main

January 10, 2025

[118]:

```python
import os
import sys

# Add paths
sys.path.append(os.path.abspath(".."))
sys.path.append(os.path.join(os.getcwd(), 'dilithium/src'))

from dilithium_py.dilithium import Dilithium2

# Generate Dilithium key pair
try:
    dilithium_keypair = Dilithium2.keygen()
    dilithium_public_key, dilithium_private_key = dilithium_keypair

    print("Dilithium2 public key length:", len(dilithium_public_key))
    print("Dilithium2 private key length:", len(dilithium_private_key))
except Exception as e:
    print(f"Dilithium key generation error: {e}")
    sys.exit(1)

sys.path.append(os.path.join(os.getcwd(), 'pyky'))
sys.path.append(os.path.join(os.getcwd(), 'src'))

from src.key_management import generate_kyber_keypair, encrypt_session_key,␣
  ↪decrypt_session_key
# Generate Kyber key pair
kyber_private_key, kyber_public_key = generate_kyber_keypair()
print("Kyber512 private key length:", len(kyber_private_key))
print("Kyber512 public key length:", len(kyber_public_key))
```

```
Dilithium2 public key length: 1312
Dilithium2 private key length: 2528
Kyber512 private key length: 3168
Kyber512 public key length: 1568
```

[119]:

```python
#################################################################
# Mock CA: Issuing PQC Certificates (Dilithium-signed X.509) #
#################################################################
```

```python
#ut will be a minimal X.509-like structure in Python and "sign" it
# with the Dilithium private key.

from datetime import datetime, timedelta
import base64
import json
import sys
import os

# Add paths for custom modules
sys.path.append(os.path.abspath(".."))
sys.path.append(os.path.join(os.getcwd(), 'dilithium/src'))
sys.path.append(os.path.join(os.getcwd(), 'pyky'))


from dilithium_py.dilithium import Dilithium2
from src.key_management import generate_kyber_keypair


# generate Dilithium key pair
try:
    dilithium_keypair = Dilithium2.keygen()
    dilithium_public_key, dilithium_private_key = dilithium_keypair

    print("Dilithium2 public key length:", len(dilithium_public_key))
    print("Dilithium2 private key length:", len(dilithium_private_key))
except Exception as e:
    print(f"Dilithium key generation error: {e}")
    sys.exit(1)

# generate Kyber key pair
try:
    kyber_private_key, kyber_public_key = generate_kyber_keypair()
    print("Kyber512 private key length:", len(kyber_private_key))
    print("Kyber512 public key length:", len(kyber_public_key))
except Exception as e:
    print(f"Kyber key generation error: {e}")
    sys.exit(1)


def create_mock_certificate(subject_name: str, pub_key: bytes, issuer_name: str, validity_days=30):
    """
    Create a mock certificate structure in JSON form.
    This is not a real X.509, but a stand-in to show the process.
    """
    cert_data = {
        "subject": subject_name,
        "issuer": issuer_name,
        "valid_from": datetime.utcnow().isoformat() + "Z",
        "valid_to": (datetime.utcnow() + timedelta(days=validity_days)).
 ↪isoformat() + "Z",
        "pqc_public_key": base64.b64encode(pub_key).decode('utf-8'),  # store␣
 ↪the PQC public key
    }
    return cert_data


def sign_certificate(cert_data: dict, private_key: bytes, signature_scheme:␣
 ↪str="Dilithium2"):
    """
    Sign the certificate data with the Dilithium private key
    using the dilithium_py package. We'll store the signature in the cert.
    """
    try:
        message_bytes = json.dumps(cert_data, sort_keys=True).encode('utf-8')
        signature = Dilithium2.sign(private_key, message_bytes)
        cert_data["pqc_signature"] = base64.b64encode(signature).decode('utf-8')
        return cert_data
    except Exception as e:
        print(f"Error signing certificate: {e}")
        sys.exit(1)

#normalize the public key to bytes
if isinstance(kyber_public_key, list):
    kyber_public_key = bytes([x % 256 for x in kyber_public_key])


# 3. create a "Root CA"certificate using Dilithium
root_subject = "CN=MyPQC-RootCA"
root_cert_struct = create_mock_certificate(
    subject_name=root_subject,
    pub_key=dilithium_public_key,
    issuer_name=root_subject,  # self-signed
    validity_days=365
)
root_cert_signed = sign_certificate(root_cert_struct, dilithium_private_key,␣
 ↪"Dilithium2")

# 4.Create an "End-Entity" certificate signed by the root
end_entity_subject = "CN=MyServer"
end_entity_cert_struct = create_mock_certificate(
    subject_name=end_entity_subject,
```

```python
    pub_key=kyber_public_key,  # Ensure this is bytes
    issuer_name=root_subject,
    validity_days=90
)
end_entity_cert_signed = sign_certificate(end_entity_cert_struct,␣
 ↪dilithium_private_key, "Dilithium2")
print("Root CA Certificate:\n", json.dumps(root_cert_signed, indent=2))
print("\nEnd-Entity Certificate:\n", json.dumps(end_entity_cert_signed,␣
 ↪indent=2))
```

```
Dilithium2 public key length: 1312
Dilithium2 private key length: 2528
Kyber512 private key length: 3168
Kyber512 public key length: 1568
Root CA Certificate:
```

```
 {
  "subject": "CN=MyPQC-RootCA",
  "issuer": "CN=MyPQC-RootCA",
  "valid_from": "2025-01-09T16:24:30.053430Z",
  "valid_to": "2026-01-09T16:24:30.053451Z",
  "pqc_public_key": "bqcuf5XtodWq8c5sp3k8M8RIf6UW7kMxDBlawi/NeMWiOAlrF98XJZRVn5q
UIkZBmG+k8CRuKYJisPDoHDZkXWl+anQ2Ex8kef0bp8bPF6SgOsV3TQ4/lIF63pzyOhhbJdpvkEk6UZU
GMv3htomXSdm99sKZkVZY8kDB0VgZL+8vjHT/wgNdBx2HkjagSnPttcNMrVG0XAaY1amDv4skdoi8mO7
aTXtuVNV9I+9n1TODlnl6r9mRabP/bE4LmlPV7pngqBJh1LdW6qEwvP+aAJUPXhlJG8n1+kN/+oPmj2c
KQRFivp6sP5GPnYZ1MjPlgEiPk8dHS09oHSSSuiaX1HHbWWuwhfWqi44vZU1962noFvnioAB62NeZL2r
JDL0VSZoLuHNIRPomUuDEp0fRj4cMCbYbvw4ZzZHqRUAYQg2sBiaLNG8nxFSi10/w/Up1G4L1xl8IXux
jdTWj8Bmi9kwd5ppRQ4/mOKGkuGTKtdk5wZjEWQWK9qmHvtRcqem5Bn1A7DPAje97xtRzliK7y3I9GKi
stzTN3DYKRncGCPRZj6tJBXZrIi5KW5yPhVfIQOh5NDgkhFZ0V5dGRry1rcsoeggw7r+nI3f3lSaoQNl
pdeW5KLakU7IcBSGCQGHbKpKVBQ4hKyM8A0vgCrDSFsN/Wn2TqcomLn0EGK7+Y3YKe0O0eCwOcjbwMup
akhXF55eyzFr8PSLyTOfKhOjrzxLI7tgtuDKDgZ96x4gOCUmAmWY+eUkD7PeqI9BFdl24KFce10ukHgV
6J2iOhNSH7VCDLDJ/QLpEMnPhs4Er7AwZG+VXcM1w+ujF341M72E7FFFAc7dC7qNbanK8M8qqvBFQsJa
JWdIIyU/TkLXUabT21SUwIGcc0RwlfDu6OVOxiw69zaAHE4+4KA49C+IY4UN84l6/SLyRl2sc+RkJOFT
7YxMpWLpmX+v2u1+L1xHqUagW0zFJiaGNsguDkxAo9RfOPWzdRNp2wD0kwhAYHa/yH+wu0uUUj1FStoe
70kFqq3N04ks8Hxj4hAD1+tJ+3luHQ+Rr7BCINwQSNj923SzUCMgS/OfDZeFzP3O18WUTmRJmSuXy7mZ
vqsYlFm4qlI9Vyc15jdHlKFQYbpd6nxMBUYvdAB3cVMSPUlrz5x1rgj8ObOi6gpwCw5kOj+kZi7/rOVn
asnmpCojzvTKqleX3apMPHtHxYQTYmY7WMDxHqBKMIMEiA+17l98dkM4JAGb9qc+YFS4GDpQ//W900gm
pw6xiUaL1DuXvds9FpOHDg+9/EbMhxJ3k+9HGyGOJrk+2vdO1HfQJXyQgt601CzvemztdomMjPn4CkuN
o9IEI6/Q0d29r91ZRcQsAVrrwlllB9CXlNnK2ptr1s7U3V7eAzqxKjyq5fFaKUDEiRpthQnlzd/v2u9+
pez4JzYIx7MdcViSDqGlHP52xb6nSwuU6HMR1Dq+z69g1aS+LhBs7T1kLF4uM47dwDk43DK5pQ57iXBa
tfO2pFV7pEIzOKiUWii+ZMJqQQJ4ukK4NfnSJbqbulWZdaP5HZGmhrbUCEZuTzsPhserj8dli/ZKGO/S
ExlR6DG9glBrcjaRkghO9p+aoQ5f5jFvm1BLoAK2Dz0N7dKtyR7SxzLyLMdbV3k1pYuVbA1LRLgQfiGW
y6rzJtznzldiLiUqnVdHONEHfM5uTbh45Tvwl6HcAswelHByrgzBU4dT2i/LHBBqYeeNqw0Q8P2ODrel
NeIfLPWuSrQ==",
  "pqc_signature": "064J2GYnGxC9yfplbPfF2JJjtnbeJ28PHBpODuwI1T6i1U656XnE5w5rLff9
Ph2YiZKdpkkROPFkHJIsJjkvkKmIJd3A5R3vokqy6iC7eF7/NwIBpZraRgFninJRKIAdWbF/juaJ5ZXj
Ctsm8vXzpMtHq0w6u61wcrxokenMEPe65Iths04I1t7a7Tiw01wIDh9mi3p5N2RoZV4A51CBmRmtK/oV
A6A1Opmgpf1daVhGdNQ/3+gVmi2NL3naYk1I7iisonBLlfLwTpHy9Sm3Wr/9SFpnpCa60YcJYhudlfPR
BW4nwnf0u+JpCgs/Y/YQAlb1iaVp22vcbmydFa20InDRPdM4hGAAdNHr6P8Be54N5EkqTq94y7w3s8RZ
qymJr28cMM2cQpqAw9+ysnGq8WIgoExWUUbajxBOOM+Kylf3Ga22dEcX2eT1vf7NJ7kH5XwV6XPQxGOA
k+DhZIBxWijERDS7AX70pW1oEIr0LovonI/x1fmeXU3IF6ynilxk1TChZpUssbuI5oFVc0n/pbbxS8BP
mz+zKlnmi0/IzZgSX0AaWQ7brPnmlMI7DUSTBI892wvNBiIkdWa9G4W7Mu/mDMv8HaeAY2BuKWfW/AGo
F93yjbZg5T4Jzg0anMb2fVKVHL0nUtyQrQoYQeXKV1zk20G+YntU9XLlunccD2dM6z9HoXbb46xmufZI
7Ym1quaShXw71E2z7yPppOHSlpDvSQ0aETbT0KK1JJn2PTq3qY4jHK60RuAM24H2pElnYo72S1VRqqEF
dy5dDv3TwsgBWhWUH5NqTUA3l2ZZuVRHb7u2cwmIZ3M/tSCkqo6bfe6n4U/RnyIE4cUHBOGDZQ/n7g/n
ExGf8m5XKxcjaa2G77H8AZ51DQKeCWNHql7dwrGoHvkMP6mSnDTsHbRBRb0Tki42LEe/KwnWobrYA19n
zXR0JVbfyObFx81F+y9IK1OlBEdFc2m9vI+WYvZhSdMoshjt1G4Yu5+M8Nz6mWqA+aFocmY/psjMbrsw
MgjhvAyn9Atk5EwrCA/OBmx8EQGEyJ6Ui4ztllPbTj0XE51i8lkpSAl30OmZvLJrHzvN6WHWPTW9XylT
Kv3/O8DH8T6tqRkvVgDoeRZk0TALEyrQJcIxB/G2gep3sxoXie1zCIHCCVCasX/vqY6SycWyrHU96Ruq
2x+RWNHHlG+5Al3secswmoAbzevBKsNuYO4LOz+m2QnFxQ730PiVGeEnBaG/JZ6vvRHJTXKp5EosJgWI
HKVL2d1v6Y+Xtc9i1+8eqVM22Mm0/IeaaZCpV0IVNsqMwg5eMzpVi5ghEac8VqiOL7iCqxI/rkuVL0Px
WoXwmI3P0FlzgOdr8+CopyyM2aGqV3iC8gtO9TVroSZ35TU56V4jJXOUj73B8zCZcDSwXJY1Hbj/Mdvu
NxwMGPIHsUaoXoXQG5Z7kCPa+4rVehUXMbIssQXZ4itYsjYnlvTigzcbJyPQ8XmyvhBr1IFxqSW6yrLU
VihLZhNFESZfs4N9bhJB2n3EDQgSiY6F9GIbPD+7KkvSnWc+43AsTESuukAlU7+H0zGFefodxD5/GJ+0
kwRUOlr5wy1UH0A1BtrUTaKhulAgCYAWLkuEvKRQTh7REyEbow+RZXOS3S5zmYei2xDLVvfE8wfKegZ7
G7/DR2N4xmidVjnXCAb1KjOR/ylUEzedL11EhmVi6pwo3gmdPFHcWtVgz1r43/pJ6BmoYWLozvLnhIOm
qHApiULULEtNDhvpZL3Iiy9wmwcdQO2E9RlIbJNfeg0uwtPS6QOmLG2EZReYi0a1mYCIO6dX5gYm5VU4
KLyfJVRdmty5qvWWTJdzMLHR+QlqYp1I0dVM2G4zfrwIiW6vTIIn3uw8LQJHyk/Fqo8DqYrMvgOGuNyd
RF/afkdWA2YdhlknVlelKyN0jz7roscg0faPKuE6vooK3fLr7GUdVTag279PTO39pjFw0taSbe1TMuyM
1nbmhkk+n8kdScWDiU4vDt+H56WPaV3vUCeranlHVFusk9rbvipwIWntuBYrJCABhAtmqU3FG6qVjJb2
qJh4QUAiTVkrKGqAWaV+B/d3MVlEUg0FUN8lh7Uwd/whCuh+z1wHQ1vg9+uRNsN8dWRDFuht0arigTQp
mpy2gdpq8ApZebIsSQdjHiYGlVVztT/kZoRsdpnA896XznIlhxDaF6vBfMyHg/rc+0BwXGOHEgf9YlWF
wMZw+Jr2pDXAH68nkaZjxQWDdfYFnvMHintUAaL/sCahbDMVAYrLkmm30S/oNXhzpJZNzI7FbNlohf6S
ZSOwWz+L4I8nFOBuA6aWFATrLv0hkCnUGkB2cf+qcjj0da1tpKEZP8uSaD9sBbRfqm63Vhztch69K5qx
P3IRUpzcFLPdfMASNvWABTVUKNNxyVTuSvceWeTLyysGdo0HpgW78dLWCkWDjIR1zFJygZ57al2VZc4O
1akCIcOyWAy/dWf3xZK0tyGXBysMux6XVdQ2/3QBewfknAau7mbGZpj94xY09P+lBxOGzrjynyJBSBcN
9YxZuyS2zCptxc8wRXcwhsp/wFlXZO79Ao+UWNewEby9VWLLuSTNg2bxUzy2dr4p6MHz8iD+ZhNjPgPC
h15BAzB9398j3sH63sCg+6DxXNMgMxi8Go4bhj0ORQMn5S7oruP9is4zPMVNBWQozvuXYwuwVGovfOFR
FgZGOuw4bDoFomnZyHgAVZuK0QPho8gTaPU0V6e2saR7qydfZAIS5PVmnyR5slD0LW7R5CH67P/9DiIY
4+PHJhDajaAi6411GZscleuGkPsBCy0GnbGN409auvojRCct01o7UW+4sYHgI2QsqZZxgZSBb47gkt+Y
x93ktYnlSPwP0q+8AlqlNBpgsB1toIk+j344opImsCEJRLJvc1oTl1OSztTwB0QRanTpJQp23o76Ym3J
Q/Me+HyAxK+GUcI+WUy9sVowrgbpLbJUVqUnrrGi9WTnNVcMyQbC4JYmGZRnjkt1hZ8D31TGY9QRACzZ
EcmwlUP+eN1TI28MhKhVmZAUc72/QMPSr9xAQX17LWLwrAxoIs0UdX3Cv18cuw92IHxHh1o30w4oEgDH
BvufXrjw2J12akAKEiAuNTc6Rk5RXoqnsLe5x9nr8fj6ExwoOFuJneLkCyw1YHOBjI2VnMbR0t/m6vcK
EDpbcXbh5eny8wAAAAAAAAAAAAAAAAAAAAAAAAAAABYfMDs="
}
```

