Measure Algebras and Applications

Increasing the efficiency of the key generation algorithm for NTRU with the help of the norm field

Reza Alimoradi[1]✉, **Mohammad Hossein Noorallahzadeh**[2], **Ahmad Gholami**[3]

1. Corresponding Author, University of Qom, Qom, Iran. Email: r.alimoradi@qom.ac.ir
2. University of Qom, Qom, Iran. Email: mh.noorallahzadeh@stu.qom.ac.ir
3. University of Qom, Qom, Iran. Email: a.gholami@qom.ac.ir

Article Info

Article type:
Research Article

Article history:
Received: 09 February 2024
Received in revised form:
16 April 2024
Accepted: 19 June 2024
Published Online:
20 August 2024

Keywords:
Post-quantum cryptographic schemes,
Lattice-based cryptographic schemes,
NTRU-based cryptographic schemes,
Algorithms based on soft field

2020 Mathematics Subject Classification: 20F05, 05C05

ABSTRACT

Conceptually, a signature scheme consists of three steps: private key generation, signature, and authentication. Private key generation in NTRU-based signature schemes on a typical laptop (Intel Core i7-6567U 3.30 GHz) takes a long time (more than one second), while signature and verification take much less time (for example, a thousandths of a second). The current paper deals with providing solutions to reduce the time of private key generation. In this paper, the previous methods are studied and then a new method based on the norm field is introduced and it is shown that the execution time is significantly reduced by using it.

Cite this article: Alimoradi, R., Noorallahzadeh, M.H., & Gholami, A. (2024). Increasing the efficiency of the key generation algorithm for NTRU with the help of the norm field. *Measure Algebras and Applications*, 2(1), 41–70. http://doi.org/10.22091/MAA.2024.10396.1014

Publisher: University of Qom
DOI: 10.22091/MAA.2024.10396.1014

Extended Abstract

Introduction

NTRU lattices have emerged as a specialized subset of general lattices, offering distinct advantages that have garnered significant attention in the realm of cryptography. Their efficient implementation of common lattice algorithms has positioned them as a favorable choice for employment in asymmetric cryptographic schemes, particularly those involving public key encryption and digital signatures. Within NTRU-based schemes, fundamental components such as keys and ciphertexts are represented as polynomials, reflecting the inherent algebraic structure of NTRU lattices.

Notably, the private key utilized in certain NTRU-based schemes is characterized by a polynomial of low degree with exceedingly small coefficients, while the public key is represented by a polynomial with large coefficients. This distinction effectively establishes short and long bases within the lattice structure, contributing to the security and efficiency of NTRU-based cryptographic systems. The unique properties of NTRU lattices have not only facilitated the development of robust cryptographic solutions but have also sparked further exploration and research in the field, holding promise for continued advancements in secure communication and data protection. Several lattice-based encryption schemes usually require solving the NTRU equation to generate keys:

$$fG-gF=q \bmod x^n+1$$

where f and g are constants, and the objective is to calculate F and G for the equation. It should be noted that the polynomials are in

$$\mathbb{Z}[x]/\left(x^n+1\right).$$

Conceptually, a signature scheme consists of three stages: key generation, signing, and verification. In the context of NTRU-based signature schemes, the process of private key generation typically consumes a significant amount of time, especially when executed on standard computing hardware such as a regular laptop (Intel Core i7-6567U 3.30 GHz), where the generation process may exceed one second. Conversely, the signing and verification stages exhibit notably lower time requirements, often on the order of milliseconds.

The current paper delves into the challenge of mitigating the time-intensive nature of private key generation in NTRU-based signature schemes. It thoroughly examines existing methods and their associated limitations, paving the way for the introduction of a novel approach rooted in the realm of number fields. This innovative method showcases a remarkable reduction in the execution time required for private key generation, presenting a compelling avenue for enhancing the overall efficiency and practicality of NTRU-based signature schemes.

The introduction of asymmetric encryption systems by Diffie and Hellman in 1976 marked a significant milestone in the evolution of cryptography. At the core of asymmetric systems lies the concept of employing a set of information along with a one-way function for encryption, which in isolation does not provide sufficient data for decryption. For the decryption process, an additional finite set of information, known as the "private key," is indispensable, while the set of information required for encryption is termed the "public key."

The prevalent one-way function in asymmetric encryption, rooted in discrete logarithm and exponentiation, serves as the cornerstone for well-known asymmetric encryption systems such as ElGamal,

ECC, and RSA. These systems hinge on number theory problems that, with the emergence of quantum computing, are susceptible to resolution. In response to this vulnerability, extensive research has been directed toward exploring the complexity of lattice problems, aiming to identify alternative approaches for asymmetric encryption that do not rely on lattice-based foundations. Notably, the NTRU encryption system, a highly efficient lattice-based system, derives its resilience from the formidable challenge of solving the Shortest Vector Problem (SVP) within lattices, presenting a compelling avenue for robust encryption in the face of advancing cryptographic landscapes. This paper also delves into the mathematical prerequisites needed for the study. Specifically, it includes a review of lattice concepts, gathering essential concepts from NTRU lattices, studying number fields and related concepts, reviewing Karatsuba multiplication (used in the paper), and finally examining ring structures.

A real-valued V module over R, which functions as a module over a set closed under addition and scalar multiplication, transforms into a lattice when it is bounded by a finite set of real numbers. The defining characteristic of a lattice lies in the presence of a bounded set of real numbers that can be added to the set, establishing its fundamental structure.

In our pursuit of optimizing computations on polynomial rings, particularly in the context of solving the NTRU equation, we have strategically employed number fields to enhance efficiency and performance. This strategic utilization of number fields carries significant practical implications, particularly for the post-quantum Falcon signature algorithm. Notably, our optimizations enable the complete utilization of the Falcon signature algorithm on small microcontrollers or even smart cards, with the algorithm requiring a mere 32 kilobytes of RAM to operate effectively. This level of resource efficiency extends to the implementation of long-term secure NTRU lattices (degree $n = 1024$), showcasing that all signature operations, including signature generation, verification, and key pair generation, can be seamlessly executed on such resource-constrained hardware environments. This breakthrough paves the way for the widespread deployment of robust cryptographic solutions in diverse computing environments, from embedded systems to IoT devices, without compromising on security or performance.
We also list several open questions below:

Non-cyclotomic polynomials: In our description, we covered cyclotomic polynomials as a covering module. This approach can be extended to other modules. In fact, for any module

$$\varphi = \varphi'(x^d)$$

for some $d > 1$, the use of a "number field" can divide the degree by d for the purposes of calculating residuals and solving the NTRU equation.

Even if φ is not irreducible in $Q[x]$, i.e., if $Q[x]/(\varphi)$ is not actually a field, the general case remains a problem for further investigation. However, the use of reducible modules in NTRU lattices is generally not recommended.

While our achievements in memory management are indeed significant, the challenge of effectively handling large integers remains a prominent concern that warrants continued exploration. From the perspective of implementation complexity, the prospect of eliminating large integers, for instance by conducting all operations in the Residue Number System (RNS), without adversely impacting the execution time and memory requirements of our algorithms, presents an intriguing area for further investigation and potential optimization. This pursuit holds the promise of streamlining computational processes and resource utilization, contributing to enhanced efficiency across a spectrum of cryptographic applications.

In addition to addressing the management of large integers, it is imperative to explore potential ap-

plications of the method proposed in this paper to enhance the efficiency of other encryption algorithms. Just as we have demonstrated a constructive application of a number field in this work, distinct from the approach in a previous study, there is merit in investigating a constructive application of lattice tracking, as opposed to a different reference. This comparative exploration can shed light on the adaptability and versatility of our proposed methodology within the broader landscape of encryption and security protocols.

Furthermore, leveraging the method outlined in this paper to enhance attacks on a specific field or even on field tracking holds the potential to yield valuable insights, opening up new possibilities for specialized analysis applications in the realm of cryptography and security. These potential directions for further exploration underscore the multifaceted implications of the research presented in this paper, offering promising avenues for continued advancements in cryptographic techniques and their practical applications. This comprehensive approach to exploring the broader implications of our work sets the stage for future breakthroughs in the field of cryptography and computational security, paving the way for innovative solutions and heightened resilience in the face of evolving security challenges.

Conclusion

We presented the use of the norm field to optimize some computations on polynomial loops, especially the results and solutions of the NTRU equation. The second practical result is that Falcon's post-quantum signature algorithm is fully usable on small microcontrollers or even smart cards since 32 KB of RAM are required to run our algorithm even for a long-term secure NTRU network (degree $n = 1024$). Enough.: All operations related to signatures (signature generation, verification, and key pair generation) can be placed on such limited hardware.

جبرهای اندازه و کاربردها
سال ۱۴۰۳، دوره ۲، شماره ۱، ص ۷۰-۴۱

افزایش کارآمدی الگوریتم تولید کلید مشبکه‌های NTRU به کمک نرم میدان

رضا علیمرادی [۱]✉ ، محمدحسین نوراله زاده [۲] ، احمد غلامی [۳]

۱. نویسندهٔ مسئول، دانشگاه قم، قم، ایران. رایانامه: r. alimoradi@qom. ac. ir
۲. دانشگاه قم، قم، ایران. رایانامه: mh. noorallahzadeh@stu. qom. ac. ir
۳. دانشگاه قم، قم، ایران. رایانامه: a. gholami@qom. ac. ir

اطلاعات مقاله

نوع مقاله:
مقالهٔ پژوهشی

تاریخ دریافت: ۱۴۰۲/۱۱/۲۰
تاریخ بازنگری: ۱۴۰۳/۱/۲۸
تاریخ پذیرش: ۱۴۰۳/۳/۳۰
تاریخ انتشار: ۱۴۰۳/۵/۳۰

کلمات کلیدی:
طرح‌های رمزنگاری پساکوانتومی،
طرح‌های رمزنگاری مشبکهٔ مبنا،
طرح‌های رمزنگاری مبتنی بر NTRU،
الگوریتم‌های مبتنی بر نرم میدان

رده‌بندی ریاضی:
20F05, 05C05

چکیده

در طراحی بسیاری از طرح‌های نامتقارن مانند کلید عمومی و امضای دیجیتال از مشبکه‌های NTRU استفاده می‌کنند. به‌صورت مفهومی یک طرح امضا از سه مرحله تشکیل می‌شود: تولید کلید خصوصی، امضا و تصدیق. برای تولید کلید خصوصی در طرح‌های امضای مبتنی بر NTRU در یک لپ‌تاپ معمولی (Intel Core i7-6567U 3.30 GHz) زمان زیادی صرف می‌شود (بیش از یک ثانیه) درحالی‌که امضا و تصدیق به‌مراتب زمان کمتری نیاز دارند (برای مثال یک‌هزارم ثانیه). مقالهٔ فعلی به ارائهٔ راهکارهایی برای کاهش زمان مرحلهٔ تولید کلید خصوصی می‌پردازد. در این مقاله، روش‌های قبلی مورد مطالعه قرار می‌گیرند و سپس یک روش جدید مبتنی بر نرم میدان معرفی می‌گردد و نشان داده می‌شود که با استفاده از آن، زمان اجرا به‌طور قابل ملاحظه‌ای کاهش پیدا می‌کند.

استناد: علیمرادی، رضا.، نوراله زاده، محمدحسین.، غلامی، احمد. (۱۴۰۳). افزایش کارآمدی الگوریتم تولید کلید مشبکه‌های NTRU به کمک نرم میدان. جبرهای اندازه و کاربردها، ۲(۱)، ۷۰-۴۱.

http://doi.org/10.22091/MAA.2024.10396.1014

ناشر: دانشگاه قم.

۱ مقدمه

مشبکه‌های NTRU حالت خاصی از مشبکه‌های عمومی هستند. بسیاری از الگوریتم‌های رایج در نظریهٔ مشبکه‌ها زمانی که برای مشبکه‌های NTRU استفاده می‌شوند، به‌صورت بسیار کارآمدتری قابل پیاده‌سازی هستند. در عمل برای طراحی بسیاری از طرح‌های نامتقارن مانند کلید عمومی و امضای دیجیتال از این مشبکه‌ها استفاده می‌شود. در طرح‌های مبتنی بر NTRU عناصر اصلی مانند کلید، متن رمزی و ... از نوع چندجمله‌ای هستند. به‌طور خاص، کلید خصوصی مورد استفاده در برخی از طرح‌های مبتنی بر NTRU یک یا دو چندجمله‌ای با ضرایب بسیار کوچک، و کلید عمومی یک یا دو چندجمله‌ای با ضرایب بزرگ است که این‌ها را می‌توان به‌عنوان پایه‌های کوتاه و بلند در یک مشبکه در نظر گرفت. تعدادی از طرح‌های رمزنگاری مبتنی بر مشبکه، معمولاً برای تولید کلید، مستلزم حل معادلهٔ NTRU هستند:

$$fG-gF=q \bmod x^n+1.$$

در اینجا f و g ثابت بوده و هدف، محاسبهٔ F و G برای معادله است. لازم به ذکر است که چندجمله‌ای‌ها در $\mathbb{Z}[x]/\left(x^n+1\right)$ قرار دارند. در این مقاله به بررسی و مطالعهٔ پیش‌نیازهای ریاضی مورد نیاز نیز خواهیم پرداخت. به‌طور خاص:

- مروری بر مفاهیم مشبکه خواهیم داشت،
- مفاهیم مورد نیاز از مشبکه‌های NTRU را گردآوری می‌کنیم،
- نرم میدان و مفاهیم مرتبط با آن را مورد مطالعه قرار می‌دهیم،
- مروری بر ضرب کاراتسوبا خواهیم داشت (در مقاله مورد استفاده قرار گرفته است)،
- و در نهایت برج حلقه‌ها را بررسی می‌کنیم.

سامانه‌های رمزنگاری نامتقارن در سال ۱۹۷۶ توسط "دیفی" و "هلمن" با ارائه در مقاله [۹] معرفی شدند. یک سامانهٔ نامتقارن بر این مفهوم بنانهاده شده است که مجموعه‌ای متناهی از اطلاعات به انضمام یک تابع یک‌طرفه برای رمزنگاری اطلاعات مورد استفاده قرار می‌گیرد ولی این اطلاعات برای رمزگشایی کافی نیست و برای رمزگشایی داده‌های رمزشده به مجموعهٔ متناهی دیگری از اطلاعات نیاز است. به مجموعهٔ اطلاعات لازم برای رمزگذاری، "کلید عمومی" و به مجموعهٔ اطلاعات مورد نیاز برای رمزگشایی "کلید خصوصی" گفته می‌شود. تابع یک‌طرفهٔ پرکاربرد در رمزگذاری نامتقارن تجزیهٔ اعداد و لگاریتم گسسته است که اساس سامانه‌های رمزنگاری نامتقارن مانند ElGamal، ECC، RSA را تشکیل می‌دهد. این سامانه‌ها بر پایهٔ مسائل نظریهٔ اعداد استوارند که با بهبود قدرت و توسعهٔ محاسبات کوانتوم، قابل‌حل هستند. تحقیقات روی دشواری مسائل مشبکه، دانشمندان را به دستیابی نامزد دیگری برای رمزنگاری نامتقارن، امیدوار می‌کند؛ که چیزی جز رمزنگاری مبتنی بر مشبکه نیست. ازجمله مسائل دشوار در مشبکه که در این مقاله معرفی خواهند شد؛ می‌توان به مسائل کوتاه‌ترین بردار[۱] و نزدیک‌ترین بردار[۲] اشاره کرد. یکی از سامانه‌های فوق‌العاده کارآمد مبتنی بر مشبکه، سامانهٔ رمزنگاری NTRU است که به آن خواهیم پرداخت. این سامانه، امنیت خود را از سختی مسئلهٔ SVP اخذ می‌کند.

در ادامه مقدمات ریاضی لازم بیان می‌گردد: یک فضای برداری V روی اعداد حقیقی R، مجموعه‌ای از بردارها است که نسبت به اعمال جمع و ضرب بسته است. یک مشبکه شبیه به فضای برداری است که در آن به ضرب بردارها با اعداد صحیح محدود شده‌ایم.

