Analisis Perbandingan Penggunaan SHA-1 dan MD5 pada aplikasi peer nTorrent dalam Komunikasi Peer to Peer

Irdham Mikhail Kenjibriel (13508111)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
irdhamkenjibriel@students.itb.ac.id

***Abstract*— *Message Digest 5* atau yang biasa disingkat dengan *MD5* adalah algoritma yang termasuk dalam algoritma enkripsi modern dan berada dalam kelompok algoritma fungsi hash kriptografi. Algoritma ini sudah terbukti sangat sulit dipecahkan karena tidak dapat didekripsi atau hanya bersifat satu arah. Oleh karena itu penggunaannya untuk mengenkripsi dank arena setiap hasil enkripsinya dapat dikatakan selalu menghasilkan nilai yang unik maka algoritma ini sering digunakan untuk perbandingan. Perbandingan yang dimaksud penulis disini adalah untuk membandingkan apakah suatu file itu sama secara isi antara satu dengan yang lainnya ddengan cara membandingkan hasil MD5. Begitu pula dengan algoritma SHA-1, yang memilki bebera sifat yang mirip dengan penggunaan MD5. Namun kedua algoritma ini memiliki beberapa perbedaan yang menjadi kelebihan dan kekurangan dari masing- masing algoritma tersebut. Pada makalah ini penulis mencoba membahas penggunaan kedua algoritma tersebut dan membandingkan kelebihan dan kekurangan masing.**

Pada makalah ini ada beberapa hal yang akan dibahas oleh penulis sebagai parameter yang dipakai untuk membandingkan kedua algoritma diatas yang pertama adalah kefisienan dan kefektifan waktu, kefisienan dan kefektifan *peforma* yang tercapai, dan yang terakhir adalah keamanan yang ada jika terjadi serangan- serangan kriptanalis.

Index Terms*— *S Efektif, Efisien, Keamanan, Kriptanalis, MD5, Peforma, SHA-1, Waktu

I. PENDAHULUAN

Aplikasi peer to peer dewasa ini mulai menjadi trend atau gaya baru dalam beberapa cara untuk berbagi file antara individu satu dengan yang lainnya. Hal tersebut didorong karena aplikasi yang sebelumnya seperti beberapa file sharing kebanyakan memiliki arsitektur *client- server* sehingga memiliki beberapa kelemahan jika penulis bandingkan dengan peer to peer. Sebagai contoh aplikai peer to peer yang mulai banyak digunakan oleh orang –orang didunia ini adalah bit torrent. Namun tahukah anda bahwa sebenarnya didalam semua aplikasi file sharing yang menggunakan arsitektur peer to peer dalam hal ini termasuk juga bit torrent menggunakan algoritma fungsi hash kriptografi yang berguna dalam membandingkan atau mencocokkan apakah file yang dikirim benar file yang akan diterima oleh setiap peer lainnya. Oleh karena itu penulis mencoba mengimplementasikan dua algoritma kedalam suatu aplikasi peer to peer yang penuli buat dan membandingkan kelebihan dan kekurangannya dalam hal ini algoritma tersebut adalah SHA-1 dan MD5.

Sebelumnya pada makalah kali ini aplikasi peer to peer yang dipakai adalah nTorrent. Aplikasi nTorrent merupakan aplikasi yang dibuat oleh penulis untuk tugas besar mata kuliah sistem terdistribusi dan makalah ke dua kriptografi ini. Aplikasi ini berguna untuk komunikasi dengan protocol peer to peer. Aplikasi ini fungsinya mirip dengan aplikasi torrent pada umumnya. Didalam aplikasi ini mengimplementasikan salah satu algoritma enkripsi yaitu SHA-1. Algoritma ini dipakai untuk mengecek ke validan dari file yang diunduh oleh user sehingga dapat dijalankan sesuai dengan file aslinya yang diunggah oleh orang lain.

Gambar 1. Beberapa Aplikasi yang memanfaatkan arsitektur peer to peer

Pada aplikasi peer to peer tersebut nTorrent, penulis akan mencoba membandingkan beberapa hal berikut ini :

- Bagaimana perbandingan tingkat kefisienan dan kefektifan waktu yang digunakan untuk

pengiriman file jika enkripsi filenya menggunakan MD5 atau SHA-1?

- Bagaimana perbandingan tingkat kefisienan dan kefektifan *peforma* yang dicapai yang digunakan untuk pengiriman file jika enkripsi filenya menggunakan MD5 atau SHA-1?
- Bagaimana perbandingan tingkat tingkat keamanan yang ada jika terjadi serangan-serangan kriptanalis pada saat pengiriman file jika enkripsi filenya menggunakan MD5 atau SHA-1?

