ARITMETIK RING POLINOMIAL UNTUK KONSTRUKSI FUNGSI HASH BERBASIS LATIS IDEAL

S. Guritman [1], N. Aliatiningtyas [2], T. Wulandari [3], dan M. Ilyas [4]

Abstrak

Sebagai hasil awal dari penelitian "konstruksi fungsi hash berbasis latis ideal", dalam artikel ini dikaji aspek komputasi ring $\mathbb{Z}_p[x]/\langle f(x)\rangle$. Diawali dari fakta bahwa ring polinomial $\mathbb{Z}_p[x]$ merupakan daerah Euclides, dapat dikonstruksi algoritme-algoritme keterbagian dalam $\mathbb{Z}_p[x]$. Kemudian, dari fakta $\mathbb{Z}_p[x]$ adalah daerah ideal utama, bisa dikonstruksi algoritme-algoritme operasi jumlah dan kali modulo $f(x)$ dalam ring $\mathbb{Z}_p[x]/\langle f(x)\rangle$. Ketika $f(x)$ berderajat n, bisa ditunjukkan pula bahwa $\mathbb{Z}_p[x]/\langle f(x)\rangle$ merupakan ruang vektor atas $\mathbb{Z}_p$ dalam operasi jumlah modulo $f(x)$ dengan basis baku $\{1, x, x^2, ..., x^{n-1}\}$, dan isomorfik ke $\mathbb{Z}_p^n$. Dari fakta yang terakhir ini, semua algoritme yang dikontruksi dapat direpresentasikan dalam data vektor. Terkait dengan kegunaan aritmetik tersebut untuk konstruksi fungsi hash, $f(x)$ dibatasi hanya polinomial yang monik, berderajat n, tak teruraikan atas $\mathbb{Z}$, dan untuk setiap vektor satuan $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_p[x]/\langle f(x)\rangle$, hasil kali ring dari $\mathbf{u}$ dan $\mathbf{v}$ merupakan vektor pendek, artinya $\|\mathbf{uv}\|$ umumnya terbatas ke $\sqrt{n}$.

Kata Kunci: Algoritme Aritmetik Ring Polinomial, Fungsi Hash, Latis Ideal

1 Pendahuluan

Menurut Menezes dkk. [8], kriptografi adalah studi teknik matematik yang berkaitan dengan tujuan keamanan informasi seperti kerahasiaan, integritas data, otentikasi entitas, dan otentikasi asal data. Salah satu primitif kriptografi yang terkait dengan tujuan keamanan integritas data dan otentikasi adalah *fungsi hash*. Fungsi hash adalah fungsi $h : D \to R$ dengan $|D| \gg |R|$ serta memunyai *sifat keamanan*: satu arah (*one-way*) dan tahan tumbukan (*collision resistance*). Fungsi h bersifat satu arah jika secara komputasi taklayak menghitung $x \in D$ dari diketahuinya $y \in R$ sehingga $y = h(x)$, dan h

[1]• S. Guritman berafiliasi pada Departemen Matematika, Institut Pertanian Bogor, Bogor

[2]• N. Aliatiningtyas berafiliasi pada Departemen Matematika, Institut Pertanian Bogor, Bogor

[3]• T. Wulandari berafiliasi pada Departemen Matematika, Institut Pertanian Bogor, Bogor

[4]• M. Ilyas berafiliasi pada Departemen Matematika, Institut Pertanian Bogor, Bogor

tahan tumbukan jika secara komputasi tak-layak menentukan $x, y \in D$ dengan $x \neq y$ sehingga $h(x) = h(y)$.

Maraknya serangan terhadap sifat keamanan fungsi hash yang konstruksinya berbasis pada aritmetik *boolean* dewasa ini, menggiatkan para peneliti kriptografi untuk beralih ke konstruksi fungsi hash yang keamanannya berlandaskan pada *problem komputasi latis*. Latis Λ berdimensi-n adalah himpunan semua kombinasi linear intejer dari n vektor bebas linear dengan problem: *secara komputasi tak-layak menentukan vektor terpendek di dalam* Λ.

Diawali oleh Ajtai [1] yang mendefinisikan keluarga fungsi hash satu arah dengan keamanan bertumpu problem komputasi latis. Dalam hal ini, untuk parameter intejer positif: d, m, n, dengan $m > n$, dan p bilangan prima, dipilih matriks $\mathbf{A} \in \mathbb{Z}_p^{n \times m}$ secara acak, fungsi hash $h_{\mathbf{A}}$ dengan *kunci* $\mathbf{A}$ didefinisikan sebagai

$$h_{\mathbf{A}} : \mathbb{Z}_d^m \rightarrow \mathbb{Z}_p^n \text{ dengan } \mathbf{y} = h_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x}$$

Memperbaki hasil kerja Ajtai, Goldreich dkk. [5] membuktikan bahwa $h_{\mathbf{A}}$ adalah tahan tumbukan. Kemudian, asumsi keamanan diperkuat di dalam: [3], [9], dan [10] sekaligus menunjukkan bahwa aspek efisiensi komputasi belum cukup jika terkait dengan aplikasi.

Berikutnya, usaha efisiensi tercantum di dalam [11] and [6] yang memperumum struktur $\mathbf{A}$ dari aritmetik *field* $\mathbb{Z}_p$ ke aritmetik ring polinomial modular $\mathbb{Z}_p[x] / \langle f(x) \rangle$ untuk sembarang polinomial $f(x) \in \mathbb{Z}[x]$. Latis dengan basis vektor-vektor kolom dari $\mathbf{A}$ dan didefinisikan dari aritmetik ring polinomial modular disebut *latis ideal*. Ternyata, fungsi hash berbasis latis ideal tidak tahan tumbukan untuk sembarang $f(x)$. Akhirnya, dibuktikan di dalam artikel [6] bahwa fungsi hash berbasiskan latis ideal dijamin tahan tumbukan jika $f(x)$ yang dipilih memenuhi dua sifat, yaitu *tak-teruraikan (irreducible) atas* $\mathbb{Z}$ dan *untuk setiap vektor satuan* $\mathbf{u}$ *dan* $\mathbf{v}$, *hasil kali ring dari* $\mathbf{u}$ *dan* $\mathbf{v}$ *adalah vektor pendek* (terbatas $\sqrt{n}$).