۱.۱ تعاریف بنیادی مشبکه و ویژگی‌های آن

تعریف ۱.۱. اگر $v_1, v_2, \ldots, v_n$ مجموعه‌ای از بردارهای مستقل خطی باشد، مشبکهٔ L تولیدشده توسط آن، مجموعهٔ تمام ترکیبات خطی از $v_1, v_2, \ldots, v_n$ است که ضرایب اعداد صحیح هستند:

$$L = L\left(v_1, v_2, \ldots, v_n\right) = \left\{a_1 v_1 + a_2 v_2 + \cdots + a_n v_n \ : a_i \in Z\right\}.$$

مجموعهٔ $v_1, v_2, \ldots, v_n$، پایه و n را بعد مشبکهٔ L می‌نامند.

گزاره ۲.۱. اگر $V = \{v_1, v_2, \ldots, v_n\}$ و $W = \{w_1, w_2, \ldots, w_n\}$ پایه‌هایی برای مشبکهٔ L باشند، آنگاه $W = AV$، که A ماتریسی با درایه‌های صحیح است و $|\det(A)| = 1$.

[1] Shortest Vector Problem (SVP)
[2] Closest Vector Problem (CVP)

اثبات. ازآنجاکه $w_i \in L$، داریم

$$w_۱ = a_{۱۱}v_۱ + a_{۱۲}v_۲ + \dots a_{۱n}v_n$$
$$w_۲ = a_{۲۱}v_۱ + a_{۲۲}v_۲ + \dots a_{۲n}v_n$$
$$\vdots$$
$$w_n = a_{n۱}v_۱ + a_{n۲}v_۲ + \dots a_{nn}v_n \ .$$

این درحالی است که $a_{ij} \in Z$. به‌عبارت‌دیگر

$$\begin{pmatrix} w_۱ \\ w_۲ \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} a_{۱۱} & a_{۱۲} & \dots & a_{۱n} \\ a_{۲۱} & a_{۲۲} & \dots & a_{۲n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n۱} & a_{n۲} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} v_۱ \\ v_۲ \\ \vdots \\ v_n \end{pmatrix} .$$

ماتریس $\begin{pmatrix} a_{۱۱} & a_{۱۲} & \dots & a_{۱n} \\ a_{۲۱} & a_{۲۲} & \dots & a_{۲n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n۱} & a_{n۲} & \dots & a_{nn} \end{pmatrix}$ را A می‌نامیم. درنتیجه $W = AV$ که درایه‌های A صحیح هستند. ازطرفی می‌توان نتیجه گرفت که درایه‌های $A^{-۱}$ نیز صحیح هستند؛ زیرا می‌دانیم ماتریس B موجود است که $V = BW$ و چون $W = AV$، داریم $V = A^{-۱}W$. در نهایت وارون‌پذیری W، تساوی $B = A^{-۱}$ را نتیجه می‌دهد. به‌عبارت‌دیگر $A^{-۱}$ ماتریسی صحیح است. ازطرفی می‌دانیم $\det\left(A^{-۱}\right) \det\left(A\right) = ۱$، پس می‌توان گفت $|\det\left(A\right)| = ۱$. درنتیجه حکم به دست می‌آید. □

تعریف ۳.۱. ماتریس‌هایی با ویژگی فوق، درایه‌هایی صحیح همراه با قدر مطلق دترمینان برابر ۱، را ماتریس‌های یونی مادولار[1] می‌نامند.

تعریف ۴.۱. فرض کنید L مشبکهٔ n بعدی با پایهٔ $\{v_۱, v_۲, \dots, v_n\}$ باشد، دامنهٔ اصلی مشبکهٔ L منسوب به این پایه را، مجموعهٔ زیر در نظر می‌گیریم

$$F\left(v_۱, v_۲, \dots, v_n\right) = \{t_۱v_۱ + t_۲v_۲ + \dots t_nv_n \ : ۰ \leq t_i < ۱\} \ .$$

مثالی از مشبکهٔ ۲ بعدی و دامنهٔ اصلی آن در شکل ۱ آورده شده است.

گزاره ۵.۱. اگر $L \subseteq R^n$ مشبکه‌ای n بعدی و F دامنهٔ اصلی آن باشد، آنگاه هر بردار $w \in R^n$ نمایش منحصربه‌فردی به‌صورت $w = t + a$ دارد که $t \in F$ و $a \in L$.

اثبات. اگر $\{v_۱, v_۲, \dots, v_n\}$ را به‌عنوان پایه برای مشبکهٔ L در نظر بگیریم؛ می‌توان w را به‌صورت ترکیب خطی از آن‌ها نوشت

$$w = b_۱v_۱ + b_۲v_۲ + \dots + b_nv_n.$$

ازطرفی می‌توان نوشت $b_i = t_i + a_i$ که $۰ \leq t_i < ۱$ و $a_i \in Z$ (برای هر $۱ \leq i \leq n$). پس می‌توان t را برابر $t_۱ + t_۲ + \dots + t_n$ و a را برابر $a_۱ + a_۲ + \dots + a_n$ در نظر گرفت. برای اثبات منحصربه‌فردی نمایش $w = t + a$، فرض می‌کنیم $w = t + a = t' + a'$؛ که $t, t' \in F$ و $a, a' \in L$. ازآنجاکه $t \in F$ داریم $t = t_۱v_۱ + t_۲v_۲ + \dots t_nv_n$ که $۰ \leq t_i < ۱$. به همین ترتیب $t' = t'_۱v_۱ + t'_۲v_۲ + \dots + t'_nv_n$ که $۰ \leq t'_i < ۱$. ازآنجاکه $a \in L$ داریم $a = a_۱v_۱ + a_۲v_۲ + \dots + a_nv_n$ که $a_i \in Z$. به همین ترتیب $a' = a'_۱v_۱ + a'_۲v_۲ + \dots + a'_nv_n$ که $a'_i \in Z$. درنتیجه داریم

$$w = (t_۱ + a_۱)\, v_۱ + \dots + (t_n + a_n)\, v_n = \left(t'_۱ + a'_۱\right) v_۱ + \dots \left(t'_n + a'_n\right) v_n \ .$$

ازطرفی به دلیل استقلال $\{v_۱, v_۲, \dots, v_n\}$، داریم $t_i + a_i = t'_i + a'_i$ (برای هر $۱ \leq i \leq n$). درنتیجه

$$t_i - t'_i = a'_i - a_i \ .$$

پس هر دو طرف تساوی صفر خواهند شد که در نهایت؛ یکتایی نمایش w را نتیجه می‌دهد. □

[1] Uni Modular

تعریف ۶.۱. فرض کنید L مشبکهٔ n بعدی و F دامنهٔ اصلی آن باشد، حجم F را که با $Vol\,(F)$ نمایش می‌دهند، دترمینان L می‌نامیم.

گزاره ۷.۱ (نامساوی هادامارد). فرض کنید L مشبکه‌ای n بعدی باشد. برای هر پایهٔ $\{v_۱, v_۲, \ldots, v_n\}$ و دامنهٔ اصلی F، داریم

$$\det(L) = Vol\,(F) \leq ||v_۱||\,||v_۲|| \ldots ||v_n||$$

و اگر پایه‌ها متعامد باشند، نامساوی بالا تبدیل به تساوی خواهد شد.

گزاره ۸.۱. اگر $L \subseteq R^n$ مشبکه‌ای با بعد n، $\{v_۱, v_۲, \ldots, v_n\}$ پایهٔ آن و همچنین F دامنهٔ اصلی L منسوب به این پایه باشد، داریم

$$Vol\,(F) = |\det(V)|\,;$$

درحالی‌که درایه‌های V (ماتریس $n \times n$)، در سطر i ام معادل با درایه‌های v_i است.

اثبات. بنابر تعریف انتگرال، داریم

$$Vol\,(F) = \int_F dx_۱\, dx_۲ \ldots dx_n\,,$$

به‌طوری‌که $X = (x_۱, x_۲, \ldots, x_n) \in F$. ازآنجاکه $X = tV$؛ می‌توان نوشت $t \in C_n$

$$\begin{aligned}
&\int_F dx_۱\, dx_۲ \ldots dx_n \\
&= \int_{C_n V} dx_۱\, dx_۲ \ldots dx_n \\
&= \int_{C_n} |\det(V)|\, dt_۱\, dt_۲ \ldots dt_n \\
&= |\det(V)| \int_{C_n} dt_۱\, dt_۲ \ldots dt_n \\
&= |\det(V)|\,,
\end{aligned}$$

درنتیجه حکم به دست می‌آید. □

نتیجه ۹.۱. اگر $L \subseteq R^n$ مشبکه‌ای با بعد n باشد، آنگاه تمام دامنه‌های اصلی L (منسوب به هر پایهٔ دلخواه)، حجم یکسانی دارند. به‌عبارت‌دیگر؛ مقدار $\det(L)$ پایا است.

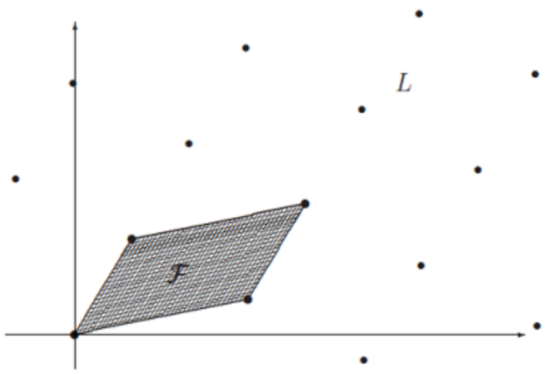
اثبات. اگر $\{v_۱, v_۲, \ldots, v_n\}$ و $\{w_۱, w_۲, \ldots, w_n\}$ دو پایه برای مشبکهٔ L باشند؛ باید نشان دهیم

$$|\det(V)| = |\det(W)|\,,$$

درحالی‌که V ماتریسی با سطرهای v_i و W ماتریسی با سطرهای w_i است. با توجه به گزاره ۲.۱، داریم $V = AW$ که $|\det(A)| = ۱$. خواهیم داشت

$$\begin{aligned}
|\det(V)| &= |\det(AW)| \\
&= |\det(A)\det(W)| \\
&= |\det(A)|\,|\det(W)| \\
&= |\det(W)|
\end{aligned}$$

درنتیجه حکم به دست می‌آید. □



شکل ۱: مشبکهٔ ۲-بعدی L و دامنهٔ اصلی آن.

۲.۱ مسائل دشوار در مشبکه

مسائل محاسباتی بنیادی در مشبکه‌ها، عبارت‌اند از پیدا کردن کوتاه‌ترین بردار ناصفر در مشبکه و یافتن نزدیک‌ترین بردار در مشبکه نسبت به برداری دلخواه.

مسئلهٔ کوتاه‌ترین بردار: مطلوب یافتن بردار ناصفر v در مشبکهٔ L است که $||v||$ کمترین مقدار ممکن باشد.

مسئلهٔ نزدیک‌ترین بردار: مطلوب یافتن بردار $v \in L$ است که نسبت به بردار داده‌شده $w \in R^n$ نزدیک‌ترین باشد. به‌عبارت‌دیگر؛ به‌ازای هر $a \in L$، داشته باشیم $||w - v|| \leq ||w - a||$.

هر دو مسئلهٔ SVP و CVP ازلحاظ محاسباتی بسیار دشوار هستند. به‌خصوص وقتی بعد مشبکه افزایش می‌یابد. از طرف دیگر؛ جواب‌های تخمینی برای این مسائل، کاربردهای بسیاری در مباحث ریاضیات کاربردی و محض دارند. لازم به ذکر است که مسئلهٔ SVP حالت خاصی از مسئلهٔ CVP است.

ازجمله مسائل دیگر در مشبکه، می‌توان به مسئلهٔ کوتاه‌ترین پایه[۱]، مسئلهٔ تقریب کوتاه‌ترین بردار[۲] و مسئلهٔ تقریب نزدیک‌ترین بردار[۳] اشاره کرد.

مسئلهٔ کوتاه‌ترین پایه: مطلوب یافتن کوتاه‌ترین پایهٔ $\{v_1, v_2, \ldots, v_n\}$ برای مشبکهٔ L است. نسخه‌های متفاوتی از SBP موجود هستند که هر یک وابسته به تعریف «اندازهٔ پایه» است.

مسئلهٔ تقریب کوتاه‌ترین بردار: فرض کنید مشبکهٔ L دارای بعد L باشد، مطلوب یافتن بردار $v \in L$ است که $||v|| < f(n)\,||v_{shortest}||$. روشن است که؛ بسته به انتخاب $f(n)$، جواب می‌تواند متفاوت باشد.

مسئلهٔ تقریب نزدیک‌ترین بردار: فرض کنید w بردار دلخواهی در R^n باشد. مطلوب یافتن بردار $u \in L$ است که به‌ازای هر $v \in L$، داشته باشیم $||w - u|| \leq \gamma\,||w - v||$.

۳.۱ تقریب‌هایی از اندازهٔ کوتاه‌ترین بردار در مشبکه

با توجه به اهمیت مسئلهٔ SVP؛ در این بخش به اندازهٔ کوتاه‌ترین بردار در مشبکه می‌پردازیم. چنان‌که با استفاده از قضایایی چون هرمیت[۴] و مینکوفسکی[۵]، کران بالایی برای جواب مسئلهٔ SVP به دست می‌آوریم. این کران وابسته به بعد و دترمینان مشبکه است.

قضیه ۱.۱۰ (هرمیت). هر مشبکهٔ L با بعد n، شامل بردار ناصفر $v \in L$ است که

$$||v|| \leq \sqrt{n}\det(L)^{\frac{1}{n}}\ .$$

تعریف ۱.۱۱. برای n داده‌شده، ثابت هرمیت γ_n، کوچکترین مقداری است که مشبکهٔ n بعدی L شامل بردار ناصفر v است که $||v||^2 \leq \gamma_n \det(L)^{\frac{2}{n}}$.

[1] Shortest Basis Problem(SBP)
[2] Approximate Shortest Vector Problem (appr. SVP)
[3] ApproximateClosest Vector Problem (appr. CVP or γ -CVP)
[4] Hermit
[5] Minkowski

بنابر قضیه ۱۰.۱، $\gamma_n \leq n$. مقدار دقیق γ_n برای $1 \leq n \leq 8$ و $n = 24$ به دست آمده است

$$\gamma_2^2 = \frac{4}{3}\ ,\ \gamma_3^3 = 2\ ,\ \gamma_4^4 = 4\ ,\ \gamma_5^5 = 8$$
$$\gamma_6^6 = \frac{64}{3},\ \gamma_7^7 = 64\ ,\ \gamma_8^8 = 256\ ,\ \gamma_{24} = 4.$$

روشن است که در سامانه‌های رمزی، مطلوب یافتن γ_nای با n بزرگ است. برای مقادیر بزرگ n، کران‌های زیر به دست آمده‌اند

$$\frac{n}{2\pi e} \leq \gamma_n \leq \frac{n}{\pi e}\ ;\pi = 3/14159\ldots,\ e = 2/71828.$$

تذکر ۱۲.۱. صورت‌های دیگری از قضیه ۱۰.۱ موجودند که با تعداد بردارهای بیشتری سروکار دارند. به‌عنوان مثال؛ می‌توان ثابت کرد پایهٔ $\{v_1, v_2, \ldots, v_n\}$ برای مشبکهٔ n بعدی L وجود دارد که

$$||v_1||\ ||v_2|| \ldots ||v_n|| \leq n^{\frac{n}{2}} \det(L)\,.$$

تعریف ۱۳.۱. نسبت هادامارد پایهٔ $V = \{v_1, v_2, \ldots, v_n\}$ برای مشبکهٔ L برابر مقدار

$$H(V) = \frac{\det(L)}{||v_1||\ ||v_2|| \ldots ||v_n||}$$

تعریف می‌شود. داریم $0 < H(V) \leq 1$ و هنگامی که بردارها متعامد باشند، این مقدار برابر ۱ می‌شود. (معکوس نسبت هادامارد را **نقص تعامد** گویند.)

برای اثبات قضیهٔ هرمیت، نتیجه‌ای از قضیهٔ مینکوفسکی مورد استفاده قرار می‌گیرد. برای شرح قضیهٔ مینکوفسکی احتیاج به تعاریف زیر داریم.

تعریف ۱۴.۱. برای هر $a \in R^n$ و $r > 0$، گوی بسته به مرکز a و شعاع r را

$$B_r(a) = \{x \in R^n\ : ||x - a|| \leq R\}$$

تعریف می‌کنیم.

