# Using the Inhomogeneous Simultaneous Approximation Problem for Cryptographic Design

Frederik Armknecht[1], Carsten Elsner[2], and Martin Schmidt[3]

[1] Universität Mannheim, 68161 Mannheim, Germany,
armknecht@informatik.uni-mannheim.de
[2] FHDW Hannover, 30173 Hannover, Germany,
carsten.elsner@fhdw.de
[3] Leibniz Universität Hannover, Institute of Applied Mathematics, 30167 Hannover, Germany,
mschmidt@ifam.uni-hannover.de

**Abstract.** Since the introduction of the concept of provable security, there has been the steady search for suitable problems that can be used as a foundation for cryptographic schemes. Indeed, identifying such problems is a challenging task. First, it should allow to build cryptographic applications on top of them. Second, it should be open and investigated for a long time to make its hardness assumption plausible. Third, it should be easy to construct hard problem instances. Not surprisingly, only a few problems are known today that satisfy all conditions, e. g., factorization, discrete logarithm, and lattice problems.

In this work, we investigate another candidate: the Inhomogeneous Simultaneous Approximation Problem (ISAP), an old problem from the field of analytic number theory. Although this problem is already known in cryptography, it has mainly been considered for *attacks* while we take a look at its hardness and applicability for cryptographic *design*. More precisely, we define a decisional problem related to ISAP, called DISAP, and show that it is NP-complete. As a starting point for concrete parameter ranges, we review the hardness of a related problem, being a computational and homogeneous variant of DISAP. Regarding the applicability, we describe as a proof of concept a bit commitment scheme where the hiding property is directly reducible to DISAP. An implementation confirms its usability in principle (e. g., size of one commitment is slightly more than 6 KB and execution time is in the milliseconds).

From our point of view, DISAP is an interesting problem that can be used for cryptographic designs. We hope to encourage further research on (D)ISAP in particular and possibly other problems from analytic number theory in general.

**Keywords:** Simultaneous Approximation Problem, Analytic Number Theory, Diophantine Approximation, Provable Security, Commitment Scheme

## 1 Introduction

**Motivation.** The concept of provable security is one cornerstone of modern cryptography. The approach is to prove the security of a cryptographic scheme

by reducing its security (in the sense of complexity theory) to another presumably hard problem. Consequently, there is a huge interest on finding appropriate problems. To be appropriate, at least the following conditions need to be met:

1. The problem is well-investigated since a long time, making the hardness conjecture trust-worthy.
2. Hard-to-solve instances can be easily constructed.
3. One can build cryptographic schemes upon them.

Different strategies are imaginable. One could start with known hard problems and look for cryptographic applications. Natural candidates are NP-complete problems, certainly meeting condition 1. However, it is not always clear how to construct hard-instances (condition 2). As an example take the homomorphic encryption scheme Polly Cracker by Fellows and Koblitz [5]. The security is based on the NP-complete problem of solving systems of nonlinear equations. But according to the current state of knowledge, all its instantiations (and variations like PollyTwo [32]) are either insecure, inefficient, or loose their homomorphic property (e. g., see [6, 17]). Another strategy could be to have a cryptographic scheme in mind and to clearly formalize the underlying problem. But then, only little may be known regarding conditions 1 and 2. It is often unclear if and to what extent newly introduced problems are examined once they have been introduced. Summing up, although a variety of problems[4] have been considered in the recent decades, only few of them fulfill all three conditions. Mainly these are connected to factorization, discrete logarithm, lattices, pairings, or error-correcting codes.

Observe that the first two, factorization and discrete logarithm, are probably the most established problems and belong both to *algebraic* number theory. Here, we would like to advert to *analytic* number theory, more precisely to the field of diophantine analysis. The adjective ”diophantine” means that one is interested in integral or rational solutions. This field emerged around 250 A.D. and had since then attracted the interest of many important and influential mathematicians like Gauss or the Fields medal winners Roth, Baker, and Faltings. Despite the enormous progress, diophantine analysis is still full of open (computational) problems. As a representative, we investigate the Simultaneous Approximation Problem (SAP) or more precisely its inhomogeneous variant (ISAP). In a nutshell, SAP (also called the Simultaneous Diophantine Approximation Problem) is to approximate a set of values by rational numbers which all share the same denominator.

[4] See the website `www.ecrypt.eu.org/wiki/index.php/Hard_Problems_in_Cryptography` for an overview.

**Related work.** SAP is known in cryptography, but has mainly been considered for *attacking* cryptosystems, e. g., knapsack systems (e. g., Shamir [29], Lagarias [13]), factorization and discrete logarithm (e. g., see Schnorr [24], Seifert [28]), and RSA (e. g., see Wiener [33]).

Regarding the *design* of cryptosystems, we are only aware of the works of Isselhorst [10] and Elsner and Schmidt [4]. Isselhorst [10] presented a public-key scheme based on fractions. He showed that the scheme could be broken in principle by solving an appropriate simultaneous approximation problem. He proposed parameters for which he suspected that the algorithm of Lagarias [14] is not capable of finding a solution. Nonetheless, the scheme was broken soon after by Stern and Toffin [30] using the LLL algorithm [15] instead. Elsner and Schmidt used continued fractions to design new S-boxes. In both cases, there was no direct reduction of the security of the scheme to the hardness of solving (I)SAP.

We want to point out that there *might* be a relation between (I)SAP and some of the lattice-based problems as both can be tackled by the LLL algorithm in principle. However, we are not aware of any result in this direction. Furthermore, the LLL algorithm applies to the homogeneous problem (SAP) only while we consider the inhomogeneous variant (ISAP) on purpose: We need the inhomogeneity for formulating appropriate problem instances. Thus, we leave the investigation of the connection between (I)SAP and lattice based problems as an open question and consider ISAP as a problem that has not been used for cryptographic design so far.

**Contribution.** In this paper, we put for the first time ISAP into the heart of a cryptosystem. Our contributions are as follows:

- *Problem Description:* We formalize/repeat the Decisional Inhomogeneous Simultaneous Approximation Problem (DISAP) and show that it is NP-complete.
- *Instance Generation:* We argue that increasing/decreasing certain parameters will probably increase the hardness of the problem and formulate an accordant assumption. Furthermore, we investigate a related computational problem and derive suggestions for concrete parameter ranges.
- *Cryptographic Application:* We demonstrate the usefulness of DISAP for cryptographic applications by constructing a bit commitment scheme on it. The scheme is perfectly binding and computationally hiding if hard DISAP instances are used.

Summing up, we demonstrate that DISAP might be a valuable addition to the existing set of established problems in cryptography and hope to encourage fur-

ther research on problems from analytic number theory in general and DISAP in particular.

