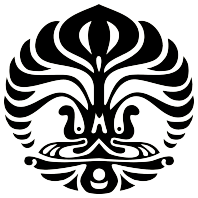
UNIVERSITAS INDONESIA

FUNGSI *HASH* KRIPTOGRAFIS DARI GRAF EKSPANDER LUBOTZKY PHILLIPS SARNAK

TESIS

SUSILA WINDARTA
0906495425

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI MAGISTER MATEMATIKA
DEPOK
JUNI 2011



UNIVERSITAS INDONESIA

FUNGSI *HASH* KRIPTOGRAFIS DARI GRAF EKSPANDER LUBOTZKY PHILLIPS SARNAK

TESIS

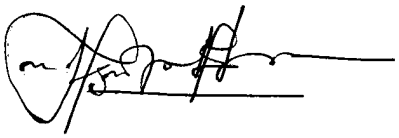
Diajukan sebagai salah satu syarat untuk memperoleh gelar
Magister Sains

SUSILA WINDARTA
0906495425

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI MAGISTER MATEMATIKA
DEPOK
JUNI 2011

HALAMAN PERNYATAAN ORISINALITAS

Tesis ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama	**:**	**Susila Windarta**
NPM	**:**	**0906495425**
Tanda Tangan	**:**	
Tanggal	**:**	**8 Juni 2011**

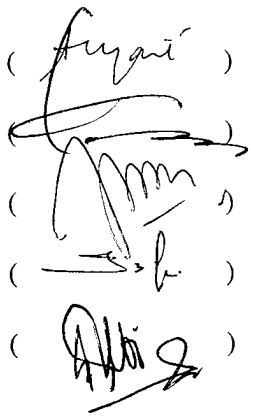
HALAMAN PENGESAHAN

Tesis ini diajukan oleh	:	
Nama	:	Susila Windarta
NPM	:	0906495425
Program Studi	:	Magister Matematika
Judul Tesis	:	Fungsi *Hash* Kriptografis dari Graf Ekspander Lubotzky Phillips Sarnak

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister pada Program Studi Magister Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing	:	Dr. Kiki Ariyanti Sugeng	( )
Penguji	:	Prof. Dr. Djati Kerami	( )
Penguji	:	Dr. Yudi Satria, M.T.	( )
Penguji	:	Dr. rer. nat. Hendri Murfi, M.Kom.	( )
Penguji	:	Dr. Alhadi Bustamam, M.Kom.	( )



Ditetapkan di	:	Depok
Tanggal	:	8 Juni 2011

KATA PENGANTAR

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

"Hai orang-orang yang beriman, apabila dikatakan kepadamu: "berlapang-lapanglah dalam majlis, maka lapangkanlah, niscaya Allah akan memberi kelapangan untukmu. Dan apabila dikatakan: "berdirilah kamu, maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan"

(QS. Al-Mujadalah: 11)

Segala puji hanyalah milik Allah SWT yang telah memberikan segala nikmat bagi seluruh umat manusia. Shalawat teriring salam semoga selalu tercurahkan bagi teladan umat manusia, Nabi Muhammad SAW, kepada keluarga, kepada sahabat, dan kepada para pengikut beliau yang istiqomah hingga hari pembalasan nanti. Rasa syukur tiada terkira penulis sampaikan kehadirat Allah SWT sehingga penulisan tesis yang berjudul Fungsi *Hash* Kriptografis dari Graf Ekspander Lubotzky Phillips Sarnak dapat penulis selesaikan tepat waktu.

Banyak pihak yang telah membantu dalam penulisan tesis ini, oleh sebab itu tidak berlebihan kiranya penulis mengucapkan terima kasih kepada pihak-pihak terkait:

1. Lembaga Sandi Negara yang telah memberikan kesempatan kepada penulis untuk mendapatkan program beasiswa selama dua tahun ini;
2. Ibu Dr. Kiki Ariyanti Sugeng atas segala arahan, bimbingan dan perhatian yang selama ini diberikan.
3. Bapak Prof. Dr. Djati Kerami selaku Ketua Program Studi Magister Matematika sekaligus penasehat akademik penulis.
4. Para dosen Program Studi Magister Matematika Universitas Indonesia.
5. Istri tercinta, Mujinah, S.Sos. dan putri tercinta, Azka Syifa Warastri Windarta atas segala doa, perhatian, kesabaran dan dukungan selama menempuh pendidikan ini.
6. Ibu Ngatinem dan bapak Sujimin yang telah memberikan semuanya untuk penulis.

7. Mas Sigit Nurianta dan Dek Bekti Sih Winarni yang selalu mendukung di segala kesempatan.

8. Rekan-rekan mahasiswa S2 angkatan 2009 atas kebersamaan selama ini, terutama rekan-rekan dari Jambi, Mas Mulyadi, Pahrin Wirnadian dan Henang Priyanto. Semoga kebaikan rekan-rekan dibalas dengan balasan yang lebih baik.

Akhir kata, penulis menyadari tesis ini masih jauh dari sempurna. Oleh karena itu segala kritik, masukan dan saran sangat penulis harapkan. Penulis berharap tesis ini dapat bermanfaat bagi yang membaca.

Depok, 8 Juni 2011

Susila Windarta

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama	:	Susila Windarta
NPM	:	0906495425
Program Studi	:	Magister Matematika
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam
Jenis Karya	:	Tesis

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

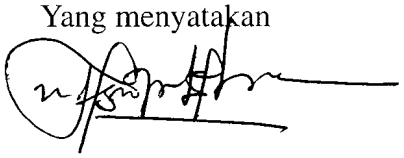
Fungsi *Hash* Kriptografis dari Graf Ekspander Lubotzky Phillips Sarnak

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 8 Juni 2011

Yang menyatakan



(Susila Windarta)

ABSTRAK

Nama	:	Susila Windarta
Program Studi	:	Magister Matematika
Judul	:	Fungsi *Hash* Kriptografis dari Graf Ekspander Lubotzky Phillips Sarnak

Salah satu aspek pengamanan informasi yang diberikan oleh kriptografi adalah layanan keutuhan data (*data integrity*). Layanan keutuhan data salah satunya dipenuhi oleh fungsi *hash* kriptografis. Saat ini banyak fungsi *hash* yang dikonstruksi berdasarkan konstruksi Merkle-Damgård, di antaranya keluarga *Secure Hash Algorithm* (SHA). Serangan yang berhasil dilakukan oleh Wang dkk. (2005) terhadap sifat *collision resistant* pada SHA1 menuntut adanya konstruksi fungsi *hash* yang baru. Pada tahun 2007, Charles dkk. mengusulkan konstruksi fungsi *hash* yang berdasarkan graf ekspander. Salah satu graf ekspander yang diusulkan adalah graf ekspander Lubotzky Phillips Sarnak (LPS).

Konstruksi, aspek keamanan, serangan serta modifikasi terhadap fungsi *hash* tersebut dibahas pada tesis ini. Serangan tersebut berdasar pada Tillich dan Zemor (2008) yang berhasil menemukan tumbukan (*collision*) secara efisien. Modifikasi dilakukan dengan mengganti setiap elemen pada himpunan generator graf dengan kuadratnya. Modifikasi tersebut mengakibatkan serangan untuk menentukan tumbukan yang berdasar Tillich dan Zemor (2008) lebih sulit untuk dilakukan.

Kata Kunci:
keutuhan data, fungsi *hash*, graf ekspander, tumbukan, *collision*, graf ekspander LPS

ABSTRACT

Name	:	Susila Windarta
Study Program	:	Magister of Mathematics
Title	:	Cryptographic Hash Function from Lubotzky Phillips Sarnak Expander Graph

One aspect of information security that given by cryptography is data integrity. Cryptographic hash function can be used to obtain data integrity. Today many hash functions are constructed based on the Merkle-Damgård construction, including family of Secure Hash Algorithm (SHA). The consequence of successful attack carried out by Wang et al. (2005) on collision-resistant properties of SHA1 is the need of new construction for hash function. In 2007, Charles et al. proposed construction of hash functions based on expander graphs. In the proposal, one of expander graph used is Lubotzky Phillips Sarnak (LPS) expander graph.

The construction of hash function based on LPS expander graph, its security aspects, an attack and modification on the hash function are discussed in this thesis. The attack is based on Tillich and Zemor (2008) who managed to find a collision efficiently. Modification is done by replacing each element in graph generator set with the square of the element, hence collision attack based on Tillich and Zemor (2008) is more difficult to do.

Keywords:
data integrity, hash function, expander graph, collision, LPS expander graph

DAFTAR ISI

DAFTAR GAMBAR

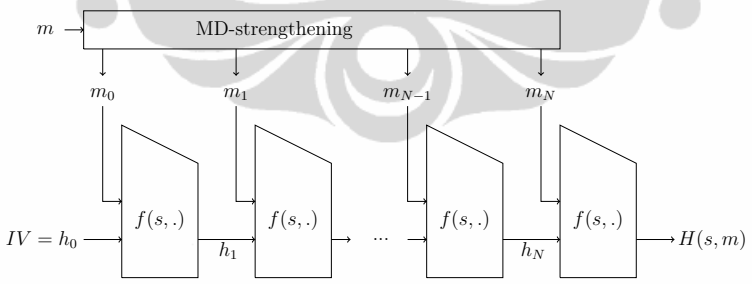
DAFTAR TABEL

DAFTAR ALGORITMA

BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

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 dkk. (1996) 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*). Skema tanda tangan digital tidak dibahas dalam tesis ini. Layanan keutuhan data dapat mendeteksi manipulasi yang dilakukan terhadap suatu data oleh pihak yang tidak berhak. Manipulasi data mencakup penyisipan (*insertion*), penghapusan (*deletion*) dan penggantian (*subtitution*).



Sumber: Petit (2009)

Gambar 1.1. Konstruksi Merkle-Damgård.

Sampai saat ini telah banyak konstruksi fungsi *hash* yang diusulkan oleh para kriptografer. Konstruksi pertama diusulkan oleh Ralph Merkle (Merkle, 1979) dan Ivan Damgård (Damgård, 1989) secara terpisah. Gambaran konstruksi Merkle-Damgård dapat dilihat pada Gambar 1.1. Fungsi f pada konstruksi ini merupakan suatu fungsi kompresi yang *collision resistant*. Nilai $h_0 = IV$ merupakan nilai awal atau vektor inisialisasi. Konstruksi ini banyak digunakan sebagai dasar pembuat-

an 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), sebuah badan riset di bawah Departemen Perdagangan, Amerika Serikat. Keluarga SHA terdiri dari SHA0, SHA1 dan SHA2. Wang dkk. (2005) 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 (NIST, 2007). Pada kompetisi tersebut banyak diusulkan berbagai konstruksi baru. Antara lain konstruksi yang berdasarkan *block cipher* oleh Ferguson dkk. (2010). Saat ini kompetisi memilih SHA3 telah sampai pada putaran final. Terdapat lima kandidat yang lolos putaran final SHA3.

Selain konstruksi pada kompetisi tersebut, terdapat konstruksi fungsi *hash* lain berdasar pada permasalahan pada graf ekspander. Konstruksi ini diusulkan oleh Charles dkk. (2007). Graf ekspander yang digunakan adalah keluarga graf Ramanujan, yaitu graf Lubotzky, Phillips dan Sarnak (LPS) dan graf Pizer. Khusus untuk fungsi *hash* berdasar graf LPS terdapat beberapa serangan, yaitu dilakukan oleh Tillich dan Zemor (2008) serta Petit dkk. (2008).

Pada tesis ini dilakukan kajian terhadap fungsi *hash* dari graf ekspander LPS. Serangan terhadap fungsi *hash* tersebut juga dibahas, yaitu serangan untuk menentukan tumbukan (*collision*). Selanjutnya dilakukan modifikasi terhadap fungsi *hash* dari graf ekspander LPS. Modifikasi ini berdasarkan ide dari Tillich dan Zemor (2008). Modifikasi yang dilakukan dapat mempersulit serangan untuk menentukan tumbukan (*collision*).

1.2 Permasalahan

Permasalahan yang dibahas pada tesis ini adalah bagaimana graf ekspander LPS dapat digunakan sebagai fungsi *hash* kriptografis dan bagaimana mengatasi serangan tersebut.

1.3 Batasan Masalah

Pembatasan masalah pada tesis ini sebagai berikut:

1. Fungsi *hash* yang dibahas adalah kelas fungsi *hash* tanpa kunci (*unkeyed hash function*).

2. Graf yang digunakan adalah graf sederhana, tanpa *loop* (gelung), tanpa busur ganda.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Mengkaji fungsi hash yang berdasar graf ekspander LPS.
2. Mengkaji serangan terhadap fungsi *hash* dari graf ekspander LPS, yaitu mencari tumbukan (*collision*).
3. Memodifikasi fungsi *hash* dari graf ekspander LPS sehingga serangan untuk mencari tumbukan (*collision*) tidak dapat dilakukan.

1.5 Manfaat Penelitian

Penulisan ini diharapkan dapat memberikan gambaran tentang penggunaan teori graf yaitu graf ekspander dalam konstruksi fungsi *hash* serta bentuk serangan yang dapat dilakukan terhadap fungsi *hash* tersebut khususnya menemukan tumbukan (*collision*).

1.6 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah telaah kepustakaan dan kajian teoritis.

1.7 Sistematika Penulisan

Tesis ini terdiri dari lima bab, yaitu

BAB 1 berisi latar belakang, permasalahan yang dibahas, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penulisan.

BAB 2 menjelaskan mengenai landasan teori yang digunakan pada bab-bab selanjutnya.

BAB 3 berisi tentang konstruksi graf ekspander LPS, konstruksi fungsi *hash* dari graf ekspander secara umum, konstruksi fungsi *hash* dari graf ekspander Cayley, serta konstruksi fungsi *hash* dari graf ekspander LPS.

BAB 4 membahas mengenai serangan pada fungsi *hash* dari graf ekspander LPS, yaitu menemukan tumbukan (*collision*) dan modifikasi yang dapat dilakukan.

BAB 5 berisi kesimpulan dan saran yang berkaitan dengan penelitian.

BAB 2
LANDASAN TEORI

Bab ini terdiri dari tujuh subbab. Subbab pertama membahas tentang teori grup, gelanggang dan lapangan. Subbab kedua membahas mengenai grup matriks atas lapangan hingga dengan order bilangan prima atau pangkat dari bilangan prima. Pada subbab selanjutnya dibahas tentang fungsi *hash* kriptografis. Beberapa teori tantang graf yang dipergunakan dalam tesis ini dibahas pada subbab keempat. Subbab kelima membahas mengenai bilangan bulat Gauss. Sedangkan subbab seanjutnya membahas mengenai quaternion dan sifat-sifat yang relevan. Pada subbab terakhir dibahas mengenai *continued fraction*. Pembuktian teorema dan lema tidak diberikan, tetapi dapat dilihat pada referensi masing-masing.

2.1 Teori Grup, Gelanggang dan Lapangan

Subbab ini akan membahas mengenai teori grup, gelanggang, dan lapangan yang akan digunakan pada bab-bab selanjutnya. Materi pada subbab ini berdasarkan Fraleigh (2002).

Definisi 2.1. *Suatu grup $\langle G, * \rangle$ adalah himpunan G yang tertutup terhadap operasi biner $*$, sehingga memenuhi aksioma-aksioma berikut:*

(1) *Untuk semua $a, b, c \in G$, berlaku*

$$(a * b) * c = a * (b * c). \quad \textbf{\textit{sifat asosiatif}}\ *$$

(2) *Terdapat sebuah elemen $e \in G$ sehingga untuk semua $x \in G$*

$$e * x = x * e = x. \quad \textbf{\textit{elemen identitas}}\ \textit{euntuk}\ *$$

(3) *Untuk setiap $a \in G$ terdapat elemen $a^{-1} \in G$ sehingga*

$$a * a^{-1} = a^{-1} a = e. \quad \textbf{\textit{invers}}\ \textit{aadalah}\ a^{-1}$$

Definisi 2.2. *Grup G disebut abelian jika operasi biner $*$ bersifat komutatif.*

Definisi 2.3. *Suatu subhimpunan tak kosong, H, dari grup G disebut subgrup dari G jika H membentuk suatu grup terhadap operasi biner $*$.*

Lema 2.4. *Suatu subhimpunan tak kosong $H \subseteq G$ adalah subgrup dari G jika dan hanya jika:*

(1) *H tertutup terhadap operasi biner $*$ dari G,*

(2) *Elemen identitas e dari G ada di H, dan*

(3) *Untuk semua $a \in H$, $a^{-1} \in H$.*

Definisi 2.5. *Jika G grup, order $|G|$ dari G adalah banyak elemen di G. Sedangkan order $o(g)$ dari suatu elemen $g \in G$ adalah bilangan bulat terkecil k sedemikian sehingga $g^k = e$, dengan e elemen identitas di G.*

Definisi 2.6. *Misalkan H adalah subgrup dari grup G. Subhimpunan $aH = \{ah | h \in H\}$ dari G adalah koset kiri dari H, untuk setiap $a \in G$. Sedangkan $Ha = \{ha | h \in H\}$ adalah koset kanan dari H, untuk setiap $a \in G$.*

Definisi 2.7. *Pemetaan ϕ dari grup G ke grup G' disebut homomorfisma jika sifat berikut terpenuhi, yaitu*

$$\phi(ab) = \phi(a)\phi(b)$$

untuk setiap $a, b \in G$.