تعریف ۱۵.۱. اگر S زیرمجموعه‌ای از R^n باشد؛
الف) S را کران‌دار گویند، هرگاه طول بردارهای S کران‌دار باشد؛ یعنی $r > 0$ موجود باشد که S داخل گوی $B_r(0)$ باشد.
ب) S متقارن است، اگر برای هر $a \in S$، $-a$ نیز متعلق به S باشد.
ج) S محدب است، اگر به‌ازای $a, b \in S$، سرتاسر خط واصل a و b نیز متعلق به S باشد.
د) S بسته است، هرگاه به‌ازای هر $a \in R^n$ و $r > 0$ که $B_r(a)$ شامل حداقل یک نقطه از S باشد، داشته باشیم $a \in S$.

قضیه ۱۶.۱ (مینکوفسکی). اگر $L \in R^n$ مشبکه‌ای n بعدی و $S \in R^n$ مجموعهٔ کران‌دار، محدب و متقارن باشد که $Vol(S) > 2^n \det(L)$، آنگاه S شامل حداقل یک بردار ناصفر از مشبکهٔ L خواهد شد. به‌علاوه؛ اگر S بسته و نامساوی فوق مختار به تساوی نیز شود، حکم همچنان برقرار است.

اثبات. برای اثبات، فرض می‌کنیم $L \in R^n$ مشبکهٔ n بعدی و همچنین S ابرمکعب در R^n به مرکز صفر با طول اضلاع $2B$ باشد. به‌عبارت‌دیگر؛

$$S = \{(x_1, x_2, \ldots, x_n) \in R^n\ :\ -B \leq x_i \leq B\}\,.$$

روشن است که S مجموعه‌ای کران‌دار، بسته، محدب و متقارن است. ازآنجاکه $Vol(S) = (2B)^n$، قرار می‌دهیم $B = \det(L)^{\frac{1}{n}}$ تا شرط $2^n \det(L) \leq Vol(S)$ برقرار شود. حال قضیه ۱۶.۱ را برای $0 \neq a \in S \cap L$ به کار می‌بریم

$$||a|| = \sqrt{a_1^2 + \cdots + a_n^2} \leq \sqrt{n}B = \sqrt{n}\det(L)^{\frac{1}{n}}\,.$$

درنتیجه حکم به دست می‌آید. □

این امکان وجود دارد که ثابت ظاهرشده در قضیه ۱۰.۱ را با به‌کارگیری قضیه ۱۶.۱ برای یک ابرکره، بهبود بخشید. به‌منظور انجام این کار؛ نیاز به دانستن حجم یک گوی در R^n داریم.

تعریف ۱۷.۱. تابع $\Gamma(s)$ برای $s > 0$ برابر است با

$$\Gamma(s) = \int_0^\infty t^s e^{-t} \frac{dt}{t}.$$

گزاره ۱۸.۱ (تقریب استرلینگ). برای مقادیر بزرگ s، تابع $\Gamma(1+s)^{\frac{1}{s}}$ به‌طور تقریبی برابر $\frac{s}{e}$ است.

قضیه ۱۹.۱. اگر $B_r(a)$ گوی به شعاع r در R^n باشد، آنگاه داریم

$$Vol(B_r(a)) = \frac{\pi^{\frac{n}{2}} R^n}{\Gamma\left(1+\frac{n}{2}\right)}.$$

تذکر ۲۰.۱. هنگامی که n مقداری بزرگ باشد، بنابر گزاره ۱۸.۱، خواهیم داشت

$$Vol(B_r(a))^{\frac{1}{n}} = \sqrt{\frac{2\pi e}{n}} r.$$

دوباره قضیه ۱۶.۱ را در نظر می‌گیریم، این بار مجموعهٔ S را گوی $B_r(0)$ قرار می‌دهیم و r را طوری انتخاب می‌کنیم که

$$2^n \det(L) \leq Vol(S) \tag{۱.۱}$$

پس می‌توان اطمینان حاصل کرد که $B_r(0)$ شامل حداقل یک بردار ناصفر مشبکهٔ L است. با فرض بزرگ بودن n، بنابر تذکر ۲۰.۱ داریم

$$Vol(B_r(0))^{\frac{1}{n}} = \sqrt{\frac{2\pi e}{n}} r.$$

ازطرفی شرط (۱.۱) ایجاب می‌کند

$$\sqrt{\frac{2n}{\pi e}} \det(L)^{\frac{1}{n}} \leq r.$$

پس می‌توان نتیجه گرفت $v \in L$ موجود است که داخل این گوی قرار می‌گیرد. به‌عبارت‌دیگر

$$||v|| \leq \sqrt{\frac{2n}{\pi e}} \det(L)^{\frac{1}{n}}.$$

مشاهده می‌شود که کران بهتری نسبت به قضیه ۱۰.۱ به دست آوردیم. اگرچه کران واقعی برای اندازهٔ کوتاه‌ترین بردار در مشبکه مجهول است، اما وقتی n بزرگ باشد؛ می‌توان اندازهٔ آن را با آرگومان‌های احتمالاتی تخمین زد. فرض کنید $B_r(0)$ گویی بزرگ به مرکز صفر باشد، آنگاه تعداد نقاط مشبکهٔ L در $B_r(0)$ به‌طور تقریبی برابر است با

$$\frac{Vol(B_r(0))}{Vol(F)}.$$

حال اگر این تعداد برابر یک باشد، داریم $Vol(F) = Vol(B_r(0))$، که بنابر تذکر ۲۰.۱ به دست می‌آید

$$r \approx \sqrt{\frac{n}{2\pi e}} \det(L)^{\frac{1}{n}}.$$

تعریف ۲۱.۱. فرض کنید L مشبکهٔ تصادفی n بعدی باشد، طول کوتاه‌ترین بردار ناصفر مورد انتظار گوسی در L به‌طور تقریبی برابر است با

$$\sigma(L) = \sqrt{\frac{n}{2\pi e}} \det(L)^{\frac{1}{n}}.$$

به‌طور دقیق‌تر؛ اگر $\epsilon > 0$ ثابت باشد، آنگاه برای تمام nهای به‌قدر کافی بزرگ و مشبکه‌های تصادفی n بعدی، داریم

$$(1-\epsilon)\sigma(L) \leq ||v_{shortest}|| \leq (1+\epsilon)\sigma(L).$$

تذکر ۲۲.۱. برای nهای کوچک، بهتر است فرمول دقیق $B_r(\circ)$ استفاده شود. پس به‌منظور دستیابی به طول کوتاه‌ترین بردار مورد انتظار گوسی، خواهیم داشت

$$\begin{aligned} & Vol\left(B_r(\circ)\right) = Vol\left(F\right) \\ \Longrightarrow \quad & \frac{\pi^{\frac{n}{2}} r^n}{\Gamma\left(1+\frac{n}{2}\right)} = \det\left(L\right) \\ \Longrightarrow \quad & r = \frac{\Gamma\left(1+\frac{n}{2}\right)^{\frac{1}{n}}}{\sqrt{\pi}} \det(L)^{\frac{1}{n}} \\ \Longrightarrow \quad & \sigma\left(L\right) = \frac{\Gamma\left(1+\frac{n}{2}\right)^{\frac{1}{n}}}{\sqrt{\pi}} \det(L)^{\frac{1}{n}}. \end{aligned}$$

به‌طور مثال؛ برای $n = 6$، مقدار تقریبی $\sigma\left(L\right)$ برابر است با $0.5927 \det(L)^{\frac{1}{6}}$، درحالی‌که مقدار دقیق آن برابر است با $0.7605 \det(L)^{\frac{1}{6}}$؛ که باهم متفاوت‌اند. اما اگر $n = 100$، مقدار تقریبی $\sigma\left(L\right)$ برابر است با $2.4420 \det(L)^{\frac{1}{100}}$، درحالی‌که مقدار دقیق آن برابر است با $2.4900 \det(L)^{\frac{1}{100}}$؛ که تفاوت ناچیزی دارند.

۴.۱ پایه‌های مطلوب در مشبکه

برای شروع؛ مشبکهٔ $L\left(v_1, v_2, \ldots, v_n\right)$ را به قسمی تجسم کنید که v_iها دوبه‌دو متعامد باشند. آنگاه مسئلهٔ SVP نه‌تنها دشوار نیست، بلکه بسیار ساده به جواب نهایی می‌رسد؛ زیرا اگر $v_1, v_2, \ldots, v_n$ دوبه‌دو متعامد باشند، برای هر مجموعهٔ ضرایب دلخواه $a_1, a_2, \ldots, a_n \in Z$، داریم

$$||a_1 v_1 + a_2 v_2 + \cdots + a_n v_n||^2 = a_1^2||v_1||^2 + a_2^2||v_2||^2 + \ldots a_n^2||v_n||^2.$$

با فرض اینکه برای هر $1 \leq j \leq n$؛ v_j کوتاه‌ترین بردار پایه باشد، خواهیم داشت

$$||a_1 v_1 + a_2 v_2 + \cdots + a_n v_n||^2 \geq ||v_j||^2 \left(a_1^2 + a_2^2 + \cdots + a_n^2\right) \geq ||v_j||^2.$$

به‌عبارت‌دیگر؛ نرم هیچ ترکیب خطی از v_iها نمی‌تواند از نرم v_j کمتر باشد. اما ازآنجاکه در مورد مشبکه‌های با بعد بالا، دستیابی به پایه‌های کاملاً متعامد بعید است، هدف خود را به یافتن پایه‌های شبه‌متعامد و تاحدامکان کوتاه، تقلیل می‌دهیم. به بیانی دیگر؛ سعی بر افزایش مقدار نسبت هادامارد تا نزدیکی آن به ۱، داریم. زیرا دترمینان مشبکه، مقداری پایاست.

۵.۱ مفهوم کاهش مشبکه

اشاره کردیم که مشبکه و خصوصاً حل مسائلی مثل CVP و SVP به دلیل کاربرد بسیار گسترده در شاخه‌های مختلف علوم محض و کاربردی، برای بیش از ۱۵۰ سال، مورد توجه ریاضیدانان بوده و با کارهای «مینکوفسکی» و «هرمیت» در ابتدای قرن بیستم به اوج بالندگی رسید. با ظهور کامپیوترها به‌عنوان ابزارهای سریع محاسبات، جستجو به دنبال الگوریتم‌هایی که لااقل بتواند تقریبی (حتی نه‌چندان خوب) از کوتاه‌ترین یا نزدیک‌ترین بردار یک مشبکه با پایه‌های مفروض به دست بیاورد در دستور کار بسیاری از پژوهشگران قرار گرفت. الگوریتم‌هایی که قادر به ارائهٔ پایه‌های مطلوب برای مشبکه باشند. تا اینکه در سال ۱۹۸۲ برادران «لنسترا»[1] به همراه نابغهٔ مجارستانی «لواش»[2]، چنین الگوریتمی را معروف به **الگوریتم LLL** ارائه کردند [۲۱]؛ الگوریتمی که بعداً مشخص شد که به‌طرز عجیب و ناشناخته‌ای بهتر از کرانی که برای آن اثبات شده عمل می‌کند.

۱.۵.۱ الگوریتم LLL

کاری که ابداع‌کنندگان الگوریتم LLL انجام دادند، تعریف هوشمندانه‌ای از پایه‌های کاهش‌یافته[3] و ارائهٔ الگوریتمی کارآمد با زمان چندجمله‌ای برای محاسبهٔ چنین پایه‌هایی بود.

[1] Lenstra
[2] Lov'asz
[3] Reduced Basis

تعریف ۲۳.۱ (پایهٔ کاهش‌یافتهٔ LLL). در یک مشبکه، پایه‌های کاهش‌یافتهٔ LLL، پایه‌هایی تقریباً متعامد، همراه با بردارهایی با نرم نسبتاً کوچک هستند؛ فرض کنید $\{v_1, v_2, \ldots, v_n\}$ پایهٔ مشبکهٔ L و $\{v_1^*, v_2^*, \ldots, v_n^*\}$ پایهٔ متعامد متناظر آن (به‌دست‌آمده از الگوریتم گرام-اشمیت) باشد. در اینجا پایهٔ کاهش‌یافتهٔ $\{v_1, v_2, \ldots, v_n\}$ از یک مشبکه مثل L، با شرایط زیر تعریف می‌شود.

۱-شرط اندازه) باید برای هر $1 \leq j < i \leq n$ داشته باشیم $\left|\frac{\langle v_i, v_j^* \rangle}{||v_j^*||^2}\right| \leq \frac{1}{2}$ و این یعنی در فرآیند گرام-اشمیت، $|\mu_{i,j}| \leq \frac{1}{2}$. به‌عبارت‌دیگر زاویهٔ هر بردار v_i با v_j^* قبل از خودش بیشتر از ۶۰ درجه است؛ زیرا شرط فوق معادل است با

$$2\left|\langle v_i, v_j^* \rangle\right| \leq ||v_j^*||^2.$$

بنابر تعریف v_j^* داریم

$$||v_j^*||^2 \leq ||v_j||\ ||v_j^*||,$$

اما ازآنجاکه v_jهای قبل از v_i به ترتیب اندازه مرتب شده‌اند، داریم

$$||v_j||\ ||v_j^*|| \leq ||v_i||\ ||v_j^*||,$$

درنتیجه

$$2\left|\langle v_i, v_j^* \rangle\right| \leq ||v_i||\ ||v_j^*||\ .$$

۲- شرط لواش) باید برای هر $1 \leq i \leq n$ داشته باشیم $||v_i^*||^2 \geq \left(\frac{3}{4} - \mu_{i,j-1}^2\right) ||v_{i-1}^*||^2$.

حال به شرط الگوریتم LLL می‌پردازیم؛ این الگوریتم، بردارهای پایه و نامتعامد $\{v_1, v_2, \ldots, v_n\}$ را به‌عنوان ورودی از مشبکهٔ L گرفته و پایه‌های جدید $\{v_1', v_2', \ldots, v_n'\}$ را به قسمی تولید می‌کند که

۱. $Span\left(\{v_1, v_2, \ldots, v_n\}\right) = Span\left(\{v_1', v_2', \ldots, v_n'\}\right)$

۲. v_i'ها در هر دو شرط **اندازه** و **لواش** صدق کنند.

```
[1]   Input a basis {v_1, ..., v_n} for a lattice L
[2]   Set k = 2
[3]   Set v_1* = v_1
[4]   Loop while k ≤ n
[5]       Loop j = 1, 2, 3, ..., k − 1
[6]           Set v_k = v_k − ⌊μ_{k,j}⌉ v_j*  [Size Reduction]
[7]       End j Loop
[8]       If ||v_k*||^2 ≥ (3/4 − μ_{k,k−1}^2) ||v_{k−1}*||^2  [Lov'asz Condition]
[9]           Set k = k + 1
[10]      Else
[11]          Swap v_{k−1} and v_k  [Swap Step]
[12]          Set k = max(k − 1, 2)
[13]      End If
[14]  End k Loop
[15]  Return LLL reduced basis {v_1, ..., v_n}
```

Note: At each step, $v_1^*, \ldots, v_k^*$ is the orthogonal set of vectors obtained by applying Gram-Schmidt to the current values of $v_1, \ldots, v_k$ and $\mu_{i,j}$ is the associated quantity $\left(v_i\ .v_j^*\right) / \left|\left|v_j^*\right|\right|^2$.

۶.۱ سامانه‌های رمزنگاری نامتقارن مبتنی بر مشبکه

در خلال سال‌های ۶۱۹۹ تا ۱۹۹۸، با اثبات آن‌که حل مسائل SVP و CVP در مشبکه‌های تصادفی مسائلی مشکل هستند، روزنهٔ امیدی برای پیاده‌سازی سامانه‌های رمزنگاری نامتقارن جدید و مبتنی بر دشواری حل این مسائل در مشبکه گشوده شد. از بین سامانه‌های رمزنگاری کلید عمومی مبتنی بر مشبکه، سامانهٔ NTRU، که رسماً در سال ۱۹۹۸ معرفی شد [۱۹] توانست نهایتاً با اصلاحات زیاد اعتماد عمومی را جلب کرده و پس از استانداردسازی با عنوان IEEE P1363.1، به صنعت راه پیدا کند [۲۸]. سامانهٔ رمز NTRU را می‌توان اولین سامانهٔ عملی دانست که امنیت خود را بر اساس حل مسئلهٔ SVP می‌داند. در مقایسه با سامانه‌های رمزی شناخته‌شده‌ای مثل RSA یا ECC، بزرگ‌ترین مزیت این سامانهٔ رمز سرعت بسیار بالا و هزینه پیاده‌سازی پایین است. چراکه در این سامانه، عملیات محاسباتی با پیچیدگی $O(N)^۲$ انجام می‌گیرد و N حداکثر ۹ بیتی است.

