

CS 305 Module Five Coding Assignment Checksum Verification Template

Instructions

Using the instructions from the Module Five Coding Assignment Checksum Verification Guidelines and Rubric, replace the bracketed text with the relevant information in your own words.

1. Algorithm Cipher

For this assignment, I have chosen the SHA-256 cryptographic hash function. SHA-256 is a member of the SHA-2 family and provides a 256-bit digest, making it highly secure and resistant to collisions. Unlike older hashing algorithms such as MD5 and SHA-1, SHA-256 ensures a high level of integrity and security, which is crucial for checksum verification.

2. Justification

SHA-256 is widely used in cryptographic applications, digital signatures, SSL certificates, and blockchain technology due to its collision resistance and security. A cryptographic hash function must be one-way, deterministic, and resistant to preimage attacks. The SHA-256 algorithm meets these criteria by producing a unique, fixed-length hash for each unique input. It is an industry-standard algorithm recommended by NIST (National Institute of Standards and Technology) and is used in many modern security protocols.

3. Generate Checksum

I implemented the checksum verification in Java using Spring Boot by modifying the Java code to include my first and last name in the checksum generation. I utilized the MessageDigest class from the java.security package to generate an SHA-256 hash and converted the resulting hash from bytes to a hexadecimal string for readability. A RESTful API endpoint (/hash) was created to display the input data string, the algorithm used, and the generated checksum value. The Spring Boot application was then configured to listen on port 8443 (HTTPS), and I verified the checksum by accessing the RESTful API via a web browser.

```
package com.snhu.sslserver;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.nio.charset.StandardCharsets;

@SpringBootApplication
public class ServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(ServerApplication.class, args);
    }
}

@RestController
class ServerController {

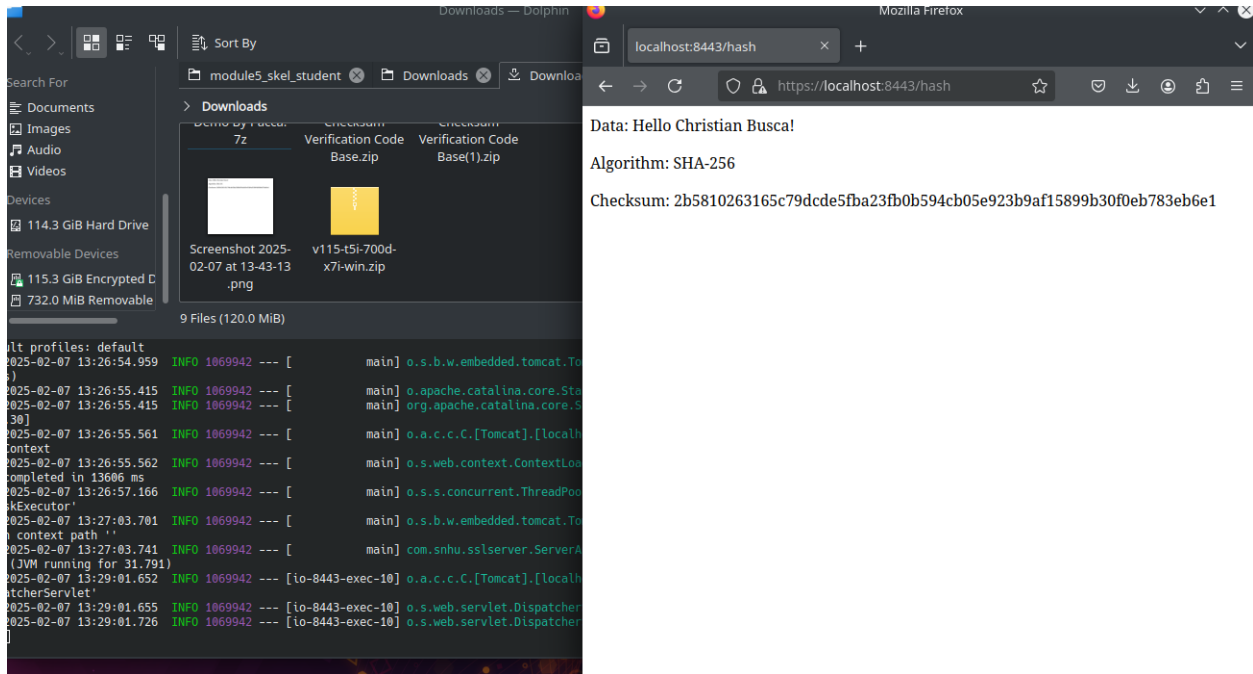
    @RequestMapping("/hash")
    public String myHash() {
        String data = "Hello Christian Busca!"; // Your name
        String checksum = generateChecksum(data);

        return "<p>Data: " + data + "</p>"
            + "<p>Algorithm: SHA-256</p>"
            + "<p>Checksum: " + checksum + "</p>";
    }

    private String generateChecksum(String data) {
        try {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hashBytes = digest.digest(data.getBytes(StandardCharsets.UTF_8));
            return bytesToHex(hashBytes);
        } catch (NoSuchAlgorithmException e) {
            throw new RuntimeException(e);
        }
    }

    private String bytesToHex(byte[] bytes) {
        StringBuilder hexString = new StringBuilder();
        for (byte b : bytes) {
            hexString.append(String.format("%02x", b));
        }
        return hexString.toString();
    }
}
```

4. Verification



The screenshot displays a verification process across three windows:

- File Explorer (Dolphin):** Shows the 'Downloads' folder containing files like 'Screenshot 2025-02-07 at 13:43-13.png' and 'v115-t5l-700d-x7l-win.zip'. It also shows a list of files with verification codes and checksums.
- Terminal Window:** Displays a series of log messages indicating the successful execution of a verification process. The messages include timestamps, log levels (INFO), and the names of the classes and methods being executed, such as 'o.s.b.w.embedded.tomcat.Yo', 'o.apache.catalina.core.Sta', 'org.apache.catalina.core.3', 'o.a.c.c.C.[Tomcat].[localh', 'o.s.web.context.ContextLoa', 'o.s.s.concurrent.ThreadPoo', 'o.s.b.w.embedded.tomcat.To', 'com.snhu.ssiserver.ServerA', and 'o.s.web.servlet.Dispatcher'.
- Web Browser (Mozilla Firefox):** Shows the URL 'https://localhost:8443/hash' and the following verification results:
 - Data: Hello Christian Busca!
 - Algorithm: SHA-256
 - Checksum: 2b5810263165c79dcde5fba23fb0b594cb05e923b9af15899b30f0eb783eb6e1