A Summary of Lattice-Based Zero-Knowledge Proofs

Charlotte Doughty

April 19, 2026

Contents

1 Notation

Notation	Meaning
$\mathcal{L}$	An integer lattice
$\mathcal{R}$	The ring equal to $\mathbb{Z}[X]/(X^N+1)$
$\mathcal{F}(\mathcal{B})$	The fundamental domain of the lattice with basis $\mathcal{B}$
$\text{Disc}(\mathcal{L})$	The discriminant of lattice $\mathcal{L}$
$\det(\mathcal{L})$	The determinant of lattice $\mathcal{L}$
v_i^*	The Gram-Schmidt orthogonoalization of vector v_i
$C,\ C(m)$	A commitment, a commitment to m

2 Introduction

Over the past fifty years as information and computing moves increasingly to digital platforms, cryptography and the mathematics of protecting sensitive information has grown as a valuable mathematical field. At the same

time, quantum computing is nearing, which would allow many more computations to be done much more easily than can be currently accomplished on classical computing systems. Quantum computing, when it arrives in large-scale form, will break many of our preexisting cryptographic protocols and problems, including the widely-used RSA [12]. Although there are many existing hard problems that quantum computers cannot solve in practical time as of yet, the fundamental lattice problems are of special interest as they are **trapdoor functions**, meaning that they can be easily computed in one direction, but are incredibly difficult to reverse. Therefore, there has been a recent exploration into and development of lattice-based cryptographic protocols and systems.

One of those types of protocols explored recently has been zero knowledge proofs. Developed in the 1980s, zero knowledge proofs are a mathematical structure to prove knowledge of a specific fact without sharing the details of that fact. It has become increasingly useful in identification schemes in cryptography, and in more applied contexts is used for blockchain privacy [14].

The combination of these two fields in cryptography is a promising and rapidly developing space. Modern cryptographers are working on developing new commitment schemes and proofs and making preexisting schemes and protocols more efficient. In this paper, we will be reviewing some of the major commitment schemes and zero knowledge proof protocols shaping the field today.

3 Background

3.1 Lattice Fundamentals

To begin, I define a **lattice** to be a discrete subgroup of $\mathbb{R}^n$. These discrete subgroups are defined by a basis of n vectors with entries in $\mathbb{R}$ as

$$\mathcal{L} = \mathcal{L}(\mathcal{B}) = \sum_{i=1}^{n} c_i b_i : c_i \in \mathbb{Z}$$

and are finitely generated free abelian groups of rank at most n. Each lattice can be defined by many different bases of vectors. Each unique basis forms an associated **fundamental domain**, the n-dimensional parallelepiped defined by the set

$$\mathcal{F}(\mathcal{B}) = \{t_1 v_1 + ... + t_n v_n : 0 \leq t_i < 1\}.$$

The volume of the fundamental domain, or the parallelepiped, is the **determinant** of the lattice, which is also equal to the absolute value of the determinant of the matrix formed by the basis vectors of the lattice. The **discriminant** is equal to the determinant squared.

3.2 Minima and Bounding

Many cryptologic lattice questions revolve around the shortest vector in the lattice, or minimum vector length. There are a series of fundamental theorems related to guaranteed small vector lengths in lattices.

Minkowski's Theorem: For a lattice $\mathcal{L} \in \mathbb{R}^n$ and a ball $S \subset \mathbb{R}^n$ with volume $\text{Vol}(S) > 2^n \det(\mathcal{L})$, then S contains a nonzero lattice vector.

Hermite's Theorem: Every lattice $\mathcal{L}$ of dimensions n contains a nonzero vector $v \in \mathcal{L}$ such that

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

Hermite's theorem derives from the special case of Minkowski's theorem where S is a regular ball centered around the origin, as presented in Theorems

7.25 and 7.28 respectively in [7]. These two theorems form the foundational understanding of upper bounds on the shortest vector length in a lattice. They combine, along with the concept of the lattice discriminant, into Theorem 14.178 in [15]:

Hermite Minkowski Theorem: There exists a constant c_n such that every lattice $\mathcal{L} \subset \mathbb{R}^n$ contains a nonzero vector satisfying

$$||v|| \leq \gamma_n^{1/2} \text{Disc}(\mathcal{L})^{1/n},$$

where

$$\gamma_n^{1/2} = \sup_{\mathcal{L} \subset \mathbb{R}^n} \frac{min\{||v|| : v \in \mathcal{L}, v \neq 0}{\text{Disc}(\mathcal{L})^{1/2}}.$$

The value of $\gamma_n^{1/2}$ is only known for $n \leq 8$ and $n = 24$. In general $\gamma_n \approx \frac{n}{\pi e}$.