۱.۶.۱ سامانهٔ رمزی NTRU

سامانهٔ رمزنگاری کلید عمومی NTRU برای اولین بار به‌طور غیررسمی در خلال نشست‌های جانبی اجلاس Crypto96، توسط ریاضیدان‌هایی از دانشگاه براون به نام‌های هافشتین[۱]، ژیل پایفر[۲] و جوزف سیلورمن[۳] معرفی و دو سال بعد جزییات آن به‌صورت رسمی در [۱۹] منتشر شد. در خلال یک دهه، علی‌رغم حملات مؤثری که علیه NTRU طراحی شد، تمامی این حملات با اصلاحات جزئی خنثی شدند. درحال‌حاضر هستهٔ NTRU نفوذناپذیر تلقی می‌شود. در اردیبهشت‌ماه سال ۱۳۸۸، IEEE نیز اولین نسخهٔ این سامانهٔ رمزنگاری کلید عمومی را با شناسه P1363.1 استانداردسازی و منتشر نمود که این خود دلیلی بر اعتماد عمومی و استقبال صنعت از این سامانهٔ سریع و بهینه است. اخیراً شرکت‌هایی مثل Intel، Cisco، Motorola، NXP، Sony و IBM در به‌کارگیری این الگوریتم در محصولات خود با شرکتی به همین نام (Ntru: Security Innovation)، همکاری خود را آغاز کرده‌اند.

۲.۶.۱ نمادها و عملگرها

عملیات پایه در سامانهٔ رمز NTRU، در حلقهٔ $R = \frac{Z[x]}{x^N - ۱}$ انجام می‌گیرد که در آن N عددی اول است. R شامل چندجمله‌ای‌هایی با درجهٔ $N - ۱$ است که ضرایب صحیح هستند. همچنین حلقه‌های چندجمله‌ای $R_p = \frac{Z_p[x]}{x^N - ۱}$ و $R_q = \frac{Z_q[x]}{x^N - ۱}$ که چندجمله‌ای‌هایی از درجهٔ $N - ۱$ با ضرایب صحیح به پیمانهٔ p و q، نیز در این سامانه مورد استفاده قرار می‌گیرند. اعداد p و q نسبت به هم اولند و q بسیار بزرگ‌تر از p است (به‌طور معمول $p = ۳$).

تعریف ۲۴.۱. برای هر عدد صحیح مثبت $d_۱$، $d_۲$، $\Gamma(d_۱, d_۲)$ را تعریف می‌کنیم؛ تمام چندجمله‌ای‌های متعلق به حلقهٔ R که دارای $d_۱$ ضریب ۱ و $d_۲$ ضریب -۱ است و بقیهٔ ضرایب صفر هستند. به این‌چنین چندجمله‌ای، چندجمله‌ای سه‌گانه گویند.

تعریف ۲۵.۱. فرض کنید a و b دو چندجمله‌ای از درجهٔ $N - ۱$ باشند، به‌طوری‌که

$$a = \sum_{j=۰}^{N-۱} a_j x^j \ , \quad b = \sum_{j=۰}^{N-۱} b_j x^j.$$

تعریف می‌کنیم

$$c = a * b = \sum_{i=۰}^{N-۱} c_i x^i; \quad c_i = \sum_{j=۰}^{N-۱} a_j b_{i-j}.$$

تعریف ۲۶.۱. فرض کنید a و b دو چندجمله‌ای از درجهٔ $N - ۱$ باشند، (a, b) را برداری $۲N$تایی تعریف می‌کنیم که N درایهٔ اول آن ضرایب چندجمله‌ای a و N درایهٔ دوم ضرایب چندجمله‌ای b باشد.

۳.۶.۱ عملکرد سامانهٔ NTRU

بر اساس تعاریف قسمت قبل، سامانهٔ NTRU را می‌توان به‌صورت زیر توصیف نمود:

[1] J. Hoffstein
[2] J. Pipher
[3] J. Silverman

- **کلید خصوصی:** دو چندجمله‌ای $f \in \Gamma(d_f, d_f - ۱)$ و $g \in \Gamma(d_g, d_g)$ به‌صورت تصادفی تولید می‌شوند. پس از انتخاب f و g، با استفاده از **الگوریتم تعمیم‌یافتهٔ اقلیدسی**، وارون f روی حلقه‌های R_p و R_q محاسبه شده و به‌ترتیب F_p و F_q نامیده می‌شوند. احتمال آن‌که چندجمله‌ای f روی این حلقه‌ها وارون‌پذیر باشد، بسیار بالا است. اما در غیر این صورت می‌توان چندجمله‌ای f جدیدی تولید کرد.

- **کلید عمومی:** کلید عمومی سامانهٔ NTRU، چندجمله‌ای h است که به‌صورت زیر محاسبه می‌شود

$$h = F_q * g \ (mod\, q)\,.$$

لازم به ذکر است که مقادیر d_g، d_f، q، p، N و d_r نیز به‌صورت عمومی منتشر می‌شوند. (مقدار d_r در قسمت رمزنگاری به کار می‌رود.)

- **رمزنگاری:** در فرایند رمزنگاری، سامانهٔ رمز ابتدا یک چندجمله‌ای تصادفی $r \in \Gamma(d_r, d_r)$ انتخاب کرده و پیام ورودی را در قالب یک چندجمله‌ای $m \in R$ با ضرایبی بین $\frac{p}{۲}$ و $-\frac{p}{۲}$ تبدیل می‌کند. متن رمزشده به‌صورت زیر محاسبه و ارسال می‌شود

$$e = p.h * r + m \ (mod\, q)\,.$$

- **رمزگشایی:** گیرنده برای محاسبهٔ چندجمله‌ای m، با در اختیار داشتن چندجمله‌ای e به‌صورت زیر عمل می‌کند

$$f * e = f * (p.h * r + m) \ (mod\, q) \Longrightarrow f * e = p.f * h * r + f * m \ (mod\, q)\,.$$

با جایگذاری h خواهیم داشت

$$e * f = p.f * F_q * g * r + f * m \ (mod\, q)$$

$$\Longrightarrow f * e = p.g * r + f * m \ (mod\, q)\,.$$

ازآنجاکه چندجمله‌ای‌های r، g، f و m دارای ضرایب کوچک هستند و همچنین مقدار p می‌توان مطمئن بود که ضرایب چندجمله‌ای فوق در بازه $\left[-\frac{q}{۲}, \frac{q}{۲}\right)$ واقع می‌شوند (احتمال عدم وقوع این پیشامد برای سامانه‌ای با پارامترهای $d_r = ۱۸$، $d_g = ۲۰$، $d_f = ۶۱$، $q = ۱۲۸$، $P = ۳$، $N = ۱۶۷$ چیزی نزدیک به $۱۰^{-۵}$ است). پس داریم

$$f * g = p.g * r + f * m.$$

در ادامه؛ گیرنده مقدار $f * e * F_p \ (mod\, p)$ را محاسبه می‌کند

$$\begin{aligned} f * e * F_p = p.g * r * F_p + m * f * F_p \ (mod\, p) \\ \Longrightarrow \ f * e * F_p = m * f * F_p \ (mod\, p) \\ \Longrightarrow \ f * e * F_p = m \ (mod\, p)\,. \end{aligned}$$

اما می‌دانیم ضرایب چندجمله‌ای m بین $-\frac{p}{۲}$ و $\frac{p}{۲}$ قرار دارند؛ پس نتیجه می‌شود

$$f * e * F_p = m.$$

۴.۶.۱ مشبکه و امنیت سامانهٔ NTRU

بنابر آنچه گفته شد؛ به‌منظور حمله بر سامانهٔ رمزی NTRU، می‌بایست با استفاده از کلید عمومی h، کلیدهای خصوصی f و g را به دست آورد. به‌عبارت‌دیگر؛ یافتن بردار (f, g) هدف حمله‌کننده به این سامانه است. نشان خواهیم داد؛ سامانهٔ رمزی NTRU، سامانهٔ مبتنی بر مشبکه بوده و حمله به این سامانه، معادل با حل مسئلهٔ SVP است. می‌توان کلید عمومی h را به‌صورت چندجمله‌ای

$h(x) = h_0 + h_1 x + \dots h_{N-1} x^{N-1}$ در نظر گرفت. مشبکهٔ منتسب به h را با L_h^{NTRU} نمایش داده که توسط ماتریس

$$M_h^{NTRU} = \begin{pmatrix} 1 & 0 & \dots & 0 & h_0 & h_1 & \dots & h_{N-1} \\ 0 & 1 & \dots & 0 & h_{N-1} & h_0 & \dots & h_{N-2} \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & 1 & h_1 & h_2 & \dots & h_0 \\ 0 & 0 & \dots & 0 & q & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & q & \dots & 0 \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & q \end{pmatrix}_{2N \times 2N}$$

تولید می‌شود.

گزاره ۲۷.۱. با فرض اینکه $f(x) * h(x) = g(x) \pmod q$، $u(x)$ متعلق به R موجود است که $f(x) * h(x) = g(x) + qu(x)$ آنگاه $M_h^{NTRU} = (f, g)$. $(f, -u)$.

اثبات. با توجه به تعریف ماتریس M_h^{NTRU} و تساوی $f(x) * h(x) = g(x) + qu(x)$ حکم ثابت می‌شود. □

نتیجه ۲۸.۱. بردار (f, g) متعلق به مشبکهٔ L_h^{NTRU} است.

گزاره ۲۹.۱. فرض کنید (N, p, q, d_f, d_g, d_r) پارامترهای سامانهٔ رمزی $NTRU$ باشد و

$$d_f = d_g = d_r = \frac{N}{3},\ q \approx 2N,$$

آنگاه (f, g) برداری کوتاه در مشبکهٔ L_h^{NTRU} خواهد بود.

اثبات. با توجه به آن‌که $f \in \Gamma(d_f, d_f - 1)$ و $g \in \Gamma(d_g, d_g)$، داریم

$$||(f, g)|| \approx \sqrt{2d_f + 2d_g} = \sqrt{\frac{4N}{3}}.$$

ازطرفی؛ شهود گوسی پیش‌بینی می‌کند که

$$\sigma(L) \approx \sqrt{\frac{N}{2\pi e}}$$

$$\Longrightarrow \frac{||(f, g)||}{\sigma(L)} \approx \frac{2/39}{\sqrt{N}}.$$

پس $||(f, g)||$ برای Nهای نسبتاً بزرگ بسیار کمتر از میانگین تخمین‌زده‌شده توسط شهود گوسی است. پس می‌توان نتیجه گرفت که بردار (f, g)، کوتاه‌ترین بردار در مشبکهٔ L_h^{NTRU} است. □

با در نظر گرفتن گزارهٔ فوق، می‌توان دریافت که سامانهٔ رمزنگاری کلید عمومی NTRU امنیت خود را از سختی مسئلهٔ SVP به عاریه گرفته است. در جدول ۲، می‌توان پارامترهای سامانهٔ NTRU را در سطوح مختلف امنیت مقایسه کرد. همچنین جدول ۳، پارامترهای این سامانه را به طور کامل معرفی می‌کند.

جدول ۱: سامانهٔ رمزنگاری NTRU

Public Parameter Creation	
A trusted party chooses public parameters (N, p, q, d) with N and p prime, $\gcd(p, q) = \gcd(N, q) = 1$, and $q > (6d+1)\,p$.	
Alice	Bob
Key Creation	
Choose private $f \in T(d+1, d)$ that is invertible in R_q and R_p. Choose private $g \in T(d, d)$. Compute F_q, the inverse of f in R_q. Compute F_p, the inverse of f in R_p. Publish the public key $h = F_q * g$.	
Encryption	
	Choose plaintext $m \in R_p$. Choose a random $r \in T(d, d)$. Use Alice's public key h to compute $e \equiv prh + m \pmod q$. Send ciphertext e to Alice.
Decryption	
Compute $f * e \equiv pg * r + f * m \pmod q$ Centerlift to $a \in R$ and compute $m \equiv F_p * a \pmod p$	

جدول ۲: پارامترهای سامانه NTRU در سطوح مختلف امنیت

	N	p	Q
امنیت پایین	۱۶۷	۳	۱۲۸
امنیت استاندارد	۲۵۱	۳	۱۲۸
امنیت بالا	۳۴۷	۳	۱۲۸
امنیت فوق بالا	۵۰۳	۳	۲۵۶

جدول ۳: پارامترهای سامانه NTRU

	N	p	q	d_f	d_g	d_r
NTRU167:3	۱۶۷	۳	۱۲۸	۶۱	۲۰	۱۸
NTRU251:3	۲۵۱	۳	۱۲۸	۵۰	۲۴	۱۶
NTRU503:3	۵۰۳	۳	۲۵۶	۲۱۶	۷۲	۵۵
NTRU167:2	۱۶۷	۲	۱۲۷	۴۵	۳۵	۱۸
NTRU251:2	۲۵۱	۲	۱۲۷	۳۵	۳۵	۲۲
NTRU503:2	۵۰۳	۲	۲۵۳	۱۵۵	۱۰۰	۶۵

۲ معرفی

مشبکه‌های NTRU دسته‌ای از مشبکه‌های دارای دریچه هستند که توسط [۱۹] به‌عنوان اساس طراحی الگوریتم رمزگذاری نامتقارن NTRUEncrypt معرفی شدند. برای یک چندجمله‌ای مونیک $\phi \in \mathbb{Z}[x]$ از درجهٔ n، مشبکه توسط دو چندجمله‌ای "کوتاه" f و g در مد ϕ ایجاد می‌شود. ضرایب f و g اعداد صحیح بسیار کوچکی هستند (در NTRUEncrypt، آن‌ها به $\{-۱, ۰, ۱\}$ محدود می‌شوند). چندجمله‌ای f و g مخفی هستند، اما نسبت بین آن‌ها

$$h = \frac{g}{f} \bmod \phi \bmod q \tag{۱.۲}$$

عدد صحیح کوچک q، مقداری عمومی و آشکار است. چندجمله‌ای f طوری انتخاب می‌شود که در مد ϕ و q معکوس‌پذیر باشد. Q لزوماً اول نیست. مشبکه‌های NTRU ویژگی‌های عملکردی خوبی را ارائه می‌دهند؛ آن‌ها در چندین طرح نامتقارن دیگر مورد استفادهٔ مجدد قرار گرفته‌اند. برخی از این طرح‌ها نیاز دارند که دریچهٔ مشبکه کامل باشد، به این معنی که فراتر از دانستن f و g، مالک کلید خصوصی باید دو چندجمله‌ای کوتاه دیگر F و G را نیز بداند که معادلهٔ NTRU زیر را تکمیل می‌کنند:

