PENYERANGAN *CRYTOGRAPHIC PROTOCOL* MENGGUNAKAN *BASIC CRYPTANALYTIC ATTACKS*

Shieny Aprilia – 13505089

Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jln. Ganesha 10, Bandung
E-mail : if15089@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang penyerangan *cryptographic protocol* (protokol kriptografik), di mana model penyerangan ini dapat bermacam – macam bentuk dan implementasinya. Model – model penyerangan yang dibahas dalam makalah ini adalah *basic cryptanalytic attacks*. *Basic cryptanalytic attacks* ini ada 7 macam, yaitu : *Ciphertext-only attack*, *Known-plaintext attack*, *Chosen-plaintext attack*, *Chosen ciphertext attack*, *Adaptive chosen-ciphertext attack*, *Chosen-key attack*, dan *Rubber-hose cryptanalysis*.

Dalam makalah ini akan dibahas masing – masing *basic cryptanalytic attacks* secara lebih mendalam. Pembahasan ini meliputi sejarah mengenai masing - masing model penyerangan, bagaimana cara penyerangan itu bekerja, terhadap kriptosistem mana saja penyerangan itu efektif untuk dilakukan, dan contoh – contoh penyerangan yang dapat dan telah dilakukan terhadap suatu kriptosistem. Dalam makalah ini dibahas juga protokol seperti apa yang riskan untuk diserang oleh masing – masing model penyerangan.

Analisis matematis dari beberapa penyerangan juga akan dibahas dalam makalah ini. Analisis matematis ini berupa analisis sederhana mengenai beberapa model penyerangan, agar cara kerja penyerangan ini dapat lebih mudah dimengerti. Selain itu, terdapat pula analisis matematis dari kriptosistem yang aman dari model penyerangan secara umum. Sebagai pengantar, akan dibahas terlebih dahulu apa itu protokol dan fungsinya, kriptografi dan macam – macam kriptografi, dan protokol kriptografik.

Kata kunci : *Cryptographic Protocol*, Kriptografi, Kriptosistem, *Basic Cryptanalytic Attacks*, enkripsi, dekripsi, kunci

1. Pendahuluan

1.1 Pengertian Protokol

Suatu protokol adalah serangkaian langkah yang melibatkan dua pihak atau lebih dan dirancang untuk menyelesaikan suatu tugas. Dari definisi ini dapat diambil beberapa arti sebagai berikut :

- protokol memiliki urutan dari awal hingga akhir
- setiap langkah harus dilaksanakan secara bergiliran
- suatu langkah tidak dapat dikerjakan bila langkah sebelumnya belum selesai
- diperlukan dua pihak atau lebih untuk melaksanakan protokol
- protokol harus mencapai suatu hasil

Selain itu, suatu protokol pun memiliki karakteristik yang lain, yaitu :

- setiap orang yang terlibat dalam protokol harus mengetahui terlebih dahulu mengenai protokol dan seluruh langkah yang akan dilaksanakan
- setiap orang yang terlibat dalam protokol harus menyetujui untuk mengikutinya
- protokol tidak boleh menimbulkan kerancuan
- protokol harus lengkap

1.2 Fungsi Protokol

Dalam kehidupan kita sehari-hari terdapat banyak sekali protokol tidak resmi, misalnya saja dalam permainan kartu, pemungutan suara dalam pemilihan umum. Akan tetapi tidak ada seorang pun yang memikirkan mengenai protokol-protokol ini, protokol-protokol ini terus berkembang, semua orang mengetahui bagaimana menggunakannya.

Saat ini, semakin banyak interaksi antar manusia dilakukan melalui jaringan komputer. Komputer ini tentu saja memerlukan suatu protokol formal agar dapat melakukan hal yang biasa dilakukan manusia tanpa berpikir. Bila kita berpindah dari satu daerah ke daerah lain dan mengetahui bahwa kartu pemilihan suaranya berbeda dengan yang biasa kita gunakan, kita dapat beradaptasi

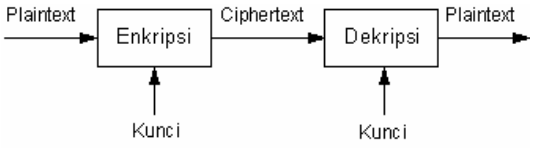
dengan mudah. Akan tetapi kemampuan ini belum dimiliki oleh komputer, sehingga diperlukan suatu protokol.

Protokol digunakan untuk mengabtraksikan proses penyelesaian suatu tugas dari mekanisme yang digunakan. Protokol komunikasi adalah sama meskipun diimplementasikan pada PC atau VAX. Bila kita yakin bahwa kita memiliki protokol yang baik, kita dapat mengimplementasikannya dalam segala benda mulai dari telepon hingga pemanggang roti cerdas.

1.3 Pengertian Kriptografi

Kriptografi, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita [Bruce Schneier - *Applied Cryptography*]. Selain pengertian tersebut terdapat pula pengertian ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data [A. Menezes, P. van Oorschot and S. Vanstone - *Handbook of Applied Cryptography*]. Tidak semua aspek keamanan informasi ditangani oleh kriptografi.

Menjaga kerahasiaan berita tersebut dilakukan dengan menyandikan pesan atau berita itu. Suatu pesan yang tidak disandikan disebut sebagai *plaintext* (plainteks) ataupun dapat disebut juga sebagai *cleartext*. Sedangkan suatu pesan yang sudah disandikan disebut sebagai *ciphertext* (cipherteks). Proses yang dilakukan untuk mengubah plainteks ke dalam cipherteks disebut *encryption* (enkripsi) atau *encipherment*. Sedangkan proses untuk mengubah *ciphertext* kembali ke *plaintext* disebut *decryption* (dekripsi) atau *decipherment*. Secara sederhana istilah-istilah di atas dapat digambarkan sebagai berikut:



Gambar 1 Proses Enkripsi – Dekripsi Sederhana

Cryptanalysis adalah suatu ilmu dan seni membuka (*breaking*) cipherteks dan orang yang melakukannya disebut *cryptanalyst*.

