Enkripsi Sederhana dengan Base64 dan Substitusi Monoalfabetik ke Huruf Non-Latin

Yusuf Adriansyah
13507120

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jalan Ganesha 10, Bandung, Jawa Barat – 40132
ysf4m1c@yahoo.co.id

ABSTRAK

Ada banyak algoritma kriptografi klasik sebelum perang dunia kedua. Namun, semuanya berusaha mengenkripsi teks dalam suatu bahasa menjadi cipherteks yang menggunakan set huruf yang sama dengan plainteksnya! Contohnya Caesar Cipher, ia mengenkripsi tulisan dalam bahasa Latin, bahasa yang dipakai bangsa Roma saat itu, menjadi cipherteks yang masih memakai huruf latin. Biarpun cipherteks tidak bisa dimengerti lagi isinya, tetapi tetap bisa dibaca, karena masih menggunakan huruf latin. Kriptanalis dengan mudah memecahkannya karena masih terbaca. Bagaimana jika cipherteks kita buat menjadi tidak terbaca?

Dalam makalah ini, penulis mencoba memperbaiki kelemahan tadi dengan cara men-*transliterasi* (alih aksara) cipherteks dari huruf latin ke huruf selain latin. Yang dibahas dalam makalah ini hanya huruf yunani, huruf kiril, huruf hangul, dan huruf-huruf Jepang.

Untuk memperkuat enkripsi, ada baiknya sebelum dialih aksara, cipherteks sudah dienkripsi dengan algoritma yang lain. Dalam contoh ini penulis memakai Base64 karena mudah dilakukan, hasilnya sendiri sudah cukup mengecoh, dan dijamin *printable* (bisa dicetak)

Kata kunci: Kriptografi klasik, Base64, *Monoalphabetic substitution*.

1. PENDAHULUAN

Setiap manusia memiliki kebutuhan yang disebut *privacy*. Ada informasi yang tidak boleh diketahui oleh orang lain, siapapun itu. Ada informasi yang boleh diketahui hanya orang-orang tertentu saja. Dan lagi, manusia terbatas. Tidak selalu seseorang bisa menyampaikan informasi rahasia ke orang yang dituju secara langsung, pesannya dibawa sendirian. Sebab, kita hidup di dunia yang tidak aman. Serangan penyamun di tengah jalan acapkali terjadi. Terlebih lagi bila jarak antara pengirim dan penerima terlalu jauh, akan repot sekali bila setiap minggu harus kirim pesan rahasia.

Setidaknya ada empat cara penyembunyian pesan, yaitu:

- **Tulisan yang buruk**
  digunakan oleh dokter ketika menulis resep obat.
- **Enkripsi**
  mengubah pesan menjadi sesuatu yang tidak bermakna.
- **Alih bahasa (*translate*)**
  mengubah pesan dari sebuah bahasa ke bahasa lain supaya tidak bisa dimengerti orang sekitar.
- **Steganografi**
  menyembunyikan pesan dalam media lain sampai tak ada yang curiga.

Enkripsi sangat umum kita temui. Enkripsi paling mudah adalah membalik teks. Misalnya, "kumpul jam lima sore" dibalik menjadi "eros amil maj lupmuk". Cukup sedikit kejelian untuk menemukan plainteksnya.

Steganografi sangat luas aplikasinya. Menggantikan kata-kata dengan kata-kata dalam *codebook* seperti "barang sudah datang" menjadi "pergi sebentar keluar" juga termasuk steganografi. Menulis dengan tinta air jeruk, mengekstrapolasi huruf pertama menjadi kata-kata, menyembunyikan teks dalam gambar, semuanya termasuk steganografi. Masih ingatkah permainan masa kecil, ketika kita diberikan tabel huruf 15 × 15 lalu bu guru berkata "Temukan 10 kata dalam huruf-huruf berikut secara mendatar, menurun, atau diagonal"? Itu sangat dekat dengan steganografi, namun belum bisa digolongkan steganografi.

Gambar 1. Contoh alih bahasa untuk menyembunyikan pesan rahasia

Gambar 1 di halaman sebelumnya adalah pesan yang saya tempel di dinding kamar. Ini adalah sebuah pengingat (*reminder*) buat saya, namun jangan ada orang lain tahu apa isinya. Maka dari itu, saya terjemahkan ke bahasa Jepang lalu saya tulis dengan kuas. Tidak dienkripsi, hanya diterjemahkan saja.

2. LANDASAN TEORI

2.1. Penyandian Base64

Base64 sejatinya bukan enkripsi, namun hanyalah sebuah standar penyandian (*encoding*). Sejarah Base64 berawal dari surat elektronik (email). Pada waktu itu, email dikirim dengan protokol SMTP (*simple mail transfer protocol*) ke mail server kita, lalu dikirim ke mailbox orang yang dituju di mail server tujuan. "Protokol" adalah tata cara mesin (komputer) saling berkomunikasi via jaringan. Supaya e-mail bisa sampai ke orang yang dituju, ia harus mengunduhnya terlebih dahulu. Proses download email menggunakan protokol POP (*post office protocol*). Saat ini POP sudah mencapai versi 3 sehingga disebut POP3. Alternatif yang lebih baik dari POP adalah IMAP (*internet mail access protocol*). IMAP sudah mencapai versi 4.