$$fG - gF = q \tag{۲.۲}$$

به‌عنوان مثال در طرح امضا NTRUSign [۱۸]، یک طرح رمزگذاری مبتنی بر هویت [۱۰]، طرح امضا Falcon [۱۳] و طرح رمزگذاری مبتنی بر هویت سلسله مراتبی LATTE [۴] به یک دریچهٔ کامل NTRU نیاز است. یافتن کوتاه‌ترین راه‌حل (برای یک نرم داده‌شده) مسئله‌ای سخت است. بااین‌حال، محاسبهٔ راه‌حلی که برای اجرای یک الگوریتم بر اساس مشبکه‌های کامل NTRU به اندازهٔ کافی کوتاه باشد، امکان‌پذیر است. به‌این‌ترتیب حل معادلهٔ NTRU بخشی از فرآیند تولید کلید محسوب می‌شود. درحالی‌که معادلهٔ NTRU ساده به نظر می‌رسد، حل آن به شیوه‌ای کارآمد مسئله‌ای بدیهی نیست. الگوریتم‌های موجود برای یافتن یک جواب [۱۸، ۲۶] در بعد n، به‌ترتیب دارای پیچیدگی زمانی و حافظهٔ حداقل از درجهٔ سه و دو هستند. برای اندازهٔ پارامترهای متداول، این در عمل یعنی نیاز به چندین مگابایت RAM که در یک کامپیوتر معمولی در حدود ۲ ثانیه زمان خواهد برد. این مانع از پیاده‌سازی در بسیاری از سامانه‌های محدود و تعبیه‌شده می‌شود. می‌توان استدلال کرد که توانایی پیاده‌سازی تولید کلید در یک دستگاه تعبیه شده چندان مهم نیست زیرا می‌توان آن را به‌صورت خارجی تولید کرد و کلید را در دستگاه کپی نمود، اما نگه‌داشتن کلید خصوصی در یک دستگاه مقاوم در برابر دست‌کاری برای چرخه عمر کامل آن اغلب برای امنیت و انطباق (به‌عنوان مثال با استاندارد FIPS 140-2 [۱۲]) مطلوب است. در این مقاله، نشان می‌دهیم که چگونه می‌توانیم از نرم میدان در حلقه‌های چندجمله‌ای، برای دستیابی به عملکردی بسیار بهبودیافته در حل معادلهٔ NTRU استفاده کنیم. این به ما اجازه می‌دهد تا دو الگوریتم جدید را برمبنای نرم میدان معرفی نماییم، که پیچیدگی (زمان و حافظه) بهتری را نسبت به الگوریتم‌های موجود با عوامل شبه‌خطی برحسب n (دقیقاً، حداقل $O(n/\log n)$) ارائه می‌دهد. به‌عنوان یک محصول جانبی، ما یک الگوریتم بهبودیافته را برای محاسبهٔ برآیندهای چندجمله‌ای، زمانی که یکی از چندجمله‌ای‌ها سیکلوتومیک باشد، توسعه دادیم. (به بخش ۳ مراجعه کنید). جدول ۴ پیچیدگی مجانبی به‌دست‌آمده توسط روش جدید ما را با روش‌های شناخته‌شدهٔ موجود مقایسه می‌کند. ما حل‌کنندهٔ کلاسیک NTRU مبتنی بر نتیجه و نیز الگوریتم جدید خود را، با بهینه‌سازی و ابزارهای مشابه پیاده‌سازی کردیم. این مسئله امکان اندازه‌گیری مستقیم ارتقای عملکرد روش ما را فراهم کرد، که تجزیه‌وتحلیل مجانبی را تائید نمود: برای یک درجهٔ معمولی ($n = ۱۰۲۴$)، روش جدید سریع‌تر و کوچک‌تر از الگوریتم‌های کلاسیک است، هر دو با یک ضریب ۱۰۰ یا بیشتر.

۱.۲ روش

الگوریتم ما به استفادهٔ مکرر از پارادایم تصویر-کن-سپس-بالا-ببر متکی است، یک پارادایم معروف در نظریهٔ اعداد الگوریتمی و تحلیل رمزی، که شامل تصویر کردن مسئله بر روی یک زیرمجموعه است که در آن آسان‌تر می‌شود، قبل از اینکه راه‌حل را به مجموعهٔ اصلی ببریم. ما بر استفاده از وجود برج‌های میدان و برج‌های حلقه تکیه می‌کنیم. به‌عنوان مثال، برج میدان‌های زیر را در نظر بگیریم:

$$\mathbb{K}_\ell / \mathbb{K}_{\ell-۱} / \ldots / \mathbb{K}_۱ / \mathbb{K}_۰ = \mathbb{Q}$$

که در آن $\forall i$ ، $\mathbb{K}_i = \mathbb{Q}[x] / \left(x^{۲^i}+۱\right)$ ، و برج حلقه‌های مرتبط (که حلقه‌هایی از اعداد صحیح از میدان‌های مربوطه هستند) با $n = ۲^\ell$:

$$\mathbb{Z}[x] / (x^n+۱) \not\supseteq \mathbb{Z}[x] / \left(x^{n/۲}+۱\right) \not\supseteq \cdots \supseteq \mathbb{Z}[x] / \left(x^۲+۱\right) \not\supseteq \mathbb{Z}.$$

می‌دانیم که نرم میدان می‌تواند هر عنصر $f \in \mathbb{Z}[x]/(x^n+1)$ را بر روی حلقهٔ کوچک‌تری از برج خود ترسیم کند. این واقعیت در حمله "NTRU بیش‌ازحد کشیده‌شده" [۱] مورد استفاده قرار می‌گیرد، جایی که مسائل به یک حلقهٔ کوچک‌تر نگاشته می‌شوند، سپس حل شده و جواب به حلقهٔ اصلی برمی‌گردد. بااین‌حال، چیزی که در این آثار مورد استفاده قرار نمی‌گیرد، این واقعیت است که نرم میدان با برج‌های میدان‌ها به‌خوبی بازی می‌کند: برای یک برج از توسعه‌های میدان $\mathbb{L}/\mathbb{K}/\mathbb{J}$ و $f \in \mathbb{L}$ ، داریم $\mathrm{N}_{\mathbb{K}/\mathbb{J}} \circ \mathrm{N}_{\mathbb{L}/\mathbb{K}}(f) = \mathrm{N}_{\mathbb{L}/\mathbb{J}}(f)$ (که در آن N نرم میدان را نشان می‌دهد). این واقعیت در قلب الگوریتم‌های ما قرار دارد. ما ابتدا به‌طور مکرر از نرم میدان برای تصویر کردن معادلاتی بر روی $\mathbb{Z}$ استفاده می‌کنیم که در اصل بر روی $\mathbb{Z}[x]/(x^n+1)$ تعریف شده‌اند؛ این مرحلهٔ نزول است. در این مرحله معلوم می‌شود که این معادلات را بر روی $\mathbb{Z}$ خیلی سریع‌تر می‌توان حل کرد. سپس از ویژگی‌های نرم میدان برای برگرداندن جواب‌هایمان به $\mathbb{Z}[x]/(x^n+1)$ استفاده می‌کنیم؛ این مرحلهٔ بلند کردن است. این اصل ساده به ما این امکان را می‌دهد که نسبت به الگوریتم‌های کلاسیک، حداقل از مرتبهٔ $\tilde{O}(n)$ بهبود به دست آوریم. ما چند ترفند مضاعف مانند تنبلی حافظه، استفاده از سامانه‌های اعداد باقیمانده، و یا این واقعیت که در میدان‌های سیکلوتومیک، مزدوج‌های گالوایی یک عنصر در نمایش FFT یا NTT، محاسبهٔ ساده و سرراستی دارد را نیز استفاده می‌کنیم. این روش‌ها پیاده‌سازی ما را سریع‌تر و از نظر حافظه کارآمدتر می‌کنند.

جدول ۴: مقایسهٔ روش جدید ما برای حل معادلهٔ NTRU با روش‌های موجود. B نشان‌دهندهٔ کران بالایی در $\log||f||$، $\log||g||$ است. تگ $[K]$ نشان می‌دهد که الگوریتم Karatsuba برای ضرب اعداد صحیح بزرگ استفاده شده است و $[SS]$ نشان می‌دهد که الگوریتم شونهاگ-استراسن استفاده شده است.

Method	Time complexity	Space complexity
Resultant [18]	$\tilde{O}(n(n2+B))$	$O(n2B)$
HNF [26]	$\tilde{O}(n3B)$	$O(n2B)$
TowerSolverR (Algorithm 4)	$O((nB)\log 2(3)\log n)$ $[K]$	$O(n(B+\log n)\log n)$
	$\tilde{O}(nB)$ $[SS]$	

۲.۲ کاربردها

الگوریتم‌های جدید ما حداقل چهار طرح مبتنی بر مشبکهٔ موجود را تحت تأثیر قرار می‌دهند.

NTRUSign. اولین طرحی که مستلزم حل این معادله در تولید کلید است، NTRUSign [۱۸] است. اگرچه در شکل فعلی، این طرح به دلایلی مستقل از تولید کلید، ناامن است.

Falcon. در طرح امضای فالکون [۱۳]، پرهزینه‌ترین بخش تولید کلید شامل حل یک معادلهٔ NTRU است. بدون روش‌های ما، برای داشتن بالاترین سطح امنیتی، در حدود ۲۳۳ کلاک پردازنده در یک لپ‌تاپ نسبتاً جدید و ۳ مگابایت حافظه نیاز است که این مسئله، کاربرد آن در دستگاه‌های تعبیه‌شده را محدود می‌کند. با توجه به اینکه ما ازلحاظ سرعت و حافظه، بهبودی از مرتبهٔ ۱۰۰ را به دست می‌آوریم، به‌طور قابل توجهی محدودهٔ دستگاه‌هایی را که می‌توان Falcon را به‌طور کامل روی آن‌ها پیاده‌سازی کرد، افزایش خواهیم داد.

DLP. مرحلهٔ راه‌اندازی طرح رمزگذاری مبتنی بر هویت DLP [۱۰] با تولید کلید فالکون یکسان است. بنابراین، آنچه در بالا ذکر شد اینجا نیز صادق است.

LATTE. اخیراً کمپبل و گرووز [۴] LATTE را معرفی کردند؛ یک طرح رمزگذاری مبتنی بر هویت سلسله مراتبی که اساساً [۱۰] را با ساختمان درختان بونسایِ [۵] ترکیب می‌کند. در هر استخراج یک کلید مخفی، LATTE باید یک معادلهٔ تعمیم‌یافتهٔ NTRU را حل کند. به‌طور دقیق‌تر، برای $f_1,\ldots,f_k \in \mathbb{Z}[x]/(\phi)$ این طرح نیاز دارد تا $F_1,\ldots,F_k \in \mathbb{Z}[x]/(\phi)$ را محاسبه نماید به‌طوری‌که:

$$\sum f_i F_i = q$$

و k ممکن است در عمل برابر با ۳ یا ۴ باشد (به [۴]، اسلاید ۲۳] مراجعه کنید). روش ما را می‌توان به‌سادگی برای حل این نوع معادله گسترش داد. تأثیر این روش‌ها بر روی LATTE حتی مهم‌تر از تأثیر آن‌ها روی طرح‌های فوق‌الذکر است، زیرا ممکن است یک مرجع نیاز به انجام استخراج‌های زیادی داشته باشد (معمولاً یک‌بار برای هر کاربر و برای هر دورهٔ تمدید کلید). برای اطلاع از مشخصات کامل‌تر LATTE به [۹] مراجعه نمایید.

الگوریتم‌های ما در تولید کلید طرح‌های دیگری همچون BAT [۱۴]، و مشتقات فالکون مانند ModFalcon [۷] و Mitaka [۱۱] مورد استفاده قرار گرفته‌اند.

۳.۲ کارهای مرتبط

معادلهٔ NTRU برای اولین بار در [۱۸] معرفی و حل شد.
روش دیگری برای حل معادلهٔ NTRU توسط Stehlé و Steinfeld [۲۶] با استفاده از فرم نرمال هرمیت پیشنهاد شد. کارآمدترین الگوریتم ازلحاظ فضا برای محاسبهٔ HNF مربوط به Miciancio و Warinschi است [۲۲]؛ بااین‌حال، مانند روش مبتنی بر نتایج، دارای پیچیدگی فضای درجهٔ دوم و پیچیدگی‌های زمانی شبه‌مکعبی / شبه-درجه-سه است و مشکل استفاده از RAM را حل نمی‌کند.
استفاده‌ای که ما از نرم میدان انجام می‌دهیم یادآور حمله "NTRU بیش‌ازحد کشیده‌شده" توسط [۱] است، با این تفاوت که این آثار، تحلیل رمزنگاری بوده و تنها یک‌بار از نرم میدان استفاده می‌کنند، درحالی‌که در کار ما به‌طور مکرر از آن استفاده می‌شود و ساختارهای رمزنگاری را بهبود می‌بخشد.

۴.۲ نقشهٔ راه

در بخش ۲، نمادها را معرفی می‌کنیم و الگوریتم کلاسیک مبتنی-بر-نتیجه را یادآوری می‌کنیم؛ ما همچنین برخی از ابزارهای ریاضی شناخته‌شده را که در الگوریتم جدید خود استفاده خواهیم کرد، توضیح می‌دهیم. در بخش ۳، یک روش جدید برای محاسبهٔ موارد خاص از نتایج را ارائه می‌کنیم؛ الگوریتم جدید ما مبتنی بر این روش بوده و در بخش ۴ توضیح داده شده است، همچنین نشان می‌دهیم که چگونه می‌توان آن را به‌عنوان یک بهینه‌سازی الگوریتم کلاسیک مبتنی-بر-نتیجه مشاهده کرد. مسائل پیاده‌سازی در بخش ۵ مورد بحث قرار گرفته‌اند.

۳ مقدمات

حلقهٔ اعداد صحیح و میدان‌های اعداد گویا، حقیقی و مختلط را با $\mathbb{Z}$، $\mathbb{Q}$، $\mathbb{R}$ و $\mathbb{C}$ نشان می‌دهیم. برای $a>0, b>1$، ما لگاریتم a در مبنای b را با $\log_b a$ نشان می‌دهیم، و قرارداد می‌کنیم $\log a=\log_2 a$. برای عدد صحیح $r>0$، حلقهٔ اعداد صحیح به پیمانهٔ r را با $\mathbb{Z}_r$ نشان می‌دهیم.

۱.۳ حلقه‌ها و میدان‌های چندجمله‌ای

فرض کنید $\mathbb{Z}[x]$ حلقهٔ چندجمله‌ای‌ها با ضرایب صحیح باشد (ازاین‌پس آن‌ها را چندجمله‌ای‌های انتگرالی خواهیم نامید). فرض کنید ϕ یک چندجمله‌ای انتگرالی مونیک غیرصفر از درجهٔ $n\geq 1$ باشد (یعنی $\phi=x^n+\sum_{i=0}^{n-1}\phi_i x^i$). تقسیم اقلیدسی هر چندجمله‌ای انتگرالی بر ϕ به‌خوبی تعریف شده است و باقیمانده‌ای یکتا از درجهٔ کمتر از n را خواهد داد؛ بنابراین می‌توانیم $\mathbb{Z}[x]/(\phi)$، حلقهٔ چندجمله‌ای‌های انتگرالی به پیمانهٔ ϕ را تعریف کنیم. به‌طور مشابه، ما $\mathbb{Z}_r[x]/(\phi)$ و $\mathbb{Q}[x]/(\phi)$,$\mathbb{C}[x]/(\phi)$ را تعریف می‌کنیم. وقتی ϕ در $\mathbb{Z}[x]$ تحویل‌ناپذیر باشد، در $\mathbb{Q}[x]$ نیز تحویل‌ناپذیر بوده و $\mathbb{Q}[x]/(\phi)$ یک میدان است. در این مقاله، روی چندجمله‌ای‌های به پیمانهٔ ϕ که در $\mathbb{Q}[x]$ تحویل‌ناپذیر هستند کار خواهیم کرد؛ اما در حالت کلی، $\mathbb{C}[x]/(\phi)$ و $\mathbb{Z}_r[x]/(\phi)$ میدان نیستند.

