Analisis Penggunaan Fungsi Hash Pada *Cryptocurrency*

Asyifa Nurul Shafira - 13521125[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13521125@std.stei.itb.ac.id

***Abstract*—Digitalisasi dalam berbagai kegiatan telah menjadi hal yang sangat wajar. Salah satunya dalam bidang keuangan. Beberapa tahun belakangan, mata uang virtual atau yang dikenal dengan *cryptocurrencies* menjadi salah satu instrumen keuangan yang banyak mendapat perhatian masyarakat global. Hal ini karena kemudahan dan keuntungan yang ditawarkannya. *Cryptocurrency* banyak digunakan dalam transaksi online maupun diperjualbelikan sebagai investasi. *Cryptocurrency* sendiri menggunakan database terdistribusi, yang dikenal sebagai 'blockchain' dalam operasinya. Ada beberapa mekanisme atau teknik yang dipakai di dalam blockchain untuk menjamin keamanannya. Salah satunya adalah pemanfaatan teknik hash dari kriptografi yang membuat blok akan memiliki nilai hash yang mengidentifikasi blok dan seluruh isinya yang bersifat unik. Penggunaan konsep ini membuat *cryptocurrency* sulit untuk dipalsukan. Meskipun demikian, tidak berarti sistem tersebut benar-benar aman dari serangan hacker. Ancaman pembajakan oleh pihak yang tidak bertanggung jawab menjadi risiko bagi seluruh kegiatan yang dilakukan secara online sejak lama.**

***Keywords*— Hash, Kriptografi, *Cryptocurrency*, SHA-256, SHA-512.**

I. PENDAHULUAN

Sumber: Is cryptocurrency the future of finance? | World Economic Forum (weforum.org)

Pesatnya perkembangan teknologi telah membawa perubahan dalam berbagai aspek kehidupan manusia. Digitalisasi dalam berbagai kegiatan telah menjadi hal yang sangat wajar. Salah satu contoh penerapannya adalah dalam bidang keuangan. Beberapa tahun belakangan, mata uang virtual atau yang dikenal dengan *cryptocurrencies* menjadi salah satu instrumen keuangan yang banyak mendapat perhatian masyarakat global. Hal ini karena kemudahan dan keuntungan yang ditawarkannya. *Cryptocurrency* banyak digunakan dalam transaksi online maupun diperjualbelikan sebagai investasi. Beberapa *cryptocurrency* yang paling banyak digunakan di seluruh dunia adalah Bitcoin, Ethereum, dan Tether.

Cryptocurrency sendiri menggunakan database terdistribusi, yang dikenal sebagai 'blockchain' dalam operasinya. Blockchain adalah kumpulan satu atau lebih blok yang membentuk rantai(Noorsanti, Yulianton, dan Hadiono, 2018). Setiap blok terdiri atas 3 elemen yaitu data, nilai hash dari blok saat ini, dan nilai hash dari blok sebelumnya. Data yang disimpan dalam blok tergantung pada tipe blok. Sebagai contoh, komponen data pada teknologi blockchain Bitcoin berisi detail transaksi seperti penerima, pengirim, dan nilai koin. Ada beberapa mekanisme atau teknik yang dipakai di dalam blockchain untuk menjamin keamanannya. Salah satunya adalah pemanfaatan teknik hash dari kriptografi yang membuat blok akan memiliki nilai hash yang mengidentifikasi blok dan seluruh isinya yang bersifat unik. Nilai hash akan sekaligus dihitung ketika blok dibuat. Teknik ini akan membuat blockchain menjadi lebih aman, karena jika ada yang mengubah salah satu blok dalam rantai blok maka nilai hashnya akan berubah dan blok berikutnya juga akan menjadi tidak valid lagi karena tidak menyimpan nilai hash yang valid dari blok sebelumnya. Perubahan yang dilakukan pada sebuah blok akan mengakibatkan seluruh rantai blok menjadi tidak valid. Penggunaan konsep ini membuat *cryptocurrency* sulit untuk dipalsukan.

Meskipun demikian, tidak berarti sistem tersebut benar-benar aman dari serangan hacker. Ancaman pembajakan oleh pihak yang tidak bertanggung jawab menjadi risiko bagi seluruh kegiatan yang dilakukan secara online sejak lama. Oleh karena itu, dalam Makalah ini akan dilakukan analisis algoritma fungsi hash yang termasuk dalam kelas SHA2 yang biasanya

digunakan pada sistem keamanan *cryptocurrency*, yaitu SHA-256 untuk dibandingkan dengan algoritma kawannya dalam kelas yang sama, yaitu SHA-512.

