Lattice-based Cryptography

Ayman Wagih Mohsen
Department of Computer and Systems Engineering
Ain Shams University
ayman.wagih@eng.asu.edu.eg

Ayman M. Bahaa-Eldin
Misr International University
On leave from Ain Shams University
ayman.bahaa@eng.asu.edu.eg

Mohamed Ali Sobh
Department of Computer and Systems Engineering
Ain Shams University
Mohamed.sobh@eng.asu.edu.eg

***Abstract*—Among the different cryptosystems, the Public Key Cryptosystems (PKCS) are of great usage and are used in many applications. Most of the current PKCS are based on a hard problem making the cryptanalysis of such cipher infeasible given the current computing and memory availability. However, Quantum Computing threatens all the current PKCS as they offer a new model that can solve those hard problems, e.g. the factorization problem of RSA or the discrete logarithm problem of ELGAMAL algorithm. Lattice Based PKCS are a promising filed to introduce an immune cipher system against quantum cryptanalysis. This paper presents a summary of recent progress in the field of lattice-based cryptography, by focusing on the recent advancement in lattice-based public key cryptosystems and key exchange mechanisms based on the learning with errors (LWE) problem and its ring variant Ring-LWE.**

***Keywords*—Post-quantum cryptography, learning with errors, public key cryptoystem, key exchange.**

I. Introduction

With the nearing of capable quantum computers being built, current security protocols should better switch their cryptosystems, key encapsulation mechanisms (KEM), digital signatures and other security schemes to ones that are immune against quantum-based attacks. The quantum computer-based attacks were first envisioned when in 1996, P. Shor derived an algorithm that uses a quantum computer to solve the hard problem of finding the discrete logarithm of a large number with polynomial-time complexity in the number of digits. Most public key cryptosystems and the key encapsulation mechanisms (such as RSA, PGP, EC-DH) have the security guarantee based on the hardness of factoring large numbers or finding the discrete logarithm. This means that when a capable quantum computer is built, it can be used to easily crack these security schemes.

There are several fields of asymmetric cryptosystems with no quantum-based attacks found to date, such as: multivariate cryptosystems, code-based cryptosystems, hash-based cryptosystems and lattice-based cryptosystems. Most of them have various drawbacks [1], for example: many multivariate systems were made, and many were later found to be not secure. Code-based cryptosystems have keys of large sizes. In hash-based cryptosystems the private key can only be used once. As for lattice-based cryptosystems, they are still not as efficient as the currently used cryptosystems, involving significant ciphertext expansion (blowup factor) at around 25, but they can have proven security and other advantages, and have been a subject of study in the past decade.

There are several attempts to build different security systems based on quantum computing like secure direct communication [21], and specific application based security like database security [22] and [23].

A. Preliminaries

Vectors are denoted in bold, matrices are denoted in bold upper case. The set of integers modulo q is denoted by $Zq = \{0,...,q-1\}$. When an integer belongs to the set Zq it is mentioned that it is in Zq. The set of all polynomials with integer coefficients is denoted by Z[x]. The ring of polynomials modulo some polynomial f(x) is denoted by $R = Z[x]/f(x)$. When all the coefficients of the polynomials are in Zq, the polynomial ring is written as $Rq = Zq[x]/f(x)$.

B. Lattices

A lattice is a periodic set of points in n-dimensions formed by all vectors that are linearly dependent on a set of linearly independent vectors b[i] called the basis. The basis vectors form the columns of the basis matrix B that describes the lattice L(B). When an n-dimensional lattice has n basis vector it's a full rank lattice. The same lattice can be described with several different basis matrices. The fundamental region of the lattice is an n-dimensional convex shape that fills the entire space with no overlapping when repeated at every lattice point. It can take any shape but always has the same volume for the same lattice. The determinant of the basis matrix is the volume of the fundamental region of the lattice. Each point p belonging to the lattice L(B) is formed by multiplying the basis matrix B by some integer combination vector s. This is equivalent to multiplying each basis vector b[i] by the corresponding component s[i] and summing them up, forming a vector linearly dependent on the basis vectors.

