Controller of Certifying Authorities
Ministry of Electronics & Information Technology
Government of India

IEEE COMPUTER SOCIETY Bangalore Chapter

IAS IEEE INDUSTRY APPLICATIONS SOCIETY Linking Research to Practice Bangalore Chapter

5th INTERNATIONAL CONFERENCE ON

PUBLIC KEY INFRASTRUCTURE AND ITS APPLICATIONS (PKIA 2024)

SEPTEMBER 5-6th, 2024

Approach to Post Quantum Cryptography Validation

Smitha G Havanur & Dr. Abey Jacob
CDAC Bangalore

AGENDA

- Introduction
- NIST/FIPS 140-3/ISO 19790
- CMVP/CAVP
- Post Quantum Algorithms
- ML-KEM Crystal Kyber
  - Parameters
  - Auxiliary functions
  - Kyber PKE
  - Kyber ML-KEM
- Test Procedure
  - Kyber 512
  - Validation results
- References

NIST - FIPS Standards & ISO/IEC

- NIST developed the FIPS Publication 140-2 as a security standard that sets forth requirements for cryptographic modules, including hardware, software, and/or firmware for U.S. federal agencies.
- FIPS 140-2 establishes a set of rigorous requirements for the design, implementation, and operation of cryptographic modules to ensure the confidentiality, integrity, and authenticity of data.
- It is a mandatory requirement for the government agencies in USA and Canada to conform with FIPS 140–2. Furthermore, vendor's equipment/devices deployed within the government agencies have to comply with FIPS 140–2.
- FIPS 140-2 ------FIPS 140-3 (2019-2020)
- FIPS 140-3 & ISO 19790 (Global acceptance)
  - (FIPS 140-3 aligns with ISO/IEC 19790:2012(E) and includes modifications of the Annexes that are allowed to CMVP).

FIPS 140-3

- FIPS 140-3 supersedes FIPS 140-2 with updated security requirements for cryptographic modules.
- FIPS 140-3 align with ISO/IEC 19790:2012(E) and include modifications of the Annexes that are allowed by the Cryptographic Module Validation Program (CMVP), as a validation authority.

1. Cryptographic Module Specification
2. Cryptographic Module Ports and Interfaces
3. Roles, Services and Authentication
4. Software/Firmware Security
5. Operational Environment
6. Physical Security
7. Non invasive Security
8. Sensitive Security Parameter (SSP) Management
9. Self Tests
10. Life-Cycle Assurance
11. Mitigation of Other Attacks

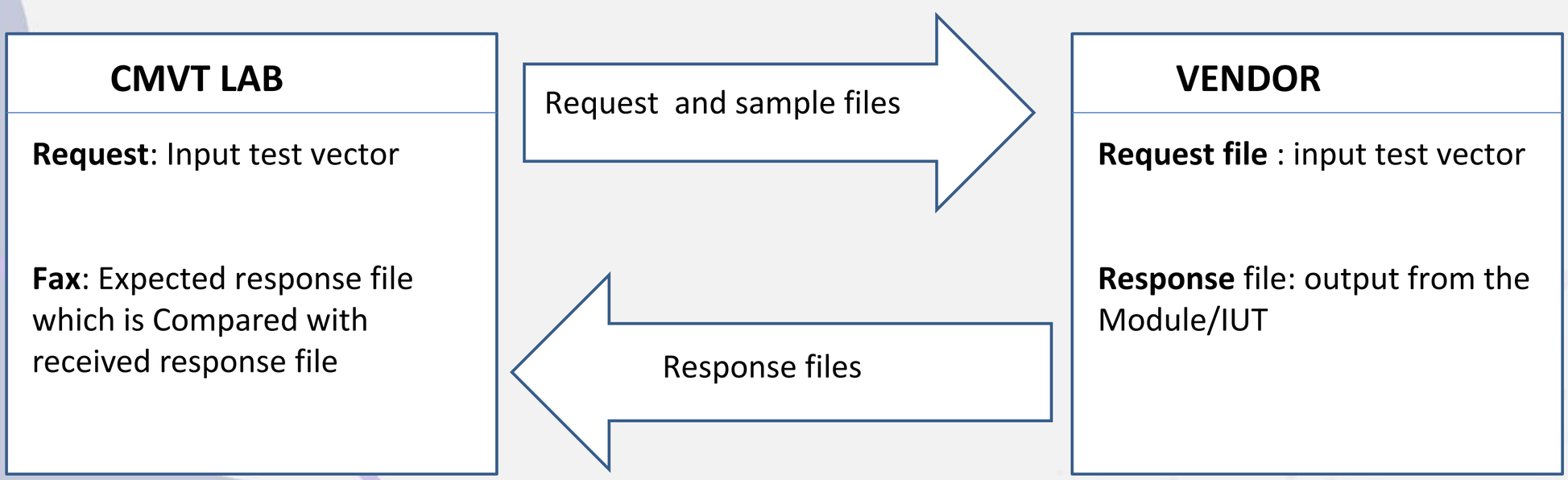
Relevant ISO Standards for Crypto Module Validation

- **ISO/IEC 19790:2012 - Information technology — Security techniques — Security requirements for cryptographic modules**
  - **Specifies security requirements intended to maintain the security provided by a cryptographic module and compliance with this International Standard within a security system protecting sensitive information in computer and telecommunication systems.**

- **ISO/IEC 24759:2017 - Information technology — Security techniques — Test requirements for cryptographic modules**
  - **To be used by testing laboratories for checking conformance of cryptographic modules as per the requirements specified in ISO/IEC 19790:2012.**
  - **The methods are developed to provide a high degree of objectivity during the testing process and to ensure consistency across the testing laboratories.**

Cryptographic Algorithm Validation Program

- **The CMVT lab uses the information supplied by the vendor to generate input vector(s). Three types of files are generated and shared with vendor**
- **The response file is generated by the vendor's algorithm implementation and submitted to CMVT lab for Validation**



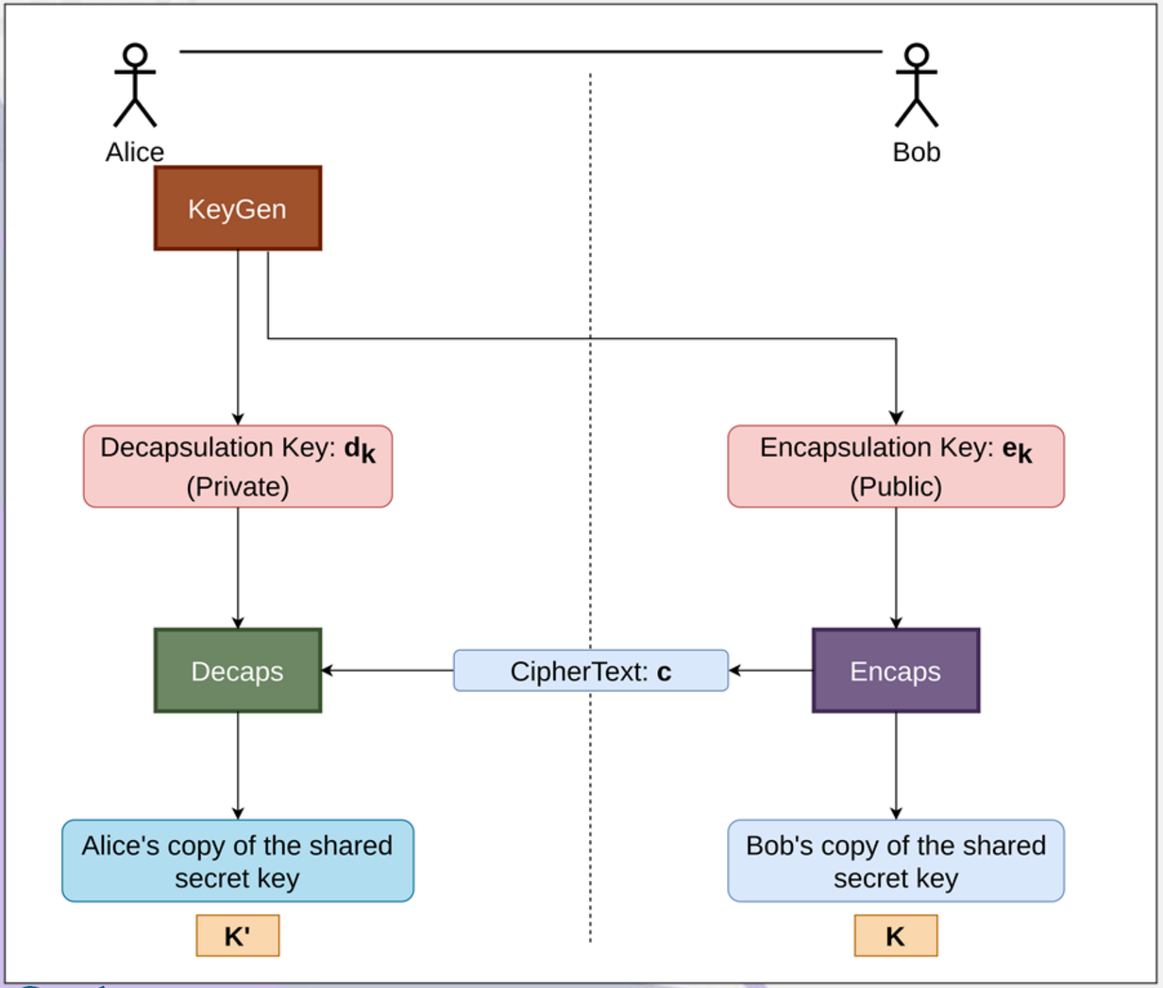
POST QUANTUM Algorithms

After 3 rounds in 2022, four algorithms were selected for standardization and widescale use. These were categorized into finalists and alternative algorithms, with draft standards released in 2023.

NIST published the final version of FIPS 203, 204, and 205, which are standards for encryption and digital signatures, using ML-KEM (based on CRYSTALS-Kyber), ML-DSA (based on CRYSTALS-Dilithium), and SLH-DSA (based on Sphincs+), respectively, with SLH-DSA serving as a backup for ML-DSA in case of vulnerabilities.

ML-KEM, derived from CRYSTALS-KYBER KEM and part of the CRYSTALS (Cryptographic Suite for Algebraic Lattices) suite, uses Module Learning with Errors over polynomial rings Zq[X]/(Xn+1)Z_q[X]/(X^n + 1)Zq[X]/(Xn+1) and for Kyber n=256n = 256n=256 and q=3329q = 3329q=3329. It is standardized in three parameter sets (ML-KEM-512, ML-KEM-768, ML-KEM-1024) aligned with NIST's security levels corresponding to AES-128, AES-192, and AES-256. These algorithms are designed to ensure long-term security even against future quantum computing threats.

Key Encapsulation Mechanism (KEM)



KeyGen:

- Alice runs the KeyGen algorithm.
- This generates two keys: a decapsulation key (private) and an encapsulation key (public).
- Alice keeps the decapsulation key secret and sends the encapsulation key to Bob.