Definisi 2.8. *Misalkan $\phi : G \to G'$ adalah suatu homomorfisma grup. Suatu subgrup $\phi^{-1}[\{e'\}] = \{x \in G | \phi(x) = e'\}$ adalah kernel dari ϕ yang dinotasikan sebagai $Ker(\phi)$.*

Definisi 2.9. *Suatu subgrup H dari grup G disebut subgrup normal jika $gH = Hg, \forall g \in G$ atau koset kiri dari H sama dengan koset kanan dari H.*

Akibat 2.10. *Jika $\phi : G \to G'$ adalah suatu homomorfisma grup, maka $Ker(\phi)$ adalah subgrup normal.*

Teorema 2.11. *Misalkan H subgrup dari grup G. Maka perkalian koset kiri, yaitu*

$$(aH)(bH) = (ab)H, \forall a, b \in G$$

terdefinisi dengan baik jika dan hanya jika koset kiri sama dengan koset kanan, yakni $aH = Ha, \forall a \in G$.

Akibat 2.12. *Misalkan H subgrup dari G yang koset kiri sama dengan koset kanan. Maka koset dari H membentuk grup G/H terhadap operasi biner $(aH)(bH) = (ab)H, \forall a, b \in G$.*

Definisi 2.13. *Grup G/H yang didefinisikan pada Akibat 2.12 disebut grup faktor atau grup kuosien dari G modulo H.*

Definisi 2.14. *Gelanggang $\langle R, +, \times \rangle$ adalah himpunan R dengan dua operasi biner $+$ dan $\times$, disebut penjumlahan dan perkalian, yang didefinisikan di R dan memenuhi aksioma-aksioma berikut:*

(1) $\langle R, + \rangle$ *adalah grup abelian;*

(2) *Operasi perkalian bersifat asosiatif terhadap operasi penjumlahan; dan*

(3) *Untuk semua $a, b, c \in R$ hukum distributif kiri $a \times (b + c) = (a \times b) + (a \times c)$ berlaku, dan hukum distribusi kanan $(a + b) \times c = (a \times c) + (b \times c)$ berlaku.*

Teorema 2.15. *Suatu subhimpunan S dari gelanggang R adalah subgelanggang jika dan hanya jika pernyataan berikut terpenuhi:*

(1) $0_R \in S$;

(2) $(a - b) \in S, \forall a, b \in S$; *dan*

(3) $ab \in S, \forall a, b \in S$.

Definisi 2.16 (Homomorfisma)**.** *Misal R dan R' adalah gelanggang. Pemetaan $\phi : R \to R'$ adalah suatu homomorfisma ring jika sifat-sifat berikut terpenuhi untuk semua $a, b \in R$:*

(1) $\phi(a + b) = \phi(a) + \phi(b)$.

(2) $\phi(ab) = \phi(a)\phi(b)$.

Definisi 2.17 (Isomorfisma)**.** *Suatu isomorfisma $\phi : R \to R'$ adalah suatu homomorfisma yang bersifat satu-satu dan pada. Gelanggang R dan R' disebut isomorfis.*

Definisi 2.18. *Gelanggang yang operasi perkaliannya komutatif disebut gelanggang komutatif. Suatu gelanggang R yang mempunyai identitas perkalian 1 sehingga $1x = x1 = x$ untuk semua $x \in R$ disebut gelanggang dengan kesatuan* (unity)*. Identitas perkalian 1 dalam suatu gelanggang disebut kesatuan* (unity).

Definisi 2.19. *Misal R gelanggang dengan kesatuan (*unity*) $1 \neq 0$. Suatu elemen $u \in R$ adalah unsur satuan dari R jika u mempunyai invers perkalian di R. Jika setiap elemen tak nol dari R adalah unsur satuan maka R disebut gelanggang pembagian* (division ring)*. Suatu lapangan* (field) *adalah gelanggang pembagian komutatif* (commutative division ring).

2.2 Grup Matriks atas Lapangan Hingga

Pada subbab ini dibahas mengenai grup yang digunakan untuk membentuk graf ekspander LPS. Materi pada subbab ini berdasarkan Davidoff dkk. (2003).

Misal K merupakan lapangan. $GL_2(K)$ merupakan grup matriks 2×2 yang dapat dibalik dengan entri-entrinya adalah 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

$$\det : GL_2(K) \to K^*.$$

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

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

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

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

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

Jadi $PSL_2(K)$ adalah grup matriks di $SL_2(K)$ yang terdiri dari matriks dan negatifnya 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 $PSL_2(\mathbb{F}_q)$. Pada Lema 2.20 dirangkum banyaknya elemen pada setiap grup.

Lema 2.20.

(1) $\left|GL_2(\mathbb{F}_q)\right| = q\,(q-1)\,(q^2-1)$

(2) $\left|SL_2(\mathbb{F}_q)\right| = \left|PGL_2(\mathbb{F}_q)\right| = q\,(q^2-1)$

(3) $\left|PSL_2(\mathbb{F}_q)\right| = \begin{cases} q\,(q^2-1) & \textit{jika } q \textit{ genap} \\ q\,(q^2-1)\,/2 & \textit{jika } q \textit{ ganjil.} \end{cases}$

2.3 Fungsi *Hash* Kriptografis

Pembahasan pada subbab ini tentang konsep fungsi *hash* tanpa kunci yang meliputi definisi, klasifikasi dan sifat-sifat. Materi pada subbab ini berdasarkan Menezes dkk. (1996, Bab 9).

Definisi 2.21. *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.*

$$H : \{0,1\}^* \rightarrow \{0,1\}^n.$$

(2) ***Mudah dihitung:*** *diberikan H dan input x, nilai $H(x)$ mudah dihitung.*

Secara umum fungsi *hash* dapat diklasifikasikan menjadi dua, yaitu:

1. *Modification Detection Codes* (MDCs)

   Disebut juga *manipulation detection codes* atau *message integrity codes* (MICs). Fungsi *hash* kategori ini digunakan untuk mendapatkan *image* atau nilai hash pesan, sehingga integritas data dapat ditentukan. MDC merupakan subkelas fungsi *hash* tanpa kunci (*unkeyed hash function*). MDC dapat dibagi menjadi 2 (dua) subkelas yakni:

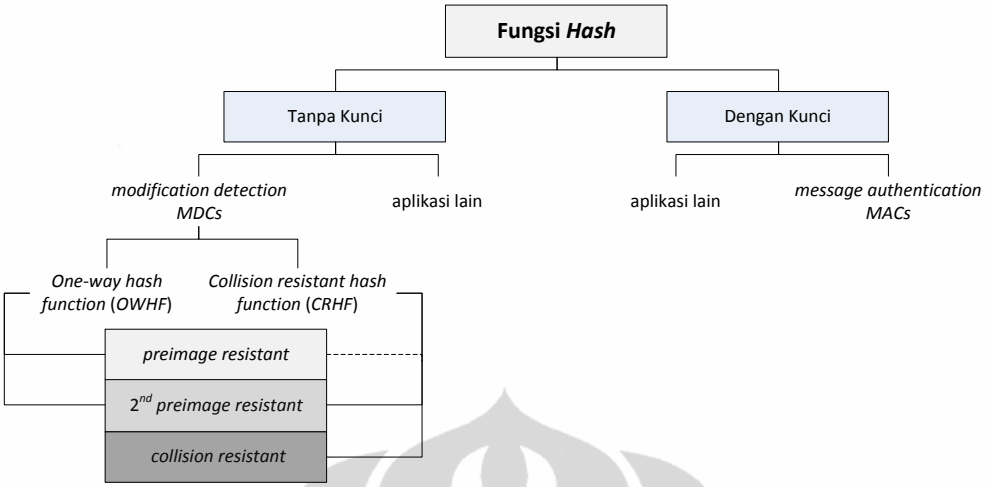
   (a) *one-way hash functions* (OWHFs): sulit menemukan input pesan dari nilai *hash* yang diberikan;

   (b) *collision resistant hash functions* (CRHFs): sulit menemukan sebarang dua input yang mempunyai nilai *hash* yang sama.

2. *Message Authentication Codes* (MACs)

   Merupakan subkelas fungsi *hash* dengan kunci (*keyed hash function*). Penggunaan MAC bertujuan untuk menjamin integritas sumber dan pesan, tanpa menggunakan mekanisme lain. Berbeda dengan MDC, fungsi *hash* jenis ini menggunakan parameter input dan kunci.

Gambar 2.1 menjelaskan secara sederhana klasifikasi fungsi *hash* di atas. Aplikasi lain fungsi *hash* antara lain digunakan dalam protokol kriptografi, seperti protokol *challenge response.*



Sumber: telah diolah kembali dari Menezes dkk. (1996)

Gambar 2.1. Klasifikasi fungsi *hash*

Selain sifat-sifat pada Definisi 2.21, 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 $H(x') = H(x)$.

Sifat *collision resistance* mengakibatkan sifat *second preimage resistance*. Pernyataan tersebut dapat dijelaskan sebagai berikut:

Misalkan H adalah fungsi *hash* yang bersifat *collision resistant*. Jika H tidak bersifat *second preimage resistant*, maka diberikan input x terdapat input lain x' sedemikian sehingga $H(x) = H(x')$. Hal ini berarti telah ditemukan suatu tumbukan (*collision*) antara x dan x'. Ini bertentangan dengan H yang bersifat *collision resistant*. Sehingga jika H bersifat *collision resistant* maka H juga bersifat *second preimage resistant*.

2.4 Teori Graf

Sebelum membahas materi yang berkaitan dengan graf ekspander terlebih dahulu dibahas tentang konsep-konsep dalam teori graf yang relevan. Materi pada subbab ini berdasarkan pada Davidoff dkk. (2003) dan Petit (2009) kecuali disebutkan lain.

Definisi 2.22. *Graf $\mathcal{G}$ adalah pasangan himpunan (V,E), dengan V adalah himpunan simpul dan $E \subset V \times V$ adalah himpunan busur. Sebarang busur $e = (v_1, v_2) \in E$ mempunyai simpul awal v_1 dan simpul akhir v_2.*

Definisi 2.23. *Dua simpul $v_1, v_2 \in V$ bertetangga jika setidaknya (v_1, v_2) atau (v_2, v_1) ada di E. Suatu gelung* (loop) *adalah busur (v_1, v_2) sedemikian sehingga $v_1 = v_2$.*

Definisi 2.24. *Misalkan graf $\mathcal{G} = (V,E)$ merupakan graf yang mempunyai n simpul. Suatu graf dikatakan sederhana jika graf tersebut tidak mempunyai gelung dan busur ganda. Graf $\mathcal{G} = (V,E)$ dikatakan tidak berarah jika untuk setiap busur $(v_1, v_2) \in E$ maka (v_2, v_1) juga ada di $(v_2, v_1) \in E$.*

Definisi 2.25. *Suatu lintasan* (path) *pada graf $\mathcal{G}$ adalah barisan busur $(e_1, e_2, \ldots, e_n)$ dengan e_i bersisian* (adjacent) *dengan e_{i+1}. Panjang lintasan $(e_1, e_2, \ldots, e_n)$ adalah n.*

Definisi 2.26. *Jarak antara simpul v_1 dan simpul v_2 adalah panjang dari lintasan terpendek antara v_1 dan v_2. Diameter dari suatu graf adalah jarak terbesar antara sembarang dua simpul pada graf. Jika dua simpul tidak mempunyai lintasan, dikatakan diameter dari graf tersebut adalah ∞.*

Definisi 2.27. *Suatu lintasan $(e_1, e_2, \ldots, e_n)$ adalah lingkaran* (cycle) *jika $e_n = e_1$. Untuk suatu graf tidak berarah,* girth *adalah panjang terpendek dari sebarang lingkaran pada graf.*

Definisi 2.28. *Suatu graf terhubung jika untuk sebarang dua simpul v_1 dan v_2 terdapat lintasan dari v_1 ke v_2 atau dari v_2 ke v_1. Graf terhubung kuat jika untuk sebarang dua simpul v_1 dan v_2 terdapat lintasan dari v_1 ke v_2 dan dari v_2 ke v_1.*

Definisi 2.29. *Graf dikatakan graf teratur$-k$ jika setiap simpul tepat mempunyai k busur, jika $\mathcal{G}$ tidak mempunyai gelung maka setiap simpul mempunyai k tetangga. Graf teratur-k mempunyai derajat k untuk setiap simpulnya.*

2.4.1 Graf Cayley

Pada subbab ini akan dibahas mengenai graf Cayley yang akan digunakan dalam konstruksi graf ekspander maupun konstruksi fungsi *hash*.

Definisi 2.30. *Misalkan G adalah grup dan $\mathcal{S}$ adalah himpunan tak kosong, himpunan bagian berhingga dari G, dengan $\mathcal{S} = \mathcal{S}^{-1}$. Graf Cayley $\mathcal{G}(G, \mathcal{S})$ adalah graf dengan himpunan simpul G dan himpunan sisi $E = \{(x,y) \in G, \exists S \in \mathcal{S} : y = xS\}$. Karena $\mathcal{S} = \mathcal{S}^{-1}$ maka graf yang dihasilkan tidak berarah.*

Beberapa sifat graf Cayley dirangkum dalam Lema 2.31.

Lema 2.31. *Misal $\mathcal{G}(G,\mathcal{S})$ adalah graf Cayley, dengan $k = |\mathcal{S}|$. Maka:*

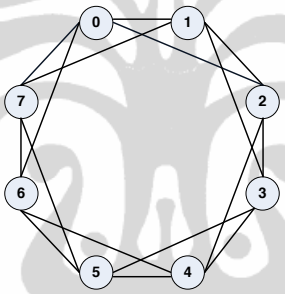
(1) *$\mathcal{G}(G,\mathcal{S})$ adalah graf sederhana dan teratur-k.*

(2) *$\mathcal{G}(G,\mathcal{S})$ tidak mempunyai gelung jika dan hanya jika $1 \notin \mathcal{S}$.*

(3) *$\mathcal{G}(G,\mathcal{S})$ terhubung jika dan hanya jika $\mathcal{S}$ membangun G.*

Contoh 2.32.

1. (Petit, 2009) Graf Cayley $\mathcal{G}(G,\mathcal{S})$ dengan $G = \mathbb{Z}/8\mathbb{Z}$, dengan $\mathcal{S} = \{1,2,6,7\}$. Graf Cayley yang dihasilkan seperti Gambar 2.2 merupakan graf sederhana, tidak berarah, dan teratur-4 dengan 8 simpul.



