Proto-Dilithium as a ZKP

attempt #1: let $R_q = \mathbb{Z}_q[x]/\langle x^n+1\rangle$, S_η denotes the set of polynomials in R_q whose coefficients are $[-\eta, \eta]$.

KeyGen	Commitment	Response(c)
$\vec{s} \xleftarrow{\$} S_\eta^l$	$\vec{y} \xleftarrow{\$} S_{\gamma_1}^l$	$\vec{z} \leftarrow c\cdot\vec{s} + \vec{y}$
$A \xleftarrow{\$} R_q^{k\times l}$	$\vec{w} \leftarrow A\vec{y}$	return $\vec{z}$
$\vec{t} \leftarrow A\vec{s}$	return $\vec{w}$	
$pk \leftarrow (A, \vec{t})$		**Verify($\vec{z}$)**
$sk \leftarrow \vec{s}$	**Challenge**	assert $A\vec{z} == c\cdot\vec{t} + \vec{w}$
return (pk, sk)	$c \leftarrow \mathrm{Ball}(\tau)$	
	return c	

SIS ZKP Attempt #1

for each ZKP we need 3 properties:

Completeness: honest verifier will accept proof from an honest prover

soundness: dishonest prover cannot make honest verifier accept

(honest verifier) **zero knowledge**: the transcript of the proof does not reveal information about the secret

Attempt #1 is <u>not sound</u> because solving $A\vec{z} = c\cdot\vec{t} + \vec{w}$ is easy where there is no constraint on $\vec{z}$ and $A \in R_q^{k\times l}$ is wide $(l \geq k)$.

solution: estimate a bound on $\vec{z}$ then pose a constraint

recall that $\vec{z} \leftarrow c\cdot\vec{s} + \vec{y}$, let $i \in \{1, 2, \cdots, l\}$ denote the index of the polynomial and $j \in \{1, 2, \cdots, n\}$ denote the index of a coefficient, then we know

$$z_i[j] = (c\cdot s_i)[j] + y_i[j]$$

we already know $-\gamma_1 \leq y_i[j] \leq \gamma_1$, it remains to establish bounds on $(c\cdot s_i)[j]$.

$$c = c[0] + c[1]x + \cdots + c[n-1]x^{n-1}$$
$$s_i = s_i[0] + s_i[1]x + \cdots + s_i[n-1]x^{n-1}$$

$$(c\cdot s_i)[j] = \sum_{\substack{0\leq a,b<n \\ a+b\equiv j \bmod n}} c[a]\,s_i[b]$$

※ only partly correct since $x^n \equiv -1 \pmod{x^n+1}$ so some coeffs need to mul with -1

for each of j the summation has n terms. Among them exactly τ of them are non-zero, and each of $s_i[b] \in [-\eta, \eta]$, so we have that $(c \cdot s_i)[j] \in [-\eta\tau, \eta\tau]$, which gives the bound for $\vec{z}$.

KeyGen	Commitment	Response(c)
$\vec{s} \xleftarrow{\$} S_\eta^l$	$\vec{y} \xleftarrow{\$} S_{\gamma_1}^l$	$\vec{z} \leftarrow c \cdot \vec{s} + \vec{y}$
$A \xleftarrow{\$} R_q^{k \times l}$	$\vec{w} \leftarrow A\vec{y}$	return $\vec{z}$
$\vec{t} \leftarrow A\vec{s}$	return $\vec{w}$	
$pk \leftarrow (A, \vec{t})$		**Verify($\vec{z}$)**
$sk \leftarrow \vec{s}$	**Challenge**	assert $A\vec{z} == c \cdot \vec{t} + \vec{w}$
return (pk, sk)	$c \leftarrow Ball(\tau)$	and $\|\vec{z}\|_\infty \le \gamma_1 + \eta\tau$
	return c	

SIS ZKP Attempt #2

Attempt #2 is somewhat sound now. This is because A is uniform random, so finding a $\vec{z} \in R_q^l$ such that $A\vec{z} = (\cdots)$ under the constraint $\|\vec{z}\|_\infty \le \beta$ is equivalent to solving an instance of the inhomogenous Module-SIS problem. $\Rightarrow$ still need to tune parameters!

