]$ and $g_{i,j}(y_1, \ldots, y_i) = 0$. Then, starting with $g_{1,1}(x_1)$ we solve $g_{1,1}(x_1) = 0$ for $y_1$ and back-substitute into one of the $g_{2,j}(x_1, x_2)$ to solve for $y_2$. That is, we solve $g_{2,j}(y_1, x_2) = 0$ for $y_2$ where $j \in \{1, 2\}$. We keep solving for roots of univariate polynomials and back-substituting until all of the desired roots are found.

For example, when $\ell = 3$ we begin with the three polynomials $p_i(x_1, x_2, x_3)$ for $i = 1, 2, 3$ and compute the three new polynomials $g_{i,j}(\cdot)$ as follows

$$\left.\begin{array}{l} p_1(x_1, x_2, x_3) \\ p_2(x_1, x_2, x_3) \\ p_3(x_1, x_2, x_3) \end{array}\right\} \begin{array}{l} g_{2,1}(x_1, x_2) = \mathrm{Res}_{x_3}(p_1, p_2) \\ g_{2,2}(x_1, x_2) = \mathrm{Res}_{x_3}(p_2, p_3) \end{array} \Bigg\} g_{1,1}(x_1) = \mathrm{Res}_{x_2}(g_{2,1}, g_{2,2}).$$

We then solve $g_{1,1}(x_1) = 0$ for all integer roots $\widehat{y}_1$. For each $\widehat{y}_1$ solve $g_{2,1}(\widehat{y}_1, x_2) = 0$ for all integer roots $\widehat{y}_2$. For each $\widehat{y}_1$ and $\widehat{y}_2$ we then solve $p_1(\widehat{y}_1, \widehat{y}_2, x_3) = 0$ for all integer roots $\widehat{y}_3$. Then, we test $f_N(\widehat{y}_1, \widehat{y}_2, \widehat{y}_3) \equiv 0 \pmod{N}$ for all $\widehat{y}_1, \widehat{y}_2$, and $\widehat{y}_3$ to find the actual $y_1, y_2$, and $y_3$.

When the polynomials are algebraically dependent, however, it is usually thought that this method cannot work. In general it does not work because the resultant of two algebraically dependent polynomials is always zero. In some special cases this algebraic dependence can be removed though. For example, suppose $g_1(x, y) = g(x, y) \cdot \widehat{g}_1(x, y)$ and $g_2(x, y) = g(x, y) \cdot \widehat{g}_2(x, y)$ where $\widehat{g}_1(x, y)$ and $\widehat{g}_2(x, y)$ are algebraically independent and $(x_0, y_0)$ is the common root we want to find. If it happens that $\widehat{g}_1(x_0, y_0) = 0 = \widehat{g}_2(x_0, y_0)$, then we can simply compute the resultant of $\widehat{g}_1(x, y)$ and $\widehat{g}_2(x, y)$ to remove one of the variables instead of trying to use $g_1(x, y)$ and $g_2(x, y)$. Also, the $\widehat{g}_i(x, y)$ are easily computed by $\widehat{g}_i(x, y) = g_i(x, y) / \gcd(g_1(x, y), g_2(x, y))$. We will call the polynomials $g_1(x, y)$ and $g_2(x, y)$ ***weakly algebraically dependent*** in this case, because we can remove the algebraic dependence. Unfortunately, if it happens that the common root is only a root of $g(x, y)$ then there is no known method of finding $x_0$ and $y_0$ from $g_1(x, y)$ and $g_2(x, y)$. In this case, we call these polynomials ***strongly algebraically dependent***.

There is currently very little theory to predict the algebraic dependence of the reduced basis vectors for a given lattice. For this reason, Coppersmith's method for finding small roots of multivariate modular polynomials is only a heuristic method. In cryptographic applications it is often assumed that the reduced basis vectors will be algebraically independent (based sometimes on only a few experiments). To date, there has been only one example, in the literature, of a cryptographic application of Coppersmith's method to multivariate polynomials that results in strongly algebraically dependent reduced basis vectors (see [6]).

### 3.1.4 Small Inverse Problem

As an example of solving a multivariate modular equation we will consider the so-called ***small inverse problem*** defined by Boneh & Durfee [8]. This is an example of solving a bivariate modular equation. An instance of the small inverse problem consists of integers $A$ and $B$ and bounds $X$ and $Y$. The problem is to find all integers $a$ that are *close* to $A$ whose inverse modulo $B$ is *small*, where close means $|a - A| < X$ and small means that $|a^{-1} \pmod e| < Y$. That is, we look for $x$ and $y$ such that $x(A + y) \equiv 1 \pmod B$, where $|x| < X$ and $|y| < Y$. Let $X = B^\alpha$ and $Y = B^\beta$ for $0 \leq \alpha, \beta \leq 1$. We are interested in finding the largest possible bounds ($\alpha$ and $\beta$) such that we can solve the problem efficiently for given $A$ and $B$.

Boneh & Durfee consider the case when $\beta = 0.5$ is fixed and try to maximize $\alpha$. They present a method that works whenever $\alpha < 0.284$ and then extend this bound to $\alpha < 0.292$. Of course, both methods are only heuristic. Their first result can be generalized by the following result.

**Theorem 3.2 (Small Inverse Problem).** *Given large integers $A$ and $B$, let $X = B^{\alpha-\epsilon_\alpha}$ and $Y = B^{\beta-\epsilon_\beta}$ where $0 < \alpha, \beta < 1$ satisfy*

$$-3\alpha^2 + (2\beta + 6)\alpha + \beta^2 + 2\beta - 3 < 0,$$

*and $\epsilon_\alpha$ and $\epsilon_\beta$ are positive real numbers. Then we can find two linearly independent polynomials $p_1(x, y)$ and $p_2(x, y)$ such that all solutions $(x_0, y_0)$ of $x(A+y) \equiv 1 \pmod B$ with $|x_0| < X$ and $|y_0| < Y$ also satisfy $p_1(x_0, y_0) = 0$ and $p_2(x_0, y_0) = 0$. Further, we can find these polynomials in polynomial time.*

*Proof.* Notice that the small inverse problem is equivalent to the problem of finding all small roots of the modular polynomial equation

$$f_B(x, y) = x(A + y) - 1 \equiv 0 \pmod B.$$

Let $X = B^\alpha$ and $Y = B^\beta$ where $0 < \alpha, \beta < 1$ are our bounds on $x$ and $y$, respectively. Also, let $m \geq 1$ and $t \geq 0$ be integers (to be determined later). Define the $x$- and $y$-shift polynomials of $f_B(x, y)$ as

$$g_{i,k}(x, y) = B^{h-k} x^i (f_B(x, y))^k \text{ and } h_{j,k}(x, y) = B^{h-k} y^j (f_B(x, y))^k,$$

respectively. Notice that each $(x_0, y_0)$ satisfying $f_B(x_0, y_0) \equiv 0 \pmod{B}$ also satisfies $g_{i,k}(x_0, y_0) \equiv 0 \pmod{B^m}$ and $h_{j,k}(x_0, y_0) \equiv 0 \pmod{B^m}$ for all $i, j \geq 0$ and $0 \leq k \leq m$. We construct the lattice $L$ whose basis matrix $M$ is made up of the coefficient vectors of the $w = (m+1)(m+2)/2 + t(m+1)$ $x$- and $y$-shift polynomials

$$\begin{aligned} &g_{i,k}(xX, yY) \quad \text{for all} \quad 0 \leq i \leq m-k \ , \ 0 \leq k \leq m, \text{ and} \\ &h_{j,k}(xX, yY) \quad \text{for all} \quad 1 \leq j \leq t \ , \ 0 \leq k \leq m. \end{aligned}$$

With a proper ordering of the polynomials we see that $M$ is triangular with all diagonal elements nonzero. Thus, the lattice is full rank with dimension $w$. A simple calculation shows that the volume of the lattice given by

$$d(L) = B^{C_B} X^{C_X} Y^{C_Y},$$

where $C_B, C_X$, and $C_Y$ are given by

$$C_B = C_X = m(m+1)(m+2)/3 + tm(m+1)/2,$$

$$C_Y = m(m+1)(m+2)/6 + t(m+1)(m+t+1)/2.$$

Using $X = B^\alpha$ and $Y = B^\beta$ and letting $t = \tau m$ for some real number $\tau \geq 0$, we see that

$$\begin{aligned} d(L) \quad = \quad & \frac{1}{6}\left(3\,\beta\,\tau^2 + (3\,\alpha + 3 + 3\,\beta)\,\tau + 2 + 2\,\alpha + \beta\right) m^3 \\ & + \frac{1}{6}\left(3\,\beta\,\tau^2 + (3\,\alpha + 3 + 6\,\beta)\,\tau + 6\,\alpha + 6 + 3\,\beta\right) m^2 \\ & + \frac{1}{6}\left(3\,\beta\,\tau + 4 + 4\,\alpha + 2\,\beta\right) m, \end{aligned}$$

and

$$m(w-1) = \frac{1}{2}\left(1 + 2\tau\right) m^3 + \frac{1}{2}\left(3 + 2\tau\right) m^2.$$

Using the LLL-algorithm, we know (from (19) and (20)) that we can find two vectors that correspond to polynomials $p_1(x, y)$ and $p_2(x, y)$ satisfying

$$\|p_1(xX, yY)\| \leq \|p_2(xX, yY)\| \leq 2^{w/4} d(L)^{1/(w-1)}.$$

A sufficient condition to apply the integer equation property, (Lemma 3.2), on these polynomials is that the right-hand side of the above inequality is bounded by $B^m/\sqrt{w}$. Thus, we insist that

$$2^{w/4} d(L)^{1/(w-1)} < B^m/\sqrt{w},$$

or

$$d(L) < 2^{-w(w-1)/4} w^{-(w-1)/2} B^{m(w-1)}.$$

We now consider when $B \gg w$ and $m$ is large. In this case, we can neglect the terms that do not depend on $B$ and only keep the higher order terms of $m$. This leaves us with

$$B^{\frac{1}{6}\left(3\,\beta\,\tau^2+(3\,\alpha+3+3\,\beta)\tau+2+2\,\alpha+\beta\right)m^3+o(m^3)} < B^{\frac{1}{2}(1+2\tau)m^3+o(m^3)}.$$

Focusing only on the exponents of $B$ and simplifying, we have

$$\frac{1}{6}\left(3\,\beta\,\tau^2+3\,(-1+\alpha+\beta)\,\tau-1+2\,\alpha+\beta\right)m^3 < 0+o(m^3).$$

For large enough $m$, it is sufficient that the coefficient of $m^3$ in the above inequality be less than zero. This happens when

$$3\,\beta\,\tau^2+3\,(-1+\alpha+\beta)\,\tau-1+2\,\alpha+\beta<0.$$

For any values of $\alpha$ and $\beta$, the left-hand side of this inequality is minimized when $\tau$ is chosen to be

$$\tau_{\text{opt}} = \frac{1-\alpha-\beta}{2\beta}.$$

Substituting this back into the inequality yields the enabling equation

$$-3\alpha^2+(2\beta+6)\alpha+\beta^2+2\beta-3<0,$$

which is the desired condition. The real numbers $\epsilon_\alpha$ and $\epsilon_\beta$ represent the neglected factors (that were independent of $B$) and lower order terms of $m$. The exact values of these error terms depend on the size of $B$ and the lattice parameters ($m$ and $t$) used. For large enough $B$, these numbers can be made arbitrarily small by using larger $m$ values (and hence larger lattice dimensions). □

The enabling equation of the preceding result is perhaps better understood when the variables are separated. This gives

$$\alpha < 1+\frac{\beta}{3}-\frac{2}{3}\sqrt{\beta^2+3\beta}, \text{ and}$$

$$\beta < 2\sqrt{\alpha^2 - \alpha - 1} - 1 - \alpha.$$

Thus, if $\alpha$ and $\beta$ satisfy the enabling equations above, the method will produce two polynomials $p_1(x, y)$ and $p_2(x, y)$ that are both root equivalent to $f_B(x, y)$ modulo $B$. Further, if $p_1(x, y)$ and $p_2(x, y)$ are also not strongly algebraically dependent then we can use resultants to solve for all the $(x_0,y_0)$. In particular, we can compute the polynomial $p_{1,2}(x) = \text{Res}_y(p_1(x, y), p_2(x, y))$ and solve $p_{1,2}(x) = 0$ for candidates of $x_0$. For each candidate $\widehat{x}_0$ we then solve $p_1(\widehat{x}_0, y) = 0$ for candidates of $y_0$. We can then test all possible candidate possibilities with the original equation $f_B(x, y) \equiv 0 \pmod{B}$.

## 3.2 Integer Equations

### 3.2.1 The Bivariate Case

The problem of finding integer solutions of bivariate integer equations was also considered by Coppersmith[20, 22] in 1996. As with the univariate modular case, the presentation took place in an unnatural space. In 2004, Coron[24] presented a simplification of the method, much like Howgrave-Graham simplified the univariate modular case, that is slightly weaker than Coppersmith's original description but much more natural. The bounds on the solution are the same, but the runtime is exponential (rather then polynomial) in the degree of the polynomial. We will follow Coron's presentation and give a sketch of his proof. The main result is as follows.

**Theorem 3.3 (Coppersmith).** *Let $f(x, y) = \sum_{i,j} a_{i,j} x^i y^j$ be an irreducible polynomial in two variables over $\mathbb{Z}$. Let $X$ and $Y$ be upper bounds on the desired integer solution $(x_0, y_0)$, and let $W = \|f(xX, yX)\|_\infty$.*

1. *If $f(x, y)$ has maximum degree $d$ in each variable separately and*

$$XY < W^{2/(3d)-\epsilon}$$

   *for some $\epsilon > 0$, then in time polynomial in $(\log W, 2^d)$, one can find all integer pairs $(x_0, y_0)$ such that $|x_0| < X$, $|y_0| < Y$, and $f(x_0, y_0) = 0$.*

2. *If $f(x, y)$ has total degree $d$ and*

$$XY < W^{1/d-\epsilon}$$

   *for some $\epsilon > 0$, then in time polynomial in $(\log W, 2^d)$, one can find all integer pairs $(x_0, y_0)$ such that $|x_0| < X$, $|y_0| < Y$, and $f(x_0, y_0) = 0$.*

Before proceeding with the proof, we will need the following result about multiples of bivariate polynomials. The result is based on a univariate result of Mignotte[62].

**Lemma 3.3 (Coron[24]).** *Let $p(x,y)$ and $q(x,y)$ be two non-zero polynomials over $\mathbb{Z}$ of maximum degree $d$ in each of $x$ and $y$, such that $q(x,y)$ is a multiple of $p(x,y)$ in $\mathbb{Z}[x,y]$. Assume that $p(0,0) \neq 0$ and $q(x,y)$ is divisible by a non-zero integer $r$ such that $\gcd(r, p(0,0)) = 1$. Then $q(x,y)$ is divisible by $r \cdot p(x,y)$ and*

$$\|q(x,y)\| \geq 2^{-(d+1)^2}|r| \cdot \|p(x,y)\|_\infty.$$

*Sketch Proof.* We will only outline the proof of the first result for a bivariate polynomial with maximum degree $d$ in each variable separately.

