Trends, Challenges and Needs for Lattice-Based Cryptography Implementations

Hamid Nejatollahi
University of California Irvine
Irvine, California 92697-3435
hnejatol@uci.edu

Nikil Dutt
University of California Irvine
Irvine, California 92697-3435
dutt@ics.uci.edu

Rosario Cammarota
Qualcomm Research
San Diego, CA 92121-1714
ro.c@qti.qualcomm.com

ABSTRACT

Advances in computing steadily erode computer security at its foundation, calling for fundamental innovations to strengthen the weakening cryptographic primitives and security protocols. At the same time, the emergence of new computing paradigms, such as Cloud Computing and Internet of Everything, demand that innovations in security extend beyond their foundational aspects, to the actual design and deployment of these primitives and protocols while satisfying emerging design constraints such as latency, compactness, energy efficiency, and agility. While many alternatives have been proposed for symmetric key cryptography and related protocols (e.g., lightweight ciphers and authenticated encryption), the alternatives for public key cryptography are limited to post-quantum cryptography primitives and their protocols. In particular, lattice-based cryptography is a promising candidate, both in terms of foundational properties, as well as its application to traditional security problems such as key exchange, digital signature, and encryption/decryption. We summarize trends in lattice-based cryptographic schemes, some fundamental recent proposals for the use of lattices in computer security, challenges for their implementation in software and hardware, and emerging needs.

KEYWORDS

Post-quantum cryptography; Public key cryptography; Lattice based cryptography; Ideal lattices; Ring-LWE.

1 INTRODUCTION

Advances in computing steadily erode computer security at its foundation, enabling prospective attackers to use ever more powerful computing systems and algorithms, and in turn making brute-force attacks against cryptography progressively more practical. On one hand there is Moore's Law, which seems to continue inexorably, [1] making traditional computing systems more capable than ever before. On the other hand there is the rise of alternative computing paradigms, such as quantum computing and its algorithms [14, 23] - an approaching reality,[2] which promise to further weaken the strength of current, standardized cryptography and its applications. As a result, the need to strengthen current practices in computer security, including strengthening and adding variety in cryptography, has become a widely accepted fact.

While there exist many alternatives for standardized symmetric key cryptography and related protocols (e.g., lightweight ciphers such as `PRINCE`[7] and `QARMA`[3]; and authenticated encryption), the alternatives for public key cryptography are limited to post-quantum cryptography primitives and their related protocols for key exchange, digital signature, encryption/decryption and homomorphic schemes [11].

Among the post-quantum cryptography candidates, lattice-based cryptography (LBC) appears to be gaining acceptance. Its applications are proliferating for both traditional security problems (e.g., key exchange and digital signature), as well as emerging security problems (e.g., homomorphic schemes). Lattice-based cryptographic algorithms and protocols promise to tackle the challenges posed by deployment across diverse computing platforms, e.g., Cloud vs. Internet-of-Things (IoT) ecosystem, as well as for diverse use cases, including the ability to perform computation on encrypted data, providing strong foundations for protocols based on asymmetric key cryptography against powerful attackers (using Quantum computers and algorithms), and to offer protection beyond the span of traditional cryptography. Indeed, lattice-based cryptography promises to enhance security for long-lived systems, e.g., critical infrastructures, as well as for safety-critical devices such as smart medical implants [9]. In this paper, we summarize trends in lattice-based cryptographic schemes, some fundamental recent proposals for the use of lattices in computer security, challenges for their implementation in software and hardware, and emerging needs.

2 TRENDS

At the computing platform level, we are seeing a diversity of computing capability, ranging from high-performance (real-time) virtualized environments, such as cloud computing resources and software defined networks, to highly resource-constrained IoT platforms to realize the vision of Internet of Everything. This poses tremendous challenges in the design and implementation of emerging cryptographic schemes, since the computing platforms exact diverging goals and constraints.

The emergence of new computing paradigms, e.g., Quantum computing, threatens to weaken even the strongest contemporary cryptographic schemes. The path toward strengthening current cryptographic schemes by increasing the key length and selecting domain parameters appropriately is neither a viable nor a practical solution to the problem [8]. While doubling the key size for symmetric key cryptography is an interim solution for non resource-constrained devices,[3] Quantum computers poses a deadly threat to the effectiveness of traditional asymmetric cryptographic algorithms.