Di dalam artikel ini dikonstruksi algoritme-algoritme tentang aritmetik ring polinomial yang nantinya akan digunakan untuk membangun algoritme fungsi hash berbasis latis ideal. Pembahasan terbagi atas tiga seksi. Seksi 2 berisi tinjauan aljabar dari struktur ring polinomial. Seksi 3 memuat tinjauan aljabar tentang aritmetik ring polinomial modular. Terakhir, Seksi 4 merupakan bahasan inti dari topik dan tujuan penelitian, yaitu aspek komputasi dari aritmetik ring polinomial modular $\mathbb{Z}_p[x] / \langle f(x) \rangle$ yang memenuhi dua sifat sebagaimana disebutkan dalam paragraf di atas.

2 Struktur Ring Polinomial

Diberikan suatu *field* $\mathbb{F}$. suatu ekspresi $a(x)$ berbentuk

$$a(x) = \sum_{i=0} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n + \cdots$$

disebut *polinomial dalam x atas $\mathbb{F}$* jika $\forall i = 0, 1, 2, .., a_i \in \mathbb{F}$. Dalam hal ini, a_i disebut *koefisien* ke-i dari *suku* (term) ke-i $(a_i x^i)$, dan x hanyalah sekedar simbol sebagai *placeholder* atau *indeterminate*. Polinomial $a(x)$ dikatakan ber*derajat* n, notasi $\deg(a(x)) = n$, jika $a_n \neq 0$ dan $a_i = 0$ untuk setiap $i > n$. Dalam hal demikian, a_n disebut *koefisien pemimpin* (leading) dan sukunya $a_n x^n$ juga disebut *suku pemimpin*. Polinomial dengan koefisien pemimpin 1 disebut *monik*. *Polinomial nol* adalah polinomial yang semua sukunya nol, notasi $O(x)$ atau cukup ditulis 0 jika konteksnya jelas, dan didefinisikan $\deg(O(x)) = -\infty$. $p(x)$ disebut *Polinomial konstan* jika $\deg(p(x)) = 0$, berarti ada $k \in \mathbb{F}$ dengan $k \neq 0$ sehingga $p(x) = k$.

Didefinisikan $\mathbb{F}[x] = \{\sum_{i=0} a_i x^i / a_i \in \mathbb{F}\}$ sebagai himpunan semua polinomial atas $\mathbb{F}$. Misalkan $a(x) = \sum_{i=0} a_i x^i$ dan $b(x) = \sum_{i=0} b_i x^i$ adalah sembarang dua anggota $\mathbb{F}[x]$, didefinisikan *persamaan:* $a(x) = b(x) \Leftrightarrow a_i = b_i$, *operasi jumlah:* $a(x) + b(x) := \sum_{i=0} (a_i + b_i) x^i$, dan *operasi kali:* $a(x).b(x) = \sum_{i+j=k} a_i b_j$.

Proposisi 1. *(Bukti bisa mengacu [12], Hal. 244) Terhadap definisi operasi jumlah dan kali di atas, $\mathbb{F}[x]$ memiliki struktur* ***daerah Integral****.*

Proposisi di atas menegaskan bahwa $\mathbb{F}[x]$ mempunyai struktur operasi yang analog dengan struktur operasi pada himpunan bilangan bulat $\mathbb{Z}$. Demikian pula dengan sifat-sifat berikut ini. (Bukti bisa mengacu [4], Hal. 13)

Proposisi 2. ***(Algoritme Pembagian)*** *Untuk $a(x), b(x) \in \mathbb{F}[x]$ dan $b(x) \neq 0$, selalu bisa ditemukan sepasang polinomial $q(x), r(x) \in \mathbb{F}[x]$ sehingga berlaku $a(x) = q(x).b(x) + r(x)$ dimana $\deg(r(x)) < \deg(b(x))$. Akibatnya, $\mathbb{F}[x]$ memiliki struktur* ***daerah Euclides****.*

Dari proposisi di atas, $q(x)$ disebut *hasil* (bagi) dan $r(x)$ disebut *sisa* (bagi) dari pembagian $a(x)$ oleh $b(x)$. Kemudian, dikatakan $b(x)$ *membagi* $a(x)$, notasi $b(x) | a(x)$, jika ada $s(x) \in \mathbb{F}[x]$ sehingga $a(x) = b(x).s(x)$. Dalam hal ini, $b(x)$ disebut *faktor* dari $a(x)$, atau $a(x)$ disebut *kelipatan* dari $b(x)$. Selanjutnya, polinomial tak-nol $c(x)$ disebut *pembagi bersama* dari $a(x)$ dan $b(x)$ jika $c(x) | a(x)$ dan $c(x) | b(x)$. Dalam hal $a(x)$ dan $b(x)$ tidak keduanya nol. $d(x)$ disebut *pembagi bersama terbesar*, notasi $d(x) = \gcd(a(x), b(x))$, jika $c(x) | d(x)$ untuk setiap pembagi bersama $c(x)$. Akhirnya, $a(x)$ dan $b(x)$ disebut (saling) *koprima* jika $\gcd(a(x).b(x)) = 1$. Sifat penting yang terkait dengan pengertian gcd polinomial diberikan dalam proposisi berikut. (Bukti bisa mengacu [12], Hal. 253)

Proposisi 3. *Jika $d(x) = \gcd(a(x).b(x))$, maka ada $r(x).s(x) \in \mathbb{F}[x]$ sehingga*

$$d(x) = r(x)\,a(x) + s(x)\,b(x)$$

Polinomial $a(x) \in \mathbb{F}[x]$ dikatakan *teruraikan* (reducible) *atas* $\mathbb{F}$ jika ada $s(x), r(x) \in \mathbb{F}[x]$ keduanya berderajat positif sehingga $a(x) = s(x)\,r(x)$. Polinomial *tak-teruraikan* (irreducible) *atas* $\mathbb{F}$ merupakan ingkaran dari polinomial teruraikan atas $\mathbb{F}$.