Let $(x_0, y_0)$ be a root of $f(x,y)$, as given in Theorem 3.3, and let $X$ and $Y$ be bounds on $x_0$ and $y_0$, respectively. First we convert the problem to a bivariate modular equation. Without loss of generality, we will assume that $f(0,0) \neq 0$ and $\gcd(f(0,0), XY) = 1$. For some positive integer $k$ we define the modulus $n = u(XY)^k$, where

$$u = W + ((1 - W) \pmod{|f(0,0)|}).$$

Let $g(x,y)$ be the polynomial

$$g(x,y) = a_{0,0}^{-1} f(x,y) \bmod n = 1 + \sum_{(i,j)\neq(0,0)} b_{i,j} x^i y^j,$$

and for non-negative integers $i$ and $j$ define the set of polynomials $g_{i,j}(x,y)$ by

$$g_{i,j}(x,y) = \begin{cases} x^i y^j X^{k-i} Y^{k-j} q(x,y) & \text{if } 0 \leq i,j \leq k \\ x^i y^j n & \text{otherwise.} \end{cases}$$

By construction we have $g_{i,j}(x_0, y_0) \equiv 0 \pmod{n}$ and $XY | g_{i,j}(xX, yY)$ for all $0 \leq i,j \leq d+k$.

Letting $w = d + k + 1$, we construct the $w^2$-dimensional lattice $L$ with basis matrix $M$ whose rows are the coefficient vectors of $g_{i,j}(xX, yY)$ for all $0 \leq i,j \leq d+k$. Thus, each element of $L$ corresponds to a polynomial having $(x_0, y_0)$ as a root modulo $n$ and is a multiple of $XY$. Using the LLL-algorithm, we look for a small element of $L$ which corresponds to a small normed polynomial $h(xX, yY)$ that satisfies both

$$\|h(xX, yY)\| < \frac{n}{\sqrt{w}} \quad \text{and} \quad \|h(xX, yY)\| < 2^{-w}(XY)^k W. \tag{30}$$

The first condition ensures that $h(x_0, y_0) = 0$ by satisfying the integer equation property, Lemma 3.2. The second condition, using Lemma 3.3 with $p(x, y) = f(xX, yY)$, $q(x, y) = h(xX, yY)$ and $r = (XY)^k$, ensures that $h(xX, yY)$ is not a multiple of $f(xX, yY)$, which implies that $h(x, y)$ is not a multiple of $f(x, y)$. Since $f(x, y)$ is irreducible, if $h(x, y)$ is not a multiple of $f(x, y)$ then we know that the two polynomials must be algebraically independent. Therefore, when both of the above conditions are met, we have two polynomials, $f(x, y)$ and $h(x, y)$, that are algebraically independent and both have the root $(x_0, y_0)$ over the integers. We can then solve for $(x_0, y_0)$ using resultants.

Using the fact that the LLL-algorithm will yield a polynomial $h(x, y)$ such that

$$\|h(xX, yY)\| \leq 2^{\frac{w^2-1}{4}} d(L)^{\frac{1}{w^2}},$$

Coron shows that a sufficient condition on $X$ and $Y$ for (30) to hold is given by

$$XY < 2^{-\beta} W^{\alpha},$$

where

$$\alpha = \frac{2(k+1)^2}{(d+k)(d+k+1)^2 - k(k+1)^2} \geq \frac{2}{3d} - \frac{2}{3(k+1)}$$

and

$$\beta = \frac{10}{4} \frac{(d+k+1)^4 + (d+k+1)^2}{(d+k)(d+k+1)^2 - k(k+1)^2} \leq \frac{4k^2}{d} + 13d.$$

Choosing $k = \lfloor 1/\epsilon \rfloor$, this reduces to

$$XY < W^{2/(3d)-\epsilon} 2^{-4/(d\epsilon^2)-13d}.$$

Doing an exhaustive search on the $4/(d\epsilon^2) - 13d$ high order bits of $x_0$ gives the desired bound. □

The case when $f(x, y)$ has total degree $d$ follows a similar method and can be found in [24, Appendix B].

### 3.2.2 General Multivariate Integer Equations

Coppersmith's method for finding small integer roots of bivariate integer equations can be heuristically extended to finding small integer roots of multivariate integer equations with more than two variables. This is discussed by both Coppersmith[20, 22] and Coron[24].

We briefly outline the method. Let $f(x_1, \ldots, x_\ell) \in \mathbb{Z}[x_1, \ldots, x_\ell]$ be a polynomial in $\ell$ variables with degree $d$ in each variable. We wish to find all integer solutions $\overline{y}$ of $f(\overline{x}) = 0$ such that $|y_i| < X_i$ for $i = 1, \ldots, \ell$.

First we construct a modulus $n$ that is a multiple of $(X_1 X_2 \cdots X_\ell)^k$ for some positive integer $k$ and a new polynomial $g(\overline{x})$ that has constant term 1 and is root equivalent to $f(\overline{x})$ modulo $n$. We then consider the lattice $L$ generated by the coefficient vectors of the polynomials

$$x_1^{\gamma_1} \cdots x_\ell^{\gamma_\ell} X^{k-\gamma_i} \cdots X^{k-\gamma_\ell} q(x_1 X_1, \ldots, x_\ell X_\ell) \quad \text{for} \quad 0 \leq \gamma_1, \ldots, \gamma_\ell \leq k,$$

and

$$(x_1 X_1)^{\gamma_1} \cdots (x_\ell X_\ell)^{\gamma_\ell} \quad \text{for} \quad (\gamma_1, \ldots, \gamma_\ell) \in [0, d+k]^\ell \setminus [0, k]^\ell.$$

We then consider the smallest $\ell - 1$ elements of $L$ as found by the LLL-algorithm. These lattice elements are the coefficient vectors of $\ell - 1$ polynomials $h_i(x_1 X_1, \ldots, x_\ell X_\ell)$. A sufficient condition on the bounds $X_i$ can then be derived so that each of these polynomials satisfy the integer equation property, Lemma 3.2, and satisfy a generalized version of Lemma 3.3. When this enabling equation is satisfied, we then have $\ell$ equations in $\ell$ variables that have the integer root $\overline{y}$ (the $\ell - 1$ polynomials $h_i(\overline{x})$ and the original equation $f(\overline{x})$). We also know that each of the $h_i(\overline{x})$ is algebraically independent of $f(\overline{x})$. The method remains a heuristic of course, since there are no general way of knowing if the $h_i(\overline{x})$ are pairwise algebraically independent or not.

### 3.2.3 Common Divisors

In 2001, Howgrave-Graham [44] extended the notion of solving integer equations to finding approximate common divisors. We define the problems below.

An instance of the ***approximate common divisor problem (ACDP)*** consists of two inputs $a_0$ and $b_0$ and bounds $X$, $Y$, and $D$ such that there exists integers $|x_0| < X$, $|y_0| < Y$, and $d > D$ satisfying $d|(a_0 + x_0)$ and $d|(b_0 + y_0)$. The problem is to find all $x_0, y_0$, and $d$ that satisfy these conditions. Without loss of generality, we assume that $X \geq Y$.

When one of the numbers is exactly known, say $Y = 0$, then we have an instance of the ***partial approximate common divisor problem (PACDP)***. Without loss of generality, we assume that $a_0 < b_0$. Notice that May's generalization of Coppersmith's univariate method, Theorem 3.1, can be used to solve the PACDP. For each solution $x_0$ that is found, the common divisor $d$ can be computed as $d = \gcd(f_d(x_0), N)$ provided that $f_d(x_0) \not\equiv 0 \pmod{N}$.

The ACDP was generalized to more than two numbers by Proos [77] in 2003. Rather then outlining the method of solution for the ACDP, we outline Proos' method for the generalized version. An instance of the ***general partial approximate common divisor problem (GPACDP)*** consists of $\ell + 1 \geq 2$ inputs $a_0, a_1, \ldots, a_\ell$ and bounds $X_1, \ldots, X_\ell$ and $D$ such that there exists integers $|y_i| < X_i$ and $d > D$ such that $d|a_0$ and $d|(a_i + y_i)$ for all $1 \leq i \leq \ell$. The problem is to find all $y_i, \ldots, y_\ell$, and $d$ that satisfy these conditions. Without loss of generality, we assume that $X_1 \geq \cdots \geq X_\ell$. Also, let $D < a_0$ since otherwise there can be no solution.

To solve the GPACDP, we proceed as follows. Let $\overline{y} = (y_1, \ldots, y_\ell) \in \mathbb{Z}^\ell$ and $d \in \mathbb{Z}$ be a solution of a given GPACDP. Choose two integers $u \geq 1$ and $h \geq u$. Letting $\gamma_0, \ldots, \gamma_\ell$, and $\sigma$ be non-negative integers, define the polynomial

$$q_{\gamma_0,\ldots,\gamma_\ell,\sigma}(\overline{x}) = a_0^{\gamma_0}(a_1 + x_1)^{\gamma_1}(a_2 + x_2)^{\gamma_2} \cdots (a_\ell + x_\ell)^{\gamma_\ell},$$

where $\gamma_0 + \gamma_1 + \cdots + \gamma_\ell = \sigma$. Notice that when $\sigma \geq u$ we have

$$q_{\gamma_0,\ldots,\gamma_\ell,\sigma}(\overline{y}) \equiv 0 \pmod{d^u}.$$

Next, construct the lattice $L$ with basis matrix $M$ whose rows are the coefficient vectors of the $\binom{h+\ell}{\ell}$ polynomials

$$\left.\begin{array}{c} q_{\gamma_0,\gamma_1,\ldots,\gamma_\ell,u}(x_1X_1, \ldots, x_\ell X_\ell) \\ q_{0,\gamma_1,\ldots,\gamma_\ell,u+1}(x_1X_1, \ldots, x_\ell X_\ell) \\ \vdots \\ q_{0,\gamma_1,\ldots,\gamma_\ell,h}(x_1X_1, \ldots, x_\ell X_\ell) \end{array}\right\} \quad \gamma_0 = 0$$

for all possible combinations of the $\gamma_i$. For example, when $\ell = 2$, $u = 2$ and $h = 3$ the basis matrix is given by

$$\left[\begin{array}{cccccccccc}
a_0^2 & & & & & & & & & \\
\hline
a_0a_1 & a_0X_1 & & & & & & & & \\
a_0a_2 & & a_0X_2 & & & & & & & \\
\hline
a_1a_2 & a_2X_1 & a_1X_2 & X_1X_2 & & & & & & \\
a_1^2 & 2a_1X_1 & & & X_1^2 & & & & & \\
a_2^2 & & 2a_2X_2 & & & X_2^2 & & & & \\
\hline
a_1^2a_2 & 2a_1a_2X_1 & a_1^2X_2 & 2a_1X_1X_2 & a_2X_1^2 & & X_1^2X_2 & & & \\
a_1a_2^2 & a_2^2X_1 & 2a_1a_2X_2 & 2a_2X_1X_2 & & a_1X_2^2 & & X_1X_2^2 & & \\
a_1^3 & 3a_1^2X_1 & & & 3a_1X_1^2 & & & & X_1^3 & \\
a_2^3 & & 3a_2^2X_2 & & & 3a_2X_2^2 & & & & X_2^3
\end{array}\right],$$

where the horizontal lines separate the 4 different values of $\gamma_1 + \gamma_2$.

With a proper ordering of the polynomials the basis matrix can always be constructed to be triangular with non-zero diagonal elements. Thus, the lattice is full rank with dimension $m = \binom{h+\ell}{\ell}$. Also, a simple calculation shows that the volume is given by

$$d(L) = a_0^{\binom{u+\ell}{\ell+1}} (X_1 \cdots X_\ell)^{\binom{h+\ell}{\ell+1}}.$$

Using the LLL-algorithm, we can obtain $\ell$ linearly independent vectors that correspond to $\ell$ linearly independent polynomials $p_i(\overline{x})$ satisfying

$$\|p_1(x_1X_1, \ldots, x_\ell X_\ell)\| \leq \cdots \leq \|p_\ell(x_1X_1, \ldots, x_\ell X_\ell)\|,$$

and

$$\|p_\ell(x_1X_1, \ldots, x_\ell X_\ell)\| \leq 2^{\frac{m+\ell-2}{4}} d(L)^{\frac{1}{m-\ell+1}}.$$

A sufficient condition to apply the integer equation property, Lemma 3.2, on each of these $\ell$ polynomials is given by

$$2^{\frac{m+\ell-2}{4}} d(L)^{\frac{1}{m-\ell+1}} < d^u/\sqrt{m}.$$

The terms $2^{\frac{m+\ell-2}{4}}$ and $\sqrt{m}$ are assumed to be small compared to $d^u$ and are neglected. Since $X_1 \geq X_2 \geq \cdots \geq X_\ell$ and $D \leq d$ the above inequality, neglecting the small terms, is then satisfied if

$$\left( a_0^{\binom{u+\ell}{\ell+1}} X_1^{\binom{h+\ell}{\ell+1}\ell} \right)^{\frac{1}{\binom{h+\ell}{\ell}-\ell+1}} \leq D^u.$$

Letting $D = a_0^\alpha$ and $X_1 = a_0^\beta$ this is equivalent to

$$\binom{u+\ell}{\ell+1} + \beta\ell\binom{h+\ell}{\ell+1} < \alpha u \left[ \binom{h+\ell}{\ell} - l + 1 \right], \tag{31}$$

which is the enabling equation for the GPACDP using this method. In terms of only $\alpha$ or only $\beta$, we see that this enabling equation can be written as

$$\alpha > \frac{\binom{u+\ell}{\ell+1} + \beta\ell\binom{h+\ell}{\ell+1}}{u\left[\binom{h+\ell}{\ell} - l + 1\right]} \quad \text{or} \quad \beta < \frac{\alpha u\left[\binom{h+\ell}{\ell} - l + 1\right] - \binom{u+\ell}{\ell+1}}{\ell\binom{h+\ell}{\ell+1}}.$$

Proos then goes on to show how to find bounds on $\beta$ for various values of $\alpha$, $\ell$, and lattice dimension bounds [77, Section 6.2.1].

# 4 Non-Linear Equations II (Applications)

## 4.1 Factoring

The original application of Coppersmith's method for bivariate integer equations [20, 22] was factoring with partial information about the factors. The main result is summarized in the following.

**Theorem 4.1 (Factoring $N = pq$).** *Let $N = pq$ be an $n$-bit product of two primes $p$ and $q$ that are roughly the same size. Given the $n/4$ most significant or least significant bits of one of the primes, $N$ can be factored efficiently.*

Let $p = p_1 2^{n/4} + p_0$ and $q = q_1 2^{n/4} + q_0$ where $0 \leq p_1, p_0, q_1, q_0 < 2^{n/4}$. That is, $p_1$ represents the $n/4$ most significant bits of $p$ and $p_0$ represents the $n/4$ least significant bits of $p$, where we assume that $p$ is an $n/2$-bit integer. A similar statement holds for $q$. Consider the two polynomials $p_0(x, y), p_1(x, y) \in \mathbb{Z}[x, y]$ defined by

$$\begin{aligned} f_0(x,y) &= \frac{1}{2^{n/4}}\Big((x2^{n/4} + p_0)(y2^{n/4} + q_0) - N\Big) \\ &= xy2^{n/4} + q_0 x + p_0 y + \frac{1}{2^{n/4}}\left(p_0 q_0 - N\right), \quad \text{and} \end{aligned}$$

$$\begin{aligned} f_1(x,y) &= (p_1 2^{n/4} + x)(q_1 2^{n/4} + y) - N \\ &= xy + q_0 2^{n/4} x + p_0 2^{n/4} y + p_0 q_0 2^{n/2} - N, \end{aligned}$$

Each polynomial has maximum degree $d = 1$ in both variables. Also, the constant term of $f_0(x, y)$ is an integer by the definition of $p_0$ and $q_0$. By construction, the polynomials satisfy

$$f_0(p_1, q_1) = 0 \quad \text{and} \quad f_1(p_0, q_0) = 0.$$

Using Coppersmith's method for bivariate integer equations we can find these roots. That is, given $f_0(x, y)$ we can find $(p_1, q_1)$ and given $f_1(x, y)$ we can find $(p_0, q_0)$. Defining the bounds $X = Y = N^{1/4-\epsilon}$ we see that

$$\|f_0(xX, yY)\|_\infty \approx \|f_1(xX, yY)\|_\infty \approx N^{3/4},$$

so letting $W = N^{3/4}$ we have

$$XY < N^{1/2-2\epsilon} \approx W^{3/2-3\epsilon} = W^{3/(2d)-3\epsilon}.$$

Doing an exhaustive search on only a couple of the most significant bits of $p_0$ when solving $f_1(x, y) = 0$ or the least significant bits of $p_1$ when solving $f_0(x, y) = 0$ ensures that all of the necessary conditions for Coppersmith's method are satisfied.