- **Encaps:**
  - Bob receives Alice's encapsulation key.
  - He uses this key with the Encaps algorithm to generate two things:
    a) A ciphertext b) Bob's copy of the shared secret key (K)
- **Ciphertext Transmission:**
  - Bob sends the ciphertext to Alice.
  - The ciphertext is related to the shared secret but doesn't reveal it.
- **Decaps:**
  - Alice receives the ciphertext from Bob.
  - She uses her decapsulation key and the Decaps algorithm to process the ciphertext.
  - This produces Alice's copy of the shared secret key (K')
- **Result:**
  - If everything works correctly, Alice's K' and Bob's K should be identical.
  - They now have a shared secret key without ever directly transmitting it.

Parameter set for CRYSTALS-Kyber algorithm

Table -1 Parameter set for CRYSTALS-Kyber algorithm												
Kyber	NIST Security Level	n	q	k	η_1	η_2	d_u	d_v	Encapsulation key ek (in Bytes)	Decapsulation key dk (in Bytes)	Ciphertext c (in Bytes)	Shared secret key (in Bytes)
512	1(AES128)	256	3329	2	3	2	10	4	800	1632	768	32
768	3(AES192)	256	3329	3	2	2	10	4	1184	2400	1088	32
1024	5(AES256)	256	3329	4	2	2	11	4	1568	3168	1568	32

We define three parameter sets for Kyber, namely Kyber512, Kyber768, and Kyber1024.
Kyber specifies two global integer constants: n = 256 and q = 3329 among different security levels.
k: specifies the sizes of the vectors in PKE key generation, denoted as "s" and "e" and also the size of the matrix Aˆ and the vectors **r,** $\boldsymbol{e_1}$, $\boldsymbol{e_2}$.
$\boldsymbol{\eta_1}$: Specifies the binomial distribution for generating specific vectors "s" and "e" during PKE key generation, as well as the vector "r" in PKE encryption.
$\boldsymbol{\eta_2}$: Specifies the binomial distribution for generating particular vectors "$\boldsymbol{e_1}$" and "$\boldsymbol{e_2}$" in PKE encryption.
$\boldsymbol{d_u}$ and $\boldsymbol{d_v}$: Specifies the inputs for the operations Compress, Decompress, ByteEncode, and ByteDecode utilized in PKE encryption and decryption.

Auxillary Functions CRYSTALS-Kyber algorithm

Kyber employs a pseudo-random function (PRF), an extendable output function (XOF), two hash functions (H and G), and a key-derivation function (KDF). These cryptographic primitives are all implemented using functions from the FIPS-202 SHA-3 Standard.

Functions J and H take a variable length input and produce a 32-byte output.

Function G: This function takes a variable-length input and produces two 32-byte outputs.

eXtendable output function (XOF): The XOF function accepts a 32-byte input as well as two 1-byte inputs, generating an output of variable length.

PRF (Pseudorandom function): This function accepts a parameter $\eta \in \{2, 3\}$, 32-byte input, and a 1-byte input, producing an output of $(64 \bullet \eta)$ bytes.

Centered Binomial Distribution (CBD): The Centered Binomial Distribution (CBD) function in Kyber is utilized to produce the secret vector s and the error (noise) vector e in the form of polynomials.

Auxillary Functions CRYSTALS-Kyber algorithm

NTT (Number-Theoretic Transform): NTT significantly speeds up the multiplication of large polynomial rings, which is needed when doing key computations. Each entry in the matrix ‘A’ and the vectors ‘s’ and ‘e’ is made of a polynomial ring denoted by Zq[X]/(X^n + 1) where q = 3329 and n = 256.

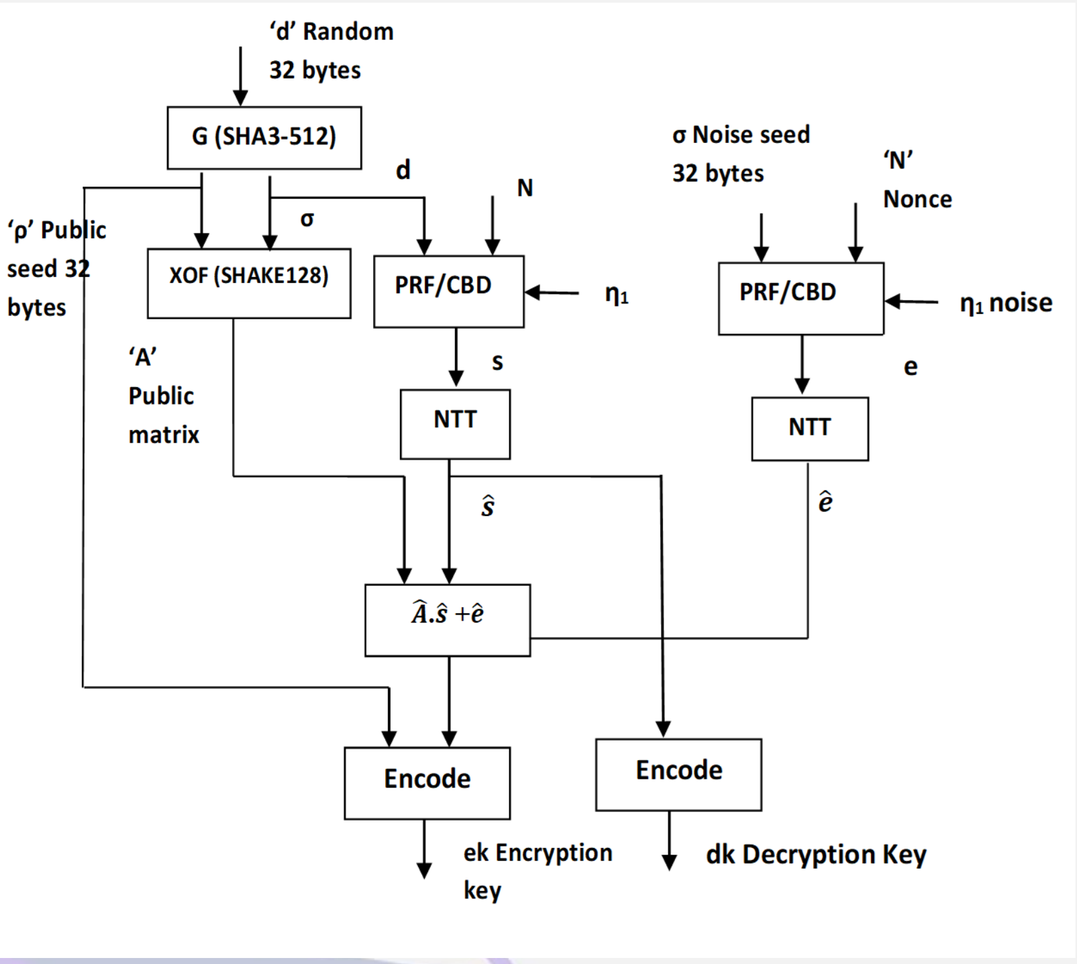
Compression and Decompression: The Compress and Decompress functions are primarily designed to discard certain low-order bits from the public key and ciphertext.

- This method aids in decreasing the overall parameter sizes, thus improving the efficiency of the cryptographic scheme.
- Performing decompression followed by compression retains the original input and compression followed by decompression only slightly alters the value.

Encoding and decoding: The ByteEncode and ByteDecode algorithms are employed for serializing and deserializing arrays of integers under modulo ‘m’.

- Each serialized array has a constant length of n = 256.
- ByteEncode converts an array of integers represented by ‘d’ bits into an array comprising of 32 bytes. Conversely, ByteDecode reverses this process, transforming an array of 32 times ‘d’ bytes back into an array of integers with d bits.

Kyber PKE key-generation algorithm



Algorithm 1:

Uses randomness to generate an encryption key and a corresponding decryption key

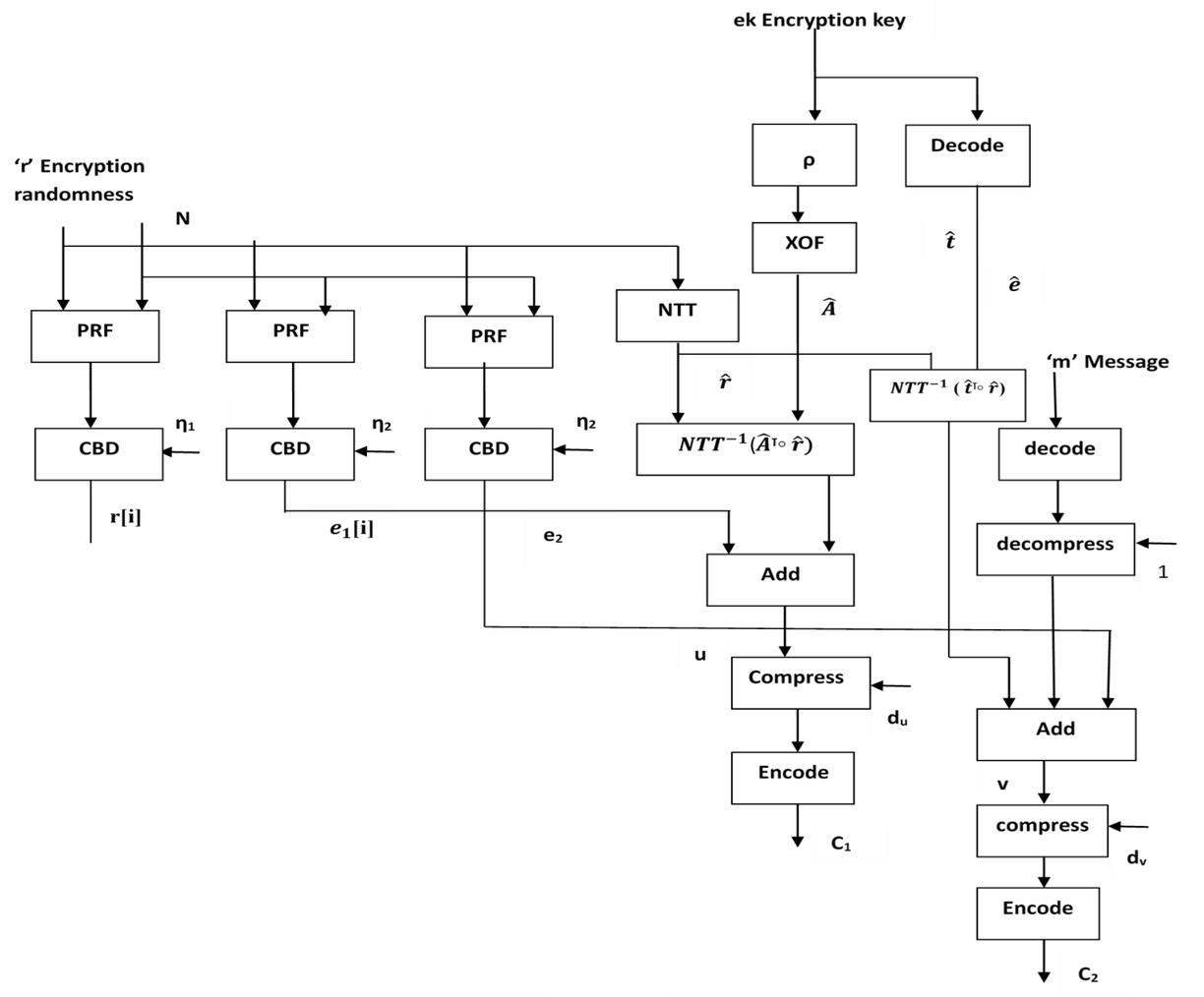
Input: randomness $d \in \mathbb{B}32$.

