**2) Queston2:**

You are working as a security analyst for a financial institution that handles sensitive customer information. Recently, the organisation has adopted SHA256 encryption to secure its data. Your task is to encrypt a piece of secret information using SHA256 and evaluate its resistance to decryption attempts.

* 1:Encrypt the secret information using SHA256 in both HashCalc and Cryptoforge.
* 2:Attempt to decode the encrypted information using various tools (e.g., Online Hash Crackers).
  1. Encrypting (Hashing) Using SHA256 in Tools
     1. Using HashCalc

Steps:

1. Open **HashCalc**
2. Input: Your secret message (e.g., MySecret123)
3. Select **SHA256** from the list of algorithms
4. Click **Calculate**
5. You will get output like: SHA256: a2c5e4fc3d8dcd8e8c0e1b998b29b4931ffbc8ff24bbd7b7c43f7335090e8a77
   * 1. Using Cryptoforge

Cryptoforge is more of a file encryption tool, but assuming you mean to generate a hash for verification:

* + - 1. Create a text file with your secret info (e.g., secret.txt)
      2. Use Cryptoforge to generate the hash (depending on its options)
      3. Select SHA256
      4. It will produce the same kind of irreversible hash output
  1. Attempting to Decode the SHA256 Hash

Since SHA256 is irreversible, **you can't decode it**, but you can **attempt to crack it** using:

1. Tools for Cracking SHA256 Hashes

|  |  |
| --- | --- |
| **Tool** | **Description** |
| CrackStation | |  | | --- | | Uses a large precomputed hash database |  |  | | --- | |  | |
| |  | | --- | | HashKiller.co.uk |  |  | | --- | |  | | |  | | --- | | Community-powered cracking tool |  |  | | --- | |  | |
| |  | | --- | | OnlineHashCrack |  |  | | --- | |  | | |  | | --- | | Advanced online cracker (free & paid) |  |  | | --- | |  | |
| |  |  |  | | --- | --- | --- | | |  | | --- | | John the Ripper |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Offline cracking via dictionary/brute-force |  |  | | --- | |  | |
| Hashcat | GPU-based offline cracker (super fast) |

Conditions for Cracking Success

Cracking **only works** if:

* The original input was weak (e.g., 123456)
* It's in the cracker’s dictionary/rainbow table
* The hash isn’t salted (SHA256 is deterministic)

Example: ef92b778bafe771e89245b89ecbc08a44a4e166c06659911881f383d4473e94f

Try this hash in **CrackStation.net**, and it will return: password123

But if you use a complex password like \*N@3k!wQd8#, cracking becomes **nearly impossible**.

**Conclusion: Resistance to Decryption**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Resistance Level | | --- |  |  | | --- | |  | |
| |  | | --- | | Brute Force |  |  | | --- | |  | | |  | | --- | | Medium (slow with strong passwords) |  |  | | --- | |  | |
| |  | | --- | | Dictionary Attack |  |  | | --- | |  | | |  | | --- | | Low if using weak/common passwords |  |  | | --- | |  | |
| |  | | --- | | Rainbow Tables |  |  | | --- | |  | | |  | | --- | | High if salted, Medium if not |  |  | | --- | |  | |
| |  | | --- | | Collision Resistance |  |  | | --- | |  | | |  | | --- | | High (unique outputs for different inputs) |  |  | | --- | |  | |
| |  | | --- | | Preimage Resistance |  |  | | --- | |  | | Very High (can’t reverse back from hash) |

**Best Practices to Increase Resistance**

* Use **salting**: Add a unique random value to each input
* Enforce **strong passwords**
* Consider **key-stretching algorithms** like bcrypt or PBKDF2 for password storage

Summary for Report

The secret data was hashed using SHA256 via HashCalc and Cryptoforge. The resulting hash was tested against several online hash-cracking tools. Due to the complexity of the input and lack of match in existing databases, the hash remained uncracked, demonstrating SHA256’s high resistance to decryption under current methods, especially when combined with salting and complex inputs.