FUNGSI HASH BERBASIS TEORI GRAF: SEBUAH SURVEI

Susila Windarta
[1]Jurusan Teknik Persandian, Sekolah Tinggi Sandi Negara
Jalan Raya Haji Usa Desa Putat Nutug Ciseeng Bogor, 16330
HP: +62 8179854715
E-mail : windarta@yahoo.com;susila.windarta@stsn-nci.ac.id[1)]

Abstrak

Layanan keutuhan data (*data integrity*) *dapat diperoleh dengan menggunakan fungsi hash kriptografis* (*cryptographic hash function*). *Sampai saat ini telah banyak konstruksi fungsi hash yang diusulkan oleh para kriptografer. Konstruksi pertama diusulkan oleh Ralph Merkle dan Ivan Damgård secara terpisah. Konstruksi ini disebut Konstruksi Merkle-Damgård. Selain konstruksi tersebut terdapat pula konstruksi fungsi hash yang berbasis pada teori graf. Konstruksi pertama diusulkan oleh Gilles Zemor pada tahun* 1991. *Konstruksi selanjutnya oleh Jean-Pierre Tillich pada tahun* 1994. *Pada makalah ini dibahas beberapa konstruksi fungsi hash yang berbasis teori graf, baik graf berarah maupun tidak berarah. Selain itu juga dibahas mengenai serangan-serangan terhadap konstruksi tersebut.*

Kata kunci: *fungsi hash kriptografis, teori graf, kriptografi, konstruksi Zemor, konstruksi Tilich-Zemor, fungsi hash LPS*

1. PENDAHULUAN

Perkembangan ilmu pengetahuan dan teknologi khususnya teknologi informasi dan komunikasi membawa perubahan besar pada gaya hidup manusia. Akan tetapi kemudahan penggunaan teknologi tersebut kurang diiringi dengan kesadaran pengamanan yang memadai. Hal ini mengakibatkan ancaman terhadap keamanan data

dan informasi sangat besar. Salah satu alat yang dapat digunakan untuk mengamankan data dan informasi adalah kriptografi. Menezes et al. [4] menyatakan kriptografi adalah studi tentang teknik-teknik matematika yang berhubungan dengan aspek-aspek pengamanan informasi seperti kerahasiaan (*confidentiality*), keutuhan data (*data integrity*), otentikasi entitas (*entity authentication*), dan otentikasi asal data (*data origin authentication*). Layanan keutuhan data salah satunya dapat diperoleh dengan menggunakan fungsi hash kriptografis (*cryptographic hash function*). Mekanisme yang lain adalah dengan menggunakan skema tanda tangan digital (*digital signature scheme*). Layanan keutuhan data dapat mendeteksi manipulasi yang dilakukan terhadap suatu data oleh pihak yang tidak berhak. Manipulasi data mencakup penyisipan (*insertion*), penghapusan (*deletion*) dan substitusi (*substitution*).

Beberapa konstruksi fungsi hash telah diusulkan oleh para kriptografer. Konstruksi pertama diusulkan oleh Ralph Merkle [5] dan Ivan Damgård [2] secara terpisah. Konstruksi ini banyak digunakan sebagai dasar pembuatan fungsi hash yang saat ini umum digunakan, seperti keluarga Merkle-Damgård (MD), dan keluarga Secure Hash Algorithm (SHA). Keluarga SHA dibuat oleh National Institute of Standard and Technology (NIST). Keluarga SHA terdiri dari SHA0, SHA1 dan SHA2. Wang et al. [11] berhasil menemukan dua input atau lebih yang mempunyai nilai hash yang sama pada keluarga SHA, yaitu SHA1. Serangan ini lebih cepat dibandingkan *exhaustive search*/ *brute force attack*. Hal ini membuat aplikasi-aplikasi yang menggunakan keluarga SHA sangat rentan terhadap serangan. Oleh karena itu pada bulan November 2007, NIST mengumumkan kompetisi SHA-3 untuk menggantikan keluarga SHA sebelumnya [6]. Pada kompetisi tersebut banyak diusulkan berbagai konstruksi baru. Setelah melalui proses yang panjang didapatkan algoritma pemenang kompetisi SHA3. Algoritma tersebut adalah Keccak yang didesain oleh Guido Bertoni, Joan Daemen, Michaël Peeters, dan Gilles Van Assche. Algoritma ini menggunakan konstruksi *sponge*.