End-Entity Certificate:

```
 {
  "subject": "CN=MyServer",
  "issuer": "CN=MyPQC-RootCA",
  "valid_from": "2025-01-09T16:24:30.065847Z",
  "valid_to": "2025-04-09T16:24:30.065851Z",
  "pqc_public_key": "9Cw7cmAcfJBu1beY1AoQ4KNw6RYv19yYUGYRfqljkuoKLeNaxrcOAKFXvtJ
YzFQdf/fA2QUwmjPCOvJYFAgW3zcxI3mkcuFKzpk2q5OL8IuIBEyv1QXOEakYOCIqPiVb2UlbjehwU5B
pk+sRQGIuKfBuQVtVj2sXliOnkvQdm0eaE7tZ1UdJAUI7xolKeEKjaUYZcIUuUZTAT1cTsJiwSDijf3y
rjggP46dUFxG49iNMn2mPgzG8b2pWnlgSRCACuNuevdC526MLpdEtEoCZJzuFfEExOCia7jSG9YBDTJu
```

```
ftPMQGwsWS+o/bhoVaLg0isfL0NpPCQtrdewubPJqdYywtimc15cx94pBephuaAJGj1ox/WeWr4vEpfo
EP7pbVGXD/QuehkqDFciTg3kz5QEVEDVgNdRPd5PIJVWHFOx0CFGlGtR+jstxKJwP7IIfkgEGzmVW1tI
4mdcUf7RwvPIgTAMc+sq4+4JjwQe3irCIeZhVkKgvLvN1HlqYWtK1lHy+48My4AB7RaxnpCBbLCEDAAY
4vsmsz9Uq0VbH6nm5iDZ42UKWVja1kXLBxfthelnN7aeUDvoqhrBuItYC3fRwUFV+zWtXn5VRxwVK6GO
NDKE/txq6HvI2TrfDGncoYECyjCxv1Jd3tVIZ16I4+IFrLZFWd3bBUFJa32WG5VUn5EUcQoBcEiYQPrq
s8NY6sKuSsKZh45p8V7WNoTGGIVKcNVPK3ytuCLJBSEtcIWl8aoIyhdkCTYxAcBuwNioE2hReXTMhcZU
cz8MkvDlModB9ItWmRCq0jnZfbbeiFFc04clSUdKWiyh5uCejx9oHkjYPusC/tatXretlV9RcM8NDh+Q
mnVcz/eYtt7yJDqIBHOcTmUmu3SKB64NIHZA+48E79gQd+Ywm1IxI2EkXL5iG91dYv7MxWEO5yOseB3R
Oh9uXXjVW+QFjXxIcz9KEz9EO7ejF2oGEp5dxITs773y8qyVOxMND6EdDGfygfemRszACj/NCbmcx4oC
4qiddlMBT6fd/BfhEKMtcZIHMJJsFssQg8mifKaZnnWoEmrwCSDtqn8CGXhh0jxMXnzM23eOLs/FbO9x
mIKBVHbq3dfkjjADGcpwtZcphNXCYNfwZP8pLk7lnG7FziGC7Z3ILu0x6ttF/AxRbCWmo6UcTvMop6gh
Oy+vMCOOV1+GK2EM7eaWttoMwgtQp+hxropa7ohRIKjtpzGlJ7kFs1cwtEwm7+hrH6Ue+Vcm+L/mKpWh
OSYysNYZDp0JtCMQxOyAgUteBuEgfU+ckXvRl7TMl/+I60jSaakRT1OUbjie/UIwYuHgKsynFzDGwl4k
A7jWv7pI38acfEQs3Q5wCvvDGJlYIUHI4I3xM0Id/RWBReiYhH4F8Uvp+QuI+WLoLPgZscUW3SKjKzCY
8xJELEpbGm7KLLRDPoqoSoiSBqMccgXJTnRwtPWoHo6mtCtYT5eGyJ4Q7TIWglbJNG/ooGmNjFctqqJf
HXuVQo/bDx2gVCtAMvqfARRpIgcSI4Yqn5XqEZLpseKlrqYoTVXwJwLp2UmB9k2V003aoRfoaugd31NJ
ochoqsdV7uVwVW+eh9yCMq2AO6dOg6VeFmco2IoIpNKxw+5Med8zL81uiccwfr0G57ki1R+pEgKNIgpN
+Jfe3jqg5+Isi1seyNugrAUsWrtNxWPMCeuMa+MouYUFlxEcwBZRtk7mDrKYEV4lUYsDEjXVhTisdH3N
FLsDFAVCzimpBV1IERZdAkZbN5zHAumJBe3U+j7S0nDRw2LcGN5Vg3XmglyqJ0UedkZFLF4Fp9NR3tYr
OWUpWKONtRfa9yhNiRcdv2xqKM2lHhJqwI8eciTtUOOptKEXGCmTF7eRZpuPLx/S4rQdmcFtLvrKEGQs
vX9m5DRyYCdQ2nAjLiCcxLikfc6kWbtgk2QMLWHMHQ8AnwaA8w4yYqPoUHaGkGByCaZSr7dqwI6K/5xW
TivWYdatEYsFBLySYEGEUOUGjb2qtY5ueSZuY4lbP3rWkRxNsc2hJ+Ll1mrUXC9mtzodTOO/6lgwItys
c06xcVw8WgKDlKfckgd1GlO8rl5AgOEo=",
  "pqc_signature": "5OnAwXjdWzdmR95FEn3AZMLqqomnC2Ot268abArDDMt63Gaqsaq47vR0DC+L
HLHHPiHwON09aS4Ues/pSQZAg3EXty5/MCCblt5DyE1aggmK+/8nkU4Yzm/cQ5wY/j4pFytRDqpe5+Qc
vUHWFONThibho/8Bf1RheZ8bkeXKPmHywYCGBejBHjmlmiBKj1JKQOFuFEcViFSH0clBAYETXaePzMw6
EK9jyaHDVhAL29fk6HuCZyFUcDWe/aQH9vQgsukZt/TiZwhMLjh8Wqtsky7II7Vbh4D2BGCJ79A0l0Pr
SCNxXxqtt/x1WSO1JMRRZiYd+QZ3WCYOy+7M81HBLAQczNlWbAgQc0EWOkf56Bi/yfoohEKIz/jG3Fz9
xr2c8ynIRnInMOp/go4GNYpn2bp+br/aOb/T3gbc/CPdmiASf1yA7z9wpIKHrlCeCq3LYlMM1Gtg7JMp
UZTez0M5AVHVQBxObUj9fmMnc+72Fac5Ew7X13bLE19DB7n2dhXWwsG6kzixZ+X8abtst1Yp5THpzLJs
2j4emLUVGTcSwP5M7a8V2SPE+I1BfozXwoJfI8TtGlVK7NuTYgfM3HOOf7Fw8F1IFHEQJq4AZDh123DO
ImIS2y90goDns7BFIGI/1mM57gmPmqnejsxf9D77vtpNBX+hsU232kVM1LwIxdDlWuFDHOv0W1hyelNW
SsfYkA/jE9oaGJgqzdlLOjraMPsJRXpMoqcpVmd2oRM0og/1VfuD8AjL9FmuUAVqYwmYyiPWmo+K+hvc
FJcgU77rrM6d1GoWwb14HubVsFfluev2fuEg843W2IYhs8Rjagt7C1574uxpgC0VH44R1k3R2w7UV3yE
lr4ro4N+JBFnPAE0weNDOtvoszK7PtswFT+0aqtEuTJ9Sh6Zf4f1X8fUiQIIwuaWIEzfPt4WeUfrPuE2
rlimbfk9P34TSEzM3TVAdq0qao8yzurnvyGBjXexqEtfynhAkdqJg03bfQr8KrUXoHr+rK1bsXuWqNDA
Owr3nSSUHoTzqiDsBGeUZkdQgPrGKf2m47zLBqPPKOqOIiXPAiUMi1dJM/fbyvaPOWrlMhow7kiCpBmI
OIe0DObHK83c6QN1dkpm54EO2w5jJjkwQMzwE7ljm+7ey0chwcexKD3j6D+gD6V1BzQ9c7SsXT1Gw8uj
ekF88O/6JI2tedhCeYBtzgEIyELKwFjYgKjjGwpSrkWLW1HlCSk5HdBZhzIPMDx5bJ2++Ckd35eMdZCo
dcAkYeD3j07HFvPAS5ZKkjkbfAIHDVVBoj3tGTmHeqK0/tghQpME/1uu9ynTQaxI+30utHRTL9adOvFw
Nh6HVJ1skHNTFMo1i4CdS29+UjGGEDK7cE40aYyypZUtGeewXinWRd5nWw9JZ/RbC01HFyhEn6NDPbqa
O39A4NLh67nbuZ246jtdjKrNYuHdRnngaDmxv7lewBWAsKY81A1c94RgWMb0ALMVKWBd8QU4YoETUo32
h8cg85zdEbsjiLhn1da35HT+3dzes7joAsXaRs4DMbo5WWg5mCgdwrJYlx6O1ivXsVzlVNZgGz1Cvsn5
NQwX476MyM3nA4ZX6dIJ8+MoHcLS0xYrn/OAXeRJiVZ4TA0TduAR+T1RV4rVzhMuF5WdOTV+b0cuI0DQ
EJwc1o/EBlOqTpABGQq3BfDFxk6oHoBViNfZJsCgD0mrK7YsrDgV8hkgueweqDmldg5pYy65AU7ZVZZT
/xUCgwWnEQi1DbUrYwzfiUAP8Ewjm5rRnFIBlkgeapUtUDgWMkEhStF2xMeosjZJQsU5/lDNBLWQj889
//4eeoBewhSSWQAny+6sZ6E7KZjqABczBMs0eBDZIF2YDbkEgs6y7j9DFBSfCEAVuQGrJ+X1nDblZXHB
K4FZO8POCYAjHjK0xDzXKhUls+e4t9f1tRPB39FXV8Tl6IwWE7IfHIEUh+UAbqHSqn9+TmIS0GtSft3q
```

```
p7j6jYEODdP7gU4yNEOnr0GsLV6NnqFjb3R5Uz6VSMY46rMz933HHpd29BP9x0/LvlC29yis99396qYB
VIqna2pzFNQWG9owwx0+4ITLVWN9+3Dr97WFA/zOB6Hf2pVnk5MYJ5/nqejQeANR1SebFp/xFvGJ44zC
qiSQsjTffFVQ3DURGJF0XXOSopkqXvpOmWsfkrOxuqZ9rZ4snqOf6CNlCrL6rNC7OOGhmYvK520y+/LT
TyEtq/qsgLxN8zN5L5ZmJ/+ABi3Plq42GhBymBxFARmzqOiv7k1YBSCP+WZOlUqvRBCR8YCUlEIl93a7
ETI9YiOL/dSvLt3clkhEikhBFs8IqW/FAccIZtTU2jP3IgltJGJFSqmnY37fNNcDl8pchzrdcFJ5hndv
6U1SbRhWDvz2MgoDoV6PevXpoDEzm7lYqw692L62YAZ+Qfjd5Oj0qum5Jtw/4/nC9u3toRk71yTXT4g5
8Jil52tPJhDKxx6U3RvqIq3Z31BvGRZHU2ToKSjEcceRAjjpuSRdYKuFr8BwswJgld/9xC57hW7766qL
tP/frHp4lm8CT3qVsMhcrIlyo+913aV/DkDCkqYf6B4IqniMZb/3GYtZNT6oD2cjf5fMiq9InQ2wftY1
FvTi6oLHl3tq1UuXndIlJlGujJPNckCcwlSurV6K+Kk5zyMLGLWHmqvRO6RfJ75v7kt5KPFafILZVZ9C
6nwZ72el+Ef8qa3qNsUqDTmWZqnnqkXSbKSEBepPE8Mdsol3pWYj1EcbTkdGBUmT64lBiEbyHa9UvVuz
A1QPwDx5zbvuKH5LA4uTPJknnG5f+AHfDopbYNey1+pSn/VsTfMh32rm9xvjZ0cGw2br0/kOI7myBMtq
ZzX0NrQPhB4Vw3QGSInDwurFBa4OZRKd6DIwrL5niSTOCKXV3Ui62ye9DAsfZa5Kyz9T48vg5GmHbijg
NFJYbSEEG9jSjK8km6cRhq/j/JXB10wmlZMNcMkCODtkz5B3B7Wlt+tZc8ZFNC3FVBB3SGKSq+jdayvc
jIn38knHvjEGJ3niHocXHZqCoLXtCcE9CcYQkHfDtb2sxyAuwrdrd8xixvRsP1+4RNCB5TSIpXqvcLBF
mEPl66uTY1WrUlQZICNQbHV5e4u0rr/N5+8PHSwyNkNJaXDQIiM0TVR0lpe5w9Ll9gcMFSdFTGCcqdXu
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA8ZJjE="
}
```

```
/var/folders/tr/m7nnfyd94_jfxmdwqvwq7hg40000gn/T/ipykernel_28007/2788386571.py:5
8: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled
for removal in a future version. Use timezone-aware objects to represent
datetimes in UTC: datetime.datetime.now(datetime.UTC).
  "valid_from": datetime.utcnow().isoformat() + "Z",
/var/folders/tr/m7nnfyd94_jfxmdwqvwq7hg40000gn/T/ipykernel_28007/2788386571.py:5
9: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled
for removal in a future version. Use timezone-aware objects to represent
datetimes in UTC: datetime.datetime.now(datetime.UTC).
  "valid_to": (datetime.utcnow() + timedelta(days=validity_days)).isoformat() +
"Z",
```

