Perbandingan Fungsi *Hash* Searah dengan Fungsi *Hash* Universal dalam Pengamanan Kriptosistem Kunci Publik dari Serangan Adaptif

Akhmad Ratriono Anggoro 13505003
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13505003@students.if.itb.ac.id

Makalah ini menyajikan metode untuk memperkuat algoritma kunci publik sedemikian rupa sehingga mereka menjadi aman terhadap serangan ciphertext adaptif. Dalam ciphertext terpilih, pihak penyerang dapat menerka-nerka algoritma dekripsi dengan ciphertexts sembarang yang diujikan, kecuali untuk objek yang hendak diserang dengan kriptoanalisis. Metode yang dimaksud adalah penggunaan fungsi *hash* universal. Metode yang diajukan disertai analisis berdasarkan kekuatan komputasi pada logaritma diskrit dalam bidang terbatas. Makalah ini juga membandingkan penggunaan fungsi hash satu arah dengan fungsi hash universal.

Kriptografi kunci public, Serangan adaptif, fungsi hash, sekuritas kunci publik

I. PENDAHULUAN

Selama lebih dari satu dekade, riset-riset secara teoritis [1] dan praktis [2] untuk mendapatkan efektifitas dan sekuritas yang terjamin dari penggunaan kriptografi kunci publik. Adapun jenis serangan yang sebenarnya sangat sulit untuk dibendung sampai saat ini adalah serangan adaptif. Dalam serangan adaptif, kriptoanalisis dilakukan dengan mencoba-coba cipherteks dan membandingkan hasilnya dengan plainteks yang dimiliki untuk menerka algoritma dari sebuah dekripsi dengan kunci publik.

Walaupun riset-riset tersebut banyak menghasilkan solusi, ekspansi dan ukuran dari cipherteks yang tidak wajar sehingga terlalu mahal untuk dijadikan solusi. Bahkan, untuk algoritma kunci publik yang memanfaatkan teknik homomorfis, hasil cipherteks yang dihasilkan bisa membengkak sampai puluhan kali [3].

Sebagai gambaran singkat mengenai serangan adaptif, mari kita lihat contoh klasik pertukaran informasi antara Bob dan Alice. Anggap Bob mengirimkan pesan terenkripsi *x* kepada Alice menggunakan kunci publik milik Alice. Dalam kondisi normal plainteks pesan *x* hanya dapat dibaca oleh Alice namun semua orang mempunyai kemungkinan untuk mengetahui cipherteks yang dikirim Bob lewat jaringan yang tidak aman. Misalkan Cathy, sebagai rival bisnis Bob, ingin mengetahui isi pesan *x*, Ia dapat melakukannya dengan sedikit percakapan dan diskusi dengan Alice. Walaupun Cathy tidak mungkin dapat mengirim pesan *x* seutuhnya karena akan menimbulkan kecurigaan besar padanya karena Bob telah mengirimkan pesan yang sama. Tetapi, Cathy dapat mengirim pesan *x* dalam *n* bagian dengan harapan mendapatkan pesan balasan yang sesuai untuk nantinya dianalisis dan diserang. Bayangkan seandainya Bob adalah nasabah, Alice adalah bank, dan Cathy adalah cracker. Maka data rekening Bob dapat diubah sesuka hati oleh Cathy dengan analisis bertahap terhadap mekanisme pengiriman perintah dari Bob ke Alice.

Makalah ini menjelaskan metode pragmatis dengan menggunakan fungsi hash universal disertai dengan contoh kriptografi kunci publik. Selain itu, perbandingan terhadap teknik *hash* satu arah juga dilakukan.

Pada bab selanjutnya, akan dijelaskan mengenai tipe-tipe serangan pada kriptografi kunci publik berdasarkan kekuatan komputasi logaritma diskrit dan definisi formal mengenai sekuritas kriptografi kunci publik. Bab III akan menjelaskan teknik fungsi *hash* universal yang dipakai untuk mengamankan kriptografi kunci publik. Bab IV akan membahas analisis pada metode yang diajukan beserta perbandingannya terhadap fungsi *hash* satu arah dan Bab V berisi kesimpulan.

