Keyboard Acoustic Emanations of Acoustic Cryptanalysis

Anita Fauziah Rahmat, Izzatul Ummah, Tina Lusiana
Sekolah Tinggi Elektronika dan Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132
E-mail : {if1208,if12014,if12048}@students.if.itb.ac.id

Abstrak

Acoustic Cryptanalysis merupakan serangan *side-channel* yang mengeksploitasi suara baik yang terdengar maupun tidak yang dihasilkan selama komputasi atau operasi *input-output*. Pada tahun 2004, Dmitri Asonov dan Rakesh Agrawal mempublikasikan bahwa tombol pada *keyboard* dan pada telepon serta mesin ATM sangat berpotensi untuk diserang dengan membedakan suara yang dihasilkan oleh tombol yang berbeda. Serangan ini didasarkan pada hipotesis bahwa suara klik diantara tombol-tombol adalah sedikit berbeda, meskipun terdengar sama dalam pendengaran manusia. Serangan ini tidak mahal dan tidak *invasive*. Fase pertama dalam serangan ialah melatih *recognizer* melalui 4 tahapan yaitu ekstraksi fitur, *unsupervised key recognition*, s*pelling and grammar checking*, dan *feedback-based training*. Untuk dapat bertahan dari serangan, keamanan fisik ruangan dan mesin harus terjamin serta harus dipastikan bahwa suara tidak dapat terekam dari luar ruangan. Penggunaan *keyboard* yang tidak ribut dan memiliki suara tombol yang seragam (sehingga sulit dibedakan) juga dapat mengurangi kemungkinan penyerangan.

Kata kunci : acoustic, *cryptanalysis, keyboard*, *side-channel*

1. Pendahuluan

Secara umum, kriptanalisis memiliki pengertian sebagai sebuah studi *chiper*, chiperteks, atau *cryptosystem* yang berusaha menyembunyikan sistem kode dengan meneliti untuk menemukan kelemahan pada sistem yang akan memungkinkan sebuah plainteks diungkap dari chiperteksnya tanpa perlu mengetahui kunci algoritma. Singkatnya, kriptanalisis berusaha memecah *cipher*, cipherteks atau *cryptosystem*.

Terdapat berbagai metode penyadapan data untuk kriptanalisis yang telah dikembangkan, yakni :

a. *Wiretapping:* Penyadap mencegat data yang ditransmisikan pada saluran kabel komunikasi dengan menggunakan sambungan perangkat keras.
b. *Electromagnetic Eavesdropping*: Penyadap mencegat data yang ditransmisikan melalui saluran wireless, misalnya radio dan *microwave*.
c. *Acoustic Eavesdropping:* Menangkap gelombang suara yang dihasilkan oleh sistem maupun suara manusia.

Dua metode terakhir memanfaatkan kebocoran informasi dalam proses transmisi yang seringkali tidak disadari dengan menggunakan teknik analisis yang sering dikenal dengan *side channel cryptanalysis*.

Teknik analisis *side channel* merupakan *tool* yang *powerful* dan mampu mengalahkan implementasi algoritma yang sangat kuat karena mengintegrasikan kepakaran istem yang sangat tinggi. Media serangan yang sering digunakan yakni :

a. *Electromagnetic Leakage*: memanfaatkan radiasi elektromagnetik yang ditangkap dengan antena
b. *Timing Attack*: serangan didasarkan pada pengukuran waktu respon sistem untuk mengurangi kemungkinan pengujian dalam menentukan *password*.
c. *Thermal Analysis*: menggunakan difusi panas yang dihasilkan *processor* untuk mengetahui aktivitas spesifik sistem dan memanfaatkan perubahan temperatur pada media *storage*.
d. *Power Analysis*: mengukur perbedaan penggunaan energi dalam periode waktu tertentu ketika sebuah *microchip* memproses sebuah fungsi untuk mengamankan informasi. Teknik ini dapat menghasilkan informasi mengenai komputasi kunci yang digunakan dalam algoritma enkripsi dan fungsi keamanan lainnya.
e. *Sound Attack*: mengeksploitasi suara/ bunyi yang dihasilkan sistem.

