Linear and Nonlinear Analysis
Volume 3, Number 2, 2017, 225–238
Yokohama Publishers
ISSN 2188-8167


# SIMULTANEOUS APPROXIMATIONS OF $p$-ADIC NUMBERS AND THEIR APPLICATIONS TO CRYPTOGRAPHY

HIROHITO INOUE, SHOICHI KAMADA, AND KOICHIRO NAITO

Abstract. In this paper we construct the multi-dimensional $p$-adic approximation lattices by using simultaneous approximation problems (SAP) of $p$-adic numbers and we estimate the $l_\infty$ norm of the $p$-adic SAP solutions theoretically by applying Dirichlet's principle and numerically by using the LLL algorithm. By using the SAP solutions as private keys, the security of which depends on NP-hardness of SAP or the shortest vector problems (SVP) of $p$-adic lattices, we propose a new lattice based cryptosystem.

## 1. Introduction

In the usual real numbers case the shortest vector problems (SVP) and the simultaneous approximation problems (SAP) have the computational complexity, NP-hardness or NP-completeness (cf. [5], [4]). The security of some modern cryptosystems is based on the hardness of these problems and the lattice-based cryptography is considered as one of the most powerful post-quantum cryptography. In the first part of this paper, extending the two-dimensional $p$-adic approximation lattices given by de Weger in [9] to multi-dimensional cases, we construct the multi-dimensional $p$-adic approximation lattices by using the SAP of $p$-adic numbers. Here the strong triangle inequality condition, which gives the significant properties in a field or a ring of $p$-adic numbers, plays the most essential role. We estimate the $l_\infty$ norm of the $p$-adic SAP solutions theoretically by applying Dirichlet's principle and numerically by using the LLL algorithm. In the second part, using the results in the first part, we propose a new lattice based cryptosystem.

The LLL algorithm, which is a multi-dimensional extension of the Gauss algorithm, approximately solves SVP within a factor of $2^{O(n)}$ for the lattice dimension $n(\geq 3)$ in polynomial time. Also the famous Minkowski's theorem gives the upper bound $\sqrt{n}(\det(\Lambda))^{1/n}$ for the norms of the shortest vectors where $\det(\Lambda)$ is the $n$-dimensional volume of the fundamental parallelepiped spanned by the basis vectors of the lattice. By applying Dirichlet's principle we prove that the $l_\infty$ norms of the shortest vectors or the SAP solutions of the $p$-adic approximation lattices have the upper bound value without the factor $\sqrt{n}$, exactly, the upper bound value is given by $p^{m/(n+1)}$ where $p$ is a prime, $m$ is the $p$-adic approximation order, $n+1$ is the

2010 *Mathematics Subject Classification.* 11J13, 11E55, 11A05, 14G50.

*Key words and phrases.* Simultaneous homogeneous approximation, $p$-adic theory, LLL algorithm, Cryptography.

dimension and $p^m$ is the determinant of our lattice. Using the LLL reduction algorithm in the open source software Sage, we also show that these numerical SVP solutions of the lattice dimensions under 60 satisfy these exact estimates in the $l_\infty$ norm.

In the second part of this paper, using these SVP or SAP solutions as private keys, we construct a cryptosystem, the security of which is based on the hardness of SVP or SAP and the $l_\infty$ estimates of these solutions. We choose a $n$-tuple of $p$-adic integers as public keys and we set the SAP solutions of these numbers as private keys where we do not apply the LLL algorithm and we can randomly choose almost all of the public keys and private keys. Since we can numerically show that the $l_\infty$ norms of the SVP solutions given by LLL in the lattices of dimensions over 60 exceed the boundary value $p^{m/(n+1)}$ of the SAP solutions, the private keys given in the lattices of dimensions over this value are considered to be secure for the attacks by LLL.

In the lower dimensional case, for instance, $n = 10, p = 13, m = 20$, taking linear combinations of the reduced basis vectors and the suitably taken integers less than $10^3$, we can obtain an extremely large number of candidates of the private keys. We can construct the case where a brute-force search attack requires almost $10^{30}(\sim 2^{100})$ exhaustive tries in the worst case to find a private key. In this case the sizes of these private keys are under $(10^3 p^{m/(n+1)})^n \sim 2^{170}$.

In the first proposed cryptosystem the cypher text is composed by a perturbed term and a message term. The perturbed term is a linear combination of rational integers, private keys, and $p$-adic integers, public keys. The message term is also a linear combination of a $\{0, 1\}$-coded plaintext and $p$-adic integers, which are also public keys. In our encryption and decryption procedures for the message term we use a knapsack type procedure. We prepare a sequence of $p$-adic integers, the $p$-adic absolute values of which is increasing, instead of a superincreasing sequence of usual real numbers used in knapsack cryptosystems.

The processing speeds of encryption and decryption procedures in our cryptosystem are given by $O(n) + O(l)$ operations where $l$ is the length of plaintexts. These numerical processing times on Sage are under 10 milliseconds even if the sizes of plaintexts are over 100 bits. To construct public keys we use Hensel's lemma, which contains single loops, and their construction times are proportional to the lengths of the keys. It takes only a few seconds ($< 10$ sec) even if $n, l \sim 100$ in our numerical calculations.

For the practical applications, to increase the security of our systems, we propose a separation of private keys between the sender Alice and the receiver Bob, that is, the sum of their private keys becomes the SAP solution. Furthermore, we propose a secret permutation of public keys and a secret decreasing integer sequence of $p$-adic absolute values in the message terms. In these practical cases we use the isosceles principle of $p$-adic absolute values in the decryption procedure. Furthermore, we can consider the prime number $p$, the $p$-adic approximation order $m$ and the precision order $M$ as private keys.