Baik POP maupun SMTP adalah protokol berbasis teks. Encoding yang digunakan adalah ASCII. Tidak masalah bila kita hanya ingin mengirim email teks saja. Masalah muncul ketika email berkembang, menjadi punya kemampuan untuk mengirim lampiran (*attachment*). Apa yang dilampirkan adalah file, dan file ini bisa file apa saja — termasuk file biner.

Kebetulan POP dan SMTP sama dalam hal terminasi pesan. Mereka menggunakan deretan karakter `CR LF . CR LF` (*carriage return – line feed* – tanda titik – *carriage return – line feed*) sebagai akhir dari pesan. Apa yang terjadi bila file biner kita di tengah-tengah terdapat byte-byte berikut: `0D 0A 2E 0D 0A`? Nilai rangkaian byte tadi adalah kode ASCII dari `CR LF . CR LF` sehingga server akan menganggap pesan yang dikirim berhenti sampai di sana. File yang kita lampirkan kita akan putus di tengah.

Untuk mengatasi masalah ini, dibuatlah penyandian Base-64. Cara kerja Base64 adalah sebagai berikut:

- Kelompokkan pesan setiap 3 karakter (3 byte = 24 bit). Bila terdapat sisa di akhir, tambahkan (*padding*) bit 0 sehingga panjangnya genap 24 bit.
- Pecah 24 bit tadi menjadi 4 kelompok yang masing-masing beranggotakan 6 bit.
- Setiap kelompok sekarang punya 2^6 kemungkinan susunan bit, berarti ada $2^6 = 64$ karakter tersedia untuk merepresentasikan 6 bit ini. Petakan setiap kelompok dengan karakter yang terdapat dalam tabel.

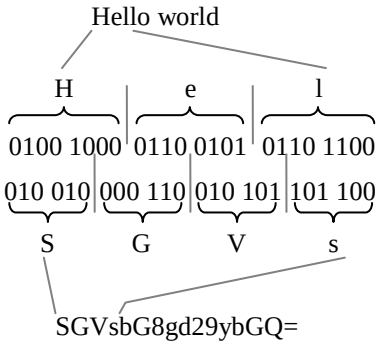
Karakter yang dipakai adalah huruf latin A-Z, huruf kecil a-z, dan angka 0-9. Semua berjumlah 62. Dua sisanya memakai simbol + dan / sehingga totalnya 64.

Tabel 1. Tabel Base64 lengkap

Bit	Desimal	Karakter	Bit	Desimal	Karakter
000 000	0	A	100 000	32	g
000 001	1	B	100 001	33	h
000 010	2	C	100 010	34	i
000 011	3	D	100 011	35	j
000 100	4	E	100 100	36	k
000 101	5	F	100 101	37	l
000 110	6	G	100 110	38	m
000 111	7	H	100 111	39	n
001 000	8	I	101 000	40	o
001 001	9	J	101 001	41	p
001 010	10	K	101 010	42	q
001 011	11	L	101 011	43	r
001 100	12	M	101 100	44	s
001 101	13	N	101 101	45	t
001 110	14	O	101 110	46	u
001 111	15	P	101 111	47	v
010 000	16	Q	110 000	48	w
010 001	17	R	110 001	49	x
010 010	18	S	110 010	50	y
010 011	19	T	110 011	51	z
010 100	20	U	110 100	52	0
010 101	21	V	110 101	53	1
010 110	22	W	110 110	54	2
010 111	23	X	110 111	55	3
011 000	24	Y	111 000	56	4
011 001	25	Z	111 001	57	5
011 010	26	a	111 010	58	6
011 011	27	b	111 011	59	7
011 100	28	c	111 100	60	8
011 101	29	d	111 101	61	9
011 110	30	e	111 110	62	+
011 111	31	f	111 111	63	/

Ditambah satu karakter khusus untuk *padding byte* yaitu simbol =. Bila dalam kelompok 3-byte itu satu byte terakhir hanya berisi *padding bit*, maka satu karakter = ditambahkan. Bila dua, maka dua karakter = (menjadi ==).

Contoh:



Standar penyandian Base64 sudah didefinisikan dalam dokumen RFC 1421 (http://tools.ietf.org/html/rfc1421)[6].

2.2. Substitusi Monoalfabetik

Subtstitusi monoalfabetik adalah algoritma enkripsi dengan cara mengganti setiap huruf pada plainteks dengan huruf pada tabel konversi. Tabel ini menjadi kunci bagi proses

enkripsi/dekripsi. Contoh yang sederhana adalah Caesar Cipher, dengan tabel kuncinya adalah:

Tabel 2. Caesar Cipher

Plain	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Cipher	D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

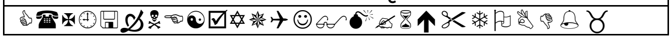
Kita bisa membuat tabel yang lain, misalnya seperti ini:

Tabel 3. Monoalphabetic substitution 1

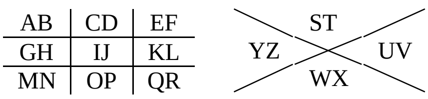
Plain	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Cipher	G J B L P N U W E C Z H A Y S Q V F R M I X T O K D

Juga tidak dilarang untuk menggunakan simbol, bukan?