Salah satu implementasi serangan pada media terakhir adalah *Acoustic Cryptanalysis*, yang merupakan serangan *side-channel* yang mengeksploitasi suara baik yang terdengar maupun tidak yang dihasilkan selama komputasi atau operasi *input-output*. Pada tahun 2004, Dmitri Asonov dan Rakesh Agrawal mempublikasikan bahwa tombol pada *keyboard* dan pada telepon serta mesin ATM sangat berpotensi untuk diserang dengan membedakan suara yang dihasilkan oleh tombol yang berbeda.
Kebanyakan sumber suara pada *keyboard* adalah tidak seragam pada jenis yang berbeda, bahkan pada model yang sama. Keyboard yang sama ataupun berbeda yang diketikkan oleh orang yang berbeda dapat menghasilkan suara yang berbeda dan hal ini mempersulit pengenalan tombol.

Serangan banyak dilakukan pada keyboard PC. Serangan ini didasarkan pada hipotesis awal bahwa suara klik diantara tombol-tombol adalah sedikit berbeda, meskipun suara klik diantara tombol yang berbeda terdengar sama dalam pendengaran manusia. Serangan ini tidak mahal karena perangkat keras yang digunakan hanyalah mikrofon *parabolic* dan tidak *invasive* karena tidak memerlukan intrusi fisik ke dalam sistem.

Dalam menganalisis sebuah teks atau kata yang diketikkan, sejumlah kata dalam bahasa yang terbatas akan membatasi kombinasi kata yang mungkin. Pertama-tama, pengelompokkan *keystroke* dengan metode *unsupervised learning* menjadi sejumlah class didasarkan pada suaranya dapat dilakukan. Dengan sampel training yang cukup, pemetaan antara class dan karakter yang sesungguhnya diketik dapat ditentukan dengan memanfaatkan konstrain bahasa.

Hal ini tidak sepele, karena terdapat tantangan berikut :

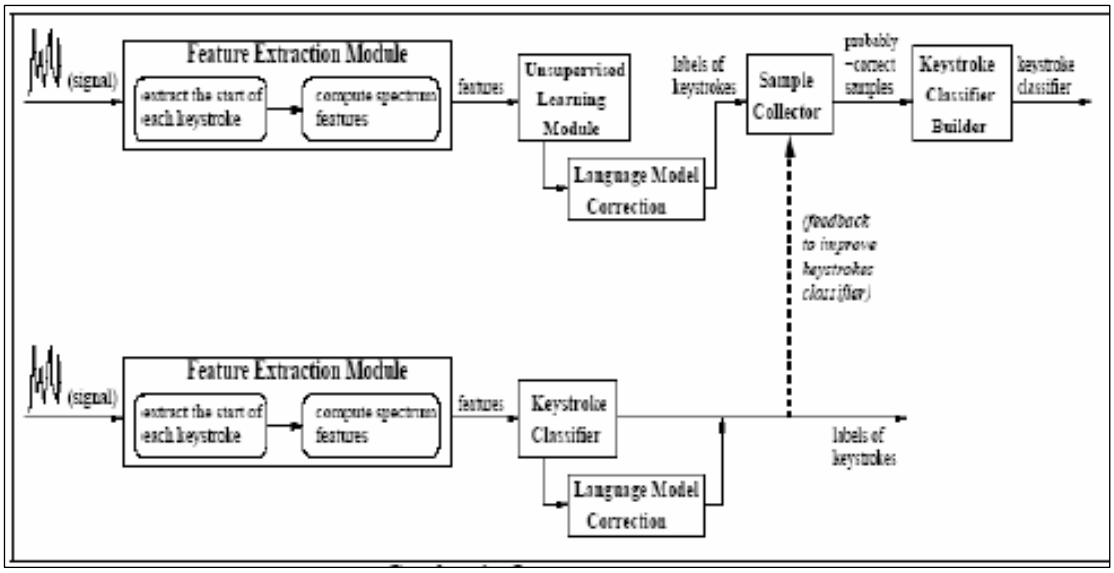
a. Bagaimana pemetaan antara konstrain bahasa dilakukan secara matematis dan bagaimana menerapkannya secara teknis?
b. Pada tahapan awal pengelompokkan berbasis suara, bagaimana mengalamatkan masalah pada banyak tombol yang dikelompokkan dalam kelas yang sama dan pada sebuah tombol yang dikelompokkan dalam banyak kelas?
c. Dapatkah pengembangan akurasi penebakan dengan algoritma pencocokan yang mencapai level sampel dikembangkan?