This trio of theorems guarantees the existence of a vector in the lattice of relatively small length; however, there may exist other shorter vectors. It will be noted that the discriminant of the lattice, $\text{Disc}(\mathcal{L})$, figures heavily in these theorems. If the lattice's basis vectors are pairwise orthogonal, then the discriminant is simply the product of the magnitudes of the basis vectors. However, since this is often not the case, the following inequality from Proposition 7.19 of [7] holds:

Hadamard's Inequality: Let $\{v_1, ..., v_n\}$ be a basis for the lattice $\mathcal{L} \subset \mathbb{R}^n$. Then

$$\text{Disc}(\mathcal{L}) \leq ||v_1|| \cdot ||v_2|| \cdots ||v_n||.$$

3.3 Bases

For a lattice $\mathcal{L}$ with basis $A = [a_1\ a_2\ ...\ a_n]$, there exist infinite other bases $B = (b_1\ b_2\ ...\ b_n]$ and a matrix S with integer entries such that $B = S * A$. As S must be invertible to translate from one basis to another, $\det(S) = \pm 1$. Therefore,

$$|\det(\mathcal{L}(B))| = |\det(\mathcal{L}(A))| = |\det(\mathcal{L})|.$$

3.4 Babai's Algorithm

The easiest way to find the shortest vector in a lattice is to simply pick out the shortest vector in the "best" basis for the lattice. By "best," we usually mean the most orthogonal, with the shortest vector lengths. However, when we are not given a best basis to start with, we have to try to get as close as possible to the best basis (since it is an established tenet of the security of lattice cryptography and lattice problems that it is computationally infeasible for an outsider to find the best basis of a lattice).

However, if one has the best basis, then finding the closest vector becomes much easier. For example, in the GGH cryptosystem, Alice, the person decrypting who has access to the best basis, uses the best basis and Babai's algorithm to find the closest vector to the target. Babai's algorithm simply attempts to go to the nearest vector point by rounding the scalar coefficients of the target vector when expressed as a span of the best basis. The algorithm is stated more formally below:

For a target vector w and basis $\mathcal{B} = \{b_1, ..., b_n\}$, write as $w = a_1b_1 + ... + a_nb_n$, where $a_i \in \mathbb{R}$. Then round each a_i, and find the approximate closest vector of

$$w* = \lfloor a_1 \rceil b_1 + ... + \lfloor a_n \rceil b_n$$

Although Babai's algorithm is most useful when integrated into a particular protocol or when the party using the algorithm has the correct, best basis, it can still be a useful tool. If used with a bad basis, it will be completely unhelpful. However, if it is used in conjunction with an almost-good basis, like one that might result from completing the Lenstra-Lenstra-Lovász Algorithm, then one may be able to approximate a close vector.

3.5 Lenstra-Lenstra-Lovász Algorithm

In lattices of dimension 2, one can use Gaussian reduction of the basis vectors to find a short vector in the lattice. However, in higher dimension, this becomes more and more difficult. So, in 1982, Lenstra, Lenstra, and Lovász

[8] developed a polynomial-time algorithm to orthogonalize and reduce the size of basis vectors in order to make a "better" basis and find a short vector.

For a basis $\mathcal{B} = \{v_1, ..., v_n\}$ for a lattice $\mathcal{L} \subset \mathbb{Z}^n$, the algorithm uses Gram-Schmidt orthogonalization denoted by v_i^* and a helper function of

$$\mu_{i,j} = \frac{v_i \cdot v_j^*}{||v_j||^2} = \frac{v_i \cdot v_j^*}{v_j * \cdot v_j *}.$$

While Gram-Schmidt is useful to orthogonalize the basis vectors, it unfortunately does not remain within the lattice domain, instead creating a new basis in the corresponding vector space, mapping a projection of each basis vector onto the span of the basis. The goal is to have a final basis that is "LLL-reduced," meaning it satisfies the following 2 conditions:

1) **Size Condition:** $\mu_{i,j} \leq \frac{1}{2}$ for $1 \leq j < i \leq n$

2) **Lovász Condition:** $||v_i^*||^2 \geq ||v_{i-1}^*||^2(\frac{3}{4} - \mu_{i,i-1}^2)$

Size reduction is achieved, as shown above, by subtracting integer multiples of earlier, smaller basis vectors from other basis vectors. The Lovász condition is achieved by swapping basis vectors that fail the condition and continue the algorithmic process. The algorithm alternates between reducing the size of basis vectors by subtracting integer multiples of other basis vectors earlier in the ordered set, and swapping adjacent vectors. The swapping of adjacent vectors ensures that "best", shortest vectors are being used to decrease the size of other basis vectors. The algorithm below runs iteratively until certain standards are met.