C. Ideal, cyclic and anti-cyclic lattices

The cyclic lattices are a special type of lattices used in ring versions of lattice based cryptosystems. A cyclic lattice has a basis matrix where each column is the previous column rotated down by one element. Cyclic lattices are ideals in the polynomial ring $Rq=Zq[x]/(x^n-1)$. This is the ring of polynomials modulo $f(x) = x^n-1$ with n coefficients modulo q. A cyclic lattice of n-dimensions, is equivalent to a polynomial ring R with n coefficients in Zq, where each element in the lattice maps to an element in the ring R. Adding two elements

978-1-5386-1191-3/17/$31.00 ©2017 IEEE

in ring R is equivalent to adding the corresponding elements in the lattice. Rotating a basis vector is equivalent taking a polynomial with same components and multiplying it by x then taking the result modulo $x^n–1$ in ring R.

Another type of ideal lattices are the anti-cyclic lattices. Where each column of the basis matrix is the previous column rotated by one element, and the element that spills over is negated. This is equivalent to multiplying by x modulo x^n+1. Hence anti-cyclic lattices are ideals in the ring $Rq = Zq[x]/(x^n–1)$. The majority of lattice-based cryptosystems use anti-cyclic lattices.

The cyclic lattices are formed from a single random vector as opposed to ordinary lattices that require a random matrix. Hence only one random vector needs to be generated and stored.

A lattice point is formed by multiplying the basis matrix B by some vector s. In cyclic lattices, this can be done by taking just the first basis vector as a polynomial in ring R and multiplying it by polynomial s. The naive matrix multiplication is done in $O(n^2)$ time, it can be improved to asymptotic time of $O(n^{2.8074})$ using Strassen algorithm. The naive polynomial multiplication is also done $O(n^2)$ time, but it can be improved to O(n) using the Number Theoretic Transform (NTT).

With anti-cyclic lattices, because of the negative wrapped convolution when multiplying polynomials mod x^n+1, before applying NTT, the components of the polynomials should be multiplied by the powers of the square root of the kernel of the transform, and the result of multiplication should be multiplied by the inverses of the powers of square root of the kernel. In order for there to be a square root of the kernel, the modulus q should be equal to 1 mod 2n.

II. Lattice-based Public Key Cryptosystems

To prove the security of a the cryptosystem when it is being designed, it should be proven that the challenge to break the cryptoystem is equivalent to solving some problem that is known to be hard. Early efforts in lattice-based cryptography focused on building a provably secure cryptoystem that is feasible. More recent works concentrate on improving efficiency and nearing the cryptosystems to practical requirements.

A. Early works

Lattice-based cryptography started in 1997 with the Ajtai-Dwork cryptosystem with provable security [3]. The cryptoystem is based on the hidden hyperplane problem (HPP).

The cryptosystem works as follows [19]: For a poitive integer *n*, the private key is defined as an n-dimesional secret vector. The secret vector describes a periodic set of parallel equidistant hyperplanes in the n-dimensional space: the secret vector is perpendicular to the hyperplanes, and the length of the vector is inversely proportional to the distance between consequent hyperplanes. The public key is defined as a set of points belonging to some of the hyperplanes. Each bit of the plaintext is encrypted in an n-dimensional point. If the message bit is 0 then the ciphertext is a uniformly random point. If the message bit is 1, the ciphertext is the centroid of a uniform subset of the public key points. Decryption is done by projecting the ciphertext point on the secret vector from the private key (scalar product with a unit vector), the message is 1 if the point is close to one of the hyperplanes, and 0 otherwise.

Since the cryptoystem was just a proof of concept it was very inefficient. Consequent works were focused on improving efficiency.

In 1997, the GGH cryptosystem was introduced by O. Goldreich, S. Goldwasser, and S. Halevi [4]. It improved the space efficiency compared to the Ajtai-Dwork cryptosystem, but did not provide a security guarantee. The private key is defined as a good basis, while the public key is defined as some bad basis. A good basis has basis vectors that are as short as possible and almost perpendicular to each other. A message is encoded as a small vector added to a lattice point (a point linearly dependent with the basis of the lattice). Then the point is reduced modulo the parallelepiped from the bad basis. With the private key, the message can be easily decrypted, by reducing the ciphertext vector modulo the fundamental parallelepiped from the good basis. But with a bad basis, decoding the message is equivalent to solving the closest vector problem (CVP) which is a hard problem. A good basis has the shortest set of linearly independent vectors. A bad basis may have vectors far away from the origin but describing the same lattice. This cryptosystem did not provide a security guarantee and later increasing attacks were discovered until the cryptosystem was insecure in any number of dimensions.