II. LANDASAN TEORI

2.1 Kriptografi

Kriptografi berasal dari Bahasa Yunani, yaitu "kryptos" yang berarti tersembunyi atau rahasia dan "graphein" yang berarti menulis. Secara harfiah, makna kriptografi adalah menulis secara tersembunyi untuk menyampaikan pesan-pesan yang perlu dijaga kerahasiaannya. Kriptografi merupakan cabang ilmu pengetahuan yang membahas tentang cara pengubahan pesan agar pesan tersebut tidak dapat dibaca oleh pihak yang tidak berwenang. Terdapat empat layanan yang dapat disediakan oleh kriptografi, yaitu kerahasiaan pesan (confidentiality), keaslian pesan (data integrity), otentikasi pengirim dan penerima pesan (authentication), dan anti penyangkalan (non repudiation).

2.2 Blockchain

Pada sebuah sistem blockchain, basis data tidak hanya dikelola satu pihak namun dikelola beberapa pihak sehingga basis data bersifat terdistribusi. Untuk memasukkan data baru ke dalam blockchain juga tidak dapat dilakukan oleh satu pihak, melainkan harus dilakukan oleh beberapa pihak. Ketika data baru masuk ke dalam blockchain, salah satu atau beberapa pengelola blockchain akan memvalidasi data tersebut. Apabila data tersebut dinyatakan valid oleh salah satu pengelola, maka data tersebut akan masuk ke dalam daftar tunggu dan disebarkan kepada seluruh pengelola.

Pengelola akan melakukan pengecekan terhadap validitas tersebut dan memasukkan ke daftar tunggunya masing-masing jika data yang diterima valid. Setelah daftar tunggu mencapai batas tertentu, pengelola yang terpilih melalui metode konsensus tertentu akan membuat blok baru menggunakan daftar tunggu yang telah diterima. Setelah blok baru ditambahkan pada blockchain yang lokal pengelola, blok baru tersebut akan disebarkan ke seluruh jaringan. Pengelola lain akan memvalidasi blok tersebut dan memasukkan ke blockchain-nya masing-masing apabila blok yang diterima valid. Misalkan terdapat suatu pihak ingin melakukan transaksi dan ingin menambahkan data transaksi tersebut ke dalam blockchain. Maka pihak tersebut akan menambahkan data transaksi tersebut ke dalam daftar tunggu miliknya kemudian menyebarkan data transaksi tersebut ke pihak lain.

Pihak pengelola blockchain lain yang menerima data transaksi akan memasukkan ke dalam daftar tunggu miliknya jika data transaksi tersebut valid. Kemudian akan terpilih salah satu pihak melalui prosedur konsensus tertentu untuk membuat block baru dari data transaksi yang telah ada pada daftar tunggu. Blok yang baru terbentuk tersebut akan di beri timestamp dan hash block sebelumnya. Blok yang baru terbentuk tersebut dikirim ke seluruh pihak yang mengelola node melalui jaringan.

Setiap pihak yang mengelola node akan melakukan pengecekan validitas dari blok baru tersebut, jika blok tersebut valid, setiap pihak akan menambah blok tersebut ke blockchain yang dikelolanya. Berikut diagram kerja sederhana blockchain tersebut. Setelah blok baru tersebut bergabung ke dalam blockchain, maka data transaksi yang tersimpan akan bersifat semipermanen. Apabila ada salah satu pihak yang melakukan perubahan pada sebuah blok maka pihak lain akan mengetahuinya melalui kode hash blok terakhir yang ditambahkan pada blockchain.

2.3 Fungsi Hash

Fungsi hash adalah fungsi yang menerima masukan string yang panjangnya sembarang dan mengonversinya menjadi string keluaran yang panjangnya tetap. Fungsi hash yang dihasilkan biasanya dituliskan dalam notasi persamaan sebagai berikut:

$$h = H(M)$$

Pada persamaan di atas,

h = nilai hash yang dihasilkan

H = fungsi hash itu sendiri

M = message atau pesan yang akan diubah dan dikonversikan menjadi nilai hash (hash value).

Suatu fungsi dikatakan sebuah fungsi hash jika memiliki sifat-sifat sebagai berikut:

1) Fungsi H dapat diterapkan pada blok data berukuran berapa pun.