Makalah ini dibagi menjadi 4 bagian. Bagian pertama membahas latar belakang penulisan makalah serta penelitian yang mendukung. Bagian kedua mengenai teori dasar yang digunakan. Penjelasan pada bagian ini dimaksudkan agar pembaca yang tidak familiar dengan istilah yang digunakan dapat memahami. Bagian ketiga membahas mengenai konstruksi umum fungsi *hash* kriptografis berdasar teori graf, konstruksi Zemor, konstruksi Tillich-Zemor, dan konstruksi Charles et al. Pada pembahasan ketiga konstruksi tersebut diberikan pula serangan-serangan yang dapat dilakukan. Pada bagian keempat diberikan simpulan dan saran dari makalah ini.

2. DASAR TEORI

Pada bagian ini dibahas teori-teori yang mendasari fungsi *hash* kriptografis berdasar teori graf.

2.1 Grup matriks atas Lapangan Hingga [3]

Misal K merupakan lapangan. $GL_2(K)$ merupakan grup matriks 2×2 yang dapat dibalik dengan entri-entri elemen K terhadap operasi perkalian matriks. $SL_2(K)$ yang merupakan subgrup $GL_2(K)$, yang terdiri dari matriks-matriks dengan determinan 1 dan merupakan kernel dari pemetaan determinan, yang didefinisikan sebagai

$$\det : GL_2(K) \mapsto K^*. \tag{1}$$

$PGL_2(K)$ merupakan grup kuosien dari $GL_2(K)$ yang didefinisikan sebagai

$$PGL_2(K) = GL_2(K) \Big/ \left\{ \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix} : a \in K^* \right\}. \tag{2}$$

Jadi $PGL_2(K)$ adalah grup matriks di $GL_2(K)$ yang terdiri dari matriks dan semua kelipatan skalarnya berbeda dalam satu kelas ekivalen.

Sedangkan $PSL_2(K)$ merupakan grup kuosien dari $SL_2(K)$ yang didefinisikan sebagai

$$PSL_2(K) = SL_2(K) \Big/ \left\{ \begin{pmatrix} e & 0 \\ 0 & e \end{pmatrix} : e \in \{-1,1\} \right\}. \tag{3}$$

Jadi $PSL_2(K)$ adalah grup matriks di $SL_2(K)$ yang terdiri dari matriks dan negatif matriks tersebut ada dalam satu kelas ekivalen.

Jika lapangan $K = \mathbb{F}_q$, lapangan hingga dengan order *q*, keempat grup di atas dinotasikan dengan $GL_2(\mathbb{F}_q)$, $SL_2(\mathbb{F}_q)$, $PGL_2(\mathbb{F}_q)$ dan $PSL_2(\mathbb{F}_q)$.

2.2 Fungsi Hash Kriptografis

Berdasarkan [4], fungsi *hash* kriptografis didefinisikan pada Definisi 1.

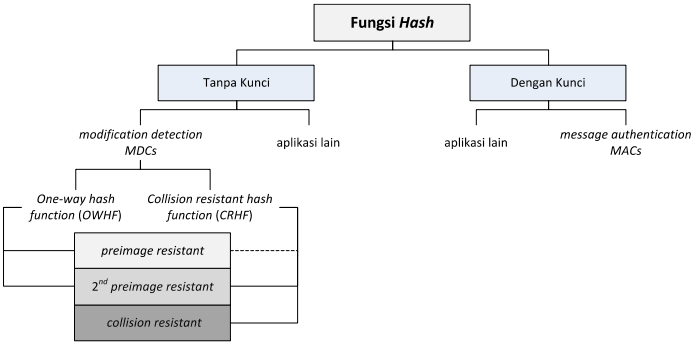
Definisi 1.*Fungsi* hash *merupakan fungsi* H *yang minimal mempunyai sifat-sifat sebagai berikut:*

1) *Kompresi*: *H memetakan sebarang input x dengan panjang bit berhingga ke output tetap* $H(x)$ *dengan panjang n,* $H : \{0,1\} \mapsto \{0,1\}^n$.
2) *Mudah dihitung*: *diberikan H dan input x, nilai* $H(x)$ *mudah dihitung.*

Secara umum klasifikasi fungsi *hash* diilustrasikan pada Gambar 1.