Sumber: telah diolah kembali dariPetit (2009)

Gambar 2.2. Graf Cayley $\mathcal{G}(G,\mathcal{S})$

2. Misalkan $G = \langle PGL_2(3), * \rangle$ dengan $*$ adalah perkalian matriks. $PGL_2(3)$ adalah grup seperti yang didefinisikan pada Subbab 2.2. Grup ini mempunyai 24 elemen. Untuk mempermudah, elemen-elemen matriks diubah menjadi bilangan bulat tebal dari **1**,..., **24**. Elemen-elemen tersebut jika dituliskan berdasarkan elemen terkecil dalam kelas ekivalen adalah :

$$\mathbf{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \mathbf{7} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \mathbf{13} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \quad \mathbf{19} = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\mathbf{2} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad \mathbf{8} = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix} \quad \mathbf{14} = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix} \quad \mathbf{20} = \begin{pmatrix} 2 & 1 \\ 2 & 0 \end{pmatrix}$$

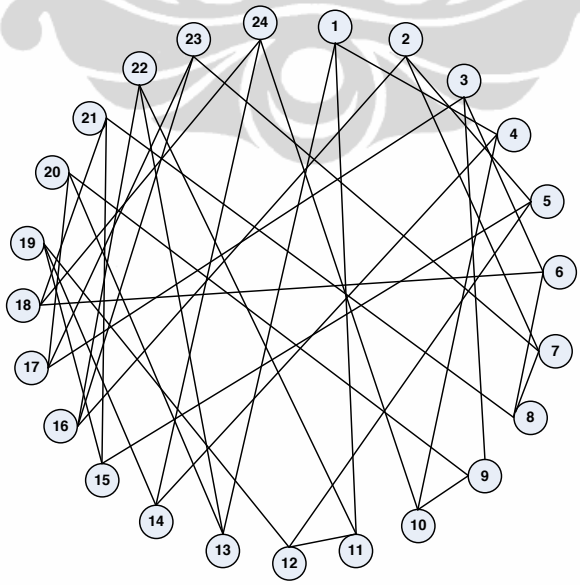
$$\mathbf{3} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} \quad \mathbf{9} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \quad \mathbf{15} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \mathbf{21} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$$

$$\mathbf{4} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \quad \mathbf{10} = \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix} \quad \mathbf{16} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix} \quad \mathbf{22} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\mathbf{5} = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \quad \mathbf{11} = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix} \quad \mathbf{17} = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \quad \mathbf{23} = \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}$$

$$\mathbf{6} = \begin{pmatrix} 1 & 0 \\ 2 & 2 \end{pmatrix} \quad \mathbf{12} = \begin{pmatrix} 0 & 1 \\ 2 & 2 \end{pmatrix} \quad \mathbf{18} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \quad \mathbf{24} = \begin{pmatrix} 2 & 1 \\ 2 & 2 \end{pmatrix}$$

Himpunan $\mathcal{S} = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix} \right\}$. Graf Cayley $\mathcal{G}(G,\mathcal{S})$ yang dihasilkan merupakan graf sederhana, berarah dan teratur-3 dengan 24 simpul. Graf ini disebut graf $X^{2,3}$, seperti pada Gambar 2.3.



Gambar 2.3. Graf Cayley $X^{2,3}$

2.4.2 Graf Ekspander

Misalkan graf $\mathcal{G} = (V,E)$ merupakan graf sederhana yang mempunyai n simpul. Untuk sub himpunan F dari himpunan simpul V, didefinisikan *edge boundary* batas busur dari F yang dinotasikan dengan ∂F sebagai himpunan busur yang menghubungkan F dengan $\overline{F}$ (komplemen F). Atau batas busur F adalah himpunan busur (v,w) sehingga $v \in F$ dan $w \notin F$.

Selanjutnya didefinisikan parameter ekspansi (*expansion parameter*) dari graf $\mathcal{G}$. Pada beberapa literatur parameter ekspansi juga disebut *expanding constant* (Goldreich) atau *isoperimetric constant* (Davidoff dkk., 2003).

Definisi 2.33 (Davidoff dkk. (2003))**.** *Parameter ekspansi dari graf $\mathcal{G}$ didefinisikan sebagai*

$$h(\mathcal{G}) = \min\left\{ \frac{|\partial F|}{\min\{|F|, |V-F|\}} : F \subseteq V \right\}.$$

dengan notasi $|.|$ menyatakan banyaknya anggota himpunan.

Pada saat berbicara mengenai graf ekspander yang dimaksud adalah koleksi tak hingga graf yang setiap graf pada koleksi tersebut mempunyai kemiripan sifat. Sebagai contoh pada saat berbicara mengenai graf ekspander teratur-k mengacu pada koleksi tak hingga graf yang setiap graf pada koleksi tersebut merupakan graf teratur-k.

Definisi 2.34 (Davidoff dkk. (2003))**.** *Misal $(\mathcal{G}_m)$ merupakan keluarga graf $\mathcal{G}_m = (V_m, E_m)$ yakni graf teratur-k, berhingga, dan terhubung dengan $m,k \in \mathbb{N}, m \geq 1, k \geq 2$. $(\mathcal{G}_m)$ adalah keluarga graf ekspander jika $|V_m| \to \infty$ pada saat $m \to \infty$ dan terdapat $\varepsilon > 0$, sedemikian sehingga $h(\mathcal{G}_m) \geq \varepsilon$ untuk setiap $m \geq 1$.*

Graf ekspander mempunyai dua sifat yang sangat berguna untuk aplikasi kriptografi. Sifat pertama berkaitan dengan *random walk* pada sembarang graf ekspander. *Walk* pada graf $\mathcal{G} = (V,E)$ adalah barisan simpul-simpul $v_1, v_2, \ldots \in V$ sedemikian sehingga simpul v_{i+1} adalah tetangga dari simpul v_i untuk setiap i. Jika v_{i+1} dipilih secara acak dengan kemungkinan yang sama dari tetangga v_i, maka *walk* semacam ini disebut *random walk* pada graf $\mathcal{G}$. Sifat tersebut dijelaskan pada Teorema 2.35 dan Akibat 2.36 berdasarkan Goldreich, Lecture 10.

Teorema 2.35. *Misal $X_0, X_1, \ldots, X_l$ merupakan* random walk *pada graf ekspander dengan n simpul yang diawali dari X_0 dengan X_i, $0 \leq i \leq n-1$ adalah label simpul. Untuk setiap δ terdapat $l = O(\log \frac{1}{\delta})$ sedemikian sehingga untuk setiap simpul v, berlaku*

$$\left| Pr(X_l = v) - \frac{1}{n} \right| < \delta.$$

Akibat 2.36. *Pada sembarang graf ekspander dengan n simpul, setelah* $l = O(\log n)$ random walk *didapatkan faktor tetap dari distribusi uniform.*

Maksudnya setelah *random walk* mencapai panjang $O(\log n)$, dengan n simpul pada suatu graf ekspander sama artinya dengan memilih l simpul secara acak. Ini mengakibatkan sebarang simpul pada graf ekspander dengan n simpul mempunyai kemungkinan yang sama untuk dipilih.

Sifat yang kedua berkaitan dengan *girth* pada graf ekspander. *Girth* merupakan panjang dari *cycle* terpendek pada graf. Graf ekspander mempunyai *girth* yang besar. Sifat ini yang berkaitan dengan keamanan dari fungsi *hash*, yaitu *collision resistant* yang dibahas pada Bab 4.

2.4.3 Graf Ramanujan

Salah satu keluarga graf ekspander adalah graf Ramanujan. Graf Ramanujan merupakan graf ekspander dengan parameter ekspansi terbesar yang diketahui jika dilihat dari nilai eigen pertama dan nilai eigen kedua pada matriks *adjacency*-nya (Davidoff dkk., 2003).

Definisi 2.37 (Davidoff dkk. (2003))**.** *Suatu graf* G *teratur-k, berhingga, dan terhubung merupakan graf Ramanujan jika untuk setiap nilai eigen* λ *selain* $\pm k$ *dari matriks* adjacency G*, maka*

$$|\lambda| \leq 2\sqrt{k-1}.$$

Beberapa hasil yang berkaitan dengan graf Ramanujan dirangkum dalam Teorema 2.38.

Teorema 2.38 (Davidoff dkk. (2003))**.** *Untuk nilai k berikut, terdapat keluarga tak hingga graf Ramanujan teratur-k:*

(1) $k = p + 1$*, dengan p bilangan prima ganjil.*

(2) $k = 3$.

(3) $k = q + 1$*, dengan q pangkat dari bilangan prima.*

Beberapa graf Ramanujan di antaranya graf ekspander LPS, yang merupakan graf teratur-$p+1$, dengan p bilangan prima ganjil; graf Pizer merupakan graf teratur-$p+1$, dengan p bilangan prima ganjil dan graf Morgenstern merupakan graf teratur-q, dengan q pangkat dari bilangan prima ganjil.

2.5 Bilangan Bulat Gauss (*Gaussian Integer*)

Materi pada subbab ini diambil dari Davidoff dkk. (2003). Pada subbab ini akan dibahas tentang karakteristik bilangan bulat sebagai jumlah dari dua bilangan kuadrat yang merupakan hasil dari Fermat dan Euler.

Definisi 2.39. *Gelanggang bilangan bulat Gauss didefinisikan sebagai berikut*

$$\mathbb{Z}[i] = \left\{a + bi : a,b \in \mathbb{Z}, i^2 = \text{-}1\right\}.$$

dengan operasi penjumlahan dan perkalian bilangan kompleks.

Gelanggang ini merupakan subgelanggang $\mathbb{C}$. Untuk $\alpha = a + bi \in \mathbb{Z}[i]$, didefinisikan *norm* dari α yang dinotasikan dengan $\mathcal{N}(\alpha)$, $\mathcal{N}(\alpha) = \alpha\overline{\alpha} = a^2 + b^2$. Dapat kita lihat suatu bilangan bulat merupakan penjumlahan dari dua bilangan kuadrat jika dan hanya jika bilangan bulat tersebut adalah *norm* dari suatu bilangan bulat Gauss. *Norm* pada gelanggang bilangan bulat Gauss adalah multiplikatif yaitu $\mathcal{N}(\alpha\beta) = \mathcal{N}(\alpha)\mathcal{N}(\beta)$.

Beberapa hasil berikut berasal dari aritmatika bilangan bulat Gauss, $\mathbb{Z}[i]$.

Teorema 2.40. *Misalkan $p \in \mathbb{N}$ adalah bilangan prima ganjil. Pernyataan-pernyataan berikut ini ekivalen:*

(1) $p \equiv 1 \pmod 4$.

(2) -1 *adalah bilangan kuadrat di $\mathbb{F}_p$, yakni kongruensi $x^2 \equiv \text{-}1 \pmod p$ mempunyai solusi di $\mathbb{Z}$.*

(3) *p adalah penjumlahan dari dua bilangan kuadrat.*

Akibat 2.41. *Suatu bilangan bulat $n \geq 2$ merupakan penjumlahan dari dua bilangan kuadrat jika dan hanya jika setiap bilangan bulat $p \equiv 3 \pmod 4$ muncul dengan pangkat genap dalam faktorisasi prima dari n.*

2.6 Quaternion

Penjelasan mengenai quaternion diambil dari Davidoff dkk. (2003) kecuali dinyatakan lain. Konsep ini sangat penting untuk mengkonstruksi graf ekspander LPS dan melakukan serangan terhadap fungsi *hash* dari graf ekspander LPS.

Definisi 2.42. *Suatu aljabar quaternion Hamiltonian atas gelanggang R, dinotasikan dengan $\mathbb{H}(R)$, merupakan aljabar asosiatif sedemikian sehingga:*

(1) $\mathbb{H}(R)$ *adalah modul bebas atas simbol* $1,i,j,k$: $\mathbb{H}(R) = \{a_0 + a_1 i + a_2 j + a_3 k : a_0, a_1, a_2, a_3 \in R\}$.

(2) 1 *merupakan unit perkalian.*

(3) $i^2 = j^2 = k^2 =$ -1.

(4) $ij =$ -$ji = k; jk =$ -$kj = i; ki =$ -$ik = j$.

Untuk $\alpha = a_0 + a_1 i + a_2 j + a_3 k \in \mathbb{H}(R)$, konjugatnya adalah $\bar{\alpha} = a_0 - a_1 i - a_2 j - a_3 k$. *Norm* dari α merupakan bilangan bulat positif $\mathcal{N}(\alpha) = \alpha\bar{\alpha} = a_0^2 + a_1^2 + a_2^2 + a_3^2$. Sama halnya dengan bilangan bulat Gauss, *norm* quaternion juga bersifat multiplikatif.

Terdapat isomorfisma antara quaternion atas lapangan hingga $\mathbb{H}(\mathbb{F}_q)$, dengan q merupakan pangkat bilangan prima ganjil dan aljabar matriks 2×2 atas $\mathbb{H}(\mathbb{F}_q)$. Hasil ini dinyatakan dalam :

Lema 2.43. *Misalkan q bilangan yang merupakan pangkat bilangan prima ganjil. Terdapat* $x, y \in \mathbb{F}_q$ *sedemikian sehingga* $x^2 + y^2 + 1 = 0$. $\mathbb{H}(\mathbb{F}_q)$ *isomorfis dengan aljabar matriks* 2×2 *atas lapangan* $\mathbb{F}_q$, $M_2(\mathbb{F}_q)$. *Isomorfisma tersebut didefinisikan sebagai*

$$\begin{array}{rcl} \psi : \mathbb{H}\left(\mathbb{F}_q\right) & \rightarrow & M_2\left(\mathbb{F}_q\right) \\ \psi(a_0 + a_1 i + a_2 j + a_3 k) & \mapsto & \begin{pmatrix} a_0 + a_1 x + a_3 y & -a_1 y + a_2 + a_3 x \\ -a_1 y - a_2 + a_3 x & a_0 - a_1 x - a_3 x \end{pmatrix} \end{array}$$

Lema berikut menjelaskan hubungan antara *norm* dari quaternion dan determinan dari matriks yang didefinisikan berdasarkan isomorfisma pada Lema 2.43.

Lema 2.44. *Misalkan* $\psi : \mathbb{H}\left(\mathbb{F}_q\right) \rightarrow M_2\left(\mathbb{F}_q\right)$ *merupakan isomorfisma yang didefinisikan pada Lema* 2.43. *Untuk* $r \in \mathbb{H}\left(\mathbb{F}_q\right)$, $\det(\psi(r)) = \mathcal{N}(r)$. *Jika r bilangan real, yakni* $r = \bar{r}$, *maka* $\psi(r)$ *adalah matriks skalar.*

Lema 2.45. *Untuk* $q \in \mathbb{H}(\mathbb{Z})$, *pernyataan berikut ekuivalen:*

(1) *q dapat dibalik di* $\mathbb{H}(\mathbb{Z})$.

(2) $N(q) = 1$.

(3) $q \in \{\pm 1, \pm i, \pm j, \pm k\}$.

Quaternion $q \in \mathbb{H}(\mathbb{Z}) - \{0\}$ dikatakan unsur satuan jika q memenuhi Lema 2.45. Dua quaternion $q, q' \in \mathbb{H}(\mathbb{Z})$ berasosiasi jika terdapat unsur satuan $u, u' \in \mathbb{H}(\mathbb{Z})$, sedemikian sehingga $q' = uqu'$. Suatu quaternion q adalah bilangan prima jika q bukan unsur satuan di $\mathbb{H}(\mathbb{Z})$ dan jika $q = rs$ maka r atau s adalah unsur satuan. q bilangan prima jika dan hanya jika $\mathcal{N}(q)$ adalah bilangan prima di $\mathbb{Z}$. Terdapat faktorisasi prima untuk sebarang quaternion akan tetapi faktorisasi ini tidak unik. Walaupun begitu terdapat faktorisasi unik untuk himpunan quaternion q dengan $\mathcal{N}(q) = p^k$ dengan p bilangan prima ganjil dan $k \in \mathbb{N}$.

