Penggunaan Bilangan Acak untuk Membangkitkan Passphrase sebagai Alternatif Sandi Lewat

Sandy Socrates 13508044[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]if18044@students.if.itb.ac.id

***Abstract*—Dengan semakin majunya jaman dan pesatnya perkembangan teknologi dan informasi, banyak hal kian dimudahkan dengan berbagai layanan yang ditawarkan Internet. Hampir seluruh layanan yang ditawarkan ini menggunakan autentikasi untuk menjaga keamanan data yang disimpan serta kepercayaan pengguna kepada penyedia layanan. Sandi lewat, sebagai sebuah solusi otentikasi, adalah sebuah hal yang sangat umum penggunaanya. Untuk menjaga agar seorang penyerang sulit mendapatkan sandi lewat, sandi lewat haruslah dibuat rumit. Tapi hal ini tentunya juga mempersulit pengguna untuk mengingat sandi lewatnya dan bahkan dapat memaksa mereka menyimpan sandi lewat di tempat lain. Hal ini justru sangat disayangkan karena tempat penyimpanan sandi lewat bisa saja tidak aman dan bisa diakses oleh penyerang. Untuk menjawab masalah ini *passphrase* diusulkan sebagai sebuah solusi. Sebuah *passphrase* pada dasarnya adalah sebuah kalimat atau frasa yang lebih aman dari sebuah sandi lewat biasa, *passphrase* bisa memiliki arti atau hanya berupa kata acak, antara sebuah kata dengan kata lainnya dipisahkan oleh sebuah karakter non huruf seperti spasi dan garis bawah. Karena berisi kumpulan kata, *passphrase* lebih mudah diingat oleh penggunanya dan juga lebih sulit ditebak oleh penyerang daripada sebuah sandi lewat biasa.**

***Index Terms*— otentikasi, Ineternet, *passphrase,* penyerang, *s*andi lewat.**

I. PENDAHULUAN

Layanan berbasis *Internet* telah mengisi hari-hari manusia. Hal ini sudah merupakan sebuah fakta yang lumrah dan dianggap hal yang biasa. Tua muda, laki wanita, dan bahkan dari golongan yang kurang mampu sudah mengenal *Internet*. Dahulu kita menggunakan pos dan surat untuk saling berkomuniaksi dengan sanak saudara dan kerabat yang bertempat tinggal jauh. Tapi saat ini *e-mail* (*electronic mail* / surat elektronik)telah menggusur surat tradisional, bahkan di beberapa negara layanan, Jepang misalnya, SMS kalah populer dibandingkan dengan email.

Ditambah dengan bermunculannya layanan jejaring sosial, jarak seakan sudah tidak dapat memendung arus informasi yang terus mengalir setiap detiknya. Dunia seakan berada di sebuah titik kecil, sehingga semua orang yang ada di dalamnya dapat saling berkomunikasi satu sama lainnya, karena ukurannya yang sangat kecil.

Informasi sama dengan sebuah mata uang yang memiliki nilai dan harga, serta bisa saja memiliki pengaruh yang sangat besar tergantung dari isi informasinya dan penerima informasi yang dimaksud. Karena nilai informasi yang tinggi ini pencurian identitas, pembobolan rekening, dan pencurian data serta kejahatan informasi lainnya kini menjadi kejahatan baru yang juga ikut berkembang.

Sebuah layanan *Internet* menggunakan informasi dan data yang bersifat pribadi bagi penggunanya. Dan bila nformasi tersebut jatuh ke tangan yang salah, penyerang misalnya, pengguna bisa saja mendapatkan kerugian yang besar. Untuk menjaga kepribadian sebuah data dan penggunaannya yang tidak sesuai, layanan *Internet* menggunakan proses autentikasi.

Sandi lewat adalah proses otentikasi yang paling banyak digunakan di *Internet*. Akan tetapi tetap saja masih ada penyerangan yang terjadi. Para penyerang ini dapat dengan mudah mendapatkan sandi lewat pengguna tanpa menggunakan *bruteforce* dari setiap sandi lewat yang mungkin. Dengan pengetahuan tentang pengguna, penyerang bisa mengira-ngira sandi lewat yang mingkin seperti nama anak, nama saudara, hobi ataupun tanggal lahir.