Setup: set $k = 2$, $v_1^* = v_1$, v_i^* indicates the Gram-Schmidt vector corresponding to v_i *recalculated at the time of reference.*

While $k \leq n$:
    $j = k - 1$
    While $j \geq 0$:
        Set $v_k = v_k - \mu_{k,j} v_j$
    If v_k^* meets the Lovász condition:
        $k+ = 1$

Else:

$$\text{Swap } v_{k-1} \text{ and } v_k \text{ and set } k = \max(k-1, 2)$$

At the end of these loops, the basis should be LLL-reduced and fit the size and Lovász conditions. While the LLL-algorithm cannot output the smallest or most orthogonal basis, it can achieve a more serviceable basis, which is helpful in getting closer to solving SVP and CVP. Combined with Babai's algorithm, LLL can help in solving the approximate CVP problem, in which one cannot find the closest vector, but an approximately close vector to the target.[1]

Let us provide an example of the LLL-algorithm for a 3-dimensional lattice. Let us choose our starting "bad" lattice basis A to be

$$B = \{v_1, v_2, v_3\},$$

$$v_1 = (1,1,0), \quad v_2 = (3,1,0), \quad v_3 = (3,-3,1)$$

We start by setting $k = 2$, $v_1^* = v_1 = (1,1,0)$, and $j = 1$ as prescribed by the algorithm. Now we must calculate

$$\begin{aligned} v_2 &= v_2 - \mu_{2,1} \cdot v_1 \\ \mu_{2,1} &= \frac{v_2 \cdot v_1^*}{||v_1||^2} = \frac{(3,1,0)\cdot(1,1,0)}{||(1,1,0)||^2} = \frac{4}{2} = 2 \\ v_2 &= (3,1,0) - 2\cdot(1,1,0) = (1,-1,0) \end{aligned}$$

We next calculate the Gram-Schmidt vector v_2^*.

$$\begin{aligned} v_2^* &= v_2 - \mu_{2,1} \cdot v_1^* \\ &= v_2 - \frac{(1,-1,0)\cdot(1,1,0)}{(1,1,0)\cdot(1,1,0)} \cdot v_1^* \\ &= (1,-1,0) - 0\cdot(1,1,0) \\ &= (1,-1,0) \end{aligned}$$

Now, we check if v_2^* satisfies the Lovász condition.

$$||v_2^*||^2 = 2, \ ||v_1^*||^2 \cdot (\frac{3}{4} - \mu_{2,1}^2) = 2 \cdot (\frac{3}{4} - 0^2) = \frac{3}{2}$$

[1] In most cryptographic applications, one ends up actually proving that one can solve the approximate CVP, since it is difficult to find the provably closest vector, even with the best possible basis.

Since $2 \geq \frac{3}{2}$, the Lovász condition is satisfied, and we can continue to $k = 3$.

For the $k = 3$ pass, we set j = 2 and begin by calculating v_3:

$$\begin{aligned} v_3 &= v_3 - \mu_{3,2} \cdot v_2 \\ &= v_3 - \frac{(3,-3,1) \cdot (1,-1,0)}{(1,-1,0)(1,-1,0)} \cdot v_2 = \\ &(3,-3,1) - 3 \cdot (1,-1,0) = (0,0,1) \end{aligned}$$

Next, we calculate v_3^*.

$$\begin{aligned} v_3^* &= v_3 - \mu_{3,2} \cdot v_2^* - \mu_{3,1} \cdot v_1^* \\ &= v_3 - \frac{(0,0,1) \cdot (1,-1,0)}{(1,-1,0) \cdot (1,-1,0)} \cdot v_2^* - \frac{(0,0,1) \cdot (1,1,0)}{(1,1,0) \cdot (1,1,0)} \cdot v_1^* \\ &= (0,0,1) - \frac{3}{2} \cdot (1,-1,0) - \frac{3}{2} \cdot (1,1,0) \\ &= (0,0,1). \end{aligned}$$

Now, we check if v_3^* satisfies the Lovász condition.

$$||v_3^*||^2 = 1, \quad ||v_2^*||^2 \cdot (\frac{3}{4} - \mu_{3,2}^2) = 2 \cdot (\frac{3}{4} - (\frac{3}{2})^2) = -3$$

As $1 \geq -3$, the Lovász condition is satisfied, and we have finished with our LLL reduction of the basis. Our final, LLL-reduced basis is

$$B' = \{(1,1,0), (1,-1,0), (0,0,1)$$

which is obviously quite a short and orthogonal basis, as this example was designed to produce for clarity.