[120]:

```python
##########################################################
# Verifying the PQC Certificate (Dilithium Signature)#
##########################################################

def verify_pqc_certificate(cert_data: dict, root_pub_key: bytes,␣
 ↪signature_scheme: str = "Dilithium2"):
    """
    Verify the 'pqc_signature' field using the root's public key.
    """
    try:
        signature_b64 = cert_data.pop("pqc_signature", None)
        if not signature_b64:
            raise ValueError("No signature found in certificate data.")

        message_bytes = json.dumps(cert_data, sort_keys=True).encode('utf-8')
        signature = base64.b64decode(signature_b64)

        #verification using Dilithium2
        is_valid = Dilithium2.verify(root_pub_key, message_bytes, signature)
        if is_valid:
            print("Certificate is VALID under PQC signature:", signature_scheme)
        else:
            print("Certificate is INVALID: Signature mismatch.")
    except Exception as e:
        print("Verification failed due to an error:", e)
    finally:
        cert_data["pqc_signature"] = signature_b64

# Verifying the root certificate (self-signed)
print("Verifying the Root CA certificate:")
verify_pqc_certificate(root_cert_signed, dilithium_public_key, "Dilithium2")

# Verify the end-entity certificate with the root CA's public key
print("\nVerifying the End-Entity certificate:")
verify_pqc_certificate(end_entity_cert_signed, dilithium_public_key,␣
 ↪"Dilithium2")
```

```
Verifying the Root CA certificate:
Certificate is VALID under PQC signature: Dilithium2

Verifying the End-Entity certificate:
Certificate is VALID under PQC signature: Dilithium2
```

0.0.1 1. Key Generation Section

[121]:

```
# Dilithium2 signature scheme initialized.

print(f"Public Key Length (Dilithium): {len(dilithium_public_key)} bytes")
print(f"Private Key Length (Dilithium): {len(dilithium_private_key)} bytes")
print(f"Signature Scheme: {Dilithium2.__class__.__name__}")
print("Dilithium2 signature scheme initialized.")
```

```
Public Key Length (Dilithium): 1312 bytes
Private Key Length (Dilithium): 2528 bytes
Signature Scheme: Dilithium
Dilithium2 signature scheme initialized.
```

[122]:

```
# KYBER

print(f"Kyber Public Key Length: {len(kyber_public_key)} bytes")
print(f"Kyber Private Key Length: {len(kyber_private_key)} bytes")
print("Signature Scheme: KYBER")
print("Kyber key encapsulation initialized.")
```

```
Kyber Public Key Length: 1568 bytes
Kyber Private Key Length: 3168 bytes
```

```
Signature Scheme: KYBER
Kyber key encapsulation initialized.
```

0.0.2 1. Authentication WITH DILITHIUM2

0.0.3 1.1 Function involved in Signing a Key with DILITHIUM2

[134]:
```
# Signing a key with Dilithium2

# a function to sign the message with dilithium
def dilithium_signature(dilithium_private_key, message):
    dilithium_signature = Dilithium2.sign(dilithium_private_key, message)
    print(f"It is ***SIGNED*** with Signature Length: {len(dilithium_signature)} bytes")
    return dilithium_signature
```

0.0.4 1.2 Function involved in Verifying with DILITHIUM2

[157]:
```
# a function to verify the signature
def dilithium_verify(pub_key, message, signature):
    if Dilithium2.verify(pub_key, message, signature):
        print("Dilithium2 signature ***VERIFIED*** successfully!")
    else:
        print("Signature ***verification failed!***")
```

[158]:
```
#example for signing and verifying the message

message = b"Post-Quantum CA Test Message"
signature= dilithium_signature(dilithium_private_key, message)
is_valid = dilithium_verify(dilithium_public_key, message, signature)
```

```
It is ***SIGNED*** with Signature Length: 2420 bytes
Dilithium2 signature ***VERIFIED*** successfully!
```

0.0.5 2. KEY EXCHANGE: ENCAP and DECAP with KYBER

0.0.6 2.1 Function to CREATE shared secret and cipher

[139]:
```
def create_secret_cipher(kyber_public_key):
    shared_secret, cipher= encrypt_session_key(kyber_public_key)
    print(f"Shared Secret Length: {len(shared_secret)} bytes")
    print(f"Cipher Length: {len(cipher)} bytes")
    return shared_secret, cipher
```

0.0.7 Now once the shared secret is created then Exchange the key using KYBER key exchange mechanism and share the cipher

0.0.8 2.2 Function to RECOVER SECRET

[146]:
```
def recovered_secret(kyber_private_key, cipher):
    recovered_secret = decrypt_session_key(kyber_private_key, cipher)
    print(f"Recovered Secret Length: {len(recovered_secret)} bytes")
    return recovered_secret
```

[147]:
```
def assert_secret(shared_secret, recovered_secret):
    assert shared_secret == recovered_secret, "Shared secret and Recovered␣
 ↪secret do not match!"
    print("Shared secret and Recovered secret match!")

print("Kyber shared secret **encapsulation and decapsulation**")
```

```
Kyber shared secret **encapsulation and decapsulation**
```

[148]:
```
#example

shared_secret, cipher = create_secret_cipher(kyber_public_key)
recovered_secret = recovered_secret(kyber_private_key, cipher)
assert_secret(shared_secret, recovered_secret)
```

```
Shared Secret Length: 32 bytes
Cipher Length: 1568 bytes
Recovered Secret Length: 32 bytes
Shared secret and Recovered secret match!
```

0.0.9 Now that Authenticaton key: Dilithium and Key exchange mechanism Kyber are established we move forward for Key Performance Testing.

