Penggunaan Fungsi *Hash Whirlpool* untuk Menghasilkan Kode Pengenal Baru dari Sidik Jari

Adiputra Sejati | 13507105[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1] e-mail: if17105@students.if.itb.ac.id; adiputra.sejati@yahoo.com

ABSTRAK

Seiring dengan pertumbuhan penduduk negara Indonesia yang sangat yang sangat pesat, data kependudukan menjadi salah satu masalah dimana terjadi penyalahgunaan terhadap kartu tanda pengenal tersebut . Kurangnya suatu hal yang unik dari sebuah KTP menjadi permasalahan pada kasus ini. Sidik jari adalah salah satu objek yang dimiliki setiap orang dimana sidik jari seseorang tidak mungkin sama dengan sidik jari orang lain. Akan tetapi apabila sidik jari dibubuhkan kedalam KTP, sidik jari tersebut tidak dapat dibedakan secara langsung dengan sidik jari orang lain.
Dalam makalah ini penulis akan membahas bagaimana menggunakan fungsi *Hash* untuk menghasilkan rangkaian kombinasi suatu huruf dan angka yang unik dengan masukkan sidik jari seseorang. Hasil dari fungsi ini digunakan sebagai suatu hal yang unik yang dapat ditambahkan dalam sebuah KTP.
Fungsu *Hash* adalah sebuah fungsi yang terdapat dalam pelajaran kriptografi dimana kriptografi merupakan sebuah ilmu untuk mengamankan sebuah pesan digital. Fungsi *Hash* merupakan fungsi yang dapat menerima sebuah masukan data dan akan mengeluarkan hasil berupa string acak. Fungsi *Hash* juga bersifat satu arah dimana hasil yang dihasilkan tidak dapat dikembalikan ke bentuk semula (*non-reversibel*). Fungsi *Hash* yang digunakan adalah fungsi *Hash WHIRLPOOL*, sehingga penulis mengharapkan dengan hasil dari fungsi ini dapat menghasilkan objek yang unik sehingga objek ini dapat dibubuhkan dalam sebuah KTP.

Kata kunci: Fungsi *Hash*, Kriptografi, *Hash WHIRLPOOL*, KTP(Kartu Tanda Penduduk).

1. PENDAHULUAN

Dalam suatu negara besar seperti Indonesia memiliki penduduk yang sangat banyak. Setiap penduduk memiliki kartu tanda pengenal yang mengidentifikasikan masing-masing personal. Nomor kependudukan yang tertera pada KTP berdasar pada tempat tinggal dan waktu pada saat mereka membuat KTP. Karena hal ini dapat menyebabkan adanya seorang penduduk yang memiliki KTP ganda yang dapat digunakan tidak dengan semestinya. Tetapi inti dari permasalahan hal ini adalah kurangnya suatu hal yang unik dari sebuah KTP tersebut. Sidik jari sesorang merupakan suatu hal yg unik dimana setiap orang memiliki sidik jari yang berbeda, akan tetapi apabila menggunakan sidik jari dalam sebuah KTP, sidik jari tersebut tidak dapat dibedakan secara langsung dengan sidik jari orang lain. Oleh karena itu pada makalah ini penulis ingin memberikan sebuah solusi untuk memberikan sebuah rangkaian nomor yang unik yang dapat digunakan sebagai nomor identifikasi dari seorang penduduk. Ditambah lagi pada masa kedatang akan digunakan e-KTP sebagai kartu tanda penduduk yang baru dimana dalam e-KTP ini terdapat sebuah chip yang digunakan untuk menyimpan data unik dari seseorang.

Dalam makalah ini penulis memberikan solusi dengan menggunakan salah satu fungsi dalam kriptografi, yaitu fungsi *Hash*. Fungsi hash yang digunakan adalah fungsi *hash WHIRLPOOL*, sehingga dengan fungsi ini dapat menghasilkan sebuah nomor unik dari setiap penduduk yang dapat digunakan sebagai kartu tanda penduduk medatang.