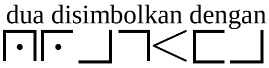
Tabel 4. Monoalphabetic substitution 2

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z


Sewaktu mengikuti pramuka ketika SD dulu, saya diajari seperti ini:



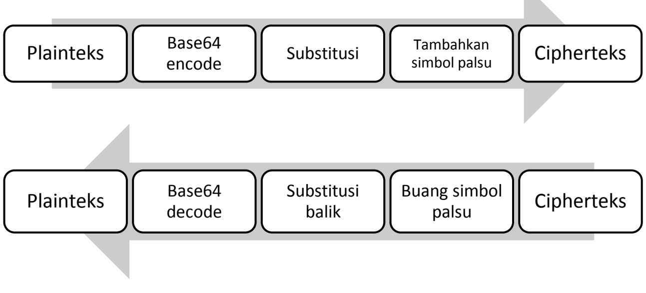
Gambar 2. Monoalphabetic substitution versi pramuka

Setiap huruf diganti dengan simbol "border" tempat huruf itu berada. Setiap sel dalam tabel terdiri dari dua huruf. Huruf pertama diganti hanya border, sedangkan huruf kedua disimbolkan dengan border dan titik. Contohnya,
 didekripsi menjadi "PRAMUKA".

Kriptanalisis untuk substitusi huruf tunggal adalah analisis frekuensi. Dalam bahasa Inggris, tiga huruf paling sering muncul adalah huruf E, T, dan A. Dalam bahasa Indonesia, tiga huruf paling sering muncul ialah A, M, dan E*. Huruf yang paling sering muncul dalam plainteks, akan berkorespondensi dengan huruf paling sering muncul di cipherteks.

3. ALGORITMA UMUM

Algoritma yang saya usulkan sederhana. Hanya balik arah.



Gambar 3. Proses enkripsi (atas) dan dekripsi (bawah)

Tentang Base64 sudah tidak perlu dibahas lagi, karena sudah ada standarnya. Lalu mengapa harus ditambahkan simbol palsu? Simbol yang saya maksud adalah spasi dan tanda baca sederhana seperti titik, koma, tanda tanya, tanda seru. Jawabannya, teks hasil Base64 adalah string panjang tidak terputus. Bagi bahasa yang tidak aglutinatif (yaitu bahasa yang menggunakan spasi sebagai pemisah kata), tidak ada kata tunggal yang sepanjang itu. Perlu dipecah-pecah supaya tidak (begitu) kelihatan aneh.

3.1. Huruf Yunani

Sejak SMP dan mulai berkenalan dengan fisika, kita sudah mengenal beberapa huruf yunani atau alfabet greek. Ada 24 huruf yunani, besar dan kecil, yaitu:
Αα Ββ Γγ Δδ Εε Ζζ Ηη Θθ Ιι Κκ Λλ Μμ
Νν Ξξ Οο Ππ Ρρ Σσ Ττ Υυ Φφ Χχ Ψψ Ωω
catatan: huruf 'sigma' kecil (σ) bila berada di akhir kata, ditulis ς bukan σ.

Saya tidak menyertakan tabel alih aksara huruf yunani ke huruf latin sebab cukup 'membingungkan'. Contohnya, beta (Ββ) dilafalkan V. Lalu eta (Ηη), upsilon (Υυ), dan iota (Ιι) sama-sama menjadi huruf I. Juga omega (Ωω) dan omicron (Οο) sama-sama menjadi huruf O. Meskipun menjadi huruf yang sama, posisi bibir dan lidah ketika mengucapkannya berbeda. Belum lagi yang "aneh" seperti gamma (γ) menjadi **g** dan zeta (ζ) menjadi **z** tetapi bila membentuk digraf γζ dibaca "gks".[11]

Bahasa Yunani juga menggunakan simbol aksen (´) di atas huruf untuk menandakan bahwa itu dibaca dengan intonasi naik. Simbol aksen ini dalam bahasa Yunani disebut τόνος (*tonos*). Huruf yang bisa terkena tonos hanya huruf vokal, yaitu α, ε, η, ι, ο, υ, dan ω. Tonos untuk huruf kecil ditulis diatas huruf, misalnya ά. Sedangkan untuk huruf besar, tonos ditulis di kiri atas huruf, misalnya ΄Ε.

Yang menarik adalah angka. Bahasa Yunani dulu juga memakai alfabet yunani itu sendiri untuk menulis angka. Bahasa Yunani modern memakai angka arab (0-9). Setiap huruf dalam alfabet mempunyai nilai angka tersendiri. Untuk 1 sampai 9, berikut angka yunani yang sesuai:

Tabel 5. Angka yunani kuno

1	2	3	4	5	6	7	8	9
α΄	β΄	γ΄	δ΄	ε΄	ς΄	ζ΄	η΄	θ΄

Tanda aksen (΄) ditulis di kanan huruf bertujuan untuk membedakan mana angka mana huruf biasa. Untuk angka 10 keatas hurufnya berbeda lagi, tidak dibahas di sini. Juga tidak ada angka nol, karena mereka menganggap nol adalah kosong (*not a thing*) dan bukan angka[4].