II. GAGASAN DAN NOTASI-NOTASI YANG DIGUNAKAN

Ambil alphabet $\sum = \{0, 1\}$ sebagai contoh himpunan alphabet yang akan banyak digunakan dalam makalah ini dengan panjang string yang dilambangkan dengan $|x|$. Konkatenasi dari dua string *x* dan *y* akan dilambangkan dengan $x\|y$ sementara bit eksklusif antara *x* dan *y* dilambangkan dengan $x \oplus y$. Bit ke i pada *x* akan dilambangkan dengan x_i dan untuk $i \leq j$, substring dari *x* dari i sampai j dilambangkan dengan $x_{[i..j]}$. #K melambangkan jumlah elemen dalam himpunan K dan *x*

$\in_R$ K berarti *x* elemen acak dan seragam dari K. Produk kartesian dari K dan L dilambangkan dalam bentuk K x L.

N melambangakan himpunan integer positif sementara *n* mewakili panjang pesan, panjang cipherteks, ataupun sebagai parameter keamanan. Seperti pada skema El-Gamal [4] *p* melambangkan bilangan prima dengan *n*-bit dan *g* adalah pembangkit grup multiplikasi GF(*p*)* dalam sebuah bidang terbatas GF(*p*). *p* dan *g* bersifat publik. Untuk menjamin keamanan dari serangan biasa, hendaknya *n* dari *p* lebih besar dari 512 bit dan *p*-1 harus memiliki setidaknya sebuah faktor prima besar [5].

Perlu diperhatikan bahwa terdapat korespondensi satu-ke-satu antara string di $\sum^n$ and elemen dalam bidang GF(*2n*). Secara alami, juga terdapat korespondensi yang sama antara $\sum^n$ dengan integer dalam [0, $\mathbf{2}^{\boldsymbol{n}}$ - l]. Karena itu, makalah ini tidak akan membedakan antara string dalam $\sum^n$, elemen dalam GF(*2n*) dan integer dalam [0, $\mathbf{2}^{\boldsymbol{n}}$ - l].

Kriptosistem dengan kunci publik, yang ditemukan oleh Diffie dan Helman, terdiri dari tiga algoritma polinomial. Pertama yaitu algoritma pembangkit kunci, dengan input *n* menghasilkan sepasang (*pk, sk*) berturut-turut adalah kunci publik dan kunci privat. Algoritma kedua adalah algoritma enkripsi, yang membangkitkan sebuah pesan rahasia *c* dari masukan *pk* dan plainteks *m.* Algoritma yang terakhir adalah algoritma dekripsi, yang dengan masukan pesan rahasia *c* dan kunci *sk*, dapat menghasilkan keluaran plainteks *m*. Algoritma enkripsi dan dekripsi memenuhi persamaan

$$\boldsymbol{D(sk, E(pk,}\ \mathrm{m})) = \boldsymbol{m} \quad (1).$$

A. SERANGAN-SERANGAN TERHADAP KRIPTOSISTEM

Terdapat lima tipe dasar serangan analitik terhadap kriptosistem, yaitu, ***ciphertext only attacks, known plaintext attacks, chosen plaintext attacks, chosen ciphertext attacks, Adaptive-chosen attack,***

Pada *ciphertext only attacks*, kriptanalis memiliki beberapa cipherteks dari beberapa pesan, semuanya dienkripsi dengan algoritma yang sama. Tugas kriptanalisis adalah menemukan plainteks sebanyak mungkin atau menemukan kunci yang digunakan untuk mengenkripsi pesan.

Diberikan: C1 = Ek(P1), C2 = Ek(P2), …, Ci = Ek(Pi)
Deduksi: P1, P2, …, Pi atau k untuk mendapatkan
Pi+1 dari Ci+1 = Ek(Pi+1).

Pada *known plaintext attacks* kriptanalis memiliki beberapa pasangan plainteks dan cipherteks yang dapat ia gunakan untuk mengenali dan memahami kriptosistem yang digunakan.