2. LANDASAN TEORI

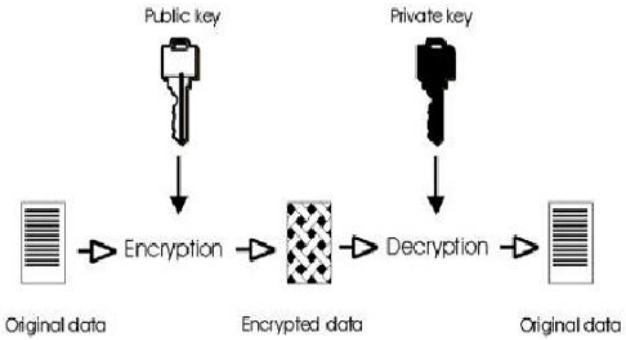
2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, yang terdiri dari dua kata yaitu kripto dan graphia. Kripto berarti rahasia(*secret*) sedangkan graphia berarti tulisan(*writing*). Sehingga kriptografi bisa diartikan sebagai “tulisan yang dirahasiakan”. Dalam kamus *hacker* (Ariyus, 2005),

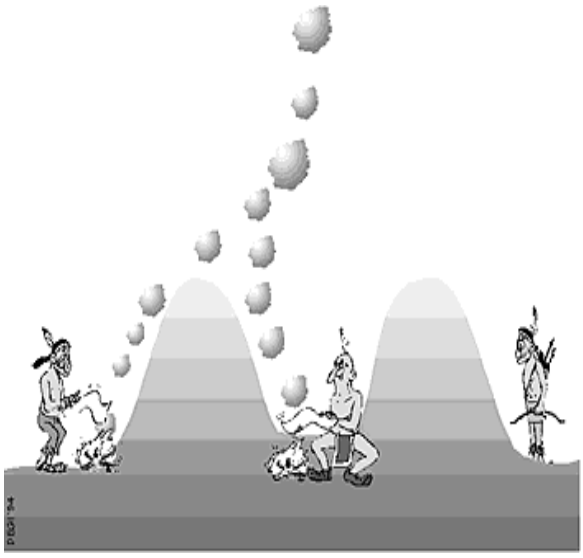
kriptografi diartikan sebagai ilmu yang mempelajari penulisan secara rahasia. Secara umum kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan. Selain itu, kriptografi juga bisa diartikan sebagai ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentifikasi data. Tidak semua aspek keamanan informasi dapat ditangani oleh kriptografi.

Dalam sebuah algoritma kriptografi, terdapat tiga unsur yaitu :

- Enkripsi, yaitu proses mengubah plaintext menjadi *ciphertext*
- Dekripsi, yaitu mengubah ciphertext menjadi *plaintext*
- Kunci, merupakan *key* yang digunakan untuk proses enkripsi maupun proses dekripsi.



Gambar 1. Skema alrian proses pada metode Kriptografi



Gambar 2. Ilustrasi serangan dari yang ada pada Kriptografi

2.2 Kartu Tanda Penduduk (KTP)

Kartu Tanda Penduduk (KTP) adalah identitas resmi Penduduk sebagai bukti diri yang diterbitkan oleh Instansi Pelaksana yang berlaku di seluruh wilayah Negara Kesatuan Republik Indonesia. Kartu ini wajib dimiliki bagi Warga Negara Indonesia (WNI) dan Warga Negara Asing (WNA) yang memiliki Izin Tinggal Tetap (ITAP) yang sudah berumur 17 tahun atau sudah pernah kawin atau telah kawin. Anak dari orang tua WNA yang memiliki ITAP dan sudah berumur 17 tahun juga wajib memilki KTP. KTP bagi WNI berlaku selama lima tahun dan tanggal berakhirnya disesuaikan dengan tanggal dan bulan kelahiran yang bersangkutan. KTP bagi WNA berlaku sesuai dengan masa Izin Tinggal Tetap. Khusus warga yang telah berusia 60 tahun dan ke atas, mendapat KTP seumur hidup yang tidak perlu diperpanjang setiap lima tahun sekali.

