4. Authenticating the signature using MD5 hash algorithm

In [2]:
```python
import hashlib
from cryptography.hazmat.primitives.asymmetric import rsa, padding
from cryptography.hazmat.primitives import serialization, hashes
# 1️⃣ Generate RSA key pair
private_key = rsa.generate_private_key(
    public_exponent=65537,
    key_size=2048,
)

public_key = private_key.public_key()

# ----------------------------
# 2️⃣ Sender side: Create message & signature
# ----------------------------
message = b"Pay Rs. 5000 to John"

# Step 1: Create MD5 hash of the message
md5_hash = hashlib.md5(message).digest()

# Step 2: Encrypt the hash using the private key to create a signature
signature = private_key.sign(
    md5_hash,
    padding.PKCS1v15(),
    hashes.MD5()  # specify MD5 hash algorithm
)

print("Original Message:", message.decode())
print("Signature (hex):", signature.hex())

# ----------------------------
# 3️⃣ Receiver side: Verify signature
# ----------------------------
try:
    # Compute MD5 hash of the received message
    md5_hash_receiver = hashlib.md5(message).digest()

    # Verify the signature using the sender's public key
    public_key.verify(
        signature,
        md5_hash_receiver,
        padding.PKCS1v15(),
        hashes.MD5()
    )

    print("\n✅ Signature is valid — message authenticated!")
except Exception as e:
    print("\n❌ Signature verification failed!", str(e))
```

```
Original Message: Pay Rs. 5000 to John
Signature (hex): aca70f23c6823cb2112a100931824ab3aad8af29feb0c115075e625470d00b84898
0d2ef8159e413dc28922db29c1f107244e0158d2d85b07200c8456e7acb9ea9fbc3fc17a48ec8129eb76
f94f5fcaf986b82e24edc3dfbf2d3bcc39429a3c94026c6e529ca5d797adb8b920c7facf056ea0528c5d
804df2d81148599bfa939879490c608a99e50d4bf3cc5f0edc6b4fc2362809ef7f671214effea8dbfdf4
789a11d7daefae37707440a874888ad45480859e201d184962d80ee2114bbc8e1f2c9f2df11941037031
b3ab5c1001464b690128154d2d8fb680f14e157f9304c1ffaaf80b9aa3f5b1ccc950d6ff66a5450fdd1a
d38a0c178b6ed8ad3642462d1

✅ Signature is valid — message authenticated!
```

In [ ]: