Studi Algoritma dan Implementasi dari Teknik *Chaffing and Winnowing*

Donnie - NIM: 13505606
Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
E-mail: if15606@students.if.itb.ac.id

Abstraksi

Makalah ini membahas tentang metode *Chaffing and Winnowing* yang diajukan oleh Ronald L. Rivest pada tahun 1998. *Chaffing* berasal dari kata bahasa Inggris, yaitu *chaff*, yang berarti bagian dari gandum yang tidak dipakai, sehingga digunakan untuk makanan ternak. Sedangkan *Chaffing* sendiri menurut Ronald L. Rivest adalah proses penambahan pesan-pesan yang tidak berguna (*chaff*) ke dalam kumpulan pesan yang ingin dikirimkan. Sedangkan *Winnowing* berarti proses pemisahan *chaff* dari kumpulan pesan yang diterima. Kedua istilah ini berasal dari istilah yang digunakan dalam pertanian gandum.

Ide dasar dari metode ini dapat dideskripsikan sebagai berikut. Ada dua bagian untuk pengiriman pesan, yaitu *authenticating* dan *Chaffing*. Pada *authenticating*, pesan yang akan dikirimkan dibagi-bagi menjadi beberapa paket. Selanjutnya, setiap paket tersebut diberikan nomor urut yang menggambarkan urutan dari bagian pesan, dan MAC (*Message Authentication Code*) yang diperoleh dari hasil fungsi isi dari paket dan kunci otentifikasi. Sedangkan pada *Chaffing*, ditambahkan paket-paket palsu dengan MAC yang salah ke dalam kumpulan paket yang sudah ada sebelumnya. Untuk penerimaan pesan, hanya ada satu bagian, yaitu *Winnowing*. Pada *Winnowing*, sistem yang berada pada bagian penerima akan membuang paket-paket yang memiliki MAC yang salah dan menyusun paket-paket yang tersisa menjadi pesan yang sempurna.

Metode *Chaffing and Winnowing* ini dapat menjadi jalan keluar dari cengkraman peraturan yang dapat membatasi teknik enkripsi yang diperbolehkan untuk dipergunakan, karena sama sekali tidak menggunakan kunci enkripsi, melainkan hanya menggunakan pesan biasa yang diotentifikasi dengan MAC. Tetapi metode ini memiliki kelemahan dalam segi efisiensi karena penambahan chaff dengan cara yang salah dapat menambah jumlah paket total menjadi sangat banyak. Untuk mengatasi masalah ini, maka dikembangkan beberapa algoritma atau skema alternatif untuk meningkatkan efisiensi dari metode ini.

Kata Kunci: *Chaff, Chaffing, Winnowing, Message Authentication Code* (MAC)

Pendahuluan

Saat ini, kriptografi banyak sekali digunakan dalam proses komunikasi, terutama melalui internet. Banyak alasan yang digunakan manusia untuk menggunakan kriptografi, tetapi alasan yang paling utama yaitu untuk merahasiakan isi pesan dari pihak-pihak lain yang ingin mengetahui isi dari pesan yang dikirimkan tersebut. Tetapi karena tidak semua manusia di dunia ini menggunakan kriptografi untuk tujuan yang baik, maka timbullah kekhawatiran dari banyak pihak dari bahaya yang dapat ditimbulkan dari kriptografi ini. Kekhawatiran ini kemudian diwujudkan dalam bentuk pembuatan regulasi dan peraturan yang membatasi penggunaan kriptografi agar pesan-pesan yang beredar dapat selalu dikontrol untuk mencegah bahaya. Tetapi hal ini tentunya merupakan pelanggaran terhadap privasi yang dimiliki oleh setiap manusia, sehingga timbullah perlombaan antara pemerintah yang membuat regulasi, dengan para ahli kriptografi yang mengembangkan metode baru yang lolos dari regulasi tersebut.

Secara umum, terdapat dua teknik utama yang digunakan untuk menjaga kerahasiaan, yaitu Enkripsi dan Steganografi. Kedua metode ini telah digunakan begitu lama untuk menyembunyikan pesan rahasia dari pihak-pihak yang tidak berkepentingan. Tetapi sebagai usaha untuk menghindari jeratan hukum tersebut, kemudian seorang profesor

universitas MIT di bidang kriptografi akhirnya menemukan sebuah teknik baru yang jika tidak mungkin, sangat sulit untuk dijerat oleh hukum.

Pada tahun 1998, Ronald L. Rivest mengajukan ide baru yaitu *Chaffing and Winnowing* untuk menjaga kerahasiaan pesan tanpa menggunakan teknik enkripsi sama sekali. Metode ini dikembangkan sebagai tantangan bagi pemerintah Amerika Serikat yang membatasi penggunaan teknik enkripsi pesan untuk dapat menghindari bahaya terorisme yang mungkin menggunakan pesan yang terenkripsi untuk melakukan proses komunikasi. Metode C*haffing and Winnowing* ini tidak menggunakan teknik enkripsi dalam pengiriman pesan karena pesan yang dikirimkan adalah teks biasa, yang menggunakan kunci yang diketahui bersama oleh pengirim dan penerima untuk proses otentifikasi.

Informasi di atas merupakan latar belakang dari pemilihan topik *Chaffing and Winnowing* ini sebagai isi dari makalah yang akan dibuat.

Dasar Teori

1. Enkripsi

   Enkripsi adalah adalah proses mentransformasikan informasi sehingga tidak dapat dibaca tanpa pengetahuan khusus. Enkripsi telah digunakan untuk melindungi jalur komunikasi selama berabad-abad, tetapi hanya individu atau organisasi tertentu saja yang memerlukan kerahasiaan yang menggunakannya. Sejak pertengahan 1970an, enkripsi berkembang dari yang tadinya hanya digunakan untuk pemeritah atau organisasi tertentu menjadi milik masyarakat umum juga, dan sekarang ini banyak digunakan dalam sistem yang banyak digunakan, seperti internet, *e-commerce*, jaringan telepon seluler, dan ATM dari bank.

   Enkripsi atau pengkodean source code juga digunakan dalam *software copy protection* untuk melawan *reversed engineering*, analisis aplikasi tanpa otorisasi, *crack* dan pembajakan perangkat lunak.

2. Steganografi

   Steganografi adalah seni dan ilmu untuk menyembunyikan pesan sehingga tidak ada pihak lain selain pihak yang seharusnya mengetahui keberadaan dari pesan rahasia tersebut. Hal ini bertentangan dengan enkripsi, dimana keberadaan dari pesan tidak disembunyikan, tetapi isi dari pesan tersebutlah yang disembunyikan.

   Kata steganografi berasal dari bahasa Yunani yang berarti tulisan yang tersembunyi. Sejarah penggunaannya dapat ditelusuri hingga 440 SM. Herodotus menyebutkan dua contoh steganografi dalam bukunya, “The Histories of Herodotus”. Demeratus mengirimkan sebuah peringatan mengenai penyerangan yang akan terjadi terhadap Yunani dengan menuliskanya pada sebuah papan kayu dan melapisinya dengan lilin. Contoh lainnya yaitu Histiaeus yang mencukur kepala budak kepercayaannya dan mentatokan sebuah pesan di atasnya. Setelah rambutnya telah tumbuh kembali, pesan tersebut telah tersembunyi. Pesan tersebut merupakan pesan yang bertujuan untuk melakukan revolusi terhadap bangsa Persia.

   Secara umum, pesan yang disembunyikan menggunakan steganografi akan tampak dengan bentuk lain: sebuah gambar, atrikel, daftar belanjaan, atau bentuk lainnya. Misalnya, sebuah pesan dapat disembunyikan dengan menggunakan tinta yang tidak tampak di antara bagian yang kosong dari sebuah dokumen.