KTP berisi informasi mengenai sang pemilik kartu, termasuk:

- Nomor Induk Kependudukan (N.I.K.)
- nama lengkap
- tempat dan tanggal lahir
- jenis kelamin
- agama
- status perkawinan
- golongan darah
- alamat
- pekerjaan
- kewarganegaraan
- foto
- masa berlaku
- tempat dan tanggal dikeluarkan KTP
- tandatangan pemegang KTP
- nama dan nomor induk pegawai pejabat yang menandatanganinya

Gambar 3. Bentuk KTP yang sekarang ada di negara kita

2.3 e-KTP

e-KTP adalah kepanjangan dari elektronik KTP, elektronik KTP merupakan salah satu pengembangan teknologi yang diberlakukan pada Kartu Tanda Penduduk kita yang biasa kita gunakan. E-KTP didesain dengan metode autentikasi dan pengamanan data tinggi. Hal ini dapat dicapai dengan menanamkan chip di dalam kartu yang memiliki kemampuan autentikasi, enkripsi dan tanda tangan digital. Autentikasidua arah dilakukan antara kartu elektronik dan perangkat pembacanya agar kartu dan pembaca dapat dipastikan sah. Sementara enkripsi digunakan untuk melindungi data yang tersimpan di dalam kartu elektronik dan tanda tangan digital untuk menjaga integritas data. Disamping itu e-KTP dilindungi dengan keamanan pencetakan seperti relief text, microtext, filter image, invisible ink dan warna yang berpendar di bawah sinar ultra violet serta anti copy design.

Tanda tangan terdigitalisasi penduduk juga disimpan di dalam rekaman elektronik berupa chip. Perekaman sidik jari dilakukan terhadap 10 sidik jari tangan yang disimpan pada basis data dan dua buah sidik jari tangan yaitu jari telunjuk kanan dan kiri pada chip kartu.

Penerapan awal KTP berbasis NIK yang dilengkapi dengan sidik jari dan chip merupakan langkah strategis menuju tertib administrasi kependudukan yang mengamanatkan adanya identitas tunggal bagi setiap penduduk dan terbangunnya basis dara kependudukan yang lengkap dan akurat.

Gambar 4. Contoh bentuk electronic id card

2.3 Fungsi Hash

Fungsi *Hash* adalah fungsi yang menerima masukkan string yang panjangnya sembarang dan mengkonversinya menjadi string keluaran yang panjangnya tetap.

Fungsi *Hash* dapat menerima masukan string apa saja. Jika string menyatakan pesan, maka sembarang pesan M berukuran bebas dikompresi oleh fungsi hash H melalui persamaan $h = H(M)$. Keluaran dari fungsi Hash disebut juga *message digest*. Pada persamaan diatas *h* adalah nilai hash atau *message digest* dari fungsi H untuk masukkan M. Dengan kata lain, fungsi hash mengkompresi sembarang pesan yang berukuran beraa saja menjadi *message digest* yang ukurannya selalu tetap.

Fungsi *Hash* sering juga disebut fungsi enkripsi satu arah, atau disebut juga dengan *message digest.* Fungsi hash digunakan untuk menjamin servis autentikasi dan integritas suatu pesan atau file. Ide utama dari fungsi hash adalah bahwa suatu nilai hash berlaku sebagai representasi dari data secara sederhana (disebut juga *message digest*, *imprint*, *digital finger print*) dari suatu input string, dan dapat digunakan hanya jika nilai hash tersebut dapat diidentifikasi secara unik dengan input string tersebut. Fungsi h adalah *many-to-one*, sehingga memungkinkan terjadinya pasangan input dengan output yang sama(*collision*).

Sifat-sifat fungsi *hash* satu-arah adalah sebagai berikut:

1. Fungsi *H* dapat diterapkan pada blok data berukuran berapa saja.
2. *H* menghasilkan nilai (*h*) dengan panjang tetap (*fixed-length output*).
3. $H(x)$ mudah dihitung untuk setiap nilai *x* yang diberikan.
4. Untuk setiap *h* yang dihasilkan, tidak mungkin dikembalikan nilai *x* sedemikian sehingga $H(x) = h$. Itulah sebabnya fungsi *H* dikatakan fungsi *hash* satu-arah (*oneway hash function*).
5. Untuk setiap *x* yang diberikan, tidak mungkin mencari $y \neq x$ sedemikian sehingga $H(y) = H(x)$.
6. Tidak mungkin mencari pasangan *x* dan *y* sedemikian sehingga $H(x) = H(y)$.

2.3 Fungsi Hash WHIRLPOOL

Algoritma *Hash* WHIRLPOOL dirancang oleh Vincent Rijmen(salah satu perancang algoritma *AES*) dan Paulo SLM Barreto. Ukuran output dari algoritma ini adalah 512 bit. Versi pertama algoritma ini adalah Whirlpool-0 yang dibuat pada November 2000. Versi kedua disebut Whirlpool-T yang dipilih untuk NESSIE (*New European Schemes for Signatures, Integrity and Encryption*).Versi ketiga fungsi ini diadopsi oleh *International Organization for Standardization* (ISO) dan IEC dalam ISO / IEC 10118-3:2004.

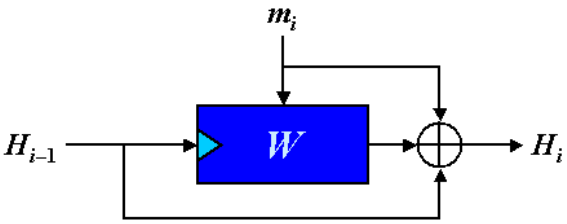
Gambar 4. Gambar yang memunculkan ide Hash Whirlpool

Fungsi *Whirlpool* menggunakan penguatan Merkle-Damgård dan skema *hash* dari Miyaguchi-Preneel yang menggunakan 512 bit blok chiper yang disebut dengan W.

Fungsi Whrilpool terdiri dari aplikasi iterasi dari fungsi compresi, berdasarkan pada 512-bit block cipher yang menggunakan kunci 512-bit. Fungsi round dan jadwal kunci yang dirancang sesuai dengan strategy wide trail. Implementasi fungsi Hash Whirlpool tidak berorientasi terhadap platform tertentu.

Cara kerja fungsi Hash Whirlpool adalah sebagai berikut.

1. Bit string yang akan di hash ditambahkan dengan sebuah ‘&lquo’bit-‘1’ kemudian dengan urutan bit-‘0’ dan dengan panjang aslinya (dalam bentuk integer 256 bit), sehingga panjang setelah dipadding menjadi kelipatan 512 bit.
2. Hasil pesan string yang dihasilkan dibagi menjadi blok-blok m1,m2,m3,...,mt dimana setiap bloknya memiliki ukuran 512 bit.
3. Hasil pesan diatas akan dihash dan menghasilkan *message digest* H0,H1,H2,H3,...,Ht. *Message digest* H0 merukapakan string bit-‘0’ sepanjang 512 bit.
4. Untuk menghitung Hi, W akan mengenkripsi mi dengan menggunakan Hi-1 sebagai kuncinya dan ciphertext yang dihasilkan akan di XOR dengan Hi-1 dan mi. Dan pada akhirnya hasil dari hash WHRILPOOl ini adalah Ht.