3 Ring Polinomial Modular

Misalkan J adalah suatu ideal dalam $\mathbb{F}[x]$, berarti untuk setiap $f(x) \in J$ dan $k(x) \in \mathbb{F}[x]$ berlaku $k(x)\,f(x) \in J$. Lagi, proposisi berikut ini menegaskan bahwa struktur $\mathbb{F}[x]$ beranalog dengan $\mathbb{Z}$. (Bukti bisa mengacu [4], Hal. 14)

Proposisi 4. $\mathbb{F}[x]$ *memiliki struktur* ***daerah ideal utama****. Artinya, untuk setiap ideal* $J \neq \{0\}$ *dalam* $\mathbb{F}[x]$, *ada tepat satu polinomial* ***monik*** $f(x)$ *dan* ***berderajat terkecil*** *dalam* J *sehingga* $J = \langle f(x)\rangle = \{k(x)\,f(x) \diagup k(x) \in \mathbb{F}[x]\}$, *terkadang dinotasikan* $J = f(x)\,\mathbb{F}[x]$.

Berdasarkan proposisi di atas, untuk selanjutnya dalam artikel ini, setiap kita sebut ideal $J = \langle f(x)\rangle$ dalam $\mathbb{F}[x]$ dimaksudkan $f(x)$ monik dan berderajat terkecil dalam J. Kemudian, untuk suatu $a(x) \in \mathbb{F}[x]$, himpunan

$$J + a(x) := \{j(x) + a(x) \diagup j(x) \in J\} = \{k(x)\,f(x) + a(x) \diagup k(x) \in \mathbb{F}[x]\}$$

disebut *koset dari* J *yang memuat* $a(x)$ dan jelas merupakan subhimpunan dari $\mathbb{F}[x]$. *Keluarga koset* dari J dinotasikan

$$\mathbb{F}[x] / \langle f(x)\rangle := \{J + a(x), J + b(x), J + c(x), ...\}$$

Ketika pada $\mathbb{F}[x] / \langle f(x)\rangle$ didefinisikan operasi *jumlah koset* dan *kali koset:*

$$(J + a(x)) + (J + b(x)) = J + (a(x) + b(x))$$
$$(J + a(x)).(J + b(x)) = J + (a(x).b(x))$$

maka kita peroleh proposisi berikut ini (Bukti bisa mengacu [12], Hal. 192).

Proposisi 5. $\mathbb{F}[x] / \langle f(x)\rangle$ *merupakan ring dan disebut* ***ring kuosen*** *(quotient ring).*

Dengan memandang bahwa $\mathbb{F}[x]$ adalah grup terhadap operasi jumlah dan J adalah subgrup normal, maka $\mathbb{F}[x] / \langle f(x)\rangle$ merupakan *partisi* dari $\mathbb{F}[x]$. Kemudian, sifat koset yang paling penting dan berlaku untuk semua ring dinyatakan dalam proposisi berikut ini (Bukti bisa mengacu [12], Hal. 193).

Proposisi 6. [5] **(Teorema Dasar Homomorfisme) Diberikan ring** R **dan** S, **jika** $\varphi : R \to S$ **adalah epimorfisme ring, maka** $R/\ker(\varphi)$ **isomorfik dengan** S **(notasi:** $R/\ker(\varphi) \cong S$**) dengan pemadanan isomorfisme**

$$[\ker(\varphi) + r] \in R/\ker(\varphi) \rightarrowtail \varphi(r) \in S$$

[5] $\varphi : R \to S$ adalah homomorfisme ring jika $(\forall x, y \in R)\ \varphi(x + y) = \varphi(x) + \varphi(y)$ dan $\varphi(xy) = \varphi(x)\varphi(y)$. Jika φ surjektif disebut epimorfisme, dan jika φ bijektif disebut isomorfisme. $\ker(\varphi) = \{x \in R \diagup \varphi(x) = 0\}$ dan $\mathrm{Im}(\varphi) = \{\varphi(x) \in S \diagup \forall x \in R\}$

Sekarang, untuk intejer positif n, ambil polinomial monik $f(x) \in \mathbb{F}[x]$ dengan $\deg(f(x)) = n$. Kita definisikan pemetaan $\Phi_{f(x)} : \mathbb{F}[x] \to \mathbb{F}[x]$ dengan rumus untuk setiap $a(x) \in \mathbb{F}[x]$,

$$\Phi_{f(x)}(a(x)) = a(x) \bmod f(x) = r(x)$$

dengan $r(x)$ dihitung sebagai sisa dari $a(x)$ dibagi $f(x)$. Maka, kita peroleh lemma berikut.

Lemma 1. *$\Phi_{f(x)}$ adalah homomorfisme ring.*

Bukti. Ambil $a_1(x), a_2(x) \in \mathbb{F}[x]$, berdasarkan Proposisi 2, berarti ada $q_1(x), q_1(x) \in \mathbb{F}[x]$ sehingga

$$a_1(x) = q_1(x) f(x) + r_1(x) \Leftrightarrow a_1(x) \bmod f(x) = r_1(x) \text{ dan}$$
$$a_2(x) = q_2(x) f(x) + r_2(x) \Leftrightarrow a_2(x) \bmod f(x) = r_2(x)$$

akibatnya ada $h(x) = (q_1(x) + q_2(x)) \in \mathbb{F}[x]$ sehingga

$$a_1(x) + a_2(x) = h(x) f(x) + (r_1(x) + r_2(x)) \Leftrightarrow$$

$$\begin{aligned}(a_1(x) + a_2(x)) \bmod f(x) &= (r_1(x) + r_2(x)) \\ &= a_1(x) \bmod f(x) + a_2(x) \bmod f(x) \Leftrightarrow\end{aligned}$$