The cloud computing and the software defined network space demand agility, high performance, and energy efficiency of cryptography, which calls for the development of new programmable

[1] http://www.telegraph.co.uk/technology/2017/01/05/ces-2017-moores-law-not-dead-says-intel-boss

[2] http://spectrum.ieee.org/tech-talk/computing/software/rigetti-launches-fullstack-quantum-computing-service-and-quantum-ic-fab

[3] See NIST and NSA recommendations on Suite B Cryptography: https://www.nsa.gov/what-we-do/information-assurance/

CODES+ISSS'17,
2017. 978-x-xxxx-xxxx-x/YY/MM…$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

accelerators capable of running not only individual cryptographic algorithms, but full protocols efficiently, with the resulting challenge of designing for agility, e.g., designing computing engines that achieve the efficiency of Application-Specific Integrated Circuits (ASICs), while retaining some level of programmability.

In the IoT space, implementations of standardized cryptography to handle increased key sizes become too expensive in terms cost, speed, and energy, but necessary, e.g., in the case of long lived systems such as medical implants. This demands the development of new and strong lightweight alternatives for both symmetric and asymmetric primitives. Furthermore, given the variety of applications and interplay with cloud, agility becomes another key requirement.

The examples above, to name a few, form a compelling argument to call for innovation in the computer security space, including and beyond the foundations, i.e., down to the actual implementation and deployment of primitives and protocols to satisfy the emerging business models and their design constraints - latency, compactness, energy efficiency, tamper resistance and, more importantly, agility.

3 BACKGROUND

A lattice is defined as a countable set of points in a n-dimensional Euclidean space with a periodic structure [20, 22]. Let b_1, b_2,..., b_n $\in \mathbb{R}^m$ be a set of linearly independent vectors; and let us define B as the $m \times n$ matrix in which i^{th} column is b_i vector such that $B = [b_1, b_2, ..., b_n]$. Thus, a lattice $\mathcal{L}$ is the set of all integer combinations that is generated by the basis B (with integer or rational entries), that is:

$$\mathcal{L}(B) = \{Bx : x \in \mathbb{Z}^n\} = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z}^n, 1 \leq i \leq n \right\}$$

Given basis B in a lattice, the shortest nonzero vector problem (SVP) in $\mathcal{L}(B)$ is defined as the problem of finding vector $v \in \mathcal{L}(B) \backslash \{0\}\}$ such that $\|v\| = \lambda_1(\mathcal{L}(B))$ [4]. The output of the SVP problem is the shortest non zero vector. The Closest Vector Problem (CVP) is the inhomogeneous version of the SVP in lattices.

One of the results by Gauss proves that there is a reduced basis for every lattice by which solving CVP is not complex. Otherwise, CVP is hard to solve with any arbitrary basis of the lattice[10]. In applications of lattices to public key cryptography, the private key is a reduced basis of the lattice, whereas a public key is another non-reduced basis. For encryption, the plain text which is a lattice point ($v \in \mathcal{L}$) is perturbed with small error e in a way that the result is not a lattice point ($w = v + e \notin \mathcal{L}$). Finding the original text v, decryption, is done by solving CVP using the reduced basis, i.e., the private key. In digital signature, the message (w) is a point that does not belong to the lattice ($w \notin \mathcal{L}$). Signer uses the reduced basis to compute the closest point in the lattice (v) to the message (w) and sends that lattice point as the signature. Verifier checks if the signature (v) is a lattice point which can be done by any basis, i.e., with the public key.

It is conjectured there is no polynomial time algorithm that can approximate lattice problems used in computer security, such as SVP, CVP [16]. The mentioned conjecture is the basis for security of lattice-based cryptography schemes. The seminal work of Ajtai [1] provides confidence for the adoption of lattice based schemes in cryptography. He proves that solving some NP-hard lattice problems, e.g., SVP, in average-case is as hard as solving them in the worst case assumption. There is a close relationship between lattice hard problems (like CVP and SVP) and two common average-case lattice-based problems: Learning With Error (LWE); the Shortest Integer Solution (SIS); and their variant on rings Ring-LWE and Ring-SIS. For LWE, $As + e = b$ where A is a $n \times n$ matrix and s and e are the secret and a small error respectively, the goal is to recover s from (A, b). Equivalently, the lattice point As is perturbed by a small error vector e and the result is $b = As + e \notin \mathcal{L}$. Recovering e and s is equivalent to solving an average case CVP.

Large key size and inefficient matrix-vector/matrix-matrix arithmetic of standard lattices are motivating reasons to adopt a more efficient alternative in cryptography, i.e., ideal lattices. An Ideal lattice is defined over a ring $R = \mathbb{Z}_q[x]/(f(x))$, where R contains all the polynomials with modulo q integer coefficients. In an n-dimensional lattice, a standard lattice is represented by n vectors, while for ideal lattices 1 vector (polynomial) suffices and $n - 1$ remaining vectors are simply built by applying simple operations (shift and negate) of that single vector. The algebraic structure of ideal lattices not only allows for fast arithmetic, e.g., by employing Fast Fourier Transforms, but also allows a reduction in the memory footprint by factor of n. For LWE, (assume $As + e = b$) in order to generate 1 extra pseudo-random number, i.e., one element of b, n random numbers (one row of A) should participate in dot product with the secret s. In Ring-LWE, n pseudo-random numbers can be produced by performing polynomial multiplication of same number of random numbers (one row of A) with secret s.