3.6 Lattice-Based Cryptography

Lattices form particularly compelling problems because they can be defined by many different vector bases. These bases are not equally useful; generally the more orthogonal and shorter the basis vectors are, the better the basis is for humans and computers to navigate the lattice. This feature is the basis of two fundamental lattice problems: the **Shortest Vector**

Problem (SVP), in which one must find the shortest vector in the lattice, and the **Closest Vector Problem (CVP)**, in which one must find the closest vector in the lattice to a specific point in $\mathbb{R}^n$. "Good" bases, or orthogonal and short bases, simply traverse the lattice and can easily find the closest vector as a simple integer linear combination of the basis vectors and find the shortest vector as, typically, the shortest basis vector. "Bad" bases, or bases with long and less orthogonal vectors, require complicated linear combinations to find the exact shortest and closest vectors, often requiring many large positive and negative integer coefficients. Algorithms like **Babai's Algorithm** that use the basis to find the shortest or closest lattice vector cannot handle these "bad" bases and return a vector that is not in fact the closest or shortest vector. Thus, if Eve has a bad basis and Alice and Bob have a good basis, then Eve cannot find the correct closest and shortest vectors in that lattice, unlike Alice and Bob.

Lattices can form the underlying structure and problems for successful cryptographic protocols as shown in findings by Miklós Ajtai [2], in which it was proven that if there exists a probabilistic, polynomial time algorithm to find a short vector in a random lattice $\mathcal{L} \subset \mathbb{Z}^n$, then there is also a probabilistic, polynomial time algorithm that solves the following three lattice problems:

1) Finding the length of the shortest nonzero vector in the lattice

2) Finding the shortest nonzero vector in the lattice

3) Finding the "best", shortest basis for the lattice

This finding proved that an average-case lattice problem is equally as difficult to solve as a worst-case lattice problem, as there exists to date no polynomial time algorithm to solve the three questions above. Thus, lattice schemes based on these problems provide good cryptographic foundations, since there exists no algorithm to "break" the problem.

Lattice-based cryptography is especially of interest in the current cryptologic community since it remains quantum-secure. Although there does not exist an algorithm or proof to determine whether lattice-based problems are definitively quantum-secure, lattice-based problems have yet to be broken by

quantum computing methods, as shown in detail by Oded Regev [13].

In modern high-volume cryptography there is a danger that too many interactions could cause an outside party to learn about the inner workings of the cryptographic protocol. Thus, it is important that the distribution of information released to the public does not show any pattern or information discernible to an outside eye. To combat this, many cryptosystems employ **rejection sampling**, a concept taken from statistical analysis. As the signer, prover, or encryptor utilizes randomness in their calculations, they check to see if their final publicized value(s) maintain the distribution principle (specific to whatever protocol is being deployed). If so, then they publish their calculations and the protocol continues. If not, they go back and choose a different random value(s) and recalculate.

3.7 Module-SIS and Module-LWE

While CVP and SVP are often described as the two fundamental problems in lattice cryptography, many modern schemes do not work directly off of these problems, but off of similar problems. These problems are adaptations of the **Short Integer Solution** (SIS) and **Learning With Errors** (LWE) that replace polynomials with vectors of polynomials in R_q^k, as explored in more detail in [10].

Module-SIS (Short Integer Solution): Find a small vector x such that $Ax = 0$. In Module-SIS, matrix A has polynomial entries, and x is a vector of polynomials. Of course, the trivial solution is automatically excluded from this problem. In 1996, Miklós Ajtai [2] showed that an average case of SIS is as hard to solve as a worst case of SVP. The matrix A is anti-circulant, meaning blocks of entries cycle through the rows from right to left. The elements of A are typically randomly selected.

Module-LWE (Learning With Errors: Find a secret vector s from a series of approximate linear equations involving s. Or alternatively, find s for $As + e = b$, where b is a known vector, A is a known matrix, and e is an unknown vector of small norm. As in SIS, A is also anti-circulant and its entries are typically randomly selected. It was introduced by Oded Regev

[13] in 2005.

3.8 Zero-Knowledge Proofs

Zero-Knowledge Proofs (ZKPs) were developed in 1985 by Goldwasser et al. [6] as a protocol to verify knowledge of a secret without sharing the details of that secret. Proofs are structured in a **commitment-challenge-answer** format, with most proofs requiring a certain threshold of challenges and correct corresponding answers in order to be accepted as valid. For example, if one picked a card from a deck and tried to prove that the card is red without revealing the card itself, the below proof scheme would follow:

1) The prover commits to the statement that the card is red.

2) The verifier challenges the prover to reveal a black card.

3) The prover reveals a black card.

This process would continue 26 times until the prover has shown the verifier all 26 black cards, which sufficiently proves that the prover is holding a red card.