Parameter yang pertama yang digunakan adalah waktu yang digunakan untuk mengenkripsi file yang akan dikirim. Dalam hal ini berkaitan dengan seberapa cepat suatu file dengan besar tertentu dapat di enkripsi dengan menggunakan algoritma fungsi hash kriptografi ini, MD5 dan SHA-1.

Parameter yang kedua yang digunakan adalah resource computer yang terpakai dalam penggunaannya saat digunakan untuk mengenkiripsi file yang ada. Hal ini dapat dilihat dari performa dari aplikasi tersebut dan resource computer yang dipakai baik untuk perhitungan dalam penggunaan algoritma fungsi hash kriptografi ini, MD5 dan SHA-1.

Parameter yang ketiga adalah tingkat keamanan yang digunakan dalam mengenkripsi file- file yang akan dikirim dengan menggunakan algoritma fungsi hash kriptografi ini, MD5 dan SHA-1. Tipe serangan yang dimaksud oleh penulis adalah brute force dan kebenaran atau pengecekan kevalidan file yang diterima atas serangan pengguna lain untuk mengubah isi yang dikirim oleh seorang pengguna kepada pengguna lainnya..

Dari ketiga para meter tersebut penulis berharap dapat mewakili dari semua factor yang ada untuk dapat menentukan algoritma mana yang diraasa lebih cocok untuk digunakan dalam mengenkripsi file- file yang akan dikirimkan melalui aplikasi peer to peer ini.

II. DASAR TEORI

A. Peer to Peer

P2P merupakan singkatan dari Peer-to-Peer atau teknologi dari “ujung” ke “ujung” pertama kali di luncurkan dan dipopulerkan oleh aplikasi-aplikasi “berbagi-berkas” (*file sharing*) seperti Napster dan KaZaA. Pada konteks ini teknologi P2P memungkinkan para pengguna untuk berbagi, mencari dan mengunduh berkas.

Sistem P2P yang sebenarnya adalah suatu sistem yang tidak hanya menghubungkan “ujung” satu dengan lainnya, namun ujung-ujung ini saling berhubungan secara dinamis dan berpartisipasi dalam mengarahkan lalu lintas komunikasi informasi-, pemrosesan-, dan penugasan pembagian bandwidth yang intensif, dimana bila sistem ini tidak ada, tugas-tugas ini biasanya diemban oleh server pusat.

Aplikasi P2P yang sebenarnya memerlukan satuan tim-tim kecil dengan ide cemerlang untuk mengembangkan perangkat lunak dan bisnis-bisnis yang mungkin dilakukan oleh perangkat tersebut – dan mungkin saja bisa membuat perusahaan besar yang sudah ada gulung tikar. P2P yang sebenarnya, bila diaplikasikan pada pasar yang sudah matang dan stabil adalah teknologi yang "mengganggu".

Ide mengenai konsep ini muncul kira-kira pada akhir dekade 1980-an, ketika jaringan komputer dan tentunya juga komputer telah mulai masuk ke dalam salah satu barang wajib dalam perusahaan, baik itu perusahaan kecil maupun besar. Tetapi, arsitektur ini berkembang dalam jaringan yang terlalu kecil untuk memiliki sebuah server yang terdedikasi, sehingga setiap komputer klien pun menyediakan layanan untuk berbagi data untuk melakukan kolaborasi antara pengguna.

Jaringan *peer-to-peer* pun mulai banyak digemari ketika Microsoft merilis sistem operasi Windows for Workgroups, meski sebelumnya sistem operasi MS-DOS (atau IBM PC-DOS) dengan perangkat MS-NET (atau PC-NET) juga dapat digunakan untuk tujuan ini. Karakteristik kunci jaringan tersebut adalah dalam jaringan ini tidak terdapat sebuah server pusat yang mengatur klien-klien, karena memang setiap komputer bertindak sebagai server untuk komputer klien lainnya. Sistem keamanan yang ditawarkan oleh metode ini terbilang lebih rendah dibandingkan dengan metode klien/server dan manajemen terhadapnya pun menjadi relatif lebih rumit.

Konsep ini pun kemudian berevolusi pada beberapa tahun terakhir, khususnya ketika jaringan Internet menjadi jaringan yang sangat besar. Hal ini mulai muncul kira-kira pada akhir dekade 1990-an, di saat banyak pengguna Internet mengunduh banyak berkas musik mp3 dengan menggunakan metode *peer-to-peer* dengan menggunakan program Napster yang menuai kritik pedas dari industri musik, seperti halnya Metallica dan banyak lainnya. Napster, pada saat dituntut oleh para pekerja industri musik, dikatakan memiliki anggota lebih dari 20 juta pengguna di seluruh dunia. Selanjutnya beberapa aplikasi juga dibuat dengan menggunakan konsep ini: eDonkey, Kazaa, BitTorrent, dan masih banyak lainnya. Meski banyak aplikasi peer-to-peer ini digunakan oleh pengguna rumahan, ternyata sistem ini juga diminati oleh banyak perusahaan juga.