Untuk mencegah penyerangan terjadi, sebuah sandi lewat haruslah kuat dan tidak mudah ditebak oleh seorang penyerang untuk menghindari serangan *bruteforce* dan dengan mengira-ngira. Kuat tidaknya sebuah sandi lewat dilihat dari ukuran seberapa efektif sandi lewat tersebut mampu menahan serangan *bruteforce* dan mengira-ngira.Semakin rumit, panjang dan tidak berpola, sebuah sandi lewat akan memiliki kekuatan yang lebih besar.

Namun pada umumnya kekuatan yang lebih besar membuat sandi lewat semakin sulit untuk diingat. Sebuah pilihan yang sulit, jika kita tidak ingat sandi lewat kita tidak bisa menggunakan layanan yang dimaksud. Sebaliknya dengan sandi lewat yang mudah justru akan membahayakan keamanan informasi dan data yang kota simpan di layanan tersebut.

Passphrase memberikan harapan yang menjanjikan untuk menggantikan sandi lewat biasa. *Passphrase* biasanya cukup panjang, karena tersusun dari beberapa kata, dan juga sulit ditebak karena kata-katanya bisa tidak memiliki arti karena diambil menggunakan bilangan acak.

II. PRINSIP DASAR

A. Bilangan Acak (Random)

Bilangan acak adalah bilangan yang tidak dapat diprediksi[1]. Disebutkan pula dalam [1] bahwa bilangan acak banyak digunakan dalam kriptografi seperti dalam pembangkitan *initialization vector (IV)* pada algoritma kunci-simetri.

Bilangan acak yang di maksud adalah sebuah barisan bilangan yang acak. Adalah hal yang sangat sulit untuk membangkitkan bilangan acak yang benar-benar acak *(true random*). Metode yang digunakan untuk memebangkitkan bilangan acak yang berupa *pseudo random number generator(PRNG)* menghasilkan sebuah barisan acak yang yang semu. Disebut semu karena pembangkitan bilangan ini dapat diulang kembali.

Kriteria untuk sebuah pembangkit bilangan acak untuk kriptofrafi tidak banyak. Seperti yang disebutkan pada [2] syarat dari sebuah pembangkit bilangan tidak banyak, hanya ada satu. Kriteria yang dimaksud yaitu jangan sampai penyerang tahu bilangan acak yang akan dibangkitkan berikutnya walaupun dia telah mengetahui bilangan-bilangan sebelumnya.

Selain dengan menggunakan *PRNG,* kita bisa menggunakan beberapa sumber untuk membangkitkan bilangan *true-random*. Seperti dengan mengambil bit dari sebuah perangkat I/O, sumber radioaktif, efek kuantum, polarisasi foton, ataupun dari *eletronic noise* yang ditangkap oleh sebuah *microphone*.

$$H = -\sum_x p_x(\log_2(p_x)) \tag{1}$$

Untuk mengukur keacakan dari sebuah pembangkit bilangan acak digunakan entropi. Entropi didefinisikan pada (1) di mana *x* adalah nilai yang mungkin dalam sebuah barisan bilangan, sebagai contoh sebuah himpunan bit dari sebuah variabel. *Px* adalah kemungkinan kemunculan *x.*

$$J = H/|x| \tag{2}$$

(2) menunjukkan bahwa *J* adalah hubungan yang menujukkan jumlah bit yang tidak bisa ditebak dibanding jumlah bit total dari sebuah *stream.*|*x*| adalah panjang dari *stream.*

$$E = min_{1 \le |x| < \infty} J \tag{3}$$

Pada (3) didefinisikan *E* atau entropi absolut sebagai entropi minimum (ketidakmungkinan untuk dikira) per bit dari *stream.*

B. Kekuatan Sandi Lewat

Kekuatan sebuah sandi lewat adalah pengukuran seberapa kuat sebuah sandi lewat dapat menahan serangan . Pada umunya diperkirakan berapa kali penyerang dapat melakukan percobaan sebelum dapat menemukan sandi lewat yang benar. Kekuatan dari sebuah sandi lewat diukur dari parameter panjang, kompleksitas, dan seberapa tidak bisa dikira sandi lewat tadi.[3]

Pengukuran kekuatan sandi lewat bisa juga didapat dari kemudahan penyerang untuk mendapatkan sandi lewat dari tempat penyimpanannya dan jumlah percobaan yang dilakukan oleh penyerang.

Seorang pengguna bisasanya membuat sandi lewat yang memiliki keterhubungan dengan dirinya, seperti tanggal lahir misalnya. Kecenderungan penggunaan angka yang familiar dan kalimat yang mudah juga sangat tinggi. Lima puluh teratas untuk penggunaan sandi lewat diduduki oleh kata-kata maupun kalimat dan angka yang mudah diingat seperti “1234567890” dan “letmein”.