$$\Phi_{f(x)}((a_1(x) + a_2(x))) = \Phi_{f(x)}(a_1(x)) + \Phi_{f(x)}(a_2(x))$$

dan ada $g(x) = (q_1(x) q_2(x) f(x) + r_1(x) q_2(x) + r_2(x) q_1(x)) \in \mathbb{F}[x]$ sehingga

$$a_1(x) . a_2(x) = g(x) f(x) + r_1(x) . r_2(x) \Leftrightarrow$$

$$\begin{aligned}(a_1(x) . a_2(x)) \bmod f(x) &= r_1(x) . r_2(x) \\ &= (a_1(x) \bmod f(x)) . (a_2(x) \bmod f(x)) \Leftrightarrow\end{aligned}$$

$$\Phi_{f(x)}((a_1(x) . a_2(x))) = \Phi_{f(x)}(a_1(x)) . \Phi_{f(x)}(a_2(x))$$

□

Teorema 2. *Ring $\mathbb{F}[x] / \langle f(x) \rangle$ dan $\operatorname{Im}(\Phi_{f(x)})$ adalah isomorfik.*

Bukti. Berdasarkan Lemma 1, $\Phi_{f(x)} : \mathbb{F}[x] \to \operatorname{Im}(\Phi_{f(x)})$ adalah epimorfisme ring. Akibatnya, dengan Proposisi 6, kita peroleh $\mathbb{F}[x] / \ker(\Phi_{f(x)}) \cong \operatorname{Im}(\Phi_{f(x)})$. Dalam hal ini, $\ker(\Phi_{f(x)}) = \{a(x) \in \mathbb{F}[x] / \Phi_{f(x)}(a(x)) = 0\} \Leftrightarrow$

$$\begin{aligned}\ker(\Phi_{f(x)}) &= \{a(x) \in \mathbb{F}[x] / a(x) \bmod f(x) = 0\} \\ &= \{k(x) f(x) \in \mathbb{F}[x] / k(x) \in \mathbb{F}[x]\} = \langle f(x) \rangle\end{aligned}$$

□

Sekarang, kita amati wujud keanggotaan $\mathrm{Im}\left(\Phi_{f(x)}\right)$, yaitu

$$\begin{aligned}\mathrm{Im}\left(\Phi_{f(x)}\right) &= \left\{\Phi_{f(x)}\left(a\left(x\right)\right) \in \mathbb{F}\left[x\right] \diagup \forall a\left(x\right) \in \mathbb{F}\left[x\right]\right\} \\ &= \left\{a\left(x\right) \bmod f\left(x\right) \diagup \forall a\left(x\right) \in \mathbb{F}\left[x\right]\right\} \\ &= \left\{r\left(x\right) \diagup r\left(x\right) = a\left(x\right) \bmod f\left(x\right), \quad \forall a\left(x\right) \in \mathbb{F}\left[x\right]\right\}\end{aligned}$$

Karena $\deg\left(f\left(x\right)\right) = n$ dan dari definisi mod, maka bisa kita nyatakan

$$\begin{aligned}\mathrm{Im}\left(\Phi_{f(x)}\right) &= \left\{r\left(x\right) \in \mathbb{F}\left[x\right] \diagup \deg\left(r\left(x\right)\right) \leq n-1\right\} \\ &= \left\{r_0 + r_1 x + r_2 x^2 + \cdots + r_{n-1} x^{n-1} \diagup r_i \in \mathbb{F}, i = 0, 1, ..., n-1\right\}\end{aligned}$$

Dengan demikian, dari Teorema 2, pemadanan isomorfisme $\mathbb{F}\left[x\right] / \left\langle f\left(x\right)\right\rangle$ dan $\mathrm{Im}\left(\Phi_{f(x)}\right)$ bisa dinyatakan sebagai berikut. Jika

$$\begin{aligned}&\left\langle f\left(x\right)\right\rangle + a\left(x\right) \leftrightarrow r\left(x\right) = a\left(x\right) \bmod f\left(x\right) \text{ dan} \\ &\left\langle f\left(x\right)\right\rangle + b\left(x\right) \leftrightarrow s\left(x\right) = b\left(x\right) \bmod f\left(x\right)\end{aligned}$$

maka

$$\begin{aligned}&\left(\left\langle f\left(x\right)\right\rangle + a\left(x\right)\right) + \left(\left\langle f\left(x\right)\right\rangle + b\left(x\right)\right) \leftrightarrow r\left(x\right) + s\left(x\right) \text{ dan} \\ &\left(\left\langle f\left(x\right)\right\rangle + a\left(x\right)\right).\left(\left\langle f\left(x\right)\right\rangle + b\left(x\right)\right) \leftrightarrow \left(r\left(x\right).s\left(x\right)\right) \bmod f\left(x\right)\end{aligned}$$

Akibatnya, kita peroleh proposisi berikut ini.

Proposisi 7. *Aritmetik ring* $\mathbb{F}\left[x\right] / \left\langle f\left(x\right)\right\rangle$ *dapat dibawa (diisomorfismekan) ke aritmetik poninomial modular* R_n, *yaitu*

$$R_n = \left\{r_0 + r_1 x + r_2 x^2 + \cdots + r_{n-1} x^{n-1} \diagup r_i \in \mathbb{F}\right\} \subseteq \mathbb{F}\left[x\right]$$

dengan operasi ***jumlah*** *dan* ***kali polinomial modulo*** $f\left(x\right)$.

Ambil sembarang $k \in \mathbb{F}$, maka k dapat dipandang sebagai anggota R_n berderajat < 1, akibatnya untuk setiap $r\left(x\right) \in R_n$, kita peroleh $k.r\left(x\right) \in R_n$. Perkalian $k.r\left(x\right)$ ini, kemudian kita definisikan sebagai *aturan kali-skalar* di dalam R_n. Akhirnya, mengingat bahwa R_n adalah grup komutatif terhadap operasi jumlah ring, maka kita peroleh teorema berikut ini yang *terlalu tidak sulit* untuk dibuktikan.