Pada *chosen plaintext attacks* penyerang memiliki akses ke algoritma enkripsi sehingga dapat dengan mudah memanipulasi plainteks apapun untuk mengirim pesan palsu kepada pemilik kunci publik. Serangan ini umum terjadi karena kunci publik dapat diketahui semua orang.

Chosen plaintext attacks, penyerang memiliki akses ke algoritma dekripsi, sehingga plainteks apapun dapat diterjemahkan oleh penyerang berdasarkan pengetahuan yang diperolehnya dalam mendekripsi cipherteks yang dapat dia miliki.

Sementara yang terakhir, bisa dilakukan pada plainteks maupun cipherteks. Namun untuk kasus ini, lebih terfokus pada penggunaan serangan adaptif terhadap cipherteks pada kriptosistem dengan kunci publik. Serangan ini dilakukan dengan memberi cipherteks-cipherteks selain dari obyek yang hendak diserang ke dalam mesin dekripsi. Cipherteks-cipherteks tadi bisa saja memiliki korelasi satu dengan yang lainnya[10].

B. GAGASAN MENGENAI KEAMANAN

Sebuah kriptosistem dapat dikatakan aman bilamana penyerang tidak dapat memulihkan seluruh plainteks dari cipherteks yang disadapnya. Berdasarkan prinsip keamanan semantik, hal ini dapat diaplikasikan dalam bentuk polinomial terikat dari bentuk yang diajukan Shannon (*perfect secrecy [5]*). Keamanan semantik juga berarti penyerang hanya memperoleh rahasia dari kriptosistem yang dapat dihitung tanpa memiliki satupun cipherteks.

Definisi dari keamanan semantik pada kriptosistem berbasis kunci publik dapat direpresentasikan dalam dua bentuk mesin Turing yang berbasis pada probabilitas polinomial; sebuah mesin pengoleksi, dan ekstraktor informasi parsial.

Kolektor berfungsi pada tahap awal dari kriptanalisis, yaitu memperoleh informasi-informasi yang dapat digunakan untuk tahap berikutnya. Adapun kolektor spesifik yang dapat diterapkan untuk tipe serangan adaptif adalah sebuah mesin £ yang memiliki input *n*, *pk*, dan sebuah obyek cipherteks. Mesin £ memiliki akses pada algoritma dekripsi dan berusaha menerka algoritmanya dengan mencoba mendekripsikan banyak cipherteks selain dari obyek yang hendak diteliti.

Ekstraktor informasi parsial adalah sebuah mesin Ŧ yang mewakili tahap kedua dari kriptanalisis, yaitu menerjemahkan obyek cipherteks dan memperoleh informasi berguna dari cipherteks tersebut. Ŧ memiliki masukan *n, pk,* dan obyek cipherteks yang hendak dipecahkan.

Definisi 1: Ambil (C, E, D) sebagai sebuah kriptosistem kunci publik. $M_n = \sum^P$ adalah ruang pesan yang diinduksikan oleh parameter keamanan *n*, dimana P

adalah polinomial dalam *n*. Asumsikan plainteks *m* dapat diperoleh dari M_n dengan kemungkinan p(*m*). Ambil V sebagai himpunan sembarang dan f_n^{pk} adalah fungsi yang memetakan M_n ke V, dengan *pk* sebagai kunci publik yang diterka secara probabilistik dari fungsi C dengan masukan *n*. $P(f_n^{pk})$ adalah probabilitas maksimum seseorang dapat menerka kunci publik tanpa mengetahui masukan *m* sama sekali.

$$P(f_n^{pk}) = \max_{v \in V} \{\Sigma_{m \in pre[f_n^{pk}(v)]} p(m)\} \quad (2)$$

Kriptosistem (C, E, D) dapat dikatakan aman secara semantik dari serangan adaptif cipherteks apabila untuk setiap kolektor cipherteks £, untuk tiap ekstraktor informasi parsial Ŧ, untuk tiap polinomial Q = Q(*n*), dan untuk tiap bilangan *n* yang besar memenuhi,

$$Pr\{Ŧ(l^n, pk, C) = f_n^{pk}\} < P(f_n^{pk}) + 1/Q \quad (3)$$

Adapun gagasan yang ekuivalen dengan keamanan semantik adalah keamanan polinom. Sebuah kriptosistem dapat dikatakan aman secara polinom apabila tidak terdapat satupun fungsi probabilistik polinom yang dapat membandingkan cipherteks yang dihasilkan oleh dua plainteks berbeda.