Now, given the $n/4$ least significant bits of $p$ (or $q$) we know the values of $p_0$ and $q_0$ since $p_0 q_0 = N \bmod 2^{n/4}$. Solving $f_0(x, y) = 0$ for $(x, y) = (p_1, q_1)$ allows for all of $p$ and $q$ to be computed. Similarly, given the $n/4$ most significant bits of $p$ (or $q$) we know $p_1$ and $q_1$ so solving $f_1(x, y) = 0$ for $(x, y) = (p_0, q_0)$ allows $p$ and $q$ to be computed.

It was later demonstrated by Howgrave-Graham [43], that the result for known most significant bits of $p$ can be obtained more simply, both theoretically and practically, using Coppersmith's method for univariate modular equations. The result is a direct application of Theorem 3.1 with the function

$$f_p(x) = (p_1 2^{n/4} + x),$$

which has root $p_0$ modulo $p$. This result was generalized to factoring integers $N = p^r q$ by Howgrave-Graham [43] and Boneh, Durfee & Howgrave-Graham [12]. For the case of $p \approx q$, they show that $N$ can be factored with knowledge of the $n/(r + 1)$ most significant bits of $p$, which recovers Coppersmith's result from above when applied with $r = 1$. This result follows from Theorem 3.1 using the function

$$f_{p^r}(x) = (p_1 2^{n/4} + x)^r,$$

which has root $p_0$ modulo $p^r$. A further generalization by May [58], uses the function

$$f_{(kp)^r}(x) = (\widetilde{p} + x)^r$$

to obtain the following result.

**Theorem 4.2 (Factoring $N = p^r q$).** *Let $N = p^r q$ where $p$ and $q$ are primes of roughly the same size. Let $k$ be an (unknown) integer that is not a multiple of $p^{r-1}q$. Given an integer $\widetilde{p}$ such that $|kp - \widetilde{p}| < N^{\frac{r}{(r+1)^2}}$, $N$ can be factored.*

That is, given a multiple of $p$ up to a small correction term we can recover $p$ and hence the factorization of $N$.

## 4.2 RSA

Many applications of Coppersmith's methods in cryptography involve attacking special instances of RSA and RSA-like cryptosystems. (see Appendix A.5 for a review of the RSA cryptosystem). Throughout this section,

unless otherwise stated, we use following notation. Let $N = pq$ be an $n$-bit RSA modulus, let $\langle e, N\rangle$ be the public key, and let $\langle d, N\rangle$ be the private key. The public/private exponents are chosen so that $ed \equiv 1 \pmod{\phi(N)}$.

### 4.2.1 Low Public Exponent

The following attacks for small public exponent RSA can only be used to recover a given plaintext. They do not expose the private key. Also, with a proper padding scheme, such as OAEP, these attacks are not relevant. We include them to show how Coppersmith's method has been applied in various settings.

**Stereotyped Messages** When most of the plaintext is fixed, or stereotyped, it is possible to recover the unknown part if both it and the public exponent are sufficiently small. For public exponent $e$, let $m = B+y$ be a plaintext where $B$ is a known fixed part of the message and let $c = m^e \bmod N$ be its corresponding ciphertext. Consider the degree $d = e$ univariate polynomial

$$f_N(x) = (B + x)^e - c,$$

which has root $x = y$ modulo $N$. Using Coppersmith's method for univariate modular polynomials, Theorem 3.1, we can recover $y$ provided that $|y| < N^{1/e-\epsilon}$. The bound can be increased to $N^{1/e}$ by doing an exhaustive search on a few bits of the solution. In particular, when $e = 3$ the full plaintext can be recovered if at least 2/3 of it is already known.

**Random Padding With Known Related Messages** When two plaintext messages $m$ and $m'$ related by the affine relation $m' = m + r$ are encrypted with the same public key, it is possible to recover $m$, given only the ciphertexts, if the public exponent and $r$ are sufficiently small.

Let $c = m^e \pmod N$ and $c' = (m + r)^e \pmod N$ be the ciphertexts of $m$ and $m'$, respectively. When $e = 3$ and $r$ is known, Franklin & Reiter[30] and Coppersmith, Franklin, Patarin & Reiter [23] show that the plaintext $m$ can be computed with the relation

$$m = \frac{r(c' + 2c - r^3)}{c' - c + 2r^3} = \frac{r(3m^3 + 3m^2r + 3mr^2)}{3m^2r + 3mr^2 + 4r^3} \pmod N.$$

When $e = 3$ and $r$ is unknown, Coppersmith [22] shows that the plaintext $m$ can still be recovered provided that $r$ is sufficiently small. In this case,

computing the resultant of $m^3 - c$ and $(m+r)^3 - c'$ with respect to $m$ yields the following degree $d = 9$ univariate modular polynomial equation

$$\begin{aligned}&\mathrm{Res}_m\left(m^3 - c, (m+r)^3 - c'\right) = \\ &\quad r^9 + (3c - 3c')r^6 + (3c^2 + 21cc' + 3c'^2)r^3 + (c - c')^3 \equiv 0 \pmod{N}.\end{aligned}$$

Using Coppersmith's method for univariate modular polynomials, Theorem 3.1, $r$ can be recovered provided $|r| < N^{1/9}$. Once $r$ is known, the method from above can be used to recover $m$. It is possible to generalize this further, but the size of $r$ becomes prohibitively small when $e > 3$ and the relation between $m$ and $m'$ becomes more complicated.

**Håstad's Broadcast Attack** One of the first non-linear applications of lattice basis reduction in cryptography was the so-called ***Håstad broadcast attack***. Håstad[38, 39] showed that knowledge of a sufficient number of ciphertexts corresponding to plaintexts that are all linearly related can be used to recover the plaintexts if each ciphertext was encrypted with a different public key. The number of ciphertexts needed was quadratic in the size of the largest public exponent. Using Coppersmith's method the number of ciphertexts needed is reduced to the largest public exponent.

Let $c_i = m_i^{e_i} \pmod{N_i}$ be the ciphertext of the plaintext $m_i$, where the $N_i$ are pairwise relatively prime and the plaintext are all of the form

$$m_i = \alpha_i m + \beta_i \pmod{N_i},$$

where $\alpha_i$ and $\beta_i$ are known values. Consider the monic polynomials $f_i(m) \in \mathbb{Z}[m]$ defined by

$$f_i(x) = \alpha_i^{-e_i}(\alpha_1 x + \beta_i)^{e_1} - c_i,$$

each with root $m$ modulo $N_i$. Using the Chinese remainder theorem, a monic degree $e = \max_i(e_i)$ polynomial equation can be derived of the form

$$F(m) \equiv 0 \pmod{\prod_i N_i}.$$

Using Coppersmith's method for univariate modular polynomials, we $m$ can be recovered provided that $|m| < N^{1/e}$. Bleichenbacher [5] further shows that when the public keys $e_i$ are different it is possible to improve this result slightly but the improvements depend on the exact values of the $e_i$ used.

### 4.2.2 Low Private Exponent

Here we outline the various attacks against low private exponent RSA.

**Boneh & Durfee: Small Inverse Attack** The small inverse problem was defined and solved by Boneh & Durfee [8, 9] in order to attack small private exponent RSA. Their attack was the first polynomial time improvement on Wiener's continued fraction attack [91] which works for private exponents $|d| < N^{0.25}$.

Let $e = N^{\alpha}$, $d < N^{\delta}$, $\phi(N) = N - \Lambda$, and recall the public/private key equation $ed - k\phi(N) = 1$, where $k$ is some positive integer. This can be written as

$$ed - k(N - \Lambda) = 1,$$

where $k$, $\Lambda$, and $d$ are the only unknowns. Reducing this equation modulo $e$ yields

$$-k(N - \lambda) \equiv 1 \pmod{e},$$

which has one less unknown. When the private exponent is small it is expected that the public exponent will be large so it is assumed that $\alpha \approx 1$. In this case the remaining unknowns satisfy

$$|k| < \frac{ed}{\phi(N)} \leq e^{1+\frac{\delta-1}{\alpha}} \approx e^{\delta},$$

and

$$|\Lambda| < 3N^{1/2} = 3e^{1/2\alpha} \approx 3e^{1/2}.$$

Defining $X = e^{\delta}$ and $Y = 3e^{1/2}$ we have an instance of the small inverse problem with $A = N$ and $B = e$ and bounds $X = e^{\delta}$ and $Y = 3e^{1/2}$. That is, we wish to find all solutions $(x_0, y_0)$ of

$$f_e(x, y) = x(N + y) - 1 \equiv 0 \pmod{e},$$

such that $|x_0| < X$ and $|y_0| < Y$. By construction, $(x_0, y_0) = (-k, -\Lambda)$ is a solution of this modular equation. Thus, solving the small inverse problem for this instance will reveal $\Lambda$ which immediately gives $\phi(N) = N - \Lambda$. Knowledge of $\phi(N)$ leads to a total break of RSA as one can directly compute the private exponent $d = e^{-1} \bmod \phi(N)$.

To solve the small inverse problem, just as in Section 3.1.4, Boneh & Durfee look for two algebraically independent polynomials that are root equivalent to $f_e(x, y)$ modulo $e$ and which also satisfy the integer equation property. These polynomials are found using lattice basis reduction. For some positive integer $m$ define the $x$- and $y$-shift polynomials of $f_e(x, y)$ by

$$g_{i,k}(x, y) = x^i f_e^k(x, y) e^{m-k} \quad \text{and} \quad h_{j,k}(x, y) = y^j f_e^k(x, y) e^{m-k},$$

respectively. Consider the lattice $L$ generated by the basis matrix $M$ whose rows are the coefficient vectors of $g_{i,k}(xX, yY)$ and $h_{k,k}(xX, yY)$ for certain values of $i, j$ and $k$. Notice that each element of $L$ corresponds to a polynomial that is root equivalent to $f_e(x, y)$ modulo $e$. Looking at the polynomials corresponding to the two smallest elements of $L$ found by the LLL-algorithm an enabling equation on $X$ and $Y$ can be derived. The lattice, and hence the bounds, depend on the particular choice of $i$,$j$, and $k$ used to define the basis matrix.

When the basis matrix consists of $x$-shifted polynomials only, the enabling equation is equivalent to $|d| < N^{0.25}$ which reproduces Wiener's result. It was observed by Blömer & May[6], however, that the two polynomials obtained in this case are always algebraically dependent.

When the basis matrix is constructed with

$$\begin{aligned} &g_{i,k}(xX, yY) \quad \text{for all} \quad 0 \le i \le m-k \ , \ 0 \le k \le m, \ \text{and} \\ &h_{j,k}(xX, yY) \quad \text{for all} \quad 1 \le j \le t \ , \ 0 \le k \le m. \end{aligned}$$

the enabling equation is as given in Section 3.1.4. Optimizing the choice for $m$ and $t$, Boneh & Durfee derive the enabling equation $|d| < N^{0.284}$. When the assumption that $\alpha \approx 1$ is relaxed the enabling equation becomes

$$|\delta| < \frac{7}{6} - \frac{1}{3}\sqrt{1+6\alpha},$$

so the attack becomes stronger with decreasing $\alpha$. This result was extended to multi-prime RSA by Hinek, Low & Teske [41], who showed that the enabling equation for $r$-prime RSA is given by

$$|\delta| < \frac{4}{3} - \frac{1}{3r} - \frac{2}{3r}\sqrt{(r-1)(r-1+2r\alpha)}.$$

Notice that the the bound on $d < N^{\delta}$ decreases with each additional prime added to the modulus.

Boneh & Durfee go on to show that when $\alpha \approx 1$ the bound $N^{0.284}$ can be improved to $N^{0.292}$ by considering a sub-lattice of $L$. Making the observation that certain rows of the basis matrix $M$ contribute more to the volume than others they look for small elements in the sub-lattice generated by the basis matrix with these rows removed. This results in a difficult computation of the enabling equation as the basis matrix is no longer square and the volume of the lattice is needed[2]. Using the concept of ***geometrically progressive***

[2]As Howgrave-Graham[44] puts it working out the determinant of a lattice given by a non-square matrix can be a major piece of work.

***matrices*** they are able to compute a bound on the lattice volume. The details can be found in [9]. This result was extended to multi-prime-RSA by Ciet, Koeunne, Laguillaumie & Quisquater [17], who show

$$|\delta| < 1 - \frac{1}{r}\sqrt{r^2 - r}.$$

Again, the bound on private exponent decreases with each additional prime added to the modulus.

Another improvement on the $N^{0.284}$ bound for $\alpha \approx 1$ was made by Blömer & May[6], who increase the bound to $N^{0.290}$. While this improvement is not as great as the previous one, the analysis is much simpler. Like Boneh & Durfee, they remove certain rows in the basis matrix $M$ that substantially contribute to the lattice volume. They also remove corresponding columns to ensure that the basis matrix remains square however, which allows for a simple volume calculation. They go on to show that the volume of the new matrix is the same as the volume of the original lattice up to a small correction term and that the size of small vectors found in the new lattice by the LLL-algorithm are the same size as those found in the original lattice up to a small correction term. The details can be found in [6]. This result was extended to multi-prime RSA by Hinek, Low & Teske [41], who show that, for $r$-prime RSA,

$$|\delta| < \frac{11}{5} - \frac{6}{5r} - \frac{3}{5}\alpha + \frac{2}{5r}\sqrt{\alpha 4(\alpha r + r - 1) + 4(r-1)^2}.$$

Besides being simpler to analyze, Blömer & May's method has the advantage that in practice smaller lattices can be used.

Wiener's continued fraction attack[91] is provably defeated if unusually large public exponents are used. In particular the attack does not work, regardless of the size of $d$, whenever $e > N^{1.5}$. Similar results can be found for the lattice attacks discussed above also. For example, for Boneh & Durfee's attack with bound $N^{0.284}$, it has been shown that whenever the public exponent is chosen so that $e > N^{1.875}$, the enabling equation cannot be satisfied for any private exponent. Unlike in Wiener's attack though, this does not mean that the attack cannot work. Since the enabling equation only provides a sufficient condition, we can only say that there is no value of $d$ for which the attack is guaranteed to work.

