Property	Puzzle-Friendly Hash Functions	Regular Hash Functions
Purpose	Designed for proof-of-work puzzles, where finding a specific hash requires computational effort	Primarily used for fast data retrieval, integrity verification, and data indexing
Collision Resistance	Emphasizes collision resistance, making it computationally difficult to find two inputs with the same hash value	Also aims for collision resistance, but may not be as computationally intensive as puzzle-friendly hash functions
Computational Intensity	Requires significant computational effort to compute the hash, making it time-consuming and resource-intensive	Designed for efficiency, aiming for quick computation to support real-time applications
Preimage Resistance	Emphasizes resistance against finding the original input given the hash value	Also aims for preimage resistance, but may not require the same level of computational intensity as puzzle-friendly hash functions
Verification Ease	Verification is straightforward and can be done quickly once the solution is found	Verification is typically fast and straightforward, ensuring quick integrity checks
Application	Mainly used in cryptocurrency mining (e.g., Bitcoin's proof-of-work) and other proof-of-work systems	Widely used in various applications, including data integrity verification, password hashing, and data indexing
Example Algorithms	Examples include Scrypt, Argon2, and Cryptonight	Examples include MD5, SHA-256, SHA-3, and Blake2