Gambar 2 *Cryptanalysis* Klasik

Cryptographic system atau *cryptosystem* adalah suatu fasilitas untuk mengkonversikan plainteks ke cipherteks dan sebaliknya. Dalam sistem ini, seperangkat parameter yang menentukan transformasi pencipheran tertentu disebut suatu set kunci. Proses enkripsi dan dekripsi diatur oleh satu atau beberapa kunci kriptografi. Secara umum, kunci – kunci yang digunakan untuk proses pengenkripsian dan pendekripsian tidak perlu identik, tergantung pada sistem yang digunakan.

Secara umum operasi enkripsi dan dekripsi dapat diterangkan secara matematis sebagai berikut :

EK (M) = C (Proses Enkripsi)

DK (C) = M (Proses Dekripsi)

Pada saat proses enkripsi kita menyandikan pesan M dengan suatu kunci K lalu dihasilkan pesan C. Sedangkan pada proses dekripsi, pesan C tersebut diuraikan dengan menggunakan kunci K sehingga dihasilkan pesan M yang sama seperti pesan sebelumnya.

Dengan demikian keamanan suatu pesan tergantung pada kunci ataupun kunci – kunci yang digunakan, dan tidak tergantung pada algoritma yang digunakan. Sehingga algoritma – algoritma yang digunakan tersebut dapat dipublikasikan dan dianalisis, serta produk – produk yang menggunakan algoritma tersebut dapat diproduksi massal. Tidaklah menjadi masalah apabila seseorang mengetahui algoritma yang kita gunakan. Selama ia tidak mengetahui kunci yang dipakai, ia tetap tidak dapat membaca pesan.

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

a. Kerahasiaan (*confidentiality*)

Kerahasiaan adalah layanan yang digunakan untuk menjaga informasi dari setiap pihak yang tidak berwenang untuk mengaksesnya. Dengan demikian informasi hanya akan dapat diakses oleh pihak – pihak yang berhak saja.

b. Integritas data (*data integrity*)

Integritas data merupakan layanan yang bertujuan untuk mencegah terjadinya pengubahan informasi oleh pihak – pihak yang tidak berwenang. Untuk meyakinkan integritas data ini harus dipastikan agar sistem informasi mampu mendeteksi terjadinya manipulasi data. Manipulasi data yang dimaksud di sini meliputi penyisipan, penghapusan, maupun penggantian data.

c. Otentikasi (*authentication*)

Otentikasi merupakan layanan yang terkait dengan identifikasi terhadap pihak – pihak yang ingin mengakses sistem informasi (*entity authentication*) maupun keaslian data dari sistem informasi itu sendiri (*data origin authentication*).

d. Ketiadaan penyangkalan (*non-repudiation*)

Ketiadaan penyangkalan adalah layanan yang berfungsi untuk mencegah terjadinya penyangkalan terhadap suatu aksi yang dilakukan oleh pelaku sistem informasi.

1.4 *Cryptographic System* (*Cryptosystem*)

Suatu *cryptosystem* terdiri dari sebuah algoritma, seluruh kemungkinan plainteks, cipherteks, dan kunci-kunci. Secara umum *cryptosystem* dapat digolongkan menjadi dua buah, yaitu :

1. *Symmetric Cryptosystem*

Dalam symmetric *cryptosystem* ini, kunci yang digunakan untuk proses enkripsi dan dekripsi pada prinsipnya identik, tetapi satu buah kunci dapat pula diturunkan dari kunci yang lainnya. Kunci-kunci ini harus dirahasiakan. Oleh karena itulah sistem ini sering disebut sebagai *secret-key ciphersystem*. Jumlah kunci yang dibutuhkan umumnya adalah :

$$ {}_nC_2 = \frac{n.(n-1)}{2} $$

dengan n menyatakan banyaknya pengguna.

Contoh dari sistem ini adalah *Data Encryption Standard* (DES), *Blowfish*, IDEA. Kriptosistem ini disebut juga Kunci Pribadi (*Private Key*). Kunci pribadi disini berarti bahwa pemegang kunci enkripsi maupun dekripsi hanyalah pihak – pihak berwenang saja. Karena melihat kembali sifatnya, bila pihak ketiga memperoleh salah satu kunci tersebut maka dia bisa memperoleh kunci yang lain.

2. *Assymmetric Cryptosystem*

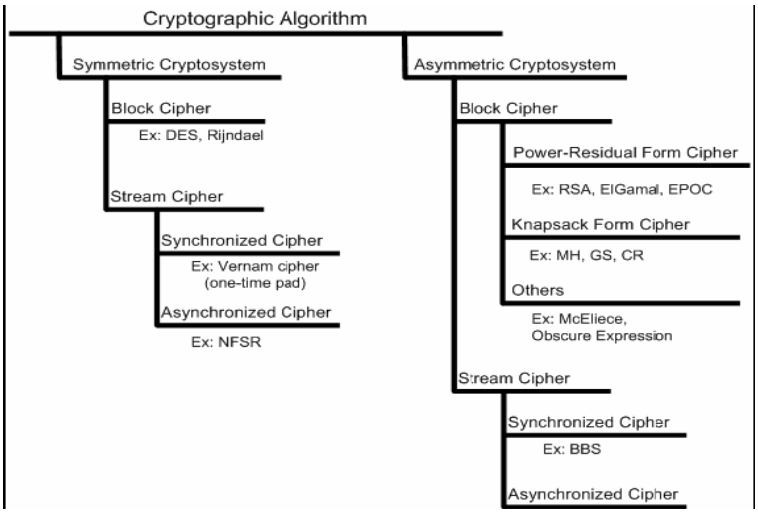
Dalam *assymmetric cryptosystem* ini digunakan dua buah kunci. Satu kunci yang disebut kunci publik (*public key*), yaitu kunci enkripsi, dapat dipublikasikan, sedang kunci yang lain yang disebut kunci privat (*private key*), yaitu kunci dekripsi, harus dirahasiakan. Proses menggunakan sistem ini dapat diterangkan secara sederhana sebagai berikut : bila A ingin mengirimkan pesan kepada B, A dapat menyandikan pesannya dengan menggunakan kunci publik B, dan bila B ingin membaca surat tersebut, ia perlu mendekripsikan surat itu dengan kunci privatnya. Dengan demikian kedua belah pihak dapat menjamin asal surat serta keaslian surat tersebut, karena adanya mekanisme ini. Contoh sistem ini antara lain RSA *Scheme* dan Merkle-Hellman *Scheme*.