Our plan of this paper is as follows. In Section 2 we give a brief review of lattices and LLL algorithm. In Section 3 we investigate the relations between the SAP of $p$-adic numbers and the SVP of $p$-adic approximation lattices and we estimate the

$l_\infty$ norm of $p$-adic SAP solutions. In Section 4 we give the numerical estimates of the SAP solutions by using the LLL reduction algorithm. In Section 5 and 6 we propose new cryptosystems based on the results in the preceding sections. In Section 7 we implement our cryptosystems and we give their numerical experiments by using Sage. In section 8 we consider the security of our systems and we give some concluding remarks.

## 2. Lattice and LLL algorithm

In this section we give a brief review on lattices and the LLL algorithm. (For details, see [5], [6].)

Given linearly independent vectors $b_1, \ldots, b_n \in \mathbb{R}^m$, the lattice generated by these vectors is defined by

$$L(b_1, \ldots, b_n) = \{\sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z}\}.$$

We refer to $b_1, \ldots, b_n$ as a basis of the lattice.

Let $B$ be the $m \times n$ matrix whose columns are $b_1, \ldots, b_n$, then the lattice generated by $B$ is

$$L(B) = \{Bx : \ x \in \mathbb{Z}^n\}.$$

We say that the rank of lattice is $n$ and its dimension is $m$. If $n = m$, the lattice is called a full-rank lattice. Hereafter we consider full-rank lattices.

For matrix $B$, $P(B) = \{Bx : x \in [0,1)^n\}$ is called the fundamental parallelepiped of $B$. Let $\Lambda = L(B)$ be a lattice of rank $n$. We define the determinant of $\Lambda$, denoted by $\det(\Lambda)$, as the $n$-dimensional volume of $P(B)$. In the full rank case, $\det(\Lambda) = |\det(B)|$.

The $i$th successive minimum of lattice $\Lambda$, $\lambda_i(\Lambda)$, is defined by

$$\lambda_i(\Lambda) = \inf\{r : \ \dim(\mathrm{span}(\Lambda \cap \overline{B}(0,r))) \geq i\}$$

where $\overline{B}(0,r)$ is a closed ball with its center 0 and its radius $r > 0$. The length of the shortest nonzero vector in the lattice is denoted by $\lambda_1(\Lambda)$ and the second minimum vector should be linearly independent to the shortest vector. The following estimate for the shortest vector is given by Minkowski's theorem in the $l_2$ norm (Euclidean norm).

$$\lambda_1(\Lambda) \leq \sqrt{n}\{\det(\Lambda)\}^{1/n}. \tag{2.1}$$

For the successive minimum in the $l_\infty$ norm we use the notation $\lambda_i^{(\infty)}(\Lambda)$ and we also use $\lambda_i^{(2)}(\Lambda)$ for those in the $l_2$ norm to distinguish it from other norms. $\| \ \|_p$ denotes the $l_p$ norm for $1 \leq p \leq \infty$.

Next we introduce the algorithm given by Lenstra, Lenstra and Lovász, which approximately solves the Shortest Vector Problem (SVP) within a factor of $2^{O(n)}$ for the lattices dimension $n$. The basic idea of LLL algorithm is to generalize Gauss's algorithm to higher dimensions. For a basis $b_1, \ldots, b_n$ of a lattice, the Gram-Schmidt orthogonalized basis $b_1^*, \ldots, b_n^*$, which satisfies

$$\mathrm{span}(b_1, \ldots, b_k) = \mathrm{span}(b_1^*, \ldots, b_k^*), k = 1, \ldots, n$$

$$b_k = \sum_{i=1}^{k} \mu_{k,i} b_i^*,\ \mu_{k,i} = \frac{(b_k, b_i^*)}{(b_i^*, b_i^*)} \text{ for } \ i \le k-1,\ \mu_{k,k} = 1,$$

is essentially used to construct the reduced basis.

**Definition 2.1.** For a constant $\delta : 1/4 < \delta < 1$, a basis $\{b_1, \ldots, b_n\}$ of a lattice is called a $\delta$-reduced basis if it satisfies the following two conditions.

- $|\mu_{k,i}| = \left|\frac{(b_k, b_i^*)}{(b_i^*, b_i^*)}\right| \le \frac{1}{2}$ for all $i < k$,
- for any pair of consecutive vectors $b_i, b_{i+1}$,

$$\delta \|\pi_i(b_i)\|_2^2 \le \|\pi_i(b_{i+1})\|_2^2$$

where we define projection operations $\pi_i$ from $\mathbb{R}^n$ onto $\mathrm{span}(b_i^*, b_{i+1}^*, \ldots, b_n^*)$ by

$$\pi_i(x) = \sum_{j=i}^{n} \frac{(x, b_j^*)}{(b_j^*, b_j^*)} b_j^*.$$

The following estimate is well-known for the first vector in a $\delta$-LLL reduced basis.

**Lemma 2.2.** *If $B = (b_1, \ldots, b_n) \in \mathbb{R}^{n\times n}$ is a $\delta$-LLL reduced basis with $\delta \in (1/4, 1)$, then*

$$\|b_1\|_2 \le \left(\frac{2}{\sqrt{4\delta - 1}}\right)^{n-1} \lambda_1(B). \tag{2.2}$$

Using the estimate (2.1), we obtain

$$\|b_1\|_2 \le \sqrt{n}\ |\det(B)|^{\frac{1}{n}} \left(\frac{2}{\sqrt{4\delta - 1}}\right)^{n-1}. \tag{2.3}$$

## 3. $p$-ADIC LATTICE

In this section we introduce $p$-adic approximation lattices and investigate simultaneous rational approximations of $p$-adic numbers. Let $p$ be a fixed rational prime number and $|\cdot|_p$ be the corresponding $p$-adic valuation, normalized so that $|p|_p = p^{-1}$. The completion of $\mathbb{Q}$ w.r.t. $|\cdot|_p$ is called the field of $p$-adic numbers, denoted by $\mathbb{Q}_p$. The strong triangle inequality

$$|a + b|_p \le \max\{|a|_p, |b|_p\}, \quad a, b \in \mathbb{Q}_p$$

is most important and essential to construct $p$-adic approximation lattices. The set of $p$-adic integers is defined by $\mathbb{Z}_p = \{z \in \mathbb{Q}_p : |z|_p \le 1\}$.