**Organization.** In Sec. 2, we present DISAP and discuss its hardness. In addition, we define and motivate an appropriate hardness assumption, named DISAP assumption. In Sec. 3, we describe a bit commitment scheme based on DISAP. Its security is proven in Sec. 4 under the DISAP assumption. In Sec. 5, we present a concrete instantiation and give implementation results. Sec. 6 concludes the paper.

## 2 The Inhomogeneous Simultaneous Approximation Problem

### 2.1 Motivation and Definition

In this section we give a short introduction to the main terms of rational diophantine approximation and motivate and define the Inhomogeneous Simultaneous Approximation Problem (ISAP). In the following, $\mathbb{N}$ will denote the set of positive integers, $\mathbb{Z}$ the ring of integers, $\mathbb{Q}$ the field of rational numbers, and $\mathbb{R}$ the field of the real numbers. We will distinguish between single values and vectors by putting the latter in bold.

In diophantine analysis the approximation of numbers $\alpha \in \mathbb{R}$ by rationals $p/q \in \mathbb{Q}$ is a main topic.[5] One of the most basic results is the approximation theorem of Dirichlet (1805–1859) [9, Theorem 185], which states that for any $\alpha \in \mathbb{R} \setminus \mathbb{Q}$ there exist infinitely many co-prime numbers $p$ and $q$ such that

$$\left|\alpha - \frac{p}{q}\right| < \frac{1}{q^2} \quad \Longleftrightarrow \quad |q\alpha - p| < \frac{1}{q}\,. \tag{1}$$

If $\alpha \in \mathbb{Q}$, the number of solutions might be finite only. A theorem of Hurwitz (1859–1919) [9, Theorem 193] states that for every $\alpha \in \mathbb{R} \setminus \mathbb{Q}$ there exist infinitely many co-prime numbers $p$ and $q$ such that

$$\left|\alpha - \frac{p}{q}\right| < \frac{1}{\sqrt{5}q^2} \tag{2}$$

holds and that for any stronger approximation quality, the number of solutions might be finite only. Interestingly, such approximations can be efficiently com-

[5] Nice introductions to this discipline can be found in [9, 18].

puted, using *continued fractions*

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots + \cfrac{1}{a_N}}}} \tag{3}$$

where the leading coefficient $a_0$ is an integer and all *partial quotients* $a_i\,(i = 1, \ldots, N)$ are positive integers. It can be shown that for $N \to \infty$ the above given expression converges to some real number $\alpha$ depending on all partial quotients $a_i$. In that case we call the expression an *infinite continued fraction*, or simply *continued fraction*. For $\alpha \in \mathbb{Q}$, the corresponding continued fraction is *finite* like in (3).

An important term is a *convergent*, which is a rational number. Given the partial quotients of the continued fraction, the corresponding convergents can easily be computed using the recurrence formulas (see [9, Theorem 149])

$$p_0 = a_0\,, \qquad p_1 = a_1 a_0 + 1\,, \qquad p_n = a_n p_{n-1} + p_{n-2} \qquad (n \geq 2)\,, \tag{4}$$

$$q_0 = 1\,, \qquad q_1 = a_1\,, \qquad q_n = a_n q_{n-1} + q_{n-2} \qquad (n \geq 2)\,. \tag{5}$$

$p_n/q_n$ is called the $n$-th convergent of the continued fraction. Observe that computing the $n$-th convergent requires $2n$ additions and multiplications of integers. It can be shown that (for irrational $\alpha$ or, if $\alpha \in \mathbb{Q}$, $n < N$)

$$\frac{1}{q_n(q_n + q_{n+1})} \leq \left|\alpha - \frac{p_n}{q_n}\right| \leq \frac{1}{q_n q_{n+1}} \leq \frac{1}{q_n^2} \tag{6}$$

holds (see [18, Chapter 10.2]). We note that from (6) it follows that the convergents satisfy inequality (1) of Dirichlet's theorem. Furthermore, it is proven that the convergents are the best rational approximations with a bounded denominator, e. g., for $\alpha \in \mathbb{R}, n > 1, 0 < q \leq q_n$ and $p_n/q_n \neq p/q$ it holds (see [9, Theorem 181])

$$\left|\alpha - \frac{p_n}{q_n}\right| < \left|\alpha - \frac{p}{q}\right|\,. \tag{7}$$

It is useful to know that $p_n$ and $q_n$ are co-prime for all convergents.

The type of approximation in (1) is called *homogeneous* in contrast to the inhomogeneous case, for which Kronecker (1828–1891) proved the following theorem (see [21, Chapter 10, Theorem 2.6]).

**Theorem 1 (Kronecker's Approximation Theorem).** *For each $\alpha \in \mathbb{R} \setminus \mathbb{Q}$, $\eta \in \mathbb{R}$, $n > 0$ and $\delta \in \mathbb{R}$ with $\delta > 0$ there are integers $p, q$ with $q > n$ such that*

$$|q\alpha - p - \eta| < \left(\frac{1}{2} + \frac{1}{\sqrt{5}} + \delta\right)\frac{1}{q}\,. \tag{8}$$

Thereby $\eta$ is called the *inhomogeneity*.

In the field of simultaneous diophantine approximation one considers more than one diophantine inequality at once and tries to approximate the given numbers $\alpha_i$ with fractions $p_i/q$ sharing a common denominator. Again, the most basic result was proved by Dirichlet (see [9, Theorem 200]): There are infinitely many solutions $(q, p_1, \ldots, p_n)$ to the system

$$\left|\alpha_i - \frac{p_i}{q}\right| < \frac{1}{q^{1+1/n}}\,, \quad \forall i \in \{1, \ldots, n\}\,, \tag{9}$$

in positive integers $q$ and integers $p_1, \ldots, p_n$ if at least one of the real numbers $\alpha_1, \ldots, \alpha_n$ is irrational. An inhomogeneous generalization about the existence of simultaneous approximations was also proved by Kronecker (see [9, Theorem 442]):

**Theorem 2 (Kronecker's Simultaneous Approximation Theorem).** *Let* $1$*,* $\alpha_1$*, ...,* $\alpha_n$ *be real numbers that are linearly independent over* $\mathbb{Q}$*. Furthermore, let* $\eta_1, \ldots, \eta_n$ *be arbitrary real numbers,* $\varepsilon > 0$ *and* $N \in \mathbb{N}$*. Then there exists integers* $p_1, \ldots, p_n$ *and a natural number* $q$ *with* $q > N$ *and*

$$|q\alpha_i - p_i - \eta_i| < \varepsilon \quad \forall i \in \{1, \ldots, n\}\,. \tag{10}$$