Setiap *cryptosytem* yang baik harus memiliki karakteristik sebagai berikut :

- Keamanan sistem terletak pada kerahasiaan kunci dan bukan pada kerahasiaan algoritma yang digunakan
- *Cryptosystem* yang baik memiliki ruang kunci (*keyspace*) yang besar
- *Cryptosystem* yang baik akan menghasilkan ciphertext yang terlihat acak dalam seluruh tes statistik yang dilakukan terhadapnya
- *Cryptosystem* yang baik mampu menahan seluruh serangan yang telah dikenal sebelumnya

Namun demikian perlu diperhatikan bahwa bila suatu *cryptosystem* berhasil memenuhi seluruh karateristik di atas belum tentu ia merupakan sistem yang baik. Banyak *cryptosystem* lemah yang terlihat baik pada awalnya. Kadang kala untuk menunjukkan bahwa suatu *cryptosystem* kuat atau baik dapat dilakukan dengan menggunakan pembuktian matematika.

Hingga saat ini masih banyak orang yang menggunakan *cryptosystem* yang relatif mudah dibuka, alasannya adalah mereka tidak mengetahui sistem lain yang lebih baik serta kadang kala terdapat motivasi yang kurang untuk



Gambar 3 Pengelompokan Kriptosistem Berikut Contohnya

menginvestasikan seluruh usaha yang diperlukan untuk membuka suatu sistem.

Baik kriptosistem simetrik maupun asimetrik, teknik kriptografinya dapat dibagi menjadi dua bagian, berdasarkan dari jenis data yang diolah, yaitu : *Block Cipher* dan *Stream Cipher*. Pada Block Cipher, sesuai namanya data Plain Teks diolah per blok data. Di lain pihak, pada Stream Cipher, data Plain Teks diolah per satuan data terkecil, misalnya per bit atau per karakter.

Teknik kriptografi kemudian dibagi lagi menjadi dua kelompok, yaitu *Synchronized Cipher* di mana Kunci Enkripsi dan Kunci Dekripsi perlu disinkronisasi, dan *Asynchronized Cipher* di mana sinkronisasi tidak diperlukan. Stream Cipher dapat dibagi menjadi dua kelompok ini, sedangkan pada Block Cipher hanya ada Synchronized Cipher.

Power-Residual Form Cipher adalah teknik di mana dalam proses enkripsi digunakan rumus matematika X^Y (mod N) atau rumus yang mirip seperti itu. *Knapsack Form Cipher* adalah teknik yang menggunakan *Knapsack Problem* yang merupakan problem komputasi kelas NP-Komplit (NP-Complete/NP-Hard).

1.5 Pengertian *Cryptographic Protocol*

Cryptographic protocol adalah suatu protokol yang menggunakan kriptografi. Protokol ini melibatkan sejumlah algoritma kriptografi, namun secara umum tujuan protokol lebih dari sekedar kerahasiaan. Pihak-pihak yang berpartisipasi mungkin saja ingin membagi sebagian rahasianya untuk menghitung sebuah nilai, menghasilkan urutan random, atau pun menandatangani kontrak secara bersamaan. Penggunaan kriptografi dalam sebuah protokol terutama ditujukan untuk mencegah atau pun mendeteksi adanya *eavesdropping* dan *cheating*.

2. Penyerangan terhadap Protokol

Penyerangan cryptographic dapat ditujukan pada beberapa hal berikut :

- Algoritma cryptographic yang digunakan dalam protokol
- Teknik cryptographic yang digunakan untuk mengimplementasikan algoritma dan protokol
- Protokol itu sendiri

Seseorang dapat mencoba berbagai cara untuk menyerang suatu protokol. Mereka yang tidak terlibat dalam protokol dapat menyadap sebagian atau seluruh protokol. Tindakan ini disebut penyerangan pasif, karena si penyerang tidak mempengaruhi atau mengubah protokol, ia hanya mengamati protokol dan berusaha untuk memperoleh informasi.

Selain itu, seorang penyerang dapat berusaha untuk mengubah protokol demi keuntungannya sendiri. Ia dapat mengirimkan pesan dalam protokol, menghapus pesan, atau bahkan mengubah informasi yang ada di dalam suatu komputer. Tindakan-tindakan ini disebut sebagai penyerangan aktif, karena ia membutuhkan suatu campur tangan aktif.

Seorang penyerang tidaklah hanya berasal dari lingkungan luar protokol, namun ia mungkin juga berasal dari dalam protokol itu sendiri, ia dapat merupakan salah satu pihak yang terlibat dalam protokol. Tipe penyerang semacam ini

disebut sebagai *cheater*. *Passive cheater* mengikuti protokol, tetapi berusaha memperoleh informasi lebih banyak daripada yang diperbolehkan protokol bagi dirinya. *Active cheater* mengubah protokol dalam usahanya untuk berbuat curang.

Usaha untuk menjaga keamanan protokol akan semakin sulit apabila pihak-pihak yang terlibat umumnya merupakan *active cheater*, oleh karena itu suatu protokol yang baik harus mampu atau pun harus aman terhadap kemungkinan *passive cheating*.

3. *Cryptanalytic Attacks*

Tujuan *cryptanalytic attack* adalah untuk mengetahui beberapa plainteks yang sesuai dengan cipherteks yang ada dan berusaha menentukan kunci yang memetakan satu dengan yang lainnya. Plainteks ini dapat diketahui karena ia merupakan standar atau karena pendugaan. Jika suatu teks diduga berada di dalam suatu pesan, posisinya mungkin tidak diketahui, tetapi suatu pesan lazimnya cukup pendek sehingga memungkinkan kriptanalis menduga plainteks yang diketahui dalam setiap posisi yang mungkin dan melakukan penyerangan pada setiap kasus secara paralel.

Suatu algoritma enkripsi yang kuat tidak hanya mampu bertahan terhadap serangan plainteks yang dikenal tetapi juga mampu bertahan terhadap *adaptive chosen* plainteks. Dalam penyerangan ini, kriptanalis berkesempatan memilih plainteks yang digunakan dan dapat melakukannya secara berulang kali, memilih plainteks untuk tahap N+1 setelah menganalisis hasil tahap N.