2) H menghasilkan nilai (h) dengan panjang tetap.

3) H(x) mudah dihitung untuk setiap nilai x yang diberikan.

4) Untuk setiap h yang dihasilkan, tidak mungkin dikembalikan nilai x sedemikian hingga H(x) = h. Itulah sebabnya fungsi H dikatakan fungsi hash satu-arah (oneway hash function)

5) Untuk setiap x yang diberikan, tidak mungkin mencari y , x sedemikian sehingga H(y) = H(x).

6) Tidak mungkin mencari pasangan x dan y sedemikian sehingga H(x) = H(y).

2.4 Secure Hash Algorithm (SHA)

Algoritma fungsi SHA dapat dideskripsikan dan dibagi menjadi dua bagian: *preprocessing* dan perhitungan *message digest*. Preprocessing melibatkan proses penambahan bit pengganjal (padding bits), membagi pesan menjadi blok-blok dengan panjang tertentu, serta mengeset nilai awal untuk digunakan pada perhitungan *message digest*. Pada proses perhitungan message digest, dilakukan proses pembangkitan *message schedule* dari pesan, dan kemudian message schedule tersebut, bersama dengan fungsi-fungsi lainnya serta konstanta-konstanta yang telah terdefinisi, digunakan secara iteratif untuk membangkitkan nilai hash akhir.

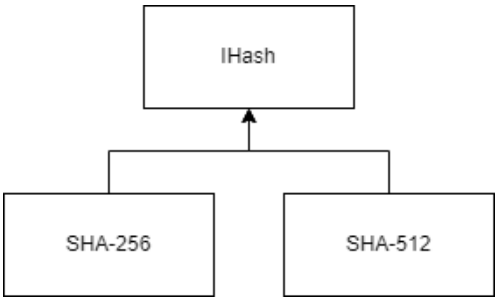
Untuk mempermudah implementasi fungsi hash, maka perlu dibuat suatu *interface* umum yang seragam yang berguna dalam pengaksesan objek-objek instansiasi dari kelas fungsi hash tersebut. Dengan cara pengaksesan yang seragam, maka implementasi kedua fungsi hash tersebut juga dapat dibuat semirip mungkin. Sehingga, kedua jenis fungsi hash yang akan

diimplementasi dapat dibandingkan secara langsung dalam segi performa.

Operasi-operasi atau metode yang diperlukan pada penggunaan algoritma SHA secara umum adalah Init(), Update(), dan Final(). Fungsi Init() akan melakukan inisiasi variabel dengan konstanta-konstanta algoritma SHA yang bersangkutan. Fungsi Update() berguna untuk menambahkan sebuah byte ke dalam pesan yang akan diproses. Fungsi Final() berguna untuk mengakhiri proses komputasi SHA sekaligus menambahkan bit-bit pengganjal, serta mengembalikan *message digest* akhir yang dihasilkan. Selain itu, dibutuhkan juga fungsi Transform() yang digunakan secara internal, untuk melakukan kalkulasi nilai hash sesuai jenis algoritma SHA yang digunakan.

Oleh karena itu, untuk mengimplementasi fungsi SHA, dibutuhkan definisi kelas abstrak IHash dalam bahasa C++ yang akan digunakan sebagai *parent class* bagi algoritma SHA yang diimplementasi (SHA-256 dan SHA-512), yang mendeklarasikan fungsi-fungsi di atas. Fungsi-fungsi tersebut dideklarasikan sebagai virtual, dan akan direalisasi di kelas anak yang merupakan implementasi konkret dari suatu algoritma SHA. Untuk lebih jelasnya, diagram kelas dapat dilihat pada Gambar 1.

Selain itu, diperlukan juga tambahan variabel-variabel dan makro-makro yang berguna pada implementasi ketiga algoritma SHA secara keseluruhan, seperti variabel untuk menyimpan nilai hash sementara, variabel panjang pesan, makro yang mendefinisikan fungsi-fungsi SHA seperti ROTR, ROTL, MAJ, serta makro untuk konversi tipe data. Kelas IHash ini diimplementasi dalam file IHash.h. Berikut ini adalah definisi makro dalam kelas IHash.