**Durfee & Nguyen: Unbalanced Primes** In 1999, Sun, Yang & Laih [86] proposed three methods for constructing instances of small private exponent RSA using unbalanced primes which thwarted the small private exponent

attacks of Wiener and Boneh & Durfee. In 2000, Durfee & Nguyen [28] presented heuristic attacks that defeated two of the proposed methods using Coppersmith's method with a trivariate modular polynomial. Looking at the public/private key equation again

$$ed = 1 + k(N + 1 - p - q),$$

they consider the polynomial $f_e(x, y, z) \in \mathbb{Z}[x, y, z]$ defined by

$$f_e(x, y, z) = x(S + y + z) - 1,$$

where $S = N+1$. Notice that $f_e(x_0, y_0, z_0) \equiv 0 \pmod{e}$ when $x_0 = k$, $y_0 = p$ and $z_0 = q$, so they look for all $(x_0, y_0, z_0)$ such that $f(x_0, y_0, z_0) \equiv 0 \pmod{e}$ where $|x_0| < X \approx ed/N$, $|y_0| < Y \approx p$ and $|z_0| < Z \approx q$. The bounds $X, Y$, and $Z$ are determined by the exact parameters of the proposed RSA instance. For integers $m \geq 1$, $a \geq 0$, and $t \geq 1$ they define the $\omega = (m+1)(m+t+1)$ polynomials

$$g_{k,i,b}(x, y, z) = e^{m-k} x^i y^a z^b f^k(x, y, z) \quad \text{for} \quad \begin{cases} k = 0, \ldots, m-1 \\ i = 1, \ldots, m-k \\ b = 0, 1 \end{cases},$$

and

$$h_{k,j}(x, y, z) = e^{m-k} y^{a+j} f^k(x, y, z) \quad \text{for} \quad \begin{cases} k = 0, \ldots, m \\ j = 0, \ldots, t, \end{cases}.$$

Consider the lattice $L$ generated by the basis matrix $M$ whose rows are the coefficient vectors of $g_{k,i,b}(xX, yY, zZ)$ and $h_{k,j}(xX, yY, zZ)$ for the above $\omega$ polynomials. Using lattice reduction they look for two elements in $L$ which correspond to polynomials that satisfy the general integer equation property. Denote these polynomials by $H_1(x, y, z)$ and $H_2(x, y, z)$. Removing the variable $x$ with a resultant computation and using $N = yz$ yields a single univariate polynomial with root $p$ over the integers. Thus, if $H_1$ and $H_2$ are not strongly algebraically dependent then $N$ can be factored which immediately reveals the private key $(N, d)$.

**May: Small CRT-Exponent** The Chinese remainder theorem can be used to speed up RSA decryption. When the primes in the modulus are not balanced, one can further speed up decryption by using a private exponent that is small modulo $p - 1$ where $p$ is the larger of the two primes. This situation is considered by May [57], and is referred to as small CRT-exponent

for unbalanced RSA. May presents three attacks on small CRT-exponent RSA. The first two differ from other methods in this section, as they are initially linear problems and the results are provable.

For this analysis, it is assumed that the public/private exponents are generated so that $ed \equiv 1 \pmod{\frac{(p-1)(q-1)}{2}}$ with $\gcd(p-1, \frac{q-1}{2}) = 1$. Since $(p-1)$ and $\frac{q-1}{2}$ are relatively prime, the exponents $e$ and $d$ must also satisfy

$$ed \equiv 1 \pmod{(p-1)} \qquad \text{and} \qquad ed \equiv 1 \pmod{\frac{q-1}{2}}.$$

Letting $d_p = d \bmod (p-1)$ and considering the first the above relations, May considers the equation

$$ed_p + k(p-1) = 1, \tag{32}$$

where $k$ is some (negative) integer. From this equation, May defines the polynomial $f_p(x, y) \in \mathbb{Z}[x, y]$ by

$$f_p(x, y) = ex - y,$$

which has root $(x_0, y_0) = (d_p, -k-1)$ modulo $p$. Letting $q = N^\beta$ and $d_p < N^\delta$, define the bounds bounds $X = N^\delta$ and $Y = N^{\beta+\delta}$ so that $|x_0| < X$ and $|y_0| < Y$. May then considers the 2-dimensional lattice $L$ generated by the rows of basis matrix

$$M_p = \begin{bmatrix} NX & \\ eX & -Y \end{bmatrix}.$$

Since the dimension is 2, Gauss' reduction algorithm finds a shortest vector in the lattice. This shortest vector corresponds to a polynomial $p(x, y)$ with root $(x_0, y_0) = (d_p, -k-1)$ modulo $p$. The following is an enabling equation on $\beta$ and $\delta$

$$3\beta + 2\delta \leq 1 - \log_N 4, \tag{33}$$

so that $p(x, y)$ also satisfies the integer equation property (i.e., $p(x_0, y_0) = 0$). Of course, this only gives a single bivariate equation. In order to recover the roots, May also shows that the polynomial $p(x, y)$ is always of the form

$$p(x, y) = (c_0 N + c_1 e)x - c_1 y,$$

where $|c_0| = k$ and $|c_1| = qd_p < N$. Computing $\gcd(c_1, N) = q$ reveals the factorization of $N$. From (33), we see that this method will work for unbalanced primes such that $q = N^\beta < N^{1/3} \approx N^{0.333}$.

May's second method is the natural extension of the first. Here, integer polynomial combinations of $f_p(x, y)$ are used to generate a lattice. For some positive integer $m$, May considers the polynomials defined by

$$g_{m,i,j}(x, y) = N^{\max(0,m-j)} x^i f_P^j(x, y).$$

Constructing a lattice $L$ with the coefficient vectors of $g_{m,i,j}(xX, yY)$ for certain values of $i$ and $j$ a short vector is found. This short vector corresponds to a polynomial $p(x, y)$ that has root $(x_0, y_0) = (d_p, -k - 1)$ modulo $p^m$. The following enabling equation is then derived:

$$3\beta - \beta^2 + 2\delta \leq 1 - \epsilon, \tag{34}$$

so that $p(x, y)$ also satisfies the integer equation property, where $\epsilon > 0$ is small for large $N$ and lattice dimension. As in the previous attack, $x_0$ and $y_0$ need to be recovered from the single bivariate polynomial $p(x, y)$. To this end, May shows that the polynomial $y_0 x - x_0 y$ divides $p(x, y)$. Thus, factoring $p(x, y)$ over $\mathbb{Z}[x, y]$ reveals $x_0$ and $y_0$. From (34), we see that this method will work for unbalanced primes such that

$$q = N^\beta < N^{\frac{3-\sqrt{5}}{2}} \approx N^{0.382}.$$

It is interesting that the previous two methods are provable in the sense that they do not rely on the algebraic independence of two polynomials found by the LLL-algorithm. A single bivariate integer polynomial is sufficient to recover the desired roots. May's third method does not possess this nice property. Two small lattice vectors will be required to recover the roots and so is only a heuristic method. Rearranging (32), $ed_p + k(p - 1) = 1$, and multiplying by $q$ yields

$$(k + 1)(N - q) - N = -ed_p q.$$

From this, May considers the polynomial

$$f_e(y, z) = y(N - z) - N,$$

which has a root $(y_0, z_0) = (k+1, p)$ modulo $e$. Define the bounds $Y = N^{\beta+\delta}$ and $Z = N^\beta$ so that $|y_0| < Y$ and $|z_0| < Z$. For some positive integer $m$, May defines the $y$- and $z$-shifted polynomials

$$g_{i,j}(y, z) = e^{m-i} y^j f_e^i(y, z), \text{ and}$$

$$h_{i,j}(y, z) = e^{m-i} x^j f_e^i(y, z).$$

May looks for two short vectors in the lattice spanned by the coefficient vectors of $g_{i,j}(yY, zZ)$ and $h_{i,j}(yY, zZ)$ for certain values of $i$ and $j$. These vectors correspond to two polynomials with root $(y_0, z_0) = (k+1, p)$ modulo $e^m$. The following enabling equation can be derived

$$\frac{5}{3}\beta + \frac{2}{3}\sqrt{3\beta - 5\beta^2} + \delta \leq 1 - \epsilon,$$

so that the two polynomials also satisfy the integer equation property, where $\epsilon > 0$ is small for large $N$ and lattice dimensions. Using resultants, $x_0$ and $y_0$ can be recovered. While this method is only a heuristic, it does allow for greater $d_p$ when $q = N^\beta < N^{0.23}$.

**May: $N = p^r q$** In 2003, May [58] presented two small private exponent attacks against RSA-like systems with modulus $N = p^r q$. The methods are direct applications of Theorems 3.1 and 4.2. Now, Euler's totient function when $N = p^r q$ is given by

$$\phi(N) = p^{r-1}(p-1)(q-1),$$

so the public/private key equation becomes

$$ed = 1 + kp^{r-1}(p-1)(q-1), \tag{35}$$

for some integer $k$. This will be the starting point for both attacks.

The first method uses the results for factoring $N = p^r q$, Theorem 4.2. Letting $E$ be the inverse of $e$ modulo $N$ (i.e., $E = e^{-1} \pmod N$) yields the equation $Ee = 1 + cN$ for some integer $c$. Multiplying (35) by $E$ and rearranging then gives

$$d - E = \left(Ekp^{r-2}(p-1)(q-1) - cp^{r-1}qd\right)p.$$

Thus, $E$ is a multiple of $p$ up to an additive error of $|d|$. Also, since the factor $Ekp^{r-2}(p-1)(q-1) - cp^{r-1}qd$ is not a multiple of $p^{r-1}q$, as shown by May, Theorem 4.2 can be directly applied. Thus, any private exponent $d$ satisfying

$$|d| < N^{\frac{1}{(r+1)^2}},$$

can be recovered.

The second method uses the generalization of Coppersmith's result for univariate modular equations, Theorem 3.1. Multiplying (35) by $E$ and rearranging we obtain

$$d - E = \left(Ek(p-1)(q-1) - cdpq\right)p^{r-1}.$$

This leads to the degree $d = 1$ univariate polynomial

$$f_{p^{r-1}}(x) = x - E$$

with the root $x_0 = d$ modulo $p^{r-1}$. Letting $b = p^{r-1}$ be the factor of $N$ (in Theorem 3.1), notice that it satisfies

$$b = p^{r-1} \geq \left(\frac{1}{2}N\right)^{\frac{r-1}{r+1}} \geq \frac{1}{2}N^{\frac{r-1}{r+1}},$$

so choosing $\beta = \frac{r-1}{r+1} - \frac{1}{\log N}$ and $c_N = 4$ (along with $d = 1$) one can apply Theorem 3.1 to $f_{p^{r-1}}(x)$. Since

$$4N^{\frac{\beta^2}{\delta}} = 4N^{\left(\frac{r-1}{r+1}\right)^2 - \frac{2(r-1)}{(r+1)\log N} + \frac{1}{\log^2 N}} \geq 4N^{\left(\frac{r-1}{r+1}\right)^2 - \frac{2}{\log N}} = N^{\left(\frac{r-1}{r+1}\right)^2},$$

any private exponent $d$ can be recovered provided that

$$|d| < N^{\left(\frac{r-1}{r+1}\right)^2}.$$

This second method is stronger than the first for all $r \geq 3$. When $N = p^2q$ however, the first method is stronger.

**Time-Line** In Table 2, we present a time-line of small private exponent attacks on RSA and RSA-like systems that use lattice basis reduction.

### 4.2.3 Partial Key-Exposure Attacks

A partial key-exposure attack is one in which some knowledge of the private key is known to the adversary. In all of the attacks discussed we only consider the cases when the adversary knows some number of the most significant bits or some number of the least significant bits of the private exponent $d$. For each of the attacks we state the main result and give a brief sketch of details leading to the results.

**Boneh-Durfee-Frankel** In 1998, Boneh, Durfee & Frankel [10] presented some partial key-exposure attacks on RSA (the current corrected version is [11]). Attacks with known least significant bits and most significant bits of the private exponent are presented. In the following, we will refer to Boneh, Durfee & Frankel as BDF.

Their main result for known least significant bits of $d$ is as follows.

| Year | Authors | | Effectiveness | Assumptions/Conditions | |
|---|---|---|---|---|---|
| 1990 | [91] | (0) | $\lvert d\rvert < N^{0.25}$ | non-lattice method | $\star$ |
| 1998 | [9] | (1) | $\lvert d\rvert < N^{0.285}$ | | |
| | | (2) | $\lvert d\rvert < N^{0.292}$ | | |
| 2000 | [28] | | specific to systems in [86] | unbalanced primes | |
| 2001 | [6] | (3) | $\lvert d\rvert < N^{0.290}$ | | |
| 2002 | [41] | (4) | $\lvert d\rvert < \frac{4}{3} - \frac{1}{3r} - \frac{2}{3r}\sqrt{4r^2 - 5r + 1}$ | $N = p_1 \cdots p_r$ , extension of (1) | |
| 2002 | [17] | (5) | $\lvert d\rvert < 1 - \frac{1}{r}\sqrt{r^2 - r}$ | $N = p_1 \cdots p_r$ , extension of (2) | |
| 2002 | [57] | | $\lvert d_q\rvert < N^{\delta}$, $p < N^{\beta}$ | unbalanced primes | |
| | | (6) | $3\beta + 2\delta \leq 1 - \log_N 4$ | $p < N^{0.333}$ | $\star$ |
| | | (7) | $3\beta - \beta^2 + 2\delta \leq 1 - \epsilon$ | $p < N^{0.382}$ | $\star$ |
| | | (8) | $\frac{5}{3}\beta + \frac{2}{3}\sqrt{3\beta - 5\beta^2} + \delta \leq 1 - \epsilon$ | $p < N^{0.23}$ | |
| 2004 | [58] | (9) | $\lvert d\rvert < N^{\frac{r}{(r+1)^2}}$ | $N = p^r q$ | $\star$ |
| | | (10) | $\lvert d\rvert < N^{\left(\frac{r-1}{r+1}\right)^2}$ | $N = p^r q$ | $\star$ |

Table 2: Time-Line of small private exponent attacks. The results are shown only for public exponent $e \approx N$. A star ($\star$) denotes that the attack is provable in the sense that it does not rely on finding two algebraically independent polynomials from the reduced lattice basis. Unless indicated, it as assumed that the primes in the modulus are balanced.

**Theorem 4.3 (BDF [11]).** *Let $N = pq$ be an $n$-bit RSA modulus. Let $1 \leq e, d \leq \phi(N)$ satisfy $ed \equiv 1 \pmod{\phi(N)}$. If $p \not\equiv q \pmod 4$ and $e \leq 2^{(n/4)-3}$ then there is an algorithm that, given the $n/4$ least significant bits of $d$, computes all of $d$ in time polynomial in $n$ and linear in $e \log e$.*

Let $d_0 = d \bmod 2^{n/4}$ be the $n/4$ least significant bits of $d$. Starting with the public/private key equation and expanding $\phi(N) = N - p - q + 1$ gives

$$ed = 1 + k(N - p - q + 1),$$

where $k$ is some positive integer. One of the factors of $N$ can be removed with the equation $N = pq$. Removing $q$ yields

$$ed = 1 + k(N - p - N/p + 1).$$

This led BDF to consider the univariate modular equation

$$kx^2 + (ed_0 - k(N+1) - 1)x + kN \equiv 0 \pmod{2^{n/4}}, \tag{36}$$

which has root $x_0 = p \bmod 2^{n/4}$. Since $k < \min(e, d)$ and $e$ is small an exhaustive search to find the correct $k$ value is feasible. For each candidate $1 \leq k' \leq e$ for $k$, BDF find all solutions of (36). Since one solution, $x_0$, is the $n/4$ least significant bits of $p$ the factoring result of Theorem 4.1 can be used to find $p$ and $q$. Testing all solutions of (36) for each $k'$ will eventually reveal the factorization. When the conditions in the theorem are met the attack can be mounted in polynomial time. The details can be found in [11]. An important condition in the result is that $p \not\equiv q \pmod 4$, which means that $p$ and $q$ cannot have more than two common least significant bits. This is because it was shown by Steinfeld & Zheng [85] that the runtime of Boneh, Durfee & Frankel's method is exponential in the number of common least significant bits of the primes $p$ and $q$.