   Kelebihan steganografi dari enkripsi yaitu pesan yang disembunyikan tidak menarik perhatian kepada pihak lain. Sebuah kode yang tidak tersembuyi, bagaimanapun baiknya terenkripsi, akan menimbulkan kecurigaan dan pada kasus lainnya dapat dianggap ilegal dalam beberapa negara tertentu.

   Penggunaan steganografi dalam komunikasi elektronik yaitu termasuk juga pengkodean steganografi dalam transport layer, seperti file MP3 dalam protokol UDP.

   Sebuah pesan steganografi biasanya dienkripsi terlebih dahulu menggunakan cara tradidional, kemudian sebuah coverteks akan dimodifikasi untuk diisi oleh pesan yang telah dienkripsi tersebut, menghasilkan stegoteks. Sebagai contoh, ukuran huruf, spasi, jenis huruf, atau karakteristik lain dari coverteks dapat dimanipulasi untuk membaa pesan tersembunyi tersebut. Hanya pihak yang seharusnya menerima pesan tersebut saja yang dapat menemukan kembali pesan

tersebut dan selanjutnya mendekripsikan pesan tersebut.

3. Otentikasi
   Otentikasi berasal dari bahasa Yunani, yang berarti asli, dari kata *authentes* yang berarti penulis. Otentikasi yaitu sebuah kegiatan melakukan konfirmasi sesuatu atau seseorang adalah otentik. Melakukan otentikasi terhadap sebuah objek berarti mengkonfirmasikan asal-usulnya, sedangkan mengkonfirmasikan seseorang berarti melakukan verifikasi terhadap identitas dari orang tersebut.

   Dalam keamanan komputer, otentikasi adalah sebuah proses memverifikasikan pengirim dari sebuah komunikasi yang dilakukan, misalnya permintaan untuk melakukan *log in*. si pengirim dapat berupa seseorang yang menggunakan sebuah komputer, atau komputer itu sendiri, atau sebuah program aplikasi komputer.

   Dalam dunia web, otentikasi adalah sebuah cara untuk memastikan bahwa pengguna adalah benar-benar orang yang mereka katakan. Seorang pengguna yang ingin melakukan suatu fungsi dalam sebuah sistem adalah pengguna yang memiliki otorisasi untuk melakukan hal itu.

   Untuk membedakan istilah otentikasi dari otorisasi, diberikan singkatan A1 untuk otentikasi, dan A2 untuk otorisasi. Istilah AuthN dan AuthZ juga digunakan untuk membedakan kedua istilah ini dalam beberapa komunitas tertentu.

   Masalahnya yaitu otorisasi seringkali dipikir identik dengan otentikasi; banyak sekali protokol standar keamanan, peraturan dan lain-lain berasumsi seperti itu.tetapi untuk lebih tepatnya, otoentikasi adalah melakukan verifikasi identitas seseorang, sedangkan otorisasi adalah melakukan verifikasi bahwa seseorang memiliki otoritas untuk melakukan sebuah operasi tertentu. Contohnya yaitu saat kita menunjukkan identitas kita kepada teller dari bank, maka kita akan diotentikasi oleh teller tersebut dan diotorisasi untuk mengakses informasi dari account bank milik kita. Kita tidak dapat mengakses sebuah account yang bukan milik kita sendiri.

   Karena otorisasi tidak dapat terjadi tanpa otentikasi, maka otorisasi terkadang digunakan untuk kombinasi dari otentikasi dan otorisasi.

4. *Message Authentication Code*
   Message Authentication Code (MAC) adalah sebuah bagian informasi pendek yang digunakan untuk melakukan otentikasi terhadap sebuah pesan. Sebuah algoritma MAC menerima masukan berupa sebuah kunci otentikasi rahasia dan pesan yang akan diotentikasi, dan menghasilkan keluaran MAC. Sebuah nilai MAC melindungi integritas dari sebuah pesan dan juga otentitas dari pesan tersebut, dimana pihak yang menerima pesan tersebut yang juga memiliki kunci otentikasi untuk mendeteksi perubahan yang terjadi pada isi dari pesan tersebut.

   Walaupun fungsi MAC mirip dengan fungsi *hash* dari kriptografi, tetapi keduanya memiliki kebutuhan keamanan yang berbeda. Agar dapat dianggap aman, maka sebuah fungsi MAC harus dapat bertahan dari pemalsuan melalui *chosen-plaintext attack*. Hal ini berarti bahwa pihak lawan tidak dapat menemukan dua pesan M dan M’ yang akan menghasilkan nilai MAC yang sama, walaupun pihak lawan memiliki sebuah akses terhadap basis data dari berbagai pesan yang ada untuk menghasilkan nilai MAC.

   MAC juga berbeda dengan tandatangan digital, dimana nilai MAC dihasilkan dan diverifikasi menggunakan kunci otentikasi yang sama. Hal ini berarti bahwa pengirim dan penerima sebuah pesan harus menyetujui sebuah kunci otentikasi sebelum memulai komunikasi. Untuk alasan yang sama, MAC tidak memberikan aspek nirpenyangkalan (*non-repudiation*) yang diberikan oleh tandatangan digital. Siapapun yang dapat melakukan verifikasi sebuah MAC juga dapat membangkitkan MAC untuk pesan lainnya.

   Algoritma MAC dapat dibangun dari algoritma kriptografi primitif lainnya, seperti fungsi h*ash* (HMAC), dan algoritma *block cipher* (OMAC, CBC-MAC, dan PMAC).

5. *Digital Rights Management*
   Digital Rigths Management (DRM) adalah sebuah istilah yang mereferensi kepada teknologi yang digunakan oleh para

publisher atau pemilik *copyright* untuk mengontrol akses dan penggunaan dari data digital dan perangkat keras. Istilah ini biasa disalahartikan dengan *copy protection* dan *technical protection measures*; dimana kedua istilah tersebut mereferensi kepada teknologi yang mengatur penggunaan dan akses data dari alat elektronik, dan merupakan bagian dari perancangan DRM.

DRM merupakan sebuah topik yang kontroversial. Para ahli hukum berargumentasi bahwa DRM diperlukan bagi pemilik *copyright* untuk mencegah penduplikasian produk mereka untuk memastikan aliran keuntungan yang berkelanjutan.beberapa kritikus dalam dunia teknologi, termasuk Free Software Foundation, mengomentari bahwa penggunaan kata "*Rights*" adalah menyesatkan dan akan memberikan sugesti kepada masyarakat. Mereka menyarankan untuk menggunakan istilah *Digital Restrictions Management.*

Enterprise Digital Rights Management (E-DRM atau ERM) adalah penggunaan teknologi DRM untuk melakukan kontrol terhadap akses ke dokumen milik suatu korporat (file-file Microsoft Word, PDF, TIFF, AutoCAD), daripada media yang dapat dijalankan oleh konsumen. Teknologi ini biasanya membutuhkan sebuah Policy Server untuk melakukan otentikasi hak dari pengguna, tetapi perangkat lunak yang digunakan akhir-akhir ini tidak lagi memerlukannya. EDRM secara umum dimaksudkan untuk melakukan pertukaran rahasia, yang sangat berbeda dengan barang yang memiliki *copyright*, yang musuh utamanya adalah mata-mata industi ataupun korporat.

Berbagai Masalah Seputar Hukum Enkripsi

Masalah mengenai hukum seputar enkripsi telah menjadi sebuah masalah pelik di berbagai negara di seluruh dunia. Di satu pihak masyarakat menginginkan kebebasan informasi dan perlindungan atas privasi mereka, sementara pihak pemerintah dengan dalih keamanan berusaha untuk membatasi kebebasan dan privasi yang dimiliki masyarakat tersebut.