Yang dimaksud *cryptanalytic attacks* adalah usaha-usaha yang dilakukan seseorang untuk memperoleh informasi ataupun data yang telah dienkripsi. Dalam hal ini, *cryptanalytic attacks* dapat digunakan untuk menyerang suatu *cryptographic protocol*. Secara ringkas terdapat tujuh macam *basic cryptanalytic attacks* berdasarkan tingkat kesulitannya bagi penyerang, dimulai dari yang paling sulit adalah *Ciphertext-only attack*, *Known-plaintext attack*, *Chosen-plaintext attack*, *Chosen ciphertext attack*, *Adaptive chosen-ciphertext attack, Chosen-key attack*, dan *Rubber-hose cryptanalysis*.

3.1 *Ciphertext-only Attack*

Ciphertext-only attack (COA) adalah model penyerangan untuk *cryptanalysis* di mana si penyerang hanya mempunyai akses ke cipherteks saja. Penyerangan dikatakan sangat berhasil apabila plainteks yang berkorespondensi dengan cipherteks dapat diketahui, atau lebih baik lagi jika kuncinya juga dapat diketahui. Kemampuan untuk memperoleh informasi dari plainteks tetap dipertimbangkan sebagai suatu keberhasilan. Sebagai contoh, jika lawan mengirimkan cipherteks terus – menerus untuk mengatur *traffic-flow security*, akan sangat berguna jika kita dapat membedakan pesan sesungguhnya dari nol.

Dalam sejarah kriptografi, sebuah sandi yang diimplementasikan menggunakan pena dan kertas, dapat dipecahkan hanya dengan menggunakan cipherteksnya saja. Para kriptografer menciptakan teknik statistik untuk penyerangan cipherteks, seperti misalnya analisis frekuensi. Alat enkripsi mekanis seperti Enigma membuat penyerangan jenis ini menjadi jauh lebih sulit (walaupun berdasarkan sejarah, para kriptografer Polandia dapat berhasil menggunakan *ciphertext-only* untuk kriptanalisis menggunakan Enigma dengan memanfaatkan protokol yang tidak aman untuk mencari informasi mengenai pengaturan pesan).

Setiap sandi modern mencoba untuk menyediakan perlidungan terhadap *ciphertext-only attacks*. Proses untuk mendesain sandi baru biasanya memerlukan beberapa tahun dan memerlukan banyak tes yang melelahkan terhadap banyak macam cipherteks untuk semua macam data statistik yang tersedia.

Meskipun demikian, penggunaan sandi yang buruk pada algoritma milik pribadi yang belum melalui berbagai penelitian secara cermat terdapat pada berbagai sistem enkripsi komputer dari masa ke masa, yang masih menjadi objek dari *ciphertext-only attack*. Contohnya sebagai berikut :

- Versi permulaan dari software *Microsoft's PPTP virtual private network* menggunakan kunci RC4 yang sama untuk pengirim dan penerimanya (versi yang setelahnya mempunyai masalah yang lain). Pada berbagai kasus di mana *stream cipher* seperti RC4 digunakan dua kali dengan kunci yang sama, membuka kesempatan untuk *ciphertext-only attack*.
- *Wired Equivalent Privacy* (WEP), keamanan protokol yang pertama untuk *Wi-Fi*, membuktikan kemudahannya untuk diserang oleh beberapa penyerangan yang kebanyakan adalah *ciphertext-only attack.*

- Baru – baru ini, beberapa desain sandi modern ditunjukkan bahwa ia mudah diserang oleh *ciphertext-only attacks*. Contohnya Akelarre.
- Sandi yang ruang untuk kuncinya terlalu sempit dapat menjadi objek *brute-force attack* dengan akses hanya pada cipherteks dengan cara mencoba – coba semua kemungkinan kunci yang sederhana. Yang dibutuhkan hanyalah suatu cara untuk membedakan plainteks yang valid dengan kata – kata pada bahasa biasa secara acak. Sebagai contoh adalah DES (*Data Encryption Standard*), yang hanya mempunyai kunci dengan 56 bit. Kebanyakan contoh yang umum adalah produk keamanan komersial yang memperoleh kunci untuk sandi yang dapat tahan serangan seperti AES (*Advanced Encryption Standard*). Karena user jarang menggunakan password yang mendekati entropi ruang kunci sandi, maka sistem seperti itu akan sering dipecahkan dengan mudah hanya dengan menggunakan cipherteks saja.

3.2 *Known-plaintext Attack*

Known-plaintext attack adalah model penyerangan untuk *cryptanalysis* di mana si penyerang mempunyai sampel baik untuk plainteks dan versi yang dienkripsi (cipherteks) dan juga si penyerang mempunyai kebebasan untuk membuka informasi rahasia lebih jauh, khususnya kunci rahasianya.

Pada Perang Dunia II, di Bletchley Park, telah diusahakan dengan keras untuk memaksa Jerman mengeluarkan pesan dengan plainteks yang telah diketahui. Plainteks yang telah diketahui ini dinamakan ”*cribs*”, dan pola untuk memaksa Jerman untuk mengeluarkan *cribs* disebut “*gardening*”. Pemecah sandi Polandia, Bureau, memanfaatkan *cribs* pada ”ANX *method*” sebelum Perang Dunia II.

Tempat penyimpanan file (*archives*) yang telah dienkripsi seperti ZIP juga sangat mudah untuk diserang dengan model penyerangan ini. Sebagai contoh, si penyerang dengan file ZIP yang telah dienkripsi hanya membutuhkan satu buah file yang belum dienkripsi dari *archive* itu, yang membentuk ”*known-plaintext*”. Lalu dengan menggunakan software publik yang tersedia, mereka dapat dengan mudah dan cepat mengetahui kunci yang dibutuhkan untuk mendekripsi seluruh *archive*.

Untuk memperoleh file yang belum dienkripsi ini, si penyerang dapat mencari file yang cocok di website, mencari dari *archive* lain yang dapat dibuka, atau secara manual mencoba membuat kembali file plainteks dengan bantuan pengetahuan mengenai nama file dari *archive* yang belum dienkripsi.

Sandi klasik khususnya mudah diserang menggunakan *known-plaintext attack*. Sebagai contoh, sandi Caesar dapat dipecahkan menggunakan sebuah huruf yang berkorespondensi dengan plainteks dan cipherteks untuk mendekripsi seluruh sandi. Sandi *monoalphabetic substitution* (penggantian huruf tunggal) yang umum membutuhkan beberapa pasangan karakter dan beberapa terkaan jika terdapat lebih sedikit dari 26 pasangan karakter yang jelas.