Teorema 3. *Terhadap operasi jumlah ring dan aturan kali-skalar,* R_n *merupakan* ***ruang vektor*** *atas* $\mathbb{F}$ *dengan basis baku* $\left\{1, x, x^2, ..., x^{n-1}\right\}$. *Selanjutnya,* R_n *dan* $\mathbb{F}^n$ *adalah dua ruang vektor yang isomorfik dengan pemadanan isomorfisme*

$$r_0 + r_1 x + r_2 x^n + \cdots + r_{n-1} x^{n-1} \in R_n \leftrightarrow \left(r_0, r_1, r_2, \cdots, r_{n-1}\right) \in \mathbb{F}^n$$

4 Aritmetik Ring Polinomial untuk Fungsi Hash Berbasis Latis Ideal

Ambil bilangan prima ganjil p, kita notasikan $\mathbb{Z}_p = \{0, 1, 2, ..., p-1\}$ sebagai *field* intejer bilangan prima (selanjutnya, cukup disebut *field prima*) dengan operasi jumlah dan kali modulo p. Berikutnya, ambil polinomial $f(x) \in \mathbb{Z}_p[x]$. Tujuan umum kita adalah membangun *cara hitung* (aritmetik) yang terkait dengan operasi ring $\mathbb{Z}_p[x] / \langle f(x) \rangle$, seperti: cara menjumlahkan (dan inversnya mengurangkan), cara mengalikan (dan invernya membagikan jika terdefinisikan), cara memangkatkan (dan inversnya melogaritmekan jika terdefinisikan), dan termasuk cara menentukan eksistensi invers. Jika dikaitkan mesin hitung (komputer), cara hitung tersebut berupa algoritme dan diharapkan yang *paling efisien* (berkecepatan hitung tinggi). Sedangkan tujuan khususnya adalah sesuai dengan tujuan penelitian, yaitu membangun algoritme aritmetik yang memenui syarat:

1. $f(x)$ adalah polinomial monik, berderajat n, tak teruraikan atas $\mathbb{Z}$,
2. untuk setiap vektor satuan $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_p[x] / \langle f(x) \rangle$ hasil kali ring dari $\mathbf{u}$ dan $\mathbf{v}$ merupakan vektor pendek, artinya $\|\mathbf{uv}\|$ terbatas $\sqrt{n}$. Catatan untuk syarat ini, nantinya anggota-anggota $\mathbb{Z}_p[x] / \langle f(x) \rangle$ kita akan padang sebagai vektor-vektor dalam $\mathbb{Z}_p^n$.
3. algoritme yang dihasilkan secara signifikan sangat efisien.

Untuk mencapai ketiga syarat tersebut, hal pertama yang akan kita bahas adalah representasi data dan pemilihan fungsi f.

4.1 Representasi Data

Berlandaskan pada Teorema 2 dan 3, demi kemudahan dan efisiensi komputasi, anggota-anggota $\mathbb{Z}_p[x] / \langle f(x) \rangle$ kita padang sebagai vektor-vektor dalam $\mathbb{Z}_p^n$, demikian pula untuk anggota $\mathbb{Z}_p[x]$. Dengan demikian, data *input/output* beserta operasi pemrosesannya merupakan data vektor. Ilustrasi, untuk $p = 5$ dan $\deg(f) = 7$. $a(x) = 2+3x^2+x^4+x^5$ direpresentasikan $(2, 0, 3, 0, 1, 1, 0, 0) \in \mathbb{Z}_5^7$, atau jika dipandang $a(x) \in \mathbb{Z}_5[x]$ kita representasi sebagai $(2, 0, 3, 0, 1, 1)$.

Terkait dengan syarat kedua, kita represetasikan $\mathbb{Z}_p = \left\{0, \pm 1, \pm 2, ..., \pm \frac{p-1}{2}\right\}$. Dalam aritmetik intejer modular, anggota-anggota $\mathbb{Z}_p$ dapat dipandang sebagai keluarga koset-koset. Dengan demikian, untuk $a \in \mathbb{Z}_p$ dengan $\frac{p-1}{2} < a \leq p-1$, dapat dinyatakan $a = -(p-a)$.

Pemilihan atau penetapan fungsi f sangat memengaruhi syarat ketiga. Fungsi hash LASH di dalam artikel [2] menetapkan $f(x) = x^n - 1$, sedangkan fungsi hash SWIFFT di dalam artikel [7] menetapkan $f(x) = x^n + 1$.

Kali ini kita pilih f sebagai keluarga *trinomial* yang kita definisikan berikut ini

$$f(x) = f_0 + f_i x^i + x^n \text{ dengan } f_i, f_0 \in \{-1, 1\} \quad (1)$$

dan intejer i dipilih dalam selang $1 \leq i \leq n-1$. Di dalam fungsi hash yang akan dikonstruksi kemudian (ditulis di dalam artikel lanjutannya), pemilihan beberapa fungsi f secara acak dari keluarga trinomial tersebut akan diperlakukan sebagai kunci, dan akan dikaji pula apakah semua f tak-teruraikan atas $\mathbb{Z}$. Sedangkan di subseksi berikut ini kita cukup menunjukkan bahwa pemilihan f juga diarahkan ke pemenuhan syarat kedua.

4.2 Konstruksi Algoritme

Misalkan $a(x), b(x) \in \mathbb{Z}_p[x]$ dengan $\deg(a(x)) = c$ dan $\deg(b(x)) = d$, maka kita representasi $a(x)$ dan $b(x)$ secara terurut sebagai vektor $\mathbf{a} \in \mathbb{Z}_p^{c+1}$ dan $\mathbf{b} \in \mathbb{Z}_p^{d+1}$ masing-masing dengan komponen terkanan tak-nol. Berdasarkan definisi operasinya, komputasi baku dari *jumlah*, *kurang*, *kali*, dan *algoritme pembagian* (Proposisi 2) dari kedua polinomial tersebut kita bisa gunakan *algoritme buku sekolah* (school book algorithm), terlalu populer untuk dideskripsikan di artikel ini. Dalam algoritme ini, mudah pula diamati bahwa, jumlah dan kurang masing-masing memerlukan $\min\{c+1, d+1\}$ *operasi jumlah modulo* p, sedangkan kali dan algoritme pembagian masing-masing memerlukan $(c+1)(d+1)$ *operasi kali modulo* p ditambah dengan $(c+1)d$ *operasi jumlah modulo* p. Dalam hal ini, setiap operasi jumlah modulo p memerlukan $\lg p$ operasi bit dan setiap operasi kali modulo p memerlukan $(\lg p)^2$ operasi bit (lg merupakan notasi dari log basis 2).