Di Inggris, terdapat peraturan mengenai intersepsi komunikasi data dan metode untuk meminta kunci enkripsi jika diminta sebagai bukti di pengadilan [6]. Di Amerika Serikat juga ada hukum yang membatasi penggunaan algoritma enkripsi, bahkan banyak kecurigaan bahwa beberapa algoritma yang pernah didesain memiliki *backdoor* yang digunakan badan pemerintah untuk mengetahui isi dari pesan yang dienkripsi. Sedangkan pada beberapa negara lainnya menerapkan larangan ekspor perangkat lunak atau algoritma enkripsi, ada juga yang mengharuskan kepemilikan lisensi untuk penggunaan perangkat lunak enkripsi, bahkan ada juga yang melarang rakyatnya sama sekali dari mengenkripsi komunikasi internet mereka.

Amerika Serikat memperlakukan enkripsi seperti senjata. Mereka ingin memastikan bahwa kekuatan asing dan kelompok teroris menggunakan enkripsi dengan tingkatan yang mereka bisa pecahkan, untuk memastikan keamanan nasional. Sistem enkripsi yang dapat diekspor hanyalah yang memiliki kunci yang sama atau lebih kecil dari 56 bit dan sistem yang memungkinkan untuk terjadinya *key recovery*. Rivest beranggapan bahwa metode enkripsi tidak seharusnya dibatasi seperti itu, karena suatu saat, seseorang pasti akan menemukan jalan untuk menghindari jeratan hukum tersebut.

Ada beberapa kasus menarik yang dapat dianggap terlalu ekstrim dalam dunia kriptografi, yaitu kasus Bernstein vs. United States dan kasus Junger vs. Daley.

Pada kasus Bernstein vs. United States, Daniel J. Bernstein yang merupakan pelajar di Berkeley University memprotes kebijakan peraturan Amerika Serikat mengenai larangan untuk mempublikasikan makalahnya beserta source code mengenai dikembangkannya. Selanjutnya, ia mengajukan tuntutan ke pengadilan untuk dapat mempublikasikan hasil kerjanya tersebut. Empat tahun kemudian, akhirnya ia memenangkan kasus tersebut [5].

Sedangkan pada kasus Junger vs. Daley, Junger yang merupakan seorang profesor di Case Western Reserve University ingin mengajarkan kuliah computer law. Tetapi dikarenakan adanya restriksi mengenai ekspor dari perangkat lunak enkripsi, maka ia tidak dapat menerima murid yang non-US ke dalam kelasnya [7].

Kedua kasus tersebut menandakan bahwa hukum yang diterapkan pada kriptografi sudah berlebihan sampai pada tahap tidak masuk akal. Tetapi, ada juga dampak positif dari penerapan hukum mengenai enkripsi ini, yaitu para ahli

kriptografi saling berlomba-lomba untuk merancang metode dan algoritma baru yang dapat lolos dari jeratan hukum dengan memanfaatkan celah-celah hukum yang ada.

Chaffing & Winnowing sebagai Teknik Baru untuk Menjaga Kerahasiaan selain Enkripsi dan Steganografi

Metode Chaffing & Winnowing merupakan metode baru yang dikembangkan oleh Ronald L. Rivest pada tahun 1998 sebagai jawaban dari hukum enkripsi yang berlaku sekarang ini. Metode ini oleh perancangnya tidak digolongkan ke dalam teknik enkripsi maupun steganografi. Hal ini disebabkan oleh adanya beberapa karakteristik khusus dari *Chaffing* & *Winnowing* yang tidak sesuai dengan kedua teknik lainnya.

Definisi umum dari Enkripsi yang diberikan oleh Rivest [9] adalah sebagai berikut:

"Mentransformasikan pesan (plainteks) menjadi cipherteks dengan menggunakan kunci sehingga pihak lawan tidak dapat mengetahui pesan aslinya."

Kunci pada definisi di atas dapat berupa kunci simetri maupun kunci publik/asimetri.

Sedangkan definisi umum dari Steganografi yang juga diberikan oleh Rivest [9] adalah sebagai berikut:

"Menyembunyikan sebuah pesan rahasia dalam sesuatu yang lebih besar sehingga pihak lawan tidak dapat mengetahui keberadaan maupun isi dari pesan rahasia tersebut"

Pertama, jika dibandingkan dengan teknik enkripsi, maka teknik *Chaffing & Winnowing* tidak menggunakan kunci enkripsi dalam prosesnya, melainkan hanya menggunakan kunci otentikasi saja. Tetapi bisa saja terdapat argumen bahwa teknik ini masih dapat digolongkan ke dalam teknik enkripsi karena msih menggunakan kunci. Tetapi pernyataan ini dapat dibantah dengan kenyataan bahwa paket pesan dapat dibaca dengan jelas walaupun tanpa menggunakan kunci otentikasi.

Kedua, jika dibandingkan dengan teknik steganografi, maka teknik *Chaffing & Winnowing* sama sekali tidak menyamarkan isi pesan yang sebenarnya dengan memasukkannya ke dalam media tertentu lainnya, seperti gambar atau suara. Memang ada sedikit kemiripan dalam prinsip, dimana pihak lawan tidak dapat mengetahui isi pesan asli karena tersimpan dalam sekumpulan pesan sampah. Perbedaannya terletak yaitu dimana pihak lawan mengetahui keberadaan pesan rahasia tersebut, tetapi tidak dapat membedakannya dengan pesan-pesan sampah yang ikut bersamanya. Sedangkan pada steganografi yang diinginkan adalah pihak lawan sama sekali tidak mengetahui keberadaan dari pesan rahasia tersebut.

Walaupun sampai saat ini belum ada metode implementasi yang resmi dari teknik ini, tetapi maksud sebenarnya dari Ronald L. Rivest yaitu menstimulasikan sebuah debat yang akan mejernihkan berbagai masalah yang sebelumnya muncul dalam diskusi masalah pembentukan peraturan.

Definisi *Chaffing & Winnowing*

Pertama, dimulai dari definisi *Chaffing & Winnowing* itu sendiri. *Chaffing* berasal dari kata dalam bahasa Inggris yaitu *Chaff*, yang berarti bagian dari gandum yang tidak berguna. Istilah *Chaff* digunakan untuk mengacu kepada paket pesan yang tidak benar, atau yang sengaja ditambahkan untuk menyesatkan pihak-pihak yang ingin melakukan intersepsi pesan. Sedangkan istilah *Wheat* digunakan untuk mengacu kepada paket pesan yang asli. Istilah *Wheat* ini digunakan untuk menjaga konsistensi dengan istilah-istilah lainnya.

Selanjutnya, istilah *Winnowing* juga berasal dari bahasa Inggris, yaitu *Winnow*, yaitu proses memisahkan bagian-bagian yang tidak berguna atau bagian yang jelek. Dalam pertanian, istilah Winnow digunakan untuk proses memisahkan gandum (*Wheat*) dari bagian-bagian yang tidak diperlukan (*Chaff*).

Prinsip Teknik *Chaffing dan Winnowing*

Untuk membahas tentang langkah-langkah yang dilakukan dalam teknik *Chaffing & Winnowing* ini, maka akan dibagi menjadi dua bagian, yaitu pada saat pengiriman pesan dan pada saat penerimaan pesan.

Pada saat pengiriman pesan, maka terdapat dua langkah yang dilakukan, pertama melakukan proses otentikasi, yaitu dengan menambahkan MAC pada paket pesan; langkah selanjutnya yaitu menambahkan *Chaff*. Proses penambahan *Chaff* dapat dilakukan dengan bervariasi, yang akan dibahas pada bagian lain dari makalah ini.

Pada proses otentikasi, yang dilakukan adalah memecah pesan asli ke dalam paket-paket pesan, kemudian menambahkan MAC pada

setiap paket pesan asli tersebut yang akan dikirimkan dengan menggunakan kunci otentikasi yang diketahui bersama baik oleh pihak pengirim maupun penerima.