3.3 *Chosen-plaintext Attack*

Chosen-plaintexkt attack (CPA) adalah model penyerangan untuk *cryptanalysis* yang mengasumsikan bahwa si penyerang mempunyai kemampuan untuk memilih plainteks sesuai keinginannya untuk dienkripsi dan memperoleh cipherteks yang berkorespondensi dengannya. Tujuan dari penyerangan ini adalah untuk memperoleh informasi yang lebih dalam yang akan mengurangi keamanan dari pola sebuah enkripsi. Pada kasus terburuk, *chosen-plaintext attack* dapat membuka kunci rahasia dari pola enkripsi itu.

Secara sekilas, tampaknya model ini merupakan model yang tidak realistik, karena sepertinya tidak mungkin bahwa si penyerang dapat membujuk seorang kriptografer manusia untuk mengenkripsi plainteks dalam jumlah besar sesuai dengan pilihan si penyerang. Tetapi, kriptografi modern diimplementasikan pada software atau hardware dan digunakan untuk bermacam – macam aplikasi. Pada banyak kasus, *chosen-plaintext attack* seringkali sangat mudah untuk dikerjakan. *Chosen-plaintext attacks* menjadi sangat penting pada konteks kriptografi kunci publik, di mana kunci enkripsinya dipublikasikan dan si penyerang dapat mengenkripsikan plainteks mana saja yang dipilihnya.

Sandi manapun yang dapat mencegah *chosen-plaintext attacks* dijamin dapat aman terhadap *known-plaintext* dan *ciphertext-only attacks*.

Bentuk - bentuk *chosen-plaintext attack* dapat dibedakan menjadi dua :

a. *Batch chosen-plaintext attack*

   Di mana kriptanalis memilih semua plainteks sebelum satupun dienkripsi. Ini berarti fungsi yang lengkap dari *chosen-plaintext attack*.

b. *Adaptive chosen-plaintext attack*

   Di mana kriptanalis membuat sejumlah seri pertanyaan yang interaktif, untuk memilih plainteks berdasarkan informasi dari enkripsi yang sebelumnya.

Algoritma enkripsi kunci publik yang tidak diacak sangat mudah untuk diserang oleh tipe penyerangan yang berjalan seperti kamus yang sederhana, di mana si penyerang membuat tabel dengan pesan dan cipherteks yang berkorespondensi dengannya. Untuk mencari dekripsi dari suatu cipherteks, si penyerang dengan mudah dapat melihat cipherteks di tabel itu. Oleh karena itu, keamanan yang menggunakan kunci publik dan dapat diserang oleh *chosen-plaintext attack* membutuhkan enkripsi probabilistik (enkripsi acak). Metode sandi simetri yang konvensional, yang menggunakan kunci yang sama untuk menenkripsi dan mendekripsi sebuah teks, juga sangat mudah diserang oleh bentuk lain dari *chosen-plaintext attack*, sebagai contoh, kriptanalisis diferensial dari sandi blok.

Teknik yang dikenal dengan *Gardening* digunakan oleh pemecah kode Aliansi pada Perang Dunia II, yang memecahkan pesan yang dienkripsi pada mesin Enigma. *Gardening* dapat dilihat sebagai *chosen-plaintext attack.*

3.4 *Chosen-ciphertext Attack*

Chosen-ciphertext attack (CCA) adalah model penyerangan untuk *cryptanalysis* di mana kriptanalis memilih cipherteks dan menyebabkannya didekripsi dengan menggunakan kunci yang belum diketahui. Bentuk spesifik dari penyerangan ini terkadang dinamakan penyerangan "*lunchtime*" atau "*midnight*", berkenaan dengan skenarionya, di mana si penyerang mendapat akses ke mesin dekripsi yang tidak diawasi. Alat yang menyediakan dekripsi pada cipherteks pilihan (baik karena kebetulan ataupun desainnya) secara umum dikenal sebagai "*decryption oracle*" .

Jelas terlihat, pihak lawan yang dapat mendekripsikan pesan pilihan (menggunakan "*decryption oracle*") dapat menurunkan kepercayaan terhadap pola enkripsi tersebut. Walau bagaimanapun, *chosen-ciphertext attack* dapat menyebabkan akibat yang signifikan untuk kriptosistem tertentu. Sebagai contoh, pada kasus yang ekstrim si penyerang mungkin dapat memperoleh kunci rahasia pola dekripsi dengan memperhatikan baik – baik cipherteks pilihannya dan menganalisis hasil pendekripsiannya. *Chosen-ciphertext attack* yang sukses dapat membahayakan keamanan dari pola kriptografi bahkan setelah "*decription oracle*" menjadi tidak dapat diakses lagi. Sebagai alternatif, penyerangan jenis ini akan efektif pada kasus di mana *decryption oracle* tidak dapat digunakan langsung untuk mendekripsikan ciphertext yang dimaksud.

Sejumlah pola kriptografi yang aman lainnya dapat dipecahkan menggunakan *chosen-ciphertext attack*. Sebagai contoh, kriptosistem El Gamal yang secara semantik aman terhadap *chosen-plaintext attack*, tetapi keamanan semantik ini dapat dipecahkan dengan menggunakan *chosen-ciphertext attack*. Versi sebelumnya dari RSA (Rivest-Shamir-Adleman) yang digunakan pada protokol SSL (*Secure Socket Layer*) sangat mudah diserang oleh *adaptive chosen-ciphertext attack*, yang membuka kunci sesi SSL. *Chosen-ciphertext attacks* juga dapat mengakibatkan proses *self-synchronizing* pada *stream cipher*. Desainer kartu pintar (*smart cards*) kriptografik yang anti penyuapan haruslah mengetahui tipe penyerangan ini, karena alat ini dapat benar – benar berada di bawah kontrol lawan, yang dapat mengeluarkan cipherteks pilihan dalam jumlah besar dalam rangka membuka kunci rahasia yang tersembunyi.