Jadi sekarang kita punya 48 huruf (besar dan kecil) ditambah 12 huruf ber-*tonos* ditambah 9 angka, semuanya 69 simbol. Cukup untuk menggantikan 65 simbol dari Base64.

* Saya belum menemukan referensi untuk ini. Huruf A sudah jelas paling sering muncul, sedangkan awalan *me-* (*me-*, *men-*, *mem-*, *meng-*, *meny-*, *menge-*) paling sering muncul dalam bahasa Indonesia.

Sayangnya ada masalah. Seandainya tulisan "Hello world" di halaman sebelumnya saya sandikan dengan Base64 lalu dilewatkan ke substitusi monoalfabetik dengan satu kunci yang saya pilih, tambahkan spasi dan sebagainya, hasilnya adalah ΣΓΩσβ Γη΄γδβ΄θ΄υβ ΓKή.

Saya tidak perlu memberitahu kuncinya pun Anda sudah merasa aneh. Kenanehan pertama, saya beri warna merah:

ΣΓΩσβ Γη΄γδβ΄θ΄υβ ΓKή

Huruf besar muncul berlebihan. Di tengah-tengah kata pula. Keanehan kedua, saya beri warna biru:

ΣΓΩσβ Γη΄γδβ΄θ΄υβ ΓKή

Masalah angka. Pertama, η΄ adalah angka 8 yang muncul di tengah kata. Lalu γδβ΄ bukanlah angka valid. Angka 342 dalam tulisan yunani yang benar adalah τμβ΄, bukan γδβ΄. Ketiga, terdapat angka berurutan β΄θ΄. Ini angka 2 dan 9. Dalam tulisan yunani yang benar, angka 29 yang benar adalah κθ΄. Masalah lain bisa muncul, misalnya huruf sigma σ yang muncul di akhir kata lupa diganti menjadi 'final sigma' ς .

Kesimpulannya, enkripsi ke huruf yunani masih mengundang kecurigaan. Tidak disarankan, tetapi boleh saja dipakai bila Anda menyembunyikan teks berbahasa Indonesia dan Anda mau mengecoh kriptanalis Indonesia.

3.2. Huruf kiril

Huruf kiril (cyrillic) dipakai di bangsa turunan Slavia seperti Rusia, Makedonia, Belarusia, Ukraina, Serbia, dan Bosnia. Huruf kiril lama-lama juga dipakai di bangsa selain Slavia seperti Mongolia, Kazakhtan, dan Uzbekistan.

Dari bahasa-bahasa yang memakai huruf kiril itu, rupanya ada perbedaan satu dengan yang lainnya. Contohnya, di Serbia ada huruf Ѓ ѓ sedangkan di Rusia tidak ada. Berhubung saya hanya hafal huruf kiril yang dipakai di Rusia saja, maka dalam makalah ini saya batasi hanya 33 huruf yang dipakai di Rusia.

Tabel 6. Huruf kiril Rusia dan alih aksaranya ke huruf latin

Huruf kiril	Alih aksara	Huruf kiril	Alih aksara	Huruf kiril	Alih aksara
А а	a	К к	k	Х х	kh
Б б	b	Л л	l	Ц ц	ts
В в	v	М м	m	Ч ч	ch
Г г	g	Н н	n	Щ щ	sh
Д д	d	О о	o	Ш ш	sch
Е е	ye	П п	p	Ъ ъ	-
Ё ё	yo	Р р	r	Ы ы	yi
Ж ж	zh	С с	s	Ь ь	-
З з	z	Т т	t	Э э	e
И и	i	У у	u	Ю ю	yu
Й й	y (i pendek)	Ф ф	f	Я я	ya

Beberapa catatan:

- Ada beberapa huruf kecil yang berubah bentuk ketika ditulis miring (*italic*) yaitu г, д, и, й, п, dan т yang menjadi *г, д, и, й, п,* dan *т*.
- Huruf Ъ adalah 'simbol keras'. Menandakan huruf sebelumnya harus dibaca dengan cepat dan menghentak.
- Huruf Ь adalah 'simbol jeda'. Gunanya memutus nafas pembacaan huruf sebelumnya dan huruf sesudahnya.

Sekarang kita punya 66 simbol huruf kiril, 33 huruf besar dan 33 huruf kecil. Cukup untuk menggantikan 65 simbol dari Base64. Tidak perlu lagi memakai angka kiril, karena angka kiril ternyata sistemnya sama seperti angka yunani. Malahan lebih susah karena mensyaratkan adanya garis diatas huruf-huruf angka, untuk membedakan mana angka mana huruf biasa.

Perlu diketahui bahwa dalam bahasa Rusia:

- Simbol keras (Ъ) jarang muncul.
- Simbol keras (Ъ) biasanya muncul setelah huruf S (dalam kiril: huruf C) dan sebelum huruf Я (ya), E (ye), Ё (yo), dan Ю (yu).
- Simbol jeda (Ь) sering muncul di akhir kata.
- Simbol jeda (Ь) tidak pernah ada di awal kata.
- Saya belum menemukan satu kata pun dalam bahasa Rusia, yang memiliki simbol jeda lebih dari tiga.