Gambar 5. Miyaguchi-Preneel compression function

W Blok Cipher yang digunakan oleh Whirlpool sangat mirip dengan algoritma AES,Rijndael. Perbedaan dari keduanya adalah sebagai berikut:

- **Ukuran block (bits)**
  Rijndael : 128, 160, 192, 224, or 256
  W : selalu 512

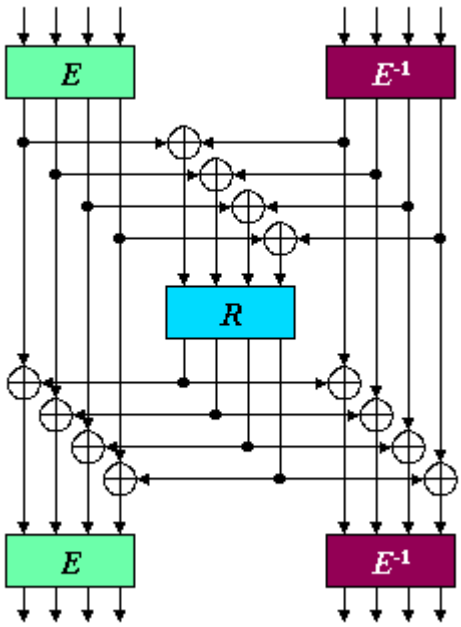
- **Banyak putaran**
  Rijndael : 10,11,12,13, atau 14
  W : selalu 10

- **Jadwal Key**
  Rijndael : algoritma prioritas
  W : pada putaran sendiri

- **GF(28) reduction polynomial**
  Rijndael : $x^8 + x^4 + x^3 + x + 1$ (0x11B)
  W : $x^8 + x^4 + x^3 + x^2 + 1$ (0x11D)

- **Asal S-box**
  Rijndael : mapping $u \rightarrow u^{-1}$ over GF(2^8), ditambah dengan affine transform
  W : rekursif
- **Asal dari konstanta putaran**
  Rijndael : polynomials x^i over GF(2^8)
  W : berturut-turut masukan dari S-box

- **Diffusion Layer**
  Rijndael : perkalian kiri dengan 4×4 circulant MDS matrix cir(2, 3, 1, 1)
  W : perkalian kanan dengan 8×8 circulant MDS matrix cir(1, 1, 4, 1, 8, 5, 2, 9)



Gambar 6. The recursive structure of the "tweaked" *S*-box

2.4 Primitive fungsi WHIRLPOOL

Primitive fungsi *Whirlpool* adalah fungsi *hashing Merkle* yang didasarkan pada sebuah blok cipher W yang beroperasi pada 512 bit *hash state* menggunakan kunci yang berantai baik yang berasal dari masukkan data.

Berikut ini adalah cara menentukan pemetaan komponen dan konstanta yang membangun fungsi *Whirlpool*.

- Input dan output
  State pada *hash* secara internal dilihat sebagai $M_{8\times8}[GF(2^8)]$. Oleh karena itu 512 bit blok data bit harus dipetakkan dari dan ke format matriks. Hal ini dilakukan oleh fungsi
  $$\mu : \mathrm{GF}(2^8)^{64} \rightarrow \mathcal{M}_{8\times8}[\mathrm{GF}(2^8)]$$
  Dan inversnya :
  $$\mu(a) = b \Leftrightarrow b_{ij} = a_{8i+j},\ 0 \leqslant i,j \leqslant 7$$

- Layer *non-linear* γ
  Fungsi γ : $M_{8\times8}[GF(2^8)] \rightarrow M_{8\times8}[GF(2^8)]$ terdiri dari aplikasi paralel dari sebuah subtitusi *non-linear* box S : $GF(2^8) \rightarrow GF(2^8)$, x ↔S[x] untuk semua byte secara terpisah : →←↔
  $$\gamma(a) = b \Leftrightarrow b_{ij} = S[a_{ij}],\ 0 \leqslant i,j \leqslant 7.$$