Selain sifat-sifat pada Definisi 1, untuk suatu fungsi *hash H* tanpa kunci dengan input x, x', dan output *y*, y' ditambah dengan sifat-sifat sebagai berikut:

1) *Preimage resistance* (one-way) — diberikan output *y*, secara perhitungan sulit menentukan input *x* sehingga $H(x) = y$.
2) *Second-preimage resistance* (*weak collision resistance*) — diberikan input *x*, secara perhitungan sulit menentukan input lain $x \neq x'$, sehingga $H(x) = H(x')$.
3) *Collision resistance* (*strong collision resistance*) — sulit secara perhitungan untuk mencari sebarang dua input $x \neq x'$, sehingga $H(x) = H(x')$.



Gambar 1. Klasifikasi fungsi hash [4]

2.3 Graf Cayley

Pada bagian ini dibahas mengenai Graf Cayley. Penjelasan pada bagian ini merujuk pada [3].

Definisi 2. Misalkan *G* adalah grup (berhingga atau tak berhingga) dan $\mathcal{S}$ adalah himpunan tidak kosong, himpunan bagian berhingga dari *G*. Graf Cayley $\mathcal{G}(G,\mathcal{S})$ adalah graf dengan himpunan simpul dan himpunan simpul $E=\{(x,y):x,y\in G,\exists S\in\mathcal{S}:y=xS\}$.

Jika $\mathcal{S}=\mathcal{S}^{-1}$ atau $\mathcal{S}$ simetrik maka Graf Cayley yang dibentuk adalah graf tidak berarah. Sedangkan jika $\mathcal{S}$ tidak simetrik maka Graf Cayley yang dibentuk adalah graf berarah.

3. KONSTRUKSI FUNGSI HASH BERDASAR TEORI GRAF

Sebelum membahas mengenai konstruksi fungsi *hash* berdasar teori graf, terlebih dahulu dibahas konstruksi umumnya.

3.1 Konstruksi Umum

Konstruksi umum dari fungsi *hash* berbasis Graf Cayley adalah dengan menggunakan sifat dari grup *G* yang membangkitkan graf tersebut. Ide konstruksi tersebut adalah mengganti karakter pada pesan yang akan di-*hash* dengan anggota grup *G* dan mengalikan anggota-anggota tersebut untuk mendapatkan nilai *hash* dari pesan. Proses ini sama artinya dengan melakukan perjalanan (*walk*) melalui simpul-simpul pada Graf Cayley. Simpul terakhir yang dikunjungi adalah nilai *hash* pesan.

3.2 Zemor (1991)

Konstruksi ini diusulkan oleh Gilles Zemor [12] pada tahun 1991.

3.2.1 Penghitungan Nilai *Hash*

Konstruksi ini berdasar pada Graf Cayley dengan grup yang digunakan adalah $SL_2(\mathbb{F}_p)$dengan *p* bilangan prima. Zemor mendefinisikan grup $SL_2(\mathbb{F}_p)$ merupakan grup yang dibangkitkan oleh pasangan:

$$A_1=\begin{bmatrix}1&1\\0&1\end{bmatrix},B_1=\begin{bmatrix}1&0\\1&1\end{bmatrix};A_2=\begin{bmatrix}1&2\\0&1\end{bmatrix},B_2=\begin{bmatrix}1&0\\2&1\end{bmatrix};A_3=\begin{bmatrix}1&1\\0&1\end{bmatrix},B_3=\begin{bmatrix}2&1\\1&1\end{bmatrix}. \quad (4)$$

Pasangan $A,$ dan B pada Persamaan (4) adalah pasangan pembangkit pada grup $SL_2(\mathbb{F}_p)$. Didefinisikan pemetaan $\theta:\{0,1\}\mapsto\{A,B\};0\mapsto A;1\mapsto B$. Misalkan $m=m_0m_1\ldots m_n$ adalah pesan dalam bit. Nilai *hash* dari pesan *m* dihitung menggunakan formula $H(m)=\theta(m_0)\cdot\theta(m_1)\cdots\theta(m_n)$. Pesan *m* dapat dianalogikan sebagai lintasan berarah pada Graf Cayley dengan elemen pada grup $SL_2(\mathbb{F}_p)$ dengan simpul awalnya adalah matriks identitas sedangkan simpul akhirnya adalah nilai *hash* pesan.

