TP4 - Ex.3

May 27, 2024

Estruturas Criptográficas

PG53886, Ivo Miguel Alves Ribeiro

A95323, Henrique Ribeiro Fernandes

3. Construir tabelas de comparações das suas implementações, para os vários níveis de segurança NIST e em termos dos seguintes parâmetros 1. Tempos: geração das chaves, produção da assinatura e verificação da assinatura. 2. Tamanhos: da chave pública, da chave privada e da assinatura.

In [ ]:
```python
import time
import pandas as pd
from pmain.dilithium import Dilithium2, Dilithium3, Dilithium5
from sphincs import sphincs

# Função para medir o tempo de execução
def measure_time(func, *args):
    start_time = time.time()
    result = func(*args)
    end_time = time.time()
    return result, end_time - start_time

# Testando diferentes níveis de segurança
security_levels = ['ML-DSA-44', 'ML-DSA-65', 'ML-DSA-87']
results = []

for level in security_levels:
    # Configure as variáveis específicas para cada nível de segurança
    if level == 'ML-DSA-44':
        dilithium = Dilithium2
    elif level ==  'ML-DSA-65':
        dilithium = Dilithium3
    elif level == 'ML-DSA-87':
        dilithium = Dilithium5

    # Medir tempos e tamanhos
    (pk, sk), keygen_time = measure_time(dilithium.keygen)
    msg = b"Your message signed by Dilithium"
    sig, sign_time = measure_time(dilithium.sign, sk, msg)
    verify_time = measure_time(dilithium.verify, pk, msg, sig)[1]

    public_key_size = len(pk)
    private_key_size = len(sk)
    signature_size = len(sig)

    results.append({
        'Name': 'Dilithium'+level,
        'Keygen Time (s)': keygen_time,
        'Sign Time (s)': sign_time,
        'Verify Time (s)': verify_time,
        'Public Key Size (bytes)': public_key_size,
        'Private Key Size (bytes)': private_key_size,
        'Signature Size (bytes)': signature_size
    })

# Medir tempos e tamanhos
(sk, pk), keygen_time = measure_time(sphincs.spx_keygen)
msg = b"Your message signed by Dilithium"
sig, sign_time = measure_time(sphincs.spx_sign, msg, sk)
verify_time = measure_time(sphincs.spx_verify, msg, sig, pk)[1]

public_key_size = len(pk)
private_key_size = len(sk)
signature_size = len(sig)

results.append({
```

```
    'Name':'Sphincs SLH-DSA-SHAKE-256s',
    'Keygen Time (s)': keygen_time,
    'Sign Time (s)': sign_time,
    'Verify Time (s)': verify_time,
    'Public Key Size (bytes)': public_key_size,
    'Private Key Size (bytes)': private_key_size,
    'Signature Size (bytes)': signature_size
})

# Criar um DataFrame com os resultados
df = pd.DataFrame(results)

# Exibir a tabela de resultados
print(df)
```

```
                       Name  Keygen Time (s)  Sign Time (s)  \
0          DilithiumML-DSA-44         0.023293       0.056795
1          DilithiumML-DSA-65         0.038092       0.201715
2          DilithiumML-DSA-87         0.039207       0.119706
3  Sphincs SLH-DSA-SHAKE-256s         1.497489      23.414620

   Verify Time (s)  Public Key Size (bytes)  Private Key Size (bytes)  \
0         0.027040                     1312                      2528
1         0.036722                     1952                      4000
2         0.062793                     2592                      4864
3         0.021906                       64                       128

   Signature Size (bytes)
0                    2420
1                    3293
2                    4595
3                   29792
```