Gambar 1. Diagram Kelas

Algoritma	Ukuran Pesan (bit)	Ukuran Block (bit)	Ukuran Word (bit)	Ukuran Massage Digest (bit)	Bit Security (bit)
SHA-256	$< 2^{64}$	512	32	256	128
SHA-512	$< 2^{128}$	1024	64	512	256

Tabel 1. Perbedaan Karakteristik Algoritma SHA-256 dan SHA-512

III. PEMBAHASAN

3.1 SHA-256

SHA-256 beroperasi seperti MD4, MD5, dan SHA-1: Pesan yang akan di-hash adalah yang pertama

(1) dipadatkan dengan panjangnya sedemikian rupa sehingga hasilnya adalah kelipatan 512 bit, lalu

(2) diuraikan menjadi blok pesan 512-bit $M^{(1)};M^{(2)};\ldots;M^{(N)}$.

Blok pesan diproses satu per satu: Dimulai dengan nilai tetap hash yang awal H(0), lalu dihitung secara berurutan

$$H^{(i)} = H^{(i-1)} + C_{M(i)}(H^{(i-1)}),$$

C adalah fungsi kompresi SHA-256 dan + berarti *'word-wise'* mod 2^{32}. $H^{(N)}$ adalah hash dari M .

Fungsi kompresi SHA-256 beroperasi pada blok pesan 512-bit dan blok pesan 256-bit. Pada dasarnya ini adalah algoritma cipher blok 256-bit yang mengenkripsi nilai hash menengah menggunakan blok pesan sebagai kunci. Karenanya ada dua komponen utama untuk dijelaskan:

(1) fungsi kompresi SHA-256, dan

(2) jadwal pesan SHA-256.

$\oplus$	bitwise XOR
$\wedge$	bitwise AND
$\vee$	bitwise OR
$\neg$	bitwise complement
$+$	mod 2^{32} addition
R^n	right shift by n bits
S^n	right rotation by n bits

Gambar 2. Notasi yang bekerja pada kata-kata 32-bit.

Nilai hash awal $H^{(0)}$ adalah urutan kata 32-bit berikut (yang diperoleh dengan mengambil bagian pecahan dari akar kuadrat dari delapan bilangan prima pertama):

$$H_1^{(0)} = 6a09e667$$
$$H_2^{(0)} = bb67ae85$$
$$H_3^{(0)} = 3c6ef372$$
$$H_4^{(0)} = a54ff53a$$
$$H_5^{(0)} = 510e527f$$
$$H_6^{(0)} = 9b05688c$$
$$H_7^{(0)} = 1f83d9ab$$
$$H_8^{(0)} = 5be0cd19$$

Konstanta dalam SHA-256 adalah sebagai berikut (64 buah):

```
428a2f98 71374491 b5c0fbcf e9b5dba5
3956c25b 59f111f1 923f82a4 ab1c5ed5
d807aa98 12835b01 243185be 550c7dc3
72be5d74 80deb1fe 9bdc06a7 c19bf174
e49b69c1 efbe4786 0fc19dc6 240ca1cc
2de92c6f 4a7484aa 5cb0a9dc 76f988da
983e5152 a831c66d b00327c8 bf597fc7
c6e00bf3 d5a79147 06ca6351 14292967

27b70a85 2e1b2138 4d2c6dfc 53380d13
650a7354 766a0abb 81c2c92e 92722c85
a2bfe8a1 a81a664b c24b8b70 c76c51a3
d192e819 d6990624 f40e3585 106aa070
19a4c116 1e376c08 2748774c 34b0bcb5
391c0cb3 4ed8aa4a 5b9cca4f 682e6ff3
748f82ee 78a5636f 84c87814 8cc70208
90befffa a4506ceb bef9a3f7 c67178f2
```

Dalam implementasinya, kelas SHA-256 didefinisikan sebagai turunan kelas IHash. Pada kelas SHA-256 didefinisikan juga dua buah makro ukuran block, yaitu SHA-256_BLOCK_SIZE dengan SHA-256_DIGEST_SIZE, serta empat buah fungsi yang digunakan pada proses komputasi SHA-256. Kemudian, seluruh fungsi yang dideklarasikan pada SHA256.h akan diimplementasi pada SHA256.cpp.

Fungsi Init() akan melakukan inisiasi variabel m_H yang menyimpan nilai hash sementara, serta menginisiasi variabel m_Length dan m_TotalLength yang menyimpan panjang pesan dengan nilai 0.