Pembangkitan sandi lewat oleh komputer tentunya dapat menghilangkan hubungan pengguna dan sandi lewat yang dimilikinya. Namunn tentu saja pembuatan sandi lewat secara acak kadang memberikan kata maupun ganbungan kata dan angka yang sangat sulit untuk diingat. Seperti misalnya seseorang me-*reset* sandi lewat untuk sebuah layanan dan mendapatkan sandi lewat baru “7Nj80rT”. Kekuatan sandi lewat ini besar jika dibandingkan dengan sandi lewat biasa namun lebih sulit untuk diingat.

Untuk memperkirakan kekuatan sebuah sandi lewat yang dibangkitkan oleh komputer dapat digunakan entropi. Sebuah sandi lewat yang acak dapat terdiri dari angka, huruf , dan simbol yang spesifik.

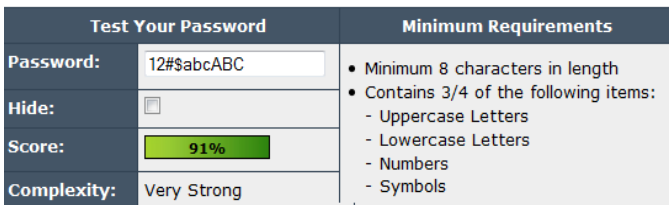
Sebuah sandi lewat yang dibangkitkan oleh komputer, yang diambil dengan cara memilih sekumpulan simbol N untuk sebuah string dengan panjang L memiliki nilai entropi seperti ditulisakan pada (4):

$$H = \log_2 N^L = L \log_2 N = L \log N / \log 2 \tag{4}$$

Pada umumnya, pengecekan sebuah password melihat kompleksitas dari karaker apa saja yang membangun sandi lewat itu .Setiap layanan pemeriksan kemananan sandi lewat misalnya [4] menggunakan pengecekan kompleksitas dengan caranya masing-masing, karena memiliki satandarnya yang berbeda-beda juga.

Satandar-standar ini tidak memberikan pengukuran yang pasti akan kekuatan sandi lewat, namun hal ini bisa dijadikan referensi oleh pengguna untuk membuat sandi lewat yang rumit dan sulit untuk ditebak dan di-*bruteforce* oleh penyerang.

Contoh : Hasil pengecekan kekeuatan sandi lwat dengan menggunakan [4]. Adapaun masukannya adalah “12#$abcABC”.



Gambar 1. Hasil pengukuran untuk masukan “12#$abcABC”

Additions	Type	Rate	Count	Bonus
Number of Characters	Flat	+(n*4)	10	+ 40
Uppercase Letters	Cond/Incr	+((len-n)*2)	3	+ 14
Lowercase Letters	Cond/Incr	+((len-n)*2)	3	+ 14
Numbers	Cond	+(n*4)	2	+ 8
Symbols	Flat	+(n*6)	2	+ 12
Middle Numbers or Symbols	Flat	+(n*2)	3	+ 6
Requirements	Flat	+(n*2)	5	+ 10
Deductions				
Letters Only	Flat	-n	0	0
Numbers Only	Flat	-n	0	0
Repeat Characters (Case Insensit	Comp	-	0	0
Consecutive Uppercase Letters	Flat	-(n*2)	2	- 4
Consecutive Lowercase Letters	Flat	-(n*2)	2	- 4
Consecutive Numbers	Flat	-(n*2)	1	- 2
Sequential Letters (3+)	Flat	-(n*3)	1	- 3
Sequential Numbers (3+)	Flat	-(n*3)	0	0
Sequential Symbols (3+)	Flat	-(n*3)	0	0

Gambar 2. Penjelasan tambahan untuk Gambar 1.

Sedangkan pada [5] pengukuran diberikan berdasarkan perhitungan permutasi dari tiap-tiap hurufnya.

```
Function estimate_time(input string s) → int
int time ← 1
  int calculationpersecond = 10000000
  if (isCommonPassword)
    → 0
  possibleChar ← checkUnicode(s)
     possibleCombination ← pow(possibleChar,
s.length)
  → possibleCombination / calculationpersecond
```

Tabel 1. Algoritma yang digunakan pada [5]

Fungsi ini akan memberikan hasil pembagian dari jumlah kombinasi yang mungkin dari total karakter yang mungkin dibagi kalkulasi yang mungkin dilakukan oleh sebuah *Personal Computer* biasa.