Misalkan p bilangan prima ganjil. Berdasarkan Teorema Jacobi (Davidoff dkk., 2003, Teorema 2.4.1), persamaan

$$a_0^2 + a_1^2 + a_2^2 + a_3^2 = p$$

mempunyai $8(p+1)$ solusi di $\mathbb{Z}$. Setiap solusi berhubungan dengan bilangan bulat quaternion $q = a_0 + a_1 i + a_2 j + a_3 k$ dengan *norm* p. Jika $p \equiv 1 (\bmod 4)$, salah satu dari a_i adalah ganjil, sedangkan sisanya genap. Jika $p \equiv 3 (\bmod 4)$, salah satu a_i genap, sedangkan lainnya ganjil. Pada setiap kasus, satu koordinat, sebut dengan a_i^0, dibedakan (*distinguished*). Jika $a_i^0 \neq 0$, maka di antara 8 bentuk uq yang berasosiasi dengan q, terdapat tepat satu yang mempunyai $\left|a_i^0\right|$ sebagai komponen ke-0-nya. Jika $a_i^0 = 0$, yang mungkin terjadi jika $p \equiv 3 (\bmod 4)$, maka dua quaternion yang berasosiasi, yaitu uq dan -uq akan mempunyai $a_0 = 0$. Pada kasus ini salah satunya dapat dijadikan sebagai yang dibedakan (*distinguished*). Oleh karena itu, terdapat $p+1$ solusi yang dibedakan (*distinguished solutions*) dari persamaan

$$a_0^2 + a_1^2 + a_2^2 + a_3^2 = p$$

sehingga quaternion yang berhubungan memenuhi $q \equiv 1 (\bmod 2)$ atau $q \equiv i + j + k (\bmod 2)$. Pada solusi ini, q dan $\overline{q}$ muncul pada saat $a_0 > 0$, dan hanya salah satu yang muncul pada saat $a_0 = 0$. Sehingga terbentuk himpunan

$$S_p = \{q_1, \overline{q_1}, \ldots, q_s, \overline{q_s}, r_1, \ldots, r_t\}$$

dengan q_i mempunyai $a_0^{(i)} > 0$, r_j mempunyai $b_0^{(j)} = 0$, dan $q_i \overline{q_i} = -r_j^2$. Pada himpunan ini $2s + t = \left|S_p\right| = p + 1$.

Definisi 2.46. *Kata tereduksi atas S_p adalah kata atas alfabet S_p, yang tidak mempunyai subkata dengan bentuk $q_i\overline{q_i}, \overline{q_i}q_i, r_j^2 (i = 1, \ldots s; j = 1, \ldots, t)$. Panjang kata adalah banyaknya simbol yang muncul.*

Teorema 2.47. *Misalkan $k \in \mathbb{N}$, misal $q \in \mathbb{H}(\mathbb{Z})$ sedemikian sehingga $\mathcal{N}(q) = p^k$. Maka q mempunyai faktorisasi unik $q = \varepsilon p^r w_m$, dengan ε adalah kesatuan di $\mathbb{H}(\mathbb{Z})$, w_m adalah kata tereduksi dengan panjang m atas S_p, dan $k = 2r + m$.*

Selanjutnya pandang himpunan Λ' yang merupakan subhimpunan $\mathbb{H}(\mathbb{Z})$ yang didefinisikan sebagai:

$$\begin{aligned} \Lambda' &= \{q = a_0 + a_1 i + a_2 j + a_3 k \in \mathbb{H}(\mathbb{Z}) : q \equiv 1 (\bmod 2) \\ &\text{atau} \\ q &\equiv i + j + k (\bmod 2), \mathcal{N}(q) \text{ adalah pangkat dari } p\}. \end{aligned}$$

Akibat 2.48. *Setiap elemen $q \in \Lambda'$ dengan $\mathcal{N}(q) = p^k$ mempunyai faktorisasi unik $q = \pm p^r w_m$, dengan $r \in \mathbb{N}$, w_m adalah kata tereduksi dengan panjang m atas S_p dan $k = 2r + m$.*

Teorema 2.47 dan Akibat 2.48 berperan sangat penting dalam menentukan tumbukan (*collision*) pada fungsi *hash* dari graf ekspander LPS yang dibahas pada Bab 4.

2.7 *Continued Fraction*

Materi pada subbab ini diambil dari Collins (1999).

Definisi 2.49. *Suatu* continued fraction *adalah persamaan yang berbentuk*

$$r = a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{a_3 + \ldots}}}$$

dengan a_i dan b_i dapat berupa bilangan rasional, bilangan real, atau bilangan kompleks. Jika $b_i = 1$ untuk semua i, maka disebut simple continued fraction. *Jika banyaknya suku pada persamaan berhingga disebut* finite continued fraction, *sebaliknya disebut* infinite continued fraction. *Bilangan a_i disebut* partial quotients. *Jika persamaan dipotong setelah* partial quotient *ke-k, maka nilai dari persamaan yang dihasilkan disebut konvergen ke-k dari* continued fraction.

Notasi yang biasa digunakan untuk *continued fraction* bilangan r adalah $r = [a_0, a_2, a_2, a_3, \ldots]$. Jika a_0 bilangan bulat biasanya dipisahkan dengan semikolon dengan koefisien yang lain, sehingga menjadi $r = [a_0; a_2, a_2, a_3, \ldots]$. Untuk memperjelas konsep ini diberikan beberapa contoh.

Contoh 2.50.

1. $\frac{3}{2} = 1 + \frac{1}{2} = [1;2]$.

2. $$\frac{19}{51} = \frac{1}{\frac{51}{19}} = \frac{1}{2+\frac{13}{19}} = \frac{1}{2+\frac{1}{\frac{19}{13}}}$$
$$= \frac{1}{2+\frac{1}{1+\frac{6}{13}}} = \frac{1}{2+\frac{1}{1+\frac{1}{\frac{13}{6}}}}$$
$$= \frac{1}{2+\frac{1}{1+\frac{1}{2+\frac{1}{6}}}}$$
$$= [0;2,1,2,6]$$

Pada bab ini telah dibahas beberapa konsep penting yang akan digunakan pada bab-bab selanjutnya. Pada bab selanjutnya dibahas mengenai konstruksi graf ekspander LPS, konstruksi umum fungsi *hash* dari graf ekspander, dan konstruksi fungsi *hash* dari graf ekspander Cayley. Selain itu juga dijelaskan mengenai konstruksi fungsi *hash* dari graf ekspander LPS serta aspek keamanannya.

BAB 3
FUNGSI *HASH* DARI GRAF EKSPANDER LPS

Seperti yang telah disinggung pada Bab 1, fungsi *hash* yang ada sekarang banyak menggunakan konstruksi Merkle-Damgård. Pada bab ini dibahas konstruksi fungsi *hash* dari graf ekspander, yang tidak menggunakan konstruksi Merkle-Damgård. Bab ini dibagi menjadi tiga subbab. Subbab pertama membahas mengenai konstruksi graf ekspander LPS, yang meliputi penentuan bilangan prima yang digunakan, grup yang digunakan dan sifat-sifat yang dimiliki oleh graf ekspander LPS. Pada subbab kedua dibahas tentang konstruksi fungsi *hash* dari graf ekspander, meliputi pembahasan mengenai konstruksi umum, konstruksi fungsi *hash* dari graf ekspander Cayley dan konstruksi fungsi *hash* dari graf ekspander LPS. Sedangkan subbab terakhir membahas mengenai keamanan fungsi *hash* dari graf ekspander LPS. Pembuktian pada bab ini berdasarkan pada referensi yang digunakan, kecuali disebutkan lain.

3.1 Konstruksi Graf Ekspander LPS

Konstruksi graf ekspander LPS berdasarkan pada grup yang telah didefinisikan pada Subbab 2.2. Konstruksi yang dijelaskan di sini berdasarkan Davidoff dkk. (2003, Bab 4) atau Lubotzky dkk. (1988).

Misalkan ℓ dan p bilangan prima berbeda dengan $\ell \equiv p \equiv 1 \pmod 4$. Berdasarkan Subbab 2.6 diperoleh $\ell+1$ solusi berbeda untuk persamaan $a_0^2+a_1^2+a_2^2+a_3^2=\ell$ dengan $a_0 > 0$ dan ganjil dan a_1, a_2, a_3 genap. Solusi tersebut bersesuaian dengan bilangan bulat quaternion $q = a_0 + a_1 i + a_2 j + a_3 k$ dengan *norm* $\mathcal{N}(q) = \ell$. Selanjutnya, untuk setiap quaternion tersebut dihubungkan dengan matriks yang entri-entrinya di gelanggang $\mathbb{Z}[i]$, yaitu

$$\widetilde{S} = \begin{pmatrix} a_0 + a_1 i & a_2 + a_3 i \\ \text{-}a_2 + a_3 i & a_0 - a_1 i \end{pmatrix} \tag{3.1}$$

Entri-entri pada $\widetilde{S}$ dipetakan ke elemen $\mathbb{F}_p$ dengan menerapkan homomorfisma gelanggang

$$\begin{array}{rcl} \phi : \mathbb{Z}[i] & \rightarrow & \mathbb{F}_p \\ a+bi & \mapsto & a+b\text{i} \end{array} \tag{3.2}$$

dengan i adalah akar kuadrat dari -1 modulo p, menjadi matriks di $PGL_2(\mathbb{F}_p)$,

$$S = \begin{pmatrix} a_0 + a_1\mathrm{i} & a_2 + a_3\mathrm{i} \\ \text{-}a_2 + a_3\mathrm{i} & a_0 - a_1\mathrm{i} \end{pmatrix}.$$

Matriks S tersebut mempunyai determinan ℓ, karena $\begin{vmatrix} a_0 + a_1\mathrm{i} & a_2 + a_3\mathrm{i} \\ \text{-}a_2 + a_3\mathrm{i} & a_0 - a_1\mathrm{i} \end{vmatrix} = a_0^2 + a_1^2 + a_2^2 + a_3^2 = \ell$. $\ell + 1$ matriks S tersebut membentuk himpunan $\mathcal{S}$. Himpunan $\mathcal{S}$ simetrik, karena setiap $S \in \mathcal{S}$ terdapat $S^{-1} \in \mathcal{S}$. Ini berasal dari fakta bahwa di $PGL_2(\mathbb{F}_p)$ didapat

$$\begin{aligned} S(a_0, a_1, a_2, a_3)S(a_0, \text{-}a_1, \text{-}a_2, \text{-}a_3) &= \begin{pmatrix} a_0 + a_1\mathrm{i} & a_2 + a_3\mathrm{i} \\ \text{-}a_2 + a_3\mathrm{i} & a_0 - a_1\mathrm{i} \end{pmatrix} \begin{pmatrix} a_0 - a_1\mathrm{i} & \text{-}a_2 - a_3\mathrm{i} \\ a_2 - a_3\mathrm{i} & a_0 + a_1\mathrm{i} \end{pmatrix} \\ &= \begin{pmatrix} a_0^2 + a_1^2 + a_2^2 + a_3^2 & 0 \\ 0 & a_0^2 + a_1^2 + a_2^2 + a_3^2 \end{pmatrix} \\ &= \begin{pmatrix} \ell & 0 \\ 0 & \ell \end{pmatrix} \\ &\equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

Himpunan $\mathcal{S}$ tidak membangkitkan seluruh grup $PGL_2(\mathbb{F}_p)$. Semua matriks di $\mathcal{S}$ mempunyai determinan ℓ yang merupakan kuadrat modulo p. Sehingga hanya matriks dengan determinant yang merupakan sisa kuadratik $(\bmod p)$ yang dibangkitkan. Berdasarkan Tillich dan Zemor (2008) subgrup yang dibangkitkan oleh himpunan $\mathcal{S}$ isomorfis dengan $PSL_2(\mathbb{F}_p)$.

Definisi 3.1 (Davidoff dkk. (2003))**.** *Graf ekspander LPS merupakan Graf Cayley seperti Definisi* 2.30, yang dinotasikan dengan $X^{\ell,p}$, didefinisikan sebagai berikut

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

Graf $X^{\ell,p}$ di atas mempunyai beberapa sifat yang telah dibuktikan oleh Lubotzky dkk. (1988).

Teorema 3.2 (Davidoff dkk. (2003))**.** *Graf $X^{\ell,p}$ seperti pada Definisi* 3.1 adalah graf teratur-$\ell + 1$ yang terhubung dan Ramanujan. Lebih lanjut,

(1) Jika ℓ adalah kuadrat modulo p, maka $X^{\ell,p}$ adalah bukan graf bipartit dengan $\frac{p(p^2-1)}{2}$ simpul, yang memenuhi perkiraan *girth* $g(X^{\ell,p}) \geq 2\log_\ell p$.

(2) Jika ℓ adalah bukan kuadrat modulo p, maka $X^{\ell,p}$ adalah graf bipartit dengan $p(p^2-1)$ simpul, yang memenuhi perkiraan *girth* $g(X^{\ell,p}) \geq 4\log_\ell p - \log_\ell 4$.

Pada Teorema 3.2 banyaknya simpul Graf $X^{\ell,p}$ didapatkan dari Lema 2.20, sedangkan graf $X^{\ell,p}$ teratur-$\ell+1$ karena dari Lema 2.31 banyaknya elemen $\mathcal{S}$ adalah $\ell+1$. Pembuktian graf $X^{\ell,p}$ adalah Ramanujan serta perkiraan *girth* dapat dilihat di Lubotzky dkk. (1988).

Teorema 3.3 (Lubotzky dkk. (1988))**.** *Jika* $n = \left|X^{\ell,p}\right|$ *maka diameter* $X^{\ell,p} \leq 2\log_\ell n + 2\log_\ell 2 + 1$.

Berdasarkan Teorema 3.3 untuk n yang relatif besar, diameter graf ekspander LPS relatif kecil.

3.2 Konstruksi Fungsi *Hash* dari Graf Ekspander

Pembahasan konstruksi fungsi *hash* dari graf ekspander diawali dengan pembahasan tentang persyaratan yang diperlukan suatu graf ekspander menjadi kandidat fungsi *hash*. Berdasarkan Petit (2009) setidaknya suatu graf ekspander yang digunakan untuk fungsi *hash* memenuhi beberapa persyaratan berikut:

1. Ekspansi yang besar: persyaratan ini menjamin nilai hash dari pesan yang relatif singkat terdistribusi dengan baik dalam himpunan output. Maksudnya semua simpul pada graf mempunyai kemungkinan yang sama untuk menjadi nilai *hash*.

2. Diameter yang kecil. Diameter yang kecil mengakibatkan semua simpul dapat menjadi output untuk pesan yang relatif pendek.

3. *Girth* yang besar. Semakin besar *girth* menjamin tidak adanya tumbukan (*collision*) yang pendek dan *girth* juga merupakan batas jarak antara dua pesan yang bertumbukan.

4. Efisiensi. Ini berkaitan dengan menghitung tetangga dari simpul yang diberikan.

5. Permasalahan *collision resistance*, *second-preimage resistance* dan *preimage resistance* harus sulit.

Selanjutnya dibahas konstruksi fungsi *hash* dari graf ekspander secara umum, konstruksi fungsi *hash* dari graf ekspander Cayley dan konstruksi fungsi *hash* dari graf ekspander LPS.

3.2.1 Konstruksi Fungsi *Hash* dari Graf Ekspander

Sebelum membahas mengenai konstruksi fungsi *hash* dari graf ekspander LPS terlebih dahulu dibahas mengenai ide konstruksi fungsi *hash* dari graf ekspander secara umum. Algoritma-algoritma dan contoh yang dibahas pada bab ini berdasarkan Petit (2009) dengan penyesuaian.

Konstruksi Secara Umum

Suatu fungsi *hash* dari graf ekspander tidak berarah ditentukan oleh deskripsi dari graf ekspander, suatu busur awal, dan suatu fungsi pengurutan ketetanggaan (*neighbor ordering function*). Tentu saja graf ekspander yang digunakan memenuhi persyaratan yang dibahas pada awal Subbab 3.2. Konstruksi umum fungsi *hash* dari graf ekspander tidak berarah dapat dilihat pada Algoritma 3.1.

Algoritma 3.1. Konstruksi umum fungsi *hash* dari graf ekspander

INPUT : Pesan m dengan panjang sembarang
OUTPUT : Nilai *hash* yang berupa simpul di graf $\mathcal{G}$

1. **Parameter awal** : $\mathcal{G}$ adalah graf ekspander tidak berarah teratur-k. Atur $k' = k-1$. Definisikan fungsi urutan ketetanggaan $\theta : E \times \{0,\ldots,k'-1\} \to V$, sedemikian sehingga untuk sembarang $e = (v_1, v_2) \in E$, himpunan $\{\theta(e,i) | 0 \leq i \leq k'-1\} \cup \{v_1\}$ adalah himpunan tetangga dari v_2. Busur awal adalah $e_0 = (v_{-1}, v_0)$.

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 $\mu - 2$,

   Hitung $v_{i+1} = \theta(e_i, m_i), e_{i+1} = (v_i, v_{i+1})$, terakhir kembalikan nilai $v_{\mu-1}$.

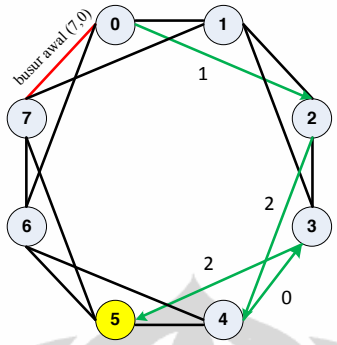
   Proses ini sama artinya dengan melakukan perjalanan (*walk*) melalui simpul-simpul $v_0, v_1, \ldots, v_{\mu-1}$.