۲.۳ ماتریس‌ها و بردارها

درحالی‌که هدف استفاده از حلقه‌های چندجمله‌ای برای نشان دادن مشبکه‌ها، اجتناب از محاسبات مربوط به ماتریس‌ها و بردارها است، ما همچنان از چنین اشیاء جبری در برخی برهان‌ها استفاده خواهیم کرد. ماتریس‌ها را با حروف بزرگ پررنگ (مثلاً $\mathbf{B}$) و بردارها را با حروف کوچک پررنگ (مثلاً $\mathbf{v}$) نشان خواهیم داد. ما بردارها را به‌صورت ردیفی نمایش می‌دهیم. p-نرم یک بردار v را با $\|\mathbf{v}\|_p$ نشان می‌دهیم و طبق قرارداد، $\|\mathbf{v}\|=\|\mathbf{v}\|_2$. یادآوری می‌کنیم که برای $\mathbf{v}\in\mathbb{C}^n$ و $0<r\leq p\leq\infty$، و با قرارداد کردن اینکه $1/\infty=0$ داریم:

$$\|\mathbf{v}\|_p\leq\|\mathbf{v}\|_r\leq n^{\left(\frac{1}{r}-\frac{1}{p}\right)}\|\mathbf{v}\|_p \tag{۱.۳}$$

برای یک چندجمله‌ای $f \in \mathbb{C}[x]/(\phi)$، که در آن ϕ یک چندجمله‌ای مونیک از درجهٔ n است، $\mathcal{C}_\phi(f)$ ماتریسی $n \times n$ را نشان می‌دهد که ردیف jام آن از ضرایب $x^{j-1} f \bmod \phi$ تشکیل شده است:

$$\mathcal{C}_\phi(f) = \begin{bmatrix} f \bmod \phi \\ xf \bmod \phi \\ \cdots \\ x^{n-1} f \bmod \phi \end{bmatrix} \tag{۲.۳}$$

هنگامی‌که ϕ از زمینهٔ بحث مشخص باشد، ما این ماتریس را به‌صورت ساده با $\mathcal{C}(f)$ نمایش می‌دهیم. می‌توان بررسی کرد که وقتی $\phi = x^n + 1$، ماتریس $\mathcal{C}_\phi(f)$ یک ماتریس چرخشی است. عملگر $f \in \mathbb{C}[x]/(\phi) \mapsto \mathcal{C}(f)$ یک هم‌ریختی حلقه بر روی تصویرش است. به‌طور خاص، برای همهٔ $f, g \in \mathbb{C}[x]/(\phi)$، داریم:

$$\begin{aligned} \mathcal{C}(f+g) &= \mathcal{C}(f) + \mathcal{C}(g) \\ \mathcal{C}(fg) &= \mathcal{C}(f)\,\mathcal{C}(g) \end{aligned} \tag{۳.۳}$$

۳.۳ ضرب سریع اعداد صحیح

روش‌های ما، زمانی که برای حل معادلهٔ NTRU به کار می‌روند، مستلزم استفاده از اعداد صحیح بزرگ هستند. هزینه‌های محاسباتی مجانبی، به پیچیدگی زمانی ضرب دو عدد صحیح بزرگ بستگی دارد. هنگامی‌که اندازهٔ بیتی این دو عدد صحیح با b محدود شود، آن پیچیدگی را با $\mathcal{M}(b)$ نشان می‌دهیم:

- اگر از الگوریتم Karatsuba استفاده کنیم، $\mathcal{M}(b) = O\left(b^{\log_2(3)}\right) \approx O\left(b^{1.585}\right)$؛
- با الگوریتم شونهاگ-استراسن [۲۴]، $\mathcal{M}(b) = \Theta(b \cdot \log b \cdot \log\log b)$.

الگوریتم کاراتسوبا برای مقادیر «کوچک» b کارآمدتر است، درحالی‌که شونهاگ-استراسن به‌طور مجانبی بهتر است. هنگام ارائهٔ پیچیدگی‌های زمانی برای الگوریتم‌های ارتقاءیافتهٔ خود، هر دو روش را در نظر می‌گیریم. باید توجه داشت که پیچیدگی مجانبی، فقط برای پارامترهای "به‌اندازهٔ کافی بزرگ" تخمین معقولی از عملکرد را به دست می‌دهد. در پیاده‌سازی‌های خود متوجه شدیم که برای پارامترهای معمولی (درجهٔ n حداکثر تا ۱۰۲۴)، گلوگاه عملکردی، ضرب عدد صحیح نیست، بلکه کاهش Babai است که مستلزم انجام عملیات اعشاری است.

۴.۳ چندجمله‌ای‌های سیکلوتومیک

اکثر الگوریتم‌های رمزنگاری مبتنی بر مشبکه که از حلقه‌های چندجمله‌ای برای نمایش مشبکه‌های ساختاریافته استفاده می‌کنند، به چندجمله‌ای‌های سیکلوتومیک متکی هستند (البته به‌استثنای برخی موارد قابل توجه مانند [۳، ۲۵]). چندجمله‌ای‌های سیکلوتومیک دارای برخی ویژگی‌ها هستند که آن‌ها را برای استفاده از نرم میدان ایده‌آل می‌کند.

تعریف ۱.۳. برای یک عدد صحیح $m \geq 1$، m-امین چندجمله‌ای سیکلوتومیک به‌صورت زیر است:

$$\Phi_m = \prod_{\substack{0 < k < m \\ \gcd(k,m) = 1}} \left(x - e^{2i\pi(k/m)}\right) \tag{۴.۳}$$

چندجمله‌ای‌های سیکلوتومیک دارای ویژگی‌های شناخته‌شدهٔ زیر هستند:

- آن‌ها در $\mathbb{Z}[x]$ بوده و در $\mathbb{Q}[x]$ تحویل‌ناپذیر هستند.
- درجهٔ Φ_m، $\varphi(m)$ است، که φ تابع اویلر را نشان می‌دهد: $\varphi(m) = |\mathbb{Z}_m^\times|$.
- اگر $n = 2^\ell$، آنگاه $\Phi_{2n} = x^n + 1$.

- اگر p یک عامل اول m باشد، آنگاه:

$$\Phi_{\mathrm{mp}}\ (x)=\Phi_m\left(x^P\right) \tag{۵.۳}$$

ازآنجایی‌که چندجمله‌ای‌های سیکلوتومیک تحویل‌ناپذیر هستند، $\mathbb{Q}[x]/\left(\Phi_m\right)$ برای همهٔ $m\geq ۱$ یک میدان است؛ ما آن‌ها را میدان‌های سیکلوتومیک می‌نامیم.

۵.۳ نرم میدان

نرم میدان ابزار اصلی‌ای است که ما در الگوریتم‌های خود استفاده می‌کنیم، و همین دلیل کارایی آن‌ها است. در این بخش، ما تعریف و همچنین چند ویژگی آن را یادآوری می‌کنیم.

تعریف ۲.۳ (نرم میدان). فرض کنید $\mathbb{K}$ یک میدان عددی باشد، و $\mathbb{L}$ یک توسیع گالوایی از $\mathbb{K}$ باشد. گروه گالوایِ توسیع میدان $\mathbb{L}/\mathbb{K}$ را با $\mathrm{Gal}(\mathbb{L}/\mathbb{K})$ نشان می‌دهیم.
نرم میدان $\mathrm{N}_{\mathbb{L}/\mathbb{K}}:\mathbb{L}\to\mathbb{K}$ نگاشتی است که برای هر $f\in\mathbb{L}$ توسط حاصل‌ضرب مزدوج‌های گالوا f تعریف می‌شود:

$$\mathrm{N}_{\mathbb{L}/\mathbb{K}}(f)=\prod_{\mathrm{g}\in\mathrm{Gal}(\mathbb{L}/\mathbb{K})}\mathrm{g}(f) \tag{۶.۳}$$

به‌طور معادل، $\mathrm{N}_{\mathbb{L}/\mathbb{K}}(f)$ را می‌توان به‌عنوان دترمینان نگاشت $\mathbb{K}$-خطی $\psi_f:a\in\mathrm{L}\mapsto fa$ تعریف کرد.

از این تعریف مشخص است که نرم میدان یک مورفیسم ضربی است. علاوه‌بر این، نرم میدان با ترکیب سازگار است: برای یک برجِ توسیع‌های $\mathbb{L}/\mathbb{K}/\mathbb{J}$، رابطهٔ $\mathrm{N}_{\mathbb{L}/\mathbb{K}}\circ\mathrm{N}_{\mathbb{K}/\mathbb{J}}(f)=\mathrm{N}_{\mathbb{L}/\mathbb{J}}(f)$ برقرار است. برای اختصار، $\mathbb{K}$ و $\mathbb{L}$ را می‌توان زمانی که از متن بحث روشن باشد، از زیرنویس حذف نمود. به‌عنوان مثال، وقتی $f\in\mathbb{L}$ و $\mathbb{K}$ بزرگ‌ترین زیرمیدان سره و یکتای $\mathbb{L}$ است، آنگاه داریم $\mathrm{N}(f)=\mathrm{N}_{\mathbb{L}/\mathbb{K}}(f)$.
به‌علاوه، اگر $f\in\mathbb{L}$ و $\mathbb{L}$ بر بالای یک برج میدان قرار بگیرد که از متن بحث مشخص باشد، آنگاه می‌توانیم i مرتبه ترکیب N را با $\mathrm{N}^i(f)$ نمایش دهیم. برای مثال، اگر برج میدان زیر را در نظر بگیریم:

$$\mathrm{Q}[x]/\left(x^n+۱\right)/\mathrm{Q}[x]/\left(x^{n/۲}+۱\right)/\ldots/\mathrm{Q}[x]/\left(x^۲+۱\right)/\mathbb{Q} \tag{۷.۳}$$

که در آن $n=۲^\ell$، آنگاه $\mathrm{N}^i(f)$ چندجمله‌ای $f\in\mathbb{Q}[x]/\left(x^n+۱\right)$ را به $\mathbb{Q}[x]/\left(x^{n/(۲^i)}+۱\right)$ می‌فرستد.
حالت توسیع‌های سیکلوتومیک. برای توسیع‌های سیکلوتومیک، نرم میدان را می‌توان به شکلی که برای ما راحت باشد بیان نمود. فرض کنید $m,n>۰$، اعدادی صحیح باشند به‌طوری‌که $n|m,\ \mathbb{L}=\mathbb{Q}[x]/\left(\Phi_m\right)$ و $\mathbb{K}=\mathbb{Q}[y]/\left(\Phi_n\right)$. مورفیسم $y\mapsto x^{m/n}$ یک توسیع میدان $\mathbb{L}/\mathbb{K}$ را تعریف می‌کند. سپس مزدوج‌های گالوای $\mathrm{g}_a(f)$ از $f\in\mathbb{L}$ به شکل:

$$\mathrm{g}_a(f)(x)=f\left(x^a\right) \tag{۸.۳}$$

خواهند بود؛ برای مجموعهٔ $a\in\mathbb{Z}_m$ که در رابطهٔ $a=۱\bmod n$ صادق باشند. این یک روش ساده و کارآمد برای محاسبهٔ نرم $\mathrm{N}_{\mathbb{L}/\mathbb{K}}(f)=\prod_a\mathrm{g}_a(f)$ به‌ویژه در FFT یا NTT را ارائه می‌دهد. به‌ویژه در حالت خاصی که $n=۲^\ell$، $\mathbb{L}=\mathbb{Q}[x]/\left(\Phi_{۲n}\right)$ و $\mathbb{K}=\mathbb{Q}[y]/\left(\Phi_n\right)$، بیان نرم میدان بسیار ساده است. هر $f\in\mathbb{L}$ را می‌توان به ضرایبی از درجه‌های زوج و فرد تفکیک کرد:

$$f=f_e\left(x^۲\right)+xf_o\left(x^۲\right) \tag{۹.۳}$$

که در آن $f_o,f_e\in\mathbb{K}$. ازآنجاکه $\psi_f:a\in\mathbb{L}\mapsto fa$، داریم

$$\mathrm{N}_{\mathbb{L}/\mathbb{K}}(f)=\det{}_{\mathbb{K}}\left(\psi_f\right)=\det\begin{bmatrix}f_e & f_o\\ yf_o & f_e\end{bmatrix}=f_e^۲-yf_o^۲ \tag{۱۰.۳}$$

۶.۳ تبدیل فوریهٔ سریع و تبدیل نظریهٔ اعدادی

تبدیل فوریهٔ سریع، و شکل دیگر آن یعنی تبدیل نظریهٔ اعدادی، ابزارهای قدرتمندی هستند که امکان محاسبات کارآمد را در حلقه‌های چندجمله‌ای فراهم می‌کنند. زمانی که عملوندها از نمایش FFT یا NTT استفاده می‌کنند، نرم میدان، به‌ویژه، می‌تواند بسیار ساده و سریع ارزیابی شود. بیشتر افزایش سرعت‌های به‌دست‌آمده توسط روش‌های ما، از تعامل بین نرم میدان و FFT/NTT حاصل می‌شود.

فرض کنید $\phi \in \mathbb{Q}[x]$ یک چندجمله‌ای مونیک از درجهٔ n، با n ریشهٔ متمایز $(\gamma_j)_{0 \le j < n}$ روی $\mathbb{C}$ باشد. برای $f \in \mathbb{C}[x]/(\phi)$، تبدیل فوریهٔ آن $\hat{f}$ به‌صورت زیر تعریف می‌شود:

$$\hat{f} = (f(\gamma_j))_{0 \le j < n} \tag{۱۱.۳}$$

تبدیل فوریه یک ایزومورفیسم بین $\mathbb{C}[x]/(\phi)$ و $\mathbb{C}^n$ است. بنابراین، برای $f, g \in \mathbb{C}[x]/(\phi)$، تبدیل فوریهٔ $f+g$ و fg را می‌توان به‌ترتیب با جمع و ضرب $\hat{f}$ و $\hat{g}$ محاسبه نمود.

تبدیل فوریهٔ سریع (یا FFT) یک الگوریتم شناخته‌شده برای محاسبهٔ تبدیل فوریهٔ f در حالت خاص $\phi = x^n + 1$ با $n = 2^\ell$ است [۱۵، ۸]. FFT دارای پیچیدگی زمانی $O(n \log n)$ عملیات در $\mathbb{C}$ است؛ تبدیل معکوس را نیز می‌توان با همین کارآمدی محاسبه کرد. به‌طور خاص، FFT امکان محاسبهٔ حاصل‌ضرب دو چندجمله‌ای به پیمانهٔ ϕ با پیچیدگی $O(n \log n)$ را فراهم می‌کند. FFT را می‌توان به پیمانه‌های دیگر، به‌ویژه چندجمله‌ای‌های سیکلوتومیک گسترش داد.

تبدیل نظریهٔ اعدادی (یا NTT)، آنالوگ تبدیل فوریه بر روی میدان متناهی $\mathbb{Z}_r$ برای یک عدد اول r است. تا زمانی که ϕ روی $\mathbb{Z}_r$ تقسیم شود، NTT خوش‌تعریف است؛ وقتی $\phi = x^n + 1$، کافی است داشته باشیم $r = 1 \bmod 2n$. مشابه حالت FFT، NTT را می‌توان در $O(n \log n)$ عملیات ابتدایی در $\mathbb{Z}_r$ برای برخی پیمانه‌ها، به‌ویژه چندجمله‌ای‌های سیکلوتومیک محاسبه کرد.

۷.۳ کاهش بابایی

قبل از اینکه نشان دهیم چگونه می‌توان معادلهٔ NTRU را حل کرد، آخرین ابزاری که نقش مهمی در این فرآیند بازی می‌کند را معرفی می‌کنیم: کاهش بابایی، یا بهتر است بگوییم تعمیمی از آن. این کاهش، یک جواب معادلهٔ NTRU را به جواب دیگری با چندجمله‌ای‌های کوتاه‌تر تبدیل می‌کند. ابتدا الحاق را تعریف می‌کنیم.

تعریف ۳.۳ (الحاق). فرض کنید $\phi \in \mathbb{Q}[x]$ مونیک با ریشه‌های متمایز (γ_j) روی $\mathbb{C}$ باشد. برای $f \in \mathbb{C}[x]/(\phi)$، ما الحاق آن f^* را به‌عنوان چندجمله‌ای منحصربه‌فردی در $\mathbb{C}[x]/(\phi)$ تعریف می‌کنیم که برای هر γ_j:

$$f^*(\gamma_j) = \overline{f(\gamma_j)} \tag{۱۲.۳}$$

که در آن $\overline{\cdot}$ نشان‌دهندهٔ مزدوج عدد مختلط است.

وجود و یکتایی به‌راحتی با توجه به این نکته به دست می‌آید که در نمایش FFT، محاسبهٔ الحاق، معادل جایگزینی هر ضریب فوریه با مزدوج آن است. اگر $f \in \mathbb{R}[x]/(\phi)$، آنگاه خواهیم داشت $f^* \in \mathbb{R}[x]/(\phi)$. در واقع، اگر γ ریشهٔ ϕ باشد، $\overline{\gamma}$ نیز ریشهٔ ϕ است، و $\overline{f(\gamma)} = f(\overline{\gamma})$. بنابراین، $f^*(\overline{\gamma}) = \overline{f^*(\gamma)}$ برای همهٔ ریشه‌های γ از ϕ. این خاصیت فقط با چندجمله‌ای‌های حقیقی به دست می‌آید، یعنی چندجمله‌ای‌هایی که ضرایب مختلط آن‌ها همهٔ اعداد حقیقی هستند. الحاق به ما امکان می‌دهد Reduce (الگوریتم ۱) را تعریف کنیم، که تعمیمی مستقیم از الگوریتم نزدیک‌ترین صفحه بابای [۲] بر روی $\mathbb{Z}[x]/(\phi)$-پیمانه‌ها است. برای ورودی‌های $f, g, F, G \in \mathbb{Z}[x]/(\phi)$، الگوریتم Reduce، F' و G' را با اندازه‌های نزدیک به حداقل محاسبه می‌کند به‌طوری‌که $fG - gF = fG' - gF'$. این نکته را ذکر می‌کنیم که گونه‌هایی از این الگوریتم قبلاً در کارهای قبلی ارائه شده‌اند، برای مثال [۱۸].