3.2.2 Serangan

Secara umum serangan dilakukan dengan memanfaatkan struktur matematika dari grup yang digunakan, yaitu $SL_2(\mathbb{F}_p)$. Jika penyerang dapat menemukan faktorisasi matriks identitas menjadi bentuk $b_1'\cdot b_2'\cdots b_t'=I_2,b_i'\in\{A,B\}$, maka penyerang dapat menyisipkan bit string tersebut ke sembarang pesan tanpa mengubah nilai *hash*-nya.Skema yang diusulkan oleh Zemor dapat diserang dengan beberapa macam cara. Pada makalah ini akan dijelaskan 2 cara.

Pertama dengan menemukan elemen grup yang mempunyai order yang kecil. Serangan dilakukan dengan menghitung nilai *hash* dari pesan acak, sampai didapatkan matriks yang dapat didiagonalkan. Misal *m* merupakan pesan yang mempunyai nilai *hash* berupa matriks *M* yang dapat didiagonalkan. Misal order dari *M* adalah *d*, maka dengan menggabungkan pesan *m* sebanyak *d* akan didapatkan nilai *hash* berupa matriks identitas. Penyerang dapat menyisipkan gabungan tersebut ke sembarang pesan tanpa mengubah nilai *hash*-nya.

Serangan kedua, berkaitan dengan nilai *hash* pesan yang berupa matriks simetrik. Pasangan elemen pembangkit pertama dan kedua pada Grup $SL_2(\mathbb{F}_p)$ yang didefinisikan Zemor saling tranpose, artinya $A=B^T$. Jika pesan *m* yang mempunyai nilai *hash* matriks simetrik, maka $\bar{m}$ akan mempunyai nilai *hash* matriks simetrik yang sama. $\bar{m}$ adalah komplemen dari *m*.

Dengan kedua cara di atas didapatkan tumbukan (*collision*) yang tak terhingga banyaknya.

3.3 Tillich-Zemor (1994)

Konstruksi ini diusulkan oleh Jean-Pierre Tillich dan Gilles Zemor pada tahun 1994 [9]. Konstruksi ini diklaim merupakan perbaikan dari skema Zemor.

3.3.1 Penghitungan Nilai *Hash*

Pada konstruksi ini digunakan Grup $SL_2\left(\mathbb{F}_{2^n}\right)$. Tillich-Zemor mendefinisikan parameter yang digunakan berupa polinomial tak-tereduksi $P_n(X)$ dengan derajat *n*. *n* dalam rentang 130-170 bit. Misalkan *A* dan *B* didefinisikan sebagai matriks $A=\begin{bmatrix} X & 1 \\ 1 & 0 \end{bmatrix}, B=\begin{bmatrix} X & X+1 \\ 1 & 1 \end{bmatrix}$.

Selanjutnya didefinisikan fungsi $\theta:\{0,1\}\mapsto\{A,B\};0\mapsto A;1\mapsto B$. Misalkan $m=m_0m_1\ldots m_n$ adalah pesan dalam bit. Nilai *hash* dari pesan *m* dihitung menggunakan formula $H(m)=\theta(m_0)\cdot\theta(m_1)\cdots\theta(m_n)$. Nilai *hash* dari pesan *m* berupa matriks yang merupakan elemen Grup $SL_2\left(\mathbb{F}_{2^n}\right)$.

3.3.2 Serangan

Penyerang dapat menemukan tumbukan pada konstruksi ini dengan menggunakan cara yang hampir sama dengan serangan pada konstruksi Zemor. Steinwandt et al. [8] menggunakan cara pertama yaitu menemukan elemen grup $SL_2\left(\mathbb{F}_{2^n}\right)$ yang mempunyai order kecil. Setelah elemen dengan order terkecil ditemukan, maka faktorisasi elemen identitas pada grup $SL_2\left(\mathbb{F}_{2^n}\right)$ dilakukan dengan mengalikan elemen tersebut sebanyak ordernya. Hasilnya dapat disisipkan pada sembarang pesan tanpa mengubah nilai *hash*-nya.