Let $n \ge 1$ be an integer and let $\Xi = \{\xi_1, \xi_2, \ldots, \xi_n\}$ be a $n$-tuple of $p$-adic integers.

**Definition 3.1.** We denote by $w_n(\Xi)$ the supremum of the real numbers $w$ such that, for some infinitely many real numbers $X_j$, which goes to infinity, the inequalities

$$0 < |a_{0,j} + a_{1,j}\xi_1 + \cdots + a_{n,j}\xi_n|_p \le X_j^{-w-1},$$
$$\max_{0\le i\le n} |a_{i,j}| \le X_j,$$

have a solution in integers $a_{0,j}, a_{1,j}, \ldots, a_{n,j}$.

*Remark* 3.2. For the case where $\xi_1 = \xi, \xi_2 = \xi^2, \ldots, \xi_n = \xi^n$ for a $p$-adic number $\xi$ the following results have been obtained (see [1]). $w_n(\Xi) = \min\{n, d-1\}$ holds if $\xi$ is algebraic of degree $d$ and $w_n(\Xi) \geq n$ for every $p$-adic number $\xi$, which is not algebraic of degree at most $n$. In [8] Sprindžuk proved that $w_n(\Xi) = n$ for almost all $\xi$ in the sense of Haar Measure.

For a positive integer $m$ we define the $p$-adic approximation lattice $\Gamma_m$ by

$$\Gamma_m = \{(a_0, a_1, \ldots, a_n) \in \mathbb{Z}^{n+1} : |a_0 + a_1\xi_1 + \cdots + a_n\xi_n|_p \leq p^{-m}\}. \tag{3.1}$$

When a $p$-adic integer $\xi_i$ has the $p$-adic expansion

$$\xi_i = \sum_{k=0}^{\infty} x_{i,k}p^k, \ 0 \leq x_{i,k} \leq p-1,$$

let $\xi_{i,m}$ be the $m$-th order approximation of $\xi_i$ defined by

$$\xi_{i,m} = \sum_{k=0}^{m-1} x_{i,k}p^k. \tag{3.2}$$

Consider the basis $\{b_{0,m}, b_{1,m}, \ldots, b_{n,m}\} \subset \mathbb{Z}^{n+1}$ of the lattice $\Gamma_m$ given by

$$b_{0,m} = (p^m, 0, \ldots, 0)^t, \quad b_{1,m} = (\xi_{1,m}, -1, 0, \ldots, 0)^t,$$
$$b_{2,m} = (\xi_{2,m}, 0, -1, 0, \ldots, 0)^t, \cdots, b_{n,m} = (\xi_{n,m}, 0, \ldots, 0, -1)^t.$$

In fact, we have $b_{k,m} \in \Gamma_m, \ \forall k$, since we can estimate

$$|\xi_{k,m} - \xi_k|_p \leq p^{-m}.$$

For $B_m = (b_{0,m}b_{1,m}\ldots b_{n,m})$ we have

$$B_m = \begin{pmatrix} p^m & \xi_{1,m} & \xi_{2,m} & \ldots & \xi_{n,m} \\ 0 & -1 & 0 & \ldots & 0 \\ 0 & 0 & -1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & -1 \end{pmatrix}, \quad |\det(B_m)| = p^m.$$

Applying the LLL algorithm for $\delta \in (1/4, 1)$, we denote $\{b_0, b_1, \ldots, b_n\}$ a reduced basis and $B = (b_0 \ b_1 \ \ldots \ b_n)$. It follows from (2.3) that the shortest vector $b_0$ in $B$ satisfies

$$\begin{aligned} \|b_0\|_2 &\leq \sqrt{n+1}\ |\det(B)|^{\frac{1}{n+1}} \left(\frac{2}{\sqrt{4\delta-1}}\right)^n \\ &= \sqrt{n+1}\ |\det(B_m)|^{\frac{1}{n+1}} \left(\frac{2}{\sqrt{4\delta-1}}\right)^n \\ &= \sqrt{n+1}\ p^{\frac{m}{n+1}} \left(\frac{2}{\sqrt{4\delta-1}}\right)^n. \end{aligned} \tag{3.3}$$

Furthermore, it is known that

$$\left(\prod_{i=0}^{n} \|b_i\|_2\right)^{\frac{1}{n+1}} \leq K_n |\det(B)|^{\frac{1}{n+1}} = K_n p^{\frac{m}{n+1}}, \quad K_n \sim 2^{O(n)} \tag{3.4}$$

for the reduced basis $\{b_0, b_1, \ldots, b_n\}$.

We can estimate the minimum norm value $\lambda_1^{(\infty)}(\Gamma_m)(= \lambda_1^{(\infty)}(L(B_m)))$ by using the famous Dirichlet principle (see [3]).

**Theorem 3.3.** *For a n-tuple of p-adic integers $\Xi = \{\xi_1, \ldots, \xi_n\}$, which are irrational and linearly independent over $\mathbb{Q}$, and each positive integer m, there exists a solution in integers $(a_{0,m}, a_{1,m}, \ldots, a_{n,m}) \in \mathbb{Z}^{n+1}$, which satisfies*

$$0 < |a_{0,m} + a_{1,m}\xi_1 + \cdots + a_{n,m}\xi_n|_p \leq p^{-m}, \tag{3.5}$$

$$\max_{0 \leq i \leq n} |a_{i,m}| \leq p^{\frac{m}{n+1}}. \tag{3.6}$$

*Consequently, we have*

$$\lambda_1^{(\infty)}(\Gamma_m) \leq p^{\frac{m}{n+1}} = \det(\Gamma_m)^{\frac{1}{n+1}} \tag{3.7}$$

*and*

$$w_n(\Xi) \geq n. \tag{3.8}$$

## 4. Numerical calculations on SAP

In this section we give the graphs which compare the numerical minimum and maximum values in the $l_\infty$ norm and the minimum values in the $l_2$ norm for the shortest vectors given by the LLL reduction basis and the values $X_m$ for the approximation orders $m$ from 5 to 40 and the dimensions $n = 10, 60, 70, 80$.

