Learning with Errors

Assume q is some large prime. This is how basically all of these stories start.

This problem gets its security from the closest vector problem. In lattice words, if Bob wants to send a message to Alice, Alice generates a random $m \times n$ basis of a rank n sublattice of $\mathbb{Z}_q^m$, call this basis A. She then generates a secret vector $s \in \mathbb{Z}_q^n$. This vector s represents a linear combination of the basis vectors. (So s is the "coordinates" of a point relative to A.) This linear combination will be $b = As$. Now, if someone got ahold of A and b, it would be super easy to find s (this is gaussian elimination) which is why we now need to perturb the point b by a little bit. We pick an error vector $e \sim \chi^m$ from a "small" distribution, and add it, creating the new point

$$\tilde{b} = b + e = As + e$$

Now, Alice publishes $\tilde{b}$ and A to the world. In other words, she publishes a random basis of her sublattice, and she publishes a point that is ever so slightly off the lattice, and closest to the point b on the lattice. The hardness of the closest vector problem gives us some assurance that nobody, given $\tilde{b}$ and A and a believable computer, can compute s in reasonable time. So, now Bob must use $\tilde{b}$ and A to somehow send a message to Alice.

Here is where the wording of the problem shifts from latticey words to linear algebra words. Bob will treat the basis A as a system of equations, where the rows of A represent the coefficients, and the rows of b represent the constant terms on the right-hand side of the equal sign. Since $b = As$, s is a solution of this equation. Of course, Bob doesn't know b, but he does know $\tilde{b}$, and so s is "almost" a solution, in that when we compute As, we get a very small vector that is almost all zeros. Bob is going to create a new equation that also has s as an almost-solution, by selecting a random subset of the rows of A and b to sum. This is often written as randomly selecting a vector $t \in \{0,1\}^m$, and computing $t^T A$ and $t^T b$. s is a solution to all equations (which are the rows of A), and the kernel of a transformation is closed under linear transformation, s is also a "solution" to any linear combination of the rows of A, and therefore also an "almost-solution." After computing a brand-new equation, Bob then adds a large number to the constant term if he wishes to send the bit 1, and doesn't add anything if he wishes to encrypt the bit 0.

Alice recieves, as the ciphertext, the new equation Bob created, along with his altered constant term. Alice plugs in her secret s, and sees if s is an almost-solution to Bob's equation. If it is, then Bob must not have altered the constant term, so Alice reads the ciphertext as being 0. If s is not an almost-solution to Bob's equation, then Bob must have added a large value to the constant term, and so Alice interprets this as an encryption of 1.