0.0.10 2. Key Performance Testing

I am trying to measure handshake latency, throughput, and resource usage for: 1. Classical EAP-TLS (e.g., RSA/ECC). 2. PQ Device-Only (Kyber for key encapsulation, Dilithium for signing). 3.

0.0.11 A. Edge-Assisted Computation Flow

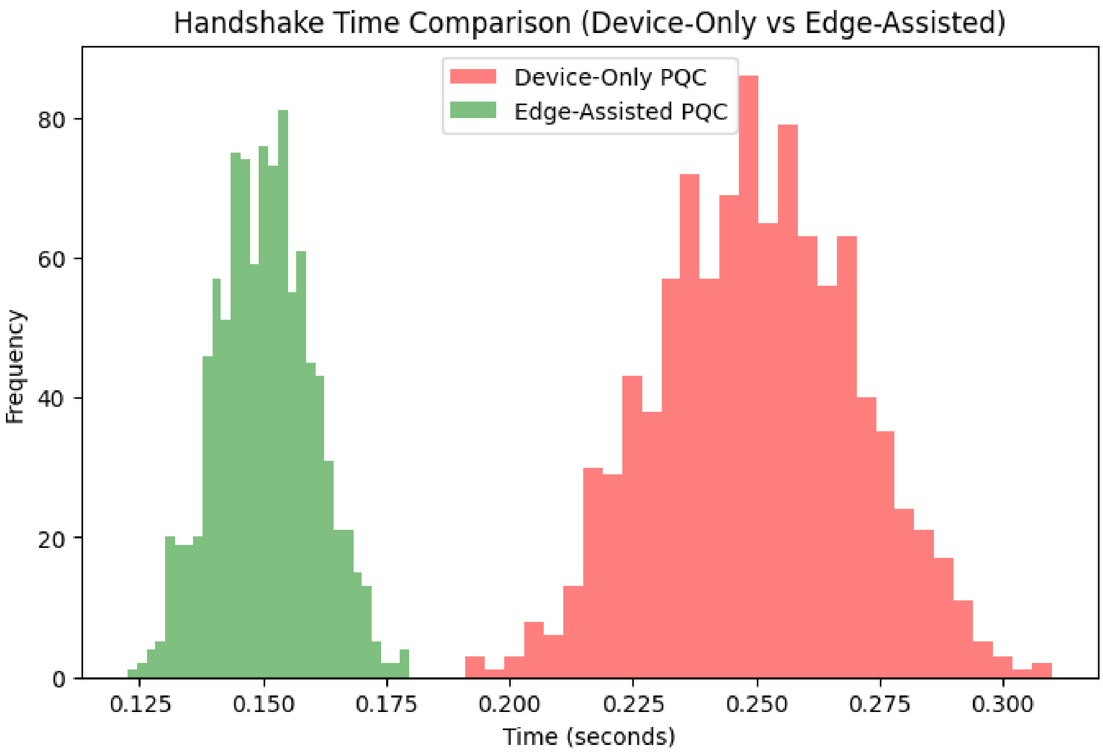
Goal: Simulate how much time and computation is saved when the edge server performs the bulk of PQC operations compared to the IoT device doing all the work.

[50]:
```
%pip install matplotlib

import numpy as np
import matplotlib.pyplot as plt

# Number of simulated IoT devices
```

```
num_devices = 1000

# Simulate handshake times (in seconds)
device_only_times = np.random.normal(0.25, 0.02, num_devices)  # Mean 250 ms␣
  ↪(device-only)
edge_offload_times = np.random.normal(0.15, 0.01, num_devices)  # Mean 150 ms␣
  ↪(edge-assisted)

# Plot the comparison
plt.figure(figsize=(8, 5))
plt.hist(device_only_times, alpha=0.5, label='Device-Only PQC', bins=30,␣
  ↪color='red')
plt.hist(edge_offload_times, alpha=0.5, label='Edge-Assisted PQC', bins=30,␣
  ↪color='green')
plt.title('Handshake Time Comparison (Device-Only vs Edge-Assisted)')
plt.xlabel('Time (seconds)')
plt.ylabel('Frequency')
plt.legend()
plt.show()

# Calculate the average time saved
avg_time_saved = np.mean(device_only_times) - np.mean(edge_offload_times)
print(f"Average Time Saved with Edge Offloading: {avg_time_saved:.4f} seconds")
```

```
Collecting matplotlib
  Using cached matplotlib-3.10.0-cp312-cp312-macosx_11_0_arm64.whl.metadata (11
kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Using cached contourpy-1.3.1-cp312-cp312-macosx_11_0_arm64.whl.metadata (5.4
kB)
Collecting cycler>=0.10 (from matplotlib)
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Using cached fonttools-4.55.3-cp312-cp312-macosx_10_13_universal2.whl.metadata
(165 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Using cached kiwisolver-1.4.8-cp312-cp312-macosx_11_0_arm64.whl.metadata (6.2
kB)
Requirement already satisfied: numpy>=1.23 in
/Users/abhisekjha/MyFolder/Github_Projects/futureg-quantum-
ca/venv/lib/python3.12/site-packages (from matplotlib) (2.2.1)
Requirement already satisfied: packaging>=20.0 in
/Users/abhisekjha/MyFolder/Github_Projects/futureg-quantum-
ca/venv/lib/python3.12/site-packages (from matplotlib) (24.0)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-11.1.0-cp312-cp312-macosx_11_0_arm64.whl.metadata (9.1 kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
```

```
  Downloading pyparsing-3.2.1-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: python-dateutil>=2.7 in
/Users/abhisekjha/MyFolder/Github_Projects/futureg-quantum-
ca/venv/lib/python3.12/site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in
/Users/abhisekjha/MyFolder/Github_Projects/futureg-quantum-
ca/venv/lib/python3.12/site-packages (from python-dateutil>=2.7->matplotlib)
(1.17.0)
Using cached matplotlib-3.10.0-cp312-cp312-macosx_11_0_arm64.whl (8.0 MB)
Using cached contourpy-1.3.1-cp312-cp312-macosx_11_0_arm64.whl (255 kB)
Using cached cycler-0.12.1-py3-none-any.whl (8.3 kB)
Using cached fonttools-4.55.3-cp312-cp312-macosx_10_13_universal2.whl (2.8 MB)
Using cached kiwisolver-1.4.8-cp312-cp312-macosx_11_0_arm64.whl (65 kB)
Downloading pillow-11.1.0-cp312-cp312-macosx_11_0_arm64.whl (3.1 MB)
                         3.1/3.1 MB
2.8 MB/s eta 0:00:00a 0:00:01
Downloading pyparsing-3.2.1-py3-none-any.whl (107 kB)
                         107.7/107.7 kB
2.7 MB/s eta 0:00:00a 0:00:01
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools,
cycler, contourpy, matplotlib
Successfully installed contourpy-1.3.1 cycler-0.12.1 fonttools-4.55.3
kiwisolver-1.4.8 matplotlib-3.10.0 pillow-11.1.0 pyparsing-3.2.1

[notice] A new release of pip is
available: 24.0 -> 24.3.1
[notice] To update, run:
pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
```



```
Average Time Saved with Edge Offloading: 0.0995 seconds
```

[164]:

```
import time
from dilithium_py.dilithium import Dilithium2  # Your PQC module

# Function to simulate PQC operations on an IoT device
def simulate_iot_device_operations():
    start_time = time.time()

    # Key generation
    # dilithium_private_key, dilithium_public_key = Dilithium2.keygen()

    message = b"Post-Quantum CA Test Message"
    signature= dilithium_signature(dilithium_private_key, message)
    is_valid = dilithium_verify(dilithium_public_key, message, signature)

    # Key encapsulation using Kyber
    ciphertext, shared_secret = create_secret_cipher(kyber_public_key)

    # assert_secret(shared_secret, recovered_secret(kyber_private_key,␣
 ↪ciphertext))
```

```
    end_time = time.time()
    time_taken = end_time - start_time
    print(f"IoT device computation time: {time_taken:.3f} seconds")
    return time_taken

# Function to simulate PQC operations on an edge server (faster computation)
def simulate_edge_server_operations():
    start_time = time.time()

     # Key generation
    # dilithium_private_key, dilithium_public_key = Dilithium2.keygen()

    message = b"Post-Quantum CA Test Message"
    signature= dilithium_signature(dilithium_private_key, message)
    is_valid = dilithium_verify(dilithium_public_key, message, signature)

    # Key encapsulation using Kyber
    ciphertext, shared_secret = create_secret_cipher(kyber_public_key)

    # assert_secret(shared_secret, recovered_secret(kyber_private_key, ciphertext))


    end_time = time.time()
    time_taken = end_time - start_time
    print(f"Edge server computation time: {time_taken:.3f} seconds")
    return time_taken

# Compare times
iot_time = simulate_iot_device_operations()
edge_server_time = simulate_edge_server_operations()

time_saved = iot_time - edge_server_time
percentage_improvement = (time_saved / iot_time) * 100

print(f"Time saved by offloading to edge server: {time_saved:.3f} seconds ({percentage_improvement:.2f}% improvement)")
```

```
It is ***SIGNED*** with Signature Length: 2420 bytes
Dilithium2 signature ***VERIFIED*** successfully!
Shared Secret Length: 32 bytes
Cipher Length: 1568 bytes
IoT device computation time: 0.043 seconds
It is ***SIGNED*** with Signature Length: 2420 bytes
Dilithium2 signature ***VERIFIED*** successfully!
Shared Secret Length: 32 bytes
Cipher Length: 1568 bytes
Edge server computation time: 0.040 seconds
```

```
Time saved by offloading to edge server: 0.004 seconds (8.27% improvement)
```