Their main results for known most significant bits of $d$ are as follows.

**Theorem 4.4 (BDF [11]).** *Let $N = pq$ be an $n$-bit RSA modulus. Let $1 \leq e, d \leq \phi(N)$ satisfy $ed \equiv 1 \pmod{\phi(N)}$*

1. *Suppose $e$ is prime in the range $\{2^t, \ldots, 2^{t+1}\}$ with $n/4 \leq t \leq n/2$. Given the $t$ most significant bits of $d$, there is an algorithm that computes all of $d$ in time polynomial in $n$.*

2. *More generally, suppose $e \in \{2^t, \ldots, s^{t+1}\}$ is the product of at most $r$ distinct primes with $n/4 \leq t \leq n/2$. Given the factorization of $e$ and the $t$ most significant bits of $d$, there is an algorithm that computes all of $d$ in time polynomial in $n$ and $2^r$.*

We outline the method for prime public exponent $e$. The method for known factorization of $e$ follows directly. Given the $t$ most significant bits of $d$, BDF show that they can compute $k$ to within a small additive constant (40). For each candidate $k'$ of $k$ they first compute

$$s' = N + 1 - k'^{-1} \pmod{e}.$$

When $k' = k$ this satisfies $s' = p + q \bmod e$. Since $\gcd(k, e) = 1$ if $s'$ does not exists the current $k'$ is rejected. Next they find a root $p'$ of the modular equation

$$x^2 - s'x + N \equiv 0 \pmod{e}.$$

This can be done efficiently since $e$ is prime. When $k' = k$ one root will be $p \bmod e$, and since $e > N^{n/4}$, this reveals the $n/4$ least significant bits $p$.

The factoring result of Theorem 4.1 will reveal both $p$ and $q$. Once $p$ and $q$ are known the entire private key $d = e^{-1} \bmod (p-1)(q-1)$ is computed. The details can be found in [11].

**Blömer-May** In 2003, Blömer & May [7] presented new partial key-exposure attacks on RSA. In addition to attacks with known least and most significant bits of $d$ they also present attacks with known most and least significant bits $d_p$ where $d_p = d \bmod p-1$ is needed to use the Chinese remainder theorem for decryption.

The main results for attacking RSA with partial knowledge of $d_p$ are given below.

**Theorem 4.5 (BM [7]).** *Let $(N, e)$ be an RSA public key with $N = pq$, public exponent $e = N^\alpha$ and private exponent $d$. Let $d_p = d \bmod (p-1)$.*

1. *Given $d_0$, $M$ such that $\alpha \in [1, poly(\log N)]$, $d_0 \equiv d \pmod M$ and $M \geq N^{1/4}$, or*
2. *Given $\widetilde{d_p}$ such that $\alpha \in [0, \frac{1}{4}]$ and $|d_p - \widetilde{p_d}| < N^{1/4-\alpha}$,*

*then $N$ can be factored in polynomial time.*

Both results make use of the factoring result, Theorem 4.2, with $r = 1$ and begin with the public/private relation $ed \equiv 1 \pmod{(p-1)}$. Converting this to an equation gives

$$ed_p = 1 + k(p-1), \tag{37}$$

where $k$ is some positive integer.

For the known least significant bits case write $d_p = p_1 M + d_0$, where $d_1 < \frac{d_p}{M} < \frac{p}{M} \leq N^{1/4}$, so that (37) becomes

$$ed_0 + k - 1 = kp - eMd_1.$$

Multiply the above equation by the inverse of $eM$ modulo $N$, denoted by $E$, to obtain

$$E(ed_0 + k - 1) + d_1 = (Ek - cqd_1)p,$$

where $c$ is some integer that satisfies $EeM = 1 + cN$. So, $E(ed_0 + k - 1)$ is a multiple of $p$ up to an error term of $d_1 < N^{1/4}$ (which is the limiting bound on the factoring method, Theorem 4.2, when $r = 1$). Since $k$ is not known, an exhaustive search is required.

For the known most significant bits case, Blömer & May compute $\widetilde{p} = e\widetilde{d_p} - 1$ which satisfies

$$|\widetilde{p} - k\widetilde{d_p}| = |e(\widetilde{d_p} - d_p) - k| \leq N^{1/4} + N^{\alpha} \leq 2N^{1/4}.$$

Applying the factoring method of Theorem 4.2 to $\widetilde{p} + N^{1/4}$ and $\widetilde{p} + N^{1/4}$ will yield the factorization of $N$.

Blömer & May's result for known most significant bits of $d$ is as follows.

**Theorem 4.6 (BM [7]).** *For every $\epsilon > 0$ there exists an integer $N_0$ such that for every $N > N_0$ the following holds: Let $(N, e)$ be an RSA public key, where $\alpha = \log_N(e)$ is in the range $[\frac{1}{2}, \frac{\sqrt{6}-1}{2}]$. Given an approximation $\widetilde{d}$ of $d$ with*

$$|d - \widetilde{d}| \leq N^{\frac{1}{8}\left(5 - 2\alpha - \sqrt{36\alpha^2 + 12\alpha - 15}\right) - \epsilon},$$

*then $N$ can be factored in polynomial time.*

Since the most significant bits of $d$ are known, Blömer & May compute an approximation $\widetilde{k} = (e\widetilde{d} - 1)/(N + 1)$ of $k$ such that

$$|\widetilde{k} - k| = \left| \frac{ed - 1}{\phi(N)} - \frac{e\widetilde{d} - 1}{N + 1} \right| \leq \frac{e}{\phi(N)} \left( N^{\delta} + 3\widetilde{d}N^{-\frac{1}{2}} \right) \leq 4N^{\alpha - \frac{1}{2}},$$

where $d < N^{\delta}$ when $\widetilde{d}N^{-1/2} \geq N^{\delta}$. Writing $d_0 = d - \widetilde{d}$, $k_0 = k - \widetilde{k}$, and expanding $\phi(N) = N - \Lambda$, the public/private key equation $ed = 1 + k\phi(N)$ can be written as

$$ed_0 + (\widetilde{k} + k_0)\Lambda + e\widetilde{d} - 1 = (\widetilde{k} + k_0)N.$$

This leads to the polynomial $f_N(x, y, z) \in \mathbb{Z}[x, y, x]$ defined by

$$f_N(x, y, z) = ex + (\widetilde{k} + y)z + e\widetilde{d} - 1,$$

which has the root $(x_0, y_0, z_0) = (d_0, k_0, \Lambda)$ modulo $N$. Defining the bounds $X = N^{\delta}$, $Y = 4N^{\alpha - \frac{1}{2}}$, and $Z = 3N^{\frac{1}{2}}$, the desired solution satisfies $|x_0| < X$, $|y_0| < Y$, and $|z_0| < Z$. For some fixed integers $m$ and $t$, Blömer & May define the following polynomials

$$g_{i,j,k}(x, y, z) = x^{j-k} z^k N^i f_N^{m-1}(x, y, z) \quad \text{for} \begin{cases} 0 \leq i \leq m \\ 0 \leq j \leq i \\ 0 \leq k \leq j \end{cases}, \text{and}$$

$$h_{i,j,k}(x, y, z) = x^j y^k N^i f_N^{m-1}(x, y, z) \quad \text{for} \begin{cases} 0 \leq i \leq m \\ 0 \leq j \leq i \\ 1 \leq k \leq t \end{cases}.$$

Using lattice basis reduction, they find three polynomials that are integer linear combinations of the $g_{i,j,k}(x,y,z)$ and $h_{i,j,k}(x,y,z)$, all having root $(x_0,y_0,z_0) = (d_0,k_0,\Lambda)$ modulo $N^m$. The enabling equation so that the equations also satisfy the integer equation property is given by the condition in the theorem.

Blömer & May present two attacks when the least significant bits of $d$ are known. The first result, which is provable, is summarized in the following.

**Theorem 4.7 (BM [7]).** *Let $N$ be an RSA modulus and let $0 \leq \alpha, \epsilon \leq \frac{1}{2}$. For all but a $\mathcal{O}(\frac{1}{N^\epsilon})$-fraction of the public exponents $e$ in the range $[3, N^\alpha]$ the following holds: Let $d$ be the private exponent. Given $d_0$,$M$ satisfying $d_0 \equiv d \pmod{M}$ with*

$$N^{\alpha+\frac{1}{2}+\epsilon} \leq M \leq 2N^{\alpha+\frac{1}{2}},$$

*then $N$ can be factored in polynomial time.*

Writing $d = d_1 M + d_0$ where $d_0$ is the known least significant bits of $d$, the public/private key equation can be written as

$$ed_1M + k\Lambda - 1 + ed_0 = kN,$$

where $k$ is a positive integer. This motivates the polynomial $f_N(x,y) \in \mathbb{Z}[x,y]$ defined by

$$f_N(x,y) = eMx + y + ed_0,$$

which has root $(x_0,y_0) = (d_1, k\Lambda - 1)$ modulo $N$. Defining the bounds $X = N^{\frac{1}{2}-\alpha-\epsilon}$ and $Y = 3N^{\frac{1}{2}+\alpha}$, the desired solution satisfies $|x_0| = |d_1| < X$ and $|y_0| = |k\Lambda - 1| < Y$. Blömer & May then consider the lattice $L$ generated by

$$M = \begin{bmatrix} N & & \\ & NX & \\ ed_0 & eMX & Y \end{bmatrix},$$

where each element of $L$ corresponds to a polynomial that has root $(x_0,y_0)$ modulo $N$. They look for two small linearly independent elements $\boldsymbol{a}M$ and $\boldsymbol{b}M$ correspond to polynomials having root $(x_0,y_0)$ over the integers, where $\boldsymbol{a} = (a_0,a_1,a_2)$ and $\boldsymbol{b} = (b_0,b_1,b_2)$ are elements of $\mathbb{Z}^3$. When these polynomials satisfy the integer equation property, they result in the system of equations

$$\begin{aligned} a_0N + a_1Nx_0 + a_2f_N(x_0,y_0) &= 0 \quad \text{and} \\ b_0N + b_1Nx_0 + b_2f_N(x_0,y_0) &= 0. \end{aligned}$$

Since $f_N(x_0, y_0) = kN$, this becomes

$$\begin{aligned} a_1 x_0 + a_2 k &= -a_0 \quad \text{and} \\ b_1 x_0 + b_2 k &= -b_0, \end{aligned}$$

which can be solved for $x_0 = d_1$ and $k$ provided that $(a_1, a_2)$ and $(b_1, b_2)$ are linearly independent. In fact, they are linearly independent since the two vectors $(a_0, a_1, a_2)M$ and $(b_0, b_1, b_2)M$ (and hence $(a_0, a_1, a_2)$ and $(b_0, b_1, b_2)$ ) are linearly independent. Therefore, as long as the two polynomials satisfy the integer equation property, the system can be solved for $d_1$, which will reveal all of $d$. Blömer & May go on to show that, for all but a $\mathcal{O}(\frac{1}{N^\epsilon})$-fraction of the public exponents $e$ in the range $[3, N^\alpha]$, this method will work. The details can be found in [7].

The second attack with known least significant bits known is stronger than the previous attack, but it is only a heuristic.

**Theorem 4.8 (BM [7]).** *For every $\epsilon > 0$ there exists an integer $N_0$ such that for every $N > N_0$ the following holds: Let $(N, e)$ be an RSA public key, where $\alpha = \log_N(e) \leq \frac{7}{8}$. Let $d$ be the private exponent. Given $d_0$, $M$ satisfying $d_0 \equiv d \pmod{M}$ with*

$$M \geq N^{\frac{1}{6} + \frac{1}{3}\sqrt{1+6\alpha} + \epsilon},$$

*then $N$ can be factored in polynomial time.*

Writing $d = d_1 M + d_0$ and $\phi(N) = N - \Lambda$, the public/private key equation can be written as

$$k(N - \Lambda) - ed_0 + 1 = eMd_1,$$

where $k$ is some integer. This leads to the polynomial $f_{eM}(y, z) \in \mathbb{Z}[y, z]$ defined by

$$f_{eM}(y, z) = y(N - z) - ed_0 + 1,$$

that has root $(y_0, z_0) = (k, \Lambda)$ modulo $eM$. For fixed integers $m$ and $t$ define the polynomials

$$\begin{aligned} g_{i,j}(y, z) &= y^j (eM)^i f_{eM}^{m-1}(, y, z) \quad \text{for} \begin{cases} 0 \leq i \leq m \\ 0 \leq j \leq i \end{cases}, \text{and} \\ h_{i,j}(y, z) &= z^j (eM)^i f_{eM}^{m-1}(y, z) \quad \text{for} \begin{cases} 0 \leq i \leq m \\ 1 \leq j \leq t \end{cases} \end{aligned}$$

Using lattice basis reduction, Blömer & May look for two polynomials that are integer combinations of the $g_{i,j}(y, z)$ and $h_{i,j}(y, z)$ that have the root

$(y_0, z_0)$ over the integers. The enabling equation is given in the theorem. The details can be found in [7].

The previous three attacks have been extended to multi-prime RSA in 2004 by Hinek [40]. The results are shown in the partial key-exposure timeline given in Table 3.

**May: $N = p^r q$** In 2004, May[58] presented some partial key-exposure attacks on RSA-like systems with modulus $N = p^r q$. These attacks are direct consequences of his small private exponent attacks on RSA-like systems with modulus $N = p^r q$, as discussed in 4.2.2.

The main results when partial information about $d_p = d \bmod (p-1)$ is known are given in the following theorem.

**Theorem 4.9 (May [58]).** *Let $N = p^r q$, where $r \geq 1$ is a known constant and $p$ and $q$ are balanced primes. Let $e$ be the public exponent and let $d_p$ satisfy $ed_p \equiv 1 \pmod{(p-1)}$.*

1. *Given $d_0$ and $M$ such that $d_0 \equiv d \pmod{M}$ and $M \geq 2N^{\frac{1}{(r+1)^2}}$, then $N$ can be factored in time $e \cdot poly(\log N)$.*
2. *Let $e = N^\alpha$, $(\alpha \in [0, \frac{r}{(r+1)^2}])$ be the public exponent and let $d_p$ satisfy $ed_p \equiv 1 \pmod{(p-1)}$. Given $\widetilde{d_p}$ such that $|d_p - \widetilde{d_p}| \leq N^{\frac{r}{(r+1)^2} - \alpha}$, then $N$ can be factored in polynomial time.*

When $N = p^r q$, Euler's totient function is given by $\phi(N) = p^{r-1}(p-1)(q-1)$. Working modulo $p-1$, the public/private key equation can be written

$$ed_p = 1 + k(p-1),$$

for some integer $k$. Let $d_0 = d_p \bmod M$ be the $M$ least significant bits of $d_p$ and write $d = d_1 M + d_0$ where $d_1 < \frac{p}{M} \leq N^{r/(r+1)^2}$. It is assumed that $d_0$ is known. The public/private key equation can be written

$$ed_1 M + ed_0 + k - 1 = kp.$$

Multiplying this equation by the inverse of $eM$ modulo $N$, denoted by $E$, yields

$$d_1 + E(ed_0 + k - 1) = (Ek - cp^{r-1}qd_1)p,$$

where $c$ is some integer satisfying $EeM = 1 + cN$. Notice that $B = E(ed_0 + k - 1)$ is multiple of $p$ up to an additive error $d_1$. Using the factoring

method of Theorem 4.2, with $B = E(ed_0 + k - 1)$, $N$ can be factored since $d_1 < N^{r/(r+1)^2}$.