III. MENGGUNAKAN FUNGSI *HASH* UNIVERSAL UNTUK MENGAMANKAN KRIPTOSISTEM KUNCI PUBLIK

Dasar pengamanan dengan menggunakan fungsi *hash* universal adalah sama seperti dengan menggunakan fungsi *hash* searah. Pengamanan ini memperkuat cipherteks dengan menambahkan properti seragam untuk setiap enkripsi. Bedanya, pada fungsi *hash* searah, properti diaplikasikan dengan fungsi *hash* yang tidak memiliki invers, sementara metode universal *hash* memilih sebuah fungsi dari kelas universal dari fungsi-fungsi *hash* yang tersedia.

Metode ini dapat akan diilustrasikan dalam skema kunci publik ElGamal. Ambil G sebagai string *pseudorandom* yang kuat secara kriptografis berdasarkan kemampuan komputasi algoritma diskrit pada ruang terbatas. G membentangkan masukan *n*-bit string menjadi sebuah keluaran yang panjangnya polinomial dalam *n*. Kasarnya, G menghasilkan keluaran O(log *n*) bit untuk setiap pemangkatan.

Misal kunci privat Alice adalah elemn x_R yang dipilih secara acak dari [1, *p - 1*], dan kunci publiknya adalah $y_R = g^{xR}$. Dapat diasumsikan bahwa semua pesan yang akan dienkripsi diperoleh dari himpunan $\sum^P$ dan P = *P(n)* memenuhi *P(n)* >= *n*. Untuk pesan yang panjangnya kurang dari *n* bit, teknik *padding* dapat diterapkan.

Sebagai tambahan, ambil *l* = *l* (*n*) sebagai polinom yang mewakili panjang tag pada pesan. Untuk alasan keamanan panjang *l* hendaknya lebih besar dari 64bit.

A. APLIKASI KUNCI HASH UNIVERSAL

Kelas H yang terdiri dari fungsi-fungsi dari $\sum^P$ sampai $\sum^l$ disebut sebagai kelas fungsi *hash* universal [6], yang memetakan masukan sebesar P-bit menjadi l-bit output jika, untuk setiap $x_1 \neq x_2 \in \sum^P$ dan setiap $y_1, y_2 \in \sum^l$, jumlah fungsi yang mengkonversikan x_1 menjadi y_1; dan x_2 menjadi y_2 adalah $\#H/2^{2l}$. Sekarang asumsikan H adalah kelas universal yang memetakan masukan sebesar P-bit menjadi *l*-bit output, Q = Q(*n*) adalah polinom, dan setiap fungsi dalam H dapat diwakili oleh string dengan panjang Q-bit. Ambil h_t sebagai fungsi pada H yang diwakili oleh string $t \in \sum^Q$, maka algoritma yang digunakan Bob untuk mengirim pesan rahasia sepanjang P-bit dengan kunci publik y_R pada Alice adalah sebagai berikut,

Algoritma 1: $E_{uhf}(y_R, p, g, m)$

1. $x \in [1, P - 1]$.
2. $r = y_R^x$
3. $z = G(r)_{[1..P]}$
4. $s = G(r)_{[(P+1)..(P+Q)]}$
5. $c_1 = g^x$
6. $c_2 = h_t$
7. $c_3 = z \oplus m$

Output yang dihasilkan (c_1, c_2, c_3). Sedangkan untuk dekripsinya. Alice yang memiliki kunci privat x_R akan menggunakan algoritma sebagai berikut,