0.1 USING DOCKER to get EGDE and IOT comparision

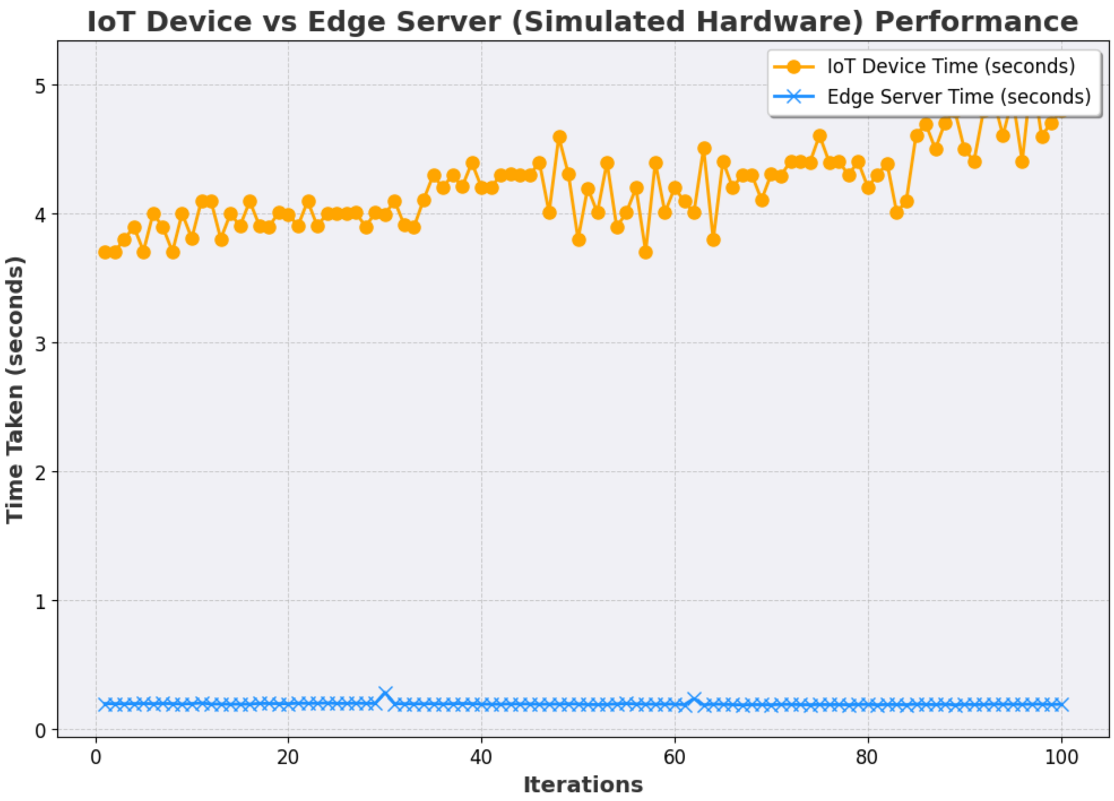
[345]:

```
import json
import matplotlib.pyplot as plt

# Directly use the absolute paths for the files
iot_results_path = "/Users/abhisekjha/MyFolder/Github_Projects/
 ↪futureg-quantum-ca/iot_results/smart_watch_results.json"

# iot_results_path = "/Users/abhisekjha/MyFolder/Github_Projects/
 ↪futureg-quantum-ca/iot_results/iot_results.json"
edge_results_path = "/Users/abhisekjha/MyFolder/Github_Projects/
 ↪futureg-quantum-ca/edge_results/edge_results.json"

# Load IoT and Edge results
with open(iot_results_path, "r") as f:
    iot_data = json.load(f)

with open(edge_results_path, "r") as f:
    edge_data = json.load(f)

print("IoT Data:", iot_data)
print("Edge Data:", edge_data)
```

```
IoT Data: {'iterations': 100, 'times': [[1, 3.7026073932647705], [2,
3.701878786087036], [3, 3.796459436416626], [4, 3.896939277648926], [5,
3.706294059753418], [6, 4.000692129135132], [7, 3.8933987617492676], [8,
3.7032155990600586], [9, 3.9998621940612793], [10, 3.8060662746429443], [11,
4.100102663040161], [12, 4.094214916229248], [13, 3.8025460243225098], [14,
3.9969677925109863], [15, 3.902538776397705], [16, 4.09399676322937], [17,
3.904853105545044], [18, 3.8990283012390137], [19, 4.006179094314575], [20,
3.992733955383301], [21, 3.9031424522399902], [22, 4.096055030822754], [23,
3.899892568588257], [24, 4.00013279914856], [25, 3.9985013008117676], [26,
4.002645015716553], [27, 4.005353689193726], [28, 3.8970108032226562], [29,
4.005146741867065], [30, 3.9936606884002686], [31, 4.096381664276123], [32,
3.910468578338623], [33, 3.892662763595581], [34, 4.104888677597046], [35,
4.295997142791748], [36, 4.19931960105896], [37, 4.299231290817261], [38,
4.206463813781738], [39, 4.392846345901489], [40, 4.2010109424591064], [41,
4.202601671218872], [42, 4.297470808029175], [43, 4.306508302688599], [44,
4.295767307281494], [45, 4.300514221191406], [46, 4.393949508666992], [47,
4.005544185638428], [48, 4.59240460395813], [49, 4.304861545562744], [50,
3.8014471530914307], [51, 4.19703483581543], [52, 4.005481958389282], [53,
4.396637678146362], [54, 3.893740653991699], [55, 4.006365060806274], [56,
4.197333812713623], [57, 3.7037181854248047], [58, 4.394512176513672], [59,
4.0061681270599365], [60, 4.198242664337158], [61, 4.0934202671051025], [62,
4.006398677825928], [63, 4.5047056674957275], [64, 3.7946438789367676], [65,
4.40449333190918], [66, 4.201554536819458], [67, 4.293876886367798], [68,
4.296973705291748], [69, 4.105252742767334], [70, 4.303713083267212], [71,
4.291173458099365], [72, 4.405004024505615], [73, 4.401947975158691], [74,
4.392736434936523], [75, 4.605201482772827], [76, 4.398755311965942], [77,
4.401961326599121], [78, 4.2994794845581055], [79, 4.400528430938721], [80,
4.205322742462158], [81, 4.296036005020142], [82, 4.389850378036499], [83,
4.005075216293335], [84, 4.100616216659546], [85, 4.6063392162323], [86,
4.69338583946228], [87, 4.498853921890259], [88, 4.704292058944702], [89,
4.800870656967163], [90, 4.499476909637451], [91, 4.403913736343384], [92,
4.801140069961548], [93, 4.890628814697266], [94, 4.604284048080444], [95,
4.896847248077393], [96, 4.403080463409424], [97, 5.0982983112335205], [98,
4.5998311042785645], [99, 4.701904773712158], [100, 4.793472528457642]]}
Edge Data: {'iterations': 100, 'times': [[1, 0.1993715763092041], [2,
0.19890546798706055], [3, 0.19808530807495117], [4, 0.19946742057800293], [5,
0.20006847381591797], [6, 0.1989424228668213], [7, 0.20043373107910156], [8,
0.19934868812561035], [9, 0.19599199295043945], [10, 0.19932126998901367], [11,
0.204268217086792], [12, 0.1957237720489502], [13, 0.19618844985961914], [14,
0.19459319114685059], [15, 0.19544410705566406], [16, 0.19495797157287598], [17,
0.2013874053955078], [18, 0.20025205612182617], [19, 0.19913768768310547], [20,
0.19860529899597168], [21, 0.20301485061645508], [22, 0.20166778564453125], [23,
0.20221638679504395], [24, 0.20445942878723145], [25, 0.20091772079467773], [26,
0.20124435424804688], [27, 0.20295238494873047], [28, 0.2014298439025879], [29,
0.20079445838928223], [30, 0.28562188148498535], [31, 0.1984252929687 5], [32,
0.1981658935546875], [33, 0.1977231502532959], [34, 0.1976611614227295], [35,
0.19887685775756836], [36, 0.19639158248901367], [37, 0.1987149715423584], [38,
0.19977617263793945], [39, 0.1991264820098877], [40, 0.19551396369934082], [41,
0.19481945037841797], [42, 0.19468188285827637], [43, 0.19720959663391113], [44,
0.19663381576538086], [45, 0.19667577743530273], [46, 0.19453811645507812], [47,
0.1952972412109375], [48, 0.19584345817565918], [49, 0.1976330280303955], [50,
0.19637322425842285], [51, 0.1925039291381836], [52, 0.19175171852111816], [53,
0.1932239532470703], [54, 0.19689440727233887], [55, 0.1998898983001709], [56,
0.19680452346801758], [57, 0.1953415870666504], [58, 0.1947925090789795], [59,
0.19572687149047852], [60, 0.1956169605255127], [61, 0.19081401824951172], [62,
0.23814129829406738], [63, 0.18921828269958496], [64, 0.19523310661315918], [65,
0.19417929649353027], [66, 0.19141554832458496], [67, 0.18909120559692383], [68,
0.19159531593322754], [69, 0.19111156463623047], [70, 0.19034862518310547], [71,
0.1932690143585205], [72, 0.19339752197265625], [73, 0.1912236213684082], [74,
0.19042038917541504], [75, 0.19237017631530762], [76, 0.19122767448425293], [77,
0.1923384666442871], [78, 0.19000792503356934], [79, 0.1928544044494629], [80,
0.19381046295166016], [81, 0.18970489501953125], [82, 0.19268798828125], [83,
0.19187259674072266], [84, 0.19071483612060547], [85, 0.19539642333984375], [86,
0.19294190406799316], [87, 0.19114112854003906], [88, 0.1950845718383789], [89,
0.18909955024719238], [90, 0.19385290145874023], [91, 0.19153189659118652], [92,
0.19254565238952637], [93, 0.1957247257232666], [94, 0.1945958137512207], [95,
0.19472932815551758], [96, 0.19367051124572754], [97, 0.1951732635498047], [98,
0.19561433792114258], [99, 0.19266295433044434], [100, 0.19439387321472168]]}
```

[351]:

```
import matplotlib.pyplot as plt

# Extract data
iot_iterations = [data[0] for data in iot_data['times']]
iot_times = [data[1] for data in iot_data['times']]

edge_iterations = [data[0] for data in edge_data['times']]
edge_times = [data[1] for data in edge_data['times']]

# Plot
# Plot with enhanced design using only Matplotlib
plt.figure(figsize=(12, 8))
plt.plot(iot_iterations, iot_times, label='IoT Device Time (seconds)',␣
 ↪marker='o', markersize=8, linewidth=2, color='orange')
plt.plot(edge_iterations, edge_times, label='Edge Server Time (seconds)',␣
 ↪marker='x', markersize=8, linewidth=2, color='dodgerblue')

# Labels and Title
plt.xlabel('Iterations', fontsize=14, fontweight='bold', color='#333333')
plt.ylabel('Time Taken (seconds)', fontsize=14, fontweight='bold',␣
 ↪color='#333333')
plt.title('IoT Device vs Edge Server (Simulated Hardware) Performance',␣
 ↪fontsize=18, fontweight='bold', color='#333333')

plt.legend(fontsize=12, loc='upper right', frameon=True, shadow=True,␣
 ↪fancybox=True)

plt.grid(True, which='major', linestyle='--', linewidth=0.7, alpha=0.6)

plt.gca().set_facecolor('#f0f0f5')

plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()

# Calculate the average time saved
avg_iot_time = sum(iot_times) / len(iot_times)
avg_edge_time = sum(edge_times) / len(edge_times)

avg_time_saved = avg_iot_time - avg_edge_time
print(f"Average Time Saved with Edge Offloading: {avg_time_saved:.4f} seconds")

# Calculate the percentage improvement
percentage_improvement = (avg_time_saved / avg_iot_time) * 100
print(f"Percentage Improvement: {percentage_improvement:.2f}%")
```



```
Average Time Saved with Edge Offloading: 4.0078 seconds
Percentage Improvement: 95.31%
```

[ ]:

0.1.1 B. Session Resumption Mechanism

Goal: Demonstrate how session caching/resumption can avoid full handshakes during repeated connections, reducing handshake times significantly.

0.1.2 C Energy Profiling and Resource Constraints

Goal: Simulate the energy consumption for different scenarios:

1. Device-only PQC (no optimization)
2. Edge-assisted PQC (partial offloading)
3. Session resumption (no full handshake)

0.1.3 Session Resumption Mechanism

0.1.4 D. Decentralized PKI Throughput

Goal: Simulate how a decentralized PKI (multiple CAs) can handle certificate issuance and renewals more efficiently than a centralized system.

0.1.5 E. Latency vs. Scalability Analysis

Goal: Simulate how handshake latency changes as the number of concurrent IoT devices increases.

1. Plot handshake times as a function of the number of devices connected.
2. Compare the scalability of device-only, edge-assisted, and centralized vs. decentralized PKI.