Outputs: Encryption key $ek_{PKE} \in B^{384k+32}$.

Outputs: Decryption key $dk_{PKE} \in B^{384k}$.

1. N ← 0
2. $\hat{t} \leftarrow \text{ByteDecode}_{12}(\text{ek}_{PKE}[0 : 384k])$
3. ρ ← ek_{PKE} [384k : 384k + 32]
4. for (i ← 0; i < k; i++)
5. for (j ← 0; j < k; j ++)
6. $\hat{A}[i, j] \leftarrow \text{SampleNTT}(\text{XOF}(\rho, i, j))$
7. end for
8. end for
9. for (i ← 0; i < k; i++)
10. $r[i] \leftarrow \text{SamplePolyCBD}_{\eta_1}(\text{PRF}_{\eta_1}(r, N))$
11. N ← N + 1
12. end for
13. for (i ← 0; i < k; i++)
14. $e_1[i] \leftarrow \text{SamplePolyCBD}_{\eta_2}(\text{PRF}_{\eta_2}(r, N))$
15. N ← N + 1
16. end for
17. $e_2 \leftarrow \text{SamplePolyCBD}_{\eta_2}(\text{PRF}_{\eta_2}(r, N))$
18. $\hat{r} \leftarrow \text{NTT}(r)$
19. $u \leftarrow NTT^{-1}(\hat{A}^{T} \circ \hat{r}) + e_1$
20. $\mu \leftarrow \text{Decompress}_1(\text{ByteDecode}_1(m))$
21. $v \leftarrow NTT^{-1}(\hat{t}^{T} \circ \hat{r}) + e_2 + \mu$
22. $c_1 \leftarrow \text{ByteEncode}_{d_u}(\text{Compress}_{d_u}(u))$
23. $c_2 \leftarrow \text{ByteEncode}_{d_v}(\text{Compress}_{d_v}(v))$
24. return $c \leftarrow (c_1 \| c_2)$

Kyber PKE Encryption Algorithm



Algorithm 2:

Uses the encryption key to encrypt a plaintext message using the randomness r.

Input: encryption key $ek_{PKE} \in B^{384k+32}$.

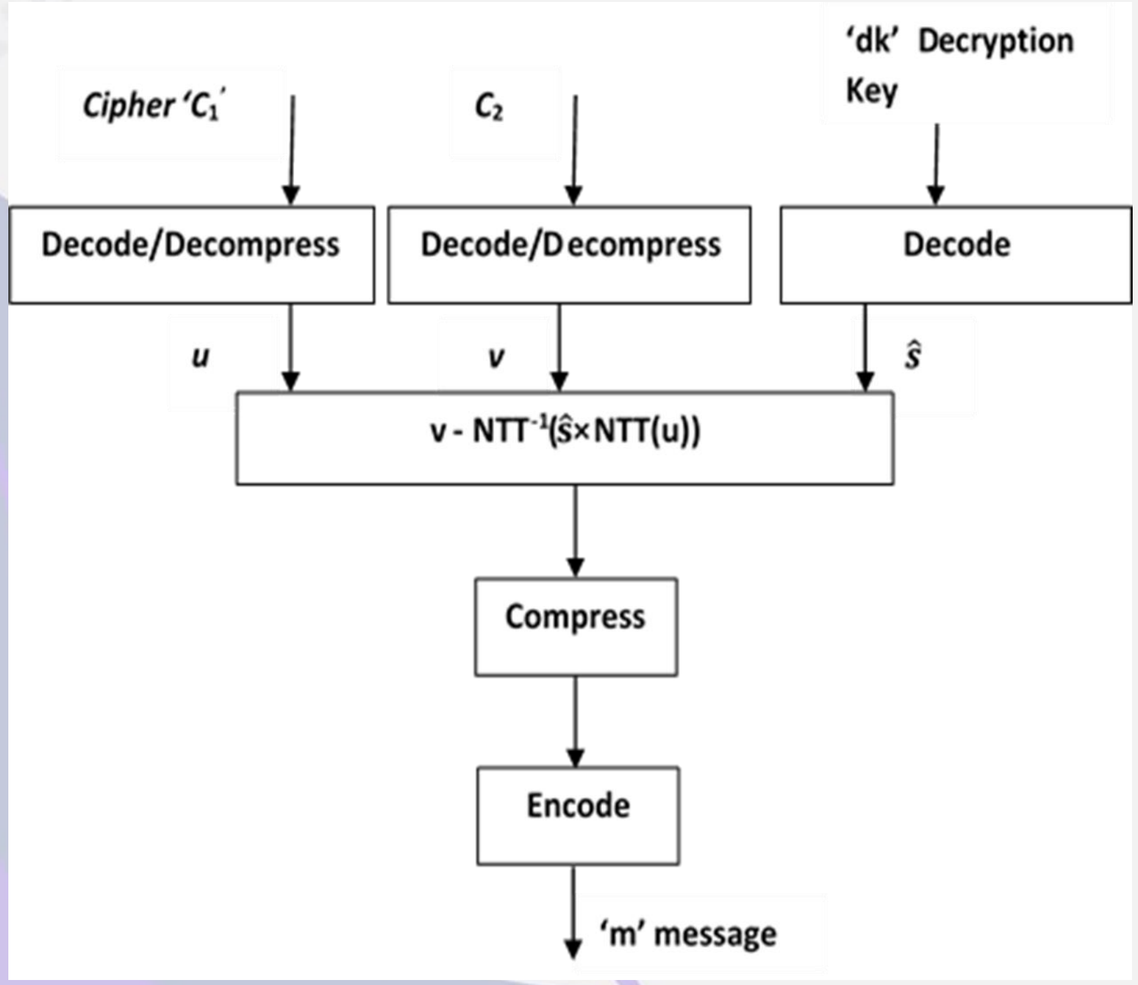
Input: message $m \in B^{32}$.

Input: encryption randomness $r \in B^{32}$.

Output: ciphertext $c \in B^{32(d_u k + d_v)}$

1. N ← 0
2. $\hat{t} \leftarrow ByteDecode_{12}(ek_{PKE}[0 : 384k])$
3. ρ ← ek_{PKE} [384k : 384k + 32]
4. for (i ← 0; i < k; i++)
5. for (j ← 0; j < k; j ++)
6. $\hat{A}$[i, j] ← SampleNTT(XOF(ρ, i, j))
7. end for
8. end for
9. for (i ← 0; i < k; i++)
10. $r[i] \leftarrow SamplePolyCBD_{\eta_1}(PRF_{\eta_1}(r, N))$
11. N ← N + 1
12. end for
13. for (i ← 0; i < k; i++)
14. $e_1[i] \leftarrow SamplePolyCBD_{\eta_2}(PRF_{\eta_2}(r, N))$
15. N ← N + 1
16. end for
17. $e_2 \leftarrow SamplePolyCBD_{\eta_2}(PRF_{\eta_2}(r, N))$
18. $\hat{r} \leftarrow$ NTT(r)
19. u ← $NTT^{-1}(\hat{A}^{T} \circ \hat{r}) + e_1$
20. μ ← $Decompress_1(ByteDecode_1(m))$
21. v ← $NTT^{-1}(\hat{t}^{T} \circ \hat{r}) + e_2 + \mu$
22. $c_1 \leftarrow ByteEncode_{d_u}(Compress_{d_u}(u))$
23. $c_2 \leftarrow ByteEncode_{d_v}(Compress_{d_v}(v))$
24. return c ← ($c_1 \| c_2$)

Kyber PKE-Decryption Algorithm



Algorithm 3:

Uses the decryption key to decrypt a ciphertext

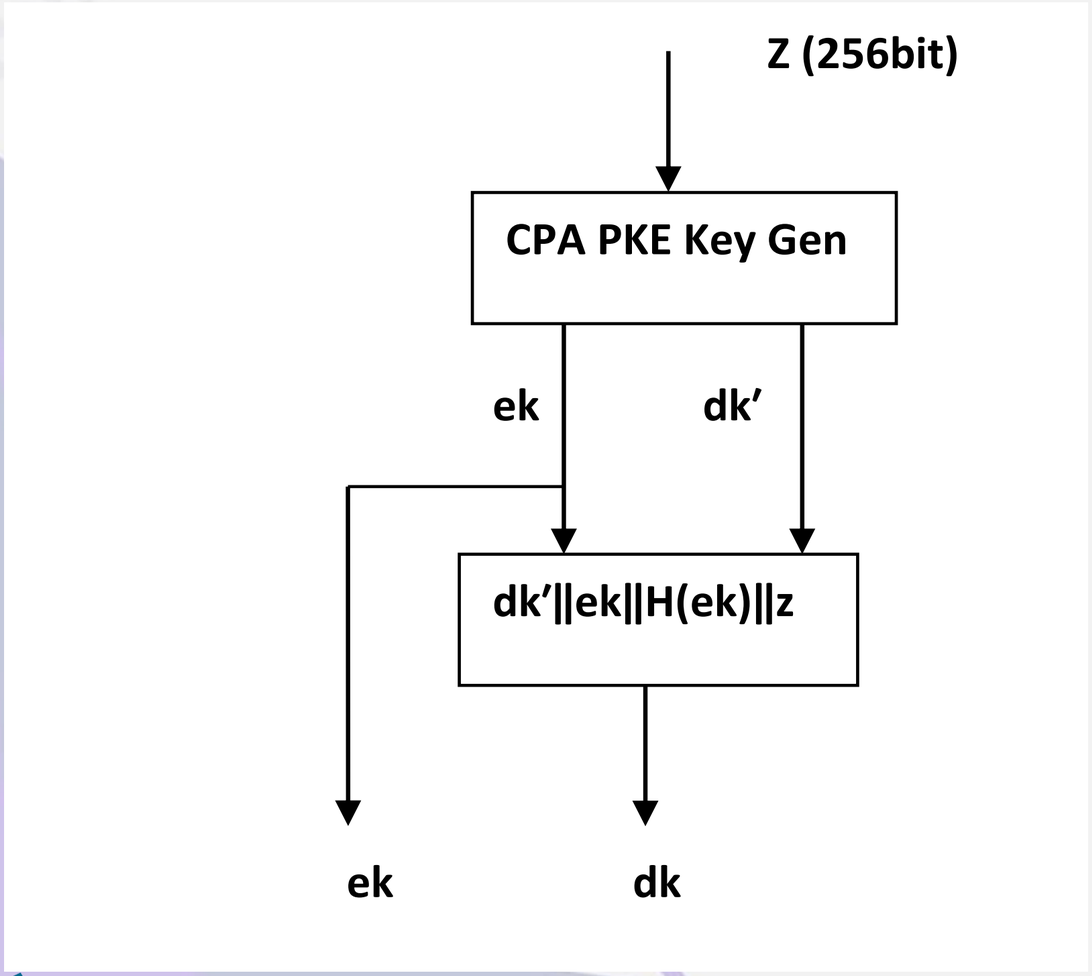
Input: decryption key $\mathbf{dk}_{PKE} \in \mathbf{B}^{384k}$.

Input: ciphertext $c \in \mathbf{B}^{32(d_u k + d_v)}$.

Output: message $m \in \mathbf{B}^{32}$.