Simbol apa yang paling jarang muncul di Base64? Jawabannya, simbol sama dengan (=). Dia hanya muncul *paling belakang* sendiri, dan hanya satu atau dua. Lalu simbol apa yang paling jarang muncul dalam abjad kiril? Simbol keras (Ъ). Maka cocoklah bila kita menyubstitusi = dengan Ъ. Kenyataannya, jika dalam teks Rusia ditemukan kata yang berakhiran dengan Ъ, huruf itu akan dianggap Ь.

Dengan mencoret huruf Ъ dan ъ dari daftar huruf kiril dan simbol = dari Base64, tersisa 64 huruf kiril dan 64 simbol Base64. Silakan susun tabel substitusi monoalfabetik yang Anda suka. Dengan memanfaatkan informasi bahwa huruf Ь sering muncul di akhir kata, maka di situlah saat yang tepat untuk menyisipkan spasi. Namun jangan melulu memasukkan spasi setelah huruf Ь karena akan membuat kecurigaan.

Misalkan tulisan "Hello world" disandikan dengan Base64 hasilnya menjadi SGVsbG8gd29ybGQ=. Lalu substitusi ke huruf kiril, tambahkan spasi setiap mulai huruf besar, hasilnya adalah СГ Всб Гщгдх Ёйб ГЧъ. Rupanya masih ada keanehan, yaitu ГЧъ (huruf besar di tengah kata).

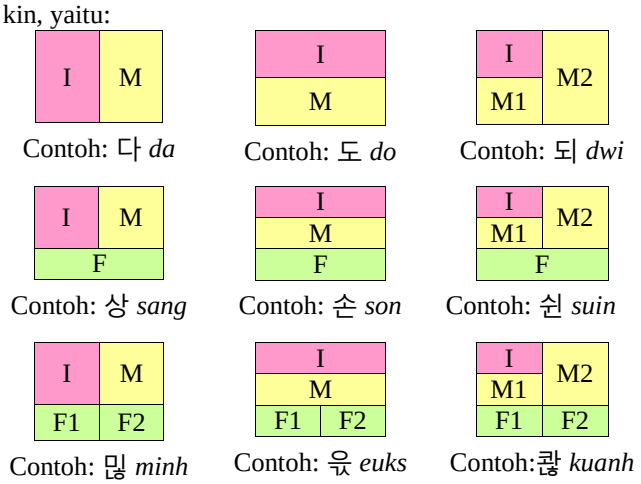
Substitusi huruf tunggal ke huruf kiril masih belum bagus, tetapi lebih baik daripada memakai huruf yunani.

3.3. Huruf hangul

Mari kita berpindah dari jenis "alfabet" ke jenis yang lain.

Bahasa Korea menggunakan dua set huruf, yaitu hangul dan hanja. Huruf hangul adalah huruf yang biasa Anda lihat di restoran-restoran Korea di kota, atau di drama-drama Korea yang bersembunyi di banyak FTP server ITB. Sedangkan huruf hanzi (Cina), kanji (Jepang), dan hanja (Korea) mengacu ke huruf yang sama, yaitu tulisan han – atau aksara Cina, lebih mudahnya. Satu suku kata dalam alfabet (latin, kiril, yunani, dsb) dibentuk oleh satu atau beberapa huruf. Sedangkan satu suku kata di bahasa Korea terdiri dari *tepat* satu huruf hangul. Beda lagi dengan Cina yang satu *kata* terdiri dari *satu* huruf.

Huruf hangul tersusun atas *jamo* (*building blocks*) yang umumnya terdiri dari tiga elemen, yaitu awalan (*initial*), penengah (*medial*), dan penutup (*final*). Elemen final tidak selalu harus ada. Elemen inisial harus ada, dan harus berupa bunyi konsonan. Elemen medial harus ada, dan harus berupa bunyi vokal. Ada 9 penempatan *jamo* yang mungkin, yaitu:



Gambar 4. Posisi penempatan *jamo*

...dengan I menyatakan inisial, M mewakili medial, dan F adalah final.

Tabel 7. Daftar jamo di bahasa Korea modern

Konsonan									
ㅂ b/p	ㄱ g/k	ㄴ n	ㄷ d	ㅅ s	ㅈ j	ㅋ k	ㅌ t	ㅊ ch	ㅍ p
ㅇ –/ng	ㄹ r/l	ㅎ h	ㅁ m						
Konsonan rangkap									
ㅃ pp	ㄲ kk	ㄸ tt	ㅆ ss	ㅉ jj					
Konsonan kombinasi — hanya boleh ada di posisi final									
ㄳ gs/ks	ㄵ nch	ㄶ nh	ㄺ lg	ㄻ lm	ㄼ lb	ㄽ ls	ㄾ lt	ㄿ lp	ㅀ lh
ㅄ bs									
Vokal									
ㅏ a	ㅐ ae	ㅣ i	ㅓ eo	ㅔ é	ㅡ eu	ㅗ o	ㅜ u		
Vokal + y									
ㅑ ya	ㅒ yae		ㅕ yeo	ㅖ ye		ㅛ yo	ㅠ yu		
Diftong/vokal ganda									
ㅘ ua (wa)	ㅙ uae (wae)	ㅚ oe (we)	ㅝ ueo (weo)	ㅞ ue (we)	ㅟ ui (wi)	ㅢ eui (yi)			

catatan: jamo *ieung* atau "bulat" (ㅇ) saya tulis "–/ng" karena tidak berbunyi bila berada di posisi inisial. Ini karena setiap huruf hangul harus diawali dengan konsonan, sehingga bila memang tidak ada awalan konsonannya maka digunakanlah ㅇ. Untuk menulis "a" saja, tidak boleh hanya ㅏ saja, tetapi harus 아. Bila ㅇ berada di posisi final, ia berbunyi *ng*.