Packet => Packet, MAC

Perlu ditegaskan sekali lagi bahwa kunci yang digunakan bukanlah pasangan kunci enkripsi/dekripsi, tetapi yaitu kunci otentikasi, yang biasa digunakan untuk melakukan otentikasi sumber dan isi dari paket pesan yang dikirimkan.

Pada proses selanjutnya, yaitu memberikan nomor serial dari paket-paket pesan asli sesuai dengan nomor urut paket dari pesan aslinya. Pemberian nomor serial ini dimaksudkan agar pihak penerima tidak kesulitan dalam menyusun pesan asli yang telah diterimanya, karena pada jaringan bisa saja terdapat paket-paket yang hilang, sehingga urutan dari paket asli dapat kacau dan sulit untuk mengetahui isi asli dari pesan karena urutannya yang sudah tidak teratur lagi.

Packet, MAC => Sequence, Packet, MAC

Langkah terakhir dalam pengiriman pesan yaitu dengan menambahkan beberapa paket Chaff sesuai dengan nomor serial dari masing-masing paket yang telah terbentuk. Hal ini jelas digunakan untuk menyamarkan paket pesan asli di antara beberapa paket Chaff yang sengaja kita buat sendiri.

Untuk lebih jelasnya, maka kita akan menggunakan sebuah contoh untuk mengilustrasikan proses pengiriman yang sudah dijelaskan di atas.

Misalkan Alice ingin mengirimkan sebuah pesan rahasia kepada Bob dengan menggunakan teknik *Chaffing & Winnowing* ini. Pesan yang akan dikirimkan yaitu:

"Hi Bob meet me at 7PM Love-Alice"

Langkah pertama yang Alice lakukan yaitu memecah pesan tersebut ke dalam beberapa bagian yang kita sebut sebagai paket *Wheat*, dan pesan tersebut akan menjadi seperti berikut ini:

(Hi Bob)
(Meet me at)
(7PM)
(Love-Alice)

Selanjutnya paket-paket *Wheat* tersebut akan dihitung nilai MACnya dengan menggunakan kunci otentikasi yang sudah disepakati sebelumnya oleh Alice dan Bob. Nilai-nilai MAC tersebut kemudian akan ditambahkan pada paket *Wheat* yang bersangkutan.

(Hi Bob,465231)
(Meet me at,782290)
(7PM,344287)
(Love-Alice,312265)

Kemudian untuk menjaga urutan dari pesan tidak terganggu, maka untuk setiap paket pesan *Wheat* diberikan nomor serial seperti di bawah ini:

(1,Hi Bob,465231)
(2,Meet me at,782290)
(3,7PM,344287)
(4,Love-Alice,312265)

Langkah terakhir dari pengiriman pesan yaitu dengan menambahkan paket-paket *Chaff* sesuai dengan nomor serial dari paket *Wheat* yang sudah ada sebelumnya.

(1,Hi Larry,532105)
(1,Hi Bob,465231)
(2,Meet me at,782290)
(2,I'll call you at,793122)
(3,6PM,891231)
(3,7PM,344287)
(4,Yours-Susan,553419)
(4,Love-Alice,312265)

Nilai MAC yang dimiliki oleh paket *Chaff* bukanlah nilai MAC yang diperoleh dari hasil perhitungan dengan menggunakan kunci otentikasi yang digunakan sebelumnya, tetapi merupakan nilai MAC yang diperoleh dengan menggunakan kunci otentikasi yang berbeda, atau merupakan sebuah nilai acak sama sekali.

Setelah selesai, maka seluruh paket *Wheat* dan *Chaff* tersebut dikirimkan melalui jaringan kepada pihak penerima, yaitu Bob. Selama proses pengiriman ini mungkin saja ada pihak lain (Charlie) yang bermaksud untuk melakukan intersepsi pesan. Tetapi Charlie mengalami kesulitan untuk menangkap isi pesan aslinya, dikarenakan paket pesan yang diterimanya dapat diartikan lebih dari satu makna. Charlie tidak memiliki kunci otentikasi yang diperlukan untuk mengetahui mana paket pesan *Wheat* dengan *Chaff*.

Kemudian saat paket pesan telah diterima oleh Bob, maka ia akan melakukan proses *Winnowing*, yaitu dengan melakukan otentikasi

setiap paket dengan kunci otentikasi yang dimilikinya untuk memisahkan antara paket-paket *Wheat* dengan paket-paket *Chaff*.

(1,Hi Larry,532105) => **Invalid**
(1,Hi Bob,465231) => Valid
(2,Meet me at,782290) => Valid
(2,I'll call you at,793122) => **Invalid**
(3,6PM,891231) => **Invalid**
(3,7PM,344287) => Valid
(4,Yours-Susan,553419) => **Invalid**
(4,Love-Alice,312265) => Valid

Selanjutnya, dengan menyatukan kembali bagian-bagian yang valid dan menghilangkan nomor serial dan nilai MAC, maka akan dihasilkan kembali pesan asli yang utuh.

"Hi Bob meet me at 7PM Love-Alice"

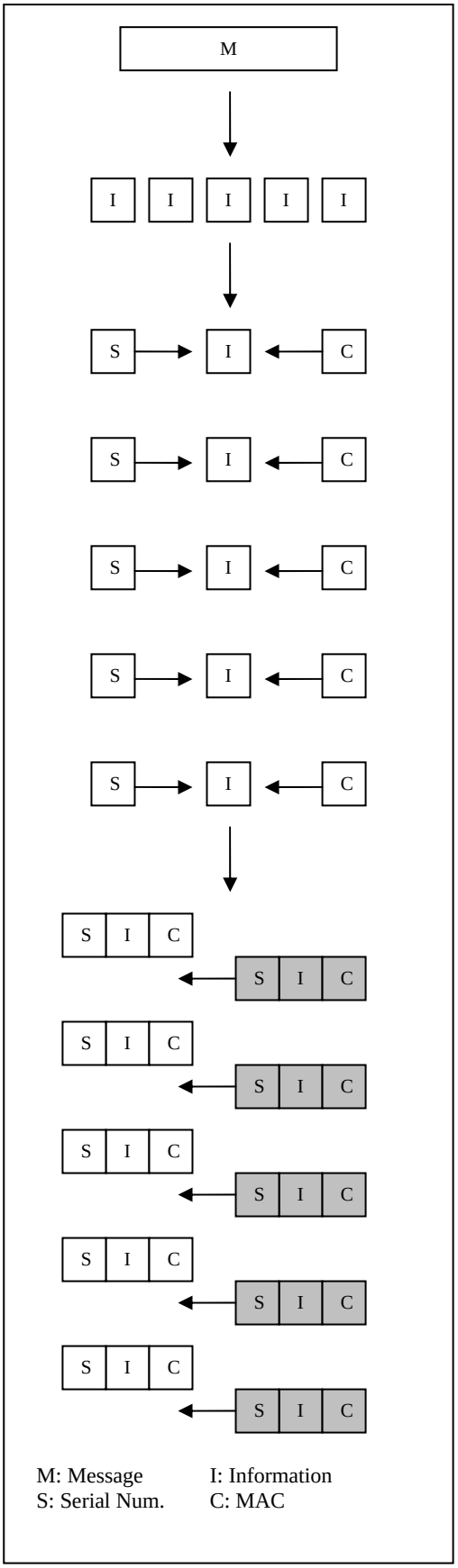
Seandainya Charlie ingin memalsukan pesan yang dikirimkan tersebut, maka karena ia tidak mengetahui kunci otentikasi yang digunakan oleh Alice dan Bob, maka saat proses *Winnowing* paket-paket palsu yang dibuat oleh Charlie tersebut akan dipisahkan dari paket *Wheat*. Maka dapat disimpulkan bahwa teknik Chaffing & Winnowing ini kebal terhadap *Man-in the middle-Attack*.