- Permutasi siklik π
  Permutasi π : $M_{8\times8}[GF(2^8)] \rightarrow M_{8\times8}[GF(2^8)]$ bergeser secara siklik setiap kolom argumen secara independen, sehingga kolom j digeser ke sebanyak j posisi.
  $$\pi(a) = b \Leftrightarrow b_{ij} = a_{(i-j) \bmod 8,j},\ 0 \leqslant i,j \leqslant 7.$$

- Layer difusi *linear* θ
  Difusi *linear* θ : $M_{8\times8}[GF(2^8)] \rightarrow M_{8\times8}[GF(2^8)]$ adalah sebuah pemetaan linear berdasar pada [16,8,9] MDS code dengan generator matriks G_C = [I C] dimana C = cir(01_x, 01_x, 04_x, 01_x, 08_x, 05_x, 02_x, 09_x), maka θ(a) = b ↔ b = a . C

- Kunci tambahan σ[k]
  Kunci tambahan σ[k] : $M_{8\times8}[GF(2^8)] \rightarrow M_{8\times8}[GF(2^8)]$ terdiri dari selain bitwise dari matriks kunci k∈ $M_{8\times8}[GF(2^8)]$:
  $$\sigma[k](a) = b \Leftrightarrow b_{ij} = a_{ij} \oplus k_{ij},\ 0 \leqslant i,j \leqslant 7.$$

- Penjadwalan kunci
  Penjadwalan kunci memperluas 512 bit cipher key k ∈ $M_{8\times8}[GF(2^8)]$ dalam sebuah sequence dari round key K^0,.....,K^R.
  $$K^0 = K,$$
  $$K^r = \rho[c^r](K^{r-1}),\ r > 0,$$

- Internal *Blok Cipher* W
  Internal *blok cipher* W[K] : $M_{8\times8}[GF(2^8)] \rightarrow M_{8\times8}[GF(2^8)]$ dengan parameter 512 *bit cipher key* K didefinisikan sebagai :
  $$W[K] = \left(\bigcirc_{1}^{r=R} \rho[K^r]\right) \circ \sigma[K^0],$$
  Dimana *round key* K^0,.....,K^R berasal dari K dari jadwal kunci. Default putarannya R = 10.

- Padding dan penguatan MD
  Sebelum menjadi sasaran operasi *hashing*, sebuah pesan M dengan panjang L< 2^{256} dipadding dengan 1bit kemudian dengan dengan 0bit yang diperlukan untuk memperoleh bit string dengan kelipatan 256 dan akhirnya sampai dengan 256bit benar-benar merupakan representasi biner L, sehingga message yang di padding hasilnya adalah m, yang dipartisi kedalam t blok m_1,...,m_t. Blok ini dipandang sebagai array byte dengan urutan grup dalam potongan 8bit.

- Fungsi Kompresi
  Whirlpool mengiterasi skema hash dari Miyaguchi-Preneel atas blok *message* yang dipadding m_i, 1 ≤ i ≤ t, menggunakan 512 bit *blok cipher* W :
  $$\eta_i = \mu(m_i),$$
  $$H_0 = \mu(IV),$$
  $$H_i = W[H_{i-1}](\eta_i) \oplus H_{i-1} \oplus \eta_i,\ 1 \leqslant i \leqslant t,$$
  Dimana IV(*Initial vector*) adalah sebuah string 512 0-bit

- Komputasi *Message Digest*
  Message digest dari fungsi whirlpool dari sebuah message M didefinisikan sebagai hasil dari H_t pada fungsi kompresi.
  $$\textsc{Whirlpool}(M) \equiv \mu^{-1}(H_t).$$