In 2001, D. Micciancio suggested to describe the public (bad) basis of the GGH cryptosystem in the Hermite normal form (HNF). There is only one HNF representation for each lattice. This halves the size of the public key because the HNF is a triangular matrix. Also this improves security slightly, because the HNF representation is considered the worst possible form to the attacker.

B. The LWE Public Key Cryptosystem

In 2005, O. Regev introduced one of the most important problems in lattice-based cryptography, the learning with errors (LWE) decision problem [5]. It states that given some pairs in the form (**a**[i], b[i]), where **a**[i] is a polynomial and b[i] is a single value – all modulo q, it is required to differentiate whether the term b[i] is uniformly random, or is dependent on the first component **a**[i] plus some small noise value: $b[i] = \mathbf{a}[i]*\mathbf{s} + e[i]$, where s is a secret vector and e[i] is a small error vector usually from the Gaussian distribution. In the ordinary LWE problem, the same pairs are given, and the goal is to find the secret vector s.

When the number of given equations is n, the decision LWE problem is equivalent to the following: given the basis matrix A of a lattice in the n-dimensional space (where the vectors **a**[i] form the rows of **A**) and a point b in n-D space, it is required to differentiate whether the point is close to some point belonging to the lattice or is it an unrelated uniformly random point.

Regev also built a cryptosystem based on the LWE problem [5]. The private key is defined as an integer combination s in Zq^n. While the public key is defined as the pair of: the good basis **A** of some lattice L(**A**), and a point belonging to the lattice, made using the secret combination plus a small Gaussian noise vector $\mathbf{b} = \mathbf{A}.\mathbf{s} + \mathbf{e}$. This interpretation translates to the set of equations: b[i] = **a**[i].**s** + e[i], where b[i] is a component of the noisy point, **a**[i] is the i^{th} row of the basis matrix, s is the secret combination, and e[i] is a small Gaussian noise that causes the hardness to an attacker when trying to find out the secret combination **s**. The multiplication in **a**[i].**s** is component-wise. The encryption is done by taking a subset of the pairs of **a**[i] and b[i] = **a**[i].**s** + e[i] and adding them up, then adding the message bit with the large amplitude floor(q/2).

Algorithm 1 The LWE cryptosystem.

Parameters:

Positive integers *n*, *m*, prime q

Generation:

Choose a uniformly random vector **s** in Zq^*n* (*n* integers modulo q), this is the private (secret) key.

Choose m independent uniformly random vectors **a**[i] in Z, i = 0,...,m–1, and a noise vector **e** in Zq^*m* (of size *m*) from a Gaussian distribution.

Define the components of vector **b** as:

$b[i] = \mathbf{a}[i].\mathbf{s} + e[i] \bmod q, \quad i = 0,...,m-1$

where the multiplication is the inner product:

$\mathbf{a}[i].\mathbf{s} = \sum a[i][j]*s[j]$.

The set of vectors **a**[i] and the vector **b** form the public key, consisting of *m* pairs (**a**[i], b[i]), i = 0,...,*m*–1, where **a**[i] is a vector of size *n* and b[i] is a single value, all values are modulo q.

Encryption:

For each bit in the plaintext message:

– A subset of indexes S is chosen uniformly from the *m* pairs (**a**[i], b[i]) of the public key. This is equivalent to forming a uniformly random bit string of m bits, where each bit decides if the corresponding pair is included in the subset.

– The pairs in the subset S are summed up, then if the message bit is true, the value floor(q/2) is added to c2.

$\mathbf{c}_1 = \sum$ i in S: **a**[i], in Zq^n

$c_2 = (\sum$ i in S: b[i]) + floor(q/2)*(message bit), in Zq

– This forms a ciphertext for a single bit consisting of a vector $\mathbf{c}_1$ of size n and a single value c_2, all values are modulo q.

Decryption:

$p = c_2 - \mathbf{c}_1.\mathbf{s} \bmod q$, in Zq

= floor(q/2)*(message bit) + ($\sum$ i in S: e[i])

message = decode(p) = abs(p) > floor(q/4) ? 1 : 0

The result is calculated modulo q in the range [–q/2, q/2]. Then in the decode function, the magnitude is compared with floor(q/4). The corresponding message bit is 1 if the magnitude was large and 0 otherwise.

In order for the message decryption to succeed, the error terms have to be small enough compared to floor(q/4). When the error terms are sampled from a Gaussian distribution with suitably small standard deviation, there is some negligible probability of decryption failure, in the unlikely event that some Gaussian error term turns out large comparable to q.