We remark that the one-dimensional theorems can all be proved in a constructive manner by using continued fractions and convergents. However, as opposed to the one-dimensional case, no constructive proofs are known for the simultaneous versions. This has lead to the formulation of a variety of related problems, e. g. [14], for some it has been proven that they are NP-complete. Nevertheless, none of them have been successfully used for cryptographic applications. The main goal of this paper is to actually remind of these and to choose one concrete problem formulation and demonstrate a possible cryptographic application. The considered basic problem is the following:

**Definition 1 (Inhomogeneous Simultaneous Approximation Problem (ISAP)).** *An instance* $I := (\boldsymbol{\alpha}, \boldsymbol{\eta}, N, n, \varepsilon)$ *of the Inhomogeneous Simultaneous Approximation Problem (ISAP) consists of a vector* $\boldsymbol{\alpha} := (\alpha_1, \ldots, \alpha_n) \in (\mathbb{Q}^*)^n$ *of non-zero rational values, a vector* $\boldsymbol{\eta} := (\eta_1, \ldots, \eta_n) \in \mathbb{Q}^n$*, a positive real value* $\varepsilon \in \mathbb{R}_{>0}$*, and a positive integer* $N \in \mathbb{N}$*.*

*A tuple* $(q, \boldsymbol{p})$ *where* $q \in \mathbb{N}_{>0}$ *and* $\boldsymbol{p} = (p_1, \ldots, p_n) \in \mathbb{Z}^n$ *is a* solution *to* $I$ *if*

$$|q\alpha_i - p_i - \eta_i| < \varepsilon \quad \forall i \in \{1, \ldots, n\} \quad \textit{and} \quad q \leq N\,. \tag{11}$$

*The value* $n$ *is called the* dimension*, and* $\varepsilon$ *the* approximation quality*. In the case that* $\eta_i = 0$ *for all* $i$*, that is in the homogeneous case, we call the problem simply the* Simultaneous Approximation Problem *(SAP).*

Although the dimension $n$ is implicitly given by the dimension of the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$, we note it explicitly for reasons of clarity. Observe that we restrict to rational and integer values on purpose: Working in practice with irrational numbers effectively means in most cases to approximate them anyway by rational numbers. We formulate a decisional problem in the context of ISAP that we will eventually propose for cryptographic design:

**Definition 2 (Decisional ISAP (DISAP)).** *Let $I$ be an ISAP-instance as explained in Def. 1. The Decisional ISAP (DISAP) is to decide whether $I$ has at least one solution.*

Next, we show that the problem class of DISAP contains indeed hard instances, i. e., instances where no efficient solving algorithms are known so far.

**Theorem 3.** *DISAP is NP-complete.*

*Proof.* We have to show that (i) DISAP is in NP, and (ii) every problem in NP is reducible to DISAP in polynomial time. The first claim is trivial. Given an DISAP instance $(\boldsymbol{\alpha}, \boldsymbol{\eta}, N, n, \varepsilon)$ and a possible solution $(q, \boldsymbol{p})$, one can check in polynomial time (i.e., polynomial in the length of the input) whether (11) is fulfilled. For the second claim, we make use of Lagarias' result [14]. He showed that DSAP[6] is NP-complete (the problem was named "Good Simultaneous Approximation problem (GSA)" there). That is, any NP problem can be reduced (in polynomial time) to an instance of DSAP. As any instance of DSAP is an instance of DISAP as well, it follows directly that any NP problem can be reduced in polynomial time to an instance of DISAP. □

Still, it remains to clarify how to generate hard instances. It is plausible to assume that increasing the dimension $n$ and/or chosing a sharper approximation quality $\varepsilon$, i.e., decreasing this value, can make the problem only harder. This motivates the following assumption:

**Definition 3 (DISAP Assumption).** *Consider a probabilistic polynomial-time (PPT) algorithm* $\mathsf{Gen}$ *that on input $N \in \mathbb{N}_{>0}$, $n \in \mathbb{N}_{>0}$, and $\varepsilon \in \mathbb{R}_{>0}$ generates an ISAP instance $I := (\boldsymbol{\alpha}, \boldsymbol{\eta}, N, n, \varepsilon)$ where $\forall i, j = 1, \ldots, n : (\alpha_i, \eta_i) \neq (\alpha_j, \eta_j)$ if $i \neq j$. Let $\mathcal{I}$ denote the set of all possible ISAP instances that can be generated by* $\mathsf{Gen}$*. We define a predicate $P : \mathcal{I} \to \{0, 1\}$ on $\mathcal{I}$ such that $P(I) = 1$ if and only if $I$ has a solution. For an algorithm $\mathcal{A}$ we define its advantage (with respect to* $\mathsf{Gen}$*) as*

$$\begin{aligned}\mathsf{Adv}_{\mathsf{Gen},\mathcal{A}}(N, n, \varepsilon) := |&Pr[I \leftarrow \mathsf{Gen}(N, n, \varepsilon), P(I) = 0 : \mathcal{A}(I) = 0] - \\ &Pr[I \leftarrow \mathsf{Gen}(N, n, \varepsilon), P(I) = 1 : \mathcal{A}(I) = 0]|\end{aligned}$$

[6] By DSAP, we refer to the straightforward restriction of DISAP to the homogenous case. In other words, a DSAP instance is a DISAP instance where $\boldsymbol{\eta} = \mathbf{0}$.

*The* decisional ISAP assumption *(with respect to* $\mathsf{Gen}$*) states that for any positive integer* $s \in \mathbb{N}_{>0}$*, being eventually the security parameter, there exist thresholds* $N^* = N^*(s) \in \mathbb{N}_{>0}$*,* $n^* = n^*(s) \in \mathbb{N}_{>0}$ *and* $\varepsilon^* = \varepsilon^*(s) \in \mathbb{R}_{>0}$ *such that* $\mathsf{Adv}_{\mathsf{Gen},\mathcal{A}}(N, n, \varepsilon)$ *is negligible in* $s$ *for all PPT* $\mathcal{A}$ *if* $N \geq N^*$*,* $n \geq n^*$*, and* $0 < \varepsilon \leq \varepsilon^*$*.*

### 2.2 Possible Parameter Choices

As explained in the previous section, a promising strategy for creating hard instances is to choose values $N$, $n$ and $\varepsilon$ which are beyond certain thresholds. Unfortunately, as DISAP has not been considered directly so far, nothing is known about concrete choices for these thresholds. However, some indication on possible choices can be derived from the fact that a somewhat related problem has been investigated since long. The key to this approach is the following Theorem.

