

# Re: incremental MD5 ?

---

*Source:* <http://www.derkeiler.com/Newsgroups/sci.crypt/2006-03/msg00862.html>

---

- *From:* "Joseph Ashwood" <[ashwood@xxxxxxx](mailto:ashwood@xxxxxxx)>
  - *Date:* Sat, 18 Mar 2006 07:39:21 GMT
- 

"slim" <[chinghai@xxxxxxxxxx](mailto:chinghai@xxxxxxxxxx)> wrote in message  
<news:1142661857.379813.193180@xx>

hi everyone!!  
i'm writing some program that backup memory data.

and i'm trying MD5 encryption, the original data is quite big and MD5 encrypted data is added to it..

Problem #1, MD5 is not an encryption algorithm, and you're actually not using it in an encryption mode (as far as I can tell). MD5 is a hash function, it'll help you if you think of it as such.

Problem #2, MD5 is considered weak, since you are attempting to use it for it's direct crypto-hash properties you need to change algorithms. I'd suggest that for a 32-bit system go with SHA-256, 64-bit system SHA-512, and either way be prepared to move to Whirlpool is the need arises.

it is like this

-----

original data

-----

MD5 encrypted data (of original data)

-----

and after reboot, it read memory data and encrypt it, and compare it with previous encrypted one.  
if it is different (maybe spoiled, damaged), then find different part of the current data, and recalculate the part with MD5, and update MD5 encrypted data on the exact part of it...i mean, update only exact part of encrypted area...

Re: incremental MD5 ?

do you think MD5 can do this method?  
do you think the whole data's encrypted data is same as previous  
partially updated one?  
incremental MD5 is implemented ?

What you're looking for is called a Merkle tree, also called the library signing algorithm. Conceptually it works like this:

Given a very large library (many floors, hundreds of thousands of books, etc)

Hash the book to get the book hash

Hash all the book hashes on a shelf to get the shelf hash

Hash all the shelf hashes in a column to get the column hash

Hash all the column hashes in an aisle to get the aisle hash

Hash all the aisle hashes in a section to get the section hash

Hash all the section hashes on a floor to get the floor hash

Hash all the floor hashes in a library to get the library hash

Sign the library hash

You can reduce the number of steps to meet your needs. The benefit for you is that when you change the value of a "book" (probably memory page) you need to perform only a few hashes of relatively small values. The downside is that you'll have to store the entire tree, but given a state and a library hash (system memory hash) you can verify whether or not the state and hash match, although without the tree you won't be able to tell which part(s) changed.

Joe

.