Faktor yang Menentukan Tingkat Kerahasiaan dari Teknik *Chaffing & Winnowing*

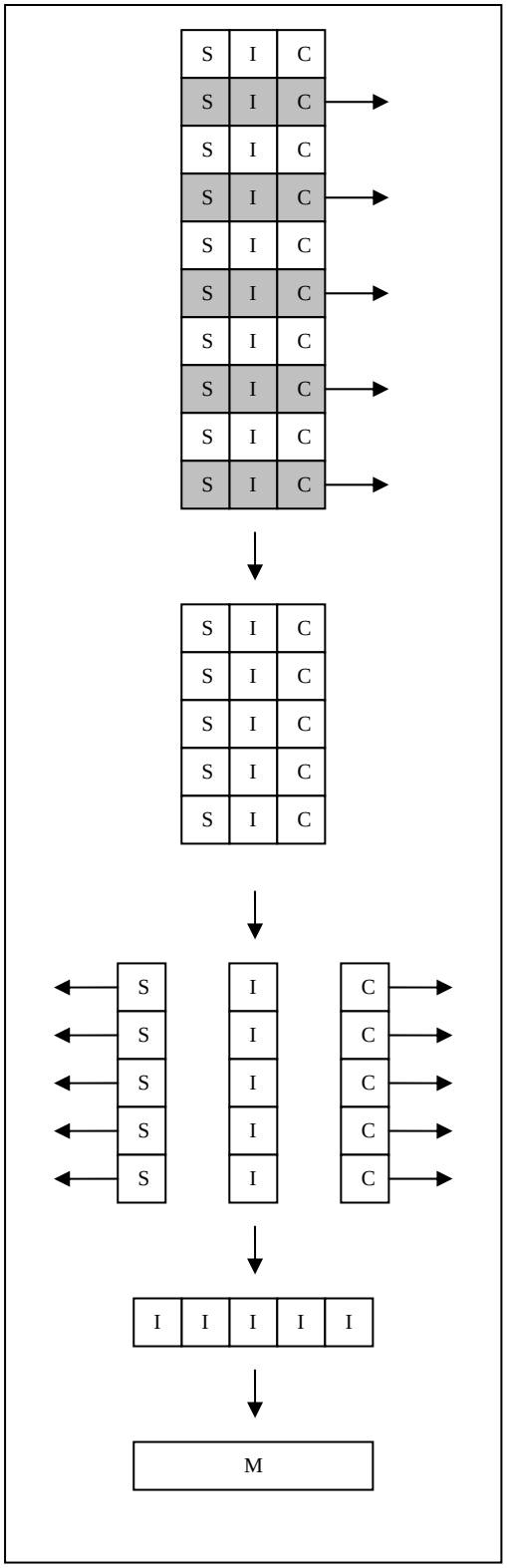
1. Penggunaan algoritma MAC yang digunakan. Algortima yang digunakan haruslah merupakan algoritma yang telah terbukti paling aman dan sedikit sekali menghasilkan nilai MAC yang duplikat.

2. Pembagian pesan menjadi paket-paket. Sebaik apapun teknik *Chaffing & Winnowing* yang digunakan, jika pembagian pesan menjadi paket dilakukan dengan sembrono, maka pihak lawan dapat dengan mudah mengetahui isi dari pesan asli yang dikirimkan.

3. Prosedur penambahan paket *Chaff* yang digunakan. Penambahan paket *Chaff* sebaiknya memiliki format yang sejenis dengan paket *Wheat*, memiliki nomor serial yang masuk akal, isi pesan yang rasional, dan memiliki MAC yang invalid terhadap kunci otentikasi yang digunakan pada paket *Wheat*.



Gambar 1. Ilustrasi Proses *Chaffing*



Gambar 2. Ilustrasi Proses *Winnowing*

Variasi Teknik *Chaffing & Winnowing*

Selain menggunakan metode yang diusulkan oleh Ronald Rivest, beberapa variasi juga dapat dikembangkan untuk memperkuat teknik ini dari berbagai serangan. Tetapi secara umum, teknik *Chaffing & Winnowing* yang baik dapat dicapai dengan merancang algoritma MAC yang baik.

Beberapa variasi yang sudah ditawarkan yaitu antara lain:

1. Untuk membuat isi paket pesan lebih sulit untuk dibaca, maka dapat dilakukan pengubahan pesan ke dalam format biner ataupun format lainnya, misalkan heksadesimal. Tambahan lainnya misalkan pesan yang formatnya sudah diubah tersebut dibagi lagi menjadi beberapa bagian. Misalnya untuk format biner, setiap satu bit pesan dijadikan satu paket pesan *Wheat*. Perhatikan contoh paket pesan berikut:

   (1,0,351216)
   (1,1,895634)
   (2,0,452412)
   (2,1,534981)
   (3,0,639723)
   (3,1,905344)
   (4,0,321329)
   (4,1,978823)

   Charlie akan mengalami kesulitan yang jauh lebih besar dalam mengartikan pesan yang diintersepsi olehnya karena pesan tersebut sama sekali tidak dapat dibaca tanpa sebelumnya digabungkan sampai utuh.

   Untuk perhitungan efisiensi, maka kita lihat sebuah contoh sebagai berikut:

   Misalkan Alice mengirimkan sebuah pesan sepanjang 1 MB. Pesan sepanjang 1 MB kira-kira akan menghasilkan sebanyak 8 juta bit pesan. Jika sebuah paket memiliki panjang 100 bit, maka ada 800 juta bit atau sebesar 100 MB yang akan dikirimkan. 100 MB paket yang dikirimkan untuk pesan asli sebesar 1 MB tentunya sudah cukup boros, dan hal ini masih belum memperhitungkan penambahan paket *Chaff* ke dalamnya. Jika diasumsikan dilakukan penambahan satu paket *Chaff* untuk setiap paket *Wheat*, maka akan ada penambahan sebesar 100 MB. Jika lebih dari satu paket *Chaff* yang ditambahkan untuk setiap paket *Wheat*, maka tinggal dikalikan saja jumlah paket *Chaff* yang akan ditambahkan untuk setiap paket *Wheat* dengan 100 MB. Jumlah ini tentunya sama sekali tidak efisien untuk diimplementasikan dalam kenyataan.

2. Menggabungkan beberapa pesan dari pengirim dan penerima yang berbeda-beda,

kemudian mengirimkannya secara bersamaan.

Dengan melakukan hal ini, maka kesulitan yang dialami oleh Charlie adalah membedakan paket-paket mana yang merupakan *Wheat* dan paket-paket mana saja yang merupakan *Chaff*. Hal ini juga cukup sulit dilakukan, karena *Wheat* bagi satu pihak merupakan Chaff bagi pihak lainnya. Hal ini dikarenakan masing-masing pihak pengirim menggunakan kunci otentikasi yang berbeda-beda, maka hanya masing-masing pihak yang dituju saja yang mampu menerima pesan yang sebenarnya. Paket-paket yang tidak ditujukan untuk satu pihak akan dianggap sebagai paket *Chaff* dan dihilangkan, sehingga ia hanya menerima paket yang dimaksudkan untuknya saja.

Di pihak lain, Charlie sebagai interseptor tidak dapat membedakan paket pesan mana yang dikirimkan oleh pihak mana yang ditujukan untuk pihak mana pula. Pembahasan metode ini lebih lanjut akan dibahas pada bagian lainnya.

3. Menggunakan metode "*All-or-Nothing Transform*". Metode ini merupakan pengembangan dari metode pertama di atas, tetapi dengan perbaikan efisiensi. Pada metode ini, seluruh pesan diubah menjadi sebuah paket, dimana paket tersebut hanya bisa dibuka jika pihak penerima memiliki keseluruhan paket. Jika pihak penerima tidak memiliki keseluruhan paket, maka ia tidak dapat mengetahui isi dari paket tersebut.