4. **Penyelesaian.** Nilai *hash* adalah simpul terakhir yang dikunjungi *walk*.

Contoh 3.4.

1. (Petit, 2009) Misalkan $G = \langle \mathbb{Z}/8\mathbb{Z}, + \rangle$, dengan $\mathcal{S} = \{1,2,6,7\}$. Pada Gambar 3.1 diilustrasikan perhitungan nilai *hash* untuk pesan $m = 1202$, dengan busur awal $(7,0)$ dan fungsi urutan ketetanggaan yang didefinisikan: untuk

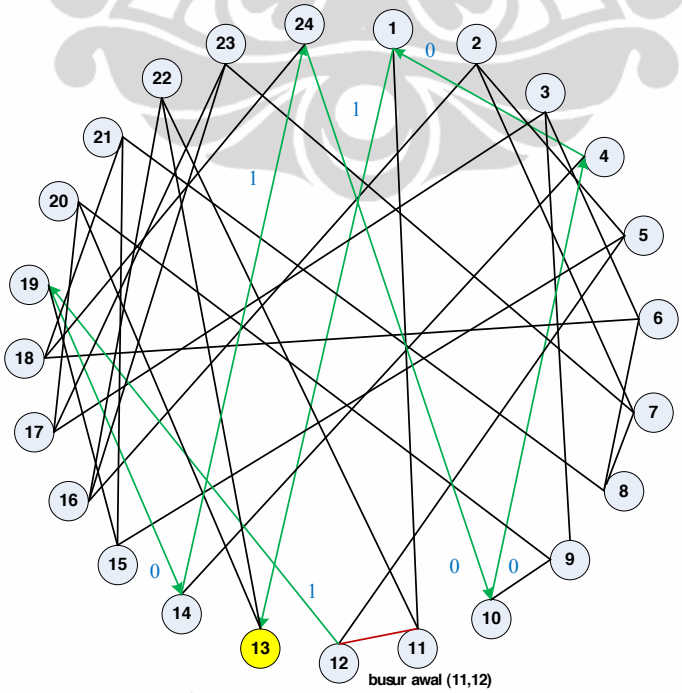
sebarang $e \in E, \theta(e,0) < \theta(e,1) < \theta(e,2)$. Nilai *hash* yang diperoleh yaitu $H(m) = 5$



Sumber: telah diolah kembali dariPetit (2009)

Gambar 3.1. Menentukan nilai *hash* untuk pesan $m = 1202$

2. Misalkan $G = \langle PGL_2(3), * \rangle$ dengan $*$ adalah perkalian matriks, dan himpunan $\mathcal{S} = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 2 \\ 2 & 1 \end{pmatrix} \right\}$. Contoh perhitungan nilai *hash* untuk pesan $m = 1010001$, busur awal $(11,12)$ dan fungsi urutan ketetanggaan yang didefinisikan: untuk sebarang $e \in E, \theta(e,0) < \theta(e,1)$. Ilustrasinya dapat dilihat pada Gambar 3.2. Nilai *hash* yang diperoleh yaitu $H(m) = 13$.



Gambar 3.2. Menentukan nilai *hash* untuk pesan $m = 1010001$

Konstruksi di atas hanya mungkin dilakukan jika tetangga dari sebarang simpul di graf dapat dihitung secara efisien, tentu saja ini berlaku untuk graf Cayley. Perlu diperhatikan juga karena nilai *hash* yang dihasilkan adalah suatu simpul, maka dibutuhkan suatu aturan untuk mengubah simpul menjadi bilangan biner atau bilangan lain.

Konstruksi Fungsi *Hash* dari Cayley Graf Ekspander

Konstruksi untuk graf ekspander Cayley dijelaskan pada Algoritma 3.2.

Algoritma 3.2. Konstruksi Fungsi *Hash* dari graf ekspander Cayley

INPUT : Pesan m dengan panjang sembarang
OUTPUT : Nilai *hash* yang berupa simpul di graf $\mathcal{G}$

1. **Parameter awal** : $\mathcal{G}$ adalah graf ekspander Cayley tidak berarah teratur-k. $k = |\mathcal{S}|$ dengan $\mathcal{S}$ adalah himpunan generator seperti yang didefinisikan pada Subbab 2.4.1. Atur $k' = k-1$. Busur awal adalah busur $\left(g_0 S_{-1}^{-1}, g_0\right)$ dengan $g_0 \in G$ dan $S_{-1} \in \mathcal{S}$. Fungsi urutan ketetanggaan didefinisikan sebagai θ : $\mathcal{S} \times \{0,\ldots,k'-1\} \to \mathcal{S}$ yang memetakan sebuah elemen di $\mathcal{S}$ dan bilangan digit-k ke elemen $\mathcal{S}$, sehingga untuk setiap $S \in \mathcal{S}$,

$$\left\{\theta(S,i) \,|\, 0 \le i \le k'-1\right\} \cup \left\{S^{-1}\right\} = \mathcal{S}.$$

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 $H(m) = g_0 \prod_{i=0}^{\mu-1} S_i = g_0 S_0 S_1 \ldots S_{\mu-1}$;

4. **Penyelesaian.** Nilai *hash* dari pesan m adalah $H(m)$.

3.2.2 Konstruksi Fungsi *Hash* dari Graf Ekspander LPS

Graf LPS merupakan graf Cayley. Graf ekspander LPS yang digunakan adalah graf yang dibentuk dari bilangan prima berbeda ℓ dan p yang kongruen 1 modulo 4. ℓ merupakan bilangan prima kecil dan bilangan kuadrat modulo p. Sedangkan p relatif besar. Besar bilangan prima p dalam Charles dkk. (2007) disarankan 1024 bit. Sehingga graf ekspander LPS $X^{\ell,p}$ yang digunakan merupakan graf teratur-$\ell+1$ dengan banyaknya simpul $\frac{p(p^2-1)}{2}$. Fungsi *hash* dari graf ekspander LPS, berdasarkan

Charles dkk. (2007) merupakan fungsi *hash* tanpa kunci seperti pada Subbab 2.3. Algoritma untuk fungsi *hash* dari graf ekspander LPS dijelaskan pada Algoritma 3.3.

Algoritma 3.3. Konstruksi Fungsi *Hash* dari graf ekspander LPS

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 yang didefinisikan pada Subbab 3.1. Misal $k = \ell + 1$ dan $k' = k - 1$. Busur awal adalah busur $\left(g_0 S_{-1}^{-1}, g_0\right)$ dengan $g_0 \in PSL_2(\mathbb{F}_p)$ dan $S_{-1} \in \mathcal{S}$. Fungsi urutan ketetanggaan didefinisikan sebagai $\theta : \mathcal{S} \times \{0,\ldots,k'-1\} \to \mathcal{S}$ yang memetakan sebuah elemen di $\mathcal{S}$ dan bilangan digit-k ke elemen $\mathcal{S}$, sehingga untuk setiap $S \in \mathcal{S}$,

$$\left\{\theta(S,i) \,|\, 0 \le i \le k'-1\right\} \cup \left\{S^{-1}\right\} = \mathcal{S}.$$

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 $H(m) = g_0 \prod_{i=0}^{\mu-1} S_i = g_0 S_0 S_1 \ldots S_{\mu-1}$;

4. **Penyelesaian.** Nilai *hash* dari pesan m adalah $H(m)$.

3.3 Keamanan Fungsi *Hash* dari Graf Ekspander LPS

Keamanan fungsi *hash* seperti yang dijelaskan pada Subbab 2.3 dapat dibuat interpretasi dalam teori graf (Petit, 2009). Untuk graf yang tidak berarah dapat dijabarkan dalam Tabel 3.1.

Tabel 3.1. Hubungan antara sifat-sifat fungsi *hash* dan graf tidak berarah

Sifat-sifat fungsi *hash*	**Sifat-sifat graf tidak berarah**
collision resistance	menemukan lingkaran
preimage resistance	menemukan lintasan
distribusi output	sifat *expanding*
jarak minimal *collision*	*girth*

Graf ekspander LPS mempunyai dua sifat yang relevan untuk keamanan fungsi *hash*, yakni:

1. Graf ekspander LPS merupakan ekspander yang baik, sehingga mempunyai parameter ekspansi yang besar. Oleh karena itu berdasarkan Akibat 2.36: *random walk* pada graf ekspander LPS mendekati distribusi uniform pada saat panjang *walk* adalah suatu konstanta dikalikan logaritma dari banyak simpul. Berdasar sifat ini dapat diketahui, distribusi nilai *hash* mendekati distribusi uniform pada saat ukuran teks adalah suatu konstanta dikalikan ukuran nilai *hash*.

2. Berdasarkan Teorema 3.2, graf ekspander LPS mempunyai *girth* yang besar, yang artinya tidak ada lingkaran pendek, sehingga untuk mencari lingkaran pada graf ini relatif sulit. Karena tumbukan (*collision*) berkaitan dengan masalah menemukan lingkaran, maka pada fungsi *hash* dari graf ekspander LPS untuk mencari tumbukan (*collision*) merupakan masalah yang sulit. Menurut Charles dkk. (2007) untuk menemukan tumbukan (*collision*) untuk fungsi *hash* dari graf ekspander LPS dapat direduksi menjadi Masalah 3.5.

   Masalah 3.5. *Misalkan p dan ℓ dua bilangan prima berbeda yang kongruen 1 modulo 4 dan ℓ adalah bilangan kuadrat modulo p. Misal $S = S_1, S_2, \ldots, S_{\ell+1}$ adalah generator $PSL_2(\mathbb{F}_p)$ yang didefinisikan pada Subbab* 3.1. *Tentukan perkalian*

$$\prod_{i=0}^{\mu-1} S_i^{e_i} = I \tag{3.3}$$

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

   Teorema 3.6 (Charles dkk. (2007))**.** *Mencari tumbukan (*collision*) pada fungsi* hash *dari graf ekspander LPS dengan ukuran input $O(\log p)$ ekuivalen dengan menyelesaikan Masalah* 3.5.

   Bukti.

   Misalkan x merupakan input dengan panjang s dan y input dengan panjang t sehingga $H(x) = H(y)$. Didapat:

$$\begin{aligned} H(x) &= H(y) \\ g_0 S_0 S_1 \ldots S_{s-1} &= g_0 S_0 S_1 \ldots S_{t-1} \\ \prod_{i=0}^{r-1} S_i &= \prod_{j=0}^{s-1} S_j \end{aligned}$$

Sehingga

$$\prod_{i=0}^{r-1} S_i \left(\prod_{j=0}^{s-1} S_j \right)^{-1} = 1$$

Berdasarkan definisi $\mathcal{S}$, invers setiap generator ada di $\mathcal{S}$, sehingga relasi di atas merupakan relasi antar generator. Relasi ini bukan relasi yang mudah karena dalam graf ekspander LPS input yang berbeda akan menghasilkan lintasan yang berbeda pula. Karena panjang masing-masing input adalah $O(\log p)$, banyaknya generator pada relasi di atas adalah $O(\log p)$, sehingga dapat disimpulkan untuk mencari tumbukan (*collision*) ekuivalen dengan Masalah 3.5. □

Pada bab ini telah dibahas mengenai konstruksi fungsi *hash* dari graf ekspander secara umum, konstruksi fungsi *hash* dari graf ekspander Cayley, dan konstruksi fungsi *hash* dari graf ekspander LPS. Pembahasan tentang keamanan fungsi *hash* dari graf ekspander LPS juga diberikan. Keamanan fungsi *hash* dari graf ekspander LPS dapat diinterpretasikan dari sifat-sifat fungsi *hash*, yaitu *collision resistant*, *second preimage resistant*, dan *preimage resistant*. Permasalahan tumbukan pada fungsi *hash* dari graf ekspander LPS dirangkum dalam Masalah 3.5. Pada bab selanjutnya akan dibahas mengenai menentukan tumbukan pada fungsi *hash* dari graf ekspander LPS beserta contoh serangan. Salah satu modifikasi juga diberikan beserta contoh serangan terhadap fungsi *hash* modifikasi.

BAB 4
TUMBUKAN (*COLLISION*) PADA FUNGSI *HASH* DARI GRAF EKSPANDER LPS

Pada bab ini dibahas mengenai serangan yang dilakukan terhadap fungsi *hash* yang berdasar graf ekspander LPS. Serangan ini berdasarkan Tillich dan Zemor (2008). Pada subbab pertama dibahas tentang menemukan tumbukan (*collision*) pada fungsi *hash* dari graf ekspander LPS yang dibagi menjadi 3 (tiga) tahap. Subbab selanjutnya membahas tentang contoh serangan, dan subbab terakhir membahas mengenai modifikasi yang dapat dilakukan untuk menghindari serangan. Pembuktian pada bab ini berdasarkan pada referensi yang digunakan, kecuali disebutkan lain.

4.1 Menentukan Tumbukan (*Collision*)

Masalah 3.5 yang dijelaskan pada Subbab 3.3 dipercaya merupakan masalah yang sulit berdasarkan Charles dkk. (2007). Akan tetapi, berdasarkan Tillich dan Zemor (2008) Masalah 3.5 dapat diselesaikan. Pada subbab ini dibahas mengenai penyelesaian Masalah 3.5 berdasarkan makalah Tillich dan Zemor (2008).

Pemfaktoran di $PSL(\mathbb{F}_p)$ secara langsung kelihatannya sulit. Strategi yang digunakan adalah dengan mengubah matriks di $PSL_2(\mathbb{F}_p)$ menjadi matriks Ω di $SL_2(\mathbb{Z}[i])$, dan kemudian memfaktorkan matriks tersebut menjadi perkalian dari generator S yang telah diubah, seperti pada Persamaan 3.1. Selanjutnya memetakan hasil pemfaktoran kembali ke $PSL_2(\mathbb{F}_p)$ akan menyelesaikan Masalah 3.5. Himpunan matriks yang relevan adalah

$$\Omega = \left\{ \left(\begin{array}{cc} a+bi & c+di \\ -c+di & a-bi \end{array} \right) \middle| (a,b,c,d) \in E_w; w > 0; w \in \mathbb{Z} \right\} \tag{4.1}$$

dengan E_w adalah himpunan 4-tupel $(a,b,c,d) \in \mathbb{Z}^4$ sedemikian sehingga

$$\left\{ \begin{array}{rcrcrcll} a^2 & + & b^2 & + & c^2 & + & d^2 & = \ell^w \\ & & & a & > 0, & a & \equiv 1 & (\mathrm{mod}\ 2) \\ & & b & \equiv & c & \equiv & d \equiv 0 & (\mathrm{mod}\ 2) \end{array} \right. \tag{4.2}$$

Sekarang pandang himpunan $\widetilde{S}$, yakni himpunan $\ell+1$ matriks $\widetilde{S}(a,b,c,d)$ dengan entri-entri di $\mathbb{Z}[i]$ seperti yang didefinisikan pada Persamaan 3.1. Himpunan

$\widetilde{S}$ merupakan subhimpunan Ω yang berhubungan dengan 4-tupel (a,b,c,d) di E_1, yang juga merupakan versi yang telah diubah dari himpunan generator S. Selanjutnya akan ditunjukkan sebarang matriks di Ω mempunyai faktorisasi yang unik.

Teorema 4.1 (Tillich dan Zemor (2008))**.** *Sebarang matriks M di Ω dapat ditulis secara unik sebagai perkalian*

$$M = \pm \ell^r \widetilde{S}_0 \widetilde{S}_1 \dots \widetilde{S}_{e-1}$$

dengan $\log_\ell(\det M) = e + 2r$ *dan* $\widetilde{S}_i \in \widetilde{S}$ *dan* $\widetilde{S}_i \widetilde{S}_{i+1} \neq \ell \widetilde{I}$ *untuk* $i \in \{0, \dots, e-2\}$.