1. Implement Session Cache for Resumption:

0.1.6 Latency vs Scalability Analysis

1. Simulate Concurrent IoT Connections:

[69]:

```python
import threading

def handshake_simulation(device_id):
    start_time = time.time()
    # Perform PQC handshake
    shared_secret, cipher = encrypt_session_key(kem_public_key)
    print(f"Device {device_id}: {time.time() - start_time:.4f} seconds")

threads = [threading.Thread(target=handshake_simulation, args=(i,)) for i in␣
 ↪range(1000)]
for thread in threads:
    thread.start()
for thread in threads:
    thread.join()
```

```
Device 0: 0.0911 seconds
Device 9: 0.1333 seconds
Device 2: 0.1670 seconds
Device 7: 0.1899 seconds
Device 6: 0.1977 seconds
Device 1: 0.2469 seconds
Device 5: 0.2394 seconds
Device 10: 0.2721 seconds
Device 11: 0.2772 seconds
Device 13: 0.3150 seconds
Device 14: 0.3349 seconds
Device 17: 0.3957 seconds
Device 4: 0.4359 seconds
Device 21: 0.3877 seconds
Device 16: 0.4335 seconds
Device 19: 0.4226 seconds
Device 8: 0.4794 seconds
Device 3: 0.4900 seconds
Device 20: 0.4860 seconds
Device 18: 0.5031 seconds
Device 12: 0.6283 seconds
Device 32: 0.3829 seconds
Device 27: 0.5667 seconds
Device 23: 0.6120 seconds
Device 31: 0.5105 seconds
Device 15: 0.7712 seconds
Device 22: 0.6876 seconds
Device 28: 0.6204 seconds
Device 25: 0.6963 seconds
Device 26: 0.6740 seconds
Device 30: 0.5786 seconds
Device 42: 0.5107 seconds
Device 43: 0.5282 seconds
Device 33: 0.6395 seconds
Device 24: 0.8229 seconds
Device 38: 0.6344 seconds
Device 48: 0.5077 seconds
Device 39: 0.6665 seconds
Device 35: 0.7317 seconds
Device 37: 0.7075 seconds
Device 36: 0.7384 seconds
Device 41: 0.6891 seconds
Device 49: 0.5824 seconds
Device 45: 0.6517 seconds
Device 44: 0.7167 seconds
Device 29: 0.9085 secondsDevice 51: 0.6552 seconds

Device 40: 0.8277 seconds
Device 52: 0.7076 seconds
Device 53: 0.7700 seconds
Device 47: 0.7917 seconds
Device 46: 0.8153 seconds
Device 50: 0.7971 seconds
Device 54: 0.8211 seconds
Device 55: 0.7342 seconds
Device 34: 1.0710 seconds
Device 61: 0.3169 seconds
Device 57: 0.6618 seconds
Device 62: 0.3913 secondsDevice 58: 0.5614 seconds
Device 63: 0.3842 seconds

Device 56: 0.8398 seconds
Device 72: 0.3388 seconds
Device 66: 0.5050 seconds
Device 70: 0.4383 seconds
Device 67: 0.5359 seconds
Device 68: 0.5201 seconds
Device 73: 0.4450 seconds
Device 59: 0.6653 seconds
Device 65: 0.5957 seconds
Device 80: 0.4522 seconds
Device 60: 0.7533 seconds
Device 78: 0.5236 seconds
Device 77: 0.5674 seconds
Device 75: 0.5815 seconds
Device 84: 0.4686 seconds
Device 69: 0.6569 seconds
Device 79: 0.5845 seconds
Device 83: 0.5038 seconds
Device 82: 0.5475 seconds
Device 85: 0.5079 seconds
Device 76: 0.6841 seconds
Device 64: 0.8698 seconds
Device 91: 0.4844 seconds
Device 74: 0.7302 seconds
Device 71: 0.8070 seconds
Device 86: 0.6155 seconds
Device 81: 0.6994 seconds
Device 90: 0.6111 seconds
Device 94: 0.5868 secondsDevice 97: 0.4996 seconds
Device 92: 0.6607 seconds
Device 87: 0.7320 seconds
Device 88: 0.7332 seconds
Device 96: 0.5802 seconds

Device 95: 0.6371 seconds
Device 93: 0.7138 seconds
Device 98: 0.5287 seconds
Device 99: 0.4191 seconds
Device 89: 0.8192 seconds
Device 100: 0.4129 seconds
Device 104: 0.3765 seconds
Device 109: 0.3116 seconds
Device 102: 0.4571 seconds
Device 110: 0.3548 seconds
Device 108: 0.4135 secondsDevice 103: 0.4899 seconds
Device 105: 0.4775 seconds

Device 106: 0.4880 seconds
Device 107: 0.5180 seconds
Device 113: 0.3197 seconds
Device 111: 0.4668 seconds
Device 114: 0.3520 seconds
Device 117: 0.3224 seconds
Device 121: 0.2874 seconds
Device 115: 0.4553 seconds
Device 122: 0.3511 seconds
Device 101: 0.7787 seconds
Device 123: 0.3324 seconds
Device 125: 0.2902 seconds
Device 118: 0.5151 seconds
Device 119: 0.4897 seconds
Device 116: 0.6243 seconds
Device 120: 0.5148 seconds
Device 112: 0.6601 seconds
Device 126: 0.4128 seconds
Device 127: 0.4434 seconds
Device 130: 0.4301 seconds
Device 129: 0.5431 seconds
Device 134: 0.4654 seconds
Device 132: 0.4912 seconds
Device 128: 0.5849 seconds
Device 133: 0.5175 seconds
Device 135: 0.5141 seconds
Device 138: 0.4618 seconds
Device 131: 0.5890 seconds
Device 124: 0.6647 seconds
Device 139: 0.4944 seconds
Device 147: 0.3779 seconds
Device 140: 0.5752 seconds
Device 151: 0.3957 seconds
Device 141: 0.5591 seconds
Device 136: 0.7027 seconds
Device 142: 0.5695 seconds
Device 146: 0.5242 secondsDevice 150: 0.5288 seconds
Device 144: 0.5646 seconds
Device 153: 0.5147 seconds
Device 148: 0.5960 seconds

Device 137: 0.8356 seconds
Device 143: 0.7064 seconds
Device 156: 0.5362 seconds
Device 149: 0.6538 seconds
Device 152: 0.6490 seconds
Device 163: 0.4817 seconds
Device 155: 0.6597 seconds
Device 171: 0.3794 seconds
Device 154: 0.7578 seconds
Device 161: 0.6047 seconds
Device 145: 0.9292 seconds
Device 158: 0.7441 seconds
Device 159: 0.7265 seconds
Device 165: 0.5998 secondsDevice 160: 0.7292 seconds
Device 173: 0.4762 seconds

Device 162: 0.7107 seconds
Device 166: 0.5853 seconds
Device 164: 0.6903 seconds
Device 168: 0.6237 seconds
Device 177: 0.5204 seconds
Device 172: 0.6726 seconds
Device 183: 0.4974 seconds
Device 178: 0.5780 seconds
Device 157: 0.9849 seconds
Device 185: 0.5212 seconds
Device 170: 0.7560 seconds
Device 174: 0.7145 seconds
Device 167: 0.7975 seconds
Device 169: 0.8593 seconds
Device 184: 0.6307 seconds
Device 180: 0.7287 seconds
Device 176: 0.8358 seconds
Device 187: 0.6646 seconds
Device 182: 0.7244 seconds
Device 186: 0.7158 seconds
Device 188: 0.5794 seconds
Device 175: 0.9227 seconds
Device 179: 0.8887 seconds
Device 190: 0.6609 seconds
Device 192: 0.6727 seconds
Device 191: 0.6880 seconds
Device 181: 0.9590 seconds
Device 193: 0.5978 seconds
Device 198: 0.4957 seconds
Device 189: 0.7404 seconds
Device 194: 0.5925 seconds
Device 195: 0.6078 seconds
Device 199: 0.5024 seconds
Device 200: 0.5378 seconds
Device 204: 0.4082 seconds
Device 197: 0.6490 seconds
Device 208: 0.4451 seconds
Device 207: 0.5089 seconds
Device 196: 0.8275 seconds
Device 202: 0.6623 seconds
Device 203: 0.6321 seconds
Device 214: 0.4534 secondsDevice 201: 0.7915 seconds
Device 206: 0.6581 seconds

Device 220: 0.4318 seconds
Device 213: 0.5300 seconds
Device 231: 0.4662 seconds
Device 222: 0.5443 seconds
Device 210: 0.7603 seconds
Device 226: 0.5617 seconds
Device 205: 0.8893 seconds
Device 236: 0.5805 seconds
Device 230: 0.6276 seconds
Device 229: 0.6580 seconds
Device 209: 0.9344 seconds
Device 212: 0.9172 seconds
Device 216: 0.7691 seconds
Device 224: 0.7462 seconds
Device 211: 0.9910 seconds
Device 225: 0.8216 seconds
Device 217: 0.8662 seconds
Device 227: 0.8214 seconds
Device 243: 0.7037 seconds
Device 219: 0.8500 seconds
Device 218: 0.9409 seconds
Device 248: 0.6976 seconds
Device 244: 0.7675 seconds
Device 223: 0.9321 secondsDevice 234: 0.8859 seconds

Device 233: 0.9273 seconds
Device 235: 0.9439 seconds
Device 232: 0.9524 seconds
Device 238: 0.9019 seconds
Device 239: 0.9165 seconds
Device 240: 0.9315 seconds
Device 250: 0.8609 seconds
Device 242: 0.9619 seconds
Device 252: 0.7632 seconds
Device 247: 0.9413 seconds
Device 228: 1.1557 seconds
Device 241: 1.0353 seconds
Device 246: 1.0134 seconds
Device 221: 1.2053 seconds
Device 237: 1.1633 seconds
Device 257: 0.7137 seconds
Device 215: 1.2774 seconds
Device 245: 1.0991 seconds
Device 249: 1.1064 seconds
Device 256: 0.8926 seconds
Device 262: 0.7315 seconds
Device 255: 1.0354 seconds
Device 264: 0.6747 seconds
Device 258: 0.9549 seconds
Device 260: 0.9203 seconds
Device 254: 1.1012 seconds
Device 269: 0.5408 seconds
Device 253: 1.1056 seconds
Device 261: 0.9541 seconds
Device 251: 1.2888 seconds
Device 267: 0.7339 seconds
Device 274: 0.5587 seconds
Device 279: 0.4623 seconds
Device 273: 0.6615 seconds
Device 263: 0.8911 seconds
Device 280: 0.4926 seconds
Device 281: 0.5313 seconds
Device 265: 0.9479 seconds
Device 268: 0.8680 seconds
Device 278: 0.7031 seconds
Device 271: 0.8137 seconds
Device 287: 0.5930 seconds
Device 283: 0.6233 seconds
Device 289: 0.6104 seconds
Device 293: 0.6293 seconds
Device 259: 1.3705 seconds
Device 266: 1.1007 seconds
Device 284: 0.7336 seconds
Device 272: 0.9544 seconds
Device 288: 0.7077 seconds
Device 294: 0.6608 seconds
Device 286: 0.7735 seconds
Device 285: 0.7835 seconds
Device 270: 1.0824 seconds
Device 282: 0.8517 seconds
Device 277: 0.9872 seconds
Device 298: 0.6087 seconds
Device 275: 1.0891 seconds
Device 297: 0.6719 seconds
Device 292: 0.8937 seconds
Device 290: 0.9742 seconds
Device 296: 0.8357 seconds
Device 315: 0.2858 seconds
Device 302: 0.6394 seconds
Device 300: 0.7356 seconds
Device 303: 0.6494 seconds
Device 295: 1.0204 seconds
Device 291: 1.0606 seconds
Device 276: 1.2804 seconds
Device 299: 0.8034 seconds
Device 312: 0.4734 seconds
Device 306: 0.6591 seconds
Device 307: 0.6714 seconds
Device 305: 0.7238 seconds
Device 308: 0.6712 seconds
Device 309: 0.6830 seconds
Device 301: 0.9184 seconds
Device 310: 0.6786 seconds
Device 311: 0.7023 seconds
Device 320: 0.5565 seconds
Device 313: 0.6899 seconds
Device 317: 0.6154 secondsDevice 323: 0.5177 seconds
Device 316: 0.6399 seconds
Device 304: 1.0460 seconds

Device 327: 0.4396 seconds
Device 329: 0.4043 seconds
Device 328: 0.4301 seconds