Algoritma 2: $D_{uhf}(x_R, p, g, c_1, c_2, c_3)$

1. $r' = c_1^{xR}$
2. $z' = G(r')_{[1, P-1]}$
3. $s' = G(r')_{[(P+1)..(P+Q)]}$
4. $m' = z \oplus c_3$
5. if $h_{t'}(m')$ *then output m, else output {}.*

Berikut ini adalah contoh sederhana kelas universal dari fungsi-fungsi hash yang berasal dari kekongruenan linear dalam bidang terbatas(lihat []). Ambil sembarang integer *k*. Untuk tiap elemen *k* + 1 $a_1, a_2, a_3, ..., a_k, b \in GF(2^l)$, ambil *t* sebagai konkatenasi dari elemen k + 1 tersebut sehingga $t = a_1||a_2||a_3||\ldots||a_k||b$, dan h_t sebagai fungsi-fungsi yang didefinisikan oleh,

$$ht(x_1,x_2,\ldots,x_k) = \sum_{i=1}^{k} a_i x_i + b \quad (4)$$

Kelas H dari fungsi h_t, yang dapat didefinisikan oleh semua elemen k + 1 dari GF(2^l) adalah kelas universal dari fungsi hash universal.

Fungsi-fungsi dalam H menkompresi masukan sebesar kl-bit menjadi l-bit string. Dengan melakukan teknik padding pada string masukan, fungsi-fungsi ini dapat diaplikasikan ke masukan yang panjangnya bukan *kl*. Kasus khusus, ketika k = [P / l], H dapat digunakan untuk mengkompresi masukan sebesar P-bit menjadi keluaran sebesar l-bit. Fungsi dalam H pada kasus ini dapat diwakili oleh sebuah string dari Q = P + (1 + α)l bits, yang memenuhi $0 \leq \alpha = P \bmod l / l < 1$. Kelas universal dari fungsi-fungsi hash sangat cocok pada kasus ketika panjang pesan P akan dienkripsi jauh lebih besar dari dari panjang tag l.

IV. ANALISIS DAN PERBANDINGAN

A. SEKILAS MENGENAI FUNGSI HASH SEARAH

Sama seperti fungsi hash universal, fungsi hash searah digunakan untuk mentransformasikan masukan string searah menjadi string keluaran yang panjangnya tetap. Dinamakan fungsi hash satu arah karena keluaran fungsi ini tidak bisa dibalik. Untuk setiap h yang dihasilkan, tidak mungkin dikembalikan nilai x sedemikian sehingga H(x) = h. Itulah sebabnya fungsi H dikatakan fungsi hash satu-arah (one-way hash function).

Adapun algoritma formal yang digunakan dalam fungsi hash searah adalah sebagai berikut. Asumsikan h adalah sebuah fungsi hash searah yang mengkompresikan masukan string menjadi keluaran sebesar l-bit. Jadi apabila Bob hendak mengirimkan pesan rahasia m sebesar P-bit kepada Alice.

Algoritma 1: $E_{uhf}(y_R, p, g, m)$

1. $x \in_R [1, P - 1]$.
2. $z = G(y^x_R)_{[1..(P + l)]}$
3. $t = h(m)$
4. $c_1 = g^x$
5. $c_2 = z \oplus (m||t)$

Output yang dihasilkan (c_1, c_2). t adalah tag yang dipakai pada keluaran. Sedangkan untuk dekripsinya. Alice yang memiliki kunci privat x_R akan menggunakan algoritma sebagai berikut,

Algoritma 2: $D_{uhf}(x_R, p, g, c_1, c_2)$

1. $z' = G(c_1^{xR})_{[1.. (P + l)]}$
2. $w = z' \oplus c_2$
3. $m' = w_{[1..P]}$.
4. $t' = w_{[(P + 1)..(P + l)]}$
5. if $h(m')= t'$ *then output* (m'), *else output* $\{\}$.