Jumlah jamo tersedia hanya 51. Masih kurang untuk menutupi 65 simbol dari Base64, jadi kita tidak bisa mengganti satu simbol Base64 dengan satu jamo. Tetapi, kita menggunakan huruf hangul yang sudah jadi saja. Huruf hangul yang sudah jadi tersedia dari unicode U+AC00 yaitu 가 *ga*, sampai U+D7A3 힣 *hih*. Semuanya berjumlah 11.172 huruf. Jumlah ini memungkinkan satu simbol Base64 diganti dengan banyak huruf hangul, sampai maksimal 171 huruf.

Tabel 8. Contoh tabel substitusi ke huruf hangul

Base64	*Substituen*
A	뀳 껇 퓛 튄 쿬 캴 츧 쨶 줌 유 웕 엾 숱
B	끫 굣 흇 혊 퓐 캿 봜 옹 숡 옛 옟 쌶 뼜
C	쓮 가 휟 헾 퉎 커 뤜 썳 찏 웹 둗 뿏 으
D	꿁 쭁 휂 혫 튕 빦 볹 숴 쬳 뭛 왞 묞 봖
…	…

Misalkan tulisan "Hello world" tadi saya enkripsi. Salah satu contoh cipherteksnya adalah 킠아응더나몁늈쌍뚬렌죠언끈시뫇팡, saya rasa tidak ada yang curiga. Tiada lagi masalah "huruf besar di tengah kata" seperti pada huruf yunani dan huruf kiril sebelumnya.

Enkripsi memakai huruf hangul dengan strategi ini cukup kuat, cukup menipu kriptanalis yang tidak mengerti bahasa Korea. Alternatifnya, Anda bisa menggunakan huruf hanzi dengan strategi yang sama. Ada sekitar 40.000 huruf hanzi, namun hanya 5089 huruf[14] yang umum dipakai di Cina.

3.4. Huruf Jepang

Bahasa Jepang ditulis dengan mencampur tiga set huruf, yaitu hiragana, katakana, dan kanji. Huruf kanji diimpor dari Cina pada abad ke-5 masehi, dan hiragana katakana sebenarnya evolusi dari kanji. Huruf hiragana dan katakana disebut huruf kana.

Tabel 9. hiragana

あ	か	が	は	ば	ぱ	ま	な	た	だ	ら	さ	ざ	や	わ
a	ka	ga	ha	ba	pa	ma	na	ta	da	ra	sa	za	ya	wa
い	き	ぎ	ひ	び	ぴ	み	に	ち	ぢ	り	し	じ		
i	ki	gi	hi	bi	pi	mi	ni	chi	ji*	ri	shi	ji		
う	く	ぐ	ふ	ぶ	ぷ	む	ぬ	つ	づ	る	す	ず	ゆ	
u	ku	gu	fu	bu	pu	mu	nu	tsu	dzu	ru	su	zu	yu	
え	け	げ	へ	べ	ぺ	め	ね	て	で	れ	せ	ぜ		
e	ke	ge	he	be	pe	me	ne	te	de	re	se	ze		
お	こ	ご	ほ	ぼ	ぽ	も	の	と	ど	ろ	そ	ぞ	よ	を
o	ko	go	ho	bo	po	mo	no	to	do	ro	so	zo	yo	wo
							ん							
							n							

Tabel 10. katakana

ア	カ	ガ	ハ	バ	パ	マ	ナ	タ	ダ	ラ	サ	ザ	ヤ	ワ
a	ka	ga	ha	ba	pa	ma	na	ta	da	ra	sa	za	ya	wa
イ	キ	ギ	ヒ	ビ	ピ	ミ	ニ	チ	ヂ	リ	シ	ジ		
i	ki	gi	hi	bi	pi	mi	ni	chi	ji*	ri	shi	ji		
ウ	ク	グ	フ	ブ	プ	ム	ヌ	ツ	ヅ	ル	ス	ズ	ユ	
u	ku	gu	fu	bu	pu	mu	nu	tsu	dzu	ru	su	zu	yu	
エ	ケ	ゲ	ヘ	ベ	ペ	メ	ネ	テ	デ	レ	セ	ゼ		
e	ke	ge	he	be	pe	me	ne	te	de	re	se	ze		
オ	コ	ゴ	ホ	ボ	ポ	モ	ノ	ト	ド	ロ	ソ	ゾ	ヨ	ヲ
o	ko	go	ho	bo	po	mo	no	to	do	ro	so	zo	yo	wo
							ン							
							n							

*huruf *ji* ぢ dan ヂ hanya dipakai pada kasus *rendaku* (連濁) saja.