Berikutnya, pandang $a(x), b(x) \in \mathbb{Z}_p[x] / \langle f(x) \rangle$ dengan representasi vektor $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_p^n$. Jika algoritme yang digunakan untuk menjumlahkan dan mengalikan $\mathbf{a}$ dan $\mathbf{b}$ mengikuti Proposisi 7, berarti kita menerapkan algoritme buku sekolah. Dengan demikian, *jumlah ring* $\mathbf{a} \oplus \mathbf{b}$ memerlukan n operasi jumlah modulo p, sedangkan *kali ring* $\mathbf{a} \odot \mathbf{b}$ melibatkan n^2 operasi jumlah modulo p ditambah dengan $n(n-1)$ operasi jumlah modulo p (yaitu menghitung kali polinomial $\mathbf{a}.\mathbf{b}$), kemudian banyaknya operasi ini dikalikan dengan 2 (karena dilanjutkan dengan menghitung sisa pembagian polinomial $\mathbf{a}.\mathbf{b}$ oleh $f(x)$).

Berdasarkan analisis komputasi di atas, dengan asumsi kita masih menggunakan algoritme untuk $\oplus$, sekarang kita turunkan algoritme untuk $\odot$ yang jauh lebih efisien dibanding algoritme buku sekolah atas dasar pemilihan trinomial f yang memenuhi Persamaan 1. Perhatikan dahulu bahwa

$$x^n = (f_0 + f_i x^i + x^n) + (-f_0 - f_i x^i) \Leftrightarrow x^n \bmod f(x) = (-f_0 - f_i x^i)$$

sehingga $xa(x) \bmod f(x) = (a_0x + a_1x^2 + ... + a_{n-2}x^{n-1} + a_{n-1}x^n) \bmod f(x)$

$$\begin{aligned}
&= \left[\left(a_0x + a_1x^2 + ... + a_{n-2}x^{n-1}\right) + \left(-f_0a_{n-1} - f_ia_{n-1}x^i\right)\right] \\
&= \left(-f_0a_{n-1} + a_0x + ... + a_{i-1}x^i + ... + a_{n-2}x^{n-1}\right) - f_ia_{n-1}x^i \\
&= \left(-f_0a_{n-1} + a_0x + ... + \left(a_{i-1} - f_ia_{n-1}\right)x^i + a_ix^i + ... + a_{n-2}x^{n-1}\right)
\end{aligned}$$

Dengan demikian, penghitungan $xa(x) \bmod f(x)$ di atas dapat kita pandang sebagai tranformasi *rotasi-substitusi* pada vektor $\mathbf{a} = (a_0, a_1, a_2, ..., a_{N-1}) \in \mathbb{Z}_p^n$ oleh trinomial f. Demi efisiensi, kita representasi $f(x) = f_0 + f_ix^i + x^n$ sebagai pasangan terurut $\mathbf{f} = (f_0, j) \in \{-1, 1\} \times \{\pm 1, \pm 2, ..., \pm(n-1)\}$ dengan $i = |j|$, $f_i = 1$ jika $j > 0$, dan $f_i = -1$ jika $j < 0$. Sebagai ilustrasi, untuk $n = 64$, $\mathbf{f} = (1, -37)$ merupakan representasi dari trinomial $f(x) = 1 - x^{37} + x^{64}$. Jadi, wujud algoritme penghitungan $xa(x) \bmod f(x)$ dengan mudah kita nyatakan berikut ini.

Algorithm 4. ***(Algoritme Rotasi-Substitusi)***
Input: *Intejer n dengan $n > 1$,prima ganjil p, pasangan terurut $\mathbf{f} = (f_0, j)$ sebagai representasi trinomial $f(x)$, dan vektor $\mathbf{a} = (a_0, a_1, a_2, ..., a_{n-1}) \in \mathbb{Z}_p^n$ sebagai representasi $a(x) \in \mathbb{Z}_p[x] / \langle f(x) \rangle$*
Output: *Vektor $\mathbf{c} = (c_0, c_1, c_2, ..., c_{n-1})$ sebagai $xa(x) \in \mathbb{Z}_p[x] / \langle f(x) \rangle$*

1. $\mathbf{c} := (\hookrightarrow \mathbf{a})$ *dimana* $\hookrightarrow \mathbf{a}$ *menotasikan putaran dari* $\mathbf{a}$ *ke kanan satu komponen.*

2. $\mathbf{c} := \mathbf{subs}(0, -f_0a_{n-1}, \mathbf{c})$ *menotasikan substitusi komponen ke-0 dari* $\mathbf{c}$ *dengan* $(-f_0a_{n-1})$.

3. *Jika* $j > 0$, *hitung* $s = a_{j-1} - a_{n-1}$, $\mathbf{c} := \mathbf{subs}(j, s, \mathbf{c})$, *dan*
   jika $j < 0$, *hitung* $s = a_{j-1} + a_{n-1}$, $\mathbf{c} := \mathbf{subs}(-j, s, \mathbf{c})$.

4. **return**($\mathbf{c}$).

Selanjutnya, untuk $b(x) = b_0 + b_1x + b_2x^2 + ... + b_{n-1}x^{n-1} \in \mathbb{Z}_p[x] / \langle f(x) \rangle$, kita amati bahwa $a(x)\,b(x) \bmod f(x)$ bisa dituliskan sebagai

$$\left(\left(b_0.a(x)\right) + b_1\left(x.a(x)\right) + b_2\left(x^2.a(x)\right) + ... + b_{n-1}\left(x^{n-1}.a(x)\right)\right) \bmod f(x)$$

Ekspresi ini menunjukkan bahwa operasi kali $a(x)\,b(x)$ dalam ring $\mathbb{Z}_p[x] / \langle f(x) \rangle$ dapat dipandang sebagai sejumlah n operasi $\oplus$ secara rekursif dari perkalian skalar b_i dengan *vektor rotasi-substitusi ke-i* dari $\mathbf{a}$. Jadi, Algoritme 4 merupakan subrutin dari algoritme operasi $\odot$. Lebih tegasnya, kita susun dalam algoritme operasi $\odot$ berikut ini.