1. $c_1 \leftarrow c[0 : 32 d_u k]$
2. $c_2 \leftarrow c[32 d_u k : 32(d_u k + d_v)]$
3. $u \leftarrow \mathbf{Decompress}_{d_u}(\mathbf{ByteDecode}_{d_u}(c_1))$
4. $v \leftarrow \mathbf{Decompress}_{d_v}(\mathbf{ByteDecode}_{d_v}(c_2))$
5. $\hat{s} \leftarrow \mathbf{ByteDecode}_{12}(\mathbf{dk}_{PKE})$
6. $w \leftarrow v - NTT^{-1}(\hat{s}^T \circ NTT(u))$
7. $m \leftarrow ByteEncode_1(Compress_1(w))$
8. return m

Kyber KEM Key generation Algorithm



Algorithm 4:

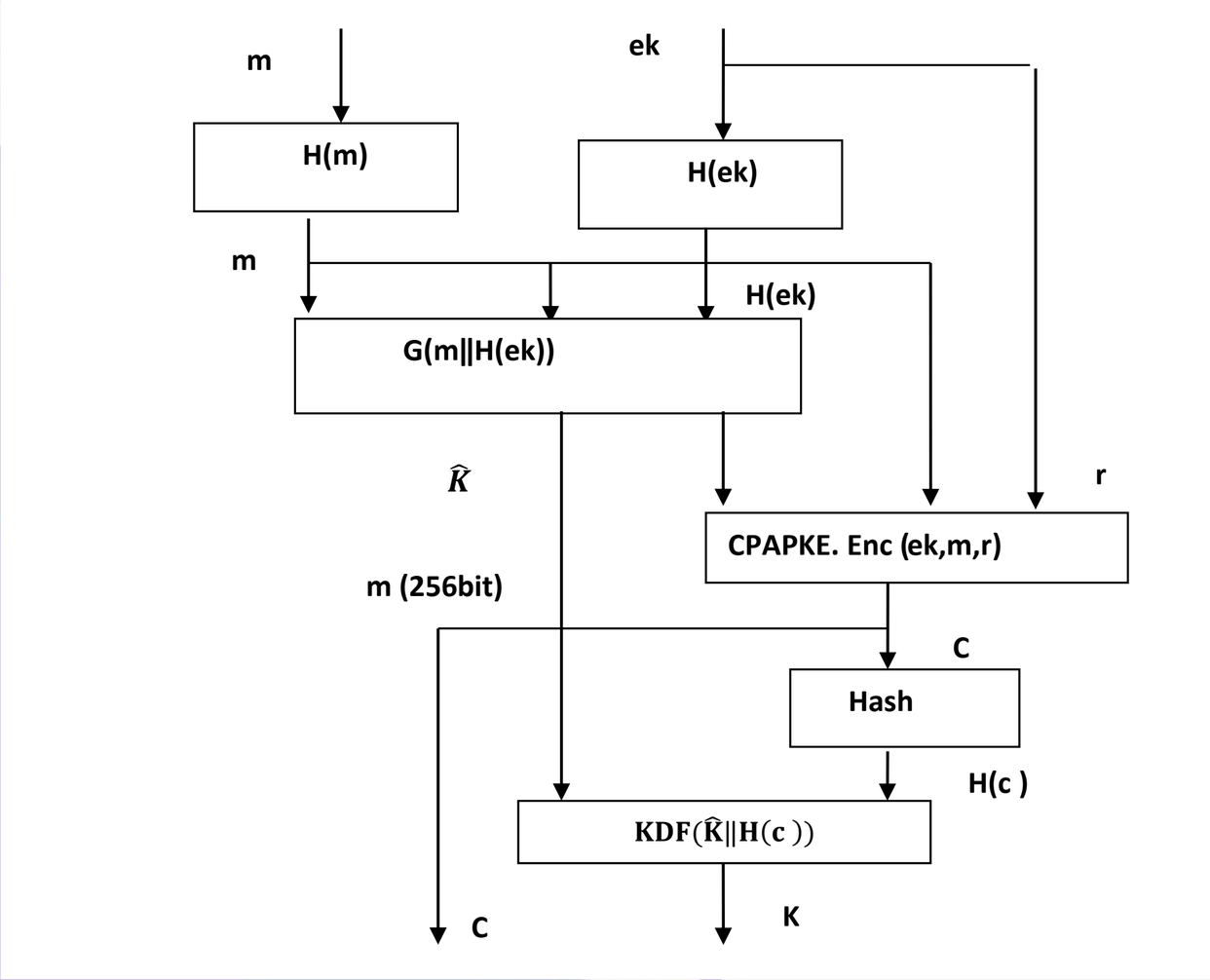
Generates an encapsulation key and a corresponding decapsulation key

Output: Encapsulation key ek ∈ $B^{384k+32}$.

Output: Decapsulation key dk ∈ $B^{768k+96}$.

1. $z \xleftarrow{\$} B^{32}$
2. $(ek_{PKE}, dk_{PKE}) \leftarrow$ PKE.KeyGen()
3. $ek \leftarrow ek_{PKE}$
4. $dk \leftarrow (dk_{PKE} \| ek \| H(ek) \| z)$
5. return (ek, dk)

Kyber KEM Key Encapsulation Algorithm



Algorithm 5:

Uses the encapsulation key to generate a shared secret key and an associated ciphertext.

Input: encapsulation key ek ∈ $B^{384k+32}$.

Output: shared key K∈ B^{32}.

Output: ciphertext c ∈ $B^{32(d_u k+d_v)}$.

1. m ← B^{32}

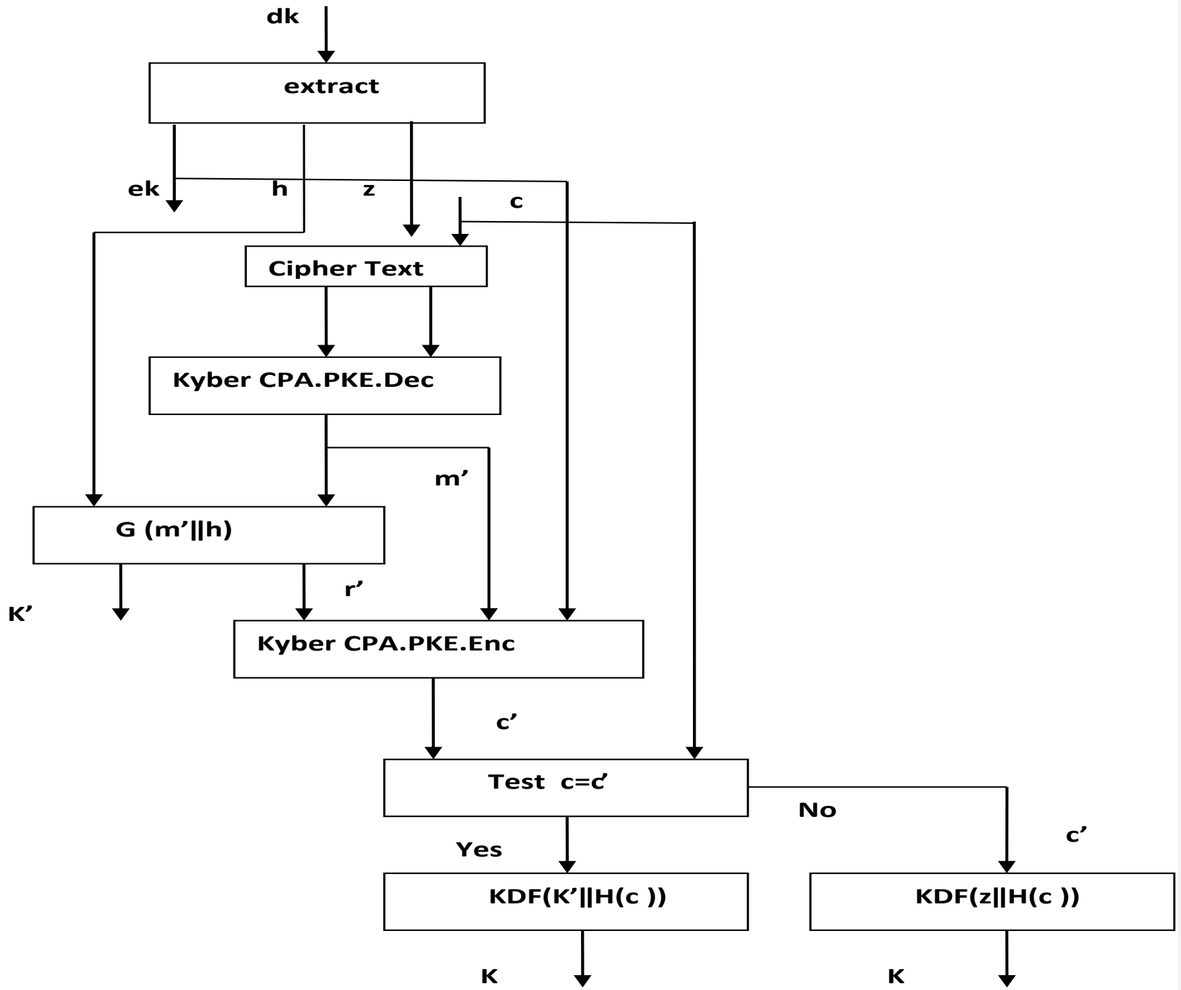
2. (K,r)← G(m‖H(ek))

3. c← PKE.Encrypt(ek, m, r)

4. K ←H(K‖H(c))

5. return (K, c)

Kyber KEM Key Decapsulation Algorithm



Algorithm 6:

Uses the decapsulation key to produce a shared secret key from a ciphertext.

Input: ciphertext c ∈ $\mathbb{B}^{32(d_u k+d_v)}$.

Input: Decapsulation key dk ∈ $B^{768k+96}$.

Output: shared key K∈ B^{32}.

1. $\mathbf{dk}_{PKE}$ ← dk[0 : 384k]
2. $\mathbf{ek}_{PKE}$ ← dk[384k:768k + 32]
3. h ← dk[768k+ 32 : 768k + 64]
4. z ← dk[768k+64:768k+ 96]
5. m′←PKE.Decrypt($\mathbf{dk}_{PKE}$, c)
6. (K′, r′) ← G(m′ ‖h)
7. $\bar{K}$← J(z‖c, 32)
8. c′ ← PKE.Encrypt($\mathbf{ek}_{PKE}$, m′ , r′)

if c ≠ c′ then

K′ ← $\bar{K}$ if cipher texts do not match, “implicitly reject”

end if

return K′

Kyber 512

Seed ‘d’ 32 random bytes
Seed ‘z’ 32 random bytes
→ **Kyber 512 Key Generation** →
‘ek’ Encapsulation key 800 bytes
‘dk’ Decapsulation key 1632 bytes

‘ek’ Encapsulation key 800 bytes
Message ‘m’ 32 random bytes
→ **Kyber 512 Encapsulation** →
Ciphertext ‘c’ 768 bytes
Shared secret key ‘K’ 32 bytes

Ciphertext ‘c’ 768 bytes
‘dk’ Decapsulation key 1632 bytes
→ **Kyber 512 Decapsulation** →
if $c \neq c'$ → “implicitly reject” and K′ ← $\overline{K}$
Return → Shared secret key K′

Shared secret key
‘K’ 32 bytes

VALIDATION TEST FOR KYBER IMPLEMENTATION

- The validation Lab need to provide the results for a sample seed in the request file asking the vendor to supply the resulting pk, sk, ct, and ss.
  - The vendor is supposed to give seed=, pk=, sk=, ct=, ss=
  - On the other hand, all intermediate values of Kyber algorithms, including various sub-functions and cryptographic primitives, need to be validated in our opinion.