   Metode ini bekerja dengan cara sebagai berikut:
   Misalkan m1, m2, …, ms adalah blok plainteks, H adalah fungsi *hash*, dan K' adalah kunci acak.

   Blok yang dikirimkan adalah:

   $$m_i^{'} = m_i \oplus H(K^{'}, i)$$

   untuk i = 1, 2, …, s

   K' dikirimkan dengan mengirimkan nilai tambahan M:

   $$M = K^{'} \oplus h_1 \oplus h_2 \oplus ... \oplus h_s$$

   dimana:

   $$h_i = H(K_0, m_i' \oplus i)$$

   untuk i = 1, 2, …, s dan K_0 adalah kunci yang diketahui secara publik.

   Misalkan Alice memaketkan pesannya yang akan dikirimkan kepada Bob. Selanjutnya Alice memecah paket tersebut menjadi blok sebesar 1024 bit, mengotentikasikan setiap blok dengan MAC, dan mengirimkan hasilnya ke Bob. Cara ini akan lebih efisien dibandingkan dengan variasi pertama, dimana setiap paket hanya berisi satu bit pesan.

   Untuk perhitungan efisiensi, maka kita lihat sebuah contoh sebagai berikut:

   Misalkan Alice ingin mengirimkan sebuah pesan sepanjang 1 MB. Alice kemudian membungkus pesannya tersebut dan membaginya menjadi 8.192 paket dengan panjang masing-masing paket sebesar 1024 bit. Misalkan panjang bit untuk nomor serial adalah 32 bit, dan untuk MAC adalah 64 bit. Maka 32 bit + 1024 bit + 64 bit = 1.100 bit untuk setiap paket. Untuk seluruh paket yang ada, maka besar totalnya adalah kira-kira 1,1 MB. Jumlah ini tentunya jauh lebih kecil dibandingkan dengan variasi pertama tadi. Walaupun mengikutsertakan *Chaff*, maka jika ditambahkan satu paket *Chaff* untuk setiap paket *Wheat*, maka jumlah totalnya hanya akan menjadi 2,2 MB. Variasi ini tentunya lebih efisien dan lebih memungkinksn untuk diimplementasikan dibandingkan dengan variasi dimana memerlukan 100 MB paket *Chaff* untuk setiap 1 MB pesan yang ada.

   Tetapi, dengan menggunakan metode *All-or-Nothing Transform* ini sebetulnya tidak diperlukan jumlah paket C*haff* yang terlalu banyak, karena jika pihak penerima tidak memiliki seluruh paket yang benar, maka ia tidak akan dapat mengetahui isi dari pesan aslinya. Semakin panjang pesan yang dikirmkan, maka jumlah paket chaff yang diperlukan akan semakin sedikit. Dari 8.192 paket *wheat*, jika ditambahkan 8 paket *chaff* saja akan menghasilkan total paket sebanyak 8.200 paket. Untuk memilih 8.192 paket yang benar dari total 8.200 paket yang ada, maka terdapat kombinasi sebanyak $5{,}0525 * 10^{26}$. Jumlah paket ini tidak akan menghasilkan besar bandwith yang terlalu berbeda jauh dengan 1,1 MB sebelumnya, tetapi tingkat kesulitan untuk memecahkannya tanpa memiliki kunci otentikasi sungguh besar.

Celah Hukum yang Dimanfaatkan dalam Teknik *Chaffing & Winnowing*

Setelah melihat berbagai variasi dari teknik *Chaffing & Winnowing*, sekarang perlu dilihat bagaimana teknik ini dapat menghindari celah hukum yang ada pada saat ini. Beberapa aspek dari celah hukum tersebut akan dijelaskan sebagai berikut ini:

1. Resiko Keamanan
   Walaupun dengan hukum yang berlaku saat ini pemerintah dapat meminta kunci enkripsi dengan paksa, tetapi tidak sama halnya dengan kunci otentikasi. Tetapi hal tersebut adalah hal yang tidak mungkin dilakukan oleh pihak pemerintah. Jika kunci otentikasi dapat diminta secara hukum, maka hal tersebut akan memungkinkan pemerintah dapat membuat paket pesan otentik yang dipalsukan untuk pihak mana saja yang sedang melakukan komunikasi.

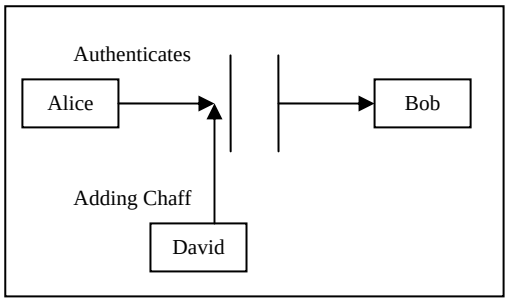
   Hal ini tentunya tidak diinginkan, karena akan menciptakan kekacauan dari struktur dan integritas dari internet. Hal ini kemudian dapat berlanjut pada *hacker* untuk menyadap seluruh pesan pribadi, bahkan untuk mengontrol komputer. Misalnya untuk mematikan sistem komputer pembangkit listrik ataupun sistem kendali lalu lintas udara. Hal ini tentunya sama sekali tidak diinginkan untuk terjadi oleh pemerintah.

   Hal ini disebabkan karena kekuatan untuk melakukan otentikasi sama saja dengan kekuatan untuk mengendalikan, dan menangani seluruh kekuatan otentikasi bagi pemerintah adalah suatu hal yang tidak masuk akal, walaupun didasarkan dengan alasan keamanan. Tidak ada satu pihakpun yang dapat menerima resiko atas keamanan yang sangat tinggi jika hal itu sampai terjadi.

2. Nir-penyangkalan (*Deniable Repudiation*)
   Salah satu aspek dari kriptografi yaitu menyangkut masalah Nir-penyangkalan (*Deniable Repudiation*), yaitu menyangkal telah melakukan pengiriman suatu pesan yang dituduhkan kepada satu pihak.

   Jika Alice dan Bob menggunakan variasi teknik *Chaffing & Winnowing* yang kedua, maka jika mereka dituntut oleh pihak pemerintah untuk memberikan kunci otentikasi yang mereka miliki, maka mereka dapat memberikan kunci otentikasi palsu kepada pemerintah agar mereka dapat tetap menyembunyikan pesan yang sebenarnya dalam tumpukan pesan *Chaff* yang ada.

   Dalam skenario lainnya, mungkin saja Alice dan Bob hanya melakukan proses otentikasi ke pesan yang mereka kirimkan, sedangkan ada pihak lain (David) yang menambahkan paket-paket *Chaff* yang berasal dari pihak lainnya karena misalkan saja Alice dan beberapa orang lainnya terletak dalam sebuah gedung yang sama, dan memiliki hanya satu jalur komunikasi keluar masuk. Dalam kasus ini, Alice dan Bob saja sama sekali tidak melakukan teknik *Chaffing dan Winnowing*, tetapi bersama dengan David barulah teknik ini dilakukan. Tetapi karena tidak ada unsur kesengajaan sama sekali, maka mereka sama sekali tidak dapat dituduh melakukan proses enkripsi ataupun C*haffing & Winnowing*.



Gambar 3. Ilustrasi Skenario Teknik *Chaffing & Winnowing* tanpa disengaja

Kelebihan dan Kekurangan Metode *Chaffing & Winnowing*

Kelebihan:

1. Tingkat keamanan yang tinggi. Pihak lawan yang ingin berusaha melakukan intersepsi terhadap pesan akan mengalami kesulitan besar dalam memahami isi keseluruhan pesan yang dikirmkan.

2. Kebal terhadap banyak serangan yang biasa dilakukan oleh pihak yang tidak berhak, misalnya *Man-in-the-middle Attack*, *Chosen-Plaintext Attack*, dan jenis-jenis serangan lainnya.

