Secure Erase protocols and methods:

- Security Erase and Enhanced Security Erase, both approved by NIST
- Secure erase overwrites all **user** data areas with binary zeroes
- Enhanced secure erase writes **predetermined** data patterns (set by the manufacturer) to all user data areas, including sectors that are no longer in use due to reallocation.

For hard disk drive (and without encryption):

Secure Erase:
The logical view of the disk as a huge sequence of numbered sectors; the "secure erase" will overwrite all these sectors (and only these sectors) once, with zeros

Enhanced Secure Erase:
It overwrites data several times with distinct bit patterns, to be sure that the data is thoroughly destroyed. It also overwrites sectors which are no longer used because they triggered an I/O error at some point and were remapped by the internal firmware of the drives

For SSD:

- Secure erase uses another method, which is more efficient, is encryption:
- When it is first powered, the drive generates a random symmetric key K and keeps it in some reboot-resistant storage space (say, some EEPROM).
- Every data read or write will be encrypted symmetrically, using K as key.
- To implement a "secure erase", the disk just needs to forget K by generating a new one and overwriting the previous one.

SSD implements "secure erase", it uses the encryption mechanism, because the "overwrite with zeros" does not make any sense, given the behavior of the flash cells and the heavy remapping/error correcting code layers used in SSD.

The implementation of the Secure Erase/ Enhanced Secure Erase: The OS (via the application) send some specific commands to the drive f/w and the internal drive's firmware does the erase. When it completed, it will report back to the OS (the application).