Sebuah kombinasi antara teknik pembelajaran mesin (*machine learning*) dan *speech recognition* dapat dilakukan untuk menunjukkan bagaimana pengenal tombol dapat memiliki tingkat pengenalan yang lebih baik dari sampel pengenal tombol yang telah diuji hanya dengan merekam suara pengetikan pengguna. Metode ini merupakan serangan klasik pada *chiper* substitusi dengan *machine learning*. Masalah utama yang muncul adalah berbedanya suara sebuah tombol setiap kali ditekan sehingga mempersulit pengenalan tombol meskipun konstrain bahasa telah diaplikasikan.

menempatkan mikrofon *wireless* pada area kerja pengguna, melatih perekaman secara kontinyu, mencoba melakukan serangan dan mengevaluasi apakah teks yang berarti dapat diperoleh.

Gambar 1 (*Overview* serangan) menunjukkan overview serangan secara global. Fase pertama melatih *recognizer* dengan tahapan berikut :

a. Ekstraksi Fitur, dengan menggunakan fitur *cepstrum* yang merupakan sebuah teknik yang dikembangkan oleh peneliti di bidang *voice recognition*.



Gambar 1. *Overview* serangan

2. Serangan

Perekaman dilakukan pada pengetikan teks berbahasa Inggris dengan keyboard dan menghasilkan sebuah *recognizer* yang dapat menentukan subsekuen tombol dengan akurasi tinggi dari suara yang direkam jika diketik oleh orang keyboard, serta kondisi perekaman yang sama. Perekaman ini dapat diimplementasikan, sebagai contoh, dengan

b. *Unsupervised Key Recognition*, dengan menggunakan data *training* yang tidak dinamai. Pengelompokkan setiap tombol ke dalam salah satu dari K buah kelas dilakukan dengan menggunakan metode pengelompokkan data standar. Jumlah K yang dipilih harus sedikit lebih banyak dari jumlah tombol *keyboard*. Jika kelas hasil pengelompokkan ini berkorespon-

den dengan tombol-tombol yang berbeda, maka pemetaan satu-ke-satu ada diantara tombol dan kelas. Sayangnya, algoritma pengelompokkan memiliki tingkat ketelitian dan ketepatan yang rendah. Kelas merupakan variabel random yang dipengaruhi oleh tombol yang diketik. Pada data yang dikelompokkan dengan baik, probabilitas dari satu atau beberapa kelas akan mendominasi pada setiap tombol. Setelah distribusi kondisional kelas ditentukan, sekuens tombol dapat dicari dengan informasi sekuens kelas untuk setiap tombol yang diketahui. Prediksi Hidden Markov Model (HMM) digunakan untuk memprediksi proses stokastik dengan menangkap korelasi antara tombol-tombol yang diketik pada sekuens tertentu.

c. *Spelling and Grammar Checking*, dengan menggunakan koreksi pengejaan dan model statistik tata bahasa berbasis kamus. Dua pendekatan ini dikombinasikan dalam HMM dan mampu meningkatkan akurasi. Pada proses ini, teks sudah mulai terbaca.
d. *Feedback-based Training*, yang menghasilkan sebuah *classifier* tombol yang tidak membutuhkan model tata bahasa dan pengejaan serta memungkinkan pengenalan teks secara random, seperti pengenalan *passwors*. Aspek heuristik digunakan untuk memilih kata-kata yang cenderung benar. Beberapa karakter pada beberapa kata yang diuji perlu diperbaiki untuk melatih *classifier*.