Ketika sebuah kriptosistem dapat dengan mudah diserang menggunakan *chosen-ciphertext attack*, si pembuat kriptosistem haruslah waspada untuk menghindari situasi di mana lawan mungkin dapat mendekripsikan cipherteks pilihannya (dengan kata lain, menghindari tersedianya *decryption oracle*). Ini dapat menjadi lebih sulit dari yang dapat dibayangkan, karena cipherteks pilihan itu dapat mengakibatkan serangan yang tidak kentara dapat menyerang sistem. Sebagai tambahan, beberapa kriptosistem (seperti RSA) menggunakan mekanisme yang sama untuk menandai pesan dan mendekripsikannya. Ini mengakibatkan penyerangan ketika proses *hashing* tidak digunakan pada pesan yang harus ditandai. Pendekatan yang lebih baik adalah untuk menggunakan kriptosistem yang sudah terbukti aman terhadap *chosen-ciphertext attack*, termasuk diantaranya RSA-OAEP, Cramer-

Shoup, dan banyak bentuk dari enkripsi simetri yang telah diautentikasi.

Pada *non-adaptive chosen-ciphertext attack*, dikenal sebagai *indifferent chosen-ciphertext attack* (penyerangan "*lunchtime*"), lawan mempunyai akses ke *decryption oracle* hanya sebelum ia memilih cipherteks spesifik yang akan diserang. Oleh karena itu, tujuan penyerangan adalah untuk memperoleh sedikit demi sedikit informasi untuk melemahkan pola kriptografi terhadap banyaknya variasi dari cipherteks. Pada skenario penyerangan yang paling sukses, penyerangan ini sukses membuka kunci dekripsi rahasia dan hal itu menghancurkan seluruhnya pola kriptografi. *Adaptive chosen-ciphertext attack* (penyerangan "*midnight*") memperluas skenario sebelumnya dengan memperbolehkan lawan menggunakan *decryption oracle* bahkan setelah ia memilih cipherteks spesifik yang akan diserang (untuk membuat penyerangan menjadi *non-trivial,* lawan dicegah untuk mendekripsikan cipherteks yang dimaksud). Tujuan dari penyerangan ini adalah untuk memperoleh informasi, termasuk dekripsi dari cipherteks target. Penyerangan ini dapat digunakan terhadap pola kunci publik, termasuk RSA. Penyerangan ini dapat dicegah melalui penggunaan yang tepat dari materi kriptografi atau pengecekan redundansi.

3.5 *Adaptive chosen-ciphertext Attack*

Adaptive-chosen-ciphertext attack (disingkat menjadi CCA2) adalah bentuk interaktif dari *chosen-ciphertext attack* di mana si penyerang mengirimkan banyaknya cipherteks yang akan didekripsi, lalu menggunakan hasil dekripsi itu untuk memilih cipherteks berikutnya. Hal itu untuk membedakannya dari *indifferent-chosen-ciphertext attack* (CCA1).

Tujuan dari penyerangan ini adalah untuk membuka informasi mengenai pesan yang telah dienkripsi atau mengenai kunci dekripsi itu sendiri. Untuk sistem dengan kunci publik, *adaptive-chosen-ciphertext* umumnya dapat digunakan hanya ketika mereka mempunyai properti dari *ciphertext malleability*, yaitu cipherteks yang dapat dimodifikasi dengan suatu cara spesifik sehingga dapat mengakibatkan efek yang dapat diduga pada proses dekripsi pesan.

Adaptive-chosen-ciphertext attacks diteliti secara teoritis sampai 1998, ketika Daniel Bleichenbacher dari Bell Laboratories mendemonstrasikan penyerangan praktek melawan sistem yang menggunakan enkripsi RSA bersama dengan fungsi penyandi PKCS #1 v1, termasuk versi dari Protokol SSL yang digunakan oleh ribuan web server pada saat itu. Penyerangan Bleichenbacher mengambil keuntungan dari cacat yang terdapat pada fungsi PKCS #1 untuk membuka isi dari pesan RSA yang telah dienkripsi. Hal ini dilakukan dengan mengirim jutaan tes cipherteks ke alat pendekripsi (contoh : web server yang dilengkapi dengan SSL). Pada prakteknya, ini berarti kunci sesi SSL dapat diketahui dalam waktu yang relatif singkat, mungkin satu hari atau kurang.

Dalam rangka mencegah *adaptive-chosen-ciphertext attacks*, adalah suatu kewajiban untuk menggunakan pola enkripsi atau pengkodean yang dapat membatasi *ciphertext malleability*. Sejumlah pola pengkodean telah dikemukakan, yang paling umum untuk enkripsi RSA adalah *Optimal Asymmetric Encryption Padding* (OAEP). Tidak seperti pola ad-hoc seperti materi yang digunakan pada PKCS #1 v1, OAEP telah dijamin aman terhadap model peramalan acak (*random oracle model*).

Semua sistem kriptografik, yang plainteksnya telah diawasi, akan aman terhadap *adaptive-chosen-ciphertext attack*.

3.6 *Chosen-key Attack*

Chosen-key attack adalah model penyerangan untuk *cryptanalysis* di mana kriptanalis pada tipe penyerangan ini memiliki pengetahuan tentang hubungan antara – antara kunci – kunci yang berbeda.

3.7 *Rubber-hose Cryptanalysis*

Pada kriptografi, *rubber-hose cryptanalysis* adalah eufemisme untuk memperoleh rahasia kriptografi dari orang dengan cara memaksa atau menyiksa, kontras dengan *cryptanalytic attack* yang matematis ataupun teknis. Nama dari penyerangan ini berkenaan dengan pemukulan pipa air dari karet, konotasi dari pemaksaan atau penyiksaan.

Walaupun namanya seperti main – main, tidak demikian dengan akibat yang ditimbulkannya. Pada kriptosistem modern, manusia seringkali merupakan yang mata rantai informasi yang terlemah. Penyerangan langsung terhadap algoritma sandi, atau protokol kriptografi yang digunakan akan jauh lebih mahal dan sulit daripada menyerang para pengguna sistem. Oleh karena itu, banyak kriptosistem dan sistem keamanan didesain dengan meminimumkan kemungkinan diserangnya manusia, seperti pada

pembangkitan kunci atau penggunaan kunci, sehingga segala ancaman pada operator atau personil lainnya akan menjadi tidak efektif untuk menghancurkan sistem.