Bukti.
Berdasarkan definisi Ω (Persamaan 4.1) terdapat korespondensi satu-satu antara Ω dengan bilangan bulat quaternion $a + bi + cj + dk$ dengan *norm* l^w seperti pada Persamaan 4.2. Juga berdasarkan Definisi 2.46 terdapat korespondensi satu-satu antara himpunan kata tereduksi atas S_p yang didefinisikan pada Subbab 2.6 dan himpunan kata atas $\widetilde{S}$ yang tidak memuat kata $\widetilde{S}_i \widetilde{S}_{i+1} \neq \ell \widetilde{I}$. Himpunan unsur satuan $\pm\{1,i,j,k\} \in \mathbb{H}(\mathbb{Z})$ berhubungan dengan himpunan matriks $\pm U$ dengan entri di $\mathbb{Z}[i]$, yaitu

$$U = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix} \right\}.$$

Di antara matriks-matriks di U hanya matriks identitas yang ada di Ω karena matriks lain tidak memenuhi satu atau lebih syarat pada Persamaan 4.2. Sehingga berdasarkan Akibat 2.48, sebarang matriks $M \in \Omega$ dapat dituliskan secara unik sebagai perkalian $M = \pm \ell^r \widetilde{S}_1 \widetilde{S}_2 \dots \widetilde{S}_e$ dengan $w = e + 2r$ dan semua $\widetilde{S}_i \in \widetilde{S}$. Sembarang perkalian matriks dengan bentuk

$$\begin{pmatrix} a+bi & c+di \\ -c+di & a-bi \end{pmatrix} \quad \text{dengan } a \equiv 1 (\bmod 2), b \equiv c \equiv d \equiv 0 (\bmod 2)$$

menghasilkan matriks dengan sifat kongruensi yang sama. Seluruh anggota $\widetilde{S}$ mempunyai sifat kongruensi tersebut dan perkalian dalam bentuk $\ell^r \widetilde{S}_1 \widetilde{S}_2 \dots \widetilde{S}_e$ juga mempunyai sifat kongruensi yang sama. Perkalian dengan elemen lain selain matriks identitas di $\pm U$ tidak menghasilkan sifat kongruensi yang sama. Sehingga hanya matriks identitas yang dapat digunakan dalam faktorisasi unik matriks M di Ω. $\square$

Selanjutnya tumbukan (*collision*) dicari dengan tahap-tahap sebagai berikut:

Tahap 1 : Mengubah matriks identitas di $PSL_2(\mathbb{F}_p)$ ke matriks M di Ω
Tahap ini untuk menentukan matriks M di Ω yang tidak dalam bentuk $\ell^r\widetilde{I}$, dengan $\widetilde{I}$ matriks identitas di Ω. Jika entri kompleks diganti dengan nilai korespondensinya di $\mathbb{F}_p$ (dengan pemetaan ϕ (Persamaan 3.2)) akan didapat matriks identitas di $PSL_2(\mathbb{F}_p)$. Tahap ini ekuivalen dengan mencari $a,b,c,d \in \mathbb{Z}$ sedemikian hingga untuk suatu bilangan bulat positif w,

$$\begin{cases} a^2 + b^2 + c^2 + d^2 = \ell^w \\ a > 0, \quad a \equiv 1 \pmod 2 \\ b \equiv c \equiv d \equiv 0 \pmod 2 \\ b^2 + c^2 + d^2 \neq 0 \end{cases} \tag{4.3}$$

Kondisi terakhir digunakan untuk mencegah matriks dengan bentuk $l^r\widetilde{I}$.

Untuk mempermudah perhitungan, eksponen w dibuat genap. Tulis $w = 2k$, demikian pula $b = 2px$, $c = 2py$, $d = 2pz$. Lalu tentukan solusi bilangan bulat (a,x,y,z) untuk persamaan

$$a^2 + 4p^2\left(x^2+y^2+z^2\right) = \ell^{2k}$$

didapat

$$\left(\ell^k - a\right)\left(\ell^k + a\right) = 4p^2\left(x^2+y^2+z^2\right)$$

Pilih $a = \ell^k - 2mp^2$ untuk suatu bilangan bulat m. Persamaan menjadi $\left(\ell^k - a\right)\left(\ell^k + a\right) = 2mp^2\left(2\ell^k - 2mp^2\right)$. Sehingga x, y, z memenuhi persamaan

$$x^2 + y^2 + z^2 = m\left(\ell^k - mp^2\right) \tag{4.4}$$

Selanjutnya pilih k yang merupakan bilangan bulat terkecil sehingga $\ell^k - 4p^2 > 0$. Kondisi ini menjamin a positif dan meminimalkan panjang faktorisasi. Selanjutnya pilih $m = 1$ atau $m = 2$ untuk memastikan a tetap positif.

Klaim 4.2 (Tillich dan Zemor (2008))**.** *Untuk $m = 1$ atau $m = 2$, bilangan $m\left(\ell^k - mp^2\right)$ adalah jumlah dari tiga bilangan kuadrat.*

Bukti.
Berdasarkan Teorema Legendre (Burton, 2007, Teorema 13.5): sebarang bilangan bulat adalah jumlah dari tiga bilangan kuadrat, kecuali dalam bentuk $4^s(8t+7)$ dengan s dan t adalah bilangan bulat. Asumsikan $m = 1$ tidak memenuhi klaim

tersebut. Dengan kata lain $\ell^k - p^2$ bukan jumlah dari tiga bilangan kuadrat. Artinya terdapat bilangan bulat tidak negatif s dan t sedemikian hingga $\ell^k - p^2 = 4^s(8t+7)$. Sebagai catatan s harus positif dalam kasus ini. Pandang sekarang bahwa $2\left(\ell^k - 2p^2\right) = 4^s(16t+14) - 2p^2$. Bilangan ini bukanlah bilangan ganjil atau kelipatan 4. Sehingga bilangan tersebut bukanlah dalam bentuk $4^u(8v+7)$. Ini menunjukkan untuk $m = 2$, $m(\ell^k - mp^2)$ merupakan jumlah dari tiga bilangan kuadrat. □

Sekarang tinggal menentukan x, y, z yang memenuhi Persamaan 4.4. Salah satu cara untuk menyelesaikan ini adalah mengurangi $m\left(\ell^k - mp^2\right)$ dengan bilangan acak x^2 dan berharap hasilnya adalah bilangan N yang merupakan jumlah dua bilangan kuadrat. Selanjutnya berdasarkan Akibat 2.41 dinyatakan bilangan bulat $n \geq 2$ dapat dinyatakan sebagai jumlah dari dua bilangan kuadrat jika dan hanya jika faktor prima yang kongruen dengan $3 \pmod 4$ muncul dengan eksponen genap. Pendekatannya adalah mencoba mencari nilai x sehingga N dalam bentuk $2^s p'$ dengan p' adalah bilangan prima yang kongruen dengan 1 modulo 4. Jika $m = 1$, dipilih x genap dan karena $\ell^k - p^2 \equiv 0 \pmod 4$ (karena ℓ dan p kongruen 1 modulo 4), cek apakah $\left(\ell^k - p^2 - x^2\right)/4$ adalah bilangan prima kongruen 1 modulo 4. Jika memenuhi nilai N yang digunakan adalah $N = (\ell^k - p^2 - x^2)/4$.

Sekarang tentukan y dan z sedemikian sehingga

$$y^2 + z^2 = N. \tag{4.5}$$

Persamaan 4.5 dapat diselesaikan dengan ekspansi *continued fraction*. Konvergensi $\frac{p_n}{q_n}$ dari ekspansi *continued fraction* bilangan real x didapat secara induksi dari formula

$$\begin{aligned} (p_{-1}, q_{-1}) &= (0,1) \\ (p_0, q_0) &= (1,0) \end{aligned}$$

dan untuk semua nilai tidak negatif dari n dengan $q_n x - p_n \neq 0$

$$\begin{aligned} a_n &= \left[-\frac{q_{n-1}x - p_{n-1}}{q_n x - p_n}\right] \\ p_{n+1} &= a_n p_n + p_{n-1} \\ q_{n+1} &= a_n q_n + q_{n-1} \end{aligned}$$

dengan $[.]$ menotasikan bagian bilangan bulat.

Barisan $(q_n)_{n\geq 0}$ *strictly increasing* dan $\frac{p_n}{q_n}$ adalah pendekatan rasional yang sa-

ngat baik untuk x, yang memenuhi

$$\left| x - \frac{p_n}{q_n} \right| < \frac{1}{q_n q_{n+1}} \tag{4.6}$$

Sesuai dengan Teorema 2.40, pada kasus ini -1 adalah sisa kuadratik modulo N, karena $N \equiv 1 (\bmod 4)$. Hasil-hasil tersebut digunakan dalam Teorema 4.3.

Teorema 4.3 (Tillich dan Zemor (2008))**.** *Misalkan N adalah bilangan prima yang kongruen* 1 *modulo* 4*, R adalah akar kuadrat dari* -1 *modulo N dan* $\xi \stackrel{def}{=} \frac{R}{N}$*. Misalkan* $\frac{p_i}{q_i}$ *adalah konvergen ke-i yang berasosiasi dengan ekspansi* continued fraction *dari* ξ*. Misalkan n merupakan bilangan bulat unik sehingga* $q_n < \sqrt{N} < q_{n+1}$*. Maka*

$$q_n^2 + (q_n R - p_n N)^2 = N.$$

Bukti.
Berdasar Pertidaksamaan 4.6, $\left| \frac{R}{N} - \frac{p_n}{q_n} \right| < \frac{1}{q_n q_{n+1}}$. Sehingga $\left| q_n \frac{R}{N} - p_n \right| < \frac{1}{q_{n+1}} < \frac{1}{\sqrt{N}}$ (karena $q_{n+1} > \sqrt{N}$). Mengakibatkan $|q_n R - p_n N| < \sqrt{N}$. Juga $q_n < \sqrt{N}$. Jumlahkan pertidaksamaan tersebut didapat $q_n^2 + (q_n R - p_n N)^2 < 2N$. Didapat

$$\begin{aligned} q_n^2 + (q_n R - p_n N)^2 &\equiv q_n^2 + q_N^2 R^2 \,(\bmod N) \\ &\equiv 0 \,(\bmod N) \qquad \text{karena } R^2 \equiv \text{-}1 \bmod N \end{aligned}$$

Karena bilangan bulat positif yang kurang dari $2N$ yang habis dibagi N adalah N, didapat

$$q_n^2 + (q_n R - p_n N)^2 = N.$$

□

Berdasarkan Tillich dan Zemor (2008), banyaknya x yang harus dicoba untuk mendapatkan N yang sesuai akan berorder $O(\log p)$; kompleksitas untuk melakukan ekspansi *continued fraction* juga berorder $O(\log p)$. Sehingga kompleksitas total untuk tahap ini sangat kecil dan akan berorder $O(\log p)$.

Tahap 2 : Faktorisasi

Tentukan faktorisasi M yang dijamin oleh Teorema 4.1 sebagai perkalian elemen-elemen di $\widetilde{S}$,

$$M = \pm \ell^r \widetilde{S}_0 \widetilde{S}_1 \ldots \widetilde{S}_{e-1}.$$

Faktorisasi M dilakukan sebagai berikut:

Pertama tentukan bilangan bulat terbesar r sehingga ℓ^r membagi keempat entri pada matriks M. Misalkan M' merupakan matriks dengan $M = \ell^r M'$. Notasikan $\ell+1$ elemen yang membentuk generator yang telah diubah dari himpunan $\widetilde{S}$ sebagai $\widetilde{S}_1, \ldots, \widetilde{S}_{\ell+1}$. Selanjutnya tentukan elemen paling kanan pada faktorisasi M' dengan menghitung semua perkalian dengan bentuk $\pm M'\widetilde{S}_i^{-1}$. Berdasarkan Teorema 4.1, salah satu perkalian tersebut akan ada di Ω. Perkalian tersebut akan berkorespondensi dengan faktorisasi M' dengan menghilangkan elemen terakhir dari faktorisasi. Untuk memperjelas pandang

$$\begin{aligned} M' &= \widetilde{S}_0\widetilde{S}_1\ldots\widetilde{S}_{e-1} \\ &\Rightarrow \widetilde{S}_0\widetilde{S}_1\ldots\widetilde{S}_{e-2} = \widetilde{S}_0\widetilde{S}_1\ldots\widetilde{S}_{e-1}\widetilde{S}_i^{-1} = M'\widetilde{S}_i^{-1} \end{aligned}$$

Oleh karena itu, $\widetilde{S}_i$ unik dengan $\pm M'\widetilde{S}_i^{-1}$ di himpunan Ω adalah elemen terakhir dari faktorisasi M'. Diberikan definisi Ω untuk memeriksa perkalian $\pm M'\widetilde{S}_i^{-1}$ ada di Ω mudah secara perhitungan. Lanjutkan proses dan proses ini akan berhenti setelah $\log_\ell(\det M) - 2r$ langkah karena pada setiap iterasi, determinan pada bagian kiri dari faktorisasi dibagi dengan ℓ dan karena determinan M' adalah ℓ^{w-2r}. Kompleksitas dari langkah ini jelas proporsional terhadap panjang faktorisasi, yakni paling besar w. Berikutnya dapat dilihat kita dapat memilih w mendekati $2\log_\ell p$, sehingga kompleksitas tidak melebihi $O(\log p)$.

Tahap 3 :
Tahap terakhir yang dilakukan adalah dengan menggunakan homomorfisma ϕ (Persamaan 3.2) pada setiap entri dari matriks-matriks hasil faktorisasi pada Tahap 2. Didapat faktorisasi dari matriks identitas di $PSL_2(\mathbb{F}_p)$ sebagai perkalian generator-generator di $\mathcal{S}$:

$$I \equiv \widetilde{M} \equiv S_0 S_1 \ldots S_{e-1}$$

Hasil faktorisasi ini menyelesaikan Masalah 3.5.

4.2 Contoh Menentukan Tumbukan (*Collision*) untuk $p > 2^{1024}$

Komputasi dilakukan dengan menggunakan Program Maple seperti pada Lampiran Tillich dan Zemor (2008). Pada contoh ini, dipilih

$$
\begin{aligned}
p = {} & 179769313486231590772930519078902473361797697894230657273430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474124377767893424865485276302219601246094119453082952085005768838150682342462881473913110540827237163350510684586298239947245938479716304835356329624224139329
\end{aligned}
$$

yang merupakan bilangan prima pertama $p > 2^{1024}$ sehingga $p \equiv 1 \pmod 4$. Selanjutnya pandang fungsi *hash* yang bersesuaian dengan graf $X^{5,p}$, dalam hal ini $\ell = 5$, bilangan prima terkecil dengan $\ell \equiv 1 \pmod 4$. Dapat dicek 5 juga merupakan sisa kuadratik modulo p yang dipilih. Keenam generator dari $X^{5,p}$, diberikan oleh matriks dengan entri-entri di $\mathbb{Z}[i]$,

$$
\widetilde{S}_1 = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \qquad \widetilde{S}_2 = \begin{pmatrix} 1+2i & 0 \\ 0 & 1-2i \end{pmatrix} \qquad \widetilde{S}_3 = \begin{pmatrix} 1 & 2i \\ 2i & 1 \end{pmatrix}
$$

$$
\widetilde{S}_4 = \begin{pmatrix} 1 & -2i \\ -2i & 1 \end{pmatrix} \qquad \widetilde{S}_5 = \begin{pmatrix} 1-2i & 0 \\ 0 & 1+2i \end{pmatrix} \qquad \widetilde{S}_6 = \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix}
$$

atau dengan S_j di $PSL_2(\mathbb{F}_p)$.