Fungsi Update() menerima dua buah parameter masukan, yang pertama adalah data yang akan ditambahkan sebagai pesan yang akan di-hash, dan yang kedua adalah panjang data tersebut. Jika data yang dimasukkan sudah mencapai panjang satu block, maka block tersebut akan langsung dimasukkan ke fungsi Transform() dan variabel m_Block akan diisi dengan nilai data selanjutnya.

Fungsi Final() menerima satu parameter yang berguna untuk menyimpan hasil keluaran dari fungsi ini yaitu hasil nilai message digest akhir. Fungsi Final() akan melakukan proses penambahan bit-bit pengganjal pada akhir pesan dan melakukan Transform() pada block terakhir untuk mendapatkan message digest. Nilai message digest tersebut kemudian akan disimpan pada variabel yang diberikan pada parameter fungsi ini.

Fungsi Transform() pada SHA-256 menerima dua parameter masukan, yaitu block pesan yang akan dikomputasi, dan panjang block pesan tersebut. Untuk membangkitkan nilai message digest, dilakukan proses komputasi yang melibatkan 64 putaran untuk tiap block.

3.2 SHA-512

SHA-512 adalah varian dari SHA-256 yang beroperasi pada delapan kata 64-bit. Pesan yang akan di-hash adalah yang pertama

(1) dipadatkan dengan panjangnya sedemikian rupa sehingga hasilnya adalah kelipatan 1024 bit, dan kemudian (2) diuraikan menjadi blok pesan 1024-bit $M^{(1)};M^{(2)};\ldots;M^{(N)}$.

Blok pesan diproses satu per satu: Dimulai dengan nilai tetap hash awal $H^{(0)}$, lalu dihitung secara berurutan

$$H^{(i)} = H^{(i-1)} + C_{M(i)} (H^{(i-1)}),$$

C adalah fungsi kompresi SHA-512 dan + berarti '*wors-wise'* mod 2^{64}. $H^{(N)}$ adalah hash dari M .

Fungsi kompresi SHA-512 beroperasi pada blok pesan 1024-bit dan nilai hash perantara 512-bit. Pada dasarnya ini adalah algoritma cipher blok 512-bit yang mengenkripsi nilai hash menengah menggunakan blok pesan sebagai kunci. Oleh karena itu, ada dua komponen utama untuk dijelaskan:

(1) fungsi kompresi SHA-512, dan

(2) jadwal pesan SHA-512.

$\oplus$	bitwise XOR
$\wedge$	bitwise AND
$\vee$	bitwise OR
$\neg$	bitwise complement
$+$	mod 2^{64} addition
R^n	right shift by n bits
S^n	right rotation by n bits

Gambar 3. Notasi yang bekerja pada kata-kata 64-bit.

Nilai hash awal $H^{(0)}$ adalah urutan kata 64-bit berikut (yang diperoleh dengan mengambil bagian pecahan dari akar kuadrat dari delapan bilangan prima pertama):

$$H_1^{(0)} = \texttt{6a09e667f3bcc908}$$
$$H_2^{(0)} = \texttt{bb67ae8584caa73b}$$
$$H_3^{(0)} = \texttt{3c6ef372fe94f82b}$$
$$H_4^{(0)} = \texttt{a54ff53a5f1d36f1}$$
$$H_5^{(0)} = \texttt{510e527fade682d1}$$
$$H_6^{(0)} = \texttt{9b05688c2b3e6c1f}$$
$$H_7^{(0)} = \texttt{1f83d9abfb41bd6b}$$
$$H_8^{(0)} = \texttt{5be0cd19137e2179}$$