Device 318: 0.7604 seconds
Device 324: 0.6364 seconds
Device 332: 0.5053 seconds
Device 321: 0.7192 seconds
Device 326: 0.6405 seconds
Device 331: 0.5183 seconds
Device 319: 0.8200 seconds
Device 314: 0.9669 seconds
Device 322: 0.8091 seconds
Device 336: 0.4197 seconds
Device 344: 0.2713 seconds
Device 335: 0.4874 seconds
Device 325: 0.8453 seconds
Device 334: 0.5798 seconds
Device 330: 0.7407 seconds
Device 337: 0.4731 seconds
Device 341: 0.4718 seconds
Device 333: 0.7435 seconds
Device 338: 0.5207 seconds
Device 343: 0.4573 seconds
Device 345: 0.4852 seconds
Device 348: 0.4749 seconds
Device 347: 0.4999 seconds
Device 352: 0.3307 seconds
Device 340: 0.6576 seconds
Device 346: 0.5843 seconds
Device 349: 0.4591 seconds
Device 350: 0.4497 seconds
Device 339: 0.7531 seconds
Device 351: 0.4838 seconds
Device 342: 0.7786 seconds
Device 360: 0.3790 seconds
Device 364: 0.3046 seconds
Device 356: 0.5090 seconds
Device 357: 0.4604 seconds
Device 355: 0.5386 seconds
Device 353: 0.6094 seconds
Device 366: 0.4126 seconds
Device 354: 0.6321 seconds
Device 359: 0.5312 seconds
Device 361: 0.5277 seconds
Device 365: 0.4683 seconds
Device 358: 0.6101 seconds
Device 362: 0.5061 seconds
Device 372: 0.3778 seconds
Device 363: 0.5725 seconds
Device 373: 0.4300 seconds
Device 369: 0.5477 seconds
Device 371: 0.5517 seconds
Device 368: 0.5914 seconds
Device 367: 0.6672 seconds
Device 370: 0.5908 seconds
Device 377: 0.4559 seconds
Device 380: 0.3648 seconds
Device 378: 0.4545 seconds
Device 375: 0.5537 seconds
Device 374: 0.6410 seconds
Device 387: 0.3745 seconds
Device 376: 0.5574 seconds
Device 390: 0.3673 seconds
Device 385: 0.4699 seconds
Device 389: 0.4643 seconds
Device 382: 0.4995 seconds
Device 393: 0.4298 seconds
Device 388: 0.5199 secondsDevice 383: 0.5731 seconds

Device 391: 0.5630 seconds
Device 379: 0.7182 seconds
Device 394: 0.4665 seconds
Device 386: 0.5936 seconds
Device 392: 0.5758 seconds
Device 405: 0.2817 seconds
Device 381: 0.7860 seconds
Device 384: 0.7408 seconds
Device 397: 0.4701 seconds
Device 396: 0.5323 seconds
Device 395: 0.5705 seconds
Device 403: 0.3820 seconds
Device 410: 0.3446 seconds
Device 398: 0.5838 seconds
Device 409: 0.4708 seconds
Device 399: 0.6495 seconds
Device 402: 0.6536 seconds
Device 407: 0.5985 seconds
Device 411: 0.5225 seconds
Device 406: 0.6117 seconds
Device 404: 0.6224 seconds
Device 400: 0.7768 seconds
Device 417: 0.5469 seconds
Device 401: 0.8030 seconds
Device 415: 0.5852 seconds
Device 429: 0.4087 seconds
Device 418: 0.6206 seconds
Device 413: 0.6839 seconds
Device 419: 0.6228 seconds
Device 414: 0.6545 seconds
Device 420: 0.6385 seconds
Device 412: 0.7702 seconds
Device 416: 0.7661 seconds
Device 408: 0.9219 seconds
Device 433: 0.5922 secondsDevice 432: 0.5994 seconds

Device 424: 0.7627 seconds
Device 425: 0.7394 seconds
Device 427: 0.6807 seconds
Device 422: 0.8008 seconds
Device 430: 0.7230 seconds
Device 428: 0.7397 seconds
Device 431: 0.7108 seconds
Device 434: 0.7163 seconds
Device 437: 0.7462 seconds
Device 421: 1.0054 seconds
Device 423: 1.0086 seconds
Device 440: 0.6849 seconds
Device 426: 0.9836 seconds
Device 435: 0.8973 seconds
Device 441: 0.6576 seconds
Device 447: 0.5669 seconds
Device 438: 0.7837 seconds
Device 442: 0.7001 seconds
Device 436: 0.9770 seconds
Device 451: 0.6114 seconds
Device 449: 0.6761 seconds
Device 445: 0.7216 seconds
Device 466: 0.3023 seconds
Device 455: 0.5051 seconds
Device 448: 0.7310 seconds
Device 443: 0.8715 seconds
Device 446: 0.8107 seconds
Device 452: 0.7430 seconds
Device 458: 0.6077 seconds
Device 454: 0.7559 seconds
Device 450: 0.8312 seconds
Device 444: 0.9561 seconds
Device 439: 1.1051 seconds
Device 457: 0.6740 seconds
Device 453: 0.8328 seconds
Device 462: 0.6070 seconds
Device 469: 0.5619 seconds
Device 468: 0.5634 seconds
Device 456: 0.8250 seconds
Device 465: 0.6846 seconds
Device 473: 0.5037 seconds
Device 461: 0.7876 seconds
Device 460: 0.8243 seconds
Device 463: 0.7505 seconds
Device 464: 0.7589 seconds
Device 467: 0.7168 seconds
Device 471: 0.7072 seconds
Device 476: 0.4783 seconds
Device 472: 0.7357 seconds
Device 475: 0.5623 seconds
Device 477: 0.5513 seconds
Device 478: 0.5304 seconds
Device 481: 0.5168 seconds
Device 459: 1.1243 seconds
Device 482: 0.4920 seconds
Device 470: 0.9283 seconds
Device 474: 0.7690 seconds
Device 480: 0.5944 seconds
Device 483: 0.5679 seconds
Device 491: 0.3169 seconds
Device 488: 0.4261 seconds
Device 479: 0.6736 seconds
Device 490: 0.5015 seconds
Device 486: 0.6346 seconds
Device 484: 0.7532 secondsDevice 487: 0.6736 seconds
Device 499: 0.4214 seconds

Device 495: 0.5899 seconds
Device 494: 0.6005 seconds
Device 485: 0.8637 seconds
Device 497: 0.6116 seconds
Device 492: 0.6628 seconds
Device 493: 0.6442 seconds
Device 500: 0.5456 seconds
Device 496: 0.6500 seconds
Device 489: 0.8342 seconds
Device 504: 0.5547 seconds
Device 511: 0.4814 seconds
Device 510: 0.5144 seconds
Device 516: 0.4066 seconds
Device 505: 0.6076 seconds
Device 503: 0.6344 seconds
Device 502: 0.6655 seconds
Device 512: 0.6311 secondsDevice 508: 0.6529 seconds
Device 501: 0.8034 seconds

Device 506: 0.7535 seconds
Device 513: 0.6340 seconds
Device 498: 0.8890 seconds
Device 522: 0.5260 seconds
Device 509: 0.8439 seconds
Device 507: 0.9018 seconds
Device 521: 0.7013 seconds
Device 527: 0.5307 seconds
Device 531: 0.3673 seconds
Device 525: 0.5561 seconds
Device 520: 0.7638 seconds
Device 523: 0.7320 seconds
Device 524: 0.6689 secondsDevice 515: 0.9195 seconds
Device 526: 0.6671 seconds
Device 519: 0.8731 seconds
Device 528: 0.6508 seconds

Device 517: 0.9235 seconds
Device 514: 0.9715 seconds
Device 529: 0.5751 seconds
Device 518: 0.9815 seconds
Device 530: 0.6435 seconds
Device 544: 0.5088 seconds
Device 535: 0.6504 seconds
Device 538: 0.6691 seconds
Device 539: 0.6543 seconds
Device 532: 0.7432 secondsDevice 534: 0.7137 seconds

Device 540: 0.6823 seconds
Device 533: 0.7588 seconds
Device 547: 0.5669 seconds
Device 543: 0.6595 seconds
Device 542: 0.7100 seconds
Device 536: 0.8500 seconds
Device 541: 0.8148 seconds
Device 551: 0.6463 seconds
Device 548: 0.6793 seconds
Device 537: 0.8993 seconds
Device 549: 0.7084 seconds
Device 550: 0.6969 seconds
Device 545: 0.8199 seconds
Device 559: 0.4641 seconds
Device 553: 0.6487 seconds
Device 546: 0.8566 seconds
Device 561: 0.5120 seconds
Device 557: 0.5838 seconds
Device 556: 0.5897 seconds
Device 552: 0.7273 seconds
Device 560: 0.6546 seconds
Device 562: 0.6288 secondsDevice 554: 0.7204 seconds

Device 566: 0.4674 seconds
Device 563: 0.5784 seconds
Device 555: 0.7491 seconds
Device 564: 0.5729 seconds
Device 558: 0.7499 seconds
Device 569: 0.4218 seconds
Device 565: 0.6201 seconds
Device 567: 0.4880 seconds
Device 571: 0.4195 seconds
Device 577: 0.2888 seconds
Device 578: 0.3547 seconds
Device 573: 0.3972 seconds
Device 570: 0.5505 seconds
Device 568: 0.6321 seconds
Device 575: 0.3952 seconds
Device 572: 0.5383 seconds
Device 576: 0.4604 seconds
Device 579: 0.4247 seconds
Device 583: 0.3899 seconds
Device 586: 0.3542 seconds
Device 574: 0.5689 seconds
Device 584: 0.4073 seconds
Device 581: 0.5195 seconds
Device 592: 0.3850 seconds
Device 587: 0.4425 seconds
Device 588: 0.4501 seconds
Device 582: 0.6339 seconds
Device 591: 0.5089 seconds
Device 580: 0.6668 seconds
Device 593: 0.4775 seconds
Device 594: 0.4261 seconds
Device 589: 0.5886 seconds
Device 597: 0.4103 seconds
Device 590: 0.5991 seconds
Device 596: 0.4541 seconds
Device 585: 0.7156 seconds
Device 595: 0.5109 seconds
Device 600: 0.3941 seconds
Device 599: 0.4544 seconds
Device 598: 0.5331 seconds
Device 603: 0.5058 seconds
Device 601: 0.5308 seconds
Device 614: 0.2950 seconds
Device 602: 0.5880 seconds
Device 613: 0.3449 seconds
Device 606: 0.4934 seconds
Device 604: 0.6694 seconds
Device 616: 0.3800 seconds
Device 610: 0.5404 seconds
Device 617: 0.4749 seconds
Device 607: 0.6349 seconds
Device 605: 0.7505 seconds
Device 611: 0.6264 seconds
Device 627: 0.4708 seconds
Device 609: 0.6543 seconds
Device 624: 0.5294 seconds
Device 620: 0.5726 seconds
Device 623: 0.5877 seconds
Device 608: 0.7774 seconds
Device 612: 0.7121 seconds
Device 632: 0.6101 seconds
Device 618: 0.7070 seconds
Device 628: 0.6591 seconds
Device 631: 0.6340 seconds
Device 622: 0.6906 seconds
Device 621: 0.7534 seconds
Device 615: 0.7962 seconds
Device 634: 0.6535 seconds
Device 640: 0.5368 seconds
Device 619: 0.8174 seconds
Device 625: 0.7798 seconds
Device 629: 0.7672 seconds
Device 633: 0.7256 seconds
Device 644: 0.4854 seconds
Device 630: 0.8288 seconds
Device 643: 0.6083 seconds
Device 641: 0.6264 seconds
Device 636: 0.8079 seconds
Device 626: 0.9691 seconds
Device 642: 0.6965 seconds
Device 638: 0.8272 seconds
Device 650: 0.4539 seconds
Device 635: 0.9417 secondsDevice 637: 0.9149 seconds
Device 645: 0.6877 seconds
Device 647: 0.6108 seconds
Device 649: 0.5858 seconds
Device 659: 0.3281 seconds
Device 660: 0.3751 seconds
Device 655: 0.5951 seconds
Device 652: 0.6624 seconds
Device 651: 0.7330 secondsDevice 654: 0.6314 seconds
Device 648: 0.8199 seconds
Device 639: 1.1537 seconds
Device 653: 0.6628 seconds
Device 646: 0.8858 seconds
Device 656: 0.6602 seconds

Device 666: 0.5392 seconds
Device 658: 0.6695 seconds