4 IMPLEMENTATION CHALLENGES

The community strives to achieve efficient and secure implementations of lattice-based schemes, to map such schemes on a range of applications from resource constrained devices in the embedded system world to resourceful platforms in the server and cloud world. In terms of the mechanics of the computation, a lattice-based implementation involves modulo arithmetic on big numbers and the extraction of the random term. Optimizations to modulo arithmetic computation can be induced by the algebraic structure of the ideal lattices which provides fast arithmetic by performing arithmetic computations on numbers in Number-Theoretic Transform (NTT) format which is highly efficient, with a time complexity of $O(n\log n)$ [17]. Standard lattice-based schemes are more amenable to arithmetic computations performed as matrix-to-matrix/vector, with time complexity of $O(n^2)$ for multiplication, which is much more expensive in terms of memory and runtime. Another component of a lattice-based scheme is extraction of the random term, which is usually implemented with a discrete noise sampler (herein after "sampler"). The drawing of the random term from a discrete distribution, e.g., the Gaussian distribution, can be implemented via rejection [25], inversion [19], Ziggurat [5], or Knuth-Yao sampling [13]. Distributions with moderate standard deviation are used for key exchange and public key encryption, and small standard deviation for digital signature to achieve compact and secure signatures.

Implementations in software are customized to map on wide vector extensions, which are ubiquitous in general purpose and many embedded processors, e.g., Intel and ARM-based processors. Practical software implementations of standard lattices for encryption [12] and key exchange [4] have been published recently.

Sampling from a perfect error distribution is impractical (in terms of memory and computation) in either hardware or software due to the infinite long tails. Consequently, an approximation of

[4] Where $\| . \|$ is the Euclidean norm in $\mathbb{R}^n$, and $\lambda_1(\mathcal{L})$ is the first successive minimum.

the perfect distribution with finite tails and negligible statistical difference to the perfect distribution is employed in sampling. Larger distribution tails culminate in more security assurance with the cost of higher memory footprint. The design choices of these lattice-based schemes is customized for their use in computer security. The design of key exchange mechanisms, such as Newhope [2] can tolerate a less accurate sampler to draw the random terms, instead of traditional high precision Gaussian sampler with some guarantee of security. Newhope uses a bi-modal sampler which offers speed and an appropriate level of security for the implementation. Indeed, Alkim et al. [2] used a less precise binomial sampler inside their key-exchange protocol which is easier to implement in either hardware or software and also protect against timing attacks.

For digital signature schemes the precision of the sampler cannot not be traded for efficiency. Digital signature consists of three steps including key generation (secret key (sk) for signer and public key (pk) for verifier), Sign_{sk}, and Verify_{pk}. Signer applies the encryption algorithm on input message M as $S = \text{Sign}_{sk}(M, sk_{signer})$ and sends (M,S) to the verifier who applies Verify_{pk} (M,S) to check validity of the signature. The Bimodal Lattice Signature Scheme (BLISS) [6], whose security is based on Ring-SIS, has been gaining attention because its efficiency compares to RSA and Elliptic Curve Digital Signature (ECDSA). For security level above 128-bits, software implementation of BLISS [6] achieves an order of magnitude better runtime compared to RSA with almost the same signature size (5kb) with 0.5× secret key (2kb) and 1.75× larger public key size (7kb). For the same security level, hardware implementations of BLISS is about an order of magnitude faster than RSA (Sign) and ECDSA (Sign and Verify) implementations with lower resource usage [21].

To address limitations in resource constrained devices, it is common practice to trade off memory footprint with security assurance, which improves both efficiency and memory consumption [12]. However, this limits the applicability of the implementation to different scenarios.

Another trade-off of security for resources can be done at a higher level in choosing ideal lattices over standard lattice. When implemented, standard lattice-based cryptography (LBC), e.g., LWE-based, schemes exhibit a relatively large memory footprint due to large key size (hundreds of kilobyte for public key), which makes implementations of standard LWE-based schemes impractical on constrained devices. The adoption of specific ring structures, e.g., Ring-LWE, offers key size reduction by a factor of n compared to Standard LWE. [15], making Ring-LWE an excellent candidate for resource constrained devices, such as Wi-Fi capable devices, including medical implants. For example, a Ring-LWE key exchange parameter set of $(n = 512, q = 25601)$ results in security level of 128-bit with public key size of 7680 bits, while $(n = 1024, q = 14336)$ achieves 256-bit security with public key size of 16384 bits [24]. However, a smaller modulus q results in faster, more secure (due to increase of error term impact) and lower memory and bandwidth overhead with the cost of higher failure probability.

