Bab 13

Matematika VI - Test Bilangan Prima

Dalam *public key cryptography* (lihat bab 16), bilangan prima yang sangat besar yang dipilih secara "acak" kerap dibutuhkan. Suatu bilangan ganjil n yang sangat besar dipilih secara acak dan $n, n+2, n+4, \ldots$ dan seterusnya ditest hingga bilangan prima pertama ditemukan. Cara naif untuk test bilangan prima yang bersifat deterministik (pasti) adalah untuk mencoba membagi bilangan yang sedang ditest (sebut saja n) dengan setiap bilangan ganjil > 2 sampai dengan $\sqrt{n}$. Jika ada yang membagi n maka n bukan bilangan prima, sebaliknya jika tidak ada yang membagi maka n merupakan bilangan prima. Tentu saja cara ini tidak praktis untuk nilai n yang sangat besar. Karena algoritma deterministik untuk test bilangan prima tidak efisien (meskipun cara paling efisien yang sudah ditemukan mempunyai kompleksitas[1] yang tergolong *polynomial* — lihat [agr04]), maka algoritma probabilistik digunakan dalam kriptografi.

13.1 Pseudoprime dan Bilangan Carmichael

Sekarang kita bahas konsep yang digunakan mayoritas algoritma probabilistik untuk test bilangan prima. Karena 2 merupakan satu-satunya bilangan prima genap, test bilangan prima fokus pada bilangan ganjil. Berbagai varian algoritma dengan konsep berikut menggunakan cara berbeda untuk mempercepat komputasi dan juga untuk dapat menjadi efektif terhadap bilangan Carmichael (akan dibahas), tetapi dasar yang digunakan adalah *Fermat's little theorem*:

$$b^{n-1} \equiv 1 \pmod{n} \tag{13.1}$$

[1]Kompleksitas asimtotik untuk test bilangan prima adalah ukuran waktu atau memori yang diperlukan untuk komputasi seiring besarnya (dalam bits) bilangan yang sedang ditest.

jika n prima dan $\gcd(b, n) = 1$. Untuk n komposit (dapat diuraikan, jadi tidak prima), persamaan 13.1 kadang berlaku, akan tetapi ini lebih jarang terjadi daripada situasi dimana persamaan tidak berlaku.

Definisi 42 (Pseudoprime) *Jika n adalah bilangan komposit ganjil dan terdapat b dengan $\gcd(b, n) = 1$ yang mematuhi persamaan 13.1, maka n disebut pseudoprime untuk base b.*

Sebagai contoh, $n = 91$ adalah *pseudoprime* untuk *base* 3 karena $3^{90} \equiv 1 \pmod{91}$, akan tetapi 91 bukan *pseudoprime* untuk *base* 2 karena $2^{90} \equiv 64 \pmod{91}$. Kita akan bahas teori yang diperlukan untuk memberi gambaran mengenai probabilitas untuk *pseudoprime*. Kita mulai dengan teorema mengenai *order* dari *base* untuk *pseudoprime*.

Teorema 88 *Untuk suatu bilangan komposit ganjil n, n adalah pseudoprime untuk base b jika dan hanya jika order dari b dalam $(\mathbf{Z}/n\mathbf{Z})^*$ membagi $n - 1$.*

Kita mulai pembuktian teorema 88 dengan mengumpamakan bahwa n *pseudoprime* untuk *base* b dan d adalah *order* dari b dalam $(\mathbf{Z}/n\mathbf{Z})^*$. Jika d tidak membagi $n - 1$ maka kita dapatkan *remainder* $r < d$ dimana $n - 1 = cd + r$ untuk suatu c, dan karena $b^{n-1} \equiv b^{cd} \equiv 1 \pmod{n}$, maka

$$\begin{aligned} b^r &= b^{n-1-cd} \\ &\equiv 1 \pmod{n}, \end{aligned}$$

sesuatu kontradiksi karena d merupakan pangkat terkecil dari b yang menghasilkan $1 \pmod{n}$, jadi $d|n-1$. Sebaliknya jika $d|n-1$, maka $n-1 = cd$ untuk suatu c, dan kita dapatkan

$$\begin{aligned} b^{n-1} &= b^{cd} \\ &\equiv 1 \pmod{n}, \end{aligned}$$

jadi n merupakan *pseudoprime* untuk *base* b. Selesailah pembuktian teorema 88. Kita buktikan satu teorema lagi sebelum kita bahas teorema utama yang memberi gambaran mengenai probabilitas *pseudoprime*.

Teorema 89 *Untuk suatu bilangan komposit ganjil n, jika n adalah pseudoprime untuk base b_1 dan base b_2, maka n adalah pseudoprime untuk base $b_1 b_2$ dan base $b_1 b_2^{-1}$ dimana b_2^{-1} adalah inverse dari $b_2 \pmod{n}$.*

Untuk menunjukkan bahwa n *pseudoprime* untuk *base* $b_1 b_2$, pertama kita harus tunjukkan bahwa $\gcd(b_1 b_2, n) = 1$. Jika $\gcd(b_1 b_2, n) = d > 1$ maka $d|b_1 b_2$, dan karena $d \not| b_1$ dan $d \not| b_2$ maka $d = d_1 d_2$ dimana $d_1 > 1$, $d_1|b_1$ dan $d_2 > 1$, $d_2|b_2$. Ini berarti $\gcd(b_1, n)$ merupakan kelipatan dari d_1 (karena d_1 juga membagi

n), sesuatu yang tidak mungkin karena $\gcd(b_1, n) = 1$. Jadi $\gcd(b_1 b_2, n) = 1$. Yang kedua, kita harus tunjukkan bahwa $(b_1 b_2)^{n-1} \equiv 1 \pmod{n}$:

$$\begin{aligned}(b_1 b_2)^{n-1} &= b_1^{n-1} b_2^{n-1} \\ &\equiv 1 \pmod{n}.\end{aligned}$$

Jadi n merupakan *pseudoprime* untuk *base* $b_1 b_2$. Berikutnya, kita ingin tunjukkan bahwa n *pseudoprime* untuk *base* b_2^{-1}. Kita mengetahui bahwa $b_2 b_2^{-1} \equiv 1 \pmod{n}$ yang berarti terdapat suatu bilangan bulat c_1 dimana

$$b_2 b_2^{-1} = c_1 n + 1.$$

Jika $\gcd(b_2^{-1}, n) = d > 1$ maka karena $d | b_2 b_2^{-1}$, terdapat bilangan bulat c_2 dimana $c_2 d = b_2 b_2^{-1}$, jadi

$$c_2 d = c_1 n + 1.$$

Karena $d|n$, terdapat bilangan bulat c_3 dimana $n = c_3 d$, jadi

$$c_2 d = c_1 c_3 d + 1$$

yang berarti $d|1$, sesuatu yang tidak mungkin. Jadi $\gcd(b_2^{-1}, n) = 1$. Berikut kita ingin tunjukkan bahwa $(b_2^{-1})^{n-1} \equiv 1 \pmod{n}$:

$$\begin{aligned}1 &\equiv 1^{n-1} \pmod{n} \\ &\equiv (b_2 b_2^{-1})^{n-1} \pmod{n} \\ &\equiv b_2^{n-1} (b_2^{-1})^{n-1} \pmod{n} \\ &\equiv (b_2^{-1})^{n-1} \pmod{n}.\end{aligned}$$

Karena n merupakan *pseudoprime* untuk *base* b_1 dan *base* b_2^{-1} maka n merupakan *pseudoprime* untuk *base* $b_1 b_2^{-1}$. Selesailah pembuktian teorema 89.