Attempt #2 is <u>not zero-knowledge</u>: certain values of $\vec{z}$ can leak information of the secret key $\vec{s}$. Consider an extreme example where $z_i[j] = \eta\tau + \gamma_1$, then we know $s_i[j]$ must be $\pm\eta$ where $c[j]$ is ± 1. Less extreme values give less but still substantial amount of information, "<u>until within a bound, then $\vec{z}$ gives no information</u>"

Recall that $z_i[j] = (cs_i)[j] + y_i[j]$, we will focus on a single coeff since the coeffs of s_i and y_i are all iid. Without knowing $z_i[j]$, an adversary only knows that $s_i \xleftarrow{\$} R_{[-\eta, \eta]}$ and $y_i[j] \xleftarrow{\$} [-\gamma_1, \gamma_1]$

if $z_i[j] = \gamma_1 + \eta\tau$, then it must be $(cs_i)[j] = \eta\tau$ and $y_i[j] = \gamma_1$

if $z_i[j] = \gamma_1 + \eta\tau - 1$, then $(cs_i[j]) = \eta\tau$ and $y_i[j] = \gamma_1 - 1$

or $(cs_i[j]) = \eta\tau - 1$ and $y_i[j] = \gamma_1$

... ...

if $z_i[j] = \gamma_1 + \eta\tau - 2\eta\tau$ then $(cs_i[j]) = \eta\tau$ and $y_i[j] = \gamma_1 - 2\eta\tau$

... ...

$(cs_i[j]) = -\eta\tau$ and $y_i[j] = \gamma_1$

intuitively speaking, when $z_i[j] > \gamma_1 - \eta\tau$, some values for $cs_i[j]$ are impossible because for such value, $y_i[j] = z_i[j] - cs_i[j]$ will be outside the allowed range $[-\gamma_1, \gamma_1]$. On the other hand when $z_i[j] = \gamma_1 - \eta\tau$, all values within $[-\eta\tau, \eta\tau]$ are possible for $cs_i[j]$ because all corresponding $y_i[j]$ values fall within the allowed range. The formal notion is expressed as follows:

Lemma if $\|\vec{z}\|_\infty \leq \gamma_1 - \eta\tau$, then the distribution of s_i is identical to the distribution of $s_i \mid c, z_i$

Proof: $P[s_i \mid c, z_i] = \sum_{s'} P[s_i \cap cs_i = s' \mid c, z_i]$

$$= \sum_{s'} P[cs_i = s' \mid c, z_i] \cdot P[s_i \mid c, z_i, cs_i = s'] \quad \cdots (1)$$

observe that:

(a) $P[s_i \mid c, z, cs_i = s'] = P[s_i \mid c, cs_i = s']$ since z_i does not give extra information when c, s' are already given

(b) $$P[s' \mid c, z_i] = \frac{P[s', c, z_i]}{P[c, z_i]} = \frac{P[z_i \mid s', c] \cdot P[c, s']}{P[c, z_i]} = \frac{P[z_i \mid s', c] \cdot P[s' \mid c]}{P[z_i \mid c]}$$ Bayes rule

(c) $P[z_i \mid s', c] = P[y_i] = \frac{1}{2\gamma_1 + 1}$ since $z = s' + y$ and for $z \in [-\gamma + \eta\tau, \gamma - \eta\tau]$ and $s' \in [-\eta\tau, \eta\tau]$, y is always in the allowed range of values ⇒ the main reason that this proof works

(d) $$\begin{aligned} P[z_i \mid c] &= \sum_{s'} P[z_i \cap s' \mid c] \\ &= \sum_{s'} P[s' \mid c] \cdot P[z_i \mid s', c] \\ &= P[y_i] \cdot \sum_{s'} P[s' \mid c] \\ &= P[y_i] \end{aligned}$$ because (c)

Putting (a) ~ (d) together:

$$\begin{aligned} P[s_i \mid c, z_i] &= \sum_{s'} P[s_i \mid s', c] \cdot \frac{P[y_i] \cdot P[s' \mid c]}{P[y_i]} \\ &= \sum_{s'} P[s_i \mid s', c] \cdot P[s' \mid c] \\ &= \sum_{s'} P[s_i, s' \mid c] \\ &= P[s_i \mid c] \\ &= P[s_i] \end{aligned}$$ $s_i \perp c$ are independent

□

now we have a complete, sound, and zk Σ protocol:

KeyGen	Commitment	Response(c)
$\vec{s} \xleftarrow{\$} S_\eta^L$	$\vec{y} \xleftarrow{\$} S_{\gamma_1}^L$	$\vec{z} \leftarrow c\cdot\vec{s} + \vec{y}$
$A \xleftarrow{\$} R_q^{k\times L}$	$\vec{w} \leftarrow A\vec{y}$	if $\|\vec{z}\|_\infty > \gamma_1 - \eta\tau$ then
$\vec{t} \leftarrow A\vec{s}$	return $\vec{w}$	return $\perp$
$pk \leftarrow (A, \vec{t})$		return $\vec{z}$
$sk \leftarrow \vec{s}$	**Challenge**	
return (pk, sk)	$c \leftarrow \mathrm{Ball}(\tau)$	**Verify($\vec{z}$)**
	return c	assert $A\vec{z} == c\cdot\vec{t} + \vec{w}$
		and $\|\vec{z}\|_\infty \leq \gamma_1 - \eta\tau$

Module-SIS ZKP attempt #3

Attempt #3 captures the idea from "Fiat-Shamir w/ abort" (Lyu'09): in the Σ-protocol, if the combination of $\vec{y}$ and c is such that $\vec{z}$ will leak information, then the prover simply refuse to release $\vec{z}$ (in practice prover can make many commits, or verifier can send many challenges).

We can estimate the probability of no abort:

$$P\left[\|\vec{z}\|_\infty \leq \gamma_1 - \eta\tau\right] = \prod_{i,j} P\left[|z_i[j]| \leq \gamma_1 - \eta\tau\right]$$

$$= \prod_{i,j} \frac{\gamma_1 - \eta\tau}{\gamma_1}$$

$$= \left(1 - \frac{\eta\tau}{\gamma_1}\right)^{L\cdot n}$$

Fiat-Shamir w/ Abort

take the Σ-protocol from v3 and apply Fiat-Shamir transformation

(a) replace $c \xleftarrow{\$} \mathrm{Ball}(\tau)$ with $c \leftarrow H(\vec{w}, m)$ where $H: R_q^k \times \mathcal{M} \to \mathrm{Ball}$ is a hash function

(b) output the entire transcript $\sigma = (\vec{w}, c, \vec{z})$ as the signature

(c) verify by checking that $\|\vec{z}\|_\infty \leq \gamma_1 - \eta\tau$ and that

$$A\vec{z} = c\vec{t} + \vec{w}$$

as it is, the signature $\sigma = (\vec{w}, c, \vec{z})$ is quite large. Notice that with an honest signature, $\vec{w} = A\vec{z} - c\vec{t}$, so we can omit $\vec{w}$ in the signature and re-derive it in the verification routine: $\sigma = (c, \vec{z})$, $\hat{\vec{w}} \leftarrow A\vec{z} - c\vec{t}$, assert $c = H(\hat{\vec{w}}, m)$. It's easy to show that producing a forgery is equivalent to breaking pre-image resistance of H.

Here is the signature scheme **Proto-Dilithium**

KeyGen	Sign(sk $=\vec{s}$, m)	Verify(pk, m, σ)
$\vec{s} \xleftarrow{\$} S_\eta^l$	$\vec{y} \xleftarrow{\$} S_{\gamma_1}^l$	$(c, \vec{z}) \leftarrow \sigma$
$A \xleftarrow{\$} R_q^{k \times l}$	$\vec{w} \leftarrow A\vec{y}$	if $\|\vec{z}\|_\infty > \gamma_1 - \eta\tau$ then
$\vec{t} \leftarrow A\vec{s}$	$c \leftarrow H(\vec{w}, m)$	return $\perp$
$sk \leftarrow \vec{s}$	$\vec{z} \leftarrow c \cdot \vec{s} + \vec{y}$	$\hat{\vec{w}} \leftarrow A\vec{z} - c\vec{t}$
$pk \leftarrow (A, \vec{t})$	if $\|\vec{z}\|_\infty > \gamma_1 - \eta\tau$ then	return $[\![H(\hat{\vec{w}}, m) = c]\!]$
return (pk, sk)	return $\perp$	
	$\sigma \leftarrow (c, \vec{z})$	
	return σ	

ZKPv3 + Fiat-Shamir + signature compression

Proto-Dilithium rev 2

the main drawback of this proto-Dilithium is that we want small values for η so $P[\text{abort}]$ is low, but that make the bound $\|\vec{z}\|_\infty \le \gamma_1 - \eta\tau$ more relaxed, which degrades security. The Dilithium team works around it by replacing the Module-SIS problem with Module-LWE.

KeyGen	Sign(sk, m)	Verify(pk, m, σ)
$\vec{s} \xleftarrow{\$} \chi_s^l$	$\vec{y} \xleftarrow{\$} \chi_y^l$	$(\hat{c}, \hat{\vec{z}}) \leftarrow \sigma$
$\vec{e} \xleftarrow{\$} \chi_e^k$	$\vec{w} \leftarrow A\vec{y}$	if $\|\hat{\vec{z}}\|_\infty > \gamma_1 - \eta\tau$:
$A \xleftarrow{\$} R_q^{k\times l}$	$c \leftarrow H(\vec{w}, m)$	return $\perp$
$\vec{t} \leftarrow A\vec{s} + \vec{e}$	$\vec{z} \leftarrow c\vec{s} + \vec{y}$	$\hat{\vec{w}} \leftarrow A\hat{\vec{z}} - c\vec{t}$
$sk \leftarrow \vec{s}$	if $\|\vec{z}\|_\infty > \gamma_1 - \eta\tau$ then	return $[\![H(\hat{\vec{w}}, m) = \hat{c}]\!]$
$pk \leftarrow (A, \vec{t})$	return $\perp$	
return (pk, sk)	return (c, z)	

the naive adaptation above breaks correctness because

$$A\vec{z} - c\vec{t} = A(c\vec{s} + \vec{y}) - c(A\vec{s} + \vec{e}) = \vec{w} - c\vec{e} \ne \vec{w}$$

so directly hashing $A\vec{z} - c\vec{t}$ will not reproduce c.

Observe that $\vec{w}$ is approximately uniformly random in R_q while $\vec{e}$ and thus $c\vec{e}$, has very small norm $\|c\vec{e}\|_\infty \le \eta\tau$, so we can hash the high-order bits of $\vec{w}$: HighBits is such that

$$\text{HighBits}(\vec{w}) = \text{HighBits}(\vec{w} - c\vec{e}) = \text{HighBits}(A\vec{z} - c\vec{t})$$