**Theorem 4.** *Assume an algorithm* $\mathcal{A}$ *that is able to compute solutions (if existent) to ISAP-instances* $(\boldsymbol{\alpha}, \boldsymbol{\eta}, N, n, N^{-\delta})$ *where* $\eta_i = \frac{\lambda_i}{\mu_i}$ *with* $0 < \mu_i \leq N^{\delta'}$ *and* $0 < \delta' \leq \delta$*. Then, there exists another algorithm* $\mathcal{B}$ *with the following property: Given* $(\boldsymbol{\beta}, N, n)$ *with* $\boldsymbol{\beta} \in \mathbb{Q}^n$*, invoke* $\mathcal{A}$ *such that any solution* $(q, \boldsymbol{p})$ *returned by* $\mathcal{A}$ *implies values* $\tilde{q} \in \mathbb{N}_{>0}$ *and* $\tilde{\boldsymbol{p}} = (\tilde{p}_1, \ldots, \tilde{p}_n) \in \mathbb{Q}^n$ *such that*

$$\left| \beta_i - \frac{\tilde{p}_i}{\tilde{q}} \right| < \frac{1}{\tilde{q}^{1+(\delta-\delta')}} \quad \forall i \in \{1, \ldots, n\} \,. \tag{12}$$

*Here, the term* $\Delta := 1 + \delta - \delta'$ *is called the* approximation order.

*Proof.* Let $(\boldsymbol{\beta}, N, n)$ be given as defined above. At first, $\mathcal{B}$ chooses some values $0 < \mu_i \leq N^{\delta'}$ and sets $\alpha_i := \beta_i / \mu_i \in \mathbb{Q}$. Furthermore, some positive integers $\lambda_i \in \mathbb{N}_{>0}$ are sampled according to some arbitrary distribution and $\eta_i := \lambda_i / \mu_i$ are defined. Then, $\mathcal{B}$ hands the ISAP-instance $(\boldsymbol{\alpha}, \boldsymbol{\eta}, N, n, N^{-\delta})$ to $\mathcal{A}$. Assume that $\mathcal{A}$ returns a solution $(q, \boldsymbol{p})$. $\mathcal{B}$ sets $\tilde{q} := q$ and $\tilde{p}_i := p_i \cdot \mu_i + \lambda_i$ and outputs $(\tilde{q}, \tilde{\boldsymbol{p}})$.

We show now that $(\tilde{q}, \tilde{\boldsymbol{p}})$ meets condition (12). By assumption, the response $(q, \boldsymbol{p})$ of $\mathcal{A}$ is a solution to the ISAP instance, i.e., $|q\alpha_i - p_i - \eta_i| < N^{-\delta}$ for $i = 1, \ldots, n$. Because of $\mu_i \leq N^{\delta'}$ and $q \leq N$, we have $\frac{1}{N^{\delta}} = \frac{1}{N^{\delta-\delta'} \cdot N^{\delta'}} \leq \frac{1}{q^{\delta-\delta'} \cdot \mu_i}$. Thus, one can show that

$$\left| q\alpha_i - \frac{\tilde{p}_i}{\mu_i} \right| = |q\alpha_i - p_i - \eta_i| < \frac{1}{N^{\delta}} \leq \frac{1}{q^{(\delta-\delta')} \cdot \mu_i} \tag{13}$$

$$\overset{\tilde{q}=q}{\Longrightarrow} \left| \beta_i - \frac{\tilde{p}_i}{\tilde{q}} \right| < \frac{1}{\tilde{q}^{1+\delta-\delta'}} \,. \tag{14}$$

Therefore, the output of $\mathcal{B}$ indeed represents a solution to (12). □

In the remainder of this section, we will derive parameter ranges where the problem explained above seems to be hard according to the current state of knowledge. First, we explain the implications for DISAP. For the sake of brevity, let us introduce some abbreviations here. By CISAP, we refer to the computational counterpart to DISAP where the challenge is to *compute* a solution instead of *deciding* the existence of a solution. Furthermore, let CSAP* denote the homogeneous variant of CISAP as expressed by Eq. (12), that is where the approximation quality $\varepsilon = q^{-\Delta}$ depends on the solution $q$. Thus, if we derive parameters where it seems that no solutions to CSAP* can be found, this includes the infeasability of finding solutions implied by CISAP. As any solution of appropriate CISAP instances imply solutions to CSAP*, this excludes the existence of efficient algorithms for CISAP (at least for the cases where some of the solutions to CSAP* can be found via CISAP). On the other hand, the infeasability of CISAP is a necessary condition for the hardness of DISAP. Therefore, adopting the parameters derived from CSAP* for DISAP and considering instances as described in Th. 4 seems to be a promising starting point for creating presumably hard instances of DISAP. We leave the determination of more appropriate values as an open question.

There are several algorithms in the literature to solve CSAP*. In the case of real algebraic and over $\mathbb{Q}$ linear independent numbers $1, \alpha_1, \ldots, \alpha_n$ and $\delta^* > 0$ arbitrary, W. Schmidt shows in [23] that there are at most finitely many $(q, \boldsymbol{p}) \in \mathbb{N} \times \mathbb{Z}^n$ with

$$\left| \alpha_i - \frac{p_i}{q} \right| < \frac{1}{q^{1+1/n+\delta^*}} \quad \forall i \in \{1, \ldots, n\} \ . \tag{15}$$

Furthermore, with $\delta^* = 0$, under these conditions the approximation order $\Delta = 1 + 1/n$ is the best possible.

There are a lot of generalizations of continued fractions for the simultaneous case, starting with the work of Jacobi [11] which lead to the Jacobi-Perron-Algorithm (JPA) [20, 27, 25, 2, 8]. However, the JPA is not able to compute solutions to such approximation quality as we will require in our proposed commitment scheme (cf. Sec. 3). For example, in the case $n = 2$ only a system with an approximation quality of $2/q^{3/2}$ is attackable with the JPA (cf. [27]). In [27] it is also mentioned that the JPA is only able to solve systems with significantly larger $\varepsilon$ in the arbitrary case ($n \geq 3$). In particular, the best affordable approximation quality $\varepsilon$ increases with the dimension $n$. Additionally, we want to mention Baldwin's numerical experiments [1] in which he computes the approximation exponent of the JPA in two dimensions – with $\Delta = 1.374$ it is significantly below the upper bound $1 + 1/2 = 1.5$ from theory.

There are some other relevant algorithms based on continued fraction generalizations, namely the ones of Güting, Brun, Selmer, and Just. The first three

ones have comparable properties like the JPA (see [27, 3, 26, 31]). Just's algorithm is much more worse concerning the approximation order ($\Delta = 1 + 1/(2n(n+1))$) [12]. Thus, the above given considerations about the JPA can also be applied to these algorithms.