Sekarang kita bahas teorema utama yang memberi gambaran mengenai probabilitas *pseudoprime*.

Teorema 90 *Untuk suatu bilangan n ganjil dan komposit, jika persamaan 13.1 tidak berlaku untuk suatu base b, maka persamaan 13.1 tidak berlaku untuk sedikitnya setengah dari semua base untuk* $(\mathbf{Z}/n\mathbf{Z})^*$.

Untuk membuktikan teorema 90, kita buat himpunan $\{b_1, b_2, \ldots, b_s\}$ sebagai himpunan *residue* dari semua *base* yang menjadikan n *pseudoprime*, jadi himpunan terdiri dari semua bilangan $0 < b_i < n$ yang lulus test persamaan 13.1. Jika b merupakan *base* yang gagal menjadikan n *pseudoprime*, maka berdasarkan teorema 89, bb_i juga gagal menjadikan n suatu *pseudoprime*, karena jika bb_i menjadikan n suatu *pseudoprime*, maka n juga merupakan *pseudoprime* untuk:

$$b \equiv (bb_i) b_i^{-1} \pmod{n},$$

suatu kontradiksi. Jadi himpunan *residue*

$$\{bb_1, bb_2, \ldots, bb_s\}$$

merupakan himpunan yang besarnya sama dengan himpunan $\{b_1, b_2, \ldots, b_s\}$ dan setiap elemennya menggagalkan n menjadi *pseudoprime*. Jadi jika terdapat *base* b yang menggagalkan n menjadi *pseudoprime*, maka sedikitnya separuh dari *residue classes* $\pmod n$ akan menggagalkan n menjadi *pseudoprime*. Selesailah pembuktian teorema 90. Kecuali jika n lulus test persamaan 13.1 untuk semua *base* b dengan $\gcd(b, n) = 1$, maka dengan probabilitas sedikitnya 50 persen, test persamaan akan gagal untuk suatu b yang dipilih secara acak. Secara garis besar, algoritma untuk test bilangan prima mengulang langkah-langkah berikut hingga terjadi kegagalan atau kita sudah cukup puas dengan probabilitas yang diberikan bahwa n merupakan bilangan prima.

1. Pilih suatu bilangan b secara acak sebagai *base* dimana $0 < b < n$.
2. Kalkulasi $d = \gcd(b, n)$ menggunakan algoritma Euclid.
3. Jika $d > 1$ maka n adalah bilangan komposit dan d merupakan faktor yang membagi n, kita selesai.
4. Jika $d = 1$ kita lakukan test persamaan 13.1 terhadap b. Jika tidak lulus maka n adalah bilangan komposit dan kita selesai. Jika lulus maka probabilitas bahwa n merupakan bilangan prima semakin besar.

Jika langkah-langkah diatas menghasilkan jawaban komposit, maka kita tahu dengan pasti bahwa n adalah bilangan komposit. Jika tidak menjawab komposit, maka probabilitas bahwa n adalah bilangan komposit yang akan gagal test persamaan 13.1 untuk suatu *base* adalah $\leq \frac{1}{2}$. Jadi jika langkah-langkah diatas diulang sebanyak k kali tanpa jawaban komposit, setiap kali dengan *base* baru yang dipilih secara acak, maka probabilitas bahwa n adalah bilangan komposit yang akan gagal test persamaan 13.1 untuk suatu *base* adalah $\leq \frac{1}{2^k}$, dan probabilitas bahwa n akan lulus test persamaan 13.1 untuk semua *base* adalah $\geq 1 - \frac{1}{2^k}$.

Adakah bilangan komposit yang lulus test persamaan 13.1 untuk semua *base*? Jawabnya ada, yaitu bilangan Carmichael (*Carmichael number*).

Definisi 43 (Carmichael) *Bilangan komposit n adalah Carmichael jika*

$$a^{n-1} \equiv 1 \pmod n$$

untuk semua $a \in (\mathbf{Z}/n\mathbf{Z})^*$ $(1 \leq a \leq n-1)$.

Definisi 44 (Carmichael Lambda Function) *Kita definisikan Carmichael lambda function sebagai*

$$\lambda(n) = exp((\mathbf{Z}/n\mathbf{Z})^*)$$

dimana $exp(G)$ adalah pangkat positif terkecil e dengan $a^e = 1$ untuk setiap a dalam finite group G, jadi e merupakan kelipatan persekutuan terkecil (lowest common multiple) semua order elemen dalam G. Kita definisikan juga bahwa $\lambda(1) = 1$.

Teorema 91 *Suatu bilangan komposit n adalah Carmichael jika dan hanya jika $\lambda(n)$ membagi $n - 1$.*

Pembuktian teorema 91 cukup mudah karena $e = \lambda(n)$ adalah pangkat terkecil yang menghasilkan 1 jika dipangkatkan ke setiap elemen dalam $(\mathbf{Z}/n\mathbf{Z})^*$, jadi $a^e \equiv 1 \pmod{n}$ untuk setiap $a \in (\mathbf{Z}/n\mathbf{Z})^*$. Definisi bilangan Carmichael mengatakan bahwa $a^{n-1} \equiv 1 \pmod{n}$ untuk setiap elemen $a \in (\mathbf{Z}/n\mathbf{Z})^*$. Jika e tidak membagi $n - 1$, maka terdapat $0 < r < e$ dimana

$$be + r = n - 1$$

untuk suatu b, dan untuk setiap $a \in (\mathbf{Z}/n\mathbf{Z})^*$:

$$a^r = a^{n-1-be} \equiv 1 \pmod{n}.$$

Ini adalah suatu kontradiksi karena e merupakan pangkat positif terkecil yang menghasilkan 1 untuk setiap $a \in (\mathbf{Z}/n\mathbf{Z})^*$. Jadi $e = \lambda(n)$ membagi $n - 1$, dan selesailah pembuktian teorema 91. Berikut adalah beberapa persamaan mengenai λ:

$$\lambda(p^e) = p^{e-1}(p-1) \text{ untuk bilangan prima ganjil } p. \tag{13.2}$$

$$\lambda(2^e) = 2^{e-2} \text{ untuk } e \geq 3. \tag{13.3}$$

$$\lambda(2) = 1. \tag{13.4}$$

$$\lambda(4) = 2. \tag{13.5}$$

$$\lambda(n) = \text{lcm}_{1 \leq i \leq k} \lambda(p_i^{e_i}) \text{ jika } n = \prod_{i=1}^{k} p_i^{e_i}. \tag{13.6}$$