The sizes of the public and private keys are (m*n+m) and n coefficients respectively. One bit of plaintext is taken at a time to produce a ciphertext of (n+1) coefficients.

In 2008, D. Micciancio and O. Regev improved the space efficiency of the LWE cryptosystem by taking the uniform subset only once per message block instead of a new uniform subset for each message bit [6]. The cryptosystem can now be described with matrix operations [Algorithm 2].

Algorithm 2 The optimized LWE cryptosystem.

Parameters:

Positive integers *n*, *m*, *l*, t, prime q

Generation:

Generate uniformly random matrices:

A in Zq^(*m*n)*, **S** in Zq^(*n*l*)

Generate matrix **E** in Zq^(*m*l*) from discrete Gaussian distribution.

$\mathbf{P} = \mathbf{A}*\mathbf{S} + \mathbf{E}$, in Zq^(*m*l*)

The private key is matrix **S**.

The public key are the matrices (**A**, **P**).

Encryption:

Plaintext message **v** in Zt^*l* (plaintext block is partitioned into *l* coefficients modulo t)

Generate a uniform vector **a** in Z(2r)^*m*

$\mathbf{u} = \mathbf{A}^T \mathbf{a}$

$\mathbf{c} = \mathbf{P}^T \mathbf{a}$ + encode(**v**)

Ciphertext (**u**, **c**)

Decryption:

$\mathbf{ev} = \mathbf{c} - \mathbf{S}^T \mathbf{u}$

$= (\mathbf{A}\,\mathbf{S} + \mathbf{E})^T \mathbf{a}$ + encode(**v**) - $\mathbf{S}^T \mathbf{A}^T \mathbf{a}$

$= \text{encode}(\mathbf{v}) + \mathbf{E}^T \mathbf{a}$

v = decode(**ev**)

The plaintext is of size (*l*log2(t)) bits, the ciphertext is of size (*n*+*l*) coefficients mod q, the sizes of the public and private keys are (*m*(*n*+*l*)) and (*n***l*) coefficients mod q respectively. With an example parameter set from [6], $l = n = 166$, $m = 1319$, $q = 4093$, and $t = 2$, the sizes of the public and private keys are 641.47 KB and 40.37 KB respectively, the plaintext size is 166 bits (20 bytes and 6 bits), the ciphertext is 498 bytes, and the blowup factor is 24.

In 2009, C. Gentry introduced a fully homomorphic encryption technique [8]. This allows to perform operations on encrypted information. This can be useful when sending data for processing at an untrusted site.

In 2010, R. Lindner and C. Peikert studied the effect of parameter selection on the security of the optimized LWE cryptosystem, and proposed more secure and more efficient parameter sets [10]. They discovered that the modulus q

doesn't need to be high for better security. Their revised LWE cryptosystem takes a message of l bits and produces a ciphertext of (n_2+l) bits. The sizes of the public and private keys are (n_1*l) and (n_2*l) coefficients respectively. And with their medium security parameter set: l = 128, n_1 = n_2 = 256, and q = 4093 < 216, the plaintext block size is 16 bytes, and with 16-bit coefficients the ciphertext is 768 bytes and the ciphertext blowup factor is 48. With coefficients stored in 12-bits the ciphertext is 576 bytes, and the blowup factor is 36.

The LWE-based cryptosystems in [6], [10] use matrix multiplication, which can be done using Strassen algorithm in $O(n^{2.8074})$ asymptotic time instead of the naive method of $O(n^3)$. There are other methods with better asymptotic complexity but they have larger constant overheads.

C. Compact-LWE

In 2017, D. Liu, N. Li, J. Kim, and S. Nepal improved the security of Regev's original public key cryptosystem by generalizing the LWE problem into the Compact-LWE problem [7]. That is, the problem on which the hardness of the cryptosystem is based is now more generalized. The original Regev's LWE cryptosystem is subject to lattice-based attacks for certain parameter values. It is possible to recover the private key by solving the closest vector problem (CVP) with a small number of dimensions. The CVP is known to be hard, but it needs a high number of dimensions to be intractable. Hence when selecting values for parameters, lattice-based attacks have to be considered in Regev's LWE cryptosystem. But the Compact-LWE cryptosystem stays secure even if CVP can be solved with the selected parameter values.