Implementations in hardware are customarily emulated on FPGA platforms, which provide flexibility and customization, but not the agility of programmability to match the expected highly variable demand for computer security in virtualized environments. Programmability is another important design choice which can be provided at different layers, e.g., to program for different sampler complexity. Implementation of lattice-based schemes as application specific integrated circuits (ASICs) offer the best energy and performance at the expense of agility [18].

5 NEEDS

Lattice-based cryptographic algorithms and protocols promise to tackle the challenges posed by deployment across diverse computing platforms, as well as for diverse use cases within reasonable security, performance and energy efficiency guarantees.

Numerous schemes and implementations tackle different trade-offs, such as memory footprint, security, performance, and energy, are mapped on a variety of platforms and are applicable to specific use cases. However, current designs are still deficient in addressing the need for agility, which is paramount to tackle the needs of emerging business models at the computing platform level. In addition, securing such platforms against physical attacks is a topic that needs to be researched.

6 ACKNOWLEDGEMENT

This work was supported in part with a gift from Qualcomm Research.

REFERENCES

[1] Miklós Ajtai. 1996. Generating hard instances of lattice problems. In *Proc. ACM symposium on Theory of computing*. ACM, 99–108.

[2] Erdem Alkim et al. 2015. Post-quantum key exchange-a new hope. *IACR Cryptology ePrint Archive* (2015).

[3] Roberto Avanzi. 2017. The QARMA Block Cipher Family. Almost MDS Matrices Over Rings With Zero Divisors, Nearly Symmetric Even-Mansour Constructions With Non-Involutory Central Rounds, and Search Heuristics for Low-Latency S-Boxes. *IACR Transactions on Symmetric Cryptology* (2017).

[4] Joppe Bos et al. 2016. Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In *Proc. ACM SIGSAC*.

[5] Johannes Buchmann et al. 2013. Discrete Ziggurat: A time-memory trade-off for sampling from a Gaussian distribution over the integers. In *Proc. SAC*.

[6] Léo Ducas et al. 2013. Lattice signatures and bimodal gaussians. In *CRYPTO*.

[7] Borghoff Julia et al. 2012. *PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications.*

[8] Daniel J. Bernstein et al. 2017. Post-quantum RSA. Cryptology ePrint Archive, Report 2017/351. (2017).

[9] Garcia-Morchon et al. 2015. *DTLS-HIMMO: Achieving DTLS Certificate Security with Symmetric Key Overhead.*

[10] CF Gauss. 1801. Disquisitiones Arithmeticae. (1801).

[11] Craig Gentry et al. 2008. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. ACM STC*.

[12] James Howe et al. 2016. Lattice-based Encryption Over Standard Lattices in Hardware. *Proc. DAC* (2016).

[13] Donald E Knuth et al. 1976. The complexity of nonuniform random number generation. *Algorithms and complexity* (1976).

[14] Gui-Lu Long. 2001. Grover algorithm with zero theoretical failure rate. *Physical Review A* (2001).

[15] Vadim Lyubashevsky et al. 2010. On ideal lattices and learning with errors over rings. In *Proc. EUROCRYPT*.

[16] Daniele Micciancio et al. 2009. Lattice-based cryptography. In *Proc. PQC*.

[17] Henri Nussbaumer. 1980. Fast polynomial transform algorithms for digital convolution. *IEEE T ACOUST SPEECH* (1980).

[18] Tobias Oder et al. 2016. Lattice-based cryptography: From reconfigurable hardware to ASIC. In *Proc. ISIC*.

[19] Chris Peikert. 2010. An efficient and parallel Gaussian sampler for lattices. In *Annual Cryptology Conference*. Springer, 80–97.

[20] Chris Peikert. 2015. A Decade of Lattice Cryptography. Cryptology ePrint Archive, Report 2015/939. (2015).

[21] Thomas Pöppelmann et al. 2014. Enhanced lattice-based signatures on reconfigurable hardware. In *Proc. CHES*.

[22] Oded Regev. 2009. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM* (2009).

[23] Peter W. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* (1997).

[24] Vikram Singh. 2015. A Practical Key Exchange for the Internet using Lattice Cryptography. (2015).

[25] John Von Neumann. 1951. Various Techniques Used in Connection With Random Digits. (1951).