- Hash Functions, eXtendable Output Functions (XOFs), and Sampling
  - SHA3-512, SHA3-256 is utilized for hashing
  - SHAKE-128 and SHAKE-256 are used as pseudo random functions.
  - Both uniform and centered binomial distributions are employed for sampling coefficients of polynomials.
  - These need to be validated using SHAVS.

VALIDATION TEST FOR KYBER IMPLEMENTATION

- It is required to test the correctness of implementation of the following modules:
  - TRNG as per NIST SP 800-90A, NIST SP 800-90B,and NIST SP 800-90C.
  - Polynomial Arithmetic functions, Encoding, Decoding and Compression and Decompression.

- ISO 19790 requires side channel resistance is to be assured by the vendor for security levels 3 and 4
  - ISO 17825, which is based on the Test Vector Leakage Assessment (TVLA) methodology.

ML-KEM-512 Key Generation

```
# KAT TEST for Kyber512
# Kyber512 KEYPAIR request file
# Generated on: Thu Aug 22 13:15:48 2024


count = 0
seed = 7f7bd0b678080787b1ea8cf1ea7d2cb658a8caa8869fa96fd249501983a622de1ba1bebe44f430bb012187d2b84045fd6679a11b7cfe6d03a7b568827f71d8df


count = 1
seed = 484992b5823094a6a9b4d888dac2d9505aeb6885e505911ce4db81c98cc82a03b3fb42ad799b6475694aa6ad5df036e87df99b38be1f0cda9de058d971fd8831
```

```
# KAT TEST for Kyber512
# Kyber512 Kwypair response file
# Generated on: Thu Aug 22 13:15:48 2024


count = 0
seed = 7f7bd0b678080787b1ea8cf1ea7d2cb658a8caa8869fa96fd249501983a622de1ba1bebe44f430bb012187d2b84045fd6679a11b7cfe6d03a7b568827f71d8df
pk = 041151032b72f69bb2ff06a682caa922270f121a8ad3311feab75038cc3dcbe7b4abd59b07e191c487b1456804689acf4e9acec7c79f5b3ba2f21c821b7294dfbb7d4f65cc9e1530151560431255a0c45cc4992ac6629b4ec7748cb6ab71c04739ca243403c081141779461c9cd97
97c3c7ec553157ce69d09570ba67659ad561dd9921b88b3ce17d18633dcbb3cbc64de2b244c877cd3089b5514627532695b734259f8a6e0e1393d03a6676c8c1f2170f71314dd12a2d337a7dd0152dd72bbf68cc7534a02de8ab09220c0efac8f652a7c4441c8faaa9ed550197290509b5
3c1415c4e3289b098457524179bc6f50637bcc3df66b01d45b7ca74be572c5129d4940568357ce42687d463015a3f8f81a557a5a3063bbf05a85acd591a92e722d09b20c559313b5c3449f314fda59e0538a088b4142b8440852b047b19139ad3ccf7b172308444f1442e7e304b2d625b6
06175ac53286b137e64d24e6689356648a4fd27a4fb0881588b1a2ddb94b7825a2e61a56cb031d755b8c126ceffa3c2ae97393b3b3b7887646bd171b48c3eeac59ad31c5bd90b4a1f352b284936dacc5930557e0534cfb26154611bbcabb133ec127e054787b8482124f667bf0180723a3
51b2200e58576c0a61c451309c78700e29b66da314c38b0722c3980b9205ba15c072acabaf0338bf9e08f05649f63239b731c8b65d76738348f9c24c97ec67b6a0b0f8a2a4c11176f40604c8cfc605b3b680acb950743c4d48630cc07c8f21acc83b37374857871a4a425fcb7d54750876
811b10c203a6888b87396c8a16c10d638bddb575e659230d97eb72b5f79c9aea35149681587fa8418c79878a0803b63bc8fd4959348d5bc23e963df0c6ad2105627b534ab534d90d185a46c611f42b792146b3ec0ad98174e410239fe129ca1a89eff458efd716517e647b6354d1689c36
dcb32e2338b9f79c83556857dd286a6f62d75259eb2b8281c211ea971afc21ca0495760dac158705578622760631773602a9048877620b8b07b32233f7bba84225e4448a46f594ac449bad6643a7ca7542071b3d555b23d2b43d953bb477b221a7531f4b25835e9a7dead674087204ec59
077fdf685e94bb996acabac
sk = 4d2b53b122053383b8c392163f16706f9663b719cd397b7537d6aca769cf821475c8d5c54f303f15dba4ec05285d60404945bd73884085524968d00bdb348a45163949cc688bb258715a553fb77e0454c2569b2a2750c651c51740671fb3d620bfe4c6e0902f481b667e075b2ab98
32ef01243dc21912a5dae1c7bd4aa8d8c7879ae9b7ec5e7c692112909872f6258a621dc4b412313dc38c323bb7c9af78c006bc0982c815a67c177f3adcfe6cf5c439fc5da8f3633cb9662494ee051cc436f7bf579ff9982fa909c6740bc4a45c2e6589354f111491954d48a51c686bfc01
a358b63127346909b7a4fe0727ce503b63226386c71ab95902cafb38508e98667d633846a2b598327615879d1a82ee08cbbcbbb4fb2e7056279399b1ab6b43762dd2129106a004dabbe52bb2a4f724f3aa359240507eb0c18d6e2a156986e1800a8d8d1031ce4c3595c9580d934288a34d
cd486afb322ffd57bcae787501c8c38f1a9944cb61093af696064389756a981322fe219c2a153bfbb9f5e8003fba595b847a613580c656b49c5619405648e66432fe77878c2cb1d9fc4254e8586add51e202bb63d4608f3f47358a80f54c3b1559a8042306cf76895ca0509d95b9614422
cbb90014f50aa825575ffc06e73c72cf46814c99ab8a6141477f05b33344ad18879aefcaf1e546f049c81ecb9791fb8c34bba65e5e45dc16b6e358aad2535c28265b800913d54468973aa8ef4ab5445bcc864f7a93791c41ad3b2b060022d20cc39b40d39453633fa4acc1c5e29903800c
023ca16ab97c669081171e2678420f752b4a94bf1a46f0b42cbaccb7dbb986cc7a7afe444c4d0519638126811f903ae5c1e659c296a39125910206927a09100544ad52401dc66b3168188acb8084535c97a00acc0bd5d86ba334665d564bcd57357f7d66c086117981636b8b3b735581d3
9e971512a5ababbc389562f9096c589a9248d396dfda094f612128db957591b681bb604717bc6d3625ddeb946c6199071386d08088b5fa87f0c5648034224589983558c60a3c1b6c004ae59f1cf50fa550cb50419c77513804c2e2276041151032b72f69bb2ff06a682caa922270f121a8
ad3311feab75038cc3dcbe7b4abd59b07e191c487b1456804689acf4e9acec7c79f5b3ba2f21c821b7294dfbb7d4f65cc9e1530151560431255a0c45cc4992ac6629b4ec7748cb6ab71c04739ca243403c081141779461c9cd9797c3c7ec553157ce69d09570ba67659ad561dd9921b88b
3ce17d18633dcbb3cbc64de2b244c877cd3089b5514627532695b734259f8a6e0e1393d03a6676c8c1f2170f71314dd12a2d337a7dd0152dd72bbf68cc7534a02de8ab09220c0efac8f652a7c4441c8faaa9ed550197290509b53c1415c4e3289b098457524179bc6f50637bcc3df66b01
d45b7ca74be572c5129d4940568357ce42687d463015a3f8f81a557a5a3063bbf05a85acd591a92e722d09b20c559313b5c3449f314fda59e0538a088b4142b8440852b047b19139ad3ccf7b172308444f1442e7e304b2d625b606175ac53286b137e64d24e6689356648a4fd27a4fb088
1588b1a2ddb94b7825a2e61a56cb031d755b8c126ceffa3c2ae97393b3b3b7887646bd171b48c3eeac59ad31c5bd90b4a1f352b284936dacc5930557e0534cfb26154611bbcabb133ec127e054787b8482124f667bf0180723a351b2200e58576c0a61c451309c78700e29b66da314c38b
0722c3980b9205ba15c072acabaf0338bf9e08f05649f63239b731c8b65d76738348f9c24c97ec67b6a0b0f8a2a4c11176f40604c8cfc605b3b680acb950743c4d48630cc07c8f21acc83b37374857871a4a425fcb7d54750876811b10c203a6888b87396c8a16c10d638bddb575e65923
0d97eb72b5f79c9aea35149681587fa8418c79878a0803b63bc8fd4959348d5bc23e963df0c6ad2105627b534ab534d90d185a46c611f42b792146b3ec0ad98174e410239fe129ca1a89eff458efd716517e647b6354d1689c36dcb32e2338b9f79c83556857dd286a6f62d75259eb2b82
81c211ea971afc21ca0495760dac158705578622760631773602a9048877620b8b07b32233f7bba84225e4448a46f594ac449bad6643a7ca7542071b3d555b23d2b43d953bb477b221a7531f4b25835e9a7dead674087204ec59077fdf685e94bb996acabacefe48b6649dbde082e1efc8
5bc30c5ecc68efd60115ac90cd07d720b3b943b6e1ba1bebe44f430bb012187d2b84045fd6679a11b7cfe6d03a7b568827f71d8df
```