3.4 Charles et al. (2007)

Pada tahun 2007, Charles et al.[1] mengusulkan fungsi *hash* yang diklaim terbukti tahan terhadap tumbukan berdasar graf ekspander. Pada makalah tersebut dibahas dua keluarga fungsi *hash*, yaitu fungsi *hash* dari keluarga graf ekspander Pizer dan fungsi *hash* dari keluarga graf ekspander Lubotzky-Phillips-Sarnak (LPS). Pada makalah ini hanya keluarga fungsi *hash* LPS yang dibahas.

Definisi 3. Graf ekspander LPS merupakan Graf Cayley, yang dinotasikan dengan $X^{\ell,p}$, didefinisikan sebagai berikut:

$$X^{\ell,p}=\begin{cases} \mathcal{G}(PSL_2(\mathbb{F}_p),\mathcal{S}) & \text{jika } \ell \text{ kuadrat modulo } p. \\ \mathcal{G}(PGL_2(\mathbb{F}_p),\mathcal{S}) & \text{jika } \ell \text{ bukan kuadrat modulo } p. \end{cases} \quad (5)$$

3.4.1 Penghitungan Nilai *Hash*

Penghitungan nilai *hash* dapat diuraikan sebagai berikut:

INPUT : Pesan *m* dengan panjang sembarang

OUTPUT : Nilai hash yang berupa simpul di graf $X^{\ell,p}$

1. Parameter awal : $X^{\ell,p}$ adalah graf ekspander LPS pada Definisi 3. Misal $k=\ell+1$ dan $k'=k-1$. Busur awal adalah busur $\left(g_0S_{-1}^{-1},g_0\right)$ dengan $g_0\in PSL_2(\mathbb{F}_p)$ dan $S_{-1}\in\mathcal{S}$. Fungsi urutan ketetanggaan didefinisikan sebagai $\theta:\mathcal{S}\times\{0,\ldots,k'-1\}\mapsto\mathcal{S}$ yang memetakan sebuah elemen di $\mathcal{S}$ dan bilangan digit-*k* ke elemen $\mathcal{S}$, sehingga untuk setiap $S\in\mathcal{S}$,
$$\{\theta(S,i)\mid 0\le i\le k'-1\}\cup\{S^{-1}\}=\mathcal{S}.$$
2. Konversi pesan : Ubah pesan *m* menjadi bilangan k'-digit, yaitu $m=m_0\ldots m_{\mu-1}$ dengan $m_i\in\{0,\ldots,k'-1\}$.
3. Penghitungan nilai hash : Untuk $i=0$ sampai $\mu-1$,

   Hitung $S_i=\theta(S_{i-1},m_i)$ dan $v_i=v_{i-1}S_i$, kemudian hitung $H(m)=g_0\prod_{i=0}^{\mu-1}S_i=g_0S_0S_1\ldots S_{\mu-1}$;
4. Penyelesaian. Nilai hash dari pesan *m* adalah *H*(*m*).

3.4.2 Serangan

Keamanan fungsi *hash* LPS dijelaskan pada Teorema 1 dan Teorema 2.

Teorema 1. *Masalah Tumbukan:* [1] *Misalkan p dan* ℓ *dua bilangan prima berbeda yang kongruen* 1 *modulo* 4 *dan* ℓ *adalah bilangan kuadrat modulo p. Misal* $\mathcal{S} = S_1, S_2, \ldots, S_{\ell+1}$ *adalah pembangkit dari* $PSL_2(\mathbb{F}_p)$. *Tentukan perkalian*

$$\prod_{i=0}^{\mu-1} S_i^{e_i} = I_2 \tag{6}$$

sehingga $\sum_i e_i$ *adalah* $O(\log p)$ *dan untuk setiap* $i, S_i \neq S_{i+1}^{-1}$.