Fungsi isCommonPassword akan mengecek apakah sandi lewat yang dimasukkan termasuk dalam list dari lima puluh sandi lewat yang paling sering digunakan.

Fungsi checkUnicode akan memberikan jumlah karakter yang mungkin dilibatkan dalam pembuatan sandi lewat. Misalkan sebuah string s = "12#$abcABC" dimasukkan dalam fungsi ini, kembalian dari fungsi ini akan sama dengan 10 (terdapat angka) + 13 (terdapat #,$, yaitu simbol yang terdapat di keyboard) + 26 (terdapat alfabet kecil) + 26 (alfabet besar) = 75.

PossibleCombination akan menghitung panjang dari string masukan dipangkatkan dengan jumlah total karakter yang mungkin. Dari sini didapatkan jumlah kemungkinan rangkaian karakter yang dapat dibentuk.

C. Passphrase

Pada [7] disebutkan oleh Arnoud Engelfriet, seorang *IT-lawyer*, berkata bahwa sebuha passphrase adalah sebuah kalimat atau frasa yang digunakan daripada sebuah sandi lewat. Karena panjangnya, sebuh *passphrase* lebih aman dibandingkan dengan sebuah sandi lewat. Dan sebuah *passphrase* tetap mudah untuk diingat.

Sebuah *passphrase* pada umunya dalah sebuah kalimat atau frasa yang menyajikan tingkat kemanan yang lebih tinggi dibandingkan dengan sebuah sandi lewat. Sandi lewat pada umumnya hanya memiliki panjang enam sampai delapan karakter, dan itu sangat tidak aman.

Dengan bilangan acak sebuah *passphrase* dapat dibangkitkan. Cara ini disebut dengan *diceware*. Kenapa disebut seperti itu, karena pengacakannya dapat menggunakan lima buah dadu

Diceware Passphrase generator adalah sebuah list berisi kata-kata yang telah diurutkan sehingga bisa dilakukan pengambilan acak sedemikian rupa dengan menggunaka lima buah dadu. Jumlah total kata dari list ini adalah 7776 kata pendek dalam bahasa Inggris. Perlu diingat bahwa pangkat lima dari enam adalah 7776.

Dengan cara ini bisa didapatkan sebuah *passphrase* dengan mengocok kelima dadu sesuai jumlah kata yang diinginkan.

III. KEKUATAN PASSPHRASE

Untuk mengukur seberapa kuat sebuah *passphrase* untuk dapat menggantikan sebuah sandi lewat maka penulis melakukan sebuah tes.

Dengan menggunakan metode *diceware* untuk membangkitkan sebuah *passphrase* dengan 3, 4, 5, 6 dan 7 huruf

A. 3 Buah Kata

Huruf yang dibangkitkan dengan acak

651435214365123

Passphrase yang dihasilkan :

yl river yemen

Pengecekan pada [4]

Score : 26%
Complexity : Weak

Pengecekan pada [5]

It would take
About 212 thousand years
for a desktop PC to crack your password

Pengecekan pada [6]

Length: 14
Strength: Reasonable - This password is fairly secure cryptographically and skilled hackers may need some good computing power to crack it. (Depends greatly on implementation!)
Entropy: 50.9 bits
Charset Size: 27 characters

B. 4 Buah Kata

Huruf yang dibangkitkan dengan acak

5423542564253265132 3

Passphrase yang dihasilkan :

sk moral feet ramp

Pengecekan pada [4]

Score : 36%

Complexity : Weak

Pengecekan pada [5]

It would take

About 97 billion years

for a desktop PC to crack your password

Pengecekan pada [6]

Length: 18
Strength: Strong - This password is typically good enough to safely guard sensitive information like financial records.
Entropy: 69 bits
Charset Size: 27 characters

C. 5 Buah Kata

Huruf yang dibangkitkan dengan acak

4652342164364356431612432

Passphrase yang dihasilkan :

pump memoir lifo wk ares

Pengecekan pada [4]

Score : 46%

Complexity : Good

Pengecekan pada [5]

It would take

About 30 quintillion years

for a desktop PC to crack your password

Pengecekan pada [6]

Length: 24
Strength: Strong - This password is typically good enough to safely guard sensitive information like financial records.
Entropy: 94.1 bits
Charset Size: 27 characters