Also the space efficiency is improved by working with more dense lattices. In the original Regev's LWE cryptosystem, the public key consisted of a set of samples (**a**[i], y[i]), where **a**[i] is a vector in Zq^n with components modulo q and y[i] is an integer in Zq. While in the Compact-LWE cryptosystem, the basis modulus b is smaller, so the vectors **a**[i] (the rows of basis matrix $\mathbf{A}^T$) have small components.

The Compact-LWE cryptosystem also eliminates the possibility of decryption failure that was present in Regev's LWE cryptosystem. Still, the Compact-LWE cryptosystem remains almost as space inefficient as Regev's.

D. The Ring-LWE Public Key Cryptosystem

In 2010, V. Lyubashevsky, C. Peikert, and O. Regev introduced a cryptoystem based on the LWE over polynomial rings (Ring-LWE) problem [9], influenced by the NTRU cryptosystem and Micciancio's one way function from ring-SIS. The Ring-LWE cryptosystem uses lattices that are ideals in a polynomial ring of the form R = Z[x]/f(x), where Z[x] is the set of polynomials of x with coefficients in Z, and f(x) is some chosen polynomial for modular reduction, thus ring R is the set of polynomials with integer coefficients modulo f(x). The most commonly used ring is $Rq = Zq[x]/(x^n+1)$ where n is a power of 2. This corresponds to the set of polynomials with n coefficients modulo q and powers of x up to n–1. In this case the set of anti-cyclic lattices in n dimensions is the ideal in ring Rq.

Polynomial multiplication in this ring Rq can be done component-wise in O(n) time by using the number theoretic transform (NTT), which is similar to FFT but instead of using complex roots of unity, the roots are integers modulo q. The transform can be done in O(n logn) time using the Cooley-Tukey algorithm just like FFT. But to use NTT in a polynomial ring efficiently, it is better when n is a power of 2, and q must satisfy the condition q = 1 (mod 2n) for there to exist an n^{th} root of unity w in Zq. The root w should also have a square root ψ in Zq in order to multiply polynomials in $Zq[x]/(x^n+1)$ with negative wrapped convolution.

Algorithm 3 The Ring-LWE cryptosystem.

Parameters:

Positive integer n, prime q.

Polynomial ring $Rq=Zq[x]/(x^n+1)$.

Generation:

Choose element a uniformly from ring Rq

Gaussian error vector e in Rq, and a secret s in Rq with small coefficients, in {-1,0,1}.

$b = a*s + e$, in Rq

Private key: s

Public key: (a, b)

Encryption:

For a message m of n bits

Generate error terms e1, e2 in Rq with coefficients from Gaussian distribution.

Generate a term t in Rq with small coefficients, in {-1,0,1}

$c_1 = a*t + e1$

$c_2 = b*t + e2 + encode(m)$

where the encoding function is:

encode(m) = m*floor(q/2)

forming a vector where each component is the corresponding message bit multiplied by the amplitude floor(q/2).

Decryption:

Calculate $p = c_2 - c_1*s$ with a small magnitude in [–q/2, q/2]

$p = c_2 - c_1*s$, in [-q/2, q/2]

m = decode(p)

where decode(p) = abs(p) > floor(q/4) ? 1 : 0.

The Ring-LWE cryptosystem takes a message of n bits and produces a ciphertext consisting of two polynomials c1 and c2 each with n coefficients modulo q. With chosen parameters n = 1024, and $q = 12286 < 2^{16}$, the plaintext is 128 bytes and the ciphertext takes up to 4096 bytes without compression, with a ciphertext blowup factor of 32.

Advantages of the ring-LWE cryptosystem over LWE:

1. Element multiplication can be done in O(n) time using the Number Theoretic Transform (NTT) instead of the naive way in $O(n^2)$. The NTT itself can be done in O(n logn) time using the Cooley-Tukey algorithm instead of the naive way in $O(n^2)$.

2. Element size is O(n) instead of $O(n^2)$, since a single vector needs to be stored instead of the entire n*n matrix.

In 2013 T. Pöppelmann and T. Güneysu [11] showed that in the Ring-LWE cryptosystem, up to 7 least significant bits of the ciphertext polynomial c2 can be discarded and the message will still be preserved at the roughly same probability. This improved space efficiency, with $n = 1024$, $q = 12286$ the ciphertext blowup factor decreases to about 25.