Another well known algorithm for solving simultaneous diophantine approximation problems is the lattice-based LLL algorithm presented by Lenstra, Lenstra Jr. and Lovasz in [15]. The LLL algorithm is able to find solutions nearly as good as the best possible. Indeed, they can compute solutions $(q, \boldsymbol{p})$ such that

$$\left|\alpha_i - \frac{p_i}{q}\right| \leq \frac{c(n)}{q^{1+1/n}} \quad \forall i \in \{1, \ldots, n\} \;, \tag{16}$$

whereas the $\alpha_i$ have to be rationals and $c(n) \in \mathcal{O}\left(2^n\right)$ (see [15], [19, Chapter 6, Theorem 8]). Thus, by choosing small enumerators for the upper bound, e.g., $= 1$ in our construction, one can construct instances that seem to fall outside of the parameter ranges that can be solved by LLL.

We conclude with the theoretical work of Lagarias. In [14] he proved that the problem of computing a denominator $q$ such that

$$|\alpha_i q - p_i| \leq s_1/s_2 \,, \quad 1 \leq q \leq N \,, \quad i = 1, \ldots, n \,, \tag{17}$$

for given positive integers $N, s_1, s_2$ and rational numbers $\alpha_i = a_i/b_i$ is in P for fixed dimension $n$. We remark that this technique has an exponential runtime in the dimension $n$. Thus, increasing $n$ is a simple method for excluding the applicability of Lagarias' algorithm.

Summing up, no efficient algorithms are known for solving CSAP* with an approximation order of $\Delta \geq 1 + 1/n$ if the dimension is high enough. In Sec. 5, we will use this observation for proposing some concrete parameters. More precisely, we will construct instances as explained in Th. 4 where the approximation quality and the dimension are too high for all algorithms mentioned above. Regarding the upper bound $N$, one has to take care that it is big enough for excluding brute force approaches. We will set $N := 2^s$ in our construction where $s$ represents the security parameter. Observe that in our scheme, we construct instances that have only one unique solution. Hence, it will not be possible to look for other solutions that might be easier to find. In this context we would like to refer to the results by Rössner and Seifert [22]: They showed that approximating the best solution is almost NP-hard. Thus, approximating the unique solution $q$ seems to be not an option either.

## 3 A Bit Commitment Scheme based on DISAP

In this section, we present a bit commitment scheme based on DISAP. In the commitment phase, the committer generates an instance of DISAP with a given

dimension and approximation quality. The crucial aspect here is that the problem instance is constructed *backwards*. That is the committer first starts with the solution $(q, \boldsymbol{p})$ that is connected to the message and then generates a problem instance $(\boldsymbol{\alpha}, \boldsymbol{\eta})$ from it where $(q, \boldsymbol{p})$ is the unique solution. Observe that the generation procedure allows for choosing the parameters outside the range that is feasible for the algorithms described in Sec. 2.2 in the *normal* direction and ensures that the instances are of the form as described in Th. 4. For this purpose, we strongly make use of the inhomogeneity $\eta$. Regarding the security, the commitment scheme is computationally hiding if the DISAP assumption holds. Furthermore, as only one solution exists, the scheme is perfectly binding.

**Setup Phase.** In the setup phase, an algorithm

$$\mathcal{P} := (N, \varepsilon, n, \mu) \leftarrow \mathsf{Setup}(s) \tag{18}$$

is executed. The purpose of this algorithm is to fix in dependence of a security parameter $s$ the bound $N$, the approximation quality $\varepsilon$, the dimension $n$, and an upper bound $\mu$ on the denominators of $\boldsymbol{\eta}$ for DISAP instances that will be used in the other phases of the commitment scheme. Starting from the DISAP assumption (Def. 3), these are chosen such that $N \geq N^*(s)$, $\varepsilon \leq \varepsilon^*(s)$ and $n \geq n^*(s)$ where $N^*(s)$, $\varepsilon^*(s)$, and $n^*(s)$ are the thresholds conjectured in the DISAP assumption (Def. 3). More precisely, we will fix $N := 2^s$ to avoid brute force guessing attacks. The bound $\mu$ will be set as described in Th. 4. We will discuss concrete parameter choices later in Sec. 5.

**Commitment Phase.** In this phase, the committer generates a commitment for a message $m \in \{0, 1\}$. The commitment algorithm has the following format:

$$((\boldsymbol{\alpha}, \boldsymbol{\eta}), (q, \boldsymbol{p})) \leftarrow \mathsf{Commit}_{\mathcal{P}}(m) \tag{19}$$

where $(\boldsymbol{\alpha}, \boldsymbol{\eta}, N, n, \varepsilon)$ specifies an instance of DISAP as defined in Def. 1 and $(q, \boldsymbol{p})$ is a solution to this instance. The tuple $(\boldsymbol{\alpha}, \boldsymbol{\eta})$ represents the commitment to the message $m$ which is made public. The tuple $(q, \boldsymbol{p})$ represents the opening information and is kept secret. The value $q$ is constructed in such a way that its least significant bit (LSB) is equal to the message $m$.

The commitment algorithm is depicted in Alg. 1. During an execution, a series of values are generated that have to fulfill certain conditions. For the sake of clarity, we separated in the description of Alg. 1 the value generation and the testing of the parameters. In real implementations, one would group this steps together to reduce the number of trials. For example, if parameter generation fails for one index $i$, one could retry other values for this index but still use the

values generated for indices $j < i$. We have to point out that it is not mathematically guaranteed that all conditions can be met. However, this was straightaway the case in almost all of our simulations (see Sec. 5 for details). Furthermore, in all other cases a small number of repetitions was sufficient to find values that fulfill the conditions.

Finally, some words on the conditions themselves. The condition $\frac{1}{\sqrt{\varepsilon}} < d_i$ (Eq. (20)) is introduced to achieve the claimed approximation quality with the given solution. The other part of the same inequality, $d_i < b_i$, is used to guarantee that the approximation $c_i/d_i$ does not give $q \cdot a_i/b_i$ again. The last conditions, given in Eq. (21), ensures that the value $q$ is uniquely determined, making the scheme perfectly binding.