Pada beberapa yurisdiksi, undang – undang mengasumsikan yang sebaliknya, bahwa operator manusia mengetahui atau mempunyai akses ke sesuatu seperti kunci sesi (*session keys*), asumsi yang sama dengan yang dimiliki oleh mereka yang mempraktekkan *rubber-hose cryptanalysis*. Sebagai contoh adalah undang - undang UK RIP, yang memutuskan siapapun yang tidak menyerahkan kunci pada permintaan resmi dari pegawai pemerintah yang diberi kuasa melalui undang – undang, dinyatakan sebagai kriminal. Para pengguna atau pemilik kriptosistem tersebut mungkin tidak dapat untuk melakukannya (karena telah membuat sesuatu yang dapat menahan serangan *rubber-hose*). Salah satu kemungkinan interpretasi untuk memecahkan ini adalah legislatif seperti RIP diharapkan untuk menggunakan akibat yang mengerikan dari penggunaan kriptografi.

4. Analisis Beberapa Tipe Penyerangan dan Kriptosistem Secara Matematis

Suatu penyerangan pasif atas *cryptosystem* adalah semua metode untuk mengungkapkan informasi tentang plainteks dan cipherteksnya dengan tanpa mengetahui kunci. Secara matematis :

a. Diberikan fungsi F, G, dan H yang terdiri dari n variabel.
b. Diberikan sistem enkripsi E.
c. Diberikan suatu distribusi plaintext dan kunci.

Suatu penyerangan atas E dengan menggunakan G dengan mengasumsikan F membagi H dengan probabilitas p adalah suatu algoritma A dengan sepasang input f,g dan satu buah output h sedemikian hingga terdapat probabilitas p atas h = $H(P_1, \ldots, P_n)$, jika kita memiliki f = $F(P_1, \ldots, P_n)$ dan g = $G(E_K(P_1), \ldots, EK(P_n))$. Perlu diperhatikan bahwa probabilitas ini tergantung pada distribusi vektor-vektor $(K,P_1,\ldots,P_n)$.

Penyerangan akan merupakan suatu trivial bila terdapat probabilitas paling sedikir p untuk h = $H(P_1, \ldots, P_n)$ jika f = F $(P_1,\ldots,P_n)$ dan g = G $(C_1,\ldots,C_n)$. Di sini $C_1,\ldots,C_n$ terletak pada ciphertext yang mungkin, dan tidak memiliki hubungan tertentu dengan $P_1,\ldots,P_n$. Dengan kata lain, suatu serangan akan merupakan trivial bila ia tidak benar-benar menggunakan enkripsi $E_K(P_1),\ldots,E_K(P_n)$.

Dengan merumuskan penyerangan secara matematis, kita dapat secara tepat memformulasikan dan bahkan membuktikan pernyataan bahwa suatu kriptosistem itu kuat. Kita katakan, sebagai contoh, bahwa suatu kriptosistem adalah aman terhadap seluruh penyerangan pasif jika sembarang penyerangan nontrivial terhadapnya tidak praktis. Jika kita dapat membuktikan pernyataan ini maka kita akan memiliki keyakinan bahwa kriptosistem kita akan bertahan terhadap seluruh teknik *cryptanalytic* pasif. Jika kita dapat mereduksi pernyataan ini hingga pada beberapa masalah yang tidak terpecahkan maka kita masih tetap memiliki keyakinan bahwa cryptosystem kita tidak mudah dibuka.

4.1 Kriptosistem *One-time Pad*

One-time Pad dikenal sebagai suatu kriptosistem yang aman. Hal ini akan dijelaskan di bawah ini.

Secara definisi, *one-time pad* adalah suatu kriptosistem di mana plainteks, cipherteks, dan kunci merupakan string (byte string) dengan panjang m, dan $E_K(P)$ adalah :

$$K \oplus P$$

Mudah untuk dibuktikan bahwa tidak ada penyerangan cipherteks tunggal non-trivial pada *one-time pad*, dengan asumsi distribusi kunci yang seragam. Dengan catatan, kita tidak perlu mengasumsikan distribusi yang seragam terhadap plainteks. Pembuktiannya sebagai berikut :

Misalkan A adalah penyerangan, algoritma kriptosistem menerima dua input f dan g lalu menghasilkan output h, dengan suatu probabilitas p, dengan h = H(P) jika f = F(P) dan g = $G(E_K(P))$ sehingga $g = G(K \oplus P)$. Lalu, karena distribusi K adalah seragam dan tidak bergantung pada P, maka distribusi dari $K \oplus P$ juga pasti seragam dan tidak bergantung pada P. Karena itu, terdapat probabilitas p yang memenuhi h = H(P) jika f = F(P) dan g = G(C), untuk semua P dan C. Sehingga A adalah trivial.

Pada kasus yang lain, *one-time pad* tidak aman jika kunci K digunakan untuk lebih dari satu plainteks, sehingga terdapat banyak penyerangan cipherteks non-trivial. Maka agar *one-time pad* tetap aman untuk digunakan, kunci K haruslah dibuang setelah satu kali enkripsi. Kunci ini juga

disebut "*pad*". Ini menjelaskan nama dari kriptosistem ini.

4.2 *Ciphertext-only Attack*

Dengan menggunakan notasi di atas, suatu *ciphertext-only attack* adalah suatu penyerangan dengan F adalah konstanta. Diberikan hanya beberapa informasi $G(E_K(P_1),..E_K(P_n))$ tentang n cipherteks, penyerangan harus memiliki kesempatan menghasilkan beberapa informasi $H(P_1,\ldots,P_n)$ tentang plainteks. Penyerangan akan merupakan suatu trivial bila ia hanya menghasilkan $H(P_1,\ldots,P_n)$ ketika diberikan $G(C_1,\ldots,C_n)$ untuk $C_1,\ldots,C_n$ acak.

Sebagai contoh, misalkan G (C) = C dan misalkan H(P) adalah bit pertama P. Kita dapat secara mudah menulis suatu penyerangan, pendugaan, yang menduga bahwa H(P) adalah 1. Penyerangan ini adalah trivial karena tidak menggunakan cipherteks, probabilitas keberhasilannya adalah 50 %. Di lain pihak, terdapat penyerangan atas RSA yang memproduksi satu bit informasi tentang P, dengan probabilitas keberhasilan 100 %, menggunakan C. Jika diberikan suatu C acak maka tingkat kesuksesan turun menjadi 50%. Inilah yang disebut penyerangan nontrivial.