3. Dapat menerapkan prinsip nirpenyangkalan (*deniable repudiation*), dimana kedua pihak dapat memberikan kunci otentikasi palsu jika dipaksa oleh pemerintah untuk menyembunyikan isi pesan yang sebenarnya.

4. Tidak dapat disentuh oleh tangan hukum, karena proses otentikasi adalah proses

yang sangat penting dalam dunia komunikasi, dimana siapapun tidak boleh memiliki kuasa penuh terhadapnya.

Kekurangan:

1. *Overhead* yang cukup besar jika tidak menggunakan metode "*All-or-Nothing Transform*".

2. Belum adanya implementasi produk nyata untuk teknik ini.

Usul Implementasi Metode *Chaffing & Winnowing*

Dalam makalah ini, dibuat sebuah aplikasi simulai untuk memperlihatkan bagaimana teknik *Chaffing & Winnowing* ini bekerja. Beberapa hal yang dijadikan pertimbangan dalam pembuatan aplikasi simulasi ini adalah sebagai berikut:

1. Pesan yang dimasukkan ke dalam setiap paket adalah satu kata dari pesan asli yang ada.

2. Paket *Chaff* yang ditambahkan merupakan pesan lain sudah ada sebelumnya, bukan berupa kata-kata yang dibangkitkan secara acak. Pertimbangan hal ini yaitu agar dapat melakukan nirpenyangkalan (D*eniable Repudiation*). Setiap pesan *Chaff* tersebut juga memiliki kunci otentikasinya masing-masing. Jumlah pesan C*haff* yang digunakan adalah sebayak dua buah pesan.

3. Algoritma MAC yang digunakan adalah HMACMD5 yang sudah terintegrasi dengan Microsoft Visual C# 2005 Express Edition, yang juga digunakan sebagai lingkungan pengembangan. Implementasi algoritma HMACMD5 dapat dilihat pada Algoritma 3.

Algoritma yang Diimplementasikan

1. Fungsi *Chaffing*
   Fungsi *Chaffing* yang diimplementasikan menggunakan dua buah pesan tambahan yang dimasukkan secara manual oleh pengguna. Kedua pesan tambahan tersebut juga dapat ditambahkan kunci otentikasi masing-masing. Implementasi fungsi *Chaffing* pada aplikasi ini dapat dilihat pada Algoritma 1.

2. Fungsi *Winnowing*
   Fungsi *Winnowing* yang diimplementasikan menggunakan pesan yang telah melalui proses *Chaffing* sebelumnya, dengan memasukkan kunci otentikasi yang tepat, maka pesan yang asli dapat didapatkan. Implementasi fungsi *Winnowing* pada aplikasi ini dapat dilihat pada Algoritma 2.

Uji Aplikasi Simulator

Untuk pengujian, maka digunakan tiga buah pesan singkat; pesan pertama akan digunakan sebagai pesan asli, dan kedua pesan lainnya akan digunakan sebagai pesan palsu.

Pesan Asli: "Jangan lupa malam ini kita beraksi."

Pesan Palsu-1: "Tolong bantu saya membetulkan pipa air."

Pesan Palsu-2: "Malam ini kita ketemu di tempat biasa."

Pesan asli diotentikasi dengan kunci "qwerty", sedangkan pesan lainnya dibiarkan tanpa kunci. Seluruh informasi tersebut kita masukkan pada tab "Chaffing" (Lihat Gambar 5). Seluruh kata diproses dan diurutkan berdasarkan nomor serial. Proses *Chaffing* akan menghasilkan format data sebagai berikut:

[nomor_serial]
[kata]
[MAC]

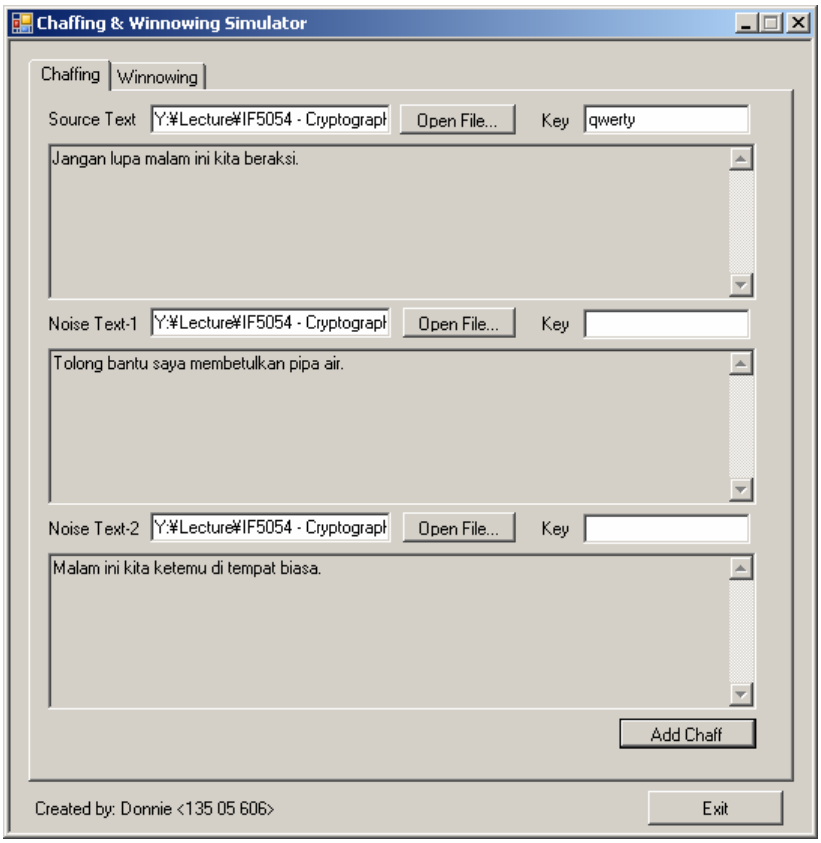
Selanjutnya, hasil dari proses *Chaffing* tersebut akan disimpan ke dalam sebuah file dengan nama test.cnw.

Untuk mendapatkan kembali pesan asli, maka kita buka kembali file test.cnw pada tab "Winnowing". Selanjutnya, masukkan kunci otentikasi yang digunakan sebelumnya. Terakhir, tekan tombol "Winnow" untuk mendapatkan pesan asli dari file tersebut. Hasil ekstraksi pesan asli tersebut dapat juga disimpan ke dalam file baru dengan menekan tombol "Save Result" (Lihat Gambar 6).

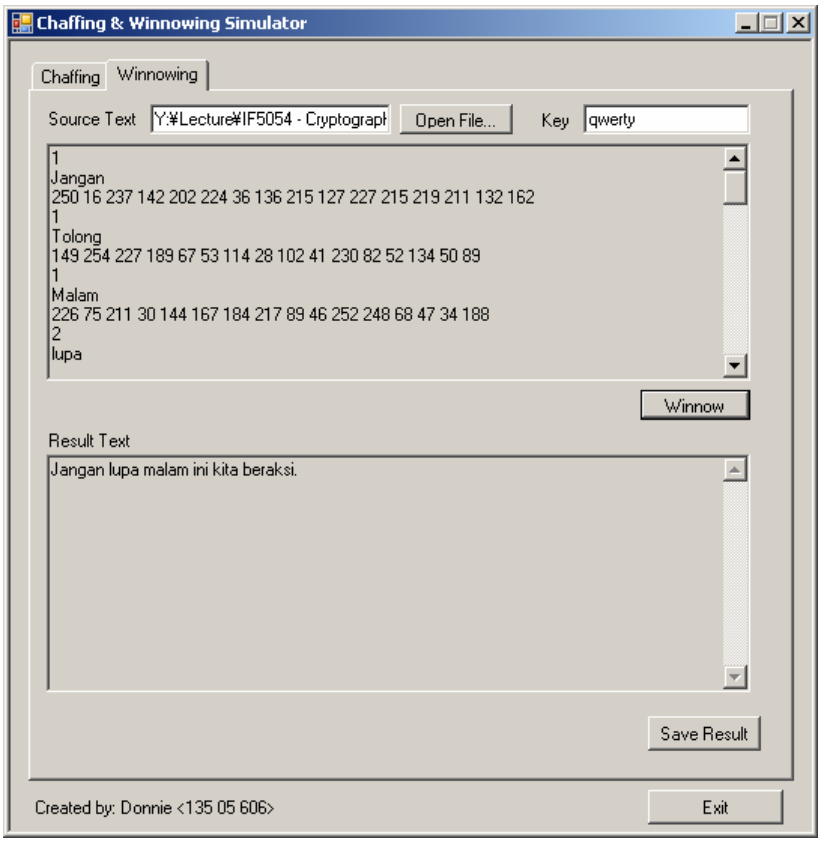
Catatan Mengenai Aplikasi Simulator yang Dibangun

1. Aplikasi yang dibangun tidak berjalan di atas jaringan, sehingga tidak mencerminkan situasi yang sesungguhnya diperuntukkan untuk teknik ini.