Algorithm 5. ***(Algoritme Operasi $\odot$)***
Input: *Vektor* $\mathbf{a} = (a_0, a_1, a_2, ..., a_{m-1})$ *dan* $\mathbf{b} = (b_0, b_1, b_2, ..., b_{n-1})$ *dalam ring* $\mathbb{Z}_p^n \cong \mathbb{Z}_p[x] / \langle f(x) \rangle$
Output: *Vektor* $\mathbf{c} = (c_0, c_1, c_2, ..., c_{n-1})$ *sebagai hasil kali dari* $\mathbf{a}$ *dan* $\mathbf{b}$ *dalam ring* $\mathbb{Z}_p^n$.

1. *Inisialisasi* $\mathbf{c} := b_0\mathbf{a}$ *(menotasikan skalar kali vektor) dan* $\mathbf{w} := \mathbf{a}$.
2. *Untuk intejer* $i = 1$ *s.d.* $i = n - 1$, *hitung:*
   (a) $\mathbf{w} := \mathbf{RotSubs}(\mathbf{w}, \mathbf{f})$ *memanggil Algoritme 4.*
   (b) *Jika* $b_i \neq 0$, *hitung* $\mathbf{c} := \mathbf{c} \oplus b_i\mathbf{w}$
3. **return**$(\mathbf{c})$.

Karena kecepatan Algoritme 4 cenderung konstan terhadap pertumbuhan nilai n, maka yang paling dominan memengaruhi efisiensi komputasi Algoritme 5 adalah banyaknya operasi jumlah dan kali modulo p. Dalam hal ini, mudah kita amati bahwa Algoritme 5 melibatkan paling banyak n^2 operasi kali modulo p ditambah paling banyak $n(n-1)$ operasi jumlah modulo p. Jadi, jika dibanding dengan algoritme buku sekolah, Algoritme 5 menghemat 50% banyaknya operasi.

Tekait dengan kegunaan Algoritme 5 sebagai subrutin dalam konstruksi algoritme fungsi hash berbasis ideal latis, maka diasumsikan bahwa $\mathbf{b}$ adalah vektor biner. Oleh karena itu, *penghematan banyaknya operasi menjadi sangat signifikan.* Ini mudah diamati bahwa Algoritme 5 hanya memerlukan sebanyak $(n-1)n$ operasi jumlah modulo p, setara dengan $n(n-1)\lg p$ operasi bit. Dengan menetapkan p sebagai parameter konstan, berarti ukuran efisiensi Algoritme 5 adalah $\mathcal{O}(n^2)$ dengan satuan operasi bit[6].

Akhirnya, teorema berikut ini menegaskan bahwa Algoritme 5 mengarah ke pemenuhan syarat kedua sebagaimana dinyatakan dalam tujuan pembangunan aritmetik ring polinomial.

Teorema 6. *Misalkan* f *adalah trinomial yang memenuhi Persamaan 1. Jika* $\mathbf{u}$ *dan* $\mathbf{v}$ *adalah sembarang dua vektor satuan anggota* $\mathbb{Z}_p^n \cong \mathbb{Z}_p[x] / \langle f(x) \rangle$ *dan* $\mathbf{h} = \mathbf{u} \odot \mathbf{v}$ *dihitung berdasarkan Algoritme 5, maka* $\|\mathbf{h}\| < n$.

Bukti. Karena $\mathbf{u}$ dan $\mathbf{v}$ adalah vektor satuan, maka representasi polinomialnya bisa dituliskan $u(x) = x^j$ dan $u(x) = x^k$ untuk $j, k = 0, 1, 2, ..., n-1$, sehingga representasi polinomial dari $\mathbf{h}$ adalah

$$h(x) = u(x)\, v(x) \bmod f(x) = x^{j+k} \bmod f(x)$$

[6] $\mathcal{O}$ (baca "oh besar") adalah ukuran matematis kecepatan algoritme secara asimtotik. Definisi formalnya bisa mengacu buku teks baku yang memuat bahasan algoritme.

Jika $j+k<n$, maka bukti selesai karena $h(x)=x^{j+k}$ yang berarti $\mathbf{h}$ adalah vektor satuan dan sehingga $\|\mathbf{h}\|=\sqrt{1}$. Jika $j+k=n$, maka bukti juga selesai karena $h(x)=(-f_0-f_ix^i)$ yang berarti $\|\mathbf{h}\|=\sqrt{(-f_0)^2+(-f_i)^2}=\sqrt{2}$. Sekarang, kita pandang untuk kasus $j+k>n$, maka diperoleh $l_1=j+k-n$ dengan $1\leq l_1\leq n-2$ sehingga

$$\begin{aligned}h(x)&=\left(x^{l_1}.x^n\right)\bmod f(x)=x^{l_1}\left(-f_0-f_ix^i\right)\bmod f(x)\\&=-f_0x^{l_1}-f_ix^{l_1+i}\bmod f(x)\end{aligned}\tag{i}$$