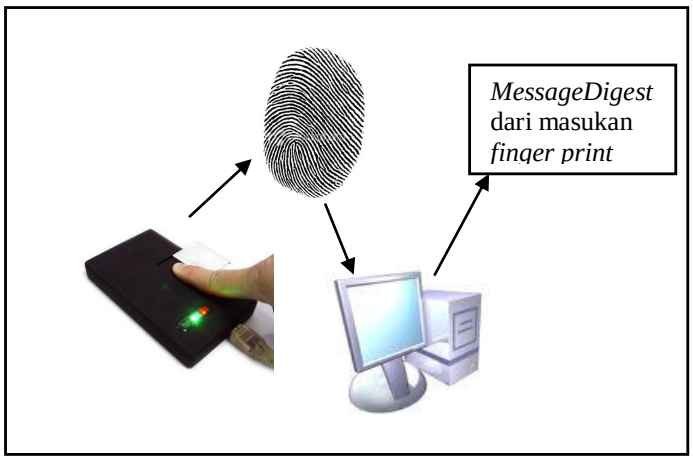
3 IMPLEMENTASI

Untuk pengimplementasian yang sebenarnya, membutuhkan alat *fingerprint* untuk mendapatkan sidik jari dari seseorang dan sidik jari tersebut akan diproses dengan fungsi *hash* yang telah dibuat oleh penulis dan akan menghasilkan *message digest* sidik jari tersebut.

3.1 Arsitektur Fungsi

Sesuai dengan proposal yang dibuat penulis mengasumsikan telah mendapat gambar dari *fingerprint* seseorang dan akaan memprosesnya menjadi kode unik

yang dapat dibubuhkan kedalam KTP atau mungkin e-KTP.



Gambar 7. Arsitektur generator sidik jadi

3.2 Proses Kerja

Proses kerja dari generator id dari *fingerprint* ini adalah sebagai berikut :

1. Jari diletakkan pada alat pendeteksi *fingerprint*
2. *Fingerprint* yang dihasilkan diproses dengan fungsi hash yang dibuat
3. *Message digest* keluaran fungsi ini digunakan sebagai kode unik pada KTP atau masukan pada e-KTP

3.3 Contoh Proses

Masukkan sidik jari I :

Proses Algoritma I :

Masukkan sidik jari II :

Proses Algoritma II :

Masukkan sidik jari III :

Proses Algoritma III :

4. ANALISIS

Fungsi *Hash Whirlpool* ini sudah dapat dianggap aman karena skema *Hash* dari Miyaguchi-Preenel adalah salah satu dari beberapa metode untuk menyusun fungsi hash yang belum terpecahkan dari masukkan blok cipher. Research terbaru oleh Black, Rogaway, dan Srimpton yang menganalisis lebih lanjut tentang sifat keamanan dari Miyaguchi-Preenel dan skema lainnya dari prespektif *black-box*, semakin menguatkan fungsi *Hash* Whirlpool.

Serangan-serangan untuk fungsi ini pasti tidak langsung diterapkan untuk fungsi ini. Serangan kunci pemulihan terbaik terhadap W dikurangi sampai 7 putaran adalah perluasan serangan oleh Gilbert dan Minier. Untuk menyerang fungsi ini dibutuhkan 2^{64} tebakan untuk sebuah kolom putaran pertama kunci x 2^{32} c-sets x 16 set nilai yang akan dienkripsi per entri tabel x 2 tabel x 2^{144} entri/tabel. Jumlah ini sampai dengan 2^{245} langkah. Hal ini dimungkinkan untuk me-mount serangan terhadap 7 putaran W, tetapu kompleksitasnya sangatlah tinggi (2^{512} lookup S-*box*, 2^{128} bit penyimpanan, O (2^{512}) plainteks). Hal ini pada dasarnya adalah kompleksitas menemukan *preimage* atau *preimage* kedua dengan *bruteforce*. Tidak ada serangan terhadap putaran W sampai dengan saat ini lebih lebih cepat dari pada *exhaustive search*.

Sejak *Whirlpool* tidak menggunakan bentuk deskripsi dari *cipher* internal W, enkripsi-dekripsi *cascade* secara eksplisit akan memberi tahu keberadaan kunci yang sedikit lemah, seperti yang enkripsi dengan satu tombol sesuai dengan dekripsi dengan kunci lain. Jadi kunci yang lemah tidak ada untuk fungsi *Hash Whirlpool* ini.