Algorithm 1 $Reduce_\phi(f, g, F, G)$

Require: $f, g, F, G \in \mathbb{Z}[x]/(\phi)$

Ensure: $F', G' \in \mathbb{Z}[x]/(\phi)$ such that $fG' - gF' = fG - gF \bmod \phi$

do

  $k \leftarrow \left\lfloor \frac{Ff^* + Gg^*}{ff^* + 9g^*} \right\rceil$

  $(F, G) \leftarrow (F - kf, G - kg)$

while $k \neq 0$

return F, G

ممکن است چند بار تکرار نیاز باشد، به‌خصوص اگر k با دقت کمی محاسبه شده باشد. در واقع، در عمل، ضرایب چندجمله‌ای‌های F, G می‌توانند قبل از کاهش، بسیار بزرگ باشند، و بنابراین محاسبهٔ k با دقت پایین (مثلاً با استفاده از مقادیر double در زبان برنامه‌نویسی C) نسبت به تقریب‌های ضرایب چندجمله‌ای کارآمدتر است: این امکانِ استفاده از نمایش FFT را فراهم می‌کند، جایی‌که ضرب‌های چندجمله‌ای و الحاق به‌راحتی محاسبه می‌شوند. سپس هر تکرار، یک مقدار تقریبی از k با ضرایب کوچک (با مقیاس‌بندی) را به دست می‌دهد. البته، استفاده از حساب اعشاری یعنی فرد ممکن است در یک حلقهٔ بی‌نهایت گیر بیافتد، اما این به‌راحتی با خروج از الگوریتم به‌محض توقف کاهش نرم (F, G) خنثی می‌شود.

به این نکته اشاره می‌کنیم که محاسبهٔ k، تقسیم چندجمله‌ای‌ها به پیمانهٔ ϕ را شامل می‌شود. در نمایش FFT، تقسیم به‌سادگی، عضوبه‌عضو اعمال می‌شود. ازآنجایی‌که ϕ روی $\mathbb{Q}[x]$ تحویل‌ناپذیر است، در اینجا هیچ تقسیم بر صفری رخ نمی‌دهد. بااین‌حال، در عمل، استفاده از مقادیر تقریبی با دقت پایین ممکن است (به‌ندرت) موقعیت‌هایی را به همراه داشته باشد که تقسیم بر صفر رخ دهد. همان‌طور که در بخش ۱.۵ توضیح داده خواهد شد، در تولید زوج کلید برای یک الگوریتم رمزنگاری، خطاهای گاه‌وبیگاه به‌راحتی قابل تحمل است. در این مقاله، ما از الگوریتم ۱ در چندجا استفاده می‌کنیم؛ هربار با چندجمله‌ای‌های f, g, F, G که در معادلهٔ NTRU (۲.۲) صدق می‌کنند. در این مورد، به‌طور غیررسمی، الگوریتم ۱ دو چندجمله‌ای F' و G' را به‌گونه‌ای محاسبه می‌کند که نرم (F', G') در حدود $O(\sqrt{n})$ بزرگ‌تر از (f, g) باشد.

۴ الگوریتم ارتقاءیافته برای حل معادلهٔ NTRU

این بخش روش و الگوریتم جدیدی را برای حل معادلهٔ NTRU ارائه می‌دهد (۲.۲). این الگوریتم از کاربرد بازگشتی نرم میدان در خود حل‌کنندهٔ NTRU کلاسیک ناشی می‌شود. ما ابتدا طرح کلی و شهود روش خود را در بخش ۱.۴ ارائه خواهیم کرد. در بخش ۲.۴، ما یک الگوریتم بازگشتی را بر اساس مشاهدات خود ارائه می‌دهیم، و در بخش ۳.۴، یک الگوریتم تکرارشوندهٔ کمی کندتر، اما از نظر حافظه کارآمدتر را معرفی می‌کنیم. در نهایت، در بخش ۴.۴، تحلیل‌هایی را برای زمان و حافظهٔ مورد نیاز الگوریتم ارائه خواهیم کرد.

۱.۴ طرح کلی

فرض کنید $m, p > 0$ اعداد صحیح باشند، $\mathbb{L}=\mathbb{Q}[x]/(\Phi_{pm})$, $\mathbb{K}=\mathbb{Q}[y]/(\Phi_m)$ و $\mathrm{N}=\mathrm{N}_{\mathbb{L}/\mathbb{K}}$. فرض کنید که یک عدد صحیح مفروض q و دو چندجمله‌ای $f, g \in \mathbb{Z}[x]/(\Phi_{pm})$ داشته باشیم، و می‌خواهیم $F, G \in \mathbb{Z}[x]/(\Phi_{pm})$ را پیدا کنیم به‌طوری‌که

$$fG - gF = q \tag{۱.۴}$$

از طرف دیگر، فرض کنید برای $\mathrm{N}(f), \mathrm{N}(g)$ که در حلقهٔ کوچک‌تر $\mathbb{Z}[y]/(\Phi_m)$ قرار دارند، می‌دانیم $F', G' \in \mathbb{Z}[y]/(\Phi_m)$ به‌گونه‌ای که:

$$\mathrm{N}(f)G' - \mathrm{N}(g)F' = q \tag{۲.۴}$$

ما ادعا می‌کنیم که می‌توانیم از راه‌حل‌های F', G' برای استنباط راه‌حل‌های F, G استفاده کنیم. در واقع، به یاد می‌آوریم که

$$N(f) = \prod_{g \in Gal(\mathbb{L}/\mathbb{K})} g(f) = ff^{\times}$$

که در آن $f^{\times} = \Pi_{\mathrm{g} \in \mathrm{Gal}(\mathbb{L}/\mathbb{K})} \mathrm{g}(f)$ نشان‌دهندهٔ حاصل‌ضرب تمام مزدوج‌های گالوای f به‌جز خودش بوده، و ما یک برابری مشابه برای g نیز داریم. سپس

$$ff^{\times}G'(x^p) - gg^{\times}F'(x^p) = q \tag{۳.۴}$$

که یک برابری در حلقهٔ بزرگ‌تر $\mathbb{Z}[x]/(\Phi_{pm})$ است. از این معادلهٔ آخر، نتیجه می‌شود که $F = g^{\times}F'(x^p)$ و $G = f^{\times}G'(x^P)$ راه‌حل‌هایی معتبر برای معادلهٔ NTRU هستند. از این مشاهدات، اکنون می‌توانیم طرح کلی الگوریتم‌های خود را برای حل معادلهٔ NTRU ارائه دهیم:

(i) از نرم میدان برای تصویر کردن آن به یک زیرحلقهٔ کوچک‌تر استفاده کنید،

(ii) معادله را در حلقهٔ کوچک‌تر حل کنید،

(iii) برای صعود جواب‌ها به حلقهٔ اصلی.

بااین‌حال، و برخلاف حمله NTRU بیش‌ازحد کشیده‌شده [۱]، ما مراحل تصویر کردن و صعود را تنها یک‌بار انجام نمی‌دهیم، بلکه به‌طور مکرر آن‌ها را تکرار خواهیم کرد. به بیان دقیق‌تر:

- f, g را روی یک زیرحلقهٔ کوچک‌تر تصویر می‌کنیم تا زمانی که به حلقهٔ اعداد صحیح $\mathbb{Z}$ برسیم؛ ما آن را مرحلهٔ نزول می‌نامیم.
- هنگامی که جواب‌ها را در $\mathbb{Z}$ به دست آوریم، با صعود آن‌ها به‌طور مکرر تا رسیدن به حلقهٔ اصلی ادامه می‌دهیم؛ ما این را مرحلهٔ صعود می‌نامیم.

تصویر کردن‌ها و صعود مکرر، کلید کارایی الگوریتم ما هستند: یک‌بار اجرای آن‌ها فقط از مرتبهٔ $O(1)$ بهبود ایجاد می‌کند، اما نشان خواهیم داد که ازلحاظ نظری، تکرار آن‌ها اجازه می‌دهد تا از مرتبه‌های بزرگ‌تر از $\tilde{O}(n)$ بهبود به دست آوریم، و در عمل این بهبود برای یک مقدار معمولی $n = ۱۰۲۴$، از مرتبهٔ ۱۰۰ خواهد بود. روال اجرای دو الگوریتم ما در شکل ۱ خلاصه شده است. فاز نزول در ستون میانی، و فاز صعود در ستون سمت راست نشان داده شده است.

$$
\begin{array}{ccccc}
\mathbb{Z}[x]/\left(x^n+1\right) & \ni & f, g & \rightarrow & F, G \\
\subsetneq & & \downarrow & & \uparrow \\
\mathbb{Z}[x]/\left(x^{n/2}+1\right) & \ni & \mathrm{N}(f), \mathrm{N}(g) & \rightarrow & F^{[1]}, G^{[1]} \\
\subsetneq & & \downarrow & & \uparrow \\
\mathbb{Z}[x]/\left(x^{n/4}+1\right) & \ni & \mathrm{N}^2(f), \mathrm{N}^2(g) & \rightarrow & F^{[2]}, G^{[2]} \\
\subsetneq & & \downarrow & & \uparrow \\
\vdots & \vdots & \vdots & & \vdots \\
\subsetneq & & \downarrow & & \uparrow \\
\mathbb{Z} & \ni & \mathrm{N}^{\ell}(f), \mathrm{N}^{\ell}(g) & \rightarrow & F^{[\ell]}, G^{[\ell]}
\end{array}
$$

شکل ۱. طرح کلی الگوریتمهای ۴ و ۵ برای حل (۲).

۲.۴ یک الگوریتم بازگشتی

در حالت خاص $\phi = x^n + ۱$ با $n = ۲^{\ell}$، می‌توانیم این فرمول‌ها را با $p = ۲$ اعمال کنیم، و سپس این کار را به‌طور مکرر روی $\phi' = x^{n/۲} + ۱$ تکرار نماییم. با این کار TowerSolverR (الگوریتم ۲) به دست می‌آید.

Algorithm 2 $\mathrm{TowerSolverR}_{n,q}(f, g)$

Require: $f, g \in \mathbb{Z}[x]/(x^n + 1)$ with n a power of two
Ensure: Polynomials F, G such that (2.2) is verified

if $n = 1$ **then**
  Compute $u, v \in \mathbb{Z}$ such that $uf - vg = GCD(f, g)$
  if $\delta = GCD(f, g)$ is not a divisor of q **then**
    abort
  $(F, G) \leftarrow (vq/\delta, uq/\delta)$
  return (F, G)
else
  $f' \leftarrow N(f)$ ▷ $f', g', F', G' \in \mathbb{Z}[x]/\left(x^{n/2} + 1\right)$
  $g' \leftarrow N(g)$
  $(F', G') \leftarrow TowerSolverR_{\frac{n}{2},q}(f', g')$
  $F \leftarrow g^{\times}(x) F'\left(x^2\right)$ ▷ $F, G \in \mathbb{Z}[x]/(x^n+1)$
  $G \leftarrow f^{\times}(x) G'\left(x^2\right)$
  $Reduce(f, g, F, G)$
 return (F, G)

توضیح غیررسمی در مورد اینکه چرا الگوریتم TowerSolverR از فضای بسیار کمتری نسبت به حل‌کنندهٔ کلاسیک (ResultantSolver) استفاده می‌کند این است که در هر مرحلهٔ بازگشت، اندازهٔ هر یک از ضرایب تقریباً دو برابر می‌شود، اما درجه نصف می‌شود، بنابراین تنها به تعداد نصف ضرایب قبل، ضریب برای ذخیره وجود خواهد داشت. این الگوریتم بر کاهش بابای (Reduce) تکیه دارد تا ضرایب محاسبه‌شدهٔ جدید (F,G) را به اندازه‌ای مشابه ضرایب (f,g) برای این سطح بازگشتی بازگرداند. یک تحلیل رسمی پیچیدگی فضا در لم ۱.۵ ارائه شده است.

صحت. اگر الگوریتم یک جواب را خروجی دهد (خاتمه در زیر نشان داده شده است)، درستی الگوریتم ۲ فوراً نتیجه می‌شود. در واقع، صحت در عمیق‌ترین سطح بازگشتی واضح است، و اگر الگوریتم برای $(f,g)\in\mathbb{Z}[x]/\left(x^{n/2}+1\right)$ صحیح باشد، به ما اطمینان می‌دهد که برای $(f,g)\in\mathbb{Z}[x]/\left(x^{n}+1\right)$ نیز صحیح خواهد بود.

۵ تحلیل پیچیدگی

اکنون به‌طور تفصیلی پیچیدگی TowerSolverR را مطالعه می‌کنیم. برای سادگی، در نظر می‌گیریم که $q=1$: معادلهٔ NTRU برای $q>1$ به‌راحتی، ابتدا با حل آن برای $q=1$ و مقیاس کردن خروجی (F,G) با عامل q حل می‌شود.

لم ۱.۵ (تحلیل پیچیدگی فضا). فرض کنید $q=1$ و نرم‌های اقلیدسی f,g دارای کران باشند: $\log\|f\|,\log\|g\|\leq B$ همچنین می‌دانیم که $\ell=\log n$. در نهایت فرض کنید:

$$\beta=\left(\frac{f^\star}{ff^\star+gg^*},\frac{g^\star}{ff^\star+gg^*}\right) \tag{۱.۵}$$

اگر $\beta=O(n\|(f,g)\|)$، آنگاه الگوریتم ۲ (*TowerSolverR*) در فضای $O(n\ell(B+\ell))$ اجرا می‌شود.

اثبات. واضح است که ما برج بازگشتی زیر را داریم:

$$\text{TowerSolverR}_{n,q}(f,g)\to\text{TowerSolverR}_{n/2,q}(\,\mathrm{N}(f),\mathrm{N}(g))\to\ldots\text{TowerSolverR}_{1,q}\left(\,\mathrm{N}^\ell(f),\mathrm{N}^\ell(g)\right)\cdots\to$$

اکنون فضای مورد نیاز متغیرهای داخلی را محدود کردیم.

۱. از (۱.۳)، هر $\mathrm{N}^i(g)$، $O(n(B+\ell))$ بیت می‌گیرد.

۲. اکنون نرم (اقلیدسی) (F,G) را محدود کردیم. ابتدا نرم آن را پس از کاهش در نظر می‌گیریم. با توجه به $V=\mathrm{Span}((f,g))$، بردار (F,G) می‌تواند به‌طور منحصربه‌فرد بر روی $V\oplus V^\perp$ تجزیه شود:

$$(F,G)=\left(\tilde{F},\tilde{G}\right)+\left(\breve{F},\breve{G}\right)$$

که در آن $(\tilde{F},\tilde{G})\in V^\perp$ و $(\breve{F},\breve{G})\in V$.