2. Aplikasi ini tidak mengacak urutan dari pesan asli maupun kedua pesan palsu lainnya, jadi jika dibaca secara teratur, maka pesan asli dapat dilihat dengan mudah.



Gambar 5. Aplikasi Simulator *Chaffing & Winnowing* Tab *Chaffing*



Gambar 6. Aplikasi Simulator *Chaffing & Winnowing* Tab *Winnowing*

```
private void ChaffButton_Click(object sender, EventArgs e){
	// Initializes new parser foe each file
	Parser P1 = new Parser(SourceTextBox.Text);
	Parser P2 = new Parser(NoiseTextBox1.Text);
	Parser P3 = new Parser(NoiseTextBox2.Text);
	// Initializes temporary container for each file
	List<string> d1;
	List<string> d2;
	List<string> d3;

	// Read a file, then insert it into containers
	P1.readFile(out d1);
	P2.readFile(out d2);
	P3.readFile(out d3);

	// Calculating Max Index value
	int max = 0;
	if (d1.Count > max) max = d1.Count;
	if (d2.Count > max) max = d2.Count;
	if (d3.Count > max) max = d3.Count;

	//Initializing Result Container
	List<string> outstr = new List<string>();

	for (int i = 0; i < max; i++)
	{
		if (i < d1.Count)
		{
			string f1 = HMACMD5Ex.EncodeText(SourceKeyBox.Text,
		d1[i]);
			outstr.Add((i + 1) + "");
			outstr.Add(d1[i]);
			outstr.Add(f1);
		}
		if (i < d2.Count)
		{
			string f2 = HMACMD5Ex.EncodeText(NoiseKeyBox1.Text,
		d2[i]);
			outstr.Add((i + 1) + "");
			outstr.Add(d2[i]);
			outstr.Add(f2);
		}
		if (i < d3.Count)
		{
			string f3 = HMACMD5Ex.EncodeText(NoiseKeyBox2.Text,
		d3[i]);
			outstr.Add((i + 1) + "");
			outstr.Add(d3[i]);
			outstr.Add(f3);
		}
	}

	// Save result from container to file
	if (saveFileDialog.ShowDialog() == DialogResult.OK)
	{
		StreamWriter outf = new
	StreamWriter(saveFileDialog.FileName);
		foreach(string s in outstr)
		outf.WriteLine(s);
		outf.Flush();
		outf.Close();
	}
}
```

Algoritma 1. Kode untuk Fungsi *Chaffing*

```
private void WinnowButton_Click(object sender, EventArgs e){
	// Insert source into temporary array
	String[] temps = SourceTextBox2.Text.Trim().Split(new Char[]
{ '\n', '\r' });
	// Initializing temporary container
	List<string> data = new List<string>();

	// Copying words from array into list container
	foreach (string s in temps)
	{
		if (s.Length != 0) data.Add(s);
	}
	// If data not complete, then fail
	if (data.Count % 3 != 0) return;

	// Initializes output container
	List<string> outr = new List<string>();
	Hashtable outh = new Hashtable();

	for (int i = 0; i < data.Count; i++)
	{
		int idx = 1;

		Int32.TryParse(data[i], out idx);
		i += 1;

		string text = data[i];
		text.Trim();
		i += 1;

		string hash = data[i];
		hash.Trim();

		string hash2 = HMACMD5Ex.EncodeText(textBox2.Text, text);

		if ( !outh.ContainsKey((int)idx) &&
	(String.Compare(hash2, hash) == 0) )
			outh.Add((int)idx, (string)text);
	}

	// Printing result into Result Text Box
	String result = "";
	for (int i = 0; i < outh.Count; i++ )
	{
		result += outh[(int)i + 1] + " ";
	}
	result.Trim();

	ResultTextBox.Text = result;

}
```

Algoritma 2. Kode untuk Fungsi *Winnowing*

```
class HMACMD5Ex{
	public static String EncodeText(String key, String sourceText){
		// Initialize the keyed hash object.
		HMACMD5 myhmacMD5 = new
	HMACMD5(Encoding.Default.GetBytes(key));

		// Compute the hash of the input file.
		byte[] hashValue =
	myhmacMD5.ComputeHash(Encoding.Default.GetBytes(sourceText));

		// Initialize hash value
		String result = "";

		// Storing hash value
		foreach (byte b in hashValue)
		{
			result += b + " ";
		}

		return result.Trim();
	}
}
```

Algoritma 3. Kode untuk Fungsi HMACMD5

Kesimpulan

Dari keseluruhan makalah ini, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Ronald L. Rivest telah memberikan sebuah teknik baru dalam memberikan kerahasiaan (*confientialty*).

2. Metode *Chaffing & Winnowing* dapat menjaga kerahasiaan tanpa menggunakan enkripsi ataupun steganografi.

3. Pemerintah Amerika Serikat tidak dapat meregulasi kerahasiaan dengan hanya meregulasikan enkripsi saja, karena akhirnya regulasi tersebut akan dikalahkan oleh perkembangan pengetahuan dalam dunia kriptografi. Ronald L. Rivest tidak memberikan bagaimana pemerintah Amerika Serikat dapat lebih efektif dalam membuat regulasi, tetapi memberikan sebuah pernyataan bahwa perdebatan seputar masalah kriptografi haruslah dilakukan dengan pendidikan yang bersifat mutual dan secara sukarela ("The cryptography debate should proceed by mutual education and voluntary action only"). Pernyataan lainnya yaitu bahwa perdebatan peraturan mengenai teknologi akan berakhir dikalahkan oleh perkembangan teknologi itu sendiri ("As usual, the policy debate about regulating technology ends up being obsoleted by technological innovations.").

Daftar Pustaka

[1] Annis, William S., "Chaffe – Chaffing and Winnowing Revision 1.5", Biomedical Computing Group, 1998.

[2] Bellare, Mihir, Alexandra Boldyreva, "The Security of Chaffing and Winnowing"Springer-Verlag, 2000.

[3] Reavis, Jim, "Chaffing and Winnowing", URL: http://www.networkworld.com/topics/security.html

[4] Science News Online, URL: http://sciencenews.org.

[5] Situs Bernstein, URL: http://cr.yp.to.

[6] Situs Hukum Inggris, Regulation of Investigatory Powers Act 2000, URL: http://www.opsi.gov.uk/acts/acts2000

[7] Situs Junger, URL: http://samsara.cwru.edu/comp_law/jvd/

[8] Spence, David, "A Review of Chaffing and Winnowing", SANS Institute, 2003.

[9] Rivest, Ronald L., "Chaffing and Winnowing: Confidentiality without Encryption", URL: http://theory.lcs.mit.edu/~rivest/chaffing.txt.