ML-KEM-512 Key Encapsulation

```
# KAT TEST for Kyber512
# Kyber512 Encapsulation Request file
# Generated on: Thu Aug 22 13:15:48 2024


count = 0
pk = 041151032b72f69bb2ff06a682caa922270f121a8ad3311feab75038cc3dcbe7b4abd59b07e191c487b1456804689acf4e9acec7c79f5b3ba2f21c821b7294dfbb7d4f65cc9e1530151560431255a0c45cc4992ac6629b4ec7748cb6ab71c04739ca243403c081141779461c9cd97
97c3c7ec553157ce69d09570ba67659ad561dd9921b88b3ce17d18633dcbb3cbc64de2b244c877cd3089b5514627532695b734259f8a6e0e1393d03a6676c8c1f2170f71314dd12a2d337a7dd0152dd72bbf68cc7534a02de8ab09220c0efac8f652a7c4441c8faaa9ed550197290509b5
3c1415c4e3289b098457524179bc6f50637bcc3df66b01d45b7ca74be572c5129d4940568357ce42687d463015a3f8f81a557a5a3063bbf05a85acd591a92e722d09b20c559313b5c3449f314fda59e0538a088b4142b8440852b047b19139ad3ccf7b172308444f1442e7e304b2d625b6
06175ac53286b137e64d24e6689356648a4fd27a4fb0881588b1a2ddb94b7825a2e61a56cb031d755b8c126ceffa3c2ae97393b3b3b7887646bd171b48c3eeac59ad31c5bd90b4a1f352b284936dacc5930557e0534cfb26154611bbcabb133ec127e054787b8482124f667bf0180723a3
51b2200e58576c0a61c451309c78700e29b66da314c38b0722c3980b9205ba15c072acabaf0338bf9e08f05649f63239b731c8b65d76738348f9c24c97ec67b6a0b0f8a2a4c11176f40604c8cfc605b3b680acb950743c4d48630cc07c8f21acc83b37374857871a4a425fcb7d54750876
811b10c203a6888b87396c8a16c10d638bddb575e659230d97eb72b5f79c9aea35149681587fa8418c79878a0803b63bc8fd4959348d5bc23e963df0c6ad2105627b534ab534d90d185a46c611f42b792146b3ec0ad98174e410239fe129ca1a89eff458efd716517e647b6354d1689c36
dcb32e2338b9f79c83556857dd286a6f62d75259eb2b8281c211ea971afc21ca0495760dac158705578622760631773602a9048877620b8b07b32233f7bba84225e4448a46f594ac449bad6643a7ca7542071b3d555b23d2b43d953bb477b221a7531f4b25835e9a7dead674087204ec59
077fdf685e94bb996acabac
m = 4cb482ffea3443a40b46cc3851d2587eb9813b21873c50fb7584b2badc6ffae4
kr = 5054adf2b77641620b81ea19425fb2235476fb85d8dd048cd8f3fe30ab2b9729e6999257697b7b6fe542b0d85a2f4a42d0a070ef0796cd9f55a248df07593f04
```

```
# KAT TEST for Kyber512
# Kyber512 Encapsulation Response file
# Generated on: Thu Aug 22 13:15:48 2024


count = 0
pk = 041151032b72f69bb2ff06a682caa922270f121a8ad3311feab75038cc3dcbe7b4abd59b07e191c487b1456804689acf4e9acec7c79f5b3ba2f21c821b7294dfbb7d4f65cc9e1530151560431255a0c45cc4992ac6629b4ec7748cb6ab71c04739ca243403c081141779461c9cd97
97c3c7ec553157ce69d09570ba67659ad561dd9921b88b3ce17d18633dcbb3cbc64de2b244c877cd3089b5514627532695b734259f8a6e0e1393d03a6676c8c1f2170f71314dd12a2d337a7dd0152dd72bbf68cc7534a02de8ab09220c0efac8f652a7c4441c8faaa9ed550197290509b5
3c1415c4e3289b098457524179bc6f50637bcc3df66b01d45b7ca74be572c5129d4940568357ce42687d463015a3f8f81a557a5a3063bbf05a85acd591a92e722d09b20c559313b5c3449f314fda59e0538a088b4142b8440852b047b19139ad3ccf7b172308444f1442e7e304b2d625b6
06175ac53286b137e64d24e6689356648a4fd27a4fb0881588b1a2ddb94b7825a2e61a56cb031d755b8c126ceffa3c2ae97393b3b3b7887646bd171b48c3eeac59ad31c5bd90b4a1f352b284936dacc5930557e0534cfb26154611bbcabb133ec127e054787b8482124f667bf0180723a3
51b2200e58576c0a61c451309c78700e29b66da314c38b0722c3980b9205ba15c072acabaf0338bf9e08f05649f63239b731c8b65d76738348f9c24c97ec67b6a0b0f8a2a4c11176f40604c8cfc605b3b680acb950743c4d48630cc07c8f21acc83b37374857871a4a425fcb7d54750876
811b10c203a6888b87396c8a16c10d638bddb575e659230d97eb72b5f79c9aea35149681587fa8418c79878a0803b63bc8fd4959348d5bc23e963df0c6ad2105627b534ab534d90d185a46c611f42b792146b3ec0ad98174e410239fe129ca1a89eff458efd716517e647b6354d1689c36
dcb32e2338b9f79c83556857dd286a6f62d75259eb2b8281c211ea971afc21ca0495760dac158705578622760631773602a9048877620b8b07b32233f7bba84225e4448a46f594ac449bad6643a7ca7542071b3d555b23d2b43d953bb477b221a7531f4b25835e9a7dead674087204ec59
077fdf685e94bb996acabac
m = 4cb482ffea3443a40b46cc3851d2587eb9813b21873c50fb7584b2badc6ffae4
kr = 5054adf2b77641620b81ea19425fb2235476fb85d8dd048cd8f3fe30ab2b9729e6999257697b7b6fe542b0d85a2f4a42d0a070ef0796cd9f55a248df07593f04
ct = 217dad151306206729b0fa74252fc90f86d49165bee7193840a57cc91c3fdcce9a3389ba8b11e79cb088c9f3a78cefb7503d6d4fbc52041a6af80c63904120350e62d566496bd162d40e6a8b308de6bd14ab53dd54247cc89f03344e27c625306d7bea49a8d5dfd22bf503aa38b4e
dcffe3d3a69dc69b5fff37a73ae727da34933b8f26f0ae255c9842c5ea190f42dfc9e2cf80c9f1e0233e12ff0d19462d29a97e47844be4de30cfbd310f03ee5607a68180889f523e3751f1a70ae298753931c06809ff9f76d22a34c1c91ead7730a8764d6ecad6b72ae2e41042613ec7ec
131fc2d7985ef6b26a1632435d14fac11a570f0b1a4c81650806406d333d785f71d443e78e9effdcc66fc15bf43332758ea33b3029eebf92986adb9628a05b7831b69de789e1eb24a55f6f568e7d5de150f1c44914a626cbf659546010dbab1ee943927e24d00812f0bd0bd8f6ab327391
93e595681579b3dd73f670038038253f6efadf322ea72994ec98fc6bb45a25577bfa0a911a372a0d2f81a2405ee075ffd7916c116460666eaf8a11d090ba6093cf2b6167597f1aad0eab3aee4f6b829438fe2acb3460a430eb13d6bf14b1a9ecb1101e548412f8b5cd8a2425a42d0dfbbc
8b9e1f1ac5cbb9e809ed63a022e779ed712121d942f8ba3306e3ff66826cacb8dd7944d0f1cce1a9346785f495734caeb008cc3181fd4f753215aa31de740ab43a154ebd38ce15b8de653a826edf3f1631f930eb3194fb268baa677f4c4edce7498458351f9277d77056a303786ed303c6
c2e76cd1df0171db65a6c3e4bffbca5633fa3820c6c6d4496a37b0138d99515b1ad25f4309fd49ca6c3d36f0c02280519ed1f05f06db963b419e7dc5ae1a0607708b3add6c0c9848fa0cd3112d85d762ad0faf39af1e618c4a376288ae7494472a44dc7dd0bbae3f50696e828065268d6d
ebbf7f30aa963782b0999b65793d36f6771470382dae3195f49b0918ff5fad53e9d3f8346d0ecc4756c53c2954576a4fcd40429fe3214620dbb749e71340ad185338882403d3e076979b35cffc2f58e807ccfaa63deca51714b314a7d
ss = 1e66f7ac6aa3316899f104ac8c8214b3badad036514477eb2e2015c29e335d3f
```

