NAME: Atharva Tamhankar
UID: 2022300127
EXP NO.: 5

AIM: To implement cryptographic hash algorithms (MD5, SHA-1, SHA-256, SHA-512) in Python to generate file checksums, verify file integrity, and detect modifications.

THEORY

Cryptographic Hash Functions

A cryptographic hash function is a mathematical process that converts input data into a unique fixed-length string known as a hash value or digest. It serves as a digital fingerprint for data. The process is one-way and irreversible, ensuring that the original data cannot be retrieved from the hash value.

Essential Characteristics

- **Determinism:** The same input always results in the same hash output.
- **Collision Resistance:** It is practically impossible for two different inputs to yield the same hash.
- **Avalanche Effect:** A tiny change in input leads to a drastically different hash output.
- **Non-Reversibility:** You cannot reconstruct original data from the hash.

File Integrity and Verification

Hashing is widely used to verify file integrity. The concept involves comparing pre-stored hash values (checksums) with newly generated ones. If both hashes match, the file is confirmed to be original and untampered. Otherwise, it has been altered or corrupted. Checksum files, such as *filename.sha256*, are commonly used for this verification.

Common Algorithms

- **MD5 & SHA-1:** These algorithms are outdated due to collision vulnerabilities and should not be used in modern security systems.
- **SHA-256 & SHA-512:** Part of the SHA-2 family, these algorithms provide strong security and are standard for digital signatures and software verification.

CONCLUSION:

Understood how hashing ensures file integrity and how checksum comparison helps detect data tampering using secure hash functions.