Let $\widetilde{d_p}$ be the known most significant bits of $d_p$. Notice that $e\widetilde{d_p}$ is an approximation of $kp$ up to a small additive error given by

$$|kp - e\widetilde{d_p}| = |e(d_p - \widetilde{d_p}) + k - 1| \leq N^{r/(r+1)^2} + N^{\alpha} \leq 2N^{r/(r+1)^2}.$$

Therefore, applying the factoring method of Theorem 4.2 with $e\widetilde{d_p}+N^{r/(r+1)^2}$ and $e\widetilde{d_p} - N^{r/(r+1)^2}$ will reveal the factorization of $N$.

May also gives partial key-exposure attacks when some of the bits of $d$ are known. The results are summarized in the following theorem.

**Theorem 4.10 (May [58]).** *Let $N = p^r q$, where $r \geq 2$ is a known constant and $p$ and $q$ are balanced primes. Let $(e, d)$ be the public/private exponents satisfying $ed \equiv 1 \pmod{\phi(N)}$. Given $\widetilde{d}$ such that*

$$|d - \widetilde{d}| \leq N^{\frac{r}{(r+1)^2}} \quad \textit{or} \quad |d - \widetilde{d}| \leq N^{\left(\frac{r-1}{r+1}\right)^2},$$

*or given $d_0$ and $M$ such that $d_0 \equiv d \pmod{M}$ and*

$$M \geq N^{1 - \frac{r}{(r-1)^2}} \quad \textit{or} \quad M \geq N^{\frac{4r}{(r+1)^2}},$$

*then $N$ can be factored in probabilistic time.*

Each of these results are simple consequences of May's small private exponent attacks when $N = p^r q$. We leave the details to [58].

**Time-Line** In Table 3, we present a time-line of partial key-exposure attacks on RSA and RSA-like systems that use lattice basis reduction.

## 4.3 ESIGN Signature Scheme

In 2003, Proos [77] showed that the ESIGN signature scheme can be reduced to a GPACDP. For a review of the ESIGN signature scheme see [32] or Appendix A.6. Given $\ell$ ESIGN signatures using the same public key $(N, e, k, H)$, if an attacker has knowledge of some of the most (or least) significant bits of the random nonces ($r_i$ for $i = 1, \ldots, \ell$) used in the signature generation, then $\ell$ linearly independent equations in $\ell$ unknowns can be found. Solving this allows the modulus $N = p^2 q$ to be factored.

| Year | Authors | | $\alpha = \log_N(e)$ | Fraction of Bits Needed | Assumptions/Conditions |
|---|---|---|---|---|---|
| 1998 | [11] | (1) | $[\frac{1}{4}, \frac{1}{2}]$ | $\alpha$ | $e$ has known factorization |
| | | (2) | $[0, \frac{1}{2}]$ | $1-\alpha$ | $\frac{d}{\phi(N)} \in \Omega(1)$ |
| | | (3) | $[0, \frac{1}{2}]$ | $\frac{3}{4}$ | $\frac{d}{\phi(N)}, \frac{\lvert p-q\rvert}{\sqrt{N}} \in \Omega(1)$ |
| 2003 | [7] | (4) | $[\frac{1}{2}, \frac{\sqrt{6}-1}{2}]$ | $\frac{5-2\alpha-\sqrt{36\alpha^2+12\alpha-15}}{8}$ | |
| | | (5) | $[0, \frac{1}{4}]$ | $\frac{1}{4}+\alpha$ | bits of $d_p$ |
| 2004 | [40] | (6) | $[\frac{1}{r}, \frac{4-3r+\sqrt{9r^2-12r+12}}{4r}]$ | $\frac{5-\alpha r-\sqrt{3(\alpha r-1)(3\alpha r+8r-11)}}{4r}$ | $N = p_1 \cdots p_r$ extension of (4) |
| 2004 | [58] | (7) | *all* | $1-\frac{r}{(r+1)^2}$ | $N = p^r q$, $r \geq 2$ |
| | | (8) | *all* | $\frac{4r}{(r+1)^2}$ | $N = p^r q$, $r \geq 2$ |
| | | (9) | $[0, \frac{r}{(r+1)^2}]$ | $1+\alpha-\frac{r}{(r+1)^2}$ | bits of $d_p$, $N = p^r q$, $r \geq 1$ |
| 1998 | [11] | (1) | $\mathcal{O}(\log\log N)$ | $\frac{1}{4}$ | $p \not\equiv q \pmod 4$ |
| 2003 | [7] | (2) | $[0, \frac{1}{2}]$ | $\frac{1}{2}-\alpha$ | all but $\mathcal{O}(N^{\alpha-\epsilon})$ of the $e$'s |
| | | (3) | $[0, \frac{7}{8}]$ | $\frac{1}{6}+\frac{1}{3}\sqrt{1+6\alpha}$ | |
| | | (4) | $\mathcal{O}(\log\log N)$ | $\frac{1}{4}$ | bits of $d_p$ |
| 2004 | [40] | (5) | $[0, \frac{1}{r}]$ | $\alpha+\frac{1}{r}+\epsilon$ | $N = p_1 \cdots p_r$ extension of (2)<br>all but $\mathcal{O}(1)$ $e$'s<br>$-\frac{2}{r} < \epsilon - 1 < \frac{1}{r}, \alpha+\epsilon+\frac{1}{r} < 1$ |
| | | (6) | $[0, \frac{4r-1}{4r(r-1)}]$ | $\frac{r-1+2\sqrt{(r-1)(3\alpha r+r-1)}}{3r}$ | $N = p_1 \cdots p_r$ extension of (3) |
| 2004 | [58] | (7) | *all* | $1-\frac{r}{(r+1)^2}$ | $N = p^r q$, $r \geq 2$ |
| | | (8) | *all* | $\frac{4r}{(r+1)^2}$ | $N = p^r q$, $r \geq 2$ |
| | | (9) | $\mathcal{O}(\log\log N)$ | $\frac{1}{(r+1)^2}+\log_N 2$ | bits of $d_p$, $N = p^r q$, $r \geq 1$ |

Table 3: Time-Line of Partial Key-Exposure Attacks on RSA and RSA-like cryptosystems. The top half of the table corresponds to attacks with known most significant bits. The bottom half of the table corresponds to attacks with known least significant bits. It is assumed that all primes are balanced.

Let $s_1, \ldots s_l$ be $\ell$ ESIGN signatures created with the same public key. By the definition of ESIGN, the $s_i$ satisfy

$$s_i = r_i + t_i pq,$$

where $0 \leq r_i < pq$ is the random nonce and $0 \leq t < p$. This leads to an instance of the GPACDP with $a_0 = N$, $a_1 = s_1$, $\ldots s_\ell = s_\ell$ where $X_i = pq$ and $D = pq - 1$. As $pq$ is not known the approximation $X_i = 2^{2k}$ and $D = 2^{2k-1}$ is used since $p$ and $q$ are both $k$-bit numbers by definition. As described by Proos, for practical values of $k$ this leads to bounds $\alpha \approx \beta \approx 2/3$ (recall from Section 3.2.3 that $X_1 = a_0^\beta$ and $D = a_0^\alpha$) and these bounds can only be reached asymptotically. If some knowledge of the random nonces $r_i$ are known, however, Proos shows that this leads to solvable instances of GPACDPs. If the adversary is given access to values $R_i$ such that $r_i = R_i + r_i'$ where $r_i' < X$ for $i \leq i \leq \ell$, then this leads to the GPACDP with $a_0 = N$, $a_1 = s_1 - R_1, \ldots, a_\ell = s_\ell - R_\ell$ where $X_i = X$ and $D = 2^{2k-1}$. For various values of $\ell$ and lattice dimensions, Proos gives bounds on $X$ such that $\ell$ linearly independent integer polynomials all sharing a common root over $\mathbb{Z}$ can be found. This common root contains the information needed to expose the secret key. In particular, $p$ and $q$ are obtained as follows: solving the system of equations $r_i$ which can be used to compute $\gcd(s_i - r_i, N) = pq$ which yields $N/(pq) = p$. For the actual bounds on $X$ see [77].

## 4.4 NBD Signature and Identification Schemes

In 2003, Proos [77] showed that both the NBD signature and identification schemes can be reduced to a GPACDP. For a review of the NBD signature and identification schemes see Nieto, Boyd & Dawson [76] or Appendix A.7. Since the signature scheme is essentially the identification scheme with the values of $s_i$ fixed by the message and $x_i$, the reduction is essentially the same for both schemes. We will only consider the signature scheme.

Let $(x_{0,1}, x_{1,1}, u_1), \ldots, (x_{0,\ell}, x_{1,\ell,u_\ell})$ be $\ell$ NBD signatures for the $\ell$ messages $m_1, \ldots, m_\ell$ all using the same public key. Letting $r_{0,j}, r_{1,j}, s_{0,j}$, and $s_{1,j}$ be the values of $r_0, r_1, s_0$, and $s_1$ be the used in the generation of $(x_{0,j}, x_{1,j}, u_j)$ leads to the following relations

$$(u_j s_{0,j}^{-1} \bmod N) \equiv r_{0,j} \pmod{q_0} \quad \text{and}$$

$$(u_j s_{1,j}^{-1} \bmod N) \equiv r_{1,j} \pmod{q_1},$$

for $1 \leq j \leq \ell$. Denoting $A_{i,j} = u_j s_{i,j}^{-1} \pmod{N}$, these relations lead to two GPACDPs given by

$$a_0 = N, a_1 = A_{0,1}, \ldots, A_{0,\ell} \quad \text{with divisor } q_0 \quad (GPACDP0), \text{ and}$$
$$a_0 = N, a_1 = A_{1,1}, \ldots, A_{1,\ell} \quad \text{with divisor } q_1 \quad (GPACDP1).$$

As in the case of ESIGN signatures, each GPACDP has bounds ($\alpha$ and $\beta$) that can only be reached asymptotically. If some of the most or least significant bits of $r_{0,j}$ or $r_{1,j}$ are known, a solvable GPACDP can be constructed. Assume that the least significant bits of $r_{i,j}$ are known. That is, the values $R_j$ and $b$ such $r_{0,j} = R_j + r'_{0,j}2^b$ where $r'_{1,j} < X$ are known for $1 \leq j \leq \ell$. A new GPACDP with $a_0 = N$, $a_1 = (A_{0,1} - R_1)2^{-b} \pmod{N}$, $\ldots$, $a_\ell = (A_{0,\ell} - R_\ell)2^{-b}$ where $X_i = X$ and $D = 2^{k-1}$. For various of values of $\ell$ and lattice dimensions, Proos gives bounds on $X$ such that $\ell$ linearly independent integer polynomials all sharing a common root over $\mathbb{Z}$ can be found. See [77] for details.

**Acknowledgements** The author would like to thank Jeff Shallit, Mark Giesbrecht and Doug Stinson for some helpful editorial comments.

# A Algorithms, Cryptosystems, Signature Schemes, etc.

## A.1 LLL-Algorithm

Before giving an algorithm for computing a reduced lattice basis, first we recall the Gram-Schmidt orthogonalization (GSO) process.

**Algorithm A.1:** GSO($\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$)

**input:** Linearly independent vectors $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n \in \mathbb{R}^n$

**output:** The Gram-Schmidt orthogonalization of $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$:

$\boldsymbol{g}_1^*, \ldots, \boldsymbol{g}_n^*$ and $\mu_{i,j}$ for $1 \leq j < i \leq n$

$\boldsymbol{g}_1^* \leftarrow \boldsymbol{g}_1$
**for** $i \leftarrow 2$ **to** $n$
**do** $\begin{cases} \boldsymbol{g}_i^* \leftarrow \boldsymbol{g}_i \\ \textbf{for } j \leftarrow 1 \textbf{ to } i-1 \\ \quad \textbf{do } \begin{cases} \mu_{i,j} \leftarrow \langle \boldsymbol{g}_i, \boldsymbol{g}_j^* \rangle / \langle \boldsymbol{g}_j^*, \boldsymbol{g}_j^* \rangle \\ \boldsymbol{g}_i^* \leftarrow \boldsymbol{g}_i^* - \mu_{i,j} \boldsymbol{g}_j^* \end{cases} \end{cases}$
**return** $\boldsymbol{g}_1^*, \ldots, \boldsymbol{g}_n^*$ and $\mu_{i,j}$ for $1 \leq j < i \leq n$

The GSO is needed in the following presentation of the LLL-algorithm as given by von zur Gathen & Gerhard in [90].

**Algorithm A.2:** LLL($\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$)

**input:** A basis $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n \in \mathbb{Z}^n$ of a lattice $\mathcal{L}$

**output:** An LLL-reduced basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{Z}^n$ for $\mathcal{L}$

**for** $i \leftarrow 1$ **to** $n$
**do** $\boldsymbol{b}_i \leftarrow \boldsymbol{f}_i$
compute GSO($\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$) (Algorithm A.1)
$i \leftarrow 2$
**while** $i \leq n$
**do** $\begin{cases} \textbf{for } j \leftarrow i-1 \textbf{ downto } 1 \\ \quad \textbf{do } \begin{cases} \boldsymbol{b}_i \leftarrow \boldsymbol{b}_i - \lceil \mu_{i,j} \rfloor \boldsymbol{b}_j \\ \text{update the GSO} \end{cases} \\ \textbf{if } i > 1 \textbf{ and } \|\boldsymbol{b}_{i-1}^*\|^2 > 2\|\boldsymbol{b}_i^*\|^2 \\ \quad \textbf{then } \begin{cases} \text{swap } \boldsymbol{b}_{i-1} \text{ and } \boldsymbol{b}_i \\ \text{update GSO}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \\ i \leftarrow i-1 \end{cases} \\ \quad \textbf{else } i \leftarrow i+1 \end{cases}$
**return** $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$

It should be noted that each call to GSO does not necessarily compute the full Gram-Schmidt orthogonalization of the input vectors. The first call to GSO does compute the full orthogonalization. For each call after the first, only the portion of the orthogonalization that changes is considered in the call to GSO (see [90] for more details).

## A.2 Knapsacks

We only present the basic Merkle-Hellman knapsack cryptosystem here. The multiple-iterated version is a simple extension of the basic version and can be found in Merkle & Hellman [60].