ML-KEM-512 Key Decapsulation

```
# KAT TEST for Kyber512
# Kyber512 Decapsulation Request file
# Generated on: Thu Aug 22 13:15:48 2024


count = 0
sk = 4d2b53b122053383b8c392163f16706f9663b719cd397b7537d6aca769cf821475c8d5c54f303f15dba4ec05285d60404945bd73884085524968d00bdb348a45163949cc688bb258715a553fb77e0454c2569b2a2750c651c51740671fb3d620bfe4c6e0902f481b667e075b2ab98
32ef01243dc21912a5dae1c7bd4aa8d8c7879ae9b7ec5e7c692112909872f6258a621dc4b412313dc38c323bb7c9af78c006bc0982c815a67c177f3adcfe6cf5c439fc5da8f3633cb9662494ee051cc436f7bf579ff9982fa909c6740bc4a45c2e6589354f111491954d48a51c686bfc01
a358b63127346909b7a4fe0727ce503b63226386c71ab95902cafb38508e98667d633846a2b598327615879d1a82ee08cbbcbbb4fb2e7056279399b1ab6b43762dd2129106a004dabbe52bb2a4f724f3aa359240507eb0c18d6e2a156986e1800a8d8d1031ce4c3595c9580d934288a34d
cd486afb322ffd57bcae787501c8c38f1a9944cb61093af696064389756a981322fe219c2a153bfbb9f5e8003fba595b847a613580c656b49c5619405648e66432fe77878c2cb1d9fc4254e8586add51e202bb63d4608f3f47358a80f54c3b1559a8042306cf76895ca0509d95b9614422
cbb90014f50aa825575ffc06e73c72cf46814c99ab8a6141477f05b33344ad18879aefcaf1e546f049c81ecb9791fb8c34bba65e5e45dc16b6e358aad2535c28265b800913d54468973aa8ef4ab5445bcc864f7a93791c41ad3b2b060022d20cc39b40d39453633fa4acc1c5e29903800c
023ca16ab97c669081171e2678420f752b4a94bf1a46f0b42cbaccb7dbb986cc7a7afe444c4d0519638126811f903ae5c1e659c296a39125910206927a09100544ad52401dc66b3168188acb8084535c97a00acc0bd5d86ba334665d564bcd57357f7d66c086117981636b8b3b735581d3
9e971512a5ababbc389562f9096c589a9248d396dfda094f612128db957591b681bb604717bc6d3625ddeb946c6199071386d08088b5fa87f0c56480342245899835 58c60a3c1b6c004ae59f1cf50fa550cb50419c77513804c2e2276041151032b72f69bb2ff06a682caa922270f121a8
ad3311feab75038cc3dcbe7b4abd59b07e191c487b1456804689acf4e9acec7c79f5b3ba2f21c821b7294dfbb7d4f65cc9e1530151560431255a0c45cc4992ac6629b4ec7748cb6ab71c04739ca243403c081141779461c9cd9797c3c7ec553157ce69d09570ba67659ad561dd9921b88b
3ce17d18633dcbb3cbc64de2b244c877cd3089b5514627532695b734259f8a6e0e1393d03a6676c8c1f2170f71314dd12a2d337a7dd0152dd72bbf68cc7534a02de8ab09220c0efac8f652a7c4441c8faaa9ed550197290509b53c1415c4e3289b098457524179bc6f50637bcc3df66b01
d45b7ca74be572c5129d4940568357ce42687d463015a3f8f81a557a5a3063bbf05a85acd591a92e722d09b20c559313b5c3449f314fda59e0538a088b4142b8440852b047b19139ad3ccf7b172308444f1442e7e304b2d625b606175ac53286b137e64d24e6689356648a4fd27a4fb088
1588b1a2ddb94b7825a2e61a56cb031d755b8c126ceffa3c2ae97393b3b3b7887646bd171b48c3eeac59ad31c5bd90b4a1f352b284936dacc5930557e0534cfb26154611bbcabb133ec127e054787b8482124f667bf0180723a351b2200e58576c0a61c451309c78700e29b66da314c38b
0722c3980b9205ba15c072acabaf0338bf9e08f05649f63239b731c8b65d76738348f9c24c97ec67b6a0b0f8a2a4c11176f40604c8cfc605b3b680acb950743c4d48630cc07c8f21acc83b37374857871a4a425fcb7d54750876811b10c203a6888b87396c8a16c10d638bddb575e65923
0d97eb72b5f79c9aea35149681587fa8418c79878a0803b63bc8fd4959348d5bc23e963df0c6ad2105627b534ab534d90d185a46c611f42b792146b3ec0ad98174e410239fe129ca1a89eff458efd716517e647b6354d1689c36dcb32e2338b9f79c83556857dd286a6f62d75259eb2b82
81c211ea971afc21ca0495760dac158705578622760631773602a9048877620b8b07b32233f7bba84225e4448a46f594ac449bad6643a7ca7542071b3d555b23d2b43d953bb477b221a7531f4b25835e9a7dead674087204ec59077fdf685e94bb996acabacefe48b6649dbde082e1efc8
5bc30c5ecc68efd60115ac90cd07d720b3b943b6e1ba1bebe44f430bb012187d2b84045fd6679a11b7cfe6d03a7b568827f71d8df
ct = 217dad151306206729b0fa74252fc90f86d49165bee7193840a57cc91c3fdcce9a3389ba8b11e79cb088c9f3a78cefb7503d6d4fbc52041a6af80c63904120350e62d566496bd162d40e6a8b308de6bd14ab53dd54247cc89f03344e27c625306d7bea49a8d5dfd22bf503aa38b4e
dcffe3d3a69dc69b5fff37a73ae727da34933b8f26f0ae255c9842c5ea190f42dfc9e2cf80c9f1e0233e12ff0d19462d29a97e47844be4de30cfbd310f03ee5607a68180889f523e3751f1a70ae298753931c06809ff9f76d22a34c1c91ead7730a8764d6ecad6b72ae2e41042613ec7ec
131fc2d7985ef6b26a1632435d14fac11a570f0b1a4c81650806406d333d785f71d443e78e9effdcc66fc15bf43332758ea33b3029eebf92986adb9628a05b7831b69de789e1eb24a55f6f568e7d5de150f1c44914a626cbf659546010dbab1ee943927e24d00812f0bd0bd8f6ab327391
93e595681579b3dd73f670038038253f6efadf322ea72994ec98fc6bb45a25577bfa0a911a372a0d2f81a2405ee075ffd7916c116460666eaf8a11d090ba6093cf2b6167597f1aad0eab3aee4f6b829438fe2acb3460a430eb13d6bf14b1a9ecb1101e548412f8b5cd8a2425a42d0dfbbc
8b9e1f1ac5cbb9e809ed63a022e779ed712121d942f8ba3306e3ff66826cacb8dd7944d0f1cce1a9346785f495734caeb008cc3181fd4f753215aa31de740ab43a154ebd38ce15b8de653a826edf3f1631f930eb3194fb268baa677f4c4edce7498458351f9277d77056a303786ed303c6
c2e76cd1df0171db65a6c3e4bffbca5633fa3820c6c6d4496a37b0138d99515b1ad25f4309fd49ca6c3d36f0c02280519ed1f05f06db963b419e7dc5ae1a0607708b3add6c0c9848fa0cd3112d85d762ad0faf39af1e618c4a376288ae7494472a44dc7dd0bbae3f50696e828065268d6d
ebbf7f30aa963782b0999b65793d36f6771470382dae3195f49b0918ff5fad53e9d3f8346d0ecc4756c53c2954576a4fcd40429fe3214620dbb749e71340ad185338882403d3e076979b35cffc2f58e807ccfaa63deca51714b314a7d
```

```
# KAT TEST for Kyber512
# Kyber512 Decapsulation Request file
# Generated on: Thu Aug 22 13:15:48 2024


count = 0
sk = 4d2b53b122053383b8c392163f16706f9663b719cd397b7537d6aca769cf821475c8d5c54f303f15dba4ec05285d60404945bd73884085524968d00bdb348a45163949cc688bb258715a553fb77e0454c2569b2a2750c651c51740671fb3d620bfe4c6e0902f481b667e075b2ab98
32ef01243dc21912a5dae1c7bd4aa8d8c7879ae9b7ec5e7c692112909872f6258a621dc4b412313dc38c323bb7c9af78c006bc0982c815a67c177f3adcfe6cf5c439fc5da8f3633cb9662494ee051cc436f7bf579ff9982fa909c6740bc4a45c2e6589354f111491954d48a51c686bfc01
a358b63127346909b7a4fe0727ce503b63226386c71ab95902cafb38508e98667d633846a2b598327615879d1a82ee08cbbcbbb4fb2e7056279399b1ab6b43762dd2129106a004dabbe52bb2a4f724f3aa359240507eb0c18d6e2a156986e1800a8d8d1031ce4c3595c9580d934288a34d
cd486afb322ffd57bcae787501c8c38f1a9944cb61093af696064389756a981322fe219c2a153bfbb9f5e8003fba595b847a613580c656b49c5619405648e66432fe77878c2cb1d9fc4254e8586add51e202bb63d4608f3f47358a80f54c3b1559a8042306cf76895ca0509d95b9614422
cbb90014f50aa825575ffc06e73c72cf46814c99ab8a6141477f05b33344ad18879aefcaf1e546f049c81ecb9791fb8c34bba65e5e45dc16b6e358aad2535c28265b800913d54468973aa8ef4ab5445bcc864f7a93791c41ad3b2b060022d20cc39b40d39453633fa4acc1c5e29903800c
023ca16ab97c669081171e2678420f752b4a94bf1a46f0b42cbaccb7dbb986cc7a7afe444c4d0519638126811f903ae5c1e659c296a39125910206927a09100544ad52401dc66b3168188acb8084535c97a00acc0bd5d86ba334665d564bcd57357f7d66c086117981636b8b3b735581d3
9e971512a5ababbc389562f9096c589a9248d396dfda094f612128db957591b681bb604717bc6d3625ddeb946c6199071386d08088b5fa87f0c56480342245899835 58c60a3c1b6c004ae59f1cf50fa550cb50419c77513804c2e2276041151032b72f69bb2ff06a682caa922270f121a8
ad3311feab75038cc3dcbe7b4abd59b07e191c487b1456804689acf4e9acec7c79f5b3ba2f21c821b7294dfbb7d4f65cc9e1530151560431255a0c45cc4992ac6629b4ec7748cb6ab71c04739ca243403c081141779461c9cd9797c3c7ec553157ce69d09570ba67659ad561dd9921b88b
3ce17d18633dcbb3cbc64de2b244c877cd3089b5514627532695b734259f8a6e0e1393d03a6676c8c1f2170f71314dd12a2d337a7dd0152dd72bbf68cc7534a02de8ab09220c0efac8f652a7c4441c8faaa9ed550197290509b53c1415c4e3289b098457524179bc6f50637bcc3df66b01
d45b7ca74be572c5129d4940568357ce42687d463015a3f8f81a557a5a3063bbf05a85acd591a92e722d09b20c559313b5c3449f314fda59e0538a088b4142b8440852b047b19139ad3ccf7b172308444f1442e7e304b2d625b606175ac53286b137e64d24e6689356648a4fd27a4fb088
1588b1a2ddb94b7825a2e61a56cb031d755b8c126ceffa3c2ae97393b3b3b7887646bd171b48c3eeac59ad31c5bd90b4a1f352b284936dacc5930557e0534cfb26154611bbcabb133ec127e054787b8482124f667bf0180723a351b2200e58576c0a61c451309c78700e29b66da314c38b
0722c3980b9205ba15c072acabaf0338bf9e08f05649f63239b731c8b65d76738348f9c24c97ec67b6a0b0f8a2a4c11176f40604c8cfc605b3b680acb950743c4d48630cc07c8f21acc83b37374857871a4a425fcb7d54750876811b10c203a6888b87396c8a16c10d638bddb575e65923
0d97eb72b5f79c9aea35149681587fa8418c79878a0803b63bc8fd4959348d5bc23e963df0c6ad2105627b534ab534d90d185a46c611f42b792146b3ec0ad98174e410239fe129ca1a89eff458efd716517e647b6354d1689c36dcb32e2338b9f79c83556857dd286a6f62d75259eb2b82
81c211ea971afc21ca0495760dac158705578622760631773602a9048877620b8b07b32233f7bba84225e4448a46f594ac449bad6643a7ca7542071b3d555b23d2b43d953bb477b221a7531f4b25835e9a7dead674087204ec59077fdf685e94bb996acabacefe48b6649dbde082e1efc8
5bc30c5ecc68efd60115ac90cd07d720b3b943b6e1ba1bebe44f430bb012187d2b84045fd6679a11b7cfe6d03a7b568827f71d8df
ct = 217dad151306206729b0fa74252fc90f86d49165bee7193840a57cc91c3fdcce9a3389ba8b11e79cb088c9f3a78cefb7503d6d4fbc52041a6af80c63904120350e62d566496bd162d40e6a8b308de6bd14ab53dd54247cc89f03344e27c625306d7bea49a8d5dfd22bf503aa38b4e
dcffe3d3a69dc69b5fff37a73ae727da34933b8f26f0ae255c9842c5ea190f42dfc9e2cf80c9f1e0233e12ff0d19462d29a97e47844be4de30cfbd310f03ee5607a68180889f523e3751f1a70ae298753931c06809ff9f76d22a34c1c91ead7730a8764d6ecad6b72ae2e41042613ec7ec
131fc2d7985ef6b26a1632435d14fac11a570f0b1a4c81650806406d333d785f71d443e78e9effdcc66fc15bf43332758ea33b3029eebf92986adb9628a05b7831b69de789e1eb24a55f6f568e7d5de150f1c44914a626cbf659546010dbab1ee943927e24d00812f0bd0bd8f6ab327391
93e595681579b3dd73f670038038253f6efadf322ea72994ec98fc6bb45a25577bfa0a911a372a0d2f81a2405ee075ffd7916c116460666eaf8a11d090ba6093cf2b6167597f1aad0eab3aee4f6b829438fe2acb3460a430eb13d6bf14b1a9ecb1101e548412f8b5cd8a2425a42d0dfbbc
8b9e1f1ac5cbb9e809ed63a022e779ed712121d942f8ba3306e3ff66826cacb8dd7944d0f1cce1a9346785f495734caeb008cc3181fd4f753215aa31de740ab43a154ebd38ce15b8de653a826edf3f1631f930eb3194fb268baa677f4c4edce7498458351f9277d77056a303786ed303c6
c2e76cd1df0171db65a6c3e4bffbca5633fa3820c6c6d4496a37b0138d99515b1ad25f4309fd49ca6c3d36f0c02280519ed1f05f06db963b419e7dc5ae1a0607708b3add6c0c9848fa0cd3112d85d762ad0faf39af1e618c4a376288ae7494472a44dc7dd0bbae3f50696e828065268d6d
ebbf7f30aa963782b0999b65793d36f6771470382dae3195f49b0918ff5fad53e9d3f8346d0ecc4756c53c2954576a4fcd40429fe3214620dbb749e71340ad185338882403d3e076979b35cffc2f58e807ccfaa63deca51714b314a7d
ss1 = 1e66f7ac6aa3316899f104ac8c8214b3badad036514477eb2e2015c29e335d3f
```