D. 6 Buah Kata

Huruf yang dibangkitkan dengan acak

4325135146315463163216431632 52

Passphrase yang dihasilkan :

n kabul grave grip chef void

Pengecekan pada [4]

Score : 50%

Complexity : Good

Pengecekan pada [5]

It would take

About 527 sextillion years

for a desktop PC to crack your password

Pengecekan pada [6]

Length: 28
Strength: Strong - This password is typically good enough to safely guard sensitive information like financial records.
Entropy: 107.6 bits
Charset Size: 27 characters

E. 7 Buah Kata

Huruf yang dibangkitkan dengan acak

46543651436154365436436436436436612

Passphrase yang dihasilkan :

puppy yl trace 2 not wordy loamy

Pengecekan pada [4]

Score :78%
Complexity : Excellent

Pengecekan pada [5]

It would take
About 8 decillion years
for a desktop PC to crack your password

Pengecekan pada [6]

Length: 33 **Strength:** Very Strong - More often than not, this level of security is overkill. **Entropy:** 140.9 bits **Charset Size:** 37 characters

E. 8 Buah Kata

Huruf yang dibangkitkan dengan acak

4643643652436565654365143654363661146431

Passphrase yang dihasilkan :

proud nov novo team yl 2 loam prong

Pengecekan pada [4]

Score 88%
Complexity : Excellent

Pengecekan pada [5]

It would take
About 374 undecillion years
for a desktop PC to crack your password

Pengecekan pada [6]

Length: 36 **Strength: Very Strong - More often than not, this level of security is overkill.** **Entropy: 153 bits** **Charset Size: 37 characters**

IV Analisis Pengujian

Dengan panjang passphrase 3 dan 4 tingkat elamanannya masih relatif rendah. Maka dari itu sebuah *passphrase* yang baik harusnya memiliki panjang lebuh dari 4 kata.

Pengecekan kata tadi hanya memiliki Charset size 27, karena yang digunakan hanya alfabet kecil dan spasi. Pengunaan huruf besar serta kombinasi dengan simbol dan angka akan meningkatkan charset size dan entropi. Karena itu ada baiknya jika spasi bisa diganti dengan tanda atau simbol lain.

V. Kesimpulan

Menggunakan sandi lewat yang aman mengurangi resiko terjadinya pencurian otorisasi oleh pihak yang tidak berwenang. Akan tetapi keamanan sebuah sandi lewat tidak dapat menggantikan keamanan dari sistem dan kontrol layanan. Tetap saja kemanan data berada di sistem. Sandi lewat yang kuat juga tidak bisa melindungi anda dari *keylogger, wiretapping, phising*, *social engineering*, dan juga orang yang berada di belakang anda. Oleh karena itu keamanan data dan informasi yang anda pakai tidak sepenuhnya bergantung pada keamanana sandi lewat namun juga keamanan sistem dan lingkungan kerja sandi lewat. Namun penggunan *passphrase* dapat meningkatkan keamanan agar sebuah sandi lewat tidak mudah untuk ditebak dan ditembus oleh penyerang.

Sebuah *passphrase* memiliki tingkat kemanan yang lebih tinggi dari pada sebuah sandi lewat biasa.

Penggunaan angka dan simbol bisa menambah keamanan sandi lewat.

VII. Acknowledgment

Puji syukur penulis panjatkan ke hadirat illahirobbi karena telah memberikan kemampuan untuk menyelesaikan tugas makalah ini.

Terima kasih untuk orang tua, keluarga, dosen mata kuliah Kriptografi serta teman teman di GCD yang telah bersedia untuk bekerja sama dengan penulis selama pengerjaan makalah ini.

DAFTARPUSTAKA

[1] Munir, Rinaldi."*Pembangkit Bilangan Acak*" unpublished

[2] Ellison, Carl. "*Cryptographic Random Number*", 1995 http://world.std.com/~cme/P1363/ranno.html.

[3] McDowell, Mindi et al. "Cyber Security Tip ST04-002". *Choosing and Protecting Passwords*. US CERT, 2004. Diakses Mei, 2011.

[4] Todnem, Jeff. *"The Password Meter"*, 2005 http://www.passwordmeter.com/pwdmeter.js

[5] Small Hadron Collider, *"How Secure is My Password" http://howsecureismypassword.net/*

[6] Nuff, E. Strength test http://rumkin.com/tools/password/passchk.php

[7] Engel, Friet. *"The Passphrase FAQ"*

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, 7 Mei 2011

Sandy Socrates
13508044