**Opening Phase.** To open the commitment, the committer sends the solution $(q, \boldsymbol{p})$ to the verifier. The verifier runs the algorithm

$$\mathrm{out} \leftarrow \mathsf{Verify}_{\mathcal{P}}\left((\boldsymbol{\alpha}, \boldsymbol{\eta}), (q, \boldsymbol{p})\right) \tag{22}$$

where $\mathrm{out} \in \{\mathsf{accept}, \perp\}$. The verifier accepts if $\mathrm{out} = \mathsf{accept}$ and rejects otherwise. The algorithm $\mathsf{Verify}_{\mathcal{P}}$ outputs $\mathsf{accept}$ if and only if

1. $|q\alpha_i - p_i - \eta_i| \leq \varepsilon$ for all $i \in \{1, \ldots, n\}$
2. There exists an index $i^* \in \{1, \ldots, n\}$ such that $N < b_{i^*}$ and $\sqrt{2b_{i^*}} < d_{i^*}$ (see Eq. (21)). Observe that the values $b_i$ are part of the commitment and the values $d_i$ can be computed from $\eta_i$ and $p_i$ by using that $c_i$ and $d_i$ are co-prime (see Sec. 2.1).

**Correctness.** The correctness of the scheme follows directly from condition $\frac{1}{\sqrt{\varepsilon}} < d_i$ (see Eq. (20)) given in Alg. 1. For any $i \in \{1, \ldots, n\}$, it holds that

$$|q\alpha_i - p_i - \eta_i| = \left|q\alpha_i - p_i - \left(\frac{c_i}{d_i} - p_i\right)\right| = \left|q\alpha_i - \frac{c_i}{d_i}\right| \overset{(6)}{\leq} \frac{1}{d_i^2} \overset{(20)}{<} \varepsilon\,. \tag{23}$$

## 4 Security

### 4.1 Binding Property

In this section, we prove that $q$ is uniquely determined by the commitment $(\boldsymbol{\alpha}, \boldsymbol{\eta})$. Thus, the scheme is perfectly binding.

**Theorem 5 (Perfectly binding).** *The commitment scheme is perfectly binding.*

**Algorithm 1** The commitment algorithm $\mathsf{Commit}_{\mathcal{P}}$

**Input:** $\mathcal{P} = (N, \varepsilon, n, \mu)$ with approximation quality $\varepsilon$, dimension $n$, and upper bound $\mu$; a message $m \in \{0,1\}$

**Output:** A commitment on $m$

1: **//Map the message**

2: Extend $m \in \{0,1\}$ to a $s$-bit value $q$, that is $[q]_2 = (r_{s-1}, \ldots, r_1, m)$ with $r_i \xleftarrow{\$} \{0,1\}$. $[q]_2$ denotes the bit representation of $q$. This implies $0 \leq q < 2^s =: N$.

3: **//Generate rational numbers** $\alpha_i := \frac{a_i}{b_i}$

4: **for** $i = 1, \ldots, n$ **do**

5: Choose co-prime integers $a_i$ and $b_i$ where $b_i$ is odd, co-prime to $q$, and less than or equal to $\mu$.

6: Set $\alpha_i := \frac{a_i}{b_i}$.

7: **end for**

8: **//Generate approximations** $\frac{c_i}{d_i}$ **of** $q \cdot \frac{a_i}{b_i}$

9: Use continued fractions to find an approximation of $\frac{c_i}{d_i}$ of $q \cdot \frac{a_i}{b_i}$ such that

$$\frac{1}{\sqrt{\varepsilon}} < d_i < b_i \,. \tag{20}$$

10: If (20) is not satisfiable, restart at line 3.

11: **//Check additional condition**

12: Beside the conditions given above, we require the existence of an index $i^* \in \{1, \ldots, n\}$ with

$$N < b_{i^*} \quad \text{and} \quad \sqrt{2b_{i^*}} < d_{i^*} \,. \tag{21}$$

13: If (21) is not satisfiable, restart at line 3.

14: **//Generate $\boldsymbol{p}$ and $\boldsymbol{\eta}$**

15: **for** $i = 1, \ldots, n$ **do**

16: Choose $p_i \in \mathbb{Z}$ arbitrary

17: Set $\eta_i := \frac{c_i}{d_i} - p_i$

18: **end for**

19: **return** A (public) commitment $(\boldsymbol{\alpha}, \boldsymbol{\eta})$ to $m$ and (secret) opening information $(q, \boldsymbol{p})$

*Proof.* Assume two solutions $(q, \boldsymbol{p})$ and $(q', \boldsymbol{p}')$. (21) ensures the existence of an index $i^*$ such that $N < b_{i^*}$ and $\sqrt{2b_{i^*}} < d_{i^*}$. We omit the index $i^*$ in the following. By definition it holds that $\eta = \frac{c}{d} - p$ and $\eta = \frac{c'}{d'} - p'$ for some appropriate integers $c, d, c', d'$ and in particular $\frac{c}{d} - \frac{c'}{d'} \in \mathbb{Z}$. Therefore, there exists an integer $z \in \mathbb{Z}$ such that

$$\frac{c}{d} - \frac{c'}{d'} = z \iff cd' - c'd = zdd' \,. \tag{24}$$

It follows that $cd' - c'd \equiv 0 \pmod{d}$, $cd' \equiv 0 \pmod{d}$, and $d' \equiv 0 \pmod{d}$. The latter holds as $c$ and $d$ are co-prime (see Sec. 2.1). Analogously, one shows that $d \equiv 0 \pmod{d'}$. As both $d$ and $d'$ are positive, we get $d = d'$. Now recall that the fractions $\frac{c}{d}$ and $\frac{c'}{d}$ are both approximations of $q \cdot \frac{a}{b}$ and $q' \cdot \frac{a}{b}$, respectively, stemming from continued fractions. With (6) we have

$$\left| q \cdot \frac{a}{b} - \frac{c}{d} \right| < \frac{1}{d^2} \quad \text{and} \quad \left| q' \cdot \frac{a}{b} - \frac{c'}{d} \right| < \frac{1}{d^2} \tag{25}$$

and in particular

$$\left| \left(q - q'\right) \frac{a}{b} - z \right| < \frac{2}{d^2} \iff \left| \left(q - q'\right) a - z \cdot b \right| < \frac{2b}{d^2} \,. \tag{26}$$

Recall that $\sqrt{2b} < d$ by Eq. (21). Thus, the right hand side of (26) is strictly less than 1 while the left hand side is an integer value. This immediately implies that $(q - q')a - z \cdot b = 0$. As $a$ and $b$ are co-prime, it follows that

$$q - q' \equiv 0 \pmod{b} \,. \tag{27}$$

With $0 \leq q, q' < N$, we have $-N < q - q' < N$. By Eq. (21), it holds that $b > N$. Thus, (27) actually implies $q - q' = 0 \Leftrightarrow q = q'$. □

### 4.2 Hiding Property

In this section, we prove that the commitment scheme is computationally hiding. Recall that this means that no efficient algorithm exists that can decide for a given commitment $(\boldsymbol{\alpha}, \boldsymbol{\eta})$ if it commits to $m = 0$ or to $m = 1$.