Gambar 2. Arsitektur umum aplikasi peer to peer

B. Message Digest 5

Dalam kriptografi, MD5 (*Message-Digest algortihm 5*) ialah fungsi hash kriptografik yang digunakan secara luas dengan *hash value* 128-bit. Pada standart Internet (RFC 1321), MD5 telah dimanfaatkan secara bermacam-macam pada aplikasi keamanan, dan MD5 juga umum digunakan untuk melakukan pengujian integritas sebuah file.

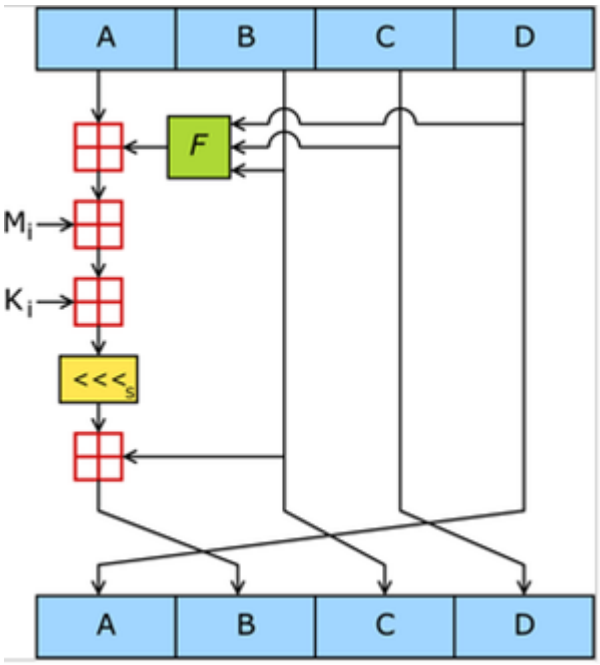
MD5 di desain oleh Ronald Rivest pada tahun 1991 untuk menggantikan *hash function* sebelumnya, MD4. Pada tahun 1996, sebuah kecacatan ditemukan dalam desainnya, walau bukan kelemahan fatal, pengguna kriptografi mulai menganjurkan menggunakan algoritma lain, seperti SHA-1 (klaim terbaru menyatakan bahwa SHA-1 juga cacat). Pada tahun 2004, kecacatan-kecacatan yang lebih serius ditemukan menyebabkan penggunaan algoritma tersebut dalam tujuan untuk keamanan jadi makin dipertanyakan.

menunjukkan perputaran bit kiri oleh *s*; *s* bervariasi untuk tiap-tiap operasi. menunjukan tambahan modulo 2^{32}. MD5 memproses variasi panjang pesan kedalam keluaran 128-bit dengan panjang yang tetap. Pesan masukan dipecah menjadi dua gumpalan blok 512-bit; Pesan ditata sehingga panjang pesan dapat dibagi 512. Penataan bekerja sebagai berikut: bit tunggal pertama, 1, diletakkan pada akhir pedan. Proses ini diikuti dengan serangkaian nol (0) yang diperlukan agar panjang pesan lebih dari 64-bit dan kurang dari kelipatan 512. Bit-bit sisa diisi dengan 64-bit integer untuk menunjukkan panjang pesan yang asli. Sebuah pesan selalu ditata setidaknya dengan 1-bit tunggal, seperti jika panjang pesan adalah kelipatan 512 dikurangi 64-bit untuk informasi panjang (panjang mod(512) = 448), sebuah blok baru dari 512-bit ditambahkan dengan 1-bit diikuti dengan 447 bit-bit nol (0) diikuti dengan panjang 64-bit.

Algoritma MD5 yang utama beroperasi pada kondisi 128-bit, dibagi menjadi empat *word* 32-bit, menunjukkan *A*, *B*, *C* dan *D*. Operasi tersebut di inisialisasi dijaga untuk tetap konstan. Algoritma utama kemudian beroperasi pada masing-masing blok pesan 512-bit, masing-masing blok melakukan pengubahan terhadap kondisi.Pemrosesan blok pesan terdiri atas empat tahap, batasan *putaran*; tiap putasan membuat 16 operasi serupa berdasar pada fungsi non-linear *F*, tambahan modular, dan rotasi ke kiri. Gambar satu mengilustrasikan satu operasi dalam putaran. Ada empat macam kemungkinan fungsi *F*, berbeda dari yang digunakan pada tiap-tiap putaran:

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

menunjukkan operasi logikan XOR, AND, OR dan NOT.