Persamaan 13.2 didapat karena $(\mathbf{Z}/p^e\mathbf{Z})^*$ dengan bilangan prima ganjil p dengan $e \geq 1$ merupakan *cyclic group* (lihat teorema 38) jadi mempunyai elemen dengan *order* terbesar $\phi(p^e) = p^{e-1}(p-1)$. Persamaan 13.3 didapat karena $(\mathbf{Z}/2^e\mathbf{Z})^*$ merupakan produk dari dua *cyclic group*, dan *order* terbesar 2^{e-2} merupakan kelipatan dari setiap *order* dalam $(\mathbf{Z}/2^e\mathbf{Z})^*$ dengan $e \geq 3$ (lihat teorema 36). Persamaan 13.4 dan 13.5 didapat karena $(\mathbf{Z}/2\mathbf{Z})^*$ dan $(\mathbf{Z}/4\mathbf{Z})^*$ merupakan *cyclic group* dengan *order* terbesar masing-masing 1 dan 2 (lihat teorema 35). Untuk menunjukkan persamaan 13.6, dimana $\prod_{i=1}^{k} p_i^{e_i}$ adalah *prime factorization* dari n, kita buat $t = \text{lcm}_{1 \leq i \leq k} \lambda(p_i^{e_i})$. Jadi

$$x^t \equiv 1 \pmod{p_i^{e_i}}$$

untuk $1 \leq i \leq k$, jadi berdasarkan *Chinese Remainder Theorem* (teorema 31) kita dapatkan

$$x^t \equiv 1 \pmod{n},$$

jadi $\lambda(n)|\text{lcm}_{1\leq i\leq k}\lambda(p_i^{e_i})$. Sekarang kita pilih a_i dengan *order* $\lambda(p_i^{e_i})$ dalam $(\mathbf{Z}/p_i^{e_i}\mathbf{Z})^*$ untuk setiap $1 \leq i \leq k$, dan kita buat $x \equiv a_i \pmod{p_i^{e_i}}$. Kita ingin tunjukkan bahwa x mempunyai *order* $t = \text{lcm}_{1\leq i\leq k}\lambda(p_i^{e_i})$ dalam $(\mathbf{Z}/n\mathbf{Z})^*$. Jika $x^{t/l} \equiv 1 \pmod{n}$ untuk $1 < l \leq t$, maka ada $\lambda(p_j^{e_j})$ yang tidak membagi t/l, dimana $1 \leq j \leq k$. Jadi $x^{\gcd(\lambda(p_j^{e_j}),t/l)} \equiv 1 \pmod{p_j^{e_j}}$. Karena $1 \leq \gcd(\lambda(p_j^{e_j}), t/l) < \lambda(p_j^{e_j})$, ini mengkontradiksi fakta bahwa $\lambda(p_j^{e_j})$ adalah pangkat terkecil dari x yang menghasilkan 1 dalam $(\mathbf{Z}/p_j^{e_j}\mathbf{Z})^*$. Jadi x mempunyai *order*

$$t = \text{lcm}_{1\leq i\leq k}\lambda(p_i^{e_i})$$

dalam $(\mathbf{Z}/n\mathbf{Z})^*$. Selesailah pembuktian persamaan 13.6. Sebagai aplikasi dari persamaan 13.6, kita tunjukkan bahwa 561 adalah bilangan Carmichael:

$$561 = 3 \cdot 11 \cdot 17$$

jadi

$$\begin{aligned} \lambda(561) &= \text{lcm}(\lambda(3), \lambda(11), \lambda(17)) \\ &= \text{lcm}(2, 10, 16) \\ &= 80. \end{aligned}$$

Karena 80 membagi $561 - 1 = 560$, maka berdasarkan teorema 91, 561 adalah suatu bilangan Carmichael.

Teorema 92 *Jika n merupakan bilangan Carmichael, maka n adalah bilangan ganjil, bebas kuadrat, dan merupakan produk dari sedikitnya 3 bilangan prima.*

Suatu bilangan disebut bebas kuadrat (*square-free*) jika tidak bisa dibagi oleh kuadrat suatu bilangan $p > 1$. Untuk menunjukkan bahwa $n > 2$ ganjil, berdasarkan persamaan 13.2 — 13.6, tidak terlalu sukar untuk melihat bahwa $2|\lambda(n)$. Karena $\lambda(n)|n-1$ (teorema 91), maka $2|n-1$, yang berarti n ganjil. Selanjutnya, jika $p^2|n$ untuk suatu bilangan ganjil $p > 2$, kita dapatkan $p|\lambda(n)$, jadi $p|n-1$ berdasarkan teorema 91. Jadi $p|n$ dan $p|n-1$ yang berarti $p|1$, suatu kontradiksi dengan $p > 2$. Yang terahir, jika $n = pq$ dengan dua bilangan prima ganjil p, q yang berlainan, maka $p-1|\lambda(n)$, jadi karena $\lambda(n)|n-1$, kita dapatkan $pq - 1 = n - 1 \equiv 0 \pmod{p-1}$. Tetapi $p \equiv 1 \pmod{p-1}$, jadi $q \equiv 1 \pmod{p-1}$, dan oleh karena itu $q \geq p$ (jika $q < p$ maka $q < p-1$, dan karena $q > 1$ maka $q \not\equiv 1 \pmod{p-1}$). Sebaliknya, dengan $q-1|\lambda(n)$ kita dapatkan $pq - 1 = n - 1 \equiv 0 \pmod{q-1}$. Tetapi $q \equiv 1 \pmod{q-1}$, jadi $p \equiv 1 \pmod{q-1}$, dan oleh karena itu $p \geq q$ (jika $p < q$ maka $p < q-1$, dan karena $p > 1$ maka $p \not\equiv 1 \pmod{q-1}$). Jadi $p = q$, suatu kontradiksi karena n bebas kuadrat. Selesailah pembuktian teorema 92.

13.2 Metode Solovay-Strassen

Banyaknya bilangan Carmichael tidak terbatas. Oleh sebab itu beberapa algoritma untuk test bilangan prima mencoba test yang lebih ketat, satu diantaranya adalah test untuk *Euler pseudoprime*. Untuk n suatu bilangan komposit ganjil, $\left(\frac{b}{n}\right)$ merupakan simbol Jacobi. Jika n merupakan bilangan prima, maka teorema 54 memberikan

$$b^{(n-1)/2} \equiv \left(\frac{b}{n}\right) \pmod{n}. \tag{13.7}$$

Definisi 45 (Euler Pseudoprime) *Bilangan komposit ganjil n yang lulus test persamaan 13.7 untuk base b disebut Euler pseudoprime untuk base b.*

Teorema 93 *Jika n merupakan Euler pseudoprime untuk base b, maka n merupakan pseudoprime untuk base b.*

Untuk membuktikan teorema 93 kita harus tunjukkan bahwa jika persamaan 13.7 berlaku, maka persamaan 13.1 juga berlaku. Ini dapat dilakukan dengan mengkuadratkan kedua sisi dari persamaan 13.7:

$$\begin{aligned} (b^{(n-1)/2})^2 &\equiv \left(\frac{b}{n}\right)^2 \pmod{n} \\ b^{n-1} &\equiv 1 \pmod{n}. \end{aligned}$$