Then in 2013 D. Cabarcas, F. Göpfert, and P. Weiden [12] studied the security impact of using error vectors with small uniform coefficients instead of sampling from discrete Gaussian distribution, in the Ring-LWE cryptosystem. Sampling discrete Gaussians requires much more calculations than uniform distribution.

In 2016, the LWE and Ring-LWE cryptosystems were given portable implementations in JavaScript by Y. Yuan, C. M. Cheng, S. Kyimoto, Y. Miyake, and T. Tagaki [14].

Then, in 2016, C. Peikert studied the weak instantiations and attacks to the Ring-LWE cryptosystem.

III. Key Encapsulation Mechanisms

A. Ding's KEM

In 2012, J. Ding. X. Xie and X. Lin [15] developed a provably secure key encapsulation mechanism (KEM) using LWE and its ring-LWE equivalent.

The KEM works by adding small even error vectors to some commonly known vector a multiplied by the first side's private key vector s. Then reconciliation functions are used that involve rounding. In total, the two ends exchange two elements of the polynomial ring Z[x]/f(x) to obtain one bit of the key.

An error reconciliation function is used. It uses an ordinary rounding function and a cross-rounding function.

In LWE-based key exchange, each of the two parties exchange a vector in Zq and a hint bit, to obtain one bit of exchanged key. In the ring-LWE equivalent, each of the two parties exchange an element in ring $Rq=Zq[x]/(x^n+1)$ and a hint binary string of n bits, to obtain a key of n bits.

Each hint bit tells if the corresponding coefficient of the session secret vector Kb is in the range [-floor(q/4)+b, floor(q/4)+b] or not, where b is a uniformly random bit. With odd modulus q, the key bits are slightly biased to zero.

B. Peikert's modification to Ding's KEM

In 2014, C Peikert modified Ding's key exchange mechanism to conserve bandwidth, and showed an actively secure version of the KEM [16]. It uses a different reconciliation mechanism to extract key bits compared to Ding's KEM. This is to eliminate the bias to zero that occurs when using an odd modulus q.

Each bit of the resultant session key is equal to the condition of whether the corresponding coefficient in the secret vector Kb is "small" (in the range [-q/2, q/2]) or not.

In 2015, J. W. Bos, C. Costello, M. Naehrig, and D. Stebila integrated Peikert's KEM into TLS, analyzed the security, and compared the performance to other KEMs used in TLS/SSL [17].

C. NewHope KEM

In 2016, E. Alkim et al. [18] showed the unauthenticated key encapsulation mechanism NewHope. It uses the polynomial ring polynomial ring $R=Zq[x]/(x^n+1)$, where $n = 1024$, $q = 12289$. The client and server each send an element from the polynomial ring of size 2048 bytes (which can be compressed to 1792 bytes) and other parameters smaller in size to obtain a common key of 32 bytes. The ratio of ciphertext and data sent, to the encapsulated key size is roughly 60 with data compression.

In this KEM, the reconciliation mechanism is changed again to decrease the probability of reconciliation failure. The reliability and security is improved from the Peikert KEM by encoding a single bit of the key in each 4 (instead of 1) coefficients of sent ring elements.

IV. Conclusion

Performance. Lattice-based cryptography remained very impractical until the introduction of Regev's LWE cryptoystem. Since then, the Ring-LWE cryptosystem significantly reduced the key sizes, from matrices to vectors.

Also, lattice-based KEM reached practical levels of reliability and security with recent efforts.

Recent efforts in key encapsulation mechanisms practical.

Security. Passive (semantic) security is guaranteed from the LWE assumption: the public key and ciphertext are indistinguishable from uniformly random distributions. Active security can be achieved either using the Fujisaki-Okamoto transformation as in Peikert's KEM, or using trapdoor (one-way) functions as in the GGH cryptosystem.

Security against quantum-based attacks.

Difference from classical PKC. The generation, encryption and decryption times in modern lattice-based cryptosystems are almost always faster than in classical PKCs. But they have a large ciphertext expansion factor of around 30, compared to classical PKCs where the ciphertext has the exact same size as the plaintext.

Future work includes efficient implementation of the Lattice Based Cryptosystem, enhancing the algorithms and formulating attack models.

REFERENCES

[1] M. Rose. “Lattice-based cryptography: A practical implementation,” 2011.

[2] M. Ajtai. “Generating hard instances of lattice problems.” In Proceedings of 28th annual ACM symposium on Theory of computing (pp. 99-108). ACM, 1996.