---

**Algorithm A.3:** MERKLE-HELLMAN–KEY–GENERATION($n$)

---

**input:** A fixed parameter $n$

**output:** A private key $(\pi, M, W, (b_1, \ldots, b_n))$ and
a public key $(a_1, \ldots, a_n)$

$(b_1, \ldots, b_n) \leftarrow$ a superincreasing sequence
$M \leftarrow$ an integer such that $M > b_1 + \cdots + b_n$ (called the modulus)
$W \leftarrow$ random integer such that $1 \leq W \leq M - 1$ and $\gcd(M, W) = 1$
$\pi \leftarrow$ random permutation of $\{1, \ldots, n\}$
**for** $i \leftarrow 1$ **to** $n$
  **do** $a_i \leftarrow W b_{\pi(i)} \mod M$
**return** private key$(\pi, M, W, (b_1, \ldots, b_n))$ and public key $(a_1, \ldots, a_n)$

---

**Algorithm A.4:** MERKLE-HELLMAN–ENCRYPT($m, (a_1, \ldots, a_n)$)

---

**input:** Public key $(a_1, \ldots, a_n)$ and plaintext $m \in \{0, 1\}^n$

**output:** Ciphertext $c \in \{0, 1\}^n$ of the plaintext $m$

let $m = m_1 \cdots m_n$
$c \leftarrow m_1 a_1 + \cdots + m_n a_n$
**return** $c$

---

**Algorithm A.5:** MERKLE-HELLMAN–ENCRYPT$(m, (a_1, \ldots, a_n))$

**input:** Private key $(\pi, M, W, (b_1, \ldots, b_n))$ and ciphertext $c$

**output:** Plaintext $m$ corresponding to $c$

$d \leftarrow W^{-1}c \mod M$
$(r_1, \ldots, r_n) \leftarrow$ integers $r_i \in \{0, 1\}$ such that $d = r_1 b_1 + \cdots r_n b_n$
**for** $i \leftarrow 1$ **to** $n$
  **do** $m_i \leftarrow r_{\pi(i)}$
**return** $m$

## A.3 DSA

The following algorithms for the Digital Signature Algorithm (DSA) are taken from Menezes, van Oorschot & Vanstone [59].

**Algorithm A.6:** DSA–KEY–GENERATION$(none)$

**output:** A private key $a$ and public key $(p, q, \alpha, y)$

$q \leftarrow$ random prime such that $2^{159} < q < 2^{160}$
$p \leftarrow$ random prime such that $2^{511+64t} < p < 2^{512+64t}$ for some $t$
  satisfying $0 \leq t \leq 8$ and $q \mid (p-1)$
**repeat**
 $g \leftarrow$ some element of $\mathbb{Z}_p^*$
 $\alpha \leftarrow g^{(p-1)/q} \mod p$
**until** $\alpha \neq 1$
**comment:** $\alpha$ generates a cyclic subgroup of order $q$ in $\mathbb{Z}_p^*$
$a \leftarrow$ random integer such that $1 \leq a \leq q-1$
$y \leftarrow \alpha^a \mod p$
**return** private key $a$ and public key $(p, q, \alpha, y)$

**Algorithm A.7:** DSA–SIGN$(m, a, (p, q, \alpha, y))$

**input:** A message $m$ to be signed

DSA private and public keys $a$ and $(p, q, \alpha, y)$

**output:** A signature $(r, s)$ for $m$

**comment:** $h : \{0,1\}^* \to \mathbb{Z}_q$ is the Secure Hash Algorithm SHA-1

$k \leftarrow$ random integer such that $0 < k < q$ (this is the nonce)
$r \leftarrow (\alpha^k \mod p) \mod q$
$s \leftarrow k^{-1}(h(m) + \alpha r) \mod 1$
**return** (r,s)

**Algorithm A.8:** DSA–VERIFY$((r, s), m, (p, q, \alpha, y))$

**input:** A signature $(r, s)$, a message $m \in \{0,1\}^*$ and a DSA

public key $(p, q, \alpha, y)$)

**output:** 1 if $(r, s)$ is a signature of $m$ for public key $(p, q, \alpha, y)$

0 otherwise

**comment:** $h : \{0,1\}^* \to \mathbb{Z}_q$ is the Secure Hash Algorithm SHA-1

**if** $0 < r < q$ **and** $0 < s < r$

**then**
$w \leftarrow s^{-1} \mod q$
$u_1 \leftarrow wh(m) \mod q$
$u_2 \leftarrow rw \mod q$
$v \leftarrow (\alpha^{u_1} y^{u_2} \mod p) \mod q$
**if** $v = r$
**then return** 1

**return** 0

## A.4 GnuPG

Full details (the actual code) of the algorithms used in GPG can be found at [34]. The algorithms described below only apply to versions of GPG up to version 1.2.3. The current stable version is 1.2.6.

| Bit-length of $p$ | 512 | 768 | 1024 | 1280 | 1536 | 1792 | 2048 |
|---|---|---|---|---|---|---|---|
| $q_{bit}$ | 119 | 145 | 165 | 183 | 198 | 212 | 225 |
| Bit-length of $p$ | 2304 | 2560 | 2816 | 3072 | 3328 | 2584 | 3840 |
| $q_{bit}$ | 237 | 249 | 259 | 269 | 279 | 288 | 296 |

Table 4: Wiener table for ElGamal primes.

**Algorithm A.9:** GPG-ElGamal–Key–Generation($k$)

**input:** Bitsize of the keys $k$

**output:** A private key $x$ and public key $(p, g, y)$

$p \leftarrow k$-bit prime such that the factorization of $p-1$ is completely known, and all factors of $\frac{p-1}{2}$ have bitlength at least $q_{bit}$ (see Table 4)
$g \leftarrow 3$
**while** $g$ is not a generator of $\mathbb{Z}_p^*$
  **do** $g \leftarrow g+1$
$x \leftarrow$ random number with bitlength $\frac{3}{2}q_{bit}$
$y \leftarrow g^x \mod p$
**return** private key $x$ and public key $(p, g, y)$

**Algorithm A.10:** GPG-ElGamal–Sign($m, x, (p, g, y)$)

**input:** A message $m \in \mathbb{Z}_p$ to be signed (see [67] for details of $m$)
private and public keys $x$ and $(p, g, y)$

**output:** A signature $(a, b)$ for $m$

$k \leftarrow$ random number with bitlength $\frac{3}{2}q_{bit}$ (this is the nonce)
**while** $\gcd(k, p-1) \neq 1$
  **do** $k \leftarrow k+1$
$a \leftarrow g^k \mod p$
$b \leftarrow (m - ax)k^{-1} \mod p-1$
**return** $(a, b)$

**Algorithm A.11:** GPG-ElGamal–Verify$((a, b), m, (p, g, y))$

**input:** A signature $(a, b)$, a message $m \in \mathbb{Z}_p$ and
a public key $(p, g, y)$

**output:** 1 if $(a, b)$ is a signature of $m$ for public key $(p, g, y)$
0 otherwise

**if** $0 < a < p$ **and** $y^a a^b \equiv g^m \pmod{p}$
**then return** 1
**else return** 0

## A.5 RSA

We only present the basic, or textbook, RSA cryptosystem as described by Rivest, Shamir & Adleman [79].

**Algorithm A.12:** RSA–Key–Generation$(k)$

**input:** The bitsize of the modulus $k$

**output:** A private key $(N, d)$ and public key $(N, e)$

**repeat**
$p, q \leftarrow$ distinct random $(\frac{k}{2})$-bit primes
$N \leftarrow pq$
**until** bitsize of $N$ is $k$
$e \leftarrow$ random element of $\mathbb{Z}_N^*$
$d \leftarrow e^{-1} \bmod \phi(N)$
**return** private key $(N, d)$ and public key $(N, e)$

**Algorithm A.13:** RSA–Encrypt$((N, e), m)$

**input:** Public key $(N, e)$ and plaintext $m$

**output:** Ciphertext $c$ of the plaintext $m$

$c \leftarrow m^e \mod N$
**return** $c$

**Algorithm A.14:** RSA–DECRYPT$((N, d), c)$

**input:** Private key $(N, d)$ and ciphertext $c$

**output:** Plaintext $m$ corresponding to ciphertext $c$

$m \leftarrow c^d \mod N$
**return** $m$

## A.6 ESIGN Signature Scheme

Full details of ESIGN can be found in [32].

**Algorithm A.15:** ESIGN–KEY–GENERATION$(k)$

**input:** A security parameter $k \geq 352$

**output:** A private key $(p, q)$ and public key $(N, e, k, H)$

**repeat**
  $p, q \leftarrow$ distinct random $k$-bit primes
  $N \leftarrow p^2 q$
**until** bitsize of $N$ is $3k$
$e \leftarrow$ an integer $\geq 8$
$H \leftarrow$ some hash function mapping $\{0,1\}^* \rightarrow \{0,1\}^{k-1}$
**return** private key $(p, q)$ and public key $(N, e, k, H)$

**Algorithm A.16:** ESIGN–Sign$(m, (p, q), (N, e, k, H))$

**input:** A message $m \in \{0,1\}^*$ to be signed
ESIGN private and public keys $(p, q)$ and $(N, e, k, H)$

**output:** A signature $s \in \{0,1\}^{3k}$ for $m$

$r \leftarrow$ random element of $\mathbb{Z}_{pq}^*$ (this is the nonce)
$z \leftarrow 0 \,\|\, H(m) \,\|\, 0^{2k}$
$a \leftarrow z - r^e \bmod N$
$w_0 \leftarrow \lceil \frac{a}{pq} \rceil$
$w_1 \leftarrow w_0 pq - a$
**if** $p < q$ **and** $w_1 \geq 2^{2k-1}$
  **then** start over
$t \leftarrow w_0 (e r^{e-1})^{-1} \bmod p$
$s \leftarrow r + tpq$
**return** $s$

**Algorithm A.17:** ESIGN–Verify$(s, m, (N, e, k, H))$

**input:** A signature $s \in \{0,1\}^{3k}$, a message $m \in \{0,1\}^*$ and
an ESIGN public key $(N, e, k, H)$

**output:** 1 if $s$ is a signature of $m$ for public key $(N, e, k, H)$
0 otherwise

$y \leftarrow s^e \bmod N$
**if** the $k$ most significant bits of $y$ are $0 \,\|\, H(m)$
  **then return** 1
  **else return** 0

## A.7 NBD Signature and Identification Schemes

For full details of the NBD signature and identification schemes see Nieto, Boyd & Dawson [76].

**Algorithm A.18:** NBD–KEY–GENERATION($k$)

**input:** A security parameter $k$

**output:** A private key $(q_0, q_1, \alpha_0, \alpha_1)$ and public key $(p, N, g_0, g_1)$

**repeat**
  $q_0, q_1 \leftarrow$ distinct random $k$-bit primes
  $p \leftarrow 2q_0q_1 + 1$
**until** $p$ is prime
$N \leftarrow q_0q_1$
$\alpha_0 \leftarrow q_1(q_1^{-1} \mod q_0)$
$\alpha_1 \leftarrow q_0(q_0^{-1} \mod q_1)$
$g_0 \leftarrow$ element of $\mathbb{Z}_p^*$ of order $q_0$
$g_1 \leftarrow$ element of $\mathbb{Z}_p^*$ of order $q_1$
**return** private key $(q_0, q_1, \alpha_0, \alpha_1)$ and public key $(p, N, g_0, g_1)$

**Algorithm A.19:** NBD–SIGN($m, (q_0, q_1, \alpha_0, \alpha_1), (p, N, g_0, g_1)$)

**input:** A message $m \in \{0,1\}^*$ to be signed
NBD private and public keys $(q_0, q_1, \alpha_0, \alpha_1)$ and $(p, N, g_0, g_1)$

**output:** A signature $(x_0, x_1, u)$ for $m$

$r \leftarrow$ random element of $\mathbb{Z}_N$ (this is the nonce)
**for** $i \leftarrow 0$ **to** 1
**do**
$r_i \leftarrow r \pmod{q_i}$
$x_i \leftarrow g_i^{r_i}$
$s_i \leftarrow H(x_i, m) \in \{1, 2, \ldots 2^t - 1\}$
  where $H$ is a hash function, $2^t - 1 < q_i$ and $t$ is defined
  to be large enough so that $H$ is one-way and collision free
$u \leftarrow r_0s_0\alpha_0 + r_1s_1\alpha_1 \pmod{N}$
**return** $(x_0, x_1, u)$

**Algorithm A.20:** NBD–VERIFY$(s, m, (p, N, g_0, g_1))$

**input:** A signature $(x_0, x_1, u)$, a message $m \in \{0,1\}^*$ and
an NBD public key $(p, N, g_0, g_1)$

**output:** 1 if $(x_0, x_1, u)$ is a signature of $m$ for public key $(p, N, g_0, g_1)$
0 otherwise

$s_0 \leftarrow H(x_0, m)$
$s_1 \leftarrow H(x_1, m)$
**if** $x_0 = g_0^{us_0^{-1} \pmod N}$ **and** $x_1 = g_1^{us_1^{-1} \pmod N}$
**then return** 1
**else return** 0

**Algorithm A.21:** NBD–IDENTIFICATION$((p, N, g_0, g_1), P, V)$

**input:** An NBD public key $(p, N, g_0, g_1)$ and two identities $P$ and $V$

**output:** A proof of knowledge for $V$ that $P$ knows the NBD private
key $(q_0, q_1, \alpha_0, \alpha_1)$ for the public key public key $(p, N, g_0, g_1)$

$P$ : $r \leftarrow$ random element of $\mathbb{Z}_N$ (the nonce)
$P$ : $r_0 \leftarrow r \pmod{q_0}$, $\quad x_0 \leftarrow g_0^{r_0}$
$P$ : $r_1 \leftarrow r \pmod{q_1}$, $\quad x_1 \leftarrow g_1^{r_1}$
$P$ : sends $x_0$ and $x_1$ to $V$
$V$ : $s_0, s_1 \leftarrow$ some elements of $\{1, 2, \ldots, 2^t - 1\}$
where $2^t - 1 < q_0, q_1$ and $t$ is large enough so that
guessing $s_0$ or $s_1$ is hard
$V$ : sends $s_0$ and $s_1$ to $P$
$P$ : $u \leftarrow r_0 s_0 \alpha_0 + r_1 s_1 \alpha_1 \pmod N$
$P$ : send $u$ to $V$
$V$ : checks that $x_0 = g_0^{us_0^{-1} \pmod N}$ and $x_1 = g_1^{us_1^{-1} \pmod N}$

## References


[1] M. Ajtai. The shortest vector problem in $l_2$ is NP-hard for randomized reductions. In *Proc. of 30th STOC*, pages 99–108. ACM, 1998. Available at `http://www.eccc.uni-trier.de/ecc/` as TR97-047.

[2] M. Ajtai, R. Kumar, and D. Sivalumar. A sieve algorithm for the shortest lattice vector problem. In *Proc. of 33rd STOC*, pages 601–610. ACM, 2001.

[3] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[4] L. Babai. On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.

[5] D. Bleichenbacher. On the security of the KMOV public key cryptosystem. In B. Kaliski, editor, *Advances in Cryptology – Proceedings of CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 235–248. Springer-Verlag, 1997.

[6] J. Blömer and A. May. Low secret exponent RSA revisited. In J. H. Silverman, editor, *Cryptography and Lattices – Proceedings of CALC '01*, volume 2146 of *Lecture Notes in Computer Science*, pages 4–19. Springer-Verlag, 2001.

[7] J. Blömer and A. May. New partial key exposure attacks on RSA. In D. Boneh, editor, *Advances in Cryptology – Proceedings of CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 27–43. Springer-Verlag, 2003.

[8] D. Boneh and G. Durfee. Cryptanalysis of RSA with private key $d$ less than $N^{0.292}$. In *Advances in Cryptology – Proceedings of EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 1–11. Springer-Verlag, 1999.

[9] D. Boneh and G. Durfee. Cryptanalysis of RSA with private key $d$ less than $N^{0.292}$. *IEEE Transactions on Information Theory*, 46(4):1339–1349, July 2000.

[10] D. Boneh, G. Durfee, and Y. Frankel. An attack on RSA given a small fraction of the private key bits. In *Advances in Cryptology – Proceed-*

*ings of ASIACRYPT '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 25–34, 1998.

[11] D. Boneh, G. Durfee, and Y. Frankel. Exposing an RSA private key given a small fraction of its bits. Revised and extended version of proceedings of ASIACRYPT '98 [10]. Available at `http://crypto.stanford.edu/~dabo/abstracts`, 2001.

[12] D. Boneh, G. Durfee, and N. A. Howgrave-Graham. Factoring $N = p^r q$ for large $r$. In *Advances in Cryptology – Proceedings of CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 326–337. Springer-Verlag, 1999.

[13] D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in diffie-hellman and related schemes. In *Advances in Cryptology – Proceedings of CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 129–142. Springer-Verlag, 1996.

[14] E. F. Brickell. Solving low density knapsacks. In D. Chaum, editor, *Advances in Cryptology – Proceedings of CRYPTO '83*, pages 25–37. Plenum Press, 1984.

[15] D. Brown and A. Menezes. A small subgroup attack on a key agreement protocol of Arazi. *Bulletin of the ICA*, 37:45–50, 2003.

[16] J. W. S. Cassels. *An Introduction to the Geometry of Numbers*. Springer-Verlag, second corrected edition, 1971.

[17] M. Ciet, F. Koeune, F. Laguillaumie, and J.-J. Quisquater. Short private exponent attacks on fast variants of rsa. UCL Crypto Group Technical Report Series CG-2002/4, Université Catholique de Louvain, 2002. Available at `http://www.dice.ucl.ac.be./crypto/tech_reports/`.

[18] H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, second edition, 1995.

[19] J. Conway and N. Sloane. *Sphere packings, lattices and groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 3rd edition, 1999.

[20] D. Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In U. Maurer, editor, *Advances in*

*Cryptology – Proceedings of EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 178–189. Springer-Verlag, 1996.

[21] D. Coppersmith. Finding a small root of a univariate modular equation. In U. Maurer, editor, *Advances in Cryptology – Proceedings of EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 155–165. Springer-Verlag, 1996.

[22] D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, 1997.

[23] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter. Low-exponent RSA with related messages. In U. Maurer, editor, *Advances in Cryptology – Proceedings of EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 1–9. Springer-Verlag, 1996.

[24] J.-S. Coron. Finding small roots of bivariate integer polynomial equations revisited. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – Proceedings of EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 492–505. Springer-Verlag, 2004.

[25] M. J. Coster, A. Joux, B. A. LaMacchia, A. M. Odlyzko, C.-P. Schnorr, and J. Stern. Improved low-density subset sum algorithms. *Computational Complexity*, 1992.

[26] M. J. Coster, B. A. LaMacchia, A. M. Odlyzko, and C.-P. Schnorr. An improved low-density subset sum algorithm. In D. W. Davies, editor, *Advances in Cryptology – Proceedings of EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 54–67. Springer-Verlag, 1991.

[27] Y. G. Desmedt, J. P. Vandewalle, and R. J. M. Govaerts. A critical analysis of the security of knapsack public-key algorithms. *IEEE Transactions on Information Theory*, IT-30(4):601–611, July 1984.

[28] G. Durfee and P. Q. Nguyen. Cryptanalysis of the RSA schems with short secret exponent from Asiacrypt '99. In T. Okamoto, editor, *Advances in Cryptology – Proceedings of ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 11–29. Springer-Verlag, 2000.

[29] R. Eier and H. Lagger. Trapdoors in knapsack cryptosystems. In T. Beth, editor, *Advances in Cryptology – Proceedings of CRYPTO '82*, volume 149 of *Lecture Notes in Computer Science*, pages 316–322. Springer-Verlag, 1983.

[30] M. K. Franklin and M. K. Reiter. A linear protocol failure for RSA with exponent three. Presented at the CRYPTO '95 Rump Session, 1995.

[31] A. M. Frieze. On the Lagarias-Odlyzko algorithm for the subset sum problem. *SIAM J. Comput.*, 15(2):536–539, 1986.

[32] E. Fujisaki, T. Kobayashi, H. Morita, H. Oguro, T. Okamoto, and S. Okazaki. ESIGN : Efficient digital signature scheme (submission to NESSIE). Available at `http://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions.html`, October 2000.

[33] M. Girault, P. Toffin, and B. Vallée. Computation of approximate $L$-th roots modulo $n$ and application to cryptography. In S. Goldwasser, editor, *Advances in Cryptology – Proceedings of CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 403–418. Springer-Verlag, 1990.

[34] GnuPG. The gnu privacy guard. `http://www.gnupg.org/`.

[35] O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In B. Kaliski, editor, *Advances in Cryptology – Proceedings of CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer-Verlag, 1997.

[36] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, second corrected edition, 1993.

[37] P. M. Gruber and C. G. Lekkerkerker. *Geometry of Numbers*, volume 37 of *North-Holland Mathematical Library*. North-Holland, second edition, 1987.

[38] J. Håstad. On using RSA with low exponent in a public key network. In H. C. Williams, editor, *Advances in Cryptology – Proceedings of CRYPTO '85*, volume 218 of *Lecture Notes in Computer Science*, pages 403–408. Springer-Verlag, 1986.

[39] J. Håstad. Solving simultaneous modular equations of low degree. *SIAM Journal on Computing*, 17(2):336–341, April 1988.

[40] M. J. Hinek. New partial key exposure attacks on RSA revisited. Technical Report CACR 2004-2, Centre for Applied Cryptographic Research, University of Waterloo, 2004. Available online at `http://www.cacr.math.uwaterloo.ca/`.

[41] M. J. Hinek, M. K. Low, and E. Teske. On some attacks on multi-prime RSA. In K. Nyberg and H. M. Heys, editors, *Selected Areas in Cryptography 2002*, volume 2595 of *Lecture Notes in Computer Science*, pages 385–404. Springer-Verlag, 2003.

[42] N. A. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Cryptography and Coding*, volume 1355 of *Lecture Notes in Computer Science*, pages 131–142. Springer-Verlag, 1997.

[43] N. A. Howgrave-Graham. *Computational Mathematics Inspired by RSA*. PhD thesis, University of Bath, 1998.

[44] N. A. Howgrave-Graham. Approximate integer common divisors. In J. H. Silverman, editor, *Cryptography and Lattices – Proceedings of CALC '01*, volume 2146 of *Lecture Notes in Computer Science*, pages 51–66. Springer-Verlag, 2001.

[45] N. A. Howgrave-Graham. A review of the ESIGN digital signature standard. Available at `http://www.ipa.go.jp/security/enc/CRYPTREC`, 2001.

[46] N. A. Howgrave-Graham and N. P. Smart. Lattice attacks on digital signature schemes. *Designs, Codes and Cryptography*, 23(3):283–290, August 2001.

[47] A. Joux and J. Stern. Improving the critical density of the Lagarias-Odlyzko attack against subset sum problems. In L. Budach, editor, *Proceedings of Fundamentals of Computation Theory '91*, volume 529 of *Lecture Notes in Computer Science*, pages 258–264. Springer-Verlag, 1991.

[48] M. Joye, F. Koeune, and J.-J. Quisquater. Takagi/Naito's algorithm revisited. UCL Crypto Group Technical Report Series CG-1997/3, Université Catholique de Louvain, 1997. Available at `http://www.dice.ucl.ac.be/crypto/tech_reports/`.

[49] C. Jutla. On finding small solutions of modular multivariate polynomial equations. In K. Nyberg, editor, *Advances in Cryptology – Proceedings of EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 158–170. Springer-Verlag, 1998.

[50] R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proc. of the 15th STOC*, pages 193–206. ACM, 1983.

[51] R. Kannan. Algorithmic geometry of numbers. *Annual Revie of Computer Science*, 2(231–267), 1987.

[52] J. C. Lagarias. Knapsack public key cryptosystems and diophantine approximation. In D. Chaum, editor, *Advances in Cryptology – Proceedings of CRYPTO '83*, pages 2–23. Plenum Press, 1984.

[53] J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *J. Assoc. Comput. Mach.*, 31(1):229–246, 1985.

[54] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[55] L. Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity*, volume 50 of *CBMS*. SIAM, 1986.

[56] J. Martinet. *Perfect Lattices in Euclidean Spaces*. Springer-Verlag, 2003. English version of the Éditions Masson 1996 French edition.

[57] A. May. Cryptanalysis of unvalanced RSA with small CRT-exponent. In M. Yung, editor, *Advances in Cryptology – Proceedings of CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 242–256. Springer-Verlag, 2002.

[58] A. May. Secret exponent attacks on RSA-type schemes with moduli $N = p^r q$. In F. Bao, R. Deng, and J. Zhou, editors, *Public Key Cryptograph - PKC 2004: 7th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2947 of *Lecture Notes in Computer Science*, pages 218–230. Springer-Verlag, 2004.

[59] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[60] R. C. Merkle and M. E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, IT-24(5):525–530, September 1978.

[61] D. Micciancio. The shortest vector problem is NP-hard to approximate within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, March 2001. A preliminary version appeared in Proc. of the 39th FOCS, 1998.

[62] M. Mignotte. An inequality about factors of polynomials. *Mathematics of Computation*, 28(128):1153–1157, October 1974.

[63] J. W. Milnor and D. Husemoller. *Symmetric Bilinear Forms*. Springer-Verlag, 1973.

[64] P. Nguyen and J. Stern. The two faces of lattices in cryptology. In J. Silverman, editor, *Cryptography and Lattices – Proceedings of CALC '01*, volume 2146 of *Lecture Notes in Computer Science*, pages 146–180. Springer-Verlag, 2001.

[65] P. Q. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto '97. In *Advances in Cryptology – Proceedings of CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 288–304. Springer-Verlag, 1999.

[66] P. Q. Nguyen. The dark side of the hidden number problem : Lattice attacks on DSA. In *Proc. Workshop on Cryptography and Comp. Number Theory (CCNT'99)*. Birkhauser, 2000.

[67] P. Q. Nguyen. Can we trust cryptographic software? Cryptographic flaws in GNU Privacy Guard v1.2.3. In C. Cachin, editor, *Advances in Cryptology – Proceedings of EUROCRYPT '94*, volume 3027 of *Lecture Notes in Computer Science*, pages 555–570. Springer-Verlag, 2004.

[68] P. Q. Nguyen and I. E. Shparlinski. The insecurity of Nyberg-Rueppel and other DSA-like signature schemes with partially known nonces. In J. H. Silverman, editor, *Cryptography and Lattices – Proceedings of CALC '01*, volume 2146 of *Lecture Notes in Computer Science*, pages 97–109. Springer-Verlag, 2001.

[69] P. Q. Nguyen and I. E. Shparlinski. The insecurity of the digital signature algorithm with partially known nonces. *Journal of Cryptology*, 15(3):151–176, 2002.

[70] P. Q. Nguyen and I. E. Shparlinski. The insecurity of the elliptic curve digital signature algorithm with partially known nonces. *Designs, Codes and Cryptography*, 30(2):201–217, September 2003.

[71] P. Q. Nguyen and D. Stehlé. Low-dimensional lattice basis reduction revisited. In D. Buell, editor, *Algorithmic Number Theory : 6th International Symposium, ANTS-VI*, volume 3076 of *Lecture Notes in Computer Science*, pages 338–357. Springer-Verlag, 2004.

[72] P. Q. Nguyen and J. Stern. Merkle-Hellman resivisted: A cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations. In B. Kaliski, editor, *Advances in Cryptology – Proceedings of CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 198–212. Springer-Verlag, 1997.

[73] P. Q. Nguyen and J. Stern. The Béguin-Quisquater server-aided RSA protocol from Crypto '95 is not secure. In K. Ohta and D. Pei, editors, *Advances in Cryptology – Proceedings of ASIACRYPT '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 371–379. Springer-Verlag, 1998.

[74] P. Q. Nguyen and J. Stern. Cryptanalysis of a fast public key cryptosystem presented at SAC '97. In S. Tavares and H. Meijer, editors, *Selected Areas in Cryptography '98*, volume 1556 of *Lecture Notes in Computer Science*, pages 213–218. Springer-Verlag, 1999.

[75] P. Q. Nguyen and J. Stern. The hardnes of the hidden subset sum problem and its cryptographics implications. In M. Wiener, editor, *Advances in Cryptology – Proceedings of CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 31–46. Springer-Verlag, 1999.

[76] J. M. G. Nieto, C. Boyd, and E. Dawson. A public key cryptosystem based on the subgroup membership problem. In *Information and Communications Security : Third International Conference, ICICS 2001, Xian, China, Novvember 13–16, 2001*, volume 2229 of *Lecture Notes in Computer Science*, pages 352–363. Springer-Verlag, 2001.

[77] J. A. Proos. *Imperfect Decryption and Partial Information Attacks in Cryptography*. PhD thesis, University of Waterloo, 2003.

[78] M. Qu and S. A. Vanstone. The knapsack problem in cryptography. In G. L. Mullen and P. J.-S. Shiue, editors, *Finite Fields: Theory, Applications, and Algorithms*, volume 168 of *Contemporary Mathematics*, pages 291–308. American Mathematics Society, 1994.

[79] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Commun. of the ACM*, 21:120–126, 1978.

[80] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.

[81] A. Shamir. A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Transactions on Information Theory*, 1984.

[82] I. Shparlinski. Exponential sums and lattice reduction: Applications to cryptograph. In *Finite Fields with Applications to Coding Theory, Cryptography and Related Area*, pages 286–298. Springer-Verlag, 2002.

[83] I. Shparlinski. Playing "hide-and-seek" in finite fields: Hidden number problem and its applications. In *Proc. 7th Spanish Meeting on Cryptology and Information Security, Univ. of Oviedo*, pages 49–72, 2002.

[84] C. L. Siegel. *Lectures on the Geometry of Numbers*. Springer-Verlag, 1989.

[85] R. Steinfeld and Y. Zheng. An advantage of low-exponent RSA with modulus primes sharing least significant bits. In D. Naccache, editor, *Progress in Cryptology - CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 52–62. Springer-Verlag, 2001.

[86] H.-M. Sun, W.-C. Yang, and C.-S. Laih. On the design of RSA with short secret exponent. In K. Y. Lam, E. Okamoto, and C. Xing, editors, *Advances in Cryptology – Proceedings of ASIACRYPT '99*, volume 1716 of *Lecture Notes in Computer Science*, pages 150–164. Springer-Verlag, 1999.

[87] T. Takagi and S. Naito. The multi-variable modular polynomial and its applications to cryptography. In *7th International Symposium on Algorithm and Computation – ISAAC'96*, volume 1178 of *Lecture Notes in Computer Science*, pages 386–396. Springer-Verlag, 1996.

[88] B. Vallée, M. Girault, and P. Toffin. How to guess $\ell$-th roots modulo $n$ by reducing lattice bases. In T. Mora, editor, *Proceedings of Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 6th International Conference, AECC-6, 1988*, volume 357 of *Lecture Notes in Computer Science*, pages 427–442. Springer-Verlag, 1989.

[89] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in lattices. Technical Report 81-04, Mathematische Instituut, University of Amsterdam, 1981. Avaliable at `http://turing.wins.uva.nl/~peter/`.

[90] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.

[91] M. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*, 36(3):553–558, 1990.