Jika n adalah bilangan komposit ganjil, kita ingin tunjukkan bahwa persamaan 13.7 akan gagal untuk sedikitnya 50 persen dari semua *base* $b \in (\mathbf{Z}/n\mathbf{Z})^*$. Kita mulai dengan menunjukkan bahwa jika *base* b_1 lulus test persamaan 13.7 dan *base* b_2 gagal, maka *base* $b_1 b_2$ akan gagal. Jika persamaan 13.7 berlaku untuk b_1 dan $b_1 b_2$:

$$\begin{aligned} b_1^{(n-1)/2} &\equiv \left(\frac{b_1}{n}\right) \pmod{n} \\ (b_1 b_2)^{(n-1)/2} &\equiv \left(\frac{b_1 b_2}{n}\right) \pmod{n} \end{aligned}$$

maka karena

$$\begin{aligned} (b_1 b_2)^{(n-1)/2} &= b_1^{(n-1)/2} b_2^{(n-1)/2} \text{ dan} \\ \left(\frac{b_1 b_2}{n}\right) &= \left(\frac{b_1}{n}\right)\left(\frac{b_2}{n}\right), \end{aligned}$$

kita dapatkan

$$b_2^{(n-1)/2} \equiv \left(\frac{b_2}{n}\right) \pmod{n}.$$

Jadi jika

$$b_2^{(n-1)/2} \not\equiv \left(\frac{b_2}{n}\right) \pmod{n}$$

maka

$$(b_1 b_2)^{(n-1)/2} \not\equiv \left(\frac{b_1 b_2}{n}\right) \pmod{n}.$$

Berikutnya kita ingin tunjukkan bahwa jika bilangan n komposit dan ganjil, maka terdapat *base* b dimana persamaan 13.7 gagal. Jika suatu bilangan komposit dan ganjil n lulus persamaan 13.7 untuk semua *base*, maka

$$\left(\frac{b}{n}\right)^2 \equiv b^{n-1} \equiv 1 \pmod{n}$$

untuk semua b, jadi n merupakan bilangan Carmichael. Menurut teorema 92, n bebas kuadrat. Kita dapat uraikan n menjadi $n = pr$ dimana p adalah bilangan prima dan $\gcd(p, r) = 1$. Kita ambil satu *quadratic non-residue* g dalam $(\mathbf{Z}/p\mathbf{Z})^*$ dan kita pilih a:

$$\begin{aligned} a &\equiv g \pmod{p}, \\ a &\equiv 1 \pmod{r}. \end{aligned}$$

Berdasarkan *Chinese Remainder Theorem* (teorema 31), a dapat dipilih. Kita dapatkan

$$\left(\frac{a}{n}\right) = \left(\frac{a}{pr}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{r}\right) = \left(\frac{g}{p}\right)\left(\frac{1}{r}\right) = (-1)(+1) = -1$$

menggunakan persamaan 11.12. Dengan asumsi persamaan 13.7 lulus untuk semua *base*, kita dapatkan

$$\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \equiv -1 \pmod{n}.$$

Karena $r|n$ maka

$$a^{(n-1)/2} \equiv -1 \pmod{r}.$$

Akan tetapi ini adalah suatu kontradiksi dengan $a \equiv 1 \pmod{r}$, jadi tidak mungkin semua *base* lulus test persamaan 13.7, jadi terdapat *base* b yang gagal. Yang terahir, kita tunjukkan bahwa jika ada *base* b yang gagal test persamaan 13.7, maka sedikitnya 50 persen dari *base* $b \in (\mathbf{Z}/n\mathbf{Z})^*$ gagal. Jika $\{b_1, b_2, \ldots, b_s\}$ merupakan himpunan semua *base* yang lulus test persamaan 13.7, maka $\{bb_1, bb_2, \ldots, bb_s\}$ merupakan himpunan yang elemennya semua gagal dan besarnya sama dengan besar himpunan $\{b_1, b_2, \ldots, b_s\}$. Jadi sedikitnya 50 persen dari semua *base* akan gagal test persamaan 13.7. Ini menjadi dasar dari algoritma Solovay-Strassen:

1. Pilih suatu bilangan b secara acak sebagai *base* dimana $0 < b < n$.
2. Kalkulasi $d = \gcd(b, n)$ menggunakan algoritma Euclid.
3. Jika $d > 1$ maka n adalah bilangan komposit dan d merupakan faktor yang membagi n, kita selesai.
4. Jika $d = 1$ kita lakukan test persamaan 13.7 terhadap b. Jika tidak lulus maka n adalah bilangan komposit dan kita selesai. Jika lulus maka probabilitas bahwa n merupakan bilangan prima semakin besar.

Jika langkah-langkah diatas menghasilkan jawaban komposit, maka kita tahu dengan pasti bahwa n adalah bilangan komposit. Jika tidak menghasilkan jawaban komposit, maka probabilitas bahwa n adalah bilangan komposit adalah $\leq \frac{1}{2}$. Jadi jika langkah-langkah diatas diulang sebanyak k kali tanpa jawaban komposit[2], setiap kali dengan *base* baru yang dipilih secara acak, maka probabilitas bahwa n adalah bilangan komposit adalah $\leq \frac{1}{2^k}$, dan probabilitas bahwa n merupakan bilangan prima adalah $\geq 1 - \frac{1}{2^k}$. Algoritma ini adalah contoh dari metode Monte Carlo yaitu algoritma probabilistik yang hasilnya hampir selalu benar. Kita dapat mengulang langkah-langkah sebanyak mungkin hingga probabilitas mendapatkan jawaban yang salah (tidak ada jawaban komposit, tetapi n komposit) adalah sangat kecil; sebagai contoh kita dapat targetkan agar probabilitas mendapatkan jawaban yang salah jauh lebih kecil dari probabilitas malfungsi hardware untuk komputasi.

13.3 Metode Miller-Rabin

Kita telah perkenalkan konsep *pseudoprime* dan *Euler pseudoprime*. Kita perkenalkan satu konsep lagi yaitu *strong pseudoprime* yang akan digunakan dalam algoritma Miller-Rabin, yang seperti Solovay-Strassen, merupakan algoritma Monte Carlo. Jika n adalah suatu bilangan komposit ganjil, maka terdapat s dan t dimana

$$n - 1 = 2^s t$$

dengan t berupa bilangan ganjil.

Definisi 46 (Strong Pseudoprime) *n adalah suatu strong pseudoprime untuk base b jika*

$$b^t \equiv 1 \pmod{n} \text{ atau} \tag{13.8}$$
$$b^{2^r t} \equiv -1 \pmod{n} \tag{13.9}$$

untuk suatu r dimana $0 \leq r < s$.

[2]Tentunya sekali jawaban komposit ditemukan, tidak ada gunanya untuk mengulang kembali langkah-langkah algoritma Solovay-Strassen.

Ide dibalik *strong pseudoprime* adalah fakta bahwa dalam $\mathbf{Z}/p^m\mathbf{Z}$ dimana p adalah bilangan prima ganjil dan $m \geq 1$, 1 mempunyai tidak lebih dan tidak kurang dari dua akar kuadrat: 1 dan -1. Jika n bukan merupakan pemangkatan bilangan prima ganjil, maka terdapat $2^{\omega(n)}$ akar kuadrat dari 1 yang berbeda, dimana $\omega(n)$ adalah banyaknya bilangan prima yang berbeda yang membagi n. Jadi *strong pseudoprime* lebih ketat lagi dari *pseudoprime*: bukan hanya $a^{n-1} \equiv 1 \pmod{n}$ saja yang ditest, akan tetapi akar kuadrat dari 1 juga ditest.