We investigate the following case.
$p = 13$: prime number,
$\xi_i = u_i^{\frac{1}{103}}$: $p$-adic number, 103rd root of $u_i$:
11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32,
33, 34, 35, 36, 37, 38, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 53, 54,
55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75,
76, 77, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 92, 93, 94, 95, 96, 97

Since the LLL reduction algorithm approximately finds the shortest vectors in the $l_2$ norm, we use their $l_\infty$ norm values as the substitutes of the shortest vectors in the $l_\infty$ norm. We use the following line styles in the graphs.

$- \cdot - \cdot - \cdot -\cdot$ : minimum norm values of the reduced basis vectors in $l_2$
- - - - - - - - : maximum norm values of the reduced basis vectors in $l_\infty$
———— : $X_m = p^{m/(n+1)}$
$\cdots\cdots\cdots\cdots$ : minimum norm values of the reduced basis vectors in $l_\infty$

These graphs show that the LLL algorithm is effective enough to obtain the solutions of SAP, which satisfy the estimate (3.6), if the dimension $n$ is under 60 (see Figure 1 and 2), but this estimate is not satisfied if $n \geq 80$ and $m \geq 30$ (see Figure 4 and [2], [7] ).

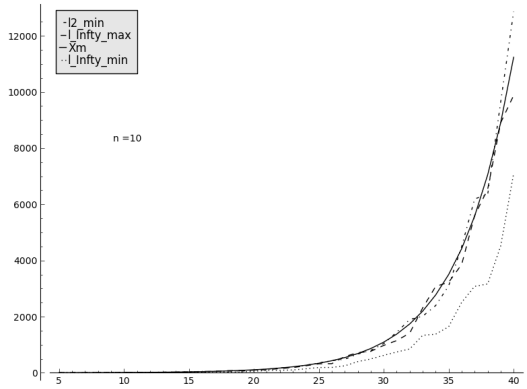


FIGURE 1. n=10

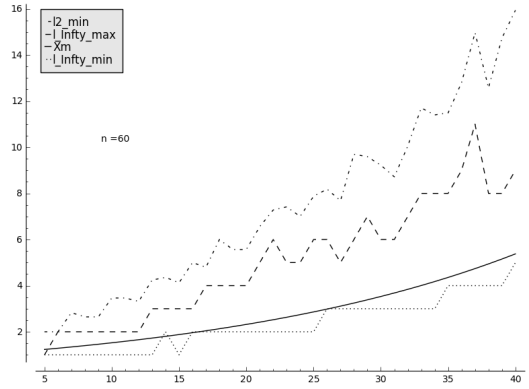


FIGURE 2. n=60

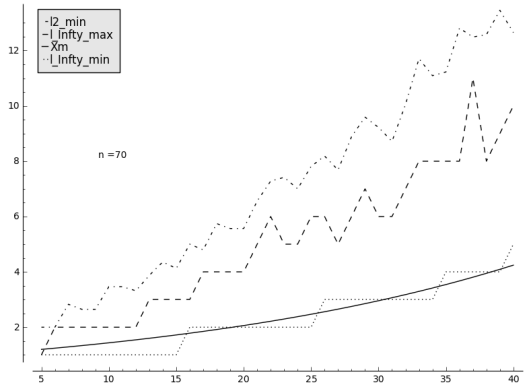


FIGURE 3. n=70

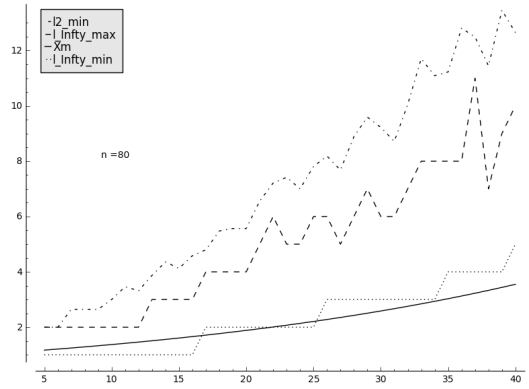


FIGURE 4. n=80

## 5. Cryptosystem I

In this section we propose a new cryptosystem, the security of which depends on the hardness of solving the SAP. Now we assume that Alice wants to send a message to Bob in this cryptosystem.

**Key generation**

First, we choose a prime number $p$ and $m \in \mathbb{N}$, which are the common private keys of Alice and Bob. For a public key we set a $l$-tuple of $p$-adic integers $\{\eta_1, \ldots, \eta_l\}$, which satisfies

$$|\eta_1|_p > |\eta_2|_p > \cdots > |\eta_l|_p, \ \eta := (\eta_1, \ldots, \eta_l), \tag{5.1}$$

and we construct a $n$-tuple of irrational $p$-adic integers $\{\xi_1, \ldots, \xi_n\}$ as a public key, linearly independent over $\mathbb{Q}$, and a $n+1$-tuple of rational integers $\{a_0, a_1, \ldots, a_n\}$ as a secret key, which satisfies $|a_i| \leq p^{m/(n+1)}, i = 0, \ldots, n$ and $|a_0 + a_1\xi_1 + \cdots + a_n\xi_n|_p \leq p^{-m}$ as follows.