Gambar 3. Operasi pada MD5

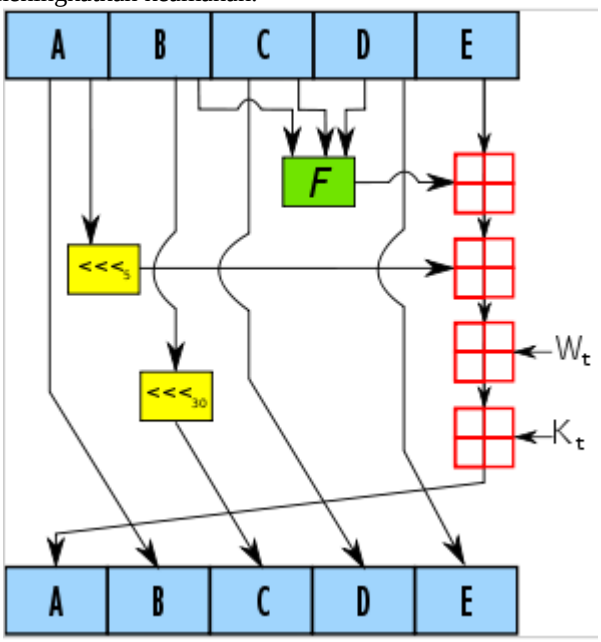
C. Secure Hash Algorithm 1

Dalam kriptografi, SHA-1 adalah fungsi hash kriptografi dirancang oleh National Security Agency dan diterbitkan oleh NIST sebagai US Federal Information Processing Standard. SHA singkatan dari Secure Hash Algorithm. Ketiga algoritma SHA yang terstruktur berbeda dan dibedakan sebagai SHA-0, SHA-1, dan SHA-2. SHA-1 ini sangat mirip dengan SHA-0, tapi mengoreksi kesalahan dalam spesifikasi hash SHA asli yang menyebabkan kelemahan signifikan. SHA-0 algoritma tidak diadopsi oleh banyak aplikasi. SHA-2 di sisi lain secara signifikan berbeda dari fungsi hash SHA-1.
SHA-1 adalah yang paling banyak digunakan pada fungsi hash SHA yang ada, dan bekerja dalam beberapa aplikasi keamanan secara luas digunakan dan protokol. Pada tahun

2005, kelemahan keamanan diidentifikasi dalam SHA-1, yaitu bahwa kelemahan matematika mungkin ada, menunjukkan bahwa fungsi hash kuat akan diinginkan [2] Meskipun tidak ada serangan yang berhasil belum dilaporkan pada varian SHA-2., mereka upaya algorithmically mirip dengan SHA-1 dan begitu juga dilakukan untuk mengembangkan alternatif perbaikan [3] [4] Sebuah standar hash baru, SHA-3, saat ini sedang dalam pengembangan -. suatu fungsi hash NIST kompetisi berlangsung dijadwalkan akan berakhir dengan pemilihan fungsi menang pada tahun 2012.

SHA-1 menghasilkan message digest 160-bit berdasarkan prinsip yang sama dengan yang digunakan oleh Ronald L. Rivest dari MIT dalam desain MD4 dan pesan algoritma MD5 mencerna, namun memiliki desain yang lebih konservatif.

Spesifikasi asli dari algoritma itu diterbitkan pada tahun 1993 sebagai Secure Hash Standard, FIPS PUB 180, oleh pemerintah AS standar lembaga NIST (Lembaga Nasional Standar dan Teknologi). Versi ini sekarang sering disebut sebagai SHA-0. Hal itu ditarik oleh NSA lama setelah publikasi dan digantikan oleh versi revisi, yang diterbitkan pada tahun 1995 di FIPS PUB 180-1 dan umumnya disebut sebagai SHA-1. SHA-1 berbeda dari SHA-0 hanya oleh rotasi bitwise tunggal dalam pesan jadwal fungsi kompresi, ini dilakukan, menurut NSA, untuk memperbaiki cacat dalam algoritma asli yang mengurangi keamanan kriptografi nya. Namun, NSA tidak memberikan penjelasan lebih lanjut atau mengidentifikasi cacat yang telah diperbaiki. Kelemahan yang kemudian dilaporkan di kedua SHA dan SHA-1. SHA-1 muncul untuk memberikan resistansi yang lebih besar terhadap serangan, mendukung pernyataan NSA yang mengubah meningkatkan keamanan.