Teorema 2. *Mencari tumbukan pada fungsi hash LPS dengan ukuran input* $O(\log p)$ *ekuivalen dengan menyelesaikan Masalah Tumbukan.*

Serangan terhadap LPS *hash* pertama kali dilakukan oleh Tillich dan Zemor [10] dan dipertegas oleh Petit et al. [7]. Serangan tersebut dijelaskan sebagai berikut:

1. Ubah matriks identitas di $PSL_2\left(\mathbb{F}_p\right)$ menjadi matriks *M* elemen himpunan matriks Ω subhimpunan $SL_2\left(\mathbb{Z}[i]\right)$.
2. Faktorkan matriks tersebut menjadi perkalian dari pembangkit $\tilde{\mathcal{S}}$.
3. Petakan hasil pemfaktoran kembali ke $PSL_2\left(\mathbb{F}_p\right)$ akan menyelesaikan Masalah Tumbukan.

Tillich dan Zemor [10] mengusulkan untuk mengganti $S_i \in \mathcal{S}, i = 1, \ldots, \ell+1$ menjadi S_i^2, sehinga skema serangan yang telah dijelaskan tidak dapat dilakukan.

4. SIMPULAN DAN SARAN

Pada bagian terakhir ini diberika simpulan dan saran dari makalah.

4.1 Simpulan

1. Fungsi *hash* berdasar teori graf yang dibahas pada makalah ini tidak memenuhi sifat *collision resistance* karena tumbukan dapat ditemukan secara efisien.
2. Serangan yang dilakukan memiliki persamaan yaitu menemukan elemen grup dengan order kecil, yang selanjutnya digunakan untuk melakukan faktorisasi terhadap identitas grup.

4.2 Saran

Untuk penelitian selanjutnya perlu dijelaskan sifat-sifat fungsi *hash* lain, yaitu *second-preimage resistance dan preimage resistance*.

5. DAFTAR RUJUKAN

[1] Charles, D., Goren, E., dan Lauter, K., 2007. Cryptographic hash functions from expander graph. *Journal of Cryptology*, 22, pp. 93–113.

[2] Damgård, I., 1989. *A design principle for hash functions*. Brassard, G., editor, Advances in Cryptology - CRYPTO 1989, Lecture Notes in Computer Science, Volume 435, pp. 416–427. Springer-Verlag.

[3] Davidoff, G., Sarnak, P., and Valette, A. (2003). *Elementary Number Theory, Group Theory, and Ramanujan Graphs.* London Mathematical Society Student Texts, No. 55. Cambridge University Press.

[4] Menezes, A., Ooschot, P. V., and Vanstone, S., 1996. *Handbook of Applied Cryptography.* CRC Press, Boca Raton.

[5] Merkle, R.C., 1989. One way hash functions and DES. In: *Proceedings on Advances in cryptology* (CRYPTO '89), Gilles Brassard (Ed.). Springer-Verlag New York, Inc., New York, NY, USA, 428-446.

[6] NIST. 2007. Notices. Federal Register/Vol.72 No. 212.

[7] Petit, C., Lauter, K., and Quisquater, J.-J., 2008. Full cryptanalysis of LPS and Morgenstern hash functions. In: Ostrovsky, R., Prisco, R. D., and Visconti, I., editor, *Security and Cryptography for Networks* 2008, Lecture Notes in Computer Science, Volume 5229, pp. 263–277. Springer.

[8] Steinwandt, R, Grassl, M, Geiselmann, W, dan Beth, T., 2000. Weaknesses in the $SL_2\left(\mathbb{F}_{2^n}\right)$ Hashing Scheme. In: *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology* (CRYPTO '00), Mihir Bellare (Ed.). Springer-Verlag, London, UK, UK, 287-299.

[9] Tillich, J.P and Zemor, G., 1994, Hashing with SL2. In: Y. Desmedt, editor, *Advances in Cryptology – CRYPTO* '94, volume 839 of Lecture Notes in Computer Science, pp. 40–49.

[10] Tillich, J.P dan Zemor, G. 2008. Collisions for the LPS expander graph hash function. Smart, N., editor, *Advances in Cryptology (EUROCRYPT'08),* The Theory and Applications of Cryptographic Techniques 27th Annual International Conference, pp. 254–269, Berlin/Heidelberg. Springer-Verlag.

[11] Wang, X., Yin, Y. L., and Yu, H., 2005. Finding collisions in the full SHA-1. *Advances in Cryptology-CRYPTO* 2005, Lecture Notes in Computer Science, Berlin/Heidelberg. Springer.

[12] Zemor, G., 1994. Hash functions and cayley graphs. *Designs, Codes and Cryptography*, 4(3), pp. 381–94.