Ex.No:8 **DSS ALGORITHM**

Date:

AIM:

To implement signature scheme using DSS algorithm.

ALGORITHM:

1. Import the required packages SHA,ECC,DSS.
2. Generate the key using ECC generate function.
3. Open and write the public and private key files and save it in the folder where the program is stored.
4. Generate the signature using DSS.
5. Verify the signature using public key at the receiver side and display the output.

PROGRAM:

```
from Crypto.PublicKey import ECC
from Crypto.Signature import DSS
from Crypto.Hash import SHA256


def init_keys(curve, format):
    key = ECC.generate(curve = curve)

    with open('prkey.pem', 'w') as prfile, open('pukey.pem', 'w') as pufile:
        prfile.write(key.export_key(format = format))
        pufile.write(key.public_key().export_key(format = format))


def get_signature(message, key, mode):
    digest = SHA256.new(message)
    signer = DSS.new(key, mode)
    signature = signer.sign(digest)
    return signature


def verify(message, key, mode, signature):
    digest = SHA256.new(message)
    verifier = DSS.new(key, mode)

    try:
        verifier.verify(digest, signature)
```

```
    except ValueError:
        return False
    else:
        return True


def main():
    CURVE = 'P-256'
    FORMAT = 'PEM'
    MESSAGE_SENDER = b'Bravo-6, going dark.'
    MESSAGE_RECEIVER = b'Bravo-6, going dark.'
    MODE = 'fips-186-3'

    init_keys(CURVE, FORMAT)
    with open('prkey.pem') as prfile, open('pukey.pem') as pufile:
        prkey = ECC.import_key(prfile.read())
        pukey = ECC.import_key(pufile.read())

    signature = get_signature(MESSAGE_SENDER, prkey, MODE)

    status = verify(MESSAGE_RECEIVER, pukey, MODE, signature)

    print('Authentic') if status else print('Not Authentic')


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

OUTPUT:

```
PS C:\Users\student\Desktop\cns> & "D:/Program Files/Python37/python.exe" c:/Users/student/Desktop/cns/dss.py
Authentic
PS C:\Users\student\Desktop\cns>
```

RESULT:

Thus the program to implement signature scheme using DSS algorithm has been verified.