We randomly choose the integers $a_0, \ldots, a_{n-1}$ which satisfy the condition $|a_i| \leq p^{m/(n+1)}, i = 0, \ldots, n-1$, and put $a_n = 1$. Next we randomly choose a linearly

independent $n$-tuple of $p$-adic integers $\{\xi_0, \xi_1, \ldots, \xi_{n-1}\}$, satisfying $|\xi_0|_p \leq p^{-m}$, and we define $\xi_n$ by $\xi_n = \xi_0 - (a_0 + a_1\xi_1 + \cdots + a_{n-1}\xi_{n-1})$. Then we have the set of these integers $\{a_0, \ldots, a_n\}$ becomes a solution of SAP :

$$0 < |a_0 + a_1\xi_1 + \cdots + a_n\xi_n|_p \leq p^{-m}, \tag{5.2}$$

$$\max_{0\leq i\leq n} |a_i| \leq p^{\frac{m}{n+1}}. \tag{5.3}$$

The security of the secret key $\{a_0, \ldots, a_n\}$ depends on the NP-hardness of the SAP.

**Encryption**

For a plaintext $\mathbf{x} = (x_1, \ldots, x_l) \in \{0,1\}^l$, Alice constructs its linear combination as a part of ciphertext $\mathbf{c}_0$ by

$$\mathbf{c}_0 := \mathbf{x} \cdot \eta = \sum_{i=1}^{l} x_i\eta_i.$$

By using $\eta = (\eta_1, \ldots, \eta_l)$, which satisfies (5.1), instead of the superincreasing sequence in the Knapsack cryptosystem Bob can easily decrypt the ciphertext $\mathbf{c}_0$ into the plaintext $\mathbf{x}$.

Alice constructs her ciphertext $\mathbf{c}$ by

$$\mathbf{c} = p^{-m}(a_0 + a_1\xi_1 + \cdots + a_n\xi_n) + \mathbf{c}_0$$

and she sends $\mathbf{c}$ to Bob.

**Decryption**

Bob obtains the part of the ciphertext $\mathbf{c}_0$ by using the public keys and the secret key from the ciphertext $\mathbf{c}$.

$$\mathbf{c} - p^{-m}(a_0 + a_1\xi_1 + \cdots + a_n\xi_n) = \mathbf{c}_0 = \mathbf{x} \cdot \eta.$$

The plaintext $\mathbf{x}$ is recovered from $\mathbf{c}_0$ step by steps as follows.

**1st-step:** If $|\mathbf{c}_0|_p \geq |\eta_1|_p$, then $x_1 = 1$, otherwise $x_1 = 0$.
**2nd-step:** If $|\mathbf{c}_0 - x_1\eta_1|_p \geq |\eta_2|_p$, then $x_2 = 1$, otherwise $x_2 = 0$.
$\vdots$
**$l$th-step:** If $|\mathbf{c}_0 - (x_1\eta_1 + \cdots + x_{l-1}\eta_{l-1})|_p \geq |\eta_l|_p$, then $x_l = 1$, otherwise $x_l = 0$.

Bob successfully gets the message from Alice.

The security of Cryptosystem I is not sufficient for practical usage, since the LLL attacks are available in the lower dimensions and we can easily have

$$\xi_n \equiv a_0 + a_1\xi_1 + \cdots + a_{n-1}\xi_{n-1} \mod p^m.$$

To increase its security we prepare Cryptosystem II in the following section and we propose some security measures in the last section. Here we give a simple partitioning method for a high dimensional case. For a given $(n+L)$-tuple of $p$-adic integers $\{\xi_i\}$ we randomly divide it into small $L$ subsets, given by $\cup_{k=1}^{L}\cup_{i=0}^{m_k}\xi_{i,k}, n = \sum_k m_k$. For each subset we can construct $m_k + 1$ number of the private keys $a_{i,k}$

and $m_k$ number of the public keys $\xi_{i,k}$ by using $p^m\xi_{0,k}$ and the same construction method as above. By the strong triangle inequality we have

$$|\sum_{k=1}^{L} a_{0,k} + \sum_{k=1}^{L}\sum_{i=1}^{m_k} a_{i,k}\xi_{i,k}|_p \le p^{-m}$$

where we can take a sufficiently small absolute value of each $a_{i,k}$.

## 6. Cryptosystem II (practical variations)

In this section we give some practical variations of Cryptosystem I to increase its security. Instead of the public keys of $p$-adic integers $\eta_1, \dots, \eta_l$, $p$-adic absolute values of which are decreasing, let $\eta_1, \dots, \eta_l$, be units and consider the two common secret keys, a permutation $\varphi(i) : \{1, \dots, l\} \to \{1, \dots, l\}$ and a strictly increasing sequence of positive integers $\{m_i\}_{1\le i\le l} :\ m_1 < m_2 < \cdots < m_l < m$.

Let the secret key $a_i$ be the sum of $\alpha_i$ and $\beta_i$, that is

$$a_i = \alpha_i + \beta_i,\ \alpha_i, \beta_i \in \mathbb{Z},\ i = 0, 1, \dots, n.$$

Alice has the secret key $\{\alpha_i\}$ and Bob has the secret key $\{\beta_i\}$.

**Encryption**

Alice constructs the part of the ciphertext $\mathbf{c}_0$ by

$$\mathbf{c}_0 = \sum_{i=1}^{l} x_i p^{m_{\varphi(i)}} \eta_{\varphi(i)}.$$

and she constructs the ciphertext $\mathbf{c}_A$ by

$$\mathbf{c}_A = \alpha_0 + \alpha_1\xi_1 + \cdots + \alpha_n\xi_n + \mathbf{c}_0.$$

**Decryption**

Bob takes the sum of $\mathbf{c}_A$ and the linear combination of $\{\xi_1, \dots, \xi_n\}$ with his secret key. Then he has

$$\begin{aligned}\mathbf{c}_A + \beta_0 + \beta_1\xi_1 + \cdots + \beta_n\xi_n &= a_0 + a_1\xi_1 + \cdots + a_n\xi_n + \sum_{i=1}^{l} x_i p^{m_{\varphi(i)}}\eta_{\varphi(i)} \\ &= a_0 + \sum_{j=1}^{n} a_j\xi_j + \sum_{i=1}^{l} x_{\varphi^{-1}(i)} p^{m_i}\eta_i := \mathbf{c}_B.\end{aligned}$$

Since $(a_0, \dots, a_n)$ is an integer solution of the SAP and $m > m_l > \cdots > m_1$, it follows from the isosceles principle that

$$|\mathbf{c}_B|_p = |a_0 + \sum_{j=1}^{n} a_j\xi_j + \sum_{i=1}^{l} x_{\varphi^{-1}(i)} p^{m_i}\eta_i|_p = |\sum_{i=1}^{l} x_{\varphi^{-1}(i)} p^{m_i}\eta_i|_p$$

if $\mathbf{x} \neq (0, 0, \dots, 0)$.