Gambar 4. Operasi pada SHA-1

III. IMPLEMENTASI

A. Spesifikasi

Struktur File Metainfo (.torrent)

Pada umumnya, data dari file metainfo ditulis dalam format bencoded dan sebuah file metainfo dapat mendeskripsikan sebuah file (single-file) atau banyak file (multi-file). Tetapi pada tugas ini, data dari file metainfo tidak perlu ditulis dalam format bencoded. Sebuah file metainfo hanya digunakan untuk mendeskripsikan sebuah file saja. Data dari file metainfo adalah sebagai berikut:

1. Info: bagian informasi yang mendeskripsikan file yang ingin didistribusikan. Informasi dari file terdiri dari:
   a. Piece length: ukuran dari setiap piece dalam byte. Ukuran setiap piece sama kecuali ukuran piece yang terakhir mungkin berbeda. Ukuran piece biasanya merupakan bilangan kelipatan pangkat 2 (2, 4, 8, 16, 32, dan seterusnya). Ukuran piece yang umum digunakan adalah 256 KB, 512 KB dan 1 MB, sedangkan ukuran minimum dan maksimum satu piece adalah 32 KB dan 4 MB.
   b. Pieces: Data dari file di-hash dengan algoritma fungsi hash kriptografi (MD5 atau SHA- 1), nilai hash memiliki panjang 20 byte. String dari nilai hash tersebut merupakan nilai dari pieces. d. file name: string dari nama file.
   c. File length: ukuran file dalam byte. Ukuran maksimum dari fie yang dapat dikirim dapat ditentukan sendiri.
2. Announce: URL dari tracker dalam format string.

Format Total dari file metainfo yang ada jadi seperti ini dibawah ini:

```
‹file name› ‹spasi› ‹file length› ‹spasi› ‹piece length›
‹announce›
‹pieces›
```

Gambar 5. Format file metainfo yang akan dibuat

Tracker

Tracker bertugas untuk memperkenalkan satu client dengan client lainnya. Client yang ingin men-download file dari sebuah file torrent harus bertanya kepada tracker untuk mengetahui alamat client lain yang memiliki file tersebut.

Spesifikasi Protokol pada Tracker :

- Tracker Request Parameters

  Request dari client ke tracker disampaikan dalam format HTTP GET request. Request ini membantu tracker untuk mendata statistik beberapa torrent sekaligus. Adapun parameter request dari client ke tracker adalah sebagai berikut:

  - Info_hash: urlencoded 20 byte SHA-1 dari nilai info pada file metainfo. Pada tugas ini, info_hash diganti dengan file_name. file_name

adalah string dari nama file yang ingin di-download. file_name didapat dari info name pada file metainfo yang sudah di-hash dengan SHA-1 atau MD5.

- Port: nomor port dimana client ini sedang listening.
- Uploaded: jumlah byte yang telah di upload oleh client khusus untuk file file_name. (terhitung mulai dari client mengirimkan started event kepada tracker).
- Downloaded: jumlah byte yang telah di download oleh client khusus untuk file file_name. (terhitung mulai dari client mengirimkan started event kepada tracker).
- Left: jumlah byte yang masih harus di download oleh client khusus untuk file file_name.
- Event: jika parameter ini ada, isinya adalah salah satu di antara started, completed, stopped, (atau kosong yang dianggap sama seperti jika parameter ini tidak ada). Jika parameter ini tidak ada, maka request ini adalah request yang dilakukan secara periodik.
  - Started: request pertama dari client ke tracker harus menggunakan event started.
  - Stopped: harus dikirim ke tracker jika client ditutup/diberhentikan dengan sengaja.
  - Completed: harus dikirim ke tracker ketika download selesai dengan sempurna. Jika file sudah utuh (100% terdownload) ketika client started, pesan completed tidak perlu dikirimkan.

- Tracker Response Parameters

Setelah mendapatkan HTTP GET request dari client, tracker akan memberikan respon berupa dokumen teks. Dokumen teks tidak perlu dituliskan dalam format bencoded seperti pada penggunaan umum dari BitTorrent. Dokumen teks terdiri dari beberapa key, yaitu:

- Failure_reason: jika terjadi error, key lain tidak perlu ada. Nilainya adalah string dari pesan error kenapa request ini gagal. Jika tidak terjadi error, key ini tidak perlu ada.
- Warning_message: sama seperti failure reason, namun hanya peringatan dan key lain tetap diproses atau diperhatikan.
- Interval: interval (detik) dimana client harus mengirimkan lagi request ke tracker.
- complete: jumlah client yang memiliki file utuh, disebut sebagai seeders.
- incomplete: jumlah client non-seeder, disebut sebagai leechers.
- peers: peers merupakan list dari ip dan port peer dapat dituliskan dalam bentuk format string (ip) dan integer (port) atau dalam format 6 byte. 4 byte pertama adalah ip, 2 byte selanjutnya adalah port. Untuk tugas ini, peers akan dituliskan dalam format string dan integer. List dari ip dan port peer yang diberikan tidak boleh mengandung ip dan port peer dari client itu sendiri. List dalam peers defaultnya berjumlah 50. Jika jumlah peer pada keadaan nyatanya lebih sedikit, isi list juga lebih sedikit. Jika jumlah peer jauh lebih banyak, tracker akan secara acak memilih peer yang akan dimasukkan ke dalam list.