Teorema 94 *Jika n merupakan strong pseudoprime untuk base b, maka n merupakan Euler pseudoprime untuk base b.*

Kita harus buktikan untuk bilangan komposit ganjil n bahwa jika persamaan 13.8 atau 13.9 berlaku untuk *base b*, maka persamaan 13.7 juga berlaku. Jika persamaan 13.8 berlaku, maka

$$\begin{aligned} b^{(n-1)/2} &\equiv 1 \pmod{n} \\ &= \left(\frac{1}{n}\right) \\ &= \left(\frac{b^t}{n}\right) \\ &= \left(\frac{b}{n}\right)^t. \end{aligned}$$

Karena t ganjil, maka $\left(\frac{b}{n}\right) = 1$, jadi

$$b^{(n-1)/2} \equiv \left(\frac{b}{n}\right) \pmod{n},$$

yang berarti persamaan 13.7 juga berlaku. Jika persamaan 13.9 berlaku dengan $r = s - 1$, maka

$$b^{(n-1)/2} = b^{2^{(s-1)}t} \equiv -1 \pmod{n},$$

jadi kita ingin tunjukkan bahwa $\left(\frac{b}{n}\right) = -1$. Jika p adalah bilangan prima ganjil yang membagi n, kita dapat uraikan $p - 1 = 2^{s'}t'$ dimana t' adalah bilangan ganjil. Kita ingin tunjukkan bahwa $s' \geq s$ dan

$$\left(\frac{b}{p}\right) = \begin{cases} -1, & \text{jika } s' = s; \\ 1 & \text{jika } s' > s. \end{cases}$$

Karena $b^{2^{s-1}t} \equiv -1 \pmod{n}$, kita pangkatkan kedua sisi persamaan dengan t' yang merupakan bilangan ganjil dan mendapatkan:

$$\begin{aligned} (b^{2^{s-1}t})^{t'} &\equiv (-1)^{t'} \pmod{n}; \\ (b^{2^{s-1}t'})^{t} &\equiv -1 \pmod{n}. \end{aligned}$$

Karena $p|n$ maka

$$(b^{2^{s-1}t'})^t \equiv -1 \pmod{p}.$$

Jika $s' < s$, maka

$$b^{p-1} = b^{2^{s'}t'} \not\equiv 1 \pmod{p},$$

sesuatu yang bertentangan dengan *Fermat's little theorem* (teorema 30). Jadi $s' \geq s$. Jika $s' = s$, maka

$$\begin{aligned}\left(\frac{b}{p}\right) &\equiv b^{(p-1)/2} \pmod{p} \text{ karena } p \text{ prima} \\ &\equiv b^{2^{s'-1}t'} \pmod{p} \\ &\equiv -1 \pmod{p} \text{ (tidak mungkin 1 karena } (b^{2^{s'-1}t'})^t \equiv -1).\end{aligned}$$

Jika $s' > s$, maka

$$\begin{aligned}((b^{2^{s-1}t'})^t)^{2^{s'-s}} &\equiv (-1)^{2^{s'-s}} \pmod{p}; \\ (b^{2^{s'-1}t'})^t &\equiv 1 \pmod{p},\end{aligned}$$

jadi

$$\begin{aligned}\left(\frac{b}{p}\right) &\equiv b^{(p-1)/2} \pmod{p} \text{ karena } p \text{ prima} \\ &\equiv b^{2^{s'-1}t'} \pmod{p} \\ &\equiv 1 \pmod{p} \text{ (tidak mungkin } -1 \text{ karena } (b^{2^{s'-1}t'})^t \equiv 1).\end{aligned}$$

Kita tulis n sebagai produk dari bilangan-bilangan prima (tidak harus semua berbeda):

$$n = \prod_{i=1}^{j} p_i.$$

Jika k merupakan banyaknya p_i yang menghasilkan $s' = s$ jika kita tulis $p_i - 1 = 2^{s'}t'$ dengan t' berupa bilangan ganjil, maka

$$\begin{aligned}\left(\frac{b}{n}\right) &= \prod_{i=1}^{j}\left(\frac{b}{p_i}\right) \\ &= (-1)^k.\end{aligned}$$

Karena $p_i = 1 + 2^{s'}t'$, maka

$$p_i = \begin{cases} 1 + 2^s t' \equiv 1 + 2^s \pmod{2^{s+1}} & \text{jika } s' = s; \\ 1 + 2^{s'}t' \equiv 1 \pmod{2^{s+1}} & \text{jika } s' > s. \end{cases}$$

Karena $n = 1 + 2^s t = \prod_{i=1}^{j} p_i$, maka

$$\begin{aligned} 1 + 2^s t &= \prod_{i=1}^{j} p_i; \\ 1 + 2^s &\equiv (1 + 2^s)^k \pmod{2^{s+1}} \\ &\equiv 1 + k2^s + \ldots \pmod{2^{s+1}} \end{aligned}$$

dimana ... merepresentasikan suku-suku yang dapat dibagi oleh 2^{s+1}. Jadi k harus berupa bilangan ganjil, akibatnya

$$\left(\frac{b}{n}\right) = (-1)^k = -1.$$

Jadi kita sudah tunjukkan bahwa jika persamaan 13.9 berlaku dengan $r = s - 1$ maka persamaan 13.7 juga berlaku. Terahir, jika persamaan 13.9 berlaku dengan $r = r'$, dimana $r' < s - 1$, maka

$$\begin{aligned} b^{2^{s-1}t} &= (b^{2^{r'}t})^{2^{s-1-r'}} \\ &\equiv (-1)^{2^{s-1-r'}} \pmod{n} \\ &\equiv 1 \pmod{n}, \end{aligned}$$

dan

$$b^{(n-1)/2} = b^{2^{s-1}t} \equiv 1 \pmod{n},$$

jadi kita ingin tunjukkan bahwa $\left(\frac{b}{n}\right) = 1$. Jika p adalah bilangan prima ganjil yang membagi n, kita dapat uraikan $p - 1 = 2^{s'}t'$ dimana t' adalah bilangan ganjil. Kita ingin tunjukkan bahwa $s' \geq r' + 1$ dan

$$\left(\frac{b}{p}\right) = \begin{cases} -1, & \text{jika } s' = r' + 1; \\ 1 & \text{jika } s' > r' + 1. \end{cases}$$

Karena $b^{2^{r'}t} \equiv -1 \pmod{n}$, kita pangkatkan kedua sisi persamaan dengan t' yang merupakan bilangan ganjil dan mendapatkan:

$$\begin{aligned} (b^{2^{r'}t})^{t'} &\equiv (-1)^{t'} \pmod{n}; \\ (b^{2^{r'}t'})^{t} &\equiv -1 \pmod{n}. \end{aligned}$$

