**Digital Signature**

from cryptography.hazmat.primitives import hashes

from cryptography.hazmat.primitives.asymmetric import ec

from cryptography.hazmat.primitives.serialization import Encoding, PublicFormat

private\_key = ec.generate\_private\_key(ec.SECP256R1())

public\_key = private\_key.public\_key()

message = b"Jspm csbs"

signature = private\_key.sign(message, ec.ECDSA(hashes.SHA256()))

try:

public\_key.verify(signature, message, ec.ECDSA(hashes.SHA256()))

print("Signature is valid!")

except Exception:

print("Signature is invalid.")

serialized\_public\_key = public\_key.public\_bytes(Encoding.PEM, PublicFormat.SubjectPublicKeyInfo)

deserialized\_public\_key = ec.load\_pem\_public\_key(serialized\_public\_key)

print("Serialized Public Key:")

print(serialized\_public\_key.decode("utf-8"))

**Output**

Signature is valid!