KeyGen	Sign(sk, m)	Verify(pk, m, σ)
$\vec{s} \xleftarrow{\$} \chi_s^l$	$\vec{y} \xleftarrow{\$} \chi_y^l$	$(\hat{c}, \hat{\vec{z}}) \leftarrow \sigma$
$\vec{e} \xleftarrow{\$} \chi_e^k$	$\vec{w}_1 \leftarrow \text{HighBits}(A\vec{y})$	if $\|\hat{\vec{z}}\|_\infty > \gamma_1 - \eta\tau$:
$A \xleftarrow{\$} R_q^{k\times l}$	$c \leftarrow H(\vec{w}, m)$	return $\perp$
$\vec{t} \leftarrow A\vec{s} + \vec{e}$	$\vec{z} \leftarrow c\vec{s} + \vec{y}$	$\hat{\vec{w}} \leftarrow \text{HighBits}(A\hat{\vec{z}} - c\vec{t})$
$sk \leftarrow \vec{s}$	if $\|\vec{z}\|_\infty > \gamma_1 - \eta\tau$ then	return $[\![H(\hat{\vec{w}}, m) = \hat{c}]\!]$
$pk \leftarrow (A, \vec{t})$	return $\perp$	
return (pk, sk)	return (c, z)	

HighBits works by dividing $\mathbb{Z}_q$ into rounding intervals each spanning $2\gamma_2$ where γ_2 is a parameter chosen to be a divisor of $q-1$, such as $\gamma_2 = (q-1)/32$). Rounding is computed using Euclidean division where for $i \in \{1, 2, \cdots, k\}$ and $j \in \{1, 2, \cdots, n\}$:

$$w_i[j] = 2\gamma_2 \cdot \text{HighBits} + \text{LowBits}$$

where $\text{LowBits} \in \{-\gamma_2, \cdots, \gamma_2\}$

This explanation is not satisfying!

Given an honest message-signature pair $(m, \sigma = (c, \vec{z}))$ one can recover the low order bits of $A\vec{z} - c\vec{t} = \vec{w} - c\vec{e}$. Since $c\vec{e}$ has small norm $\eta\tau \ll \gamma_2$, we know

$$\text{Low}(A\vec{z} - c\vec{t}) = \text{Low}(\vec{w} - c\vec{e}) = \text{Low}(\vec{w}) - c\vec{e}$$

and from the definition of LowBits we know $|\text{Low}(\vec{w})| \leq \gamma_2$, so we run into the same problem as in ZKPv2, where $A\vec{z} - c\vec{t}$ may leak information about $\vec{e}$. This also means that we can apply the same "abort if $\|A\vec{z} - c\vec{t}\|_\infty$ gets too big" fix:

KeyGen	Sign(sk, m)	Verify(pk, m, σ)
$\vec{s} \xleftarrow{\$} \chi_s^l$	$\vec{y} \xleftarrow{\$} \chi_y^l$	$(\hat{c}, \hat{\vec{z}}) \leftarrow \sigma$
$\vec{e} \xleftarrow{\$} \chi_e^k$	$\vec{w}_1 \leftarrow \text{HighBits}(A\vec{y}, 2\gamma_2)$	if $\|\hat{\vec{z}}\|_\infty > \gamma_1 - \eta\tau$ then
$A \xleftarrow{\$} R_q^{k \times l}$	$c \leftarrow H(\vec{w}, m)$	return $\perp$
$\vec{t} \leftarrow A\vec{s} + \vec{e}$	$\vec{z} \leftarrow c\vec{s} + \vec{y}$	$\hat{\vec{w}} \leftarrow \text{HighBits}(A\vec{z} - c\vec{t}, 2\gamma_2)$
$sk \leftarrow \vec{s}$	if $\|\vec{z}\|_\infty > \gamma_1 - \eta\tau$ then	return $[\![H(\hat{\vec{w}}, m) = \hat{c}]\!]$
$pk \leftarrow (A, \vec{t})$	return $\perp$	
return (pk, sk)	if $\|\text{LowBits}(A\vec{y} - c\vec{e})\|_\infty > \gamma_2 - \eta\tau$ then	
	return $\perp$	
	return $(c, \vec{z})$	

Dilithium w/o pk compression