ML-KEM-512 shared secret key verification

```
# KAT TEST for Kyber512
# Kyber512 TEXT REFERENCE file
# Generated on: Thu Aug 22 13:15:48 2024


count = 0
seed = 7f7bd0b678080787b1ea8cf1ea7d2cb658a8caa8869fa96fd249501983a622de1ba1bebe44f430bb012187d2b84045fd6679a11b7cfe6d03a7b568827f71d8df
pk = 041151032b72f69bb2ff06a682caa922270f121a8ad3311feab75038cc3dcbe7b4abd59b07e191c487b1456804689acf4e9acec7c79f5b3ba2f21c821b7294dfbb7d4f65cc9e1530151560431255a0c45cc4992ac6629b4ec7748cb6ab71c04739ca243403c081141779461c9cd97
97c3c7ec553157ce69d09570ba67659ad561dd9921b88b3ce17d18633dcbb3cbc64de2b244c877cd3089b5514627532695b734259f8a6e0e1393d03a6676c8c1f2170f71314dd12a2d337a7dd0152dd72bbf68cc7534a02de8ab09220c0efac8f652a7c4441c8faaa9ed550197290509b5
3c1415c4e3289b098457524179bc6f50637bcc3df66b01d45b7ca74be572c5129d4940568357ce42687d463015a3f8f81a557a5a3063bbf05a85acd591a92e722d09b20c559313b5c3449f314fda59e0538a088b4142b8440852b047b19139ad3ccf7b172308444f1442e7e304b2d625b6
06175ac53286b137e64d24e6689356648a4fd27a4fb0881588b1a2ddb94b7825a2e61a56cb031d755b8c126ceffa3c2ae97393b3b3b7887646bd171b48c3eeac59ad31c5bd90b4a1f352b284936dacc5930557e0534cfb26154611bbcabb133ec127e054787b8482124f667bf0180723a3
51b2200e58576c0a61c451309c78700e29b66da314c38b0722c3980b9205ba15c072acabaf0338bf9e08f05649f63239b731c8b65d76738348f9c24c97ec67b6a0b0f8a2a4c11176f40604c8cfc605b3b680acb950743c4d48630cc07c8f21acc83b37374857871a4a425fcb7d54750876
811b10c203a6888b87396c8a16c10d638bddb575e659230d97eb72b5f79c9aea35149681587fa8418c79878a0803b63bc8fd4959348d5bc23e963df0c6ad2105627b534ab534d90d185a46c611f42b792146b3ec0ad98174e410239fe129ca1a89eff458efd716517e647b6354d1689c36
dcb32e2338b9f79c83556857dd286a6f62d75259eb2b8281c211ea971afc21ca0495760dac1587055786227606317736O2a9048877620b8b07b32233f7bba84225e4448a46f594ac449bad6643a7ca7542071b3d555b23d2b43d953bb477b221a7531f4b25835e9a7dead674087204ec59
077fdf685e94bb996acabac
sk = 4d2b53b122053383b8c392163f16706f9663b719cd397b7537d6aca769cf821475c8d5c54f303f15dba4ec05285d60404945bd73884085524968d00bdb348a45163949cc688bb258715a553fb77e0454c2569b2a2750c651c51740671fb3d620bfe4c6e0902f481b667e075b2ab98
32ef01243dc21912a5dae1c7bd4aa8d8c7879ae9b7ec5e7c692112909872f6258a621dc4b412313dc38c323bb7c9af78c006bc0982c815a67c177f3adcfe6cf5c439fc5da8f3633cb9662494ee051cc436f7bf579ff9982fa909c6740bc4a45c2e6589354f111491954d48a51c686bfc01
a358b63127346909b7a4fe0727ce503b63226386c71ab95902cafb38508e98667d633846a2b598327615879d1a82ee08cbbcbbb4fb2e7056279399b1ab6b43762dd2129106a004dabbe52bb2a4f724f3aa359240507eb0c18d6e2a156986e1800a8d8d1031ce4c3595c9580d934288a34d
cd486afb322ffd57bcae787501c8c38f1a9944cb61093af696064389756a981322fe219c2a153bfbb9f5e8003fba595b847a613580c656b49c5619405648e66432fe77878c2cb1d9fc4254e8586add51e202bb63d4608f3f47358a80f54c3b1559a8042306cf76895ca0509d95b9614422
cbb90014f50aa825575ffc06e73c72cf46814c99ab8a6141477f05b33344ad18879aefcaf1e546f049c81ecb9791fb8c34bba65e5e45dc16b6e358aad2535c28265b800913d54468973aa8ef4ab5445bcc864f7a93791c41ad3b2b060022d20cc39b40d39453633fa4acc1c5e29903800c
023ca16ab97c669081171e2678420f752b4a94bf1a46f0b42cbaccb7dbb986cc7a7afe444c4d0519638126811f903ae5c1e659c296a39125910206927a09100544ad52401dc66b3168188acb8084535c97a00acc0bd5d86ba334665d564bcd57357f7d66c086117981636b8b3b735581d3
9e971512a5ababbc389562f9096c589a9248d396dfda094f612128db957591b681bb604717bc6d3625ddeb946c6199071386d08088b5fa87f0c5648034224589983558c60a3c1b6c004ae59f1cf50fa550cb50419c77513804c2e2276041151032b72f69bb2ff06a682caa922270f121a8
ad3311feab75038cc3dcbe7b4abd59b07e191c487b1456804689acf4e9acec7c79f5b3ba2f21c821b7294dfbb7d4f65cc9e1530151560431255a0c45cc4992ac6629b4ec7748cb6ab71c04739ca243403c081141779461c9cd9797c3c7ec553157ce69d09570ba67659ad561dd9921b88b
3ce17d18633dcbb3cbc64de2b244c877cd3089b5514627532695b734259f8a6e0e1393d03a6676c8c1f2170f71314dd12a2d337a7dd0152dd72bbf68cc7534a02de8ab09220c0efac8f652a7c4441c8faaa9ed550197290509b53c1415c4e3289b098457524179bc6f50637bcc3df66b01
d45b7ca74be572c5129d4940568357ce42687d463015a3f8f81a557a5a3063bbf05a85acd591a92e722d09b20c559313b5c3449f314fda59e0538a088b4142b8440852b047b19139ad3ccf7b172308444f1442e7e304b2d625b606175ac53286b137e64d24e6689356648a4fd27a4fb088
1588b1a2ddb94b7825a2e61a56cb031d755b8c126ceffa3c2ae97393b3b3b7887646bd171b48c3eeac59ad31c5bd90b4a1f352b284936dacc5930557e0534cfb26154611bbcabb133ec127e054787b8482124f667bf0180723a351b2200e58576c0a61c451309c78700e29b66da314c38b
0722c3980b9205ba15c072acabaf0338bf9e08f05649f63239b731c8b65d76738348f9c24c97ec67b6a0b0f8a2a4c11176f40604c8cfc605b3b680acb950743c4d48630cc07c8f21acc83b37374857871a4a425fcb7d54750876811b10c203a6888b87396c8a16c10d638bddb575e65923
0d97eb72b5f79c9aea35149681587fa8418c79878a0803b63bc8fd4959348d5bc23e963df0c6ad2105627b534ab534d90d185a46c611f42b792146b3ec0ad98174e410239fe129ca1a89eff458efd716517e647b6354d1689c36dcb32e2338b9f79c83556857dd286a6f62d75259eb2b82
81c211ea971afc21ca0495760dac1587055786227606317736O2a9048877620b8b07b32233f7bba84225e4448a46f594ac449bad6643a7ca7542071b3d555b23d2b43d953bb477b221a7531f4b25835e9a7dead674087204ec59077fdf685e94bb996acabacefe48b6649dbde082e1efc8
5bc30c5ecc68efd60115ac90cd07d720b3b943b6e1ba1bebe44f430bb012187d2b84045fd6679a11b7cfe6d03a7b568827f71d8df
m = 4cb482ffea3443a40b46cc3851d2587eb9813b21873c50fb7584b2badc6ffae4
kr = 5054adf2b77641620b81ea19425fb2235476fb85d8dd048cd8f3fe30ab2b9729e6999257697b7b6fe542b0d85a2f4a42d0a070ef0796cd9f55a248df07593f04
ct = 217dad151306206729b0fa74252fc90f86d49165bee7193840a57cc91c3fdcce9a3389ba8b11e79cb088c9f3a78cefb7503d6d4fbc52041a6af80c63904120350e62d566496bd162d40e6a8b308de6bd14ab53dd54247cc89f03344e27c625306d7bea49a8d5dfd22bf503aa38b4e
dcffe3d3a69dc69b5fff37a73ae727da34933b8f26f0ae255c9842c5ea190f42dfc9e2cf80c9f1e0233e12ff0d19462d29a97e47844be4de30cfbd310f03ee5607a68180889f523e3751f1a70ae298753931c06809ff9f76d22a34c1c91ead7730a8764d6ecad6b72ae2e41042613ec7ec
131fc2d7985ef6b26a1632435d14fac11a570f0b1a4c81650806406d333d785f71d443e78e9effdcc66fc15bf43332758ea33b3029eebf92986adb9628a05b7831b69de789e1eb24a55f6f568e7d5de150f1c44914a626cbf659546010dbab1ee943927e24d00812f0bd0bd8f6ab327391
93e595681579b3dd73f670038038253f6efadf322ea72994ec98fc6bb45a25577bfa0a911a372a0d2f81a2405ee075ffd7916c116460666eaf8a11d090ba6093cf2b6167597f1aad0eab3aee4f6b829438fe2acb3460a430eb13d6bf14b1a9ecb1101e548412f8b5cd8a2425a42d0dfbbc
8b9e1f1ac5cbb9e809ed63a022e779ed712121d942f8ba3306e3ff66826cacb8dd7944d0f1cce1a9346785f495734caeb008cc3181fd4f753215aa31de740ab43a154ebd38ce15b8de653a826edf3f1631f930eb3194fb268baa677f4c4edce7498458351f9277d77056a303786ed303c6
c2e76cd1df0171db65a6c3e4bffbca5633fa3820c6c6d4496a37b0138d99515b1ad25f4309fd49ca6c3d36f0c02280519ed1f05f06db963b419e7dc5ae1a0607708b3add6c0c9848fa0cd3112d85d762ad0faf39af1e618c4a376288ae7494472a44dc7dd0bbae3f50696e828065268d6d
ebbf7f30aa963782b0999b65793d36f6771470382dae3195f49b0918ff5fad53e9d3f8346d0ecc4756c53c2954576a4fcd40429fe3214620dbb749e71340ad185338882403d3e076979b35cffc2f58e807ccfaa63deca51714b314a7d
ss = 1e66f7ac6aa3316899f104ac8c8214b3badad036514477eb2e2015c29e335d3f
ss1 = 1e66f7ac6aa3316899f104ac8c8214b3badad036514477eb2e2015c29e335d3f
Verification Success!!

count = 1
seed = 484992b5823094a6a9b4d888dac2d9505aeb6885e505911ce4db81c98cc82a03b3fb42ad799b6475694aa6ad5df036e87df99b38be1f0cda9de058d971fd8831
```

References

1. FIPS 203 Federal Information Processing Standards Publication Module-Lattice-based Key-Encapsulation Mechanism Standard.
2. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehle, D. CRYSTALS Kyber Algorithm Specifications and SupportingDocumentation.Availableonline:https://pqcrystals.org/kyber/data/kyberspecification-round3-20210131.pdf (accessed on 8 February 2022).

THANK YOU