EXPERIMENT 5
SECURE HASHING ALGORITHM

3.1 Aim

To write a program for implementing the SHA-1 algorithm.

3.2 Algorithm

1. START
2. Create a Node.js project to work .
3. Create the sha.js file.
4. Import the internal crypto module of Node.js .
5. Use the desired algorithm for hashing (sha1) and assign to a variable named *algorithm.*
6. Declare a constant variable *message* as input message.
7. Create a Hash object that can be used to generate hash digests with the given algorithm using the crypto module.
8. Pass the message to hash in the *update* function of the *hashObject .*
9. Update function calculates the digest of all of the data passed to be hashed.
10. Use function digest on the hashedData with the required format (hex) as parameter.
11. *hashObject* use is disabled further by *digest* function .
12. Log the generated hashed message in console

3.3 Program

```
//Loading the crypto module in node.js
let crypto = require('crypto');
//define message to hash
const message = "this is a sample";
//log the plain text in console
console.log("Plain text : ", message);
//define the hashing algorithm
const algorithm = "sha1";
//creating hash object
let hashObject = crypto.createHash(algorithm);
//passing the data to be hashed
let data = hashObject.update(message, 'utf-8');
```

```
//Creating the hash in the required format
const generatedHash = data.digest('hex');
//log the output on the console
console.log("Hashed message : " + generatedHash);
```

3.4 Output

```
PS C:\Users\cinoy\OneDrive\Documents\sha> node sha1.js
Plain text :  this is a sample
Hashed message : 5435c304f07425026b89ae4e0ce71590907c4671
PS C:\Users\cinoy\OneDrive\Documents\sha>
```

3.5 Result

The SHA-1 algorithm was implemented successfully.