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)	$c \leftarrow Ball(\tau)$	
	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 as 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 $\vec{z}$.

<u>KeyGen</u>

$\vec{s} \xleftarrow{\$} S_\eta^l$
$A \xleftarrow{\$} R_q^{k \times l}$
$\vec{t} \leftarrow A\vec{s}$
$pk \leftarrow (A, \vec{t})$
$sk \leftarrow \vec{s}$
return (pk, sk)

<u>Commitment</u>

$\vec{y} \xleftarrow{\$} S_{\gamma_1}^l$
$\vec{w} \leftarrow A\vec{y}$
return $\vec{w}$

<u>Challenge</u>

$c \leftarrow \text{Ball}(\tau)$
return c

<u>Response(c)</u>

$\vec{z} \leftarrow c \cdot \vec{s} + \vec{y}$
return $\vec{z}$

<u>Verify($\vec{z}$)</u>

assert $A\vec{z} == c \cdot \vec{t} + \vec{w}$
and $\|\vec{z}\|_\infty \leq \gamma_1 + \eta\tau$

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 $\|\vec{z}\|_\infty \leq \beta$ 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 $\|\vec{z}\|_\infty \le \gamma_1 - \eta\tau$, then the distribution of s_i is identical to the distribution of $s_i | C, z_i$

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

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

observe that:

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

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

(c) $P[z_i | s', c] = P[y_i] = \frac{1}{2\gamma_1 + 1}$ 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 $\Rightarrow$ the main reason that this proof works

(d) $$P[z_i | c] = \sum_{s'} P[z_i \cap s' | c] = \sum_{s'} P[s' | c] \cdot P[z_i | s', c] = P[y_i] \cdot \sum_{s'} P[s' | c] = P[y_i]$$ because (c)

Putting (a) ~ (d) together:

$$P[s_i | C, z_i] = \sum_{s'} P[s_i | s', c] \cdot \frac{P[y_i] \cdot P[s' | c]}{P[y_i]}$$

$$= \sum_{s'} P[s_i | s', c] \cdot P[s' | c]$$

$$= \sum_{s'} P[s_i, s' | c]$$

$$= P[s_i | c] \quad \} \; s_i \perp c \text{ are independent}$$

$$= P[s_i]$$

$\square$

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

KeyGen

$\vec{s} \xleftarrow{\$} S_\eta^L$
$A \xleftarrow{\$} R_q^{k\times L}$
$\vec{t} \leftarrow A\vec{s}$
$pk \leftarrow (A, \vec{t})$
$sk \leftarrow \vec{s}$
return $(pk.sk)$

Commitment

$\vec{y} \xleftarrow{\$} S_{\gamma_1}^L$
$\vec{w} \leftarrow A\vec{y}$
return $\vec{w}$

Challenge

$c \leftarrow \mathrm{Ball}(\tau)$
return c

Response(c)

$\vec{z} \leftarrow c\cdot\vec{s} + \vec{y}$
if $\|\vec{z}\|_\infty > \gamma_1 - \eta\tau$ then
  return $\perp$
return $\vec{z}$

Verify$(\vec{z})$

assert $A\vec{z} == c\cdot\vec{t} + \vec{w}$
  and $\|\vec{z}\|_\infty \le \gamma_1 - \eta\tau$

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:

$$P\left[\|\vec{z}\|_\infty \le \gamma_1 - \eta\tau\right] = \prod_{i,j} P\left[|z_i[j]| \le \gamma_1 - \eta\tau\right]$$
$$= \prod_{i,j} \frac{\gamma_1 - \eta\tau}{\gamma_1}$$
$$= \left(1 - \frac{\eta\tau}{\gamma_1}\right)^{L\cdot n}$$

Proto-Dilithium

parameters: $R_q = \mathbb{Z}_q[x]/\langle x^n+1\rangle$, $S_\eta = R_{[-\eta,\eta]}$, $l \geq k$, $\mathrm{Ball}(\tau)$, γ_1

KeyGen	Sign(sk $=\vec{s}$, m)	Verify(pk, m, σ)
$\vec{s} \xleftarrow{\$} S_\eta^l$	$\vec{y} \xleftarrow{\$} S_{\gamma_1}^l$	$(\vec{w}, \vec{z}) \leftarrow \sigma$
$A \xleftarrow{\$} R_q^{k\times l}$	$\vec{w} \leftarrow A\vec{y}$	if $\|\vec{z}\|_\infty > \gamma_1 - \eta\tau$ then
$\vec{t} \leftarrow A\vec{s}$	$c \leftarrow H(\vec{w}, m)$	return 0
$sk \leftarrow \vec{s}$	$\vec{z} \leftarrow c\cdot\vec{s} + \vec{y}$	$\hat{c} \leftarrow H(\vec{w}, m)$
$pk \leftarrow (A, \vec{t})$	if $\|\vec{z}\|_\infty > \gamma_1 - \eta\tau$ then	if $A\vec{z} = \hat{c}\cdot\vec{t} + \vec{w}$ then
return (pk, sk)	return $\perp$	return 1
	$\sigma \leftarrow (\vec{w}, \vec{z})$	return 0
	return σ	

proto-Dilithium derived from applying Fiat-Shamir to the ZKP in attempt #3

problem: signing routine might fail, the probability of success:

$$P[\mathrm{Sign} \neq \perp] = \left(1 - \frac{\eta\tau}{\gamma_1}\right)^{ln}$$

to decrease the probability of abort we want:

- a smaller value for η, but a smaller value for η means the bound for $\|\vec{z}\|_\infty \leq \gamma_1 - \eta\tau$ is relaxed, which makes the underlying Module-SIS problem easier to solve $\Rightarrow$ less security

solution: use LWE, where $A \xleftarrow{\$} R_q^{k\times l}$ but $\vec{t} \leftarrow A\vec{s} + \vec{e}$. Commit is still $\vec{w} \leftarrow A\vec{y}$ and response is still $\vec{z} \leftarrow c\cdot\vec{s} + \vec{y}$.
Verification will need tweaking:

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

e is secret!

the verifier instead checks whether $\|A\vec{z} - c\cdot\vec{t} - \vec{w}\|_\infty$ is "small enough"