# 1 IBSAS LATTICE

Author: Hideharu Kojima[1]

# 2 algorithms

Algorithms Setup, KeyDerivation, Signing, Verification are based on Crystals-Dilithium[1][2]. Regular font letters represent elements in $R$ or $R_q$. These include elements in $\mathbb{Z}$ and $\mathbb{Z}_q$. Bold upper case letters represent matrices. Bold lower case letters represent column vectors with coefficients. Parameters $\rho$, $\eta$, $\gamma_1$, and $\gamma_2$ are as same as Dilithium. `Uniform`, `ExpandA`, `CRH`, `ExpandMask`, `HighBits`, and `Decompose` are also the same as Dilithium. In Algorithm KeyDerivation, `invmod` derives a reverse element of $|\mathbf{A}|$. And `adjugate` calculates an adjugate matrix of $\mathbf{A}$. Then, $\mathbf{sk}_{id1}$ is derived from the execution at the line 7.

**Algorithm 1** Setup

**function** Setup<br>
  $\rho \leftarrow \{0,1\}^{256}$, $K \leftarrow \{0,1\}^{256}$<br>
  $(\mathbf{s}_1, \mathbf{s}_2) \leftarrow S_\eta^k \times S_\eta^k$<br>
  $\mathbf{A} \in R_q^{k\times k} := \mathtt{ExpandA}(\rho)$<br>
  $\mathbf{t} := \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$<br>
  $tr \in \{0,1\}^{384} := \mathtt{CRH}(\rho || \mathbf{t})$<br>
  **return** $(mpk = (\rho, \mathbf{t}), msk = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t})$<br>
**end function**

**Algorithm 2** KeyDerivation

**Require:** $msk = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t})$, $ID$<br>
1: **function** KeyDerivation<br>
2: $\mathbf{t}_{id} := \mathbf{t} \cdot Hash(ID)$<br>
3: $\mathbf{sk}_{id2} := \mathbf{s}_2 \cdot \mathtt{Uniform}(Hash(ID))_\eta$<br>
4: $\mathbf{A} \in R_q^{k\times k} := \mathtt{ExpandA}(\rho)$<br>
5: $inv_{det} := \mathtt{invmod}_q(|\mathbf{A}|)$<br>
6: $\tilde{\mathbf{A}} := \mathtt{adjugate}_q(\mathbf{A})$<br>
7: $\mathbf{sk}_{id1} := inv_{det}\tilde{\mathbf{A}}(\mathbf{t}_{id} - \mathbf{sk}_{id2})$<br>
8: $tr_{id} \in \{0,1\}^{384} := \mathtt{CRH}(\rho || \mathbf{t}_{id})$<br>
9: **return** $sk_{id} = (\rho, K, tr_{id}, \mathbf{sk}_{id1}, \mathbf{sk}_{id2})$<br>
10: **end function**

# References

[1] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," IACR Transactions on Cryptographic Hardware and Embedded Systems, pp.238–268, 2018.

[1] Faculty of Information Science and Technology, Osaka Institute of Technology

[2] Cryatals-Dilithium `https://github.com/pq-crystals/dilithium`

**Algorithm 3** Signing

**Require:** $sk_{id}, M, \sigma'$

**function** SIGNING
  Parse $sk_{id}$=$(\rho, K, tr_{id}$, $\mathbf{sk}_{id1}$, $\mathbf{sk}_{id2})$
  $\mathbf{A} \in R_q^{k\times k}$ := $\texttt{ExpandA}(\rho)$
  $\mu \in \{0,1\}^{384}$ := $\texttt{CRH(}tr_{id}||M\texttt{)}$
  $\kappa := 0$, $\mathbf{z} := \perp$
  **while** $\mathbf{z} = \perp$ **do**
    $\mathbf{y} \in S_{\gamma_1-1}^{k}$ := $\texttt{ExpandMask}(K||\mu||\kappa)$
    $\mathbf{w} := \mathbf{Ay}$
    $\mathbf{w}^1 := \texttt{HighBits}_q(\mathbf{w}, 2\gamma_2)$
    $c \in B_{60} := \texttt{H(}\mu||\mathbf{w}^1\texttt{)}$
    $\mathbf{z} := \mathbf{y} + c\mathbf{sk}_{id1}$
    $(\mathbf{r}^1, \mathbf{r}^0) := \texttt{Decompose}_q(\mathbf{w} - c\mathbf{sk}_{id2}, 2\gamma_2)$
    **if** $||\mathbf{r}^0||_\infty \geq \gamma_2 - \beta$ or $\mathbf{r}^1 \neq \mathbf{w}^1$ **then** $\mathbf{z} := \perp$
    **end if**
    $\kappa := \kappa + 1$
  **end while**
  Parse $\sigma' = (c', \mathbf{z}', w'=\{\mathbf{w}_1, ...\})$
  **if** $id = 1$ **then**
    $\sigma' := (c' = 0, \mathbf{z}' = \mathbf{0}, w' = \emptyset)$
  **end if**
  $c_{agg} := c' + c$, $\mathbf{z}_{agg} := \mathbf{z}' + \mathbf{z}$, $w_{agg} := w' \bigcup \{\mathbf{w}\}$
  **return** $\sigma = (c_{agg}, \mathbf{z}_{agg}, w_{agg})$
**end function**

**Algorithm 4** Verification

**Require:** $\sigma, mpk$, List $((id_1, m_1), ..., (id_n, m_n))$ of ID Info. and Message

**function** VERIFICATION
  Parse $mpk = (\rho, \mathbf{t})$
  Parse $\sigma := (c_{agg}, \mathbf{z}_{agg}, w_{agg} = \{\mathbf{w}_1, ..., \mathbf{w}_N\})$
  $c_N := 0$, $\mathbf{w}_N := \mathbf{0}$, $\mathbf{ct}_N$
  **for** $i := 1$; $i \leq n$; $i$++ **do**
    $\mathbf{t}_{id} := \mathbf{t} \cdot Hash(id_i)$
    $tr := \texttt{CRH}(\rho||\mathbf{t}_{id})$
    $\mu := \texttt{CRH}(tr||m_i)$
    $\mathbf{w}_i^1 := \texttt{HighBits}_q(\mathbf{w}_i, 2\gamma_2)$
    $c := \texttt{H}(\mu|| \mathbf{w}_i^1)$
    $c_N := c_N + c$
    $\mathbf{w}_N := \mathbf{w}_N + \mathbf{w}_i$
    $\mathbf{ct}_N := \mathbf{ct}_N + c\mathbf{t}_{id}$
  **end for**
  $\mathbf{A} \in R_q^{k\times k}$ := $\texttt{ExpandA}(\rho)$
  $\mathbf{Az}^1 := \texttt{HighBits}_q(\mathbf{Az}_{agg} - \mathbf{ct}_N, 2\gamma_2)$
  $\mathbf{w}_N^1 := \texttt{HighBits}_q(\mathbf{w}_N, 2\gamma_2)$
  **return** $c_N = c_{agg}$ && $\mathbf{w}_N^1 = \mathbf{Az}^1$
**end function**