Jika $l_1+i<n$, maka bukti selesai karena $h(x)=-f_0x^{l_1}-f_ix^{l_1+i}$ yang berarti $\mathbf{h}$ adalah vektor dengan $\|\mathbf{h}\|=\sqrt{2}$. Demikian pula jika $l_1+i=n$, maka h memunyai bentuk

$$h(x)=-f_0x^{l_1}-f_i\left(-f_0-f_ix^i\right)=f_0f_1-f_0x^{l_1}+x^i$$

sehingga nilai $\|\mathbf{h}\|$ yang paling mungkin adalah $\sqrt{3}$ ($\|\mathbf{h}\|$ bisa bernilai $\sqrt{5}$ yaitu kalau $i=l_1$ dan $f_0=-1$, kemungkinan ini sangat kecil apalagi ketika n cukup besar). Selanjutnya, untuk kasus $l_1+i>n$, maka diperoleh $l_2=l_1+i-n$ dengan $1\leq l_2\leq n-3$ dan dari Persamaan (i) kita peroleh

$$\begin{aligned}h(x)&=-f_0x^{l_1}-f_i\left(x^{l_2}.x^n\right)\bmod f(x)=-f_0x^{l_1}-f_ix^{l_2}\left(-f_0-f_ix^i\right)\bmod f(x)\\&=-f_0x^{l_1}+f_0f_ix^{l_2}+x^{l_2+i}\bmod f(x)\end{aligned}\tag{ii}$$

Jika $l_2+i<n$, maka bukti selesai karena $h(x)=-f_0x^{l_1}+f_0f_ix^{l_2}+x^{l_2+i}$. Demikian pula jika $l_2+i=n$, maka h memunyai bentuk

$$h(x)=-f_0x^{l_1}+f_0f_ix^{l_2}+\left(-f_0-f_ix^i\right)=-f_0-f_0x^{l_1}+f_0f_ix^{l_2}-f_ix^i$$

Untuk kasus $l_2+i>n$, maka diperoleh $l_3=l_2+i-n$ dengan $1\leq l_3\leq n-4$ dan dari Persamaan (ii) kita peroleh

$$\begin{aligned}h(x)&=-f_0x^{l_1}+f_0f_ix^{l_2}+x^{l_3}\left(-f_0-f_ix^i\right)\bmod f(x)\\&=-f_0x^{l_1}+f_0f_ix^{l_2}-f_0x^{l_3}-f_ix^{l_3+i}\bmod f(x)\end{aligned}$$

sehingga

$$h(x)=\begin{cases}-f_0x^{l_1}+f_0f_ix^{l_2}-f_0x^{l_3}-f_ix^{l_3+i}\text{ jika }l_3+i<n\\f_0f_i-f_0x^{l_1}+f_0f_ix^{l_2}-f_0x^{l_3}+x^i\text{ jika }l_3+i<n\end{cases}$$

Demikian seterusnya, mengikuti pola dari uraian di atas dijamin ada inejer $s\in\{1,2,\ldots,n-2\}$ dan ada l_s dengan $1\leq l_s\leq n-(s+1)$ sedemikian sehingga h memiliki bentuk

$$\begin{aligned}h(x)&=h_1x^{l_1}+h_2x^{l_2}+\cdots+h_sx^{l_s}+h_{s+1}x^{l_s+i}\text{ atau}\\h(x)&=h_0+h_1x^{l_1}+h_2x^{l_2}+\cdots+h_sx^{l_s}+h_{s+1}x^i\end{aligned}$$

dengan $h_u \in \{1, -1\}$ untuk $u = 0, 1, ..., s + 1$. Oleh karena itu, jelas bahwa nilai $\|\mathbf{h}\|$ terbesar terjadi ketika $s = n - 2$ dengan representasi polinomial

$$h(x) = h_0 + h_1 x^{l_1} + h_2 x^{l_2} + \cdots + h_{n-2} x^{l_{n-2}} + h_{n-1} x^{i}$$

seihingga $\|\mathbf{h}\| < n$. □

Walaupun Teorema 6 menyatakan bahwa $\|\mathbf{h}\|$ terbatas ke n, akan tetapi berdasarkan buktinya, nilai $\|\mathbf{h}\|$ cenderung jauh lebih kecil dari n. Dengan kata lain, vektor $\mathbf{h}$ cenderung merupakan vektor pendek, umumnya terbatas ke $\sqrt{n}$. Hal ini perlu kajian lebih jauh dengan menggunakan analisis peluang.

Daftar Pustaka

[1] M. Ajtai. "Generating hard instance of lattice problems". In *Complexity of Computations and Proofs*, vol. 3 of *Quad. Math.*, p. 1-32. Dept. Math., Sconda University Napoli, Caserta, 2004. Preliminary version in STOC 1996.

[2] K. Bebtahar, D. Page, J. Silverman, M. Saarinen, and N. Smart. "LASH". In *Technical Report*, 2nd NIST Cryptografic Hash Function Workshop, 2006.

[3] J. Y. Cai and A. Nerurkar. "An improved worst-case to avarage-case connection for lattice problems". In *Proc. 38th IEEE Symp. on foundations of Computer Sci. (FOCS)*., p. 468-477, 1997.

[4] P. A. Fuhrmann, "A Polynomial Approach to Linear Algebra". *Springer-Verlag.*, 1996. ISBN: 0-387-94643-8.

[5] O. Goldreich, S. Goldwasser, and S. Halevi. "Collision-free hashing from lattice problems". In *Technical Report TR96-056, Electronic Collouqium on Computational Complexity* (ECCC), 1996.

[6] V. Lyubashevsky and D. Micciancio. "Generalized compact knapsacks are collision resistant". In *33rd International Colloquium on Automata, Languages and Programming* (ICALP), 2006.

[7] V. Lubyashevsky, D. Micciancio, C. Peikert, and A. Rosen. "SWIFFT: modest proposal for FFT hashing". In *FSE 2008*, 2008.

[8] R. P. Menezes, P. C. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography." *CRC Press, Inc.*, 1997.

[9] D. Micciancio, "Improved crytographic hash functions with worst-case and average-case connections". In *Proc. 34th ACM Symp. on Theory of Computing (STOC)*, p. 609-618. 2002.

[10] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures". In *Proc. 45th Annual IEEE Symp. on foundations of Computer Sci. (FOCS)*. p. 609-618, 2002.

[11] C. Peikert and A. Rosen. "Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices". In *3rd Theory of Cryptography Conference* (TCC), p. 145-166. 2006.

[12] C. C. Pinter, "A Book of Abstract Algebra". *McGraw-Hill Inc.*, 1990. ISBN: 0-07-100855-1.