Fase *recognition* mengenali sampel *training* kembali. Pengenalan kedua ini secara khusus memberikan tingkat akurasi tombol yang lebih tinggi. Sejumlah koreksi pengejaan dan tata bahasa merupakan indikator kualitas *classifier*. Fase ini menggunakan *classifier* tombol yang telah dilatih untuk mengenali perekaman suara baru. Jika teks mengandung string acak, hasil akan langsung dikeluarkan. Jika teks yang dihasilkan adalah sebuah kata bahasa, maka pemodelan ejaan dan tata bahasa digunakan untuk memperbaiki hasil. Dalam menentukan apakah sebuah string adalah acak atau bukan, koreksi dilakukan dan hasilnya dilihat apakah menghasilkan teks yang berarti. Sampel baru dan sampel yang telah ada dapat digunakan bersama untuk memperoleh *classifier* tombol yang lebih akurat.

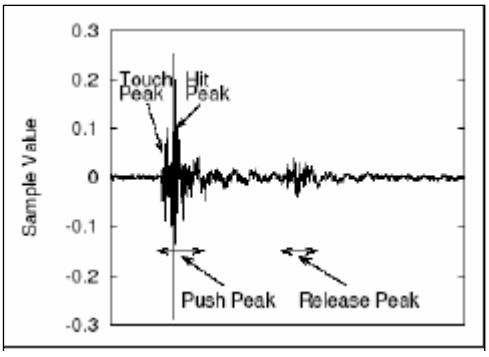
3. Aspek Detail

3.1 Ekstraksi Fitur Tombol

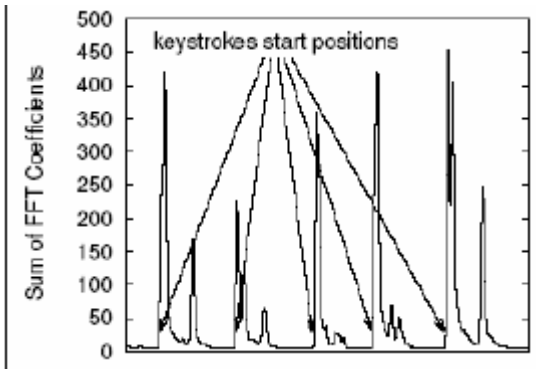
3.1.1 Ekstraksi Penekanan Tombol

Pengguna pada umumnya mengetik hingga 300 karakter per menit. Proses mengetik tombol terdiri dari menekan dan melepas tombol. Terdapat waktu jeda diantara penekanan tombol yang cukup besar untuk membedakan diantara tombol yang ditekan. Gambar berikut menunjukkan sinyal akustik dari puncak menekan dan puncak melepas tombol.

Nilai *Discrete Fourier Transform* sinyal dihitung dan koefisien dari semua FFT dijumlahkan sebagai energi. Nilai ambang digunakan untuk mendeteksi awal penekanan tombol.



Gambar 2. Sinyal audio penekanan tombol



Gambar 3. Tingkat energi pada penekanan 5 tombol

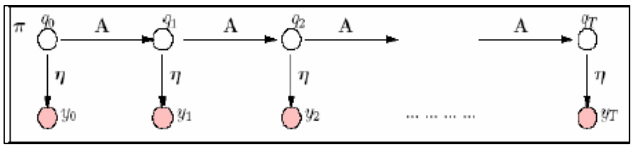
3.1.2 Fitur : Cepstrum vs FFT

Kedua fitur ini dapat diekstraksi dari sinyal audio pada periode dari posisi wav hingga posisi wav $+\Delta T$. Fitur FFT dengan $\Delta T \approx 5$ms berkoresponden dengan *touch peak* tombol, yaitu ketika jari menyentuh tombol. *Hit peak* yang merupakan waktu ketika tombol menyentuh lempeng tombol dapat digunakan, namun waktu sinyal ini sulit untuk disasarkan. Fitur *Cepstrum* telah digunakan dalam analisis dan pengenalan suara. Fitur ini telah diverifikasi secara empirik sehingga lebih efektif daripada koefisien FFT biasa pada sinyal suara. Setelah ekstraksi fitur ini, setiap penekanan tombol direpresentasikan sebagai sebuah fitur vektor.