Karena $p|n$ maka

$$(b^{2^{r'}t'})^t \equiv -1 \pmod{p}.$$

Jika $s' < r' + 1$, maka

$$b^{p-1} = b^{2^{s'}t'} \not\equiv 1 \pmod{p},$$

sesuatu yang bertentangan dengan *Fermat's little theorem* (teorema 30). Jadi $s' \geq r' + 1$. Jika $s' = r' + 1$, maka

$$\begin{aligned}\left(\frac{b}{p}\right) &\equiv b^{(p-1)/2} \pmod{p} \text{ karena } p \text{ prima} \\ &\equiv b^{2^{s'-1}t'} \pmod{p} \\ &\equiv b^{2^{r'}t'} \pmod{p} \\ &\equiv -1 \pmod{p} \text{ (tidak mungkin 1 karena } (b^{2^{r'}t'})^t \equiv -1 \pmod{p}).\end{aligned}$$

Jika $s' > r' + 1$, maka

$$\begin{aligned}((b^{2^{r'}t'})^t)^{2^{s'-r'-1}} &\equiv (-1)^{2^{s'-r'-1}} \pmod{p}; \\ (b^{2^{s'-1}t'})^t &\equiv 1 \pmod{p},\end{aligned}$$

jadi

$$\begin{aligned}\left(\frac{b}{p}\right) &\equiv b^{(p-1)/2} \pmod{p} \text{ karena } p \text{ prima} \\ &\equiv b^{2^{s'-1}t'} \pmod{p} \\ &\equiv 1 \pmod{p} \text{ (tidak mungkin } -1 \text{ karena } (b^{2^{s'-1}t'})^t \equiv 1 \pmod{p}).\end{aligned}$$

Kita tulis n sebagai produk dari bilangan-bilangan prima (tidak harus semua berbeda):

$$n = \prod_{i=1}^{j} p_i.$$

Jika k merupakan banyaknya p_i yang menghasilkan $s' = r' + 1$ jika kita tulis $p_i - 1 = 2^{s'}t'$ dengan t' berupa bilangan ganjil, maka

$$\begin{aligned}\left(\frac{b}{n}\right) &= \prod_{i=1}^{j}\left(\frac{b}{p_i}\right) \\ &= (-1)^k.\end{aligned}$$

Karena $p_i = 1 + 2^{s'}t'$, maka

$$p_i = \begin{cases} 1 + 2^{r'+1}t' \equiv 1 + 2^{r'+1} \pmod{2^{r'+2}} & \text{jika } s' = r' + 1; \\ 1 + 2^{s'}t' \equiv 1 \pmod{2^{r'+2}} & \text{jika } s' > r' + 1. \end{cases}$$

Karena $n = 1 + 2^s t = \prod_{i=1}^{j} p_i$, maka

$$1 + 2^s t = \prod_{i=1}^{j} p_i;$$

$$\begin{aligned} 1 &\equiv (1+2^{r'+1})^k \pmod{2^{r'+2}} \\ &\equiv 1+k2^{r'+1}+\ldots \pmod{2^{r'+2}} \end{aligned}$$

dimana ... merepresentasikan suku-suku yang dapat dibagi oleh $2^{r'+2}$. Jadi k harus berupa bilangan genap, akibatnya

$$\left(\frac{b}{n}\right) = (-1)^k = 1.$$

Jadi kita sudah tunjukkan bahwa jika persamaan 13.9 berlaku dengan $r = r' < s-1$ maka persamaan 13.7 juga berlaku. Selesailah pembuktian teorema 94. Sebelum kita bahas teorema mengenai probabilitas *strong pseudoprime*, akan kita buktikan lebih dahulu dua teorema yang akan digunakan.

Teorema 95 *Terdapat $d = \gcd(k, m)$ elemen dalam group $\{g, g^2, g^3, \ldots, g^m = 1\}$ (g merupakan generator) yang mematuhi persamaan $x^k = 1$.*

Untuk membuktikan teorema 95, kita mengetahui bahwa elemen g^j (dengan $1 \le j \le m$) mematuhi persamaan diatas jika dan hanya jika $g^{jk} = 1$, dengan kata lain jika dan hanya jika $m|jk$, jadi

$$g^{jk} = 1 \iff m|jk \iff \frac{m}{d}|j\frac{k}{d}.$$

Karena $\frac{m}{d}$ koprima dengan $\frac{k}{d}$, maka

$$g^{jk} = 1 \iff \frac{m}{d}|j \ (j \text{ merupakan kelipatan } \frac{m}{d}).$$

Terdapat d bilangan $i\frac{m}{d}$ dimana $1 \le i\frac{m}{d} \le m$:

$$\frac{m}{d}, \frac{2m}{d}, \frac{3m}{d}, \ldots, \frac{dm}{d}.$$

Selesailah pembuktian teorema 95.

Teorema 96 *Jika p adalah bilangan prima ganjil, dan kita tuliskan $p-1 = 2^{s'}t'$ dimana t' adalah bilangan ganjil, maka banyaknya elemen $x \in (\mathbf{Z}/p\mathbf{Z})^*$ yang mematuhi persamaan*

$$x^{2^r t} \equiv -1 \pmod{p},$$

dimana t adalah bilangan ganjil, adalah 0 jika $r \ge s'$, atau $2^r \gcd(t, t')$ jika $r < s'$. Dalam notasi formal:

$$\sharp\{x \in (\mathbf{Z}/p\mathbf{Z})^* | x^{2^r t} \equiv -1 \pmod{p}\} = \begin{cases} 0 & \text{jika } r \ge s', \\ 2^r \gcd(t, t') & \text{jika } r < s'. \end{cases}$$

Untuk membuktikan teorema 96, jika g adalah *generator* untuk $(\mathbf{Z}/p\mathbf{Z})^*$, maka $x = g^j$ untuk suatu j dengan $0 \leq j < p-1$. Karena $g^{(p-1)/2} \equiv -1 \pmod{p}$ dan $p - 1 = 2^{s'}t'$, maka relasi *congruence* dalam teorema ekuivalen dengan

$$2^r tj \equiv 2^{s'-1}t' \pmod{2^{s'}t'}, \tag{13.10}$$

dimana kita harus mencari j. Persamaan 13.10 mempunyai solusi jika dan hanya jika $2^{s'}t'$ membagi $2^r tj - 2^{s'-1}t'$, jadi harus ada bilangan bulat m dengan

$$\begin{aligned} 2^{s'}t'm &= 2^r tj - 2^{s'-1}t'; \\ m &= 2^{r-s'}\frac{t}{t'}j - \frac{1}{2}. \end{aligned}$$

Jika $r \geq s'$ maka tidak ada bilangan bulat j yang dapat membuat m menjadi bilangan bulat (karena $2^{r-s'}$ adalah bilangan bulat dan t' adalah bilangan ganjil, jadi $2^{r-s'}\frac{t}{t'}j$ tidak mungkin menjadi kelipatan dari $\frac{1}{2}$). Jadi persamaan 13.10 tidak mempunyai solusi jika $r \geq s'$. Jika $r < s'$, kita bagi persamaan 13.10 dengan $2^r d$ dimana $d = \gcd(t, t')$ mendapatkan:

$$\frac{t}{d}j \equiv 2^{s'-r-1}\frac{t'}{d} \pmod{2^{s'-r}\frac{t'}{d}}, \tag{13.11}$$

dimana $\frac{t}{d}$ dan $\frac{t'}{d}$ merupakan bilangan bulat. Karena $\gcd(\frac{t}{d}, 2^{s'-r-1}\frac{t'}{d}) = 1$ maka persamaan 13.11 mempunyai solusi yang unik untuk j. Jadi persamaan 13.10 mempunyai $2^r d$ solusi untuk j. Selesailah pembuktian teorema 96.