The plaintext $\mathbf{x}$ is recovered from $\mathbf{c}_B$ step by steps and Bob can easily recover the message $\mathbf{x}$ from $\mathbf{c}_0$ by using the secret keys $\varphi(i), \{m_i\}$.

**1st-step:** If $|\mathbf{c}_B|_p \ge p^{-m_1}$, then $x_{\varphi^{-1}(1)} = 1$,
otherwise $x_{\varphi^{-1}(1)} = 0$.

**2nd-step:** If $|\mathbf{c}_B - x_{\varphi^{-1}(1)}p^{m_1}\eta_1|_p \geq p^{-m_2}$, then $x_{\varphi^{-1}(2)} = 1$, otherwise $x_{\varphi^{-1}(2)} = 0$.

$\vdots$

**$l$th-step:** If $|\mathbf{c}_B - \sum_{i=1}^{l-1} x_{\varphi^{-1}(i)}p^{m_i}\eta_i|_p \geq p^{-m_l}$, then $x_{\varphi^{-1}(l)} = 1$, otherwise $x_{\varphi^{-1}(l)} = 0$.

**Permutation:** $\varphi : x_{\varphi^{-1}(i)} \to x_i,\ i = 1, \ldots, l$.

Bob successfully gets the message from Alice.

## 7. Numerical experiments

Using the open source software Sage, we can implement Cryptosystem I, II and we show how it works. We use some approximations of $p$-adic numbers in this numerical calculation on Sage. For a $p$-adic integer $\xi = \sum_{i=0}^{\infty} c_i p^i$ we use its approximation $\xi_M = \sum_{i=0}^{M-1} c_i p^i$ for some large $M \in \mathbb{N}$, which is called "precision", and we denote $\xi := \xi_M$ for simplicity. Here we consider the constant $M$ as a common private key.

### Cryptosystem I

#### Key generation

We choose parameters as follows.

$p = 13, n = 10, l = 8, m = 20, M = 50$,

The private key:

$\{a_0, ..., a_{n-1}\}$ : randomly taken, satisfying $|a_i| \leq p^{m/(n+1)}$, $a_n = 1$.

$p^{m/(n+1)} = 106$

$\{a_i\} = \{29, -9, -14, 43, -41, 38, 71, -74, -74, 0, 1\}$

$u_i = \{107, 3, 5, 29, 31, 53, 89, 101, 103, 106\}$,

$v_i = \{121, 126, 127, 129, 131, 133, 134, 139\}$, $\xi_i = u_i^{1/k}, \nu_i = v_i^{1/k}, k = 103$.

$\xi_0 = 107^{1/103} = 3 + 13 + 13^2 + 2 \cdot 13^3 + \cdots + 3 \cdot 13^{47} + 8 \cdot 13^{48} + O(13^{50})$,

$\xi_1 = 3^{1/103} = 3 + 9 \cdot 13 + 7 \cdot 13^2 + \cdots + 2 \cdot 13^{46} + 8 \cdot 13^{48} + O(13^{50})$,

$\vdots$

$\xi_9 = 106^{1/103} = 11 + 8 \cdot 13 + 9 \cdot 13^2 + \cdots + 5 \cdot 13^{47} + 3 \cdot 13^{48} + 6 \cdot 13^{49} + O(13^{50})$,

$\xi_{10} := \xi_0 - (a_0 + a_1\xi_1 + \cdots + a_9\xi_9)$

$= 2 + 6 \cdot 13 + 12 \cdot 13^2 + \cdots + 11 \cdot 13^{47} + 13^{48} + 5 \cdot 13^{49} + O(13^{50})$

Since each $\nu_i$ is a $p$-adic unit, we multiply $p^{i-1}$ by $\nu_i$ $(i = 1, \ldots, 8)$ to satisfy condition (5.1) and we put $\eta_i := p^{i-1}\nu_i$.

$\eta_1 = 4 + 4 \cdot 13^2 + 7 \cdot 13^3 + \cdots + 13^{47} + 8 \cdot 13^{48} + 10 \cdot 13^{49} + O(13^{50})$,

$\vdots$

$\eta_8 = 9 \cdot 13^7 + 5 \cdot 13^8 + 2 \cdot 13^{10} + \cdots + 12 \cdot 13^{47} + 7 \cdot 13^{48} + 6 \cdot 13^{49} + O(13^{50})$.

