

Hashing

Overview

Cryptographic hash functions transform data of any size into a fixed-length message digest or checksum, with SHA256 producing a 256-bit (64 hexadecimal digit) output regardless of input size. These functions are deterministic and demonstrate the avalanche effect—even a single-bit change in input produces a completely different hash value. Hash functions serve critical security purposes including secure password storage and detecting file modifications or tampering. Older algorithms like MD5 and SHA-1 are now cryptographically broken and vulnerable to collision attacks.

Key Information

- Fixed Output Size: Hash functions always produce the same length output regardless of input file size (e.g., SHA256 always produces 256 bits or 64 hex digits)
 - Avalanche Effect: A single-bit change in input data produces a drastically different hash value, enabling reliable tamper detection
 - Primary Applications: Password storage (with salting), file integrity verification, and detecting both intentional tampering and transfer errors
 - Secure vs. Broken Algorithms: SHA224, SHA256, SHA384, SHA512, and RIPEMD160 are currently secure; MD5 and SHA-1 are cryptographically broken and susceptible to hash collisions
 - HMAC Authentication: HMAC combines a hash function with a secret key using inner/outer padding (ipad/opad) to provide message authentication.
-

Notes

```
sha256sum file
```

```
hmac256 key file
```

Task

1. What is the SHA256 checksum of the file `order.json`?
 1. `sha256sum order.json`
 2. `2c34b68669427d15f76a1c06ab941e3e6038dacdfb9209455c87519a3ef2c660`
 2. Open the file `order.json` and change the amount from `1000` to `9000`. What is the new SHA256 checksum?
 1. `sha256sum order.json`
 2. `11faeec5edc2a2bad82ab116bbe4df0f4bc6edd96adac7150bb4e6364a238466`
 3. Using SHA256 and the key `3RfDFz82`, what is the HMAC of `order.txt`?
 1. `hmac256 3RfDFz82 order.json`
 2. `c7e4de386a09ef970300243a70a444ee2a4ca62413aeaeb7097d43d2c5fac89f`
-

Conclusion

Cryptographic hash functions are fundamental security tools that provide both data integrity verification and secure password storage mechanisms. Understanding the difference between secure algorithms (SHA-256 family) and broken ones (MD5, SHA-1) is essential for implementing modern cybersecurity solutions. HMAC extends basic hashing by incorporating secret keys, making it suitable for message authentication in scenarios requiring both integrity and authenticity verification.

Resources

- **TryHackMe:** [Intro to Cryptography](#)
-
-