Ketika pesan tepat sebesar n bit, sehingga P = n, fungsi eksponen dapat dapat digunakan untuk membangkitkan tag t. Model enkripsi juga dapat diubah sehingga langkah 2 dari proses enkripsi dapat menjadi $z = G(y^x_R)_{[1..2n]}$ dan langkah 3 diubah menjadi $c_1 = g^m$.

B. KEAMANAN MENGHADAPI SERANGAN ADAPTIF PADA CIPHERTEKS

Iingat bahwa output dari algoritma enkripsi untuk kriptosistem dengan hash satu arah adalah (c_1, c_2), dimana $c_1 = g^x$, $c_2 = z \oplus (m||t)$, $t = h(m)$, dan h adalah fungsi hash satu arah. Algoritma enkripsi tersebut mendefinisikan fungsi yang memetakan elemen (x, m) dari [1, P - 1] x $\sum^P$ ke elemen (c_1, c_2 dalam [1, P - 1] x $\sum^{P+l}$). Karena terdapat keterlibatan fungsi hash $h(m)$, cipherteks tidak mungkin direkayasa tanpa pengetahuan akan x dan m. Observasi serupa tentu sama pada kriptosistem dengan fungsi hash universal. Fakta ini memotivasi Damgard untuk menggunakan gagasan yang bernama *sole-samplable space* untuk menguji keamanan kriptografi kunci publiknya[7].

ambil f sebagai fungsi dari $D = U_nD_n$ ke $R = U_nR_n$, dengan $D_n \subseteq \sum^n$, $R_n \subseteq \sum^{Q1}$ dan $Q_1 = Q_1(n)$ adalah polinom. $R = U_nR_n$ adalah bidang yang diinduksi oleh fungsi f. Secara informal,$R_n = U_nR_n$ dapat dikatan sebagai sole-samplable ketikat tidak ada cara lain untuk membangkitkan sebuah elemen y di R_n selain dari memilih elemen x di D_n terlebih dahulu dan mengevaluasi fungsi pada poin x. Untuk mendefinisikan sole-samplability secara formal, dua tipe mesin Turing: *pembangkit sampel* dan *ekstraktor persepsi awal* harus digunakan.

Pembangkit sampel dalam bidang $R = U_nR_n$, dilakukan dengan fungsi f sebagai mesin Turing probabilisting polinom S sehingga bila terdapat n sebagai masukan dan akses ke Oracle O_R pada bidang R, keluarannya adalah Q_1-bit string. Oracle kemudan mencetak satu sampel string y elmnt R_n sebagai jawaban dari request $n \in \mathbf{N}$. S dapat memberikan query ke oracle hanya dengan menulis $n \in \mathbf{N}$ pada pita khusus dan akan membaca jawaban dari oracle $y \in R_n$ pada pita yang terpisah.

Ekstraktor persepsi dari sebuah pembangkit sample S adalah mesin turing probabilistik polinom S yang mempunyai akses ke semua pita milik S dan dapat mengamati semua komputasi dalam S. Masukan X adalah integer $n \in \mathbf{N}$ dan keluaran X adalah n-bit string. Adapun kedua mesin turing ini didefinisikan berdasarkan gagasan keamanan semantik yang disebutkan sebelumnya.

Definisi 2: ambil f sebagai fungsi dari $D = U_nD_n$ ke $R = U_nR_n$. $D_n \subseteq \sum^n$, $R_n \subseteq \sum^{Q1}$ dan $Q_1 = Q_1(n)$ adalah polinom. ruang $R = U_nR_n$ diinduksikan dari fungsi f yang *sole-sampable* bila, untuk tiap pembangkit sampel S dan untuk sembarang polinom $Q_2 = Q_2(n)$, terdapat ekstraktor persepsi X untuk pembangkit sampel S sehingga untuk semua bilangan n besar, berlaku

$$\Pr\{X(1^n, S)\} \geq 1 - 1/Q_2 \quad (5)$$

$\Pr\{X(1^n, S)\}$ adalah probabilitas probabilitas ketika output S adalah contoh y dari R_n yang berbeda dengan yang diberikan oleh oracle O_R, X menghasilkan sebuah string x elmt Dn sehingga $y = f(x)$.