Format respon total adalah sebagai berikut:

```
failure_reason <spasi> <failure_reason_value>
warning_message <spasi> <warning_reason_value>
interval <spasi> <interval_value>
complete <spasi> <complete_value>
incomplete <spasi> <incomplete_value>
peers <spasi> <ip-peer-1>:<port-peer-1> <spasi> <ip-peer-2>:<port-peer-2> <spasi> ... <spasi>
<ip-peer-N>:<port-peer-N>
```

Gambar 6. Format tracker respon yang akan dibuat

Peer

Komunikasi peer-to-peer menggunakan TCP. Pada umumnya, komunikasi peer-to-peer digunakan untuk pertukaran piece yang dideskripsikan pada file metainfo. Peer/Client dapat mengatur state dari setiap koneksi kepada peer lainnya. State yang sudah didefinisikan adalah choked dan interseted.

Spesifikasi Protokol pada Peer :

- Data Types
  Semua integer pada protokol dituliskan dalam format 4 byte big-endian.
- Message flow
  Protokol ini dimulai dengan handshake. Kemudian, peer berkomunikasi dengan peer lain menggunakan message dengan format length-prefixed message. Handshake dan message dikirimkan dalam bentuk binary.
- Handshake
  Handshake digunakan untuk pesan yang pertama kali ditransimisikan oleh peer. Panjang handshake adalah 49+len(pstr) byte. Format handshake:
  <pstrlen><pstr><reserved><info_hash><peer_id>
  Keterangan:
  - Pstrlen: panjang dari string pstr, ditulis dalam byte. Panjang dari pstr adalah 19.
  - Pstr: string identifier dari protokol. pstr = “BitTorrent protocol”.
  - Reserved: reserved byte, dengan panjang 8 byte. Dapat diisikan dengan 8 buah byte 0.
  - Info_hash: 20 byte SHA-1 dari nilai info pada file metainfo. Pada tugas ini, info_hash diganti dengan file_name. file_name adalah string dari nama file yang ingin di-download. file_name didapat dari

info name pada file metainfo yang sudah di-hash dengan SHA-1 atau MD5 .

- peer_id: 20-byte string yang merupakan ID dari client. (tidak perlu) peer_id tidak perlu dimasukkan ke dalam pesan handshake.

Catatan: jika sebuah peer menerima handshake dengan file_name yang tidak dimilikinya, peer tersebut dapat mendrop koneksinya.

Message

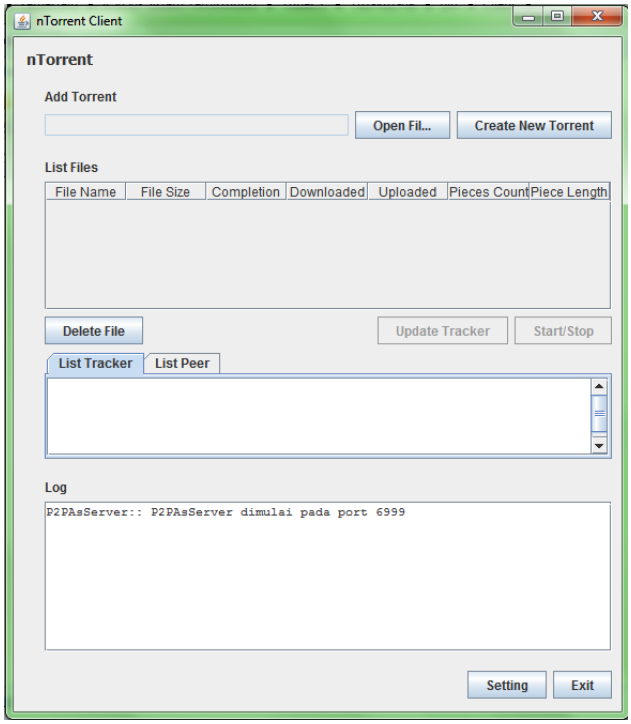
Message yang digunakan pada protokol ini memiliki format length-prefixed message. Format length-prefixed message: <length prefix><message ID><payload>.