Selain itu masih ada kana kombinasi, yaitu menambah ゃ (*ya* kecil), ゅ (*yu* kecil), dan ょ (*yo* kecil) untuk hiragana. Untuk katakana, lebih banyak lagi kombinasinya. Teks Jepang hampir selalu mengandung kana kombinasi, sehingga terasa aneh bila cipherteks tidak mengandung huruf ini.

Tabel 11. hiragana kombinasi

	+ や	+ ゆ	+ よ
k-	きゃ kya	きゅ kyu	きょ kyo
g-	ぎゃ gya	ぎゅ gyu	ぎょ gyo
h-	ひゃ hya	ひゅ hyu	ひょ hyo
b-	びゃ bya	びゅ byu	びょ byo
p-	ぴゃ pya	ぴゅ pyu	ぴょ pyo
m-	みゃ mya	みゅ myu	みょ myo
n-	にゃ nya	にゅ nyu	にょ nyo
ch-	ちゃ cha	ちゅ chu	ちょ cho
sh-	しゃ sha	しゅ shu	しょ sho
j-	じゃ ja	じゅ ju	じょ jo
r-	りゃ rya	りゅ ryu	りょ ryo

Tabel 12. katakana kombinasi

	-a	-i	-u	-e	-o
k-	キャ kya		キュ kyu	キェ kye	キョ kyo
g-	ギャ gya		ギュ gyu	ギェ gye	ギョ gyo
h-	ヒャ hya		ヒュ hyu	ヒェ hye	ヒョ hyo
f-	ファ fa	フィ fi		フェ fe	フォ fo
b-	ビャ bya		ビュ byu	ビェ byé	ビョ byo
p-	ピャ pya		ピュ pyu	ピェ pye	ピョ pyo
m-	ミャ ya		ミュ myu	ミェ mye	ミョ myo
n-	ニャ nya		ニュ nyu	ニェ nye	ニョ nyo
ch-	チャ cha		チュ chu	チェ che	チョ cho
sh-	シャ sha		シュ shu	シェ sye	ショ sho
j-	ジャ ja		ジュ ju	ジェ je	ジョ jo
r-	リャ rya		リュ ryu	リェ rye	リョ ryo
t-		ティ ti	トゥ tu		
d-		ディ di	ドゥ du		
z-		ゼィ zi			

Hiragana ada 71 + 33 = 104 buah, cukup untuk mengganti 65 simbol Base64. Selain itu masih ada 2232 huruf[14] kanji yang dipakai sehari-hari di Jepang. Kita bisa mengganti satu simbol Base64 dengan beberapa huruf, campuran dari hiragana dan kanji.

Huruf katakana sejatinya dipakai untuk menulis kata asing (kata serapan) dalam Bahasa Jepang, termasuk nama orang asing. Misalnya kata "*inkjet printer*" ditulis dengan katakana menjadi インクジェットプリンター (*inkujetto purintaa*). Yang ingin saya tekankan disini adalah, *huruf katakana tidak pernah muncul sendirian* kecuali untuk menulis *iroha*. Apa itu *iroha* tidak akan dibahas di sini. Dengan memperhatikan properti ini, kita bisa mengurangi keanehan cipherteks dengan cara menyorot (seleksi) sederetan cipherteks *hiragana-only* lalu mengubahnya ke katakana.

Ada simbol perpanjangan vokal (*chou-on*) untuk katakana, bentuknya garis panjang ー。*Tanda chou-on tidak pernah muncul di awal kata dan tidak pernah muncul setelah huruf n* (ン). Selain itu ada tanda perpanjangan konsonan (*sokuon*), yaitu huruf 'tsu' kecil (っ untuk hiragana dan ッ untuk katakana). Fungsinya menggandakan konsonan milik huruf sesudahnya. Contoh, ふき *fuki* menjadi ふっき *fu**k**ki* dan きて *kite* menjadi きって *ki**t**te*. Sekarang, peraturannya adalah *tanda sokuon tidak pernah muncul sebelum huruf vokal dan huruf n* (ん)*, dan juga tidak pernah muncul di awal kata/kalimat*.

Sampai di tahap ini, tabel substitusi sudah jadi, cipherteks juga sudah terbentuk. Untuk mengurangi kecurigaan, tambahkan simbol-simbol palsu yang terdiri dari:

- Huruf kanji yang tidak didalam tabel substitusi.
- Tanda *chou-on* dan *sokuon* (perhatikan *constraint* di paragraf sebelumnya)

- Tanda titik (。), koma (、), tanda tanya atau seru. Boleh menggunakan tanda kurung siku Jepang 「seperti ini」. Tanda ini berfungsi seperti tanda kutip di teks berhuruf latin. Untuk tanda tanya, tempat meletakkannya harus setelah huruf hiragana *ka* (か), karena dalam bahasa Jepang partikel *ka* berfungsi membentuk kalimat tanya.
- Jangan tambahkan spasi! Bahasa Jepang adalah bahasa aglutinatif, yaitu bahasa yang tidak menggunakan batas antar kata.