#### Encryption

For the plaintext $\mathbf{x} = (1, 0, 1, 1, 1, 1, 0, 1) \in \{0, 1\}^8$, Alice calculates the ciphertext

$$\begin{aligned}\mathbf{c} &:= p^{-20}(a_0 + a_1\xi_1 + \cdots + a_{10}\xi_{10}) + \mathbf{x} \cdot \eta \\ &= 3 + 13 + 2 \cdot 13^2 + 2 \cdot 13^3 + \cdots + 2 \cdot 13^{47} + 12 \cdot 13^{48} + 12 \cdot 13^{49} + O(13^{50})\end{aligned}$$

Then, Alice sends to Bob

$$\mathbf{c} = \mathbf{c}_M = 3 + 13 + 2 \cdot 13^2 + 2 \cdot 13^3 + \cdots + 2 \cdot 13^{47} + 12 \cdot 13^{48} + 12 \cdot 13^{49}.$$

**Decryption**

Bob calculates

$$\begin{aligned}\mathbf{x} \cdot \eta &= \mathbf{c} - p^{-20} \cdot (a_0 + a_1\xi_1 + \cdots + a_{10}\xi_{10}) \\ &= 4 + 13^2 + 5 \cdot 13^4 + 7 \cdot 13^5 + \cdots + 2 \cdot 13^{47} + 12 \cdot 13^{48} + 12 \cdot 13^{49} + O(13^{50})\end{aligned}$$

Applying the Knapsack-like decryption procedure for this $\mathbf{x} \cdot \eta$, Bob obtains the plaintext $\mathbf{x} = (1, 0, 1, 1, 1, 1, 0, 1)$ correctly.

The following tables show the processing times of Cryptosystem I for parameters $n$ and $l$.

TABLE 1. Processing times of cryptosystem I

| Parameter | Create Key 1 (secs) | Create Key 2 (secs) | Encrypt (millisecs) | Decrypt (millisecs) |
|---|---|---|---|---|
| $n = 20$ | 1.419 | 0.637 | 0.635 | 0.982 |
| 40 | 2.476 | 0.639 | 0.850 | 1.041 |
| 60 | 3.817 | 0.613 | 1.757 | 2.287 |
| 80 | 5.048 | 0.624 | 1.435 | 3.031 |
| 100 | 6.206 | 0.601 | 1.736 | 1.968 |

$(l = 10, m = 80, M = 110)$

TABLE 2. Processing times of cryptosystem I

| Parameter | Create Key 1 (secs) | Create Key 2 (secs) | Encrypt (millisecs) | Decrypt (millisecs) |
|---|---|---|---|---|
| $l = 20$ | 1.288 | 1.209 | 1.060 | 1.607 |
| 40 | 1.250 | 2.380 | 0.652 | 1.254 |
| 60 | 1.285 | 3.623 | 1.303 | 2.848 |
| 80 | 1.220 | 4.840 | 0.797 | 3.248 |
| 100 | 1.222 | 6.060 | 0.854 | 2.320 |

$(n = 20, m = 80, M = 110)$

Notes for Table 1 and Table 2:

(1) Key creation, Encryption and Decryption is implemented on Intel Core i7: 2×2.7 GHz, 4.0GB Memory machine. Operating system is Mac OS X 10.9.5 and Sage version is 6.2.
(2) Create Key 1 is the wall time of creating $(a_0, \ldots, a_n)$ and $\{\xi_1, \ldots, \xi_n\}$.
(3) Create Key 2 is the wall time of creating the key $\{\eta_1, \ldots, \eta_l\}$.
(4) The parameters $l = 10, m = 80, M = 110$ are fixed on the Table 1.
(5) The parameters $n = 20, m = 80, M = 110$ are fixed on the Table 2.

**Cryptosystem II**

The set of common private keys of Alice and Bob is $\{\{m_i\}, \varphi, p, m, M\}$.

**Key generation**

We choose parameters as follows.
$p = 13, n = 10, m = 20, M = 50$,
$u_i = \{61, 3, 5, 9, 12, 29, 31, 41, 50, 53\}$,
$v_i = \{121, 126, 127, 129, 131, 133, 134, 139, 140\}$, $\xi_i = u_i^{1/k}, \nu_i = v_i^{1/k}, k = 103$,
$\varphi$ : [[0, 7], [1, 8], [2, 2], [3, 5], [4, 3], [5, 1], [6, 6], [7, 4], [8, 0]]
$\{m_i\}$: $\{1, 2, 7, 9, 15, 17, 18, 19, 20\}$
$\xi_0 = 61^{1/103}, \xi_1 = 3^{1/103}, ..., \xi_9 = 53^{1/103}$, $\xi_{10} := \xi_0 - (a_0 + a_1\xi_1 + \cdots + a_9\xi_9)$.
The private keys:
$\{a_0, ..., a_{n-1}\}$: randomly taken, satisfying $|a_i| \leq p^{m/(n+1)}$, $a_n = 1$,
$\{a_i\} = \{-33, 81, 4, 22, 65, 30, 106, -55, 40, -91, 1\}$
Alice's private key:
$\{\alpha_i\} = \{-62, 83, -55, 50, 29, -70, -72, 66, 45, 91, -86\}$
Bob's private key:
$\{\beta_i\} = \{29, -2, 59, -28, 36, 100, 178, -121, -5, -182, 87\}$

**Encryption**

For the plaintext $\mathbf{x} = (1, 1, 1, 1, 0, 0, 0, 1, 1) \in \{0, 1\}^9$, Alice calculates the ciphertext

$$\begin{aligned}\mathbf{c}_A &= \alpha_0 + \alpha_1\xi_1 + \cdots + \alpha_n\xi_n + \sum_{i=1}^{l} x_i p^{m_{\varphi(i)}}\eta_{\varphi(i)} \\ &= 2 + 9 \cdot 13^2 + 4 \cdot 13^3 + \cdots + 9 \cdot 13^{47} + 12 \cdot 13^{48} + 11 \cdot 13^{49} + O(13^{50})\end{aligned}$$

Alice sends

$$\mathbf{c}_A = \mathbf{c}_{A,M} = 2 + 9 \cdot 13^2 + 4 \cdot 13^3 + \cdots + 9 \cdot 13^{47} + 12 \cdot 13^{48} + 11 \cdot 13^{49}$$

to Bob.