3.2 Pengenalan Penekanan Tombol Tunggal *Unsupervised*

Pengenalan *unsupervised* ini mengenal penekanan tombol hanya dengan menggunakan data audio yang direkam dan tanpa data *training* atau bahasa. Langkah pertama yakni pengelompokkan vektor fitur menjadi K buah kelas. Algoritma yang memungkinkan yakni *K-means* dan EM pada G*aussian Mixture*. Nilai K = 50 akan memberikan hasil yang optimal. Nilai K yang lebih besar memberikan lebih banyak informasi dari sampel suara, namun mengakibatkan sistem lebih sensitif terhadap *noise*. Langkah kedua yakni pemulihan teks dari kelas-kelas ini dengan menggunakan HMM. Dalam rantai Markov, status *next* bergantung pada status *current*.

Untuk proses yang dimodelkan dengan HMM, status sistem yang sesungguhnya tidak diketahui dan direpresentasikan dengan variabel acak *hidden*. Variabel yang diketahui adalah yang bergantung pada status dan direpresentasikan dengan variabel output. Masalah utama dalam HMM adalah inferensi dimana variabel status yang tidak diketahui diinferensi dari sekuens observasi dan dapat dipecahkan dengan algoritma Viterbi. Masalah lainnya adalah *parameter estimation problem*, dimana parameter distribusi kondisional diestimasi dari sekuens observasi. Masalah ini diselesaikan dengan algoritma EM (*Expectation Maximization*). Contoh penggunaan HMM ditunjukkan pada gambar berikut :



Gambar 4. Hidden Markov Model untuk pengenalan tombol *Unsupervised*

HMM direpresentasikan dalam sebuah model grafis statistik. Lingkaran merepresentasikan variabel random. Lingkaran berwarna (*yi*) adalah observasi sedangkan yang tidak berwarna (*qi*) adalah variabel status yang tidak diketahui dan akan diinferensi. *Qi* adalah nama tombol ke-i dalam sekuens dan *yi* adalah kelas penekanan tombol hasil pengelompokkan. Panah dari *qi* ke *qi*+1 dan dari *qi* ke *yi* mengindikasikan bahwa yang berikutnya bergantung pada kondisi sebelumnya. Nilai pada panah adalah *entry* dari matriks probabilitas A dengan persamaan $p(qi+1|qi) = Aqi,qi+1$ yang

menunjukkan bahwa tombol *qi*+1 muncul setelah tombol *qi*. Matriks A adalah sebuah cara merepresentasikan data distribusi bigram plainteks dan ditentukan oleh tata bahasa dan diperoleh dari sekumpulan teks bahasa.

Terdapat pula persamaan $p(yi|qi) = \eta qi,yi$, yang menunjukkan probabilitas tombol *qi* dikelompokkan ke dalam kelas *yi* pada langkah sebelumnya. Dengan nilai *yi* yang diketahui dan output matriks η tidak diketahui, kita perlu menginferensi nilai *qi*. Algoritma EM dan Viterbi digunakan untuk mengestimasi parameter (menghasilkan matriks η) dan menginferensi *qi*.

Tombol *space* mudah dibedakan oleh pendengaran karena memiliki suara yang unik dan cukup sering digunakan. Penandaan sejumlah tombol *space*, pencarian kelas yang telah dikelompokkan untuk masing-masing tombol, penghitungan estimasi probabilitas untuk setiap anggota kelas dan penyimpanan nilai sebagai η kemudian dilakukan untuk memberi hasil yang baik.

3.3 Koreksi Error dengan Model Bahasa

Koreksi error adalah tahapan krusial dalam memperbaiki hasil dan digunakan dalam *unsupervised training, supervised training* dan pengenalan teks bahasa.