4.3 *Known-plaintext Attack*

Penyerangan *known-plaintext* klasik memiliki $F(P_1,P_2) = P1$, $G(C_1,C_2) = (C_1,C_2)$, dan $H(P_1,P_2)$ tergantung hanya pada P_2. Dengan kata lain, bila diberikan dua cipherteks C_1 dan C_2 dan satu dekripsi P_1, penyerangan known-plainteks seharusnya menghasilkan informasi tentang dekripsi P_2.

4.4 *Chosen-plaintext Attack*

Penyerangan dengan model *chosen-plaintext* memberikan kebebasan pada kriptanalis untuk memilih plainteks dan mengetahui cipherteks yang berkorespondensi dengannya, lalu mengulanginya sampai ia mengetahui bagaimana caranya mendekripsikan pesan manapun. Contoh yang lebih ekstrim dari penyerangan model ini adalah *chosen-key attack* dan *chosen-system attack*. Dari sini kita mengetahui bahwa *chosen-key attack*, yang termasuk ke dalam *basic cryptanalytic attacks* sebenarnya merupakan bagian dari *chosen-plaintext attacks*.

Perlu diingat bahwa, penyerangan dengan tipe ini merupakan penyerangan aktif, sehingga, seperti yang telah dijelaskan di atas, si penyerang dapat mengubah protokol yang sedang diserang. Hal penting dari bentuk penyerangan aktif adalah penyerangan dengan perubahan pesan (*message corruption attack*), di mana si penyerang mencoba untuk mengubah cipherteks melalui suatu cara untuk membuat perubahan yang bermanfaat pada plainteks.

Ada banyak cara mudah untuk mengacaukan penyerangan ini. Secara sederhana, mengenkripsikan plainteks P sebagai :

$$T, E_K(h(T \oplus R \oplus P), R, P)$$

Di mana T adalah *time-key* (rangkaian angka) yang dipilih mulai dari awal untuk setiap pesan, R adalah angka acak, dan h adalah fungsi hash satu arah. Di sini tanda koma berarti konkatenasi.

4.5 *Brute-force Attacks*

Umpamakan penyerangan *known-plaintext* berikut. Kita diberikan sejumlah plainteks $P_1,\ldots,P_{n-1}$ dan cipherteks $C_1,\ldots,C_{n-1}$. Kita juga diberikan sebuah cipherteks Cn. Kita jalankan seluruh kunci K. Bila kita temukan K sedemikian sehingga $E_K(P_1) = C_i$ untuk setiap I<n, kita cetak $D_K(C_n)$.

Jika n cukup besar sehingga hanya satu kunci yang bekerja, penyerangan ini akan sukses untuk seluruh input yang valid pada setiap waktu, sementara ia akan menghasilkan hasil yang tepat hanya sekali untuk input acak. Penyerangan ini adalah nontrivial, masalahnya ia sangat lambat bila terdapat banyak kemungkinan kunci.

4.6 *Key-guessing Attack*

Misalkan seseorang menggunakan *one-time pad*, tetapi tidak memilih kunci secara acak dan seragam dari seluruh pesan sepanjang m-bit, untuk membuktikan suatu keamanan kriptosistem. Pada kenyataannya, katakanlah diketahui bahwa ia akan memilih kunci yang merupakan suatu kata dalam Bahasa Inggris. Lalu seorang kriptanalis dapat menggunakan semua kata dalam Bahasa Inggris sebagai kemungkinan kunci. Penyerangan ini seringkali digunakan untuk menggantikan model penyerangan lainnya, dan penyerangan ini jauh lebih cepat dibandingkan dengan pencarian secara *brute-force*, pada seluruh ruang kunci.

4.7 Entropi

Kita dapat mengukur seberapa buruk sebuah distribusi kunci dengan menghitung entropi. E melambangkan banyaknya bit efektif penyimpan informasi kunci. Seorang kriptanalis biasanya melalui 2^E taksiran untuk menemukan kunci

yang tepat. E merupakan jumlah dari $-p_K \log_2 p_K$, di mana p_K adalah peluang kunci K.

DAFTAR PUSTAKA

[1] Answers.com, world's greatest encyclodictionalmanacapedia. (2006). Cryptographic Attacks. *http://answers.com*. Tanggal akses : 27 Desember 2006 pukul 1:15.

[2] BlinkBits. (2006). Rubber-hose Cryptanalysis. *http://www.blinkbits.com*. Tanggal akses : 27 Desember 2006 pukul 00:45.

[3] Cryptography FAQ. (2003). Mathematical Cryptology. *http://www.faqs.org/*. Tanggal akses : 27 Desember 2006 pukul 00:50.

[4] Felix, Fidens. (2006). Dasar Kriptografi. *http://IlmuKomputer.com*. Tanggal Akses : 24 Desember 2006 pukul 15.04.

[5] Gillogly, James. (1995). Ciphertext-only Cryptanalysis of Enigma. http://www.fortunecity.com/. Tanggal akses : 27 Desember 2006 pukul 00:33.

[6] Munir, Rinaldi. (2004). Diktat Kuliah IF2151 Matematika Diskrit Edisi Keempat. Departemen Teknik Informatika, Institut Teknologi Bandung.

[7] RSA Laboratories. (2004). CryptoFAQ. *http://rsasecurity.com*. Tanggal akses : 26 Desember 23.22.

[8] Scheneier, Bruce. (1996). Applied Cryptography 2nd. New Jersey : John Wiley & Sons, Inc.

[9] Tedi Heriyanto. (1999). Pengenalan Kriptografi. *http://tedi.hariyanto.net/papers/p_kripto.html*. Tanggal akses : 24 Desember 2006 pukul 14.42.

[10] Wikipedia, the free encyclopedia. (2006). Cryptographic Attacks. *http://en.wikipedia.org*. Tanggal akses : 27 Desember 2006 pukul 1:07.

[11] Wikipedia Indonesia, ensiklopedia bebas berbahasa Indonesia. (2006). Enkripsi. *http://id.wikipedia.org/wiki/Enkripsi.htm*. Tanggal akses : 24 Desember 2006 pukul 14.51.

[12] Wikipedia Indonesia, ensiklopedia bebas berbahasa Indonesia. (2006). Kriptografi. *http://id.wikipedia.org/wiki/Kriptografi.htm*. Tanggal akses : 24 Desember 2006 pukul 14.52.