Jpeg file that I used : 2.65 MB

My source code :

```
import subprocess
import time
import os
import matplotlib.pyplot as plt


def generate_key_pair(key_type):
    if key_type == 'rsa':
        subprocess.run(['openssl', 'genpkey', '-algorithm', 'RSA', '-out',
'rsa_private.pem', '-pkeyopt' , 'rsa_keygen_bits:2048'], check=True)
        subprocess.run(['openssl', 'rsa', '-pubout', '-in',
'rsa_private.pem', '-out', 'rsa_public.pem'], check=True)
    elif key_type == 'dsa':
        # Generate DSA private key

        subprocess.run(['openssl', 'dsaparam', '-out', 'dsaparam.pem',
'2048'], check=True)
        subprocess.run(['openssl', 'gendsa', '-out', 'dsa_private.pem',
'dsaparam.pem'], check=True)

        # Generate DSA public key
        subprocess.run(['openssl', 'dsa', '-in', 'dsa_private.pem', '-
outform' ,'DER', '-pubout', '-out', 'dsa_public.pem'], check=True)


def sign_and_verify(key_type, file_to_sign):
    signing_times = []
    verifying_times = []

    for _ in range(500):
        # Signing
        start_time = time.time()
        subprocess.run(['openssl', 'dgst', '-sha256', '-sign',
f'{key_type}_private.pem', '-out', f'signature_{key_type}.bin',
file_to_sign], check=True)
        signing_time = time.time() - start_time
        signing_times.append(signing_time)

        # Verifying
        start_time = time.time()
        subprocess.run(['openssl', 'dgst', '-sha256', '-verify',
f'{key_type}_public.pem', '-signature', f'signature_{key_type}.bin',
file_to_sign], check=True)
        verifying_time = time.time() - start_time
        verifying_times.append(verifying_time)

    average_signing_time = sum(signing_times) / len(signing_times)
    average_verifying_time = sum(verifying_times) / len(verifying_times)
    return average_signing_time, average_verifying_time

def main():
    # Specify the path to your image file
    file_to_sign = "homework3.jpeg"

    # Disable hardware acceleration
    os.environ["OPENSSL_ENGINES"] = "disable"

    # Step 1: Generate RSA and DSA key pairs
    generate_key_pair('rsa')
    generate_key_pair('dsa')

    # Step 2: Evaluate efficiency
    rsa_signing_time, rsa_verifying_time = sign_and_verify('rsa',
file_to_sign)
    dsa_signing_time, dsa_verifying_time = sign_and_verify('dsa',
file_to_sign)

    # Step 3: Print results
    print(f'RSA Signing Time (Average): {rsa_signing_time:.6f} seconds')
    print(f'RSA Verifying Time (Average): {rsa_verifying_time:.6f} seconds')
    print(f'DSA Signing Time (Average): {dsa_signing_time:.6f} seconds')
    print(f'DSA Verifying Time (Average): {dsa_verifying_time:.6f} seconds')

    # Step 4: Plot results
    labels = ['RSA Signing', 'RSA Verifying', 'DSA Signing', 'DSA Verifying']
    times = [rsa_signing_time, rsa_verifying_time, dsa_signing_time,
dsa_verifying_time]

    plt.bar(labels, times, color=['blue', 'blue', 'green', 'green'])
    plt.ylabel('Average Time (seconds)')
    plt.title('RSA vs DSA Efficiency Comparison')
    plt.show()

    # Optional: Clean up generated files
    os.remove('rsa_private.pem')
    os.remove('rsa_public.pem')
    os.remove('dsa_private.pem')
    os.remove('dsa_public.pem')
    os.remove('dsaparam.pem')
    os.remove('signature_rsa.bin')
    os.remove('signature_dsa.bin')

if __name__ == "__main__":
    main()
```

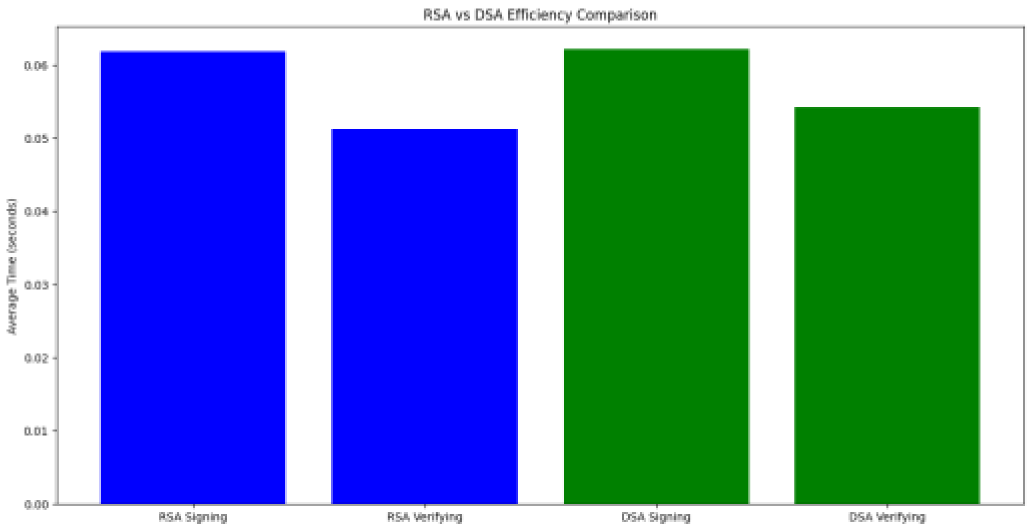
my result (500 iteration) :

RSA Signing Time (Average): 0.061786 seconds

RSA Verifying Time (Average): 0.051271 seconds

DSA Signing Time (Average): 0.062136 seconds

DSA Verifying Time (Average): 0.054261 seconds



Conclusion :

It appears that RSA generally has slightly faster average signing and verifying times compared to DSA in this specific experiment. The differences are relatively small, but RSA is showing slightly better performance in both signing and verifying.

But actual performance can depend on various factors, including the specific implementation of the cryptographic algorithms, the key sizes used, and the hardware/software environment. The results might vary in different scenarios, and it's essential to consider factors like security requirements and key management practices when choosing between RSA and DSA. Additionally, it's worth noting that RSA is more widely used and supported in various applications, which might influence the choice of algorithm based on compatibility and interoperability.