**Theorem 6 (Hiding).** *Let* Gen *denote the algorithm that generates DISAP instances as explained in Alg. 1 and let* $\mathsf{Gen}^*$ *denote the algorithm that first invokes* Gen *and then replaces* $\boldsymbol{\alpha}$ *by* $2\boldsymbol{\alpha}$*. If the DISAP assumption (Def. 3) holds with respect to* $\mathsf{Gen}^*$*, the commitment scheme is computationally hiding.*

*Proof.* Recall that the DISAP assumption tells that it is hard to decide whether a given instance has a solution or not. Furthermore, by definition the committed message equals to the least significant bit of $q$, for short: $\mathrm{LSB}(q)$. Thus, breaking the hiding property is equivalent to deciding the LSB of $q$. Let $I^* := (2\boldsymbol{\alpha}, \boldsymbol{\eta}, N, n, \varepsilon)$ where $I := (\boldsymbol{\alpha}, \boldsymbol{\eta}, N, n, \varepsilon)$ is the instance generated by Gen. Observe as only the values $\boldsymbol{\alpha}$ are changed, instance $I^*$ fulfills all conditions derived in Sec. 2.2 if this is the case for $I$. We show now that the LSB of $q$ is equal to 0 if and only if $I^*$ has a solution.

Assume that LSB of $q$ is zero. That is we can write $q = 2q^*$ and one sees easily that it holds for all $i = 1, \ldots, n$:

$$|q \cdot \alpha_i - p_i - \eta_i| < \varepsilon \Leftrightarrow |(2q^*) \cdot \alpha_i - p_i - \eta_i| < \varepsilon \Leftrightarrow |q^* \cdot (2\alpha_i) - p_i - \eta_i| < \varepsilon \tag{28}$$

Thus, if $q$ is a solution to $I$ with LSB$(q)$=0, then there exists a solution to $I^*$.

Contrariwise, assume that $I^*$ has a solution $q^*$. Then, with (28) it follows that $q = 2q^*$ is a solution to $I$. Moreover, as we have shown in Theorem 5, $q$ is the only unique solution. Thus, the existence of a solution $q^*$ implies that the LSB of the solution of $I$ is equal to zero. □

## 5 A Concrete Instantiation and Implementation

In this section we want to fix some values for the thresholds $\varepsilon^*$ and $n^*$. Due to our discussion of the algorithmic landscape in Sec. 2.2 and because of $q^{-(1+1/2)} \leq q^{-(1+1/n)}$ for all $n \geq 2$, we know that there exists no algorithm with a runtime polynomial in $n$ that given $(\boldsymbol{\beta}, N, n)$, finds integers $\tilde{q}$ and $\tilde{\boldsymbol{p}}$ such that

$$\left|\beta_i - \frac{\tilde{p}_i}{\tilde{q}}\right| < \frac{1}{\tilde{q}^{1+\delta-\delta'}} \tag{29}$$

with $\delta - \delta' = 1/2$. We set $\varepsilon^* := N^{-\delta} = 2^{-\delta s}$ and mention the upper bound on $\mu_i$ of $\mu^* := N^{\delta'} = 2^{\delta' s}$. In [14] it is stated that the used algorithm of Lenstra Jr. [16] has a runtime that grows exponentially in the dimension. This motivates us to set $n^* := \log(s)$. Observe that the effort of the commitment scheme grows linearly with $n$. Thus, increasing $n$ in the case of need induces only a linear overhead.

Looking back to Alg. 1, we set $\varepsilon := \varepsilon^*$ and $n := n^*$ in the following as concrete parameters. Next, we compute the size of a commitment and thereby get a hint how to choose $\delta'$. Due to the fact that the sizes of $a_i$ and $p_i$ do not effect the proofs of binding and hiding in Sec. 4 we are free in the choice of their bounds. Thus, we choose $a_i$ and $p_i$ equally distributed from the same interval as $q$, namely $[0, 2^s)$. Only for the $b_i$ we have to pay attention that $b_i \leq \mu$ holds.

The commitment consists of the quantities $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$. The $\alpha_i := a_i/b_i$ require $s + \delta' s$ bits because $a_i \in [0, 2^s)$ and $b_i \in [0, \mu) = \left[0, 2^{\delta' s}\right)$. Moreover, the denominators $d_i$ of the second part $\boldsymbol{\eta}$ of the commitment require $\delta' s$ bits due to $0 < d_i < b_i < 2^{\delta' s}$ (see condition (20) in Alg. 1). Finally we consider the expanded numerators $c_i - p_i d_i \in [-p_i d_i, c_i]$ and note that we need $s + \delta' s$ bits for the negative range because $p_i d_i < 2^s b_i < 2^{s+\delta' s}$ and $2s$ bits for the positive range ($c_i < q a_i < 2^{2s}$). Subsuming $\eta_i$ requires $3s + 2\delta' s$ bits leading to a complete commitment size of

$$|(\boldsymbol{\alpha}, \boldsymbol{\eta})|_2 = n\left(s + \delta' s\right) + n\left(3s + 2\delta' s\right) = ns\left(4 + 3\delta'\right) .$$

Because of $\delta' > 1$ we have the lower bound of $7ns$ bits for the commitment size. We see that we minimize the commitment size by minimizing $\delta'$ with respect to

$\delta' > 1$. By setting $\delta' := 1 + \delta''$ with $\delta'' > 0$ we get $|(\boldsymbol{\alpha}, \boldsymbol{\eta})|_2 = 7ns + 3ns\delta''$, leading to $(3ns)^{-1}$ as a minimal choice for $\delta''$.

We implemented the scheme[7] and made about $10^6$ test runs on a AMD Athlon X2 Dual-Core QL-62 with 2 GHz per core with $n = 7, s = 128$ and minimal $\delta'' = (3 \cdot 128 \cdot 7)^{-1}$. This gives a commitment size of 6273 bit. The algorithm restarts the computation of the commitment on an average of 3.0579 times in order to satisfy (21) (cf. line 13 in Alg. 1). The maximal number of restarts to compute a single commitment was 23. Condition (20) was always fulfilled. Furthermore, all operations are really cheap in software – leading to running times in the milliseconds not measurable in seconds.

## 6 Future Work and Conclusions

In this work, we focused on one particular problem from analytic number theory, namely the Decisional Inhomogeneous Simultaneous Approximation Problem (DISAP). The problem is NP-complete and one can efficiently generate presumably hard instances. Observe that the difficulty can be easily increased, e. g., by raising the dimension $n$. As a proof of concept, we constructed a bit commitment scheme on DISAP. However, other schemes would have been imaginable.