[3] M. Ajtai, C. Dwork. “A public key cryptosystem with worst-case/average-case equivalence.” In Proceedings of the 29th annual ACM symposium on Theory of computing (pp. 294-293). ACM, 1997.

[4] O. Goldreich, S. Goldwasser, and S. Halevi. “Public-key cryptosystems from lattice reduction problems.” In Advances in Cryptography CRYPTO'97: 17th Annual International Cryptography Conference, Santa Barbara, California, USA, August 1997. Proceedings (p. 112). Springer Berlin/Heidelberg 1997.

[5] O. Regev. “On lattices, learning with errors, random linear codes, and cryptography.” In Journal of the ACM, 56(6) p. 34, 2009 Sept 1.

[6] D. Micciancio, and O. Regev. “Lattice-based cryptography.” In Post-quantum cryptography (pp. 147-191). Springer Berlin Heidelberg 2009.

[7] D. Liu, N. Li, J. Kim, and S. Nepal. “Compact-LWE: Enabling practically lightweight public key encryption for leveled IoT device authentication.” In Cryptology ePrint Archive 2017: 685.

[8] C. Gentry. “Fully homomorphic encryption using ideal lattices.” STOC vol. 9 No. 2009. 31 May 2009.

[9] V. Lyubashevsky, C. Peikert, and O. Regev. “On ideal lattices and learning with errors over rings.” In Annual International Conference on the Theory and Applications of Cryptographic Techniques (STOC, pp. 169-178). 20 May 2010.

[10] R. Linder, and C. Peikert. “Better key sizes (and attacks) for LWE-based encryption.” In CT-RSA. Vol. 6558, pp. 319-339. 2011.

[11] T. Pöppelmann, and T. Güneysu. “Towards practical lattice-based public-key encryption on reconfigurable hardware.” In International Conference on Selected Areas in Cryptography (pp. 68-85). Springer, Berlin, Heidelberg, 2013.

[12] D. Cabarcas, F. Göpfert, and P. Weiden. “Provably secure LWE encryption with smallish uniform noise and secret.” In Proceedings of the 2nd ACM Workshop on ASIA public-key cryptography (pp. 33-42). ACM, 2014.

[13] V. Lyubashevsky, C. Peikert, and O. Regev. “A toolkit for ring-LWE cryptography.” In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 35-54). Springer, Berlin, Heidelberg, 2013.

[14] Y. Yuan, C.-M. Cheng, S. Kyimoto, Y. Miyake, and T. Tagaki. “Portable implementation of lattice-based cryptography using JavaScript.” International Journal of Networking and Computing 6.2 (pp. 309-327). 2016.

[15] J. Ding, X. Xie, and X. Lin. “A simple provably secure key exchange scheme based on the LWE problem.” In IACR Cryptology ePrint Archive 2012: 688.

[16] C. Peikert. “Lattice Cryptography for the Internet.” In international workshop on post-quantum cryptography (pp 197-219). Springer, Cham 2014 Oct 1.

[17] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila. “Post-quantum key exchange for the TLS protocol from the RLWE problem.” In Security and Privacy (SP), 2015 IEEE Symposium on IEEE (pp. 553-570), 2015.

[18] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. “Post-quantum key exchange – a new hope.” In USENIX Security Symposium (pp. 327-343). 2016 Jan 1.

[19] C. Peikert. "A decade of lattice cryptography." Foundations and Trends® in Theoretical Computer Science 10.4 (2016): pp. 283-424.

[20] C. Peikert. "How (not) to instantiate Ring-LWE." In International Conference on Security and Cryptography for Networks. Springer International Publishing (pp. 411-430). 2016.

[21] Hegazy, Ola M., Ayman M. Bahaa-Eldin, and Yasser H. Dakroury. "Quantum Secure Direct Communication using Entanglement and Super Dense Coding." arXiv preprint arXiv:1402.6219 (2014).

[22] Hamouda, Israa, Ayman M. Bahaa-Eldin, and Hazem Said. "A generalized Grover's algorithm with access control to quantum databases." 2016 11th International Conference on Computer Engineering & Systems (ICCES), pp. 281-285. IEEE, 2016.

[23] Hamouda, Israa, Ayman M. Bahaa-Eldin, and Hazem Said. "Quantum databases: Trends and challenges." 2016 11th International Conference on Computer Engineering & Systems (ICCES), pp. 275-280. IEEE, 2016.