Every zero knowledge proof must satisfy two conditions:

1) **Completeness:** If the prover does know the secret they are committing to, then the verifier should always accept the proof

2) **Soundness:** If the prover does *not* know the secret the are committing to (or is some kind of outside party), then the probability that the verifier will accept all of the prover's responses should be quite small

These two concepts hold up the foundational security of zero knowledge proofs, allowing them to be useful, highly reliable tools in cryptography. Without completeness, no secret can be definitely proven to be known. Without soundness, the problem underlying the zero knowledge scheme is solved,

if an outside actor can simulate the proof.

Related to this concept of zero-knowledge proofs is the **commitment scheme**, which is a process by which a prover commits to a statement or value (used in the first step of zero-knowledge proofs). The two features that are important for commitment schemes to function in earnest are:

1) **Hiding:** From the final value representing the commitment, outsiders cannot decipher what value was committed to.

2) **Binding:** Once a commitment is made, it cannot be altered.

Hiding and binding is extremely important in the security of commitment schemes. First, hiding is essential, as it protects the secret statement that is being committed, which is of course the fundamental principle behind zero knowledge proofs. Binding protects from deception by outside parties, preventing replication of commitments.

While some zero knowledge proof protocols and purposes work well with the interactive nature of challenges and responses, some protocols struggle under so much information being passed. To circumvent challenges, a **non-interactive zero-knowledge proof** strategy was devised by Amos Fiat and Adi Shamir [4] using hash functions, with the end result mimicking digital signatures. This was done by hashing the commitment value and/or the public values. Then this hash, which must by nature seem random and be unable to reverse-engineer or predict, replaces the verifier's random challenge to the prover. Then the prover makes the same calculation with the hash as they would in the interactive version of the proof with the challenge, and then send that calculation over to the verifier. Then the verifier confirms the calculation just as they would in an interactive protocol. The hash by the prover of the commitment value allows each party to compute without multi-step communication.

The privacy of zero knowledge proofs even between communicating parties makes them highly attractive for modern cryptography. Although protocols are being designed and refined on many different mathematical problems, we will be focusing on those built on $\mathcal{NP}$-hard lattice problems.

4 Lattice-Based Zero-Knowledge Proof Schemes

If one were to attempt to apply preexisting zero knowledge proof techniques to lattices and their problems, it may become computationally costly and inefficient, as proven in detail in [5]. So, since the foundational work by Ajtai, Regev, and Goldwasser et al., researchers recently have begun developing and refining specific lattice-based zero knowledge proof schemes and protocols, ranging from theoretical to practical, with most of the contemporary work focused on increasing efficiency. Below, I describe some key research and protocols.

4.1 Ajtai's Commitment Scheme

Although not explicitly stated in "Generating Hard Instances of Lattice Problems," a commitment scheme was implicitly proposed. While this scheme is fundamentally built bit by bit, it has expanded to a full k-bit message commitment, and even beyond. The scheme involves two matrices, A_1 and A_2 with randomly selected entries, a message $m \in \{0,1\}^m$ and randomness $r \in \{0,1\}^m$. The commitment [11] is calculated below:

$$C(m,r) = A_1 \cdot m + A_2 \cdot r \mod q$$

An important key to this proof is that both m and r are small, meaning that they must have small coefficients modulo q. A great benefit of this commitment scheme is that the dimension of m has no affect on the size of the commitment. For more on this detail, see [9].

4.2 Zero Knowledge Proofs of SVP and CVP

One of the first foundational proof schemes in lattice-based zero knowledge proofs was the work of Daniele Micciancio and Salil Vadhan in proving knowledge of a short vector in GapSVP and knowledge of a close vector in GapCVP.

GapSVP and GapCVP are similar to standard SVP and CVP, but instead of finding exactly the shortest or closest vector, it requires a few other parameters. Both GapSVP and GapCVP [1] require a distance bound $d \in \mathbb{Z}$, and a factor γ. GapSVP requires that given a lattice, if the length of the shortest vector is less than d, then a YES is returned, and if the length of the shortest vector is greater than γd, then a NO is returned. GapCVP similarly requires that given a lattice and a target vector, if the distance from the target vector to the closest vector in the lattice is less than d, then a YES is returned, and if the distance from the target vector to the closest vector int he lattice is greater than γd, then a NO is returned.

4.3 Proof of Plaintext Knowledge in Ajtai-Dwork

In 2005, Shafi Goldwasser and Dmitriy Kharchenko used Micciancio and Vadhan's lattice-based proof framework to provide a **plaintext proof of knowledge** - a zero knowledge proof that the plaintext m corresponds with the ciphertext c, for Ajtai and Dwork's lattice cryptosystem. As Ajtai and Dwork's cryptosystem uses bit-based plaintext, Goldwasser and Kharchenko constructed two zero knowledge proofs: one to prove that the ciphertext decrypted to '0' and another to prove that the ciphertext decrypted to '1'.