**Decryption**

Bob calculates

$$\begin{aligned}\mathbf{c}_B &:= \mathbf{c}_{A,M} + \beta_0 + \beta_1\xi_1 + \cdots + \beta_n\xi_n \\ &= 12 \cdot 13^{20} + 5 \cdot 13^{21} + 11 \cdot 13^{22} + \cdots \\ &\quad +10 \cdot 13^{47} + 6 \cdot 13^{48} + 12 \cdot 13^{49} + O(13^{50})\end{aligned}$$

Bob can get the $p$-adic absolute value of the message term by estimating the value $|\mathbf{c}_B|_p$ and using the isosceles principle. By applying the Knapsack-like decryption procedure for this $|\mathbf{c}_B|_p$ and using the common private keys $\varphi$ and $\{m_i\}$ he can obtain the plaintext $\mathbf{x} = (1, 1, 1, 1, 0, 0, 0, 1, 1)$ correctly.

## 8. Security considerations and concluding remarks

In our numerical calculations, while the $l_\infty$ norms of the SVP solutions given by LLL in the lattices of dimensions under 60 are less than the boundary value $p^{m/(n+1)}$

of the SAP solutions, in the lattices of dimension $n = 80$ these norms exceed this boundary value for the $p$-adic approximation order $m \geq 30$. So our private keys in Cryptosystem I given in the lattices of dimensions over 80 with $m \geq 30$ are considered to be secure against the LLL attacks. When $p = 2, n = 80$ and the precision size $M = 50$, the maximum size of public keys is equal to $p^{nM} \sim 2^{4000}$. Furthermore, if the prime numbers are larger than 5, their sizes exceed more than ten thousands bits. However, in our numerical calculations on the processing times it takes a few seconds ($< 10$ sec) to construct the public and private keys for the case $p = 13, n = 100, M = 110$ by $2 \times 2.7$ GHz CPU machine. (In preparation of this paper we found the remark in [6] on the run-time of exact SVP, quoted from [2], that up to dimension 60 the shortest vector problem could be solved within an hour, whereas dimension 100 seemed out of reach. )

In the lower dimensional case, $n = 10, p = 13, m = 20$, taking the linear combinations of the reduced basis vectors and the integer constants, which are arbitrarily taken under some suitable value, say $10^3$, we can obtain an extremely large number $\sim 10^{30}$ of candidates of the private keys. In this case a brute-force search attack requires almost $10^{30} (\sim 2^{100})$ exhaustive tries in the worst case to find a private key. When we put $M = m + 30$, we have $p^{nM} \sim 2^{1850}$ for $p = 13, n = 10, m = 20$. Here the key sizes are a few thousands bits, which are almost equal to the sizes of keys used in RSA, and the security level against offline attacks is over $2^{80}$ in this lower dimensional case. However, the keys constructions in our systems may contain a lot of severe problems for their practical applications when the parameters $p, m, M$ become large.

We can take these parameters $p, m, M$ as private keys. The offline attack on $p$ and $M$ by using the maximum value of a public key, which is given by $p^M$, requires many repeated tries, the number of which are given by primes from 2 to $q$ where $q$ satisfies $q^{10} \sim p^M$. Here we consider the case where the minimum value of the precision must be greater than or equal to 10. Using the prime number theorem, we can show that the number of the required tries of the offline attack is almost equal to $q/\log q \sim 5.3 \times 10^9$ for $p = 13, M = 100$. However, the extreme large prime numbers should cause a lot of problems on the processing times and the size of public keys. In our forthcoming tasks, further numerical and theoretical estimates on the numbers of the triple $\{p, m, M\}$, which satisfies some suitable conditions, will be required to estimate the security levels against the brute-force attacks on these private keys. We hope that Moore's law is correct and we will be able to easily construct the public and private keys of large size for the corresponding large parameters in the near future.

## References

[1] Y. Bugeaud, *Approximation by Algebraic Numbers*, Cambridge Tracts in Mathematics, Cambridge University Press, 2004.

[2] N. Gama and P. Q. Nguyen, *Predicting lattice reduction*, Nigel P. Smart:editor, EUROCRYPT, vol. 4965 Lecture Notes in Computer Science, Springer, 2008, pp. 31–51.

[3] H. Inoue and K. Naito, *The shortest vector problems in p-adic lattices and simultaneous approximation problems of p-adic numbers*, to appear in Linear Nonlinear Anal.

[4] J. C. Lagarias, *The computational complexity of simultaneous diophantine approximation problems*, SIAM Journal on Computing **14** (1985), 196–209.
[5] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems, a Cryptographic Perspective*, Springer International Series in Engineering and Computer Science, vol. 671. Springer, 2002
[6] P.Q. Nguyen and B. Vallee (Eds.), *The LLL Algorithm, Survey and Applications*, Springer, 2010.
[7] J. van de Pol and Nigel P. Smart, *Estimating key sizes for high dimensional lattice-based systems*, IMA Int. Conf., vol. 8308, Lecture Notes in Computer Science, Springer, 2013, pp. 290–303.
[8] V.G. Sprindžuk, *Mahler's problem in metric number theory*. Izdat. " Nauka i Tehnika", Minsk, 1967 (in Russian). English translation by B. Volkmann, Translations of Mathematical Monographs, Vol. 25, American Mathematical Society, Providence, R.I., 1969
[9] B.M.M. de Weger, *Approximation Lattices of p-adic Numbers*, J. Number Theory **24** (1986), 70–88.

---

*Manuscript received 2 March 2015*
*revised 4 November 2015*

Hirohito Inoue
Department of Applied Mathematics, Graduate School of Science and Technology, Kumamoto University, Kurokami 2-39-1, Kumamoto, Japan
*E-mail address*: hiro886@gmail.com

Shoichi Kamada
Department of Mathematics, Graduate School of Science and Technology, Kumamoto University, Chuo-ku, Kurokami 2-39-1, Kumamoto, Japan
*E-mail address*: 168d9309@st.kumamoto-u.ac.jp

Koichiro Naito
Department of Applied Mathematics, Graduate School of Science and Technology, Kumamoto University, Kurokami 2-39-1, Kumamoto, Japan
*E-mail address*: knaito@gpo.kumamoto-u.ac.jp