## Required properties:

* Minimize Hash Collisions: In the event of a hash collision, multiple passwords would generate the same hash code, rendering them interchangeable[[1]](#footnote-1).
* Pre-Image Resistant (One-Way): It should not be possible to discern the password from the hash code.
* Slow: It is possible to determine an accepted password through brute-force[[2]](#footnote-2). Furthermore, the attacker only needs a valid password, irrespective of its user. Hence, hash functions must be slow, to slow down such attacks[[3]](#footnote-3).

## sdbm Hash Function (derived from C code)[[4]](#footnote-4) :

BEGIN sdbm

hash = 0

FOR char IN str

hash = char + (LEFTSHIFT hash by 6) + (LEFTSHIFT hash by 16) - hash

ENDFOR

RETURN hash

END sdbm

hash is an unsigned long as otherwise, the bit shifts may result in a negative value that would have to be accounted for, which would complicate the logic significantly.

Bit shifting and subtraction is performed on the hash code in order to vary the bits with each character. This ensures the order of the characters affects the output code.

1. "Network Security" Princeton University, accessed October 2, 2015, http://www.cs.princeton.edu/courses/archive/spr11/cos461/docs/rec08-net-security.pdf [↑](#footnote-ref-1)
2. Halderman, J., Waters, B., and Felten, E. *A Convenient Method for Securely Managing Passwords* (Chiba: International World Wide Web Conference Committee, 2005) [↑](#footnote-ref-2)
3. Halderman, J., Waters, B., and Felten, E. *A Convenient Method for Securely Managing Passwords* (Chiba: International World Wide Web Conference Committee, 2005) [↑](#footnote-ref-3)
4. “Hash Functions” York University, accessed October 2, 2015, http://www.cse.yorku.ca/~oz/hash.html [↑](#footnote-ref-4)