Konstanta dalam SHA-512 adalah sebagai berikut (80 buah):

```
428a2f98d728ae22  7137449123ef65cd
b5c0fbcfec4d3b2f  e9b5dba58189dbbc
3956c25bf348b538  59f111f1b605d019
923f82a4af194f9b  ab1c5ed5da6d8118
d807aa98a3030242  12835b0145706fbe
243185be4ee4b28c  550c7dc3d5ffb4e2
72be5d74f27b896f  80deb1fe3b1696b1
9bdc06a725c71235  c19bf174cf692694
e49b69c19ef14ad2  efbe4786384f25e3
0fc19dc68b8cd5b5  240ca1cc77ac9c65
2de92c6f592b0275  4a7484aa6ea6e483
5cb0a9dcbd41fbd4  76f988da831153b5
983e5152ee66dfab  a831c66d2db43210
b00327c898fb213f  bf597fc7beef0ee4
c6e00bf33da88fc2  d5a79147930aa725
06ca6351e003826f  142929670a0e6e70
27b70a8546d22ffc  2e1b21385c26c926
4d2c6dfc5ac42aed  53380d139d95b3df
650a73548baf63de  766a0abb3c77b2a8
81c2c92e47edaee6  92722c851482353b
```

```
a2bfe8a14cf10364  a81a664bbc423001
c24b8b70d0f89791  c76c51a30654be30
d192e819d6ef5218  d69906245565a910
f40e35855771202a  106aa07032bbd1b8
19a4c116b8d2d0c8  1e376c085141ab53
2748774cdf8eeb99  34b0bcb5e19b48a8
391c0cb3c5c95a63  4ed8aa4ae3418acb
5b9cca4f7763e373  682e6ff3d6b2b8a3
748f82ee5defb2fc  78a5636f43172f60
84c87814a1f0ab72  8cc702081a6439ec
90befffa23631e28  a4506cebde82bde9
bef9a3f7b2c67915  c67178f2e372532b
ca273eceea26619c  d186b8c721c0c207
eada7dd6cde0eb1e  f57d4f7fee6ed178
06f067aa72176fba  0a637dc5a2c898a6
113f9804bef90dae  1b710b35131c471b
28db77f523047d84  32caab7b40c72493
3c9ebe0a15c9bebc  431d67c49c100d4c
4cc5d4becb3e42b6  597f299cfc657e2a
5fcb6fab3ad6faec  6c44198c4a475817
```

Dalam implementasinya, kelas SHA-512 didefinisikan sebagai turunan kelas IHash. Sama seperti SHA-256, pada kelas SHA-512 didefinisikan juga dua buah makro ukuran block, yaitu SHA-256_BLOCK_SIZE dengan SHA-256_DIGEST_SIZE, serta empat buah fungsi yang digunakan pada proses komputasi SHA-512. Perbedaan utama dengan kelas SHA-256 adalah tipe data yang digunakan, dimana pada SHA-512 digunakan unsigned long long yang berukuran 64-bit.

Seluruh fungsi yang dideklarasikan pada SHA-512.h akan diimplementasi pada SHA-512.cpp.

Fungsi Init() akan melakukan inisiasi variabel m_H yang menyimpan nilai hash sementara, serta menginisiasi variabel m_Length dan m_TotalLength yang menyimpan panjang pesan dengan nilai 0.

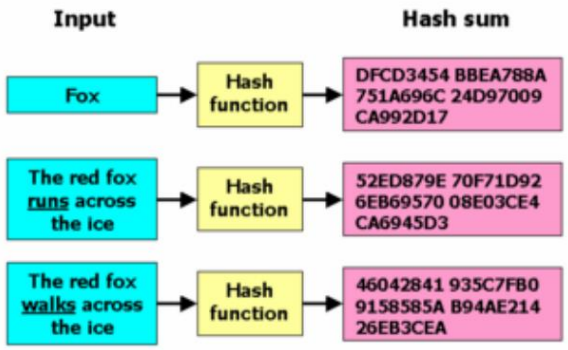
Fungsi Update() menerima dua buah parameter masukan, yang pertama adalah data yang akan ditambahkan sebagai pesan yang akan di-hash, dan yang kedua adalah panjang data tersebut. Jika data yang dimasukkan sudah mencapai panjang satu block, maka block tersebut akan langsung dimasukkan ke fungsi. Transform() dan variabel m_Block akan diisi dengan nilai data selanjutnya.

Fungsi Final() menerima satu parameter yang berguna untuk menyimpan hasil keluaran dari fungsi ini yaitu hasil nilai message digest akhir. Fungsi Final() akan melakukan proses penambahan bit-bit pengganjal pada akhir pesan dan melakukan Transform() pada block terakhir untuk mendapatkan message digest. Nilai message digest tersebut kemudian akan disimpan pada variabel yang diberikan pada parameter fungsi ini.

Fungsi Transform() pada SHA-512 menerima dua parameter masukan, yaitu block pesan yang akan dikomputasi, dan panjang block pesan tersebut. Untuk membangkitkan nilai message digest, dilakukan proses komputasi yang melibatkan 80 putaran untuk tiap block.