- در [۱۰]، لم ۳] نشان داده شده است که $\|(\tilde{F},\tilde{G})\|=\beta$. با فرض، مشخص می‌شود که $\|(\tilde{F},\tilde{G})\|=O(n\|(f,g)\|)$.
- ما $\|(\tilde{F},\tilde{F}G)\|$ را محدود کردیم: پس از کاهش (F,G) با استفاده از الگوریتم ۱، نابرابری مثلث تضمین می‌کند که

$$\|(\breve{F},\breve{G})\|\leq n/2\|(f,g)\|.$$

نتیجه می‌شود که

$$\|\,(F,G)\,\|^2=\|(\tilde{F},\tilde{G})\|^2+\|(\breve{F},\breve{G})\|^2=O\left(n^2\|\,(f,g)\,\|^2\right) \tag{۲.۵}$$

و بنابراین (F,G) را می‌توان در فضای $O(n(B+\ell))$ ذخیره کرد. البته، ما همچنین باید (F,G) را زمانی که از $F',G',f^\times,g^\times$ محاسبه می‌شود و هنوز کاهش نیافته است، کنترل کنیم. بنابراین خواهیم داشت:

$$\|F\|\leq\sqrt{\frac{n}{2}}\,\|F'\|\,\|g\|\quad\text{and}\quad\|G\|\leq\sqrt{\frac{n}{2}}\,\|G'\|\,\|f\|.$$

□

دربارهٔ لم۳. نسخهٔ قبلی این اثر نسخه‌ای از لم ۱.۵ را ارائه کرد که به شرط $\beta=\|(\tilde{F},\tilde{G})\|$ نیازی نداشت. بااین‌حال، اثبات، حاوی کران بالایی اشتباه برای β بود. این نسخهٔ به‌روزشده با اضافه کردن شرط $\beta=O(n\|(f,g)\|)$ این مورد را تصحیح می‌کند. این اثبات را تمام می‌کند و همچنین آن را بسیار ساده‌تر می‌کند. نقطهٔ ضعف آشکار این است که گزارهٔ اثبات‌شده کلیت کمتری دارد. بااین‌حال، این محدودیت را می‌توان به‌طور پیشگیرانه بررسی کرد که امکان نمونه‌گیری مجدد (f,g) را در زمانی که اجازه داریم، می‌دهد. یک استدلال ابتکاری در [۱۰] بیان می‌کند که وقتی ضرایب f، g بر اساس گاوسی نمونه‌برداری می‌شوند، می‌توانیم به‌طور متوسط انتظار داشته باشیم $(\beta=O(1/\|(f,g)\|)$. ما از توماس اسپیتاو برای اشاره به کران بالای اشتباه روی $\|(\tilde{F},\tilde{G})\|$ سپاسگزاریم.

لم ۲.۵ (تحلیل پیچیدگی زمانی). با شرایط لم ۱.۵، پیچیدگی‌های زمانی الگوریتم ۲ (*TowerSolverR*) عبارت‌اند از:

- $\tilde{O}(nB)$ برای الگوریتم ۲ با *Schönhage-Strassen*،
- $O\left((nB)^{\log_2(3)}\ell\right)$ برای الگوریتم ۲ با *Karatsuba*

توجه می‌کنیم که درحالی‌که پیچیدگی‌های داده‌شده با شونهاگ-اشتراسن بسیار بهتر از کاراتسوبا هستند، اما گمراه‌کننده هستند زیرا $\tilde{O}$ عوامل ثابت و لگاریتمی را پنهان می‌کند که در عمل قابل چشم‌پوشی نیستند. پیچیدگی‌های ارائه‌شده با *Karatsuba*، زمان‌های اجرایی را که ما برای مقادیر معمولی n و B مشاهده می‌کنیم، با دقت بیشتری منعکس می‌کند. در الگوریتم ۲، صعود پرهزینه‌ترین بخش به‌عنوان هر مرحلهٔ جداگانه است، کمی گران‌تر از فرود. سپس پیچیدگی زمانی آن $\sum_{0\le i<\ell} R_i$ است که به اثبات الگوریتم ۲ پایان می‌دهد.

حالت کلی برای q. تحلیل فوق شرایطی را پوشش می‌دهد که در آن سمت راست معادلهٔ NTRU، $q=1$ است. در حالت کلی، ممکن است مقدار دیگری از q را هدف قرار دهیم، معمولاً یک عدد صحیح کوچک. این کار با ضرب مقادیر در q در نقطه‌ای از فاز صعود انجام می‌شود. در توضیح الگوریتم TowerSolverR، این ضرب درست بعد از GCD انجام شد، اما بعدازآن نیز می‌توان آن را انجام داد. در هر صورت، ضرب در q اندازهٔ ضرایب چندجمله‌ای را به‌صورت $\log q$ افزایش می‌دهد (از نظر بیتی) و کاهش Babai در عمل این بیت‌ها را جذب می‌کند. در بدترین حالت، بیت‌های $\log q$ تا آخرین مرحله باقی می‌مانند، که به معنای فضای سربار حداکثر بیت‌های $O(n\log q)$ است.
احتمال شکست. در نسخهٔ قبلی این کار، ما به اشتباه بیان کردیم که الگوریتم ۲ راه‌حلی برای معادلهٔ NTRU (۲.۲) با ورودی‌های (f,g) پیدا می‌کند، اگر و فقط اگر چنین راه‌حلی وجود داشته باشد. این لزوماً درست نیست؛ الگوریتم‌های ۱ و ۲ راه‌حلی را پیدا می‌کنند اگر و فقط اگر $q|\gcd(\mathrm{N}(f),\mathrm{N}(g))$. بااین‌حال، (۲.۲) ممکن است چنین راه‌حلی را بپذیرد حتی اگر $q\nmid\gcd(\mathrm{N}(f),\mathrm{N}(g))$.
به‌عنوان مثال، در حلقهٔ $\mathbb{Z}[x]/\left(x^4+1\right)$، عناصر حلقه را در نظر بگیرید:

$$(f,g,F,G)=\left(x-2,9x^3+x^2,x,x^2+2x+4\right).$$

می‌توان بررسی کرد که $fG-gF=1$، درحالی‌که $\mathrm{N}(f)=17$ و $\mathrm{N}(g)=17386$.
کیفیت خروجی. یک مفهوم مهم کیفیت جواب‌ها (F,G) است، برای مثال در نرم اقلیدسی یا در نرم گرام اشمیت (همان‌طور که در [۱۰، ۱۶] تعریف شده است). برای هر یک از این معیارها، الگوریتم‌های ما جواب‌هایی با کیفیتی مشابه با الگوریتم‌های موجود خروجی می‌دهند.
در واقع، مجموعهٔ جواب‌ها به شکل $\{(F_0+rf,G_0+rg)\mid r\in\mathbb{Z}[x]/(x^n+1)\}$ است، که در آن (F_0,G_0) یک جفت جواب دلخواه را نشان می‌دهد. برای هر عنصر در این مجموعه، الگوریتم ۱ همان جواب را خروجی می‌دهد، بنابراین نرم اقلیدسی خروجی برای الگوریتم‌های ۱ و ۲ یکسان خواهد بود.
از سوی دیگر، برای یک ورودی ثابت (f,g)، همهٔ جواب‌های معادلهٔ NTRU دارای نرم گرام اشمیت یکسان هستند (به‌عنوان مثال [۱۰]، لم ۳] را ببینید).

۶ نتیجه‌گیری و مسائل باز

ما استفاده از نرم میدان را برای بهینه‌سازی برخی از محاسبات روی حلقه‌های چندجمله‌ای، به‌ویژه نتیجه‌ها و حل معادلهٔ NTRU ارائه کردیم. نتیجه عملی دومی این است که الگوریتم امضای پساکوانتومی فالکون به‌طور کامل بر روی میکروکنترلرهای کوچک یا حتی کارت‌های هوشمند قابل استفاده است، زیرا ۳۲ کیلوبایت RAM برای اجرای الگوریتم ما حتی برای یک شبکهٔ NTRU با امنیتی طولانی‌مدت (درجه $n=1024$)، کافی است. تمام عملیات مربوط به امضاها (تولید امضا، تأیید، و تولید زوج کلید) می‌توانند بر روی چنین سخت‌افزارهای

محدودی قرار گیرند.

در زیر تعدادی از سؤالات باز را فهرست می‌کنیم.

چندجمله‌ای‌های غیر سیکلوتومیک. در توصیف خود، مورد چندجمله‌ای سیکلوتومیک را به‌عنوان مدول پوشش دادیم. این روش را می‌توان به مدول‌های دیگر گسترش داد. در واقع، برای هر مدول $\phi=\phi' \left(x^d\right)$ برای مقدار $d> ۱$، استفاده از "نرم میدان" می‌تواند درجه را بر d برای اهداف محاسبهٔ برآیندها و حل معادلهٔ NTRU تقسیم کند. حتی اگر ϕ در $\mathbb{Q}[x]$ تقلیل‌ناپذیر نباشد، یعنی اگر $\mathbb{Q}[x]/(\phi)$ در واقع یک فیلد نباشد، صادق است. شرح حالت کلی همچنان به‌عنوان مسئله برای بررسی باقی مانده است. بااین‌حال، استفاده از مدول‌های کاهش‌پذیر در شبکه‌های NTRU معمولاً توصیه نمی‌شود.

اعداد صحیح بزرگ. درحالی‌که دستاوردهای ما، از نظر حافظه، قابل توجه است، ما هنوز باید اعداد صحیح بزرگ را مدیریت کنیم. از نقطه‌نظر پیچیدگی پیاده‌سازی، خلاص شدن از شر اعداد صحیح، برای مثال با انجام تمام عملیات در RNS، بدون تأثیر منفی بر زمان اجرا و نیازهای حافظهٔ الگوریتم‌های ما، جالب خواهد بود.

کاربردهای دیگر برای ساختارهای رمزنگاری. به نظر می‌رسد که بررسی اینکه آیا روش ذکرشده در این مقاله می‌تواند کارایی سایر الگوریتم‌های رمزنگاری را بهبود بخشند، ارزشمند باشد. علاوه‌بر این، درست همان‌طور که در این مقاله یک کاربرد سازنده از نرم میدان (در مقابل [۱]) ارائه کردیم، به نظر می‌رسد که، کاربرد سازندهٔ ردیابی (در مقابل [۶]) بسیار جالب خواهد بود. در پایان، [۲۰] نشان داد که دیدگاه جبری در مورد [۱] ضروری نیست. این سؤال را مطرح می‌کند که آیا در مورد این کار ارائه‌شده در این مقاله همچنین است؟

کاربردهای مختص تحلیل. به نظر می‌رسد که استفاده از روش ارائه‌شده در این مقاله برای بهبود حملات بر اساس نرم میدان [۱] یا حتی بر روی ردیابی میدان [۶] ارزشمند باشد.

References

[1] Albrecht, M., Bai, S., & Ducas, L. (2016). A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and graded encoding schemes. *In Annual International Cryptology Conference, Berlin, Heidelberg: Springer Berlin Heidelberg*, 9814, 153–178. DOI: https://doi.org/10.1007/978-3-662-53018-4_6.

[2] Babai, L. (1986). On Lovász'lattice reduction and the nearest lattice point problem. *Combinatorica*, 6, 1–13. DOI: https://doi.org/10.1007/BF02579403.

[3] Bernstein, D.J., Chuengsatiansup, C., Lange, T., & van Vredendaal, C. (2016). NTRU Prime. *Tech. rep., National Institute of Standards and Technology*.

[4] Campbell, P., & Groves, M. (2017). Practical post-quantum hierarchical identity-based encryption. *16th IMA International Conference on Cryptography and Coding*.

[5] Cash, D., Hofheinz, D., Kiltz, E., & Peikert, C. (2010). Bonsai trees, or how to delegate a lattice basis. *In: Gilbert, H. (eds) Advances in Cryptology – EUROCRYPT 2010. EUROCRYPT 2010. Lecture Notes in Computer Science, vol 6110. Springer, Berlin, Heidelberg*. DOI: https://doi.org/10.1007/978-3-642-13190-5_27.

[6] Cheon, J.H., Jeong, J., & Lee, C. (2016). An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low level encoding of zero. *LMS Journal of Computation and Mathematics*, 19, 255–266. DOI: https://doi.org/10.1112/S1461157016000371.

[7] Chuengsatiansup, C., Prest, T., Stehlé, D., Wallet, A., & Xagawa, K. (2020). ModFalcon: Compact signatures based on module-NTRU lattices. *In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 853–866. DOI: https://doi.org/10.1145/3320269.3384758.

[8] Cooley, J.W., & Tukey, J.W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19, 297–301.

[9] Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE transactions on Information Theory*, 22, 644–654. DOI: https://doi.org/https://doi.org/10.1109/TIT.1976.1055638.

[10] Ducas, L., Lyubashevsky, V., & Prest, T. (2014). Efficient identity-based encryption over NTRU lattices. *In Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, ROC, December 7-11, 2014, Proceedings, Part II, 20, 22–41*. DOI: https://doi.org/10.1007/978-3-662-45608-8_2.

[11] Espitau, T., Fouque, P.A., Gérard, F., Rossi, M., Takahashi, A., Tibouchi, M., Wallet, A., & Yu, Y. (2022). Mitaka: A simpler, parallelizable, maskable variant of Falcon. *In Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cham: Springer International Publishing*, 222–253. DOI: https://doi.org/10.1007/978-3-031-07082-2_9.

[12] FIPS. (2001). NIST: Security Requirements for Cryptographic Modules.

[13] Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., & Zhang, Z. (2017). Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU (tech. rep.).

[14] Fouque, P.A., Kirchner, P., Pornin, T., & Yu, Y. (2022). Bat: Small and Fast KEM over NTRU Lattices. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 240–265. DOI: https://doi.org/10.46586/tches.v2022.i2.240-265.

[15] Gentleman, W.M., & Sande, G. (1966). Fast Fourier transforms: for fun and profit. *In Proceedings of the November 7-10, fall joint computer conference*, 563–578.

[16] Gentry, C., Peikert, C., & Vaikuntanathan, V. (2008). Trapdoors for hard lattices and new cryptographic constructions. *In Proceedings of the fortieth annual ACM symposium on Theory of computing*, 197–206. DOI: https://doi.org/10.1145/1374376.1374407.

[17] Harvey, D., & Van Der Hoeven, J. (2021). Integer multiplication in time O(nlog n). *Annals of Mathematics*, 193, 563–617. DOI: https://doi.org/10.4007/annals.2021.193.2.4.

[18] Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J.H., & Whyte, W. (2003). NTRUSIGN: Digital signatures using the NTRU lattice. *In Cryptographers' track at the RSA conference, Berlin, Heidelberg: Springer Berlin Heidelberg*, 122–140. DOI: https://doi.org/https://doi.org/10.1007/3-540-36563-X_9.

[19] Hoffstein, J., Pipher, J., & Silverman, J.H. (1998). NTRU: A ring-based public key cryptosystem. *In International algorithmic number theory symposium, Berlin, Heidelberg: Springer Berlin Heidelberg*, 267–288. DOI: https://doi.org/https://doi.org/10.1007/BFb0054868.

[20] Kirchner, P., & Fouque, P.A. (2017). Revisiting lattice attacks on overstretched NTRU parameters. *In Annual International Conference on the Theory and Applications of Cryptographic*

Techniques, Cham: Springer International Publishing, 3–26. DOI: https://doi.org/10.1007/978-3-319-56620-7_1.

[21] Lenstra, A.K., Lenstra, H.W., & Lovász, L. (1982). Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(ARTICLE), 515–534. DOI: https://doi.org/10.1007/BF01457454.

[22] Micciancio, D., & Warinschi, B. (2001). A Linear Space Algorithm for Computing the Hermite Normal Form. *In Proceedings of the 2001 international symposium on Symbolic and algebraic computation*, 231–236. DOI: https://doi.org/10.1145/384101.384133.

[23] NIST. (2016). Submission requirements and evaluation criteria for the post-quantum cryptography standardization process.

[24] Schönhage, A., & Strassen, V. (1971). Fast multiplication of large numbers. *Computing*, 7, 281–292. DOI: https://doi.org/10.1007/BF02242355.

[25] Smart, N.P., Albrecht, M.R., Lindell, Y., Orsini, E., Osheter, V., Paterson, K., & Peer, G. (2017). Lima: A PQC encryption scheme. *National Institute of Standards and Technology*.

[26] Stehle, D. & Steinfeld, R. (2013). Making NTRUEnrypt and NTRUSign as secure as standard worst-case problems over ideal lattices. *Cryptology ePrint Archive, Report 2013/004*.

[27] Von Zur Gathen, J., & Gerhard, J. (2013). Modern Computer Algebra (3. Ed.). *Cambridge University Press*.

[28] Working Group of the C/MM Committee. (2009). *IEEE P1363. 1 Standard Specifications for Public-Key Cryptographic Techniques Based on Hard Problems over Lattices*.

[29] Zhao, R.K., McCarthy, S., Steinfeld, R., Sakzad, A., & O'Neill, M. (2023). Quantum-safe HIBE: does it cost a Latte?. *IEEE Transactions on Information Forensics and Security*. DOI: https://eprint.iacr.org/2021/222.