4.4 BDLOP

While all the above protocols were focused on proving secrets related to preexisting problems and encryption schemes, the BDLOP commitment scheme can be used to prove knowledge of an independent secret. The BDLOP [3] commitment scheme is a zero-knowledge proof of knowledge on the secret witness inside of a commitment, known as an **opening scheme**. In this scheme, we work over the ring $R_q = \mathbb{Z}[X]/(X^N + 1)$ to commit to and prove knowledge of a message $x \in R_q^\ell$. To set up, we create $A_1 \in R_q^{n \times k}$ and $A_2 \in R_q^{\ell \times k}$ such that

$$A_1 = [I_n \ A_1'] \text{and} A_2 = [0^{\ell \times n} \ I_\ell A_2']$$

where A_1' and A_2' are randomly selected to fill out A_1 and A_2. The commitment to x involves a randomly selected $r \in \mathbb{Z}^k/(X^{N+1})$ with a norm below a specified bound. The commitment value that is then generated and shared is computed by

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} r + \begin{bmatrix} 0^n \\ x \end{bmatrix}$$

The proof is completed in 3 moves, beginning with the prover calculating and sending to the verifier $t = A_1 y$ for $y \in$ a discrete normal distribution over R centered at 0. The verifier then challenges the prover with d randomly selected from the space of commitments. The prover then calculates $z = y + dr$ and sends z to the verifier. Finally, the verifier accepts if:

$$A_1 z = t + dc_1$$

and t's ℓ_2-norm is below a certain bound. This verification simplifies to

$$A_1 y + A_1 dr = t + dc_1,$$

which is clear based on our generation of t and c_1.

4.5 ABDLOP

The ABDLOP scheme improves on other preexisting schemes in proving knowledge of a short vector s such that

$$As = t \mod q,$$

building on the Module-SIS and Module-LWE problems. In this section we give a summary of the key material in the article [9]. The innovation in the proof is in combining the Ajtai and BDLOP commitments into one. The high-level thought is to combine the strengths of each system: Ajtai's ability to handle secrets in high dimensions without adverse effects to the commitment size, and BDLOP's ability to take large coefficients modulo q for secrets. So we commit to a message s_1 with a small norm (Ajtai) and a commitment to a message m with large coefficients at the same time using a small random s_2 by computing:

$$\begin{bmatrix} A_1 \\ 0 \end{bmatrix} s_1 + \begin{bmatrix} A_2 \\ B \end{bmatrix} s_2 + \begin{bmatrix} 0 \\ m \end{bmatrix} = \begin{bmatrix} t_A \\ t_B \end{bmatrix} \mod q,$$

$$s_1 \in \mathcal{R}_q^{m_1},\ s_2 \in \mathcal{R}_q^{m_2},\ m \in \mathcal{R}_q^{\ell}$$
$$A_1 \in \mathcal{R}_q^{n \times m_1},\ A_2 \in \mathcal{R}_q^{n \times m_2},\ B \in \mathcal{R}_q^{\ell \times m_2}$$

As in the original Ajtai and BDLOP commitment schemes, the entries for matrices A_1, A_2, and B are randomly selected, and s_2 is a random vector with a small norm. s_1 will hold messages of high dimension but small coefficients, and m will hold messages of low dimension but large coefficients modulo q. It is also important to connect s_1 and m, satisfying:

$$R_1 s_1 + R_m m = u \mod q,$$
$$R_1 \in \mathcal{R}_q^{N \times m_1},\ R_m \in \mathcal{R}_q^{N \times \ell}$$

for public R_1, R_m, and u.

We will now show an example of this ABDLOP commitment scheme in a four-dimensional lattice. As $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$, the elements of each matrix and vector will be polynomials in $\mathcal{R}_q$. For simplicity, let N be small, so we choose $N = 5$. Similarly, we will also choose m_1, m_2, and ℓ to be $3, 3$, and 2 respectively, and choose $q = 7$.