3.3 Aspek Keamanan

Kedua algoritma SHA tersebut sudah tergolong aman. Algoritma SHA dikatakan aman karena tidak mungkin secara komputasional untuk menemukan pesan yang berpasangan dengan sebuah message digest tertentu. Selain itu, tidak mungkin juga secara komputasional untuk menghasilkan dua pesan berbeda yang menghasilkan message digest yang sama. Sedikit perubahan pada pesan kemungkinan besar akan menghasilkan message digest yang sangat berbeda.



Ganbar 4. Avalance Effects

Pada gambar diatas, mengubah satu kata bagian pesan dari runs menjadi walks dapat menghasilkan nilai message digest yang jauh berbeda.

Keamanan algoritma SHA dibuktikan jika terjadi perubahan sedikit pada pesan, maka message digest yang dihasilkan dapat berbeda jauh. Akan tetapi, tetap saja tidak ada keamanan yang sempurna. Algoritma SHA masih dapat diserang. SHA-2 (SHA-256 dan SHA-512) dirancang untuk menutupi kekurangan yang dimiliki oleh kelas SHA sebelumnya, dengan menambah jumlah putaran (loop) dalam algoritmanya dan meningkatkan panjang message digest yang dihasilkannya. Sampai saat ini belum ditemukan collision pada algoritma SHA-256 maupun SHA-512. Jadi,

dapat dikatakan bahwa SHA-256 dan SHA-512 lebih aman dibandingkan kelas SHA sebelumnya, karena tingkat kompleksitasnya yang lebih tinggi serta message digest yang dihasilkannya lebih panjang.

V. Kesimpulan

Berdasarkan pembahasan sebelumnya, dapat diambil kesimpulan bahwa fungsi hash SHA-256 dan SHA-512 memiliki karakteristik dan implementasi yang hampir mirip. Akan tetapi, keduanya menggunakan konstanta, panjang variabel, fungsi, jumlah loop, serta menghasilkan nilai message digest yang panjangnya berbeda. Dengan mengeksekusi program pada mesin berarsitektur 64-bit, waktu eksekusi SHA-512 dapat lebih cepat. Hal ini disebabkan SHA-512 menggunakan variabel dengan panjang 64-bit. Dalam aspek keamanan, SHA-256 dan SHA-512 tidak jauh berbeda, di mana keduanya dianggap aman karena tingkat kompleksitas yang tinggi dan message digest yang dihasilkan lebih panjang sehingga dapat digunakan pada sistem keamanan untuk *cryptocurrency*.

VI. Ucapan Terima Kasih

Penulis mengucapkan puji syukur kepada Tuhan Yang maha Esa atas karunia-Nya sehingga penulis dapat menyelesaikan makalah ini. Penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu penulis dalam menyelesaikan makalah ini. Penulis juga menyampaikan terima kasih kepada seluruh dosen pengampu mata kuliah Matematika Diskrit, utamanya Ibu Fariska Zakhralativa Ruskanda sebagai dosen pengampu kelas K2 yang telah mengajarkan penulis berbagai ilmu penting yang penulis butuhkan untuk menyusun makalah ini. Selain itu, penulis ingin mengucapkan terima kasih kepada orang-orang yang membahas terkait materi yang dibahas penulis melalui internet yang penulis jadikan sebagai referensi dalam pembuatan makalah ini karena telah menambah wawasan penulis mengenai topik yang penulis bawa di makalah ini.

References

[1] Choiri, E. (2020). Pengertian Kriptografi, Sejarah & Jenis Algoritmanya. Diakses pada 11 Desember 2022, dari Pengertian Kriptografi, Sejarah & Jenis Algoritmanya - Qwords

[2] Descriptions of SHA-256, SHA-384, and SHA-512. Diakses pada 11 Desember 2022, dari yarbrelease.dvi (ethereum.org)

[3] Gladman, Brian. (2006). SHA1, SHA2, HMAC and Key Derivation in C. Diakses pada 12 Desember 2022.

[4] Regina Dionne Aurelia Hadiprodjo, (2022). Analisis Penerapan Kriptografi pada Masalah Keamanan Bitcoin. Diakses pada 11 Desember 2022.

[5] Federal Information Processing Standards Publication. Federal Information Processing Standards (FIPS) Publication 180-2 : Secure Hash Standard. National Institute of Standards and Technology, USA. Diakses pada 11 Desember 2022.

[6] Eastlake, D., Jones, P. (2001). Request For Comment (RFC)-3174 : US Secure Hash Algorithm 1 (SHA1). The Internet Society. Diakses pada 11 Desember 2022.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Desember 2022

Asyifa Nurul Shafira
13521125