Length prefix ditulis dalam format 4 byte big-endian. Message ID dituliskan 1 byte. Payload setiap message dapat bervariasi. Message-message yang ada adalah:

- Keep-alive: <len=0000>
  Message keep-alive harus digenerate dan dikirim secara periodik jika peer tidak berkomunikasi lagi dalam selang periode tersebut. Periode yang biasa digunakan adalah dua menit.
- Have: <len=0005><id=4><piece index>
  Payload message have adalah zero-based piece index yang dimiliki oleh peer.
- Bitfield: <len=0001+X><id=5><bitfield>
  Message bitfield adalah message pertama yang dikirimkan setelah handshake selesai dilakukan dan harus dikirimkan sebelum message-message yang lain dikirimkan. Tetapi, message bitfield tidak perlu dikirimkan jika peer sama sekali tidak memiliki piece. Message bitfield memiliki panjang yang bervariasi, X adalah panjang dari bitfield. Payload message ini adalah bitfield dimana bitfield merepresentasikan piece-piece yang dimiliki oleh peer. Byte pertama pada bitfield merepresentasikan piece index 0 – 7 (high – low bit), byte selanjutnya 8 -15, dan seterusnya. Untuk setiap piece yang dimiliki oleh peer, bit yang berkorespondensi diset menjadi 1 dan selain itu diset menjadi 0. Sisa bit (Spare bit) pada bagian akhir diisi dengan bit 0.
- Request:
  <len=0013><id=6><index><begin><length>
  Message request digunakan untuk me-request sebuah block. Payload message request adalah sebagai berikut:
  - Index: integer yang menunjukkan zero-based piece index.
  - Begin: integer yang menunjukkan zero-based offset dari piece.
  - Length: panjang dari block data dari piece.
- Piece:
  <len=0009+X><id=7><index><begin><block>
- Message piece memiliki panjang yang bervariasi, X adalah panjang dari block. Payload message piece adalah sebagai berikut:
  - index: integer yang menunjukkan zero-based piece index.
  - Begin: integer yang menunjukkan zero-based offset dari piece.
  - Block: block data dari piece.
- Cancel:
  <len=0013><id=8><index><begin><length>

Message cancel digunakan untuk membatalkan block request. Payloadnya sama dengan payload pada message request.

B. Tampilan Antar Muka



Gambar 7. Tampilan untuk aplikasi peer to peer nTorrent

Diatas merupakan tampilan antar muka yang penulis buat untuk melakukan upload dan download file. Semua status pengiriman dan pencocokan akan terlihat pada text field dibawah label log. Disana hasil perbandingan antara SHA-1 atau MD5 akan di tuliskan.

Sedangkan untuk antar muka tracker tidak dibuat dan hanya ada textual command yang dijalankan melalui command prompt.

C.Cara Kerja

Pertama kali data yang akan dikirim akan dibagi- bagi berdesarkan inputan user. Sebagai contoh file yang ukurannya 50 Mb maka piecesnya berkisar antara 256 KB, 512 KB dan 1 MB, sedangkan ukuran minimum dan maksimum satu piece adalah 32 KB dan 4 MB. Dengan Begitu terbentuklah potongan file yang besarnya telah ditentukan.

Selanjutnya pieces- pieces yang ada sebelum dikirimkan kita akan menggunakan salah satu fungsi hash kriptografi

(dalam hal ini SHA-1 dan MD5) untuk digunakan mengenkripsi satu file yang telah dibagi perpieces- pieces seperti yang telah disebutkan diatas tersebut. Pieces-pieces yang sudah filenya sudah dienkripsi tersebut kemudian akan dikirimkan kepada peer lain yang membutuhkan beserta nilai hash dari file asli tersebut.

Selanjutnya setelah peer yang menerima file mendapatkan pieces- piecesnya secara lengkap dengan urutan yang sudah teratur maka peer tersebut akan mencocokan dengan hash yang telah dimilikinya yang didapat dari si pengirim dan mencocokan dengan hasil dari hash file yang baru diterima atau diunduh oleh peer tersebut.

IV. PENGUJIAN

Pada bagian ini penulis akan menguji dan membandingkan kedua algoritma yang dipakai oleh penguji untuk menghash data yang akan dikirim oleh aplikasi yang dibuat oleh penulis. Pengujian ini dilakukan dengan beberapa parameter yang akan dibahas dibawah ini dan dibandingkan hasil—hasil nya antara MD5 dan SHA-1.

A. Waktu

Dilakukan pengujian dengan cara menghitung lama waktu dari enkripsi yang dilakukan pada tiap file yang akan dikirim dan didapatkan hasil sebagai berikut:

Pengujian dilakukan pada sebuah laptop dengan spesifikasi processor intel core 2 duo (2 GHz, 800 MHz). Untuk file dengan besar 1 Mb.