Paragraf ini tentang penyisipan huruf kanji palsu (yang tidak ada di tabel substitusi). Mungkin ada yang bertanya, kanji mana yang cocok untuk disisipkan? Tujuan kita adalah mengurangi kecurigaan, bukan menerjemahkan cipherteks ke dalam bahasa Jepang. Sisipkan kanji apa saja, tanpa perlu memperhatikan huruf-huruf di belakangnya. Dalam bahasa Jepang sebenarnya, huruf hiragana yang mengikuti kanji bisa mengubah cara pembacaan kanji tersebut. Hal ini tidak usah dipedulikan dalam enkripsi ini.

Misalkan "Hello world" dilewatkan ke Base64 lalu disubstitusi ke huruf Jepang, ditambahkan kanji palsu dan sebagainya, anggap saja hasilnya 博ちむじゃぼたマージメ合わっせ真「青」そら。Cipherteks sudah tak terlihat aneh.

4. KESIMPULAN

Enkripsi dengan substitusi huruf tunggal ke huruf non-latin bertujuan untuk menyembunyikan cipherteks supaya tidak terbaca orang sekitar. Kita tidak peduli apakah orang yang membaca cipherteks kita mengerti bahasa asing yang kita pakai atau tidak. Andaikan orang Indonesia mengenkripsi teks berbahasa Indonesia ke huruf hanzi misalnya, lalu ada temannya yang mengerti bahasa Kanton (*Cantonese*, salah satu bahasa di Cina) kebetulan melihat cipherteksnya, biarkan saja ia tertawa karena jelas teksnya tidak bisa diartikan ke bahasa natural yang benar, yang punya makna. Yang diutamakan adalah plainteksnya aman tersembunyi.

Dalam makalah ini tidak diteliti tentang *dictionary-based monoalphabetic substitution*, karena:

1. Cipherteks yang punya makna (valid secara semantik dalam bahasa tujuan) boleh terlihat keren, tetapi implementasinya sulit sekali.
2. Ukuran file kamus sangat besar. Contohnya, kamus bahasa Jepang-Inggris yang saya punya yaitu EDICT, berukuran 15 MB. Mencari teks dengan Boyer-Moore memang cepat, tetapi mencari didalam teks sebanyak 15 MB itu tetap membutuhkan waktu yang lama.
3. Tujuan kita adalah membuat cipherteks terlihat se-'natural' mungkin, bukan menerjemahkan cipherteks. Jika memang yang diinginkan adalah enkripsi plus alih bahasa, maka gunakanlah kode (substitusi kata dalam *codebook*) lalu terjemahkan cipherwords ke dalam bahasa tujuan. Misalnya plainteks "kumpul jam lima" disandikan menjadi "ayo cepat bangun", terjemahkan saja ke dalam bahasa Jepang menjadi「早く起きなさいよ」、 Anda sudah dapat yang Anda inginkan.

Substitusi ke huruf non-latin mensyaratkan huruf tujuan mempunyai simbol *minimal sama jumlahnya* dengan huruf plain/cipherteks asal. Dalam makalah ini Base64 menggunakan huruf latin 46 huruf, ditambah angka dan 3 simbol, sehingga huruf tujuan harus yang punya minimal 65 simbol. Telah dibahas huruf yunani yang hanya 24 huruf, lalu huruf kiril yang 33 huruf, hangul, dan terakhir huruf-huruf Jepang. Anda tidak terbatas hanya 4 huruf ini saja. Masih banyak huruf atau sistem penulisan yang bisa dieksplorasi, misalnya

- Huruf hanzi (Cina), sekitar 5000 huruf,
- Huruf hanacaraka (Jawa), ada 20 huruf dasar + 20 *modifier* pembentuk huruf mati + 5 aksara *swara* + 5 aksara *rekam* + 8 aksara *murda* + 10 angka + 12 *sandhangan*. Menarik.
- Huruf thai (Thailand)
- Huruf asyiria/ibrani (Israel)

Atau mungkin ada yang tertarik meneliti sistem penulisan yang kompleks, seperti

- Huruf hijayah (Arab)
- Huruf devanagari (India, Nepal)

REFERENSI

Sumber pustaka

[1] Rinaldi Munir, *Kriptografi*. 2005, Departemen Teknik Informatika ITB

Sumber internet

[2] *Bahasa Rusia*
http://yusufat.blogspot.com/2008/07/bahasa-rusia.html#hurufcyril
[3] Learn Korean Online – *Learn hangul*
http://www.learnkoreanlanguage.com/learn-hangul.html
[4] *Greek Numerals*
http://www.foundalis.com/lan/grknum.htm

Sumber wikipedia

[5] http://en.wikipedia.org/wiki/hangul
[6] http://en.wikipedia.org/wiki/Base64
[7] http://en.wikipedia.org/wiki/Cyrillic_numerals
[8] http://en.wikipedia.org/wiki/Greek_alphabet
[9] http://en.wikipedia.org/wiki/Polytonic_orthography
[10] http://en.wikipedia.org/wiki/Modern_Greek
[11] http://en.wikipedia.org/wiki/Romanization_of_Greek
[12] http://en.wikipedia.org/wiki/Post_office_protocol
[13] http://en.wikipedia.org/wiki/Roman_Empire

Software pembantu

[14] *Wakan* 和漢, *freeware tool for learning Japanese and Chinese*. © Filip Kábrt, 2003. http://wakan.manga.cz