Tahap Pertama

Tentukan a, b, c, d yang memenuhi Persamaan 4.3. Pilih k yang merupakan bilangan bulat yang lebih besar dari $\log_5(2p^2)$ untuk membuat sisi kanan dari Persamaan 4.4 bernilai positif. Didapatkan $k = 885$. Selanjutnya hitung $5^k - p^2$ dalam bentuk $4u$ dengan u ganjil. Didapat $u \not\equiv 7 \pmod 8$, yang artinya u dapat dituliskan dalam jumlah tiga bilangan kuadrat. Diperoleh $u \equiv 1 \pmod 4$, kemudian kurangkan dari u suatu kuadrat dengan bentuk $4v^2$ dan cek apakah $u - 4v^2$ bilangan prima. Jika telah didapatkan bilangan prima, bilangan tersebut akan kongruen 1 modulo 4, sehingga dapat dituliskan sebagai jumlah dari dua bilangan kuadrat. v pertama sehingga $N = u - 4v^2$ adalah bilangan prima adalah $v = 1359$.

$$N = 961077099389994091403699389514599846739356619387360496720$$
$$638298830053889431370436914694339083946394772569050806697$$
$$773918573784053299739321092529478047541381566589627894897$$
$$434183976030112772310378873116134592981859183897935909138 1$$
$$444110368295605432723685358966253184757851034009950050858$$
$$486229566302988461087419965626346022851090927042184794017 2$$
$$953820591473923333605158431648029181324234805090838458855 0$$
$$965301861057342692490431549785186548354710315681833870619 9$$
$$907293342692791265665917261114865908104212017141124490806 6$$
$$203957409499211524818105891551295094259012382828993478933$$
$$475176669898290177424846213262396893520697$$

Selanjutnya akan dituliskan N sebagai jumlah dari dua bilangan kuadrat. Pertama tentukan akar kuadrat dari -1 modulo N. Akar kuadrat dipilih secara acak sehingga representasi di $\{1,\ldots,N-1\}$ adalah yang terbesar. Akar kuadrat -1 modulo N yang terbesar adalah

$$R = 898202571311019024315403325434100887061527989671428747353 8$$
$$104469444608364339521870596870741289763947777785504466931$$
$$428606278929554087724869282413511892466665506147393809333$$
$$689798127442633133387330091526135172120564171497298432712 0$$
$$206892240739092438325004287888948210118810788157288999123 5$$
$$457033348878181562857318410600012902302756589822857910854 5$$
$$416139103187874664834695346309395594824905072871071693669 1$$
$$536460506179949395665961270042615590769825319719416602011 9$$
$$637241300012551867927927406795424924630957758373955828600$$
$$630677828765563337241374279247168942747581278799514191517 6$$
$$373996787216637095154383830571320161161 23$$

Selanjutnya ekspan R/N ke dalam *continued fraction* dan hitung n terbesar sedemikian sehingga p_n/q_n adalah konvergen dari ekspansi *continued fraction* yang ke-n dan $q_n < \sqrt{N}$. Pada kasus ini didapat $n = 638$. Nilai p_n dan q_n yang berhubungan adalah:

$$p_n = 880986188840627248473909766507146786880982031336646046992$$
$$463508716580587181258382278025049778868887704800277904314 1$$
$$871816367700450485164672438203121267209775211446384827639 1$$
$$561443280598300211778924622775943737537470455568134200246$$
$$629135828736370919963408747551972872753796499527472381602$$
$$4845125499320205661121$$

$$
\begin{aligned}
q_n &= 94265556347468059512199261224467485647223762784754397836533031522268527366611438539610825608699478704342095253642008126030826301443533038869603524802913757871628076844951011381898308423636740552460792941348312507988865922100841216960530031212094671183052963181226462870959919063701421822357304468948828069905 1
\end{aligned}
$$

Selanjutnya atur

$$
\begin{aligned}
x &= 2q_n \\
&= 18853111269493611902439852244893497129444752556950879567306606304453705473322287707922165121739895740868419050728401625206165260288706607773920704960582751574325615368990202276379661684727348110492158588269662501597773184420168243392106006242418934236610592636245292574191983812740284364471460893789765613981 02 \\
y &= 2(p_nN - q_nR) \\
&= 53843323835077832200181655642120259940740845443239064438478273988022780542201506179105388020315602967387633239031172668654198496820535571586033398452755397068058926959879926468899198769435035532186888071331070855322680217446641109287078727085964196045373858927991233965666941591128488318630645885222450035587 2 \\
z &= 4 \times 1359 \\
&= 5436
\end{aligned}
$$

dan diperoleh $5^k - p^2 = 4u = x^2 + y^2 + z^2$. Selanjutnya atur

$$
\begin{aligned}
a &= \sqrt{5^{2k} - 16up^2} \\
b &= 2px \\
c &= 2py \\
d &= 2pz
\end{aligned}
$$

Tahap Kedua

Faktorkan matriks M di Ω

$$
M = \begin{pmatrix} a+bi & c+di \\ -c+di & a-bi \end{pmatrix}
$$

menjadi perkalian dari $\widetilde{S}_j$. Berdasarkan Teorema 4.1, terdapat cara unik untuk memfaktorkan M, dan panjang faktorisasi pada contoh ini adalah $2k = 1770$, yakni

$$M = \pm M_0 \dots M_{2k-1}$$

dalam hal ini $M_j = \widetilde{S}_{\sigma_j}$ dengan $\sigma_j \in \{1,2,\dots,6\}$. Untuk menghitung σ, pertama kalikan pada sisi kanan M dengan enam matriks $\widetilde{S}_1 \dots \widetilde{S}_6$ dan periksa apakah semua entri merupakan kelipatan 5. Jika hal itu terjadi kita telah menemukan $\widetilde{S}_{\sigma(1769)}$. Selanjutnya hitung $M' = M\widetilde{S}^{-1}_{\sigma(1769)} = \frac{1}{5}M\widetilde{S}_{7-\sigma(1769)}$ dan lakukan secara rekursif, memeriksa $M'\widetilde{S}_j$ paling banyak enam kali untuk mendapatkan $\sigma(1768)$ dan seterusnya. Setelah 1770 iterasi didapat matriks identitas I atau -I.

Nilai $\sigma_j, j = 0,\dots 1769$ adalah:

1, 1, 1, 1, 5, 6, 2, 3, 3, 2, 1, 3, 1, 5, 3, 6, 4, 1, 3, 5, 5, 6, 5, 1, 3, 5, 3, 1, 2,
3, 2, 6, 6, 6, 5, 3, 1, 5, 3, 1, 4, 2, 3, 5, 4, 4, 5, 6, 5, 5, 5, 6, 6, 6, 4, 2, 1, 3,
3, 1, 3, 3, 6, 5, 1, 2, 1, 1, 5, 1, 4, 5, 6, 5, 5, 5, 1, 2, 6, 6, 2, 2, 4, 2, 6, 2, 3,
1, 5, 1, 1, 1, 5, 5, 5, 6, 5, 6, 5, 3, 5, 1, 1, 1, 2, 2, 1, 3, 3, 3, 6, 4, 1, 2, 6, 2,
6, 3, 1, 1, 2, 3, 1, 2, 2, 4, 4, 4, 2, 1, 3, 3, 1, 4, 5, 5, 3, 2, 3, 3, 1, 2, 4, 5, 4,
1, 3, 5, 6, 3, 6, 5, 3, 3, 6, 2, 1, 2, 2, 2, 2, 6, 3, 6, 6, 5, 4, 2, 2, 2, 3, 1, 4, 5,
6, 5, 6, 4, 5, 4, 1, 2, 4, 4, 2, 3, 6, 6, 4, 6, 3, 1, 5, 5, 4, 6, 3, 3, 3, 1, 2, 3, 5,
3, 3, 1, 1, 5, 1, 4, 2, 3, 6, 5, 5, 6, 6, 6, 2, 3, 1, 5, 4, 5, 3, 1, 2, 2, 6, 3, 2, 2,
4, 2, 3, 6, 4, 6, 5, 6, 4, 4, 1, 2, 3, 5, 1, 3, 5, 1, 5, 6, 5, 3, 5, 3, 2, 6, 2, 1, 5,
5, 1, 5, 3, 1, 5, 6, 3, 1, 5, 1, 4, 5, 3, 3, 2, 1, 5, 5, 4, 6, 5, 1, 2, 2, 6, 2, 4, 5,
1, 5, 3, 1, 2, 4, 5, 4, 6, 4, 6, 3, 3, 3, 6, 5, 3, 6, 6, 6, 4, 1, 2, 1, 2, 2, 1, 1, 2,
3, 3, 1, 5, 5, 6, 3, 6, 5, 1, 3, 3, 6, 2, 2, 4, 1, 1, 1, 4, 6, 5, 3, 1, 4, 6, 6, 6, 4,
5, 4, 1, 4, 1, 2, 6, 5, 4, 5, 6, 4, 6, 3, 2, 4, 5, 6, 5, 3, 5, 3, 3, 5, 4, 4, 4, 1, 3,
2, 3, 3, 6, 4, 6, 4, 2, 1, 4, 2, 3, 1, 1, 3, 2, 3, 6, 2, 4, 6, 3, 2, 3, 6, 4, 5, 3, 1,
4, 1, 4, 6, 5, 6, 4, 2, 2, 3, 5, 1, 4, 5, 4, 6, 3, 2, 3, 6, 5, 5, 3, 3, 6, 5, 1, 3, 1,
2, 2, 3, 3, 5, 5, 6, 6, 3, 5, 3, 5, 4, 5, 1, 3, 3, 1, 2, 1, 5, 4, 4, 4, 6, 5, 5, 4, 5,
6, 3, 2, 6, 5, 5, 5, 5, 4, 6, 2, 6, 3, 3, 2, 3, 2, 2, 4, 6, 5, 3, 1, 4, 1, 3, 2, 4, 5,
5, 1, 5, 3, 6, 4, 6, 3, 5, 1, 2, 3, 2, 2, 3, 5, 3, 2, 4, 5, 4, 1, 1, 1, 4, 1, 1, 3, 1,
4, 6, 6, 5, 6, 6, 4, 1, 3, 3, 5, 4, 1, 3, 5, 5, 3, 5, 5, 3, 2, 1, 1, 4, 4, 4, 2, 4, 6,

3, 5, 4, 6, 3, 3, 5, 6, 3, 1, 3, 3, 1, 5, 6, 5, 4, 2, 4, 5, 4, 2, 6, 4, 1, 4, 1, 4, 6,
6, 4, 5, 3, 6, 5, 6, 6, 5, 3, 1, 1, 4, 2, 3, 5, 1, 5, 6, 5, 6, 4, 4, 6, 2, 4, 5, 3, 3,
5, 4, 4, 5, 5, 1, 4, 5, 1, 3, 3, 6, 6, 5, 3, 2, 2, 2, 4, 4, 2, 4, 2, 6, 6, 5, 4, 5, 6,
4, 4, 2, 1, 3, 6, 6, 5, 5, 5, 6, 5, 3, 1, 4, 1, 2, 6, 6, 5, 4, 6, 6, 3, 3, 1, 3, 5, 1,
5, 3, 6, 6, 6, 4, 6, 6, 5, 4, 6, 2, 4, 5, 4, 4, 5, 4, 5, 3, 5, 4, 5, 4, 6, 3, 5, 4, 1,
3, 3, 5, 4, 4, 5, 6, 4, 5, 6, 4, 4, 4, 2, 1, 1, 5, 6, 3, 5, 4, 1, 2, 2, 6, 2, 6, 4, 6,
3, 5, 6, 4, 5, 1, 4, 2, 6, 3, 5, 4, 4, 1, 5, 3, 3, 1, 1, 4, 1, 2, 2, 4, 6, 3, 2, 3, 3,
5, 5, 3, 1, 4, 4, 2, 2, 3, 6, 2, 6, 5, 5, 4, 2, 1, 2, 1, 4, 5, 1, 4, 4, 5, 1, 1, 1, 4,
6, 6, 5, 5, 5, 6, 3, 6, 2, 2, 2, 3, 6, 4, 6, 3, 3, 2, 6, 4, 5, 1, 2, 2, 6, 2, 4, 4, 4,
2, 6, 4, 2, 4, 1, 1, 2, 1, 3, 1, 1, 2, 2, 1, 2, 2, 4, 1, 5, 4, 1, 2, 2, 6, 3, 3, 3, 3,
1, 4, 5, 4, 4, 6, 6, 4, 6, 3, 2, 6, 3, 6, 3, 6, 5, 1, 1, 3, 1, 5, 3, 5, 3, 1, 1, 4, 4,
1, 1, 1, 5, 6, 5, 5, 4, 6, 5, 6, 6, 6, 2, 1, 1, 1, 1, 5, 6, 2, 3, 3, 2, 1, 3, 1, 5, 3,
6, 4, 1, 3, 5, 5, 6, 5, 1, 3, 5, 3, 1, 2, 3, 2, 6, 6, 6, 5, 3, 1, 5, 3, 1, 4, 2, 3, 5,
4, 4, 5, 6, 5, 5, 5, 6, 6, 6, 4, 2, 1, 3, 3, 1, 3, 3, 6, 5, 1, 2, 1, 1, 5, 1, 4, 5, 6,
5, 5, 5, 1, 2, 6, 6, 2, 2, 4, 2, 6, 2, 3, 1, 5, 1, 1, 1, 5, 5, 5, 6, 5, 6, 5, 3, 5, 1,
1, 1, 2, 2, 1, 3, 3, 3, 6, 4, 1, 2, 6, 2, 6, 3, 1, 1, 2, 3, 1, 2, 2, 4, 4, 4, 2, 1, 3,
3, 1, 4, 5, 5, 3, 2, 3, 3, 1, 2, 4, 5, 4, 1, 3, 5, 6, 3, 6, 5, 3, 3, 6, 2, 1, 2, 2, 2,
2, 6, 3, 6, 6, 5, 4, 2, 2, 2, 3, 1, 4, 5, 6, 5, 6, 4, 5, 4, 1, 2, 4, 4, 2, 3, 6, 6, 4,
6, 3, 1, 5, 5, 4, 6, 3, 3, 3, 1, 2, 3, 5, 3, 3, 1, 1, 5, 1, 4, 2, 3, 6, 5, 5, 6, 6, 6,
2, 3, 1, 5, 4, 5, 3, 1, 2, 2, 6, 3, 2, 2, 4, 2, 3, 6, 4, 6, 5, 6, 4, 4, 1, 2, 3, 5, 1,
3, 5, 1, 5, 6, 5, 3, 5, 3, 2, 6, 2, 1, 5, 5, 1, 5, 3, 1, 5, 6, 3, 1, 5, 1, 4, 5, 3, 3,
2, 1, 5, 5, 4, 6, 5, 1, 2, 2, 6, 2, 4, 5, 1, 5, 3, 1, 2, 4, 5, 4, 6, 4, 6, 3, 3, 3, 6,
5, 3, 6, 6, 6, 4, 1, 2, 1, 2, 2, 1, 1, 2, 3, 3, 1, 5, 5, 6, 3, 6, 5, 1, 3, 3, 6, 2, 2,
4, 1, 1, 1, 4, 6, 5, 3, 1, 4, 6, 6, 6, 4, 5, 4, 1, 4, 1, 2, 6, 5, 4, 5, 6, 4, 6, 3, 2,
4, 5, 6, 5, 3, 5, 3, 3, 5, 4, 4, 4, 1, 3, 2, 3, 3, 6, 4, 6, 4, 2, 1, 4, 2, 3, 1, 1, 3,
2, 3, 6, 2, 4, 6, 3, 2, 3, 6, 4, 5, 3, 1, 4, 1, 4, 6, 5, 6, 4, 2, 2, 3, 5, 1, 4, 5, 4,
6, 3, 2, 3, 6, 5, 5, 3, 3, 6, 5, 1, 3, 1, 2, 2, 3, 3, 5, 5, 6, 6, 3, 5, 3, 5, 4, 5, 1,
3, 3, 1, 2, 1, 5, 4, 4, 4, 6, 5, 5, 4, 5, 6, 3, 2, 6, 5, 5, 5, 5, 4, 6, 2, 6, 3, 3, 2,
3, 2, 2, 4, 6, 5, 3, 1, 4, 1, 3, 2, 4, 5, 5, 1, 5, 3, 6, 4, 6, 3, 5, 1, 2, 3, 2, 2, 3,
5, 3, 2, 4, 5, 4, 1, 1, 1, 4, 1, 1, 3, 1, 4, 6, 6, 5, 6, 6, 4, 1, 3, 3, 5, 4, 1, 3, 5,
5, 3, 5, 5, 3, 2, 1, 2, 1, 4, 4, 4, 2, 4, 6, 3, 5, 4, 6, 3, 3, 5, 6, 3, 1, 3, 3, 1, 5,
6, 5, 4, 2, 4, 5, 4, 2, 6, 4, 1, 4, 1, 4, 6, 6, 4, 5, 3, 6, 5, 6, 6, 5, 3, 1, 1, 4, 2,
3, 5, 1, 5, 6, 5, 6, 4, 4, 6, 2, 4, 5, 3, 3, 5, 4, 4, 5, 5, 1, 4, 5, 1, 3, 3, 6, 6, 5,
3, 2, 2, 2, 4, 4, 2, 4, 2, 6, 6, 5, 4, 5, 6, 4, 4, 2, 1, 3, 6, 6, 5, 5, 5, 6, 5, 3, 1,
4, 1, 2, 6, 6, 5, 4, 6, 6, 3, 3, 1, 3, 5, 1, 5, 3, 6, 6, 6, 4, 6, 6, 5, 4, 6, 2, 4, 5,
4, 4, 5, 4, 5, 3, 5, 4, 5, 4, 6, 3, 5, 4, 1, 3, 3, 5, 4, 4, 5, 6, 4, 5, 6, 4, 4, 4, 2,