Teorema 97 *Suatu bilangan komposit ganjil n adalah strong pseudoprime untuk maksimum 25 persen dari base b dengan $0 < b < n$.*

Pembuktian teorema 97 terdiri dari tiga situasi yang berbeda:

1. Situasi dimana n dapat dibagi oleh kuadrat dari suatu bilangan prima ganjil p ($p^2|n$).

2. Situasi dimana n merupakan produk dari dua bilangan prima ganjil p dan q yang berbeda ($n = pq$).

3. Situasi dimana n merupakan produk dari tiga atau lebih bilangan prima ganjil yang berbeda ($n = p_1 p_2 \ldots p_k$, $k \geq 3$).

Jika p adalah suatu bilangan prima ganjil dan $p^2|n$, kita ingin tunjukkan bahwa n merupakan *strong pseudoprime* untuk tidak lebih dari $(n-1)/4$ *base* b, dengan $0 < b < n$. Ini kita lakukan dengan menunjukkan bahwa n merupakan *pseudoprime* untuk tidak lebih dari $(n-1)/4$ *base* b (ingat bahwa suatu *strong pseudoprime* juga merupakan *Euler pseudoprime* menurut teorema 94,

jadi menurut teorema 93 juga merupakan *pseudoprime*). Teorema 38 mengatakan bahwa $(\mathbf{Z}/p^2\mathbf{Z})^*$ merupakan suatu *cyclic group*, jadi terdapat *generator* g dimana

$$(\mathbf{Z}/p^2\mathbf{Z})^* = \{g, g^2, g^3, \ldots g^{p(p-1)}\}.$$

Menurut teorema 95, banyaknya b dengan $0 \leq b < p^2$ dimana

$$b^{n-1} \equiv 1 \pmod{p^2} \tag{13.12}$$

adalah $d = \gcd(p(p-1), n-1)$. Karena $p|n$ maka $p \not| n-1$, jadi $p \not| d$. Akibatnya $d \leq p-1$, jadi proporsi b dari 1 sampai dengan $n-1$ yang mematuhi persamaan 13.12 adalah

$$\frac{p-1}{p^2-1} = \frac{1}{p+1} \leq \frac{1}{4}.$$

Karena proporsi b dari 1 sampai dengan $n-1$ yang mematuhi $b^{n-1} \equiv 1 \pmod{n}$ tidak lebih dari proporsi b yang mematuhi persamaan 13.12, maka n merupakan *pseudoprime* untuk tidak lebih dari $1/4$ *base* b dengan $0 < b < n$. Selesailah pembuktian untuk situasi $p^2|n$.

Jika $n = pq$, kita tulis $p - 1 = 2^{s'}t'$ dan $q - 1 = 2^{s''}t''$, dengan t', t'' berupa bilangan ganjil dan $s' \leq s''$ (kita dapat memilih p dan q sedemikian rupa). $0 < b < n$ merupakan *base* yang menjadikan n *strong pseudoprime* jika

$$b^t \equiv 1 \pmod{p} \text{ dan } b^t \equiv 1 \pmod{q}$$

atau

$$b^{2^r t} \equiv -1 \pmod{p} \text{ dan } b^{2^r t} \equiv -1 \pmod{q}$$

untuk suatu r dengan $0 \leq r < s$, dimana $n - 1 = 2^s t$ dengan t berupa bilangan ganjil. Banyaknya $0 < b < n$ untuk kemungkinan pertama adalah banyaknya $0 < b < p$ yang mengakibatkan $b^t \equiv 1 \pmod{p}$ dikalikan dengan banyaknya $0 < b < q$ yang mengakibatkan $b^t \equiv 1 \pmod{q}$. Menggunakan teorema 95, banyaknya b untuk kemungkinan pertama adalah

$$\begin{aligned} \gcd(t, p-1)\gcd(t, q-1) &= \gcd(t, 2^{s'}t')\gcd(t, 2^{s''}t'') \\ &= \gcd(t, t')\gcd(t, t'') \\ &\leq t't''. \end{aligned}$$

Untuk setiap r dengan $r < s' \leq s'' \leq s$, banyaknya b yang mematuhi $b^{2^r t} \equiv -1 \pmod{n}$, menggunakan teorema 96, adalah

$$2^r \gcd(t, t')2^r \gcd(t, t'') < 4^r t't''.$$

Karena $n - 1 > \phi(n) = 2^{s'+s"}t't''$, maka proporsi b dalam $0 < b < n$ yang menjadikan n *strong pseudoprime* adalah $\leq$

$$\frac{t't'' + \sum_{r=0}^{s'-1} 4^r t't''}{2^{s'+s''}t't''} = \frac{1 + (1 + 4 + 4^2 + \ldots 4^{s'-1})}{2^{s'+s''}}$$

$$= \quad 2^{-s'-s''}\left(1+\frac{4^{s'}-1}{4-1}\right).$$

Jika $s'' > s'$ maka ini adalah $\leq$

$$\begin{aligned} 2^{-2s'-1}\left(1+\frac{4^{s'}-1}{3}\right) &= 2^{-2s'-1}\left(\frac{2}{3}+\frac{4^{s'}}{3}\right) \\ &\leq 2^{-3}\frac{2}{3}+\frac{1}{6} \\ &= \frac{1}{12}+\frac{1}{6} \\ &= \frac{1}{4}. \end{aligned}$$

Jika $s'' = s'$ maka

$$\gcd(t,t') < t' \text{ atau } \gcd(t,t'') < t''$$

karena jika $t'|t$ dan $t''|t$ maka dari

$$n-1 = 2^s t = pq-1 = q(p-1)+q-1 \equiv q-1 \equiv 0 \pmod{t'}$$

kita dapatkan $t'|q-1$, jadi $t'|t''$, dan dari

$$n-1 = 2^s t = pq-1 = p(q-1)+p-1 \equiv p-1 \equiv 0 \pmod{t''}$$

kita dapatkan $t''|p-1$, jadi $t''|t'$, alhasil $t' = t''$ dan $p = q$. Tetapi ini bertentangan dengan asumsi $p \neq q$, jadi

$$\gcd(t,t') < t' \text{ atau } \gcd(t,t'') < t''.$$

Jika $\gcd(t,t') < t'$ maka $\gcd(t,t') \leq \frac{1}{3}t'$ karena gcd membagi t', lebih kecil dari t' dan merupakan bilangan ganjil. Demikian juga jika $\gcd(t,t'') < t''$ maka $\gcd(t,t'') \leq \frac{1}{3}t''$. Jadi proporsi b dalam $0 < b < n$ yang menjadikan n *strong pseudoprime* adalah $\leq$

$$\begin{aligned} \frac{t't''+\sum_{r=0}^{s'-1}4^r t't''}{3(2^{2s'}t't'')} &= \frac{1}{3}2^{-2s'}\left(1+\frac{4^{s'}-1}{4-1}\right) \\ &\leq \frac{1}{18}+\frac{1}{9} \\ &= \frac{1}{6} \\ &< \frac{1}{4}. \end{aligned}$$

Selesailah pembuktian untuk situasi $n = pq$.