To begin we select random $A_1 \in \mathcal{R}_7^{4\times 3},\ A_2 \in \mathcal{R}_7^{4\times 3},\ B \in \mathcal{R}_7^{2\times 3}$ below:

$$A_1 = \begin{bmatrix} 3 + 2X + 6X^2 & 1 + 5X^4 & 4X + X^2 \\ 6 + X & 2X^3 + 3 & 5 + 6X + X^4 \\ 1 + 4X^2 + 2X^3 & 3X + 2X^4 & 6 + 6X^2 \\ 2 + 5X & 1 + X^2 + 3X^3 & 4 + 2X^4 \end{bmatrix}$$
$$A_2 = \begin{bmatrix} 5 + X^2 & 2 + 3X + 6X^4 & X + 2X^3 + 6X^4 \\ 1 + 4X & 3X^2 + 5X^3 & 1 + 4X \\ 6 + 2X^3 & 1 + X + X^4 & 3 + 2X^4 \\ 3X + 4X^2 & 5 + 6X^3 & 1 \end{bmatrix}$$
$$B = \begin{bmatrix} 2 + X + 3X^4 & 6 + 2X^2 & 1 + 5X^3 \\ 4X + 6X^2 & 3 + X^3 & 2 + 2X^4 \end{bmatrix}$$

Let us now select $s_1 \in \mathcal{R}_7^3, s_2 \in \mathcal{R}_7^3$, and $m \in \mathcal{R}_7^2$ such that s_1 and s_2 have small magnitudes, and m has a low dimensionality but can carry larger entries.

$$s_1 = (1 + X, 6, X^2), \qquad s_2 = (X, 1 + 6X^4),$$
$$m = (1 + X + X^2 + X^3 + X^4, 3 + 2X^2 + 4X^3)$$

Now that we have assembled our pieces, we can put them together to form the first part of the commitment.

$$\begin{bmatrix} A_1 \\ 0 \end{bmatrix} s_1 = \begin{bmatrix} 3+2X+6X^2 & 1+5X^4 & 4X+X^2 \\ 6+X & 2X^3+3 & 5+6X+X^4 \\ 1+4X^2+2X^3 & 3X+2X^4 & 6+6X^2 \\ 2+5X & 1+X^2+3X^3 & 4+2X^4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} (1+X, 6, X^2)$$

$$= \begin{bmatrix} 2+5X+X^2+3X^3+3X^4 \\ 3+X+6X^2+4X^3 \\ 1+5X+3X^2+6X^3+6X^4 \\ 1+2X+X^2+4X^3 \\ 0 \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} A_2 \\ B \end{bmatrix} s_2 = \begin{bmatrix} 5+X^2 & 2+3X+6X^4 & X+2X^3+6X^4 \\ 1+4X & 3X^2+5X^3 & 1+4X \\ 6+2X^3 & 1+X+X^4 & 3+2X^4 \\ 3X+4X^2 & 5+6X^3 & 1 \\ 2+X+3X^4 & 6+2X^2 & 1+5X^3 \\ 4X+6X^2 & 3+X^3 & 2+2X^4 \end{bmatrix} (X, 1+6X^4)$$

$$= \begin{bmatrix} 6+X+2X^3+4X^4 \\ 5X+2X^2+5X^3 \\ 6X^3+2X^4 \\ 5+4X^2+3X^3+2X^4 \\ 2+3X^2+X^4 \\ 3+3X^2+4X^4 \end{bmatrix},$$

$$
\begin{aligned}
&\begin{bmatrix} A_1 \\ 0 \end{bmatrix} s_1 + \begin{bmatrix} A_2 \\ B \end{bmatrix} s_2 + \begin{bmatrix} 0 \\ m \end{bmatrix} \\
&\qquad = \begin{bmatrix} 2 + 5X + X^2 + 3X^3 + 3X^4 \\ 3 + X + 6X^2 + 4X^3 \\ 1 + 5X + 3X^2 + 6X^3 + 6X^4 \\ 1 + 2X + X^2 + 4X^3 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 6 + X + 2X^3 + 4X^4 \\ 5X + 2X^2 + 5X^3 \\ 6X^3 + 2X^4 \\ 5 + 4X^2 + 3X^3 + 2X^4 \\ 2 + 3X^2 + X^4 \\ 3 + 3X^2 + 4X^4 \end{bmatrix} \\
&\qquad\qquad + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 + X + X^2 + X^4 \\ 3 + 2X^2 + 4X^3 \end{bmatrix} \\
&\qquad = \begin{bmatrix} 1 + 6X + X^2 + 5X^3 \\ 3 + 6X + X^2 + 2X^3 \\ 1 + 5X + 3X^2 + 5X^3 + X^4 \\ 6 + 2X + 5X^2 + 2X^4 \\ 3 + X + 4X^2 + 2X^4 \\ 6 + 5X^2 + 4X^3 + 4X^4 \end{bmatrix}.
\end{aligned}
$$

This final vector we have calculated is equivalent to $\begin{bmatrix} t_A \\ t_B \end{bmatrix}$, such that:

$$
t_A = \begin{bmatrix} 1 + 6X + X^2 + 5X^3 \\ 3 + 6X + X^2 + 2X^3 \\ 1 + 5X + 3X^2 + 5X^3 + X^4 \\ 6 + 2X + 5X^2 + 2X^4 \end{bmatrix}, \qquad t_B = \begin{bmatrix} 3 + X + 4X^2 + 2X^4 \\ 6 + 5X^2 + 4X^3 + 4X^4 \end{bmatrix}.
$$