Perlu diperhatikan bahwa fungsi f bukan fungsi satu arah sehingga terdapat fungsi inverse f^{-1} dari f yang dapat dihitung dalam probabilistik polinom, bidang R yang diinduksi oleh f adalah sebuah sole sampable, karena semua bisa menghitung preimage x emnt Dn dari sebuah elemen $y \in R_n$, implikasinya adalah terdapat satu cara untuk memilih R_n, yaitu mengambil x terlebih dahulu lalu menghitung $y = f(x)$.

KOndisi yang dibutuhkan untuk bidang $R = U_n R_n$, yang diberikan oleh fungsi satu arah f sebagai contoh sederhana adalah R harus sparse, dalam artian $\#R_n/2 \ Q_1 < 1/Q_2$ untuk setiap polinomial $Q_2 = Q_2(n)$ dan untuk setiap n. Walaupun sparseness bukan kondisi mutlak yang dibutuhkan untuk sebuah sole sampable namun ini adalah properti yang dapat ditemukan dalam hashing. Asumsi berikut akan digunakan untuk membandingkan keamanan antara dua kriptosistem yang dibahas di sini. Asumsi ini berkaitan dengan sole-sampability dari bidang yang dibangkitkan oleh fungsi dari algoritma enkripsi yang dijelaskan di bab sebelumnya. Asumsi-asumsi ini dibuat berdasarkan fakta pemampatan pesan dalam panjang yang tetap pada pembuatan cipherteks dari plainteks.

Asumsi 1: bidang yang dibentuk oleh algoritma enkripsi yang memanfaatkan fungsi hash searah memenuhi kriteria sole samplability

asumsi 2: bidang yang dibentuk algoritma enkripsi yang memanfaatkan fungsi hash universal memenuhi kriteria sole samplability

Apabila kedua asumsi ini dapat diterapkan, sesuai dengan sifat sole-sampability yang telah dijelaskan dalam bab ini, maka keduanya sebenarnya memiliki tingkat keamanan yang kurang lebih sama untuk pengamanan terhadap serangan adaptif dalam kondisi ideal.

Kembali ke mesin Turing yang digunakan untuk melakukan serangan adaptif (Ŧ, £). Fakta tersebut akan dimanfaatkan untuk membuktikan hashing akan membuat sebuah kriptosistem dengan kunci publik menjadi aman secara semantik. Misalkan Ŧ telah mendapatkan akses ke algoritma dekripsi dan melakukan query sebanyak $Q = Q(n)$, masing-masing dengan menggunakan cipherteks berbeda. Anggap cipherteks pertama yang dipakai adalah c_1. karena bidang yang dihasilkan dari proses enkripsi bersifat sole samplable, gambaran awal terhadap c_1, yang merupakan bagian dari plainteks m_1, dapat dihitung dalam probabilitas polinomial dari data yang diperolah pada komputasi yang dilakukan Ŧ. Dengan kata lain, berusaha mendapatkan respon yang bermanfaat dengan query c_1 pada mesin dekripsi tidak akan menghasilkan hal-hal baru yang dapat dipelajari. Hal ini berlaku juga untuk query cipherteks c_2, c_3,c_4, ..., c_Q. Karena a sudah tidak berguna, kita dapat menganggap kolektor pada serangan ini tidak memiliki akses terhadap mesin dekripsi. Hal ini dapat kita representasikan dengan (Ŧ', £). Perhatikan bahwa input pada a' masih sama seperti Ŧ, yaitu n, kunci publik pk, dan obyek cipherteks yang hendak dianalisis. sementara masukan £ terdiri dari n, pk, obyek cipherteks dan keluaran yang diberikan oleh Ŧ '. Perbedaan utama yang dapat dilihat disini adalah Ŧ' memiliki obyek cipherteks yang hendak dianalisis. Memanfaatkan semua properti yang didiskusikan di atas kalau Ŧ' hanya memberikan informasi yang sebenarnya bisa diperoleh tanpa mengetahui n dan pk, keberadaan £ sendiri bisa dihilangkan. Induksi ini membuktikan, menghadapi serangan adaptif terhadap kekuatan komputasi logaritma diskrit pada bidang yang sole-sampable sangat sulit dilakukan.