1, 1, 5, 6, 3, 5, 4, 1, 2, 2, 6, 2, 6, 4, 6, 3, 5, 6, 4, 5, 1, 4, 2, 6, 3, 5, 4, 4, 1,
5, 3, 3, 1, 1, 4, 1, 2, 2, 4, 6, 3, 2, 3, 3, 5, 5, 3, 1, 4, 4, 2, 2, 3, 6, 2, 6, 5, 5,
4, 2, 1, 2, 1, 4, 5, 1, 4, 4, 5, 1, 1, 1, 4, 6, 6, 5, 5, 5, 6, 3, 6, 2, 2, 2, 3, 6, 4,
6, 3, 3, 2, 6, 4, 5, 1, 2, 2, 6, 2, 4, 4, 4, 2, 6, 4, 2, 4, 1, 1, 2, 1, 3, 1, 1, 2, 2,
1, 2, 2, 4, 1, 5, 4, 1, 2, 2, 6, 3, 3, 3, 3, 1, 4, 5, 4, 4, 6, 6, 4, 6, 3, 2, 6, 3, 6,
3, 6, 5, 1, 1, 3, 1, 5, 3, 5, 3, 1, 1, 4, 4, 1, 1, 1, 5, 6, 5, 5, 4, 6, 5, 6, 6, 6, 2

Akhirnya didapat pemfaktoran matriks identitas dengan melakukan pemetaan ϕ (Persamaan 3.2) terhadap hasil di atas:

$$I = S_{\sigma_0} \dots S_{\sigma_{1769}}.$$

4.3 Modifikasi Fungsi *Hash*

Serangan yang dilakukan oleh Tillich dan Zemor (2008) dapat dihindari dengan melakukan modifikasi pada himpunan generator $\mathcal{S}$. Untuk setiap $S_i \in \mathcal{S}, i = 1, \dots, \ell+1$ ganti dengan S_i^2. Modifikasi ini diusulkan oleh Tillich dan Zemor di akhir makalah. Hal ini mengakibatkan faktorisasi yang dijamin pada Teorema 4.1 tidak dapat dilakukan. Hal ini dapat dijelaskan sebagai berikut:

Karena setiap $M \in \Omega$ pada Teorema 4.1 merupakan perkalian dari S_i sebelum dimodifikasi, sedangkan S_i yang digunakan sekarang adalah S_i^2 dan berdasarkan Definisi 2.46, elemen kata tereduksi S_p tidak memuat bentuk $q_i\overline{q_i}, \overline{q_i}q_i$, dan r_j^2 sedangkan dengan mengganti S_i^2 maka kata tereduksi S_p memuat bentuk r_j^2 sehingga Teorema 2.47 tidak dapat diterapkan dalam faktorisasi M. Ini berakibat serangan yang diusulkan oleh Tillich dan Zemor (2008) tidak dapat dilakukan, dengan kata lain permasalahan untuk menentukan tumbukan pada fungsi *hash* hasil modifikasi tidak dapat diselesaikan.

Algoritma modifikasi untuk menghitung nilai *hash* dapat dilihat pada Algoritma 4.1.

Algoritma 4.1. Konstruksi Fungsi *Hash* Modifikasi

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. Ubah $S_i \in \mathcal{S}$ menjadi S_i^2 untuk $i = 1,\ldots,\ell+1$. Misal $k = \ell+1$ dan $k' = k-1$. Busur awal adalah busur $\left(g_0 S_{-1}^{-1}, g_0\right)$ dengan $g_0 \in PSL_2(\mathbb{F}_p)$ dan $S_{-1} \in \mathcal{S}$. Fungsi urutan ketetanggaan didefinisikan sebagai $\theta : \mathcal{S} \times \{0,\ldots,k'-1\} \to \mathcal{S}$ yang memetakan sebuah elemen di $\mathcal{S}$ dan bilangan digit-k ke elemen $\mathcal{S}$, sehingga untuk setiap $S \in \mathcal{S}$,

$$\left\{\theta(S,i) \,|\, 0 \le i \le k'-1\right\} \cup \left\{S^{-1}\right\} = \mathcal{S}.$$

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 $H(m) = g_0 \prod_{i=0}^{\mu-1} S_i = g_0 S_0 S_1 \ldots S_{\mu-1}$;

4. **Penyelesaian.** Nilai *hash* dari pesan m adalah $H(m)$.

Contoh 4.4 (Serangan pada Modifikasi Fungsi *Hash*).

Fungsi *hash* yang akan diserang menggunakan parameter pada Subbab 4.2, perbedaannya adalah mengganti setiap S_i dengan S_i^2 atau $\widetilde{S}_i$ dengan $\widetilde{S}_i^2$. Pada contoh ini akan ditunjukkan faktorisasi terhadap matriks $M \subset \Omega$ tidak menghasilkan pemfaktoran yang diinginkan, yaitu hasil kali generator pada faktorisasi tidak sama dengan -I atau I, dengan I matriks identitas di $SL(\mathbb{Z}[i])$. Generator awal adalah:

$$\widetilde{S}_1 = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix} \qquad \widetilde{S}_2 = \begin{pmatrix} 1+2i & 0 \\ 0 & 1-2i \end{pmatrix} \quad \widetilde{S}_3 = \begin{pmatrix} 1 & 2i \\ 2i & 1 \end{pmatrix}$$

$$\widetilde{S}_4 = \begin{pmatrix} 1 & -2i \\ -2i & 1 \end{pmatrix} \qquad \widetilde{S}_5 = \begin{pmatrix} 1-2i & 0 \\ 0 & 1+2i \end{pmatrix} \quad \widetilde{S}_6 = \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix}$$

Setelah dikuadratkan menjadi:

$$\widetilde{S}_1 = \begin{pmatrix} -3 & 4 \\ -4 & -3 \end{pmatrix} \qquad \widetilde{S}_2 = \begin{pmatrix} -3+4i & 0 \\ 0 & -3-4i \end{pmatrix} \quad \widetilde{S}_3 = \begin{pmatrix} -3 & 4i \\ 4i & -3 \end{pmatrix}$$

$$\widetilde{S}_4 = \begin{pmatrix} -3 & -4i \\ -4i & -3 \end{pmatrix} \qquad \widetilde{S}_5 = \begin{pmatrix} -3-4i & 0 \\ 0 & -3+4i \end{pmatrix} \quad \widetilde{S}_6 = \begin{pmatrix} -3 & -4 \\ 4 & -3 \end{pmatrix}$$

Tahapan yang dilakukan sama seperti pada Subbab 4.2. Matriks $M = \begin{pmatrix} a+bi & c+di \\ -c+di & a-bi \end{pmatrix}$ adalah matriks yang dihasilkan pada Tahap Kedua Subbab 4.2, dengan

$$\begin{aligned} a &= \sqrt{5^{2k} - 16up^2} \\ b &= 2px \\ c &= 2py \\ d &= 2pz \end{aligned}$$

Selanjutnya memfaktorkan M menjadi perkalian dari $\widetilde{S}_i$, dengan $i = 1, \ldots, 6$. Faktorisasi dilakukan dengan cara yang sama seperti pada Subbab 4.2. Setelah 1770 iterasi ternyata matriks yang dihasilkan bukan merupakan matriks identitas I atau -I, melainkan matriks $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ dengan

a_{11} = 381199139148866535831408268136972943499698237487972650285042284989269090258661385676558015492426266562658748526586919765480101665404846636327502975980371724314572763421228874064415392270069958940869640689998785986845879952499393282148230922390369691900356194064923759809752602808865927758588427412207145404868636679835592609540928146845991118499633957632261185364861300863401185310621386169065725459684543686546468451071499153937549683203139254870355039598682902606566283381831390537952610880977031661252844935170443157111322559912085985697726333173987842877263245502747898976723832686497292745727161315044655679906264 3+677842173999280550635035550366853879638605956541712196805918103392657247229869968646258711254550657807289331197972107444548401558286367293408081317464242693216547075754756307172060510108386762278065324041598241123377764082641312336420536160520848991976063222303397772530646978112035651451331188119853873371243078694889050391033929231376566937929273611970068001621314493610642265860053094187170042721349696858646569461233629529304977172440748678082763924322227789996607424283257958968326257817966053306746728679376941158564327943746717718081506804811551918135303482019764385857243254768433128049786928663042917683071160i

a_{12} = 193587547232975843939757501504284167542364043337827339821
87474902905985267068906232680652016703894658560826177971 6
55223525466843853630104207968047157529718553199318881417 55
18615417887213607542603309419662073704877486685998982233 54
62308452642160432483063376103589762402069278189490411227 82
91039548473301088527942239952286502021251850452419048263 75
54286709614198548914007185517894855218390368824011434390 21
56468239210094854358283617091423642491857411527992766341 07
95295097699379380683638681706623978487469780606425330803 4
64517024988992359538792244883290019954530836016973483236 2
22051379637173014738403889123850682257977 6+19544519762 22
30985488330060342582769038946457150607570587673184234686 9
65135740647117880656544921473162161787810196178858446636 23
66550337519742237265912362680235092537315137555923957731 50
47475352666939178550681827188083742184272564473843833377 9
98737224399467521628222344647064578431514756661699940156 74
564842784888

a_{21} = -193587547232975843939757501504284167542364043337827339821
87474902905985267068906232680652016703894658560826177971 6
55223525466843853630104207968047157529718553199318881417 55
18615417887213607542603309419662073704877486685998982233 54
62308452642160432483063376103589762402069278189490411227 82
91039548473301088527942239952286502021251850452419048263 75
54286709614198548914007185517894855218390368824011434390 21
56468239210094854358283617091423642491857411527992766341 07
95295097699379380683638681706623978487469780606425330803 4
64517024988992359538792244883290019954530836016973483236 2
22051379637173014738403889123850682257977 6+19544519762 22
30985488330060342582769038946457150607570587673184234686 9
65135740647117880656544921473162161787810196178858446636 23
66550337519742237265912362680235092537315137555923957731 50
47475352666939178550681827188083742184272564473843833377 9
98737224399467521628222344647064578431514756661699940156 74
564842784888

$$
\begin{aligned}
a_{22} \quad = \quad & 381199139148866535831408268136972943499698237487972650285 0 \\
& 4228498926909025866138567655801549242626656265874852658 69 \\
& 197654801016654048466363275029759803717243145727634212288 7 \\
& 40644153922700699589408696406899987859868458799524993932 8 \\
& 21482309223903696919003561940649237598097526028088659277 5 \\
& 858842741220714540486863667983559260954092814684599111849 9 \\
& 633957632261185364861300863401185310621386169065725459684 5 \\
& 43686546468451071499153937549683203139254870355039598682 9 \\
& 026065662833818313905379526108809770316612528449351704431 5 \\
& 711132255991208598569772633317398784287726324550274789897 6 \\
& 7238326864972927457271613150446556799062643-6778421739992 \\
& 805506350355503668538796386059565417121968059181033926572 4 \\
& 722986996864625871125455065780728933119797210744454840155 8 \\
& 286367293408081317464242693216547075754756307172060510108 3 \\
& 867622780653240415982411233777640826413123336420536160520 8 \\
& 489919760632223033977725306469781120356514513311881198538 7 \\
& 337124307869488905039103392923137656693792927361197006800 1 \\
& 621314493610642265860053094187170042721349696858646569461 2 \\
& 33629529304977172440748678082763924322227789996607424283 2 \\
& 57958968326257817966053306746728679376941158564327943746 7 \\
& 177180815068048115519181353034820197643858572432547684331 2 \\
& 8049786928663042917683071 16i.
\end{aligned}
$$

Sedangkan nilai $\sigma_j, j = 0, \ldots 1769$ adalah:

5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,

5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,

5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,

BAB 5
PENUTUP

Pada bab ini diberikan kesimpulan dan saran yang berkaitan dengan penelitian yang dilakukan. Saran pada bab ini diharapkan dapat dijadikan acuan untuk melakukan pengembangan lebih lanjut.

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Fungsi *hash* dari graf ekspander LPS mempunyai ekspansi yang besar, diameter yang kecil, *girth* yang besar dan efisien dalam menghitung tetangga jika diberikan sebarang simpul sehingga memenuhi persyaratan menjadi fungsi *hash* dari graf ekspander.

2. Fungsi *hash* dari graf ekspander LPS tidak memenuhi sifat *collision resistance* karena tumbukan (*collision*) dapat ditemukan secara efisien. Ini dilakukan dengan melakukan pemfaktoran dalam himpunan matriks yang entri-entrinya di $\mathbb{Z}[i]$ dari pada memfaktorkan di $PSL_2(\mathbb{F}_p)$.

3. Salah satu modifikasi yang dapat dilakukan adalah dengan mengganti himpunan matriks generator $S_i, i = 1, \ldots, \ell + 1$ menjadi S_i^2 yang mengakibatkan faktorisasi unik di himpunan $\Omega \subset SL_2(\mathbb{Z}[i])$ sulit dilakukan.

5.2 Saran

1. Perlu dilakukan kajian mengenai sifat fungsi *hash* yang lain yaitu *second-preimage resistance* dan *preimage resistance*.

2. Perlu dilakukan kajian graf LPS hasil modifikasi apakah masih memenuhi sifat-sifat graf LPS awal. Ini untuk mengetahui seberapa signifikan ekspansi yang besar diperlukan dalam desain fungsi *hash* dari graf ekspander.

3. Perlu dilakukan penelitian mengenai aspek komputasi fungsi *hash* dari graf ekspander LPS khususnya, dan dari graf ekspander Cayley pada umumnya. Karena berdasarkan Petit (2009), perhitungan fungsi *hash* dari graf ekspander Cayley dapat dilakukan secara paralel (*parallel computing*).

DAFTAR REFERENSI

Burton, D. (2007). *Elementary Number Theory,* edisi ke-6. McGraw-Hill Higher Education.

Charles, D., Goren, E., dan Lauter, K. (2007). Cryptographic hash functions from expander graph. *Journal of Cryptology*, volume 22, halaman 93–113.

Collins, D. C. (1999). Continued Fractions. http://www-math.mit.edu/phase2/UJM/vol1/COLLIN~1.PDF. diakses 22 Maret 2011, 12.19 WIB.

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

Davidoff, G., Sarnak, P., dan Valette, A. (2003). *Elementary Number Theory, Group Theory, and Ramanujan Graphs*. London Mathematical Society Student Texts, nomor 55. Cambridge University Press.

Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., dan Walker, J. (2010). The Skein Hash Function Family. http://www.skein-hash.info/sites/default/files/skein1.3.pdf, Submission to NIST (Round 3).

Fraleigh, J. B. (2002). *A First Course in Abstract Algebra*. Addidon-Wesley.

Goldreich, O. Basic facts about expander graph. www.wisdom.weizmann.ac.il/~oded/COL/expander.pdf. diakses pada 9 November 2010, 7:45:30 WIB.

Goldreich, O. Randomized Methods in Computation. Lecture Notes. http://www.wisdom.weizmann.ac.il/~oded/rnd-sum.html. diakses 5 Januari 2011, 08.24 WIB.

Lubotzky, A., Phillips, R., dan Sarnak, P. (1988). Ramanujan graphs. *Combinatorica*, 8:261–277. 10.1007/BF02126799.

Menezes, A., Ooschot, P. V., dan Vanstone, S. (1996). *Handbook of Applied Cryptography*. CRC Press, Boca Raton.

Merkle, R. (1979). *Secrecy, Authentication, and Public key systems*. PhD thesis, Stanford.

NIST (2007). Notices. Federal Register/Vol.72 No. 212.

Petit, C. (2009). *On graph-based cryptographic hash functions*. PhD thesis, Universite Catholique de Louvain, Louvain-la-Neuve.

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

Tillich, J. dan Zemor, G. Z. (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, halaman 254–269, Berlin/Heidelberg. Springer-Verlag.

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