Device 673: 0.4544 seconds
Device 664: 0.5920 seconds
Device 668: 0.6135 seconds
Device 657: 0.8700 seconds
Device 665: 0.6996 seconds
Device 662: 0.7626 seconds
Device 676: 0.6047 seconds
Device 661: 0.8434 seconds
Device 677: 0.6058 seconds
Device 679: 0.6198 seconds
Device 669: 0.7412 seconds
Device 672: 0.7179 seconds
Device 675: 0.6949 seconds
Device 667: 0.8167 seconds
Device 671: 0.8219 seconds
Device 674: 0.7730 seconds
Device 663: 0.9324 seconds
Device 683: 0.7406 seconds
Device 670: 0.8941 seconds
Device 678: 0.7786 seconds
Device 685: 0.7137 seconds
Device 692: 0.5077 seconds
Device 687: 0.7070 seconds
Device 681: 0.8239 seconds
Device 688: 0.7050 seconds
Device 694: 0.5857 seconds
Device 684: 0.8901 secondsDevice 696: 0.6054 seconds
Device 682: 0.9672 seconds

Device 691: 0.7207 seconds
Device 690: 0.7495 seconds
Device 680: 1.0469 seconds
Device 693: 0.7673 seconds
Device 695: 0.7395 seconds
Device 705: 0.5276 seconds
Device 702: 0.5669 seconds
Device 700: 0.6472 seconds
Device 686: 1.0590 seconds
Device 701: 0.6461 seconds
Device 698: 0.7252 seconds
Device 704: 0.6585 seconds
Device 703: 0.7021 seconds
Device 709: 0.5485 seconds
Device 689: 1.1072 seconds
Device 706: 0.7391 seconds
Device 714: 0.5789 seconds
Device 710: 0.6575 seconds
Device 699: 0.9520 seconds
Device 697: 1.0635 seconds
Device 707: 0.7269 seconds
Device 723: 0.5044 seconds
Device 726: 0.4991 seconds
Device 724: 0.5328 seconds
Device 720: 0.6771 seconds
Device 711: 0.7796 seconds
Device 728: 0.4367 seconds
Device 712: 0.7573 seconds
Device 715: 0.7162 seconds
Device 708: 0.8598 seconds
Device 717: 0.7522 seconds
Device 729: 0.4989 seconds
Device 718: 0.7477 seconds
Device 716: 0.8293 seconds
Device 721: 0.8545 seconds
Device 722: 0.8658 seconds
Device 713: 0.9699 seconds
Device 733: 0.5950 seconds
Device 730: 0.6150 seconds
Device 736: 0.5077 seconds
Device 727: 0.7126 seconds
Device 725: 0.9006 seconds
Device 737: 0.6317 seconds
Device 735: 0.6881 seconds
Device 719: 1.0553 seconds
Device 732: 0.7545 seconds
Device 743: 0.5867 seconds
Device 731: 0.7958 seconds
Device 742: 0.5875 seconds
Device 740: 0.6946 seconds
Device 739: 0.7083 seconds
Device 750: 0.4385 seconds
Device 747: 0.4706 seconds
Device 734: 0.8799 seconds
Device 754: 0.3430 seconds
Device 752: 0.5055 secondsDevice 745: 0.6290 seconds
Device 749: 0.5463 seconds
Device 741: 0.8262 seconds

Device 744: 0.7320 seconds
Device 746: 0.6452 seconds
Device 738: 0.9618 seconds
Device 753: 0.5500 seconds
Device 751: 0.6394 seconds
Device 756: 0.4442 seconds
Device 757: 0.4537 seconds
Device 755: 0.5393 seconds
Device 748: 0.7604 seconds
Device 760: 0.3563 seconds
Device 761: 0.3589 seconds
Device 764: 0.3948 seconds
Device 758: 0.4963 seconds
Device 769: 0.2489 seconds
Device 765: 0.3567 seconds
Device 770: 0.2657 secondsDevice 763: 0.5167 seconds

Device 762: 0.5511 seconds
Device 759: 0.6156 seconds
Device 778: 0.2531 seconds
Device 767: 0.4707 seconds
Device 779: 0.3221 seconds
Device 775: 0.3831 seconds
Device 772: 0.4509 seconds
Device 771: 0.4636 seconds
Device 766: 0.5897 seconds
Device 777: 0.4220 seconds
Device 768: 0.6035 seconds
Device 776: 0.4385 seconds
Device 773: 0.4964 seconds
Device 774: 0.5047 seconds
Device 780: 0.5267 seconds
Device 787: 0.2875 seconds
Device 781: 0.5070 seconds
Device 784: 0.4834 seconds
Device 783: 0.5429 seconds
Device 788: 0.4099 seconds
Device 786: 0.4933 seconds
Device 797: 0.3316 seconds
Device 785: 0.5856 seconds
Device 796: 0.4314 secondsDevice 794: 0.4673 seconds
Device 782: 0.7556 seconds
Device 791: 0.4966 seconds
Device 792: 0.5795 seconds
Device 805: 0.4964 seconds
Device 798: 0.5469 seconds
Device 795: 0.6015 seconds
Device 790: 0.6427 seconds
Device 789: 0.7066 seconds
Device 815: 0.4317 seconds
Device 804: 0.5570 seconds
Device 808: 0.5654 seconds
Device 802: 0.6653 seconds
Device 801: 0.6700 seconds
Device 800: 0.7157 seconds
Device 809: 0.6653 seconds
Device 811: 0.6975 seconds
Device 807: 0.7600 seconds
Device 812: 0.7079 seconds
Device 806: 0.7703 seconds
Device 820: 0.6679 secondsDevice 810: 0.7904 seconds
Device 793: 0.9497 seconds
Device 819: 0.6820 seconds
Device 799: 0.8918 seconds

Device 813: 0.8752 seconds
Device 833: 0.5789 seconds
Device 803: 1.0182 seconds
Device 816: 0.8830 seconds
Device 821: 0.7695 seconds
Device 829: 0.6799 seconds
Device 826: 0.7647 seconds
Device 827: 0.7875 seconds
Device 823: 0.8261 seconds
Device 824: 0.8587 seconds
Device 828: 0.7994 seconds
Device 839: 0.5344 seconds
Device 832: 0.8044 seconds
Device 838: 0.5670 seconds
Device 814: 1.0580 seconds
Device 818: 1.0174 seconds
Device 822: 0.9167 seconds
Device 825: 0.9425 seconds
Device 836: 0.6936 secondsDevice 831: 0.9554 seconds

Device 841: 0.4846 seconds
Device 830: 0.9765 seconds
Device 835: 0.7419 seconds
Device 817: 1.1913 seconds
Device 834: 0.8754 seconds
Device 842: 0.5788 seconds
Device 840: 0.7298 seconds
Device 849: 0.4183 seconds
Device 845: 0.5128 seconds
Device 837: 0.8987 seconds
Device 848: 0.4360 seconds
Device 856: 0.3189 seconds
Device 855: 0.3865 seconds
Device 851: 0.5737 seconds
Device 846: 0.6058 seconds
Device 844: 0.7814 seconds
Device 852: 0.6039 seconds
Device 868: 0.3662 seconds
Device 850: 0.6503 seconds
Device 854: 0.5831 seconds
Device 847: 0.7133 seconds
Device 853: 0.6550 seconds
Device 857: 0.6453 seconds
Device 859: 0.5801 seconds
Device 858: 0.6890 seconds
Device 843: 1.0780 seconds
Device 861: 0.6255 seconds
Device 864: 0.6152 seconds
Device 866: 0.6155 seconds
Device 865: 0.6214 seconds
Device 867: 0.6608 seconds
Device 869: 0.6308 seconds
Device 860: 0.7080 seconds
Device 871: 0.5732 seconds
Device 862: 0.7147 seconds
Device 872: 0.6397 seconds
Device 863: 0.7755 seconds
Device 876: 0.6303 seconds
Device 884: 0.4271 seconds
Device 874: 0.6978 seconds
Device 870: 0.8616 seconds
Device 896: 0.2584 seconds
Device 885: 0.4984 seconds
Device 880: 0.6640 seconds
Device 873: 0.7550 seconds
Device 881: 0.7115 seconds
Device 878: 0.8730 seconds
Device 888: 0.6612 seconds
Device 889: 0.6606 seconds
Device 879: 0.8679 seconds
Device 887: 0.6799 seconds
Device 875: 0.9271 seconds
Device 883: 0.8870 seconds
Device 877: 0.9690 seconds
Device 898: 0.5400 seconds
Device 882: 0.9512 seconds
Device 895: 0.6259 seconds
Device 893: 0.6909 seconds
Device 904: 0.5271 seconds
Device 891: 0.7830 secondsDevice 899: 0.6539 seconds
Device 897: 0.7252 seconds
Device 906: 0.6285 seconds

Device 892: 0.8671 seconds
Device 890: 0.8874 seconds
Device 905: 0.6929 seconds
Device 907: 0.6584 seconds
Device 894: 0.8605 seconds
Device 913: 0.5670 seconds
Device 886: 1.1819 seconds
Device 903: 0.8762 seconds
Device 909: 0.8058 seconds
Device 921: 0.6902 seconds
Device 916: 0.7982 secondsDevice 908: 0.8827 seconds

Device 902: 0.9840 seconds
Device 915: 0.8142 seconds
Device 911: 0.8270 seconds
Device 917: 0.8138 seconds
Device 901: 1.0464 seconds
Device 900: 1.0836 seconds
Device 912: 0.8871 seconds
Device 918: 0.8559 seconds
Device 926: 0.7447 seconds
Device 927: 0.7447 seconds
Device 920: 0.9127 seconds
Device 928: 0.7867 seconds
Device 924: 0.8333 seconds
Device 930: 0.8104 seconds
Device 929: 0.8443 seconds
Device 925: 0.8931 seconds
Device 931: 0.8757 seconds
Device 922: 1.0160 seconds
Device 910: 1.1779 seconds
Device 923: 1.0393 seconds
Device 919: 1.1721 seconds
Device 933: 0.8720 seconds
Device 914: 1.2453 secondsDevice 946: 0.4953 seconds
Device 935: 0.8245 seconds
Device 934: 0.9255 seconds
Device 940: 0.6753 seconds
Device 936: 0.8761 seconds
Device 941: 0.7091 seconds
Device 943: 0.6493 seconds
Device 937: 0.8967 seconds
Device 938: 0.8405 seconds
Device 932: 1.0626 seconds
Device 949: 0.4652 seconds
Device 947: 0.6404 seconds
Device 939: 0.9055 seconds
Device 952: 0.4853 seconds
Device 960: 0.4661 seconds
Device 945: 0.8109 seconds
Device 948: 0.6626 secondsDevice 942: 0.9244 seconds

Device 958: 0.5625 seconds
Device 950: 0.6794 seconds
Device 954: 0.6580 seconds
Device 957: 0.6125 seconds
Device 955: 0.6524 seconds
Device 953: 0.6631 seconds
Device 956: 0.6247 seconds
Device 951: 0.6849 seconds
Device 944: 1.0030 seconds
Device 962: 0.4845 seconds
Device 961: 0.5824 seconds
Device 959: 0.6879 seconds
Device 963: 0.4688 seconds
Device 967: 0.2173 seconds
Device 964: 0.5054 seconds
Device 970: 0.2965 seconds
Device 966: 0.4092 seconds
Device 977: 0.2934 seconds
Device 965: 0.6364 seconds
Device 980: 0.3099 seconds
Device 983: 0.2792 seconds
Device 975: 0.3537 seconds
Device 972: 0.3939 seconds
Device 971: 0.4564 seconds
Device 968: 0.5577 seconds
Device 976: 0.4122 seconds
Device 973: 0.4320 seconds
Device 969: 0.5652 seconds
Device 974: 0.5329 seconds
Device 992: 0.4437 seconds
Device 995: 0.4255 seconds
Device 978: 0.5189 seconds
Device 985: 0.5083 seconds
Device 981: 0.5553 seconds
Device 984: 0.5386 seconds
Device 987: 0.5236 seconds
Device 990: 0.5208 seconds
Device 989: 0.5878 seconds
Device 996: 0.5752 seconds
Device 994: 0.5925 seconds
Device 991: 0.6045 seconds
Device 982: 0.6577 seconds
Device 986: 0.6607 seconds
Device 979: 0.7149 seconds
Device 988: 0.6578 seconds
Device 999: 0.5205 seconds
Device 998: 0.5751 seconds
Device 993: 0.7057 seconds
Device 997: 0.6894 seconds
```