Walaupun kedua hashing memiliki potensi kekebalan yang sama dalam menangani serangan cipherteks adaptif, yang membedakan adalah kolisi yang mungkin terjadi dari proses hashing. Kolisi adalah sebuah kasus (yang biasanya langka) ketika dua plainteks memiliki dua hashing value yang sama. Ini sangat berbahaya dan pada kasus-kasus tertentu dapat digunakan untuk menyerang algoritma sehingga dapat menyebabkan kerugian parah seperti data loss dan mampu digunakan untuk menyerang sebuah kriptosistem dalam bentuk *collision attack*[8].

Perhitungan kasar, karena universal hash menggunakan pengambilan secara acak sebuah fungsi hash dari kelas hash universal yang memiliki N fungsi, kemungkinan kolisi dapat diperkecil sedemikian rupa sehingga probabilitas untuk terjadinya kolisi pada fungsi hash universal lebih sedikit N kali dibandingkan dengan fungsi hash satu arah.

V. KESIMPULAN

Metode yang telah dijelaskan dan dibandingkan pada makalah ini berguna untuk menangkal serangan cipherteks adaptif. Gagasan dari sole-samplability dijelaskan secara formal berdasarkan yang dilakuikan oleh Damgard. Adapun properti hashing yang bermanfaat untuk pengamanan sistem adalah memperkuat kriptosistem memanfaatkan sole samplability dan kekuatan komputer dalam menghitung logaritma diskrit dalam bidang yang terbatas. Universal hash lebih aman dibandingkan dengan fungsi hash satu arah, karena kemungkinan kolisi lebih kecil.

REFERENSI

[1] **M.** Blum, P. Feldman, and **S.** Micali, “Non-interactive zero-knowledge proof systems and applications,” in ***Proc. 20th Annu. ACM Symp. Theory Computing.*** 1988, pp. **103-1** 12

[2] I. Damgird, “Towards practical public key systems secure against chosen ciphertext attacks,” in ***Advances in Cryptology-froceedings of Crypto’91,*** Lecture Notes in Computer Science, vol. 576, J. Feigenbaum, Ed. New York: Springer-Verlag. 1992, pp. 445-456.

[3] AR, Anggoro, “Studi Mengenai *Fully Homomorphic Encryption* dan Perkembangannya dari RSA sebagai Enkripsi Homomorfis Populer,” Institut Teknologi Bandung, Teknik Informatika, 2010.

[4] T. EIGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,’’ ***IEEE Trans. Inform. Theory,*** vol. IT-31, pp. 469472, 1985.

[5] B. A. LaMacchia and **A.** M. Odlyzko, “Computation of **discrete** logarithms in prime fields,” ***Designs, Codes Cryptography,*** vol. 1, pp. 47-62, 1991.

[6] Y. Zheng and J. Seberry, ” **Immunizing Public Key Cryptosystems Against Chosen Ciphertext Attacks**,” IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 1 I , NO. 5, JUNE 1993

[7] D. R. Stinson, "Combinatorial techniques for universal hashing," Rep. Series 127, Dep. Comput. Sci., Univ. Nebraska, Lincoln, Nov. 1990. (Also submitted to J. Comput. Syst. Sci.), accessed April 27, 2010

[8] www.apa.com Alexander Sotirov, Marc Stevens, Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, Benne de Weger, ”**MD5 considered harmful today: Creating a rogue CA certificate**”, accessed March 29, 2010

[9] Mihir Bellare, Anand Desai, David Pointcheval, and Philip Rogaway, ”**Relations among Notions of Security for Public-Key Encryption Schemes, in Advances in Cryptology**” -- CRYPTO '98, Santa Barbara, California, pp. 549-570.

”

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, 29 April 2010

ttd

Akhmad Ratriono Anggoro 13505003