Message Digest yang dihasilkan dari program yang dibuat adalah 512 bit dengan kata lain sama dengan 128 karakter. Sebagai kode unik untuk sebuah kartu tanda penduduk menurut saya terlalu panjang untuk dibuat menjadi sebuah identitas. Oleh karena itu perlu adanya suatu proses tambahan untuk memperpendek *message digest* yang dihasilkan.

Untuk mencoba apakan fungsi *Hash Whirlpool* benar-benar dapat digunakan untuk mengenerate sidik jari setiap orang dan menghasilkan hasil yang berbeda membutuhkan banyak sekali percobaan.

Supaya percobaan lebih valid sebaiknya masukkan sidik jari harus mempunyai posisi/pola yang konsisten sehingga perbedaan sidik jari seseorang dapat terlihat dengan jelas.

5. KESIMPULAN

Fungsi *Hash Whirlpool* merupakan fungsi *Hash* yang paling dapat diukur(*scalable*) dari pada fungsi hash modern yang lain. Meskipun fungsi hash ini tidak secara khusus berorientasi pada platform manapun fungsi *Hash Whirlpool* masih lebih efisien dari pada fungsi *hash* yang lain, dengan struktur paralel luas mendukung pelaksanaan komponen pemetaan. Fungsi H*a*sh *Whirlpool* juga tidak memerlukan *space* penyimpanan yang berlebihan sehingga fungsi ini dapat dikatakan efisien. Hal ini menyebabkan fungsi ini dapat diimplementasikan dalam lingkungan yang sempit seperti *smartcard*.

Dengan adanya fungsi *Hash* Whirpool ini, dapat membantu pembuatan kode yang unik pada setiap Kartu Tanda Penduduk sehingga tidak terjadi penduplikasian.

Untuk menguji apakan fungsi *Hash Whirlpool* dapat menjadi fungsi yang valid untuk digunakan secara sah sebagai generator sidik jari yang akan digunakan, membutuhkan ujicoba yang sangat banyak.

Fungsi *Hash Whirlpool* seharushnya menghasilkan *message digest* yang unik sehingga dapat digunakan menghasilkan karakter unik dari setiap data masukkan.

REFERENSI

[1] http://www.kependudukancapil.go.id. Diakses pada 13 Mei 2010 pukul 18.00

[2] http://www.larc.usp.br/~pbarreto/WhirlpoolPage.html. Diakses pada 13 Mei 2010 oukul 18.45

[2] http://sovianchoeruman.wordpress.com/2010/02/01/bahan-e-ktp-untuk-indonesia/. Diakses pada 13 Mei 2010 pukul 18.30

[3] http://anangss.blogspot.com/2010/01/sekilas-tentang-e-ktp.html. Diakses pada 13 Mei 2010 pukul 20.00

[4] http://amanjdac.wordpress.com/2007/12/19/fungsi-hash/. Diakses pada 13 Mei 2010 pukul 20.37

[5] http://saluc.engr.uconn.edu/refs/algorithms/hashalg/bareto00whirlpool.pdf. Diakses pada 16 Mei 2010 pada pukul 16.00

[5] Munir, Rinaldi. (2006). Kriptografi.Program Studi Teknik Informatika,Institut Teknologi Bandung. http://saluc.engr.uconn.edu/refs/algorithms/hashalg/barreto00whirlpool.pdf

6. ACKNOWLEDGMENT

Dengan berakhirnya makalah ini, saya Adiputra Sejati ingin mengucapkan terimakasih sebesar-besarnya kepada bapak Rinaldi Munir yang selama ini telah mengajarkan kepada saya banyak hal tentang kriptografi. Saya juga mengucapkan banyak terima kasih kepada teman-teman saya yang telah memberikan banyak ide untuk kemajuan makalah ini.

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

Adiputra Sejati | 13507105