With this commitment scheme, one can provide a proof of knowledge of the values s_1, s_2, and m that the prover committed to, by following the steps below:

1) The prover chooses y_1 and y_2 from Gaussian distributions, and calculates and sends to the verifier

$$w = A_1 \cdot y_1 + A_2 \cdot y_2$$

$$v = R_1 \cdot y_1 - R_m \cdot B \cdot y_2$$

2) The verifier randomly chooses c from the challenge space $\mathcal{C} \subset \mathcal{R}_q^\ell$, where $c = t \cdot r$ for $t \in \mathcal{R}_q$ and $r \in \mathcal{R}_q^\ell$ and sends c to the prover.

3) The prover calculates

$$z_1 = c \cdot s_1 + y_1$$

$$z_2 = c \cdot s_2 + y_2$$

If z_1 and z_2 satisfy the rejection sampling criteria, then the prover sends z_1 and z_2 to the verifier. Otherwise, the prover returns to step (1) and restarts the process.

4) The verifier then accepts if the following 3 requirements are satisfied:

a) $||z_1||$ and $||z_2||$ are below a given upper bound
b) $A_1 \cdot z_1 + A_2 \cdot z_2 - c \cdot t_A = w$
c) $R_1 \cdot z_1 + R_m(c \cdot t_B - B \cdot z_2) - c \cdot u = v$

While the proof above is simply a proof of knowledge of the secret components of the commitment scheme, what makes the ABDLOP commitment scheme especially promising is that it can also be used in combination with different verification methods to prove linear relations, quadratic relations, and inner product relations.

References

[1] Dorit Aharonov and Oded Regev. *GapSVP and Gap CVP are in NP*. 2004. URL: https://qipconference.org/2004/presentations/regev.pdf.

[2] M. Ajtai. "Generating hard instances of lattice problems (extended abstract)". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 99–108. ISBN: 0897917855. DOI: 10.1145/237814.237838. URL: https://doi.org/10.1145/237814.237838.

[3] Carsten Baum et al. *More Efficient Commitments from Structured Lattice Assumptions*. Cryptology ePrint Archive, Paper 2016/997. 2016. URL: https://eprint.iacr.org/2016/997.

[4] Amos Fiat and Adi Shamir. "How To Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *Advances in Cryptology — CRYPTO' 86*. Ed. by Andrew M. Odlyzko. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, pp. 186–194. ISBN: 978-3-540-47721-1.

[5] Shafi Goldwasser and Dmitriy Kharchenko. "Proof of Plaintext Knowledge for the Ajtai-Dwork Cryptosystem". In: *Springer-Verlag Berlin Heidelberg* (2005).

[6] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof-Systems". In: *SIAM Journals on Computing* (1985).

[7] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An Introduction to Mathematical Cryptography*. 2nd ed. Spring Science+Business Media, 2008.

[8] A.K. Lenstra, H.W. Lenstra, and L. Lovász. "Factoring polynomials with rational coefficients". In: *Mathematische Annalen* (1982).

[9] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plancon. *Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General*. Cryptology ePrint Archive, Paper 2022/284. 2022. URL: https://eprint.iacr.org/2022/284.

[10] Alfred Menezes. *Lecture 7. Module-SIS and Module-LWE*. YouTube. 2025. URL: https://www.youtube.com/watch?v=iCgX3HWTrjM.

[11] Daniele Micciancio. *The SIS Problem and Cryptographic Applications*. 2020. URL: https://simons.berkeley.edu/sites/default/files/docs/14967/sis.pdf.

[12] *Quantum Computing - How it Changes Encryption as We Know It*. 2024. URL: https://it.umd.edu/security-privacy-audit-risk-and-compliance-services-sparcs/topic-week/quantum-computing-how-it-changes-encryption-we-know-it.

[13] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*. STOC '05. Baltimore, MD, USA: Association for Computing Machinery, 2005, pp. 84–93. ISBN: 1581139608. DOI: 10.1145/1060590.1060603. URL: https://doi.org/10.1145/1060590.1060603.

[14] Tjerand Silde and Akira Takahashi. *Zero Knowledge Proofs: Challenges, Applications, and Real-world Deployment*. 2024. URL: https://csrc.nist.gov/csrc/media/presentations/2024/wpec2024-3b1/images-media/wpec2024-3b1-slides-akira-tjerand--ZKP-Overview.pdf.

[15] Joseph H. Silverman. *Abstract Algebra: An Integrated Approach*. American Mathematical Society, 2022.