For example, observe that if $q$ is known, the enumerators $\boldsymbol{p}$ can be directly computed. Thus, one could modify the commitment scheme to get a stream cipher where $q$ would be the secret key and $p_i$ the individual plaintexts. Whenever the sender wants to encrypt a plaintext $p_i$, he computes the other values $\alpha_i$, etc. as described and uses the tuple $(\alpha_i, \eta_i)$ as ciphertext for the current plaintext block. Observe that the values $\alpha_i$ and $c_i/d_i$ can be precomputed for accelerating the scheme. Although we did not check it in detail, we are optimistic that a proof of security should be possible that is similar to the proof given in this paper, at least for the known-ciphertext scenario. The development of other schemes might be interesting as well, e. g., authentication schemes giving a proof of knowledge on $q$.

Despite DISAP, other problems and results from analytic number theory might be worth to be investigated as well. For example, one can easily transform a rational number from its binary representation to continued fractions and vice versa. But only little is known on the relations between changes in one representation and the corresponding changes in the other representation. This "fragility" might be used to construct a collision-resistant compression function. Furthermore, several results exist on the periodicity of certain representations. The construction of bitstream generators based on these might be an interesting question.

[7] We used the GNU MP (`http://gmplib.org/`) and MPFR [7] library for arbitrary large integers and arbitrary precise floating point arithmetic.

Concluding, we think that the established discipline of analytic number theory contains many interesting open problems and results that only wait to be (re-)discovered for cryptographic applications. We hope to encourage further research into this direction.

## References


1. P. R. Baldwin. A convergence exponent for multidimensional continued-fraction algorithms. *Journal of Statistical Physics*, 66(5/6):1507–1526, 1992.
2. L. Bernstein. *The Jacobi-Perron algorithm, it's theory and application*, volume 207 of *Lecture Notes in Mathematics*. Springer Verlag, Berlin, Heidelberg, New York, 1971.
3. A. J. Brentjes. Multi-dimensional continued fraction algorithms. *Mathematical Centre Tracts*, 145, 1981.
4. C. Elsner and M. Schmidt. KronCrypt - a new symmetric cryptosystem based on Kronecker's approximation theorem. Cryptology ePrint Archive, Report 2009/416, 2009. `http://eprint.iacr.org/`.
5. M. Fellows and N. Koblitz. Combinatorial cryptosystems galore! *Contemporary Mathematics*, 168:51–61, 1993.
6. C. Fontaine and F. Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP J. Inf. Secur.*, 2007(1):1–15, 2007.
7. Laurent Fousse, Guillaume Hanrot, Vincent Lefèvre, Patrick Pélissier, and Paul Zimmermann. MPFR: A multiple-precision binary floating-point library with correct rounding. *ACM Trans. Math. Softw.*, 33(2):13, 2007.
8. R. Gärtner. Zur Geometrie des Jacobi-Perron Algorithmus. *Arch. Math.*, 39:134–146, 1982.
9. G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers*. Clarendon Press, Oxford, 3rd ed. edition, 1954.
10. H. Isselhorst. The use of fractions in public-key cryptosystems. In *EUROCRYPT*, pages 47–55, 1989.
11. C. G. J. Jacobi. Allgemeine Theorie der kettenbruchähnlichen Algorithmen, in welchen jede Zahl aus drei vorhergehenden gebildet wird. *Journal für die reine und angewandte Mathematik (Crelle's Journal)*, 69:29–64, 1868.
12. B. Just. Generalizing the continued fraction algorithm to arbitrary dimensions, 1992.
13. J. C. Lagarias. Knapsack public key cryptosystems and diophantine approximation. In *CRYPTO*, pages 3–23, 1983.
14. J. C. Lagarias. The computational complexity of simultaneous diophantine approximation problems. *SIAM J. Comput.*, 14(1):196–209, 1985.
15. A. K. Lenstra, H. W. Lenstra Jr., and L. Lovasz. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.
16. H. W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, Nov. 1983.
17. F. Levy-dit-Vehel, M. Marinari, L. Perret, and C. Traverso. *Gröbner Bases, Coding Theory, and Cryptography*, chapter A Survey on Polly Cracker systems. RISC Book Series. Springer, Heidelberg, 2009.
18. Hua Loo Keng. *Introduction to number theory*. Springer Verlag, Berlin, Heidelberg, New York, fifth edition, 1982.
19. P. Q. Nguyen and B. Valle, editors. *The LLL Algorithm. Survey and Applications*. Information Security and Cryptography. Springer, 2010.
20. O. Perron. Grundlagen für eine Theorie des Jacobischen Kettenbruchalgorithmus. *Math. Ann.*, 64:1–76, 1907.

21. G. J. Rieger. *Zahlentheorie*. Vandenhoeck & Ruprecht, Göttingen, 1976.
22. C. Rössner and J.-P. Seifert. Approximating good simultaneous diophantine approximations is almost NP-hard. In Wojciech Penczek and Andrzej Szalas, editors, *MFCS*, volume 1113 of *Lecture Notes in Computer Science*, pages 494–505. Springer, 1996.
23. W. Schmidt. *Diophantine approximations*. Springer-Verlag, Berlin, 1980.
24. C.-P. Schnorr. Factoring integers and computing discrete logarithms via diophantine approximations. In *EUROCRYPT*, pages 281–293, 1991.
25. F. Schweiger. *The metrical theory of Jacobi-Perron algorithm*, volume 334 of *Lecture Notes in Mathematics*. Springer Verlag, Berlin, Heidelberg, New York, 1973.
26. F. Schweiger. *Multidimensional continued fractions*. Oxford University Press, 2000.
27. F. Schweiger. Was leisten mehrdimensionale Kettenbrüche? *Mathematische Semesterberichte*, 53:231–244, 2006.
28. J.-P. Seifert. Using fewer qubits in Shor's factorization algorithm via simultaneous diophantine approximation. In *CT-RSA 2001: Proceedings of the 2001 Conference on Topics in Cryptology*, pages 319–327, London, UK, 2001. Springer-Verlag.
29. A. Shamir. A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. In *SFCS '82: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 145–152, Washington, DC, USA, 1982. IEEE Computer Society.
30. J. Stern and P. Toffin. Cryptanalysis of a public-key cryptosystem based on approximations by rational numbers. In *EUROCRYPT*, pages 313–317, 1990.
31. C. Szekeres. Multidimensional continued fractions. *Ann. Univ. Sci. Budap. Eötös, Sect. Math.*, 13:113–140, 1980.
32. L. Van Ly. Polly two : A new algebraic polynomial-based public-key scheme. *Appl. Algebra Eng. Commun. Comput.*, 17(3–4):267–283, 2006.
33. M. J. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*, 36:553–558, 1990.