3.3.1 Koreksi Ejaan Probabilistik Sederhana

Menggunakan pemeriksa ejaan -seperti *Aspell*- adalah cara termudah dalam mengeksploitasi pengetahuan tentang bahasa, meskipun ketersediaan ejaannya dan kapabilitasnya cukup terbatas. Pemeriksa ini didesain untuk mengatasi kesalahan umum yang sering terjadi pada ketikan dan bukan kesalahan sumber akustik yang dilakukan oleh *classifier*. Untungnya, terdapat pola kesalahan yang seringkali dilakukan oleh *classifier* sehingga akurasi prediksi teks dapat meningkat. Kemudian, dijalankan *classifier* pada beberapa data *training* dan merekam semua hasil klasifikasi dan *error* yang terjadi.Selanjutnya, nilai matriks *E* dikalkulasi dengan persamaan :

$$E_{ij} = \hat{p}(y = i|x = j) = \frac{N_{x=j,y=i}}{N_{x=j}} \quad (1)$$

dengan $\hat{p}(\cdot)$ menunjukkan probabilitas yang diestimasi, *x* merupakan tombol yang diketik dan *y* tombol yang dikenali. $Nx=j,y=i$ adalah frekuensi dimana $x = j$, $y = i$. Kolom *E* memberikan distribusi probabilitas kondisional *y* yang diestimasi dengan nilai *x* yang diberikan. Dengan mengasumsikan bahwa huruf independen satu sama lain dan bernilai sama untuk huruf yang sama, probabilitas kondisional dari kata **Y** yang dikenali dapat dikalkulasi dengan tiap kata **X** yang diberikan, dengan persamaan :

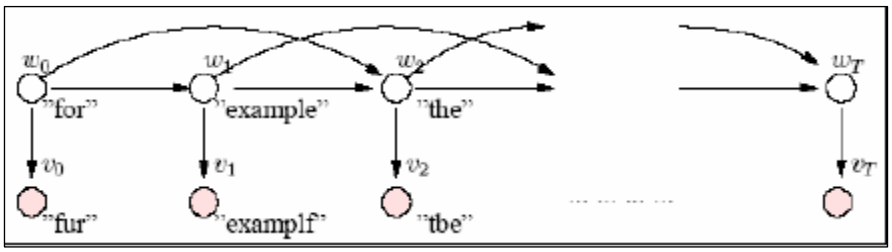
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{i=1}^{\text{length of X}} p(\mathbf{Y}_i|\mathbf{X}_i) \approx \prod_i E_{y_i,x_i} \quad (2)$$

Karena jumlah data *training* yang terbatas, akan terdapat banyak nol pada E jika persamaan langsung diimplementasikan.Hal ini tidak diinginkan karena berbagai kombinasi dapat terjadi dalam pengenalan data. Masalah ini mirip dengan masalah *zero-occurrence* pada model *n-gram*.

3.3.2 Penambahan Model Bahasa n-gram

Skema koreksi ejaan di atas tidak memperhitungkan frekuensi relative kata dan isu tata bahasa, seperti koreksi terhadap frase "fur example" sebagai ejaan yang benar karena "fur" adalah kata kamus meskipun frase sebenarnya adalah "for example". Jalan untuk memperbaiki adalah dengan menggunakan pemodelan bahasa *n-gram* yang memodelkan frekuensi kata dan relasi

antara kata-kata yang berdekatan secara probabilistik. Trigram dikombinasikan dengan koreksi ejaan di atas dan memodelkan kalimat dengan model grafis seperti pada gambar berikut :



Gambar 5. Model bahasa Trigram dengan koreksi

Variabel *hidden wt* adalah kata-kata pada kalimat asli. *vt* adalah kata-kata yang dikenali. *p*(*vt*|*wt*) dikalkulasi dengan persamaan diatas.

3.4 Supervised Training and Recognition

Supervised training merupakan proses *training* yang dilakukan pada data *training* yang dinamai. Proses *training* berbasis umpan balik dilakukan secara iteratif dengan karakter *previous* yang dikenali digunakan pada setiap iterasi sebagai sampel *training* untuk meningkatkan akurasi *classifier*. *Training* dilakukan untuk menerima vektor fitur dan label yang berkoresponden serta menghasilkan model yang digunakan dalam *recognition*. Proses *recognition* menerima vektor fitur dan melatih model klasifikasi serta menghasilkan label untuk setiap fitur vektor.

3.4.1 Neural Network

Metode yang digunakan adalah *neural network* probabilistik terbaik yang dapat menangani klasifikasi.

3.4.2 Linear Classification (Discriminant)

Metode ini mengasumsikan data sebagai Gaussian dan mencoba menemukan *hyperplane* untuk membagi kelas.