MD5 = 0.014 second
SHA-1 = 0.026 second

Dapat dilihat disni penulis menghitung kecepatan pada awal fungsi hash tersebut mengenkripsi sampai setelah menghasilkan nilai hash dari file tersebut.

B. Peforma

Dilakukan pengujian dengan cara menghitung peforma dari aplikasi yang dibuat oleh penulis yakni nTorrent.adalah dengan cara mengukur tingkat kecepatan proses mulai dari melakukan hashing pada file yang akan dikirm sampai kepada validasi akan kebenaran dari file yang diterima dan didapatkan hasil sebagai berikut:

MD5 = 16.532 second
SHA-1 = 10.012.second

Pengujian dilakukan pada sebuah laptop dengan spesifikasi processor intel core 2 duo (2 GHz, 800 MHz). Untuk file dengan besar 50 Mb.

C. Keamanan

Kemanan yang dimaksud disini adalah keamanan dari segi kriptanalis yang mencoba untuk menemukan dua buah file atau data yang berbeda tetapi masih memiliki niali hash yang sama diantara keduanya pada pengujian untuk parameter keamanan ini penulis tidak dapat menemukan dua data yang berbeda dengan nilai hash yang sama baik pada data yang dienkripsi dengan algoritma MD5 maupun SHA-1.

V. ANALISIS

Pada bagian ini penulis akan mengkaji hasil yang telah didapat pada bagian sebelumnya yakni pada tahap pengujian dengan menggunakan dasar teori yang telah penulis utarakan pada bagian sebelumnya.

A. Waktu

Kecepatan. Kedua algoritma bekerja pada modulo 232 sehingga keduanya bekerja baik pada arsitektur 32 bit. SHA-1 mempunyai langkah lebih banyak dibandingkan MD5 (80 dibanding MD5 64) dan harus memproses 160 bit buffer dibanding DM5 128 bit buffer, sehingga SHA-1 bekerja lebih lambat dibanding MD5 pada perangkat keras yang sama. Jadi secara kecepatan untuk mengenkripsi datang yang dikirim kemampuan MD5 lebih cepat jika dibandingkan dengan SHA-1.

B. Peforma

Untuk peforma memang pada saat melakukan hashing pada file yang akan dikirim memang MD5 memilki kecepatan enkripsi yang lebih baik dari pada SHA-1. Namun jangan lupa karena MD5 memiliki format little endian sedangkan SHA-1 yang berupa big endian maka SHA-1 memiliki peforma yang lebih tinggi karena formatnya sudah sesuai dengan format standard penilaian pada jaringan yakni big endian oleh karena itu dari segi peforma pengirimannya maka SHA-1 memilki nilai yang lebih tinggi jika dibandingkan dengan MD5.

C. Keamanan

Untuk keamanan jika dibandingkan antara MD5 dan SHA-1 maka SHA-1 dinilai lebih tinggi jika dibandingkan dengan MD5. Keamanan terhadap serangan brute-force. Hal yang paling penting adalah bahwa SHA-1 menghasilkan diggest 32-bit lebih panjang dari MD5. Dengan brute-force maka SHA-1 lebih kuat dibanding MD5. Kelemahan MD5 ada pada design sehingga lebih mudah dilakukan kriptanalisis dibandingkan SHA-1

IV. KESIMPULAN

Pada makalah ini penulis membahas perbandingan algoritma fungsi hash kriptografi yang diimplementasikan pada salah satu aplikasi peer to peer yang dibuat oleh pengguna yakni nTorrent. Parameter yang dipakai untuk membandingkan kedua algoritma tersebut adalah:

- Waktu
  Waktu yang digunakan oleh MD5 terbukti lebih baik daripada SHA-1.
- Performansi
  Tingkat performansi yang ditunjukkan oleh algoritma SHA-1 lebih baik jika dibandingkan dengan MD5.
- Serangan

SHA-1 lebih aman jika dibandingkan dengan MD5.

Dengan ketiga parameter tersebut dapat ditarik kesimpulan bahwa algoritma fungsi hash kriptografi SHA-1 terbuktui lebih unggul jika diterapkan dalam aplikasi peer to peer nTorrent jika dibandingkan dengan MD5. Hal ini karena dari tiga parameter yang diukur SHA-1 unggul pada dua parameter diantaranya.

REFERENCES

[1] Munir, Rinaldi, Diktat Kuliah IF5054 Kriptografi, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, 2006.

[2] Andrew S. Tanenbaum, "Computer Networks", New Jersey, Prentice Hall, 2001.

[3] http://www.bittorrent.org/beps/bep_0003.html tanggal akses 4 mei 2011.

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, 9 Mei 2011

Nama dan NIM