3.4.3 Gaussian Mixtures

Metode ketiga ini lebih rumit dari klasifikasi linear dengan mengasumsikan bahwa setiap kelas berkoresponden dengan gabungan dari beberapa distribusi Gaussian. Hal ini menunjukkan fakta bahwa setiap tombol dapat memiliki beberapa suara yang sedikit berbeda, bergantung pada gaya mengetik. Algoritma EM dapat digunakan untuk melatih model gabungan Gaussian ini. Gabungan Gaussian yang lebih banyak akan memberikan model potensial yang lebih akurat namun memerlukan lebih banyak parameter yang dilatih, memerlukan data *training* lebih banyak dan terkadang membuat algoritma ini kurang stabil.

4. Pengembangan

4.1 Pengembangan Serangan

Serangan tidak memperhitungkan tombol khusus seperti *Shift, Backspace* dan *Capslock*. Terdapat dua isu penting, yaitu apakah penekanan tombol untuk tombol khusus di atas terpisah dari penekanan tombol lain pada waktu pemrosesan sinyal dan bagaimana tombol pengubah karakter seperti Shift dapat menyesuaikan skema koreksi ejaan. Salah satu solusi singkat adalah dengan mengganti tombol *Shift* atau

Capslock dengan tombol *Space*. Pertimbangan lain adalah bagaimana teks akhir setelah mengaplikasikan *backspace* dengan mempertahankan agar kompleksitas algoritma koreksi tidak tinggi. Hal yang menarik adalah mendeteksi penekanan tombol pada aplikasi khusus, seperti *visual editor* dan *software development environment*. Pemeriksaan teks yang diketik pada lingkungan ini cukup sulit karena tombol khusus atau tambahan lebih sering digunakan.

Metode alternatif untuk prosedur *feedback training* adalah Hierarchical Hidden Markov Models (HHMM) yang memiliki level tata bahasa dan ejaan yang dibangun dalam sebuah model tunggal. Serangan akan kurang berhasil jika terdapat *noise* seperti musik yang didengarkan ketika mengetik. Pengembangan pemrosesan sinyal yang memisahkan sebuah suara dengan suara lain dalam kanal yang sama dapat dilakukan lebih lanjut.

4.2 Defenses

Untuk dapat bertahan dari serangan, keamanan fisik ruangan dan mesin harus terjamin.Selain itu, harus dipastikan bahwa suara tidak dapat terekam dari luar ruangan. Penggunaan *keyboard* yang tidak ribut dan memiliki suara tombol yang seragam (sehingga sulit dibedakan) dapat mengurangi kemungkinan penyerangan.

5. Kesimpulan

Serangan jenis baru ini, yakni *keyboard emanations*, hanya memerlukan perekaman akustik dari proses pengetikan yang menggunakan keyboard untuk kemudian menyelidiki tipe isi pengetikan tersebut. Dibandingkan metode lainnya, serangan ini lebih umum dan natural. Serangan *acoustic emanations* ini berhasil diaplikasikan oleh Dmitri Asonov dan Rakesh Agrawal pada tahun 2004, selain pada keyboard komputer, juga pada keyboard noteboard, pad telepon, dan pad ATM. *Countermeasure* untuk serangan *acoustic emanations* ini dapat berupa keyboard yang menghasilkan suara seminimal mungkin (bebas-suara). Jenis keyboard ini tentunya tidak murah serta tidak familiar bagi *user*. Riset lebih luas dalam bidang ini dapat dilakukan dalam mengeksplorasi variabel-variabel pada lingkungan yang berpengaruh pada berhasilnya serangan ini dilakukan, atau meneliti bagaimana peralatan input yang menghasilkan suara seperti keyboard itu dapat terkena serangan.

REFERENSI:

[1] D.Asonov, *et al*., *Keyboard Acoustic Emanations*, IBM Almaden Research Center, 2004.

[2] L. Zhuang, *et al., Keyboard Acoustic Emanations Revisited*, University of California, Berkeley, 2005.

[3] C. Karlof, *et al.*, *Hidden Markov Model Cryptanalysis*, Department of Computer Science, University of California, Berkeley, USA, 2003.