Untuk $n = p_1 p_2 \dots p_k$, $k \geq 3$, dan $p_i \neq p_j$ jika $i \neq j$, kita tulis

$$p_i - 1 = 2^{s_i} t_i$$

dengan t_i bilangan ganjil untuk $1 \leq i \leq k$. Kita pilih urutan p_i sedemikian rupa sehingga $s_1 \leq s_i$ untuk $2 \leq i \leq k$. Ini merupakan generalisasi dari situasi $n = pq$, dan proporsi b dalam $0 < b < n$ yang menjadikan n *strong pseudoprime* adalah $\leq$

$$\begin{aligned} 2^{-s_1-s_2-\dots-s_k}\left(1+\frac{2^{ks_1}-1}{2^k-1}\right) &\leq 2^{-ks_1}\left(\frac{2^k-2}{2^k-1}+\frac{2^{ks_1}}{2^k-1}\right) \\ &= 2^{-ks_1}\frac{2^k-2}{2^k-1}+\frac{1}{2^k-1} \\ &\leq 2^{-ks}\frac{2^k-2}{2^k-1}+\frac{1}{2^k-1} \\ &= 2^{1-k} \end{aligned}$$

jadi $\leq \frac{1}{4}$ untuk $k \geq 3$. Selesailah pembuktian teorema 97. Teorema 97 menjadi dasar dari algoritma Miller-Rabin dengan $n-1 = 2^s t$ dimana t adalah bilangan ganjil:

1. Pilih suatu bilangan b secara acak sebagai *base* dimana $0 < b < n$.

2. Kalkulasi

   $$b_0 = b^t \pmod{n}, b_1 = b_0^2 \pmod{n}, \dots b_k = b_{k-1}^2 \pmod{n}$$

   hingga $k = s$ atau $b_k \equiv 1 \pmod{n}$.

3. Jika $k = s$ dan $b_k \not\equiv 1 \pmod{n}$ maka n adalah bilangan komposit dan kita selesai.

4. Jika $k \neq 0$ dan $b_{k-1} \not\equiv -1 \pmod{n}$ maka n adalah bilangan komposit dan kita selesai.

Jika $b^t \equiv 1 \pmod{n}$ maka $k = 0$ dan langkah 3 dan langkah 4 tidak menghasilkan jawaban komposit. Jika terdapat r dengan $0 \leq r < s$ dimana $b^{2^r t} \equiv -1 \pmod{n}$ maka $k = r + 1$ dan langkah 3 dan langkah 4 tidak menghasilkan jawaban komposit. Jika $b^t \not\equiv 1 \pmod{n}$ dan tidak ada r dengan $0 \leq r < s$ dimana $b^{2^r t} \equiv -1 \pmod{n}$ maka

- $k = s$ dan $b^k \not\equiv 1 \pmod{n}$, jadi langkah 3 menghasilkan jawaban komposit; atau

- $k \neq s$, $b^k \equiv 1 \pmod{n}$ tetapi $b^{k-1} \not\equiv -1 \pmod{n}$, jadi langkah 4 menghasilkan jawaban komposit.

Jadi jika langkah-langkah diatas menghasilkan jawaban komposit, maka kita tahu dengan pasti bahwa n adalah bilangan komposit. Jika tidak menghasilkan jawaban komposit, maka probabilitas bahwa n adalah bilangan komposit adalah $\leq \frac{1}{4}$. Jadi jika langkah-langkah diatas diulang sebanyak k kali tanpa jawaban komposit[3], setiap kali dengan *base* baru yang dipilih secara acak, maka probabilitas bahwa n adalah bilangan komposit adalah $\leq \frac{1}{4^k}$, dan probabilitas bahwa n merupakan bilangan prima adalah $\geq 1 - \frac{1}{4^k}$. Algoritma Miller-Rabin dianggap lebih baik dari algoritma Solovay-Strassen karena

- algoritma Solovay-Strassen lebih sukar untuk diprogram karena perlu kalkulasi dengan simbol Jacobi;
- probabilitas jawaban yang salah jika n adalah bilangan komposit lebih kecil dengan algoritma Miller-Rabin; dan
- jika algoritma Solovay-Strassen memberi jawaban komposit untuk suatu *base* b, maka algoritma Miller-Rabin juga memberikan jawaban komposit, sedangkan sebaliknya tidak selalu.

13.4 Test Deterministik

Jika Extended Riemann Hypothesis (ERH)[4] benar, maka algoritma Solovay-Strassen atau algoritma Miller-Rabin dapat digunakan untuk test bilangan prima secara deterministik dengan melakukan test untuk setiap *base* b:

$$2 \leq b \leq 2(\log n)^2.$$

Juga terdapat algoritma Cohen-Lenstra untuk test bilangan prima secara deterministik (lihat [coh84]) yang, walaupun secara teoritis bukan *order polynomial* dalam $\log n$, tetapi dalam prakteknya dapat melakukan test bilangan prima untuk bilangan dengan ratusan *digit* dalam hitungan detik dengan komputer masa kini.

Ada juga algoritma deterministik yang dapat dengan cepat menentukan bahwa bilangan adalah prima jika memang benar bilangan prima. Algoritma yang banyak digunakan pertama dikembangkan oleh Goldwasser dan Kilian (lihat [gol86]), dan kemudian dibuat lebih efisien Atkin dan diimplementasi oleh Atkin dan Morain (lihat [atk93]). Algoritma yang didasarkan pada *elliptic curve* ini bisa digunakan untuk memastikan bahwa suatu bilangan yang telah

[3]Tentunya sekali jawaban komposit ditemukan, tidak ada gunanya untuk mengulang kembali langkah-langkah algoritma Miller-Rabin.

[4]Kebenaran dari ERH belum pernah dibuktikan.

ditest oleh algoritma probabilistik (contohnya Miller-Rabin) memang benar prima. Algoritma ini dijuluki Atkin-Goldwasser-Kilain-Morain *certificate*.

13.5 Ringkasan

Bab ini telah membahas test bilangan prima, topik yang sangat penting untuk kriptografi *public key*. Meskipun telah ditemukan algoritma test bilangan prima yang bersifat deterministik dengan kompleksitas *polynomial*, dalam prakteknya algoritma yang bersifat probabilistik jauh lebih cepat. Algoritma probabilistik (Monte Carlo) untuk test bilangan prima menggunakan konsep *pseudoprime* dan dua diantaranya adalah algoritma Solovay-Strassen [sol77] dan algoritma Miller-Rabin [rab80]. Algoritma Miller-Rabin secara umum dianggap lebih baik dibandingkan algoritma Solovay-Strassen. Untuk memastikan bahwa bilangan yang telah ditest menggunakan algoritma probabilistik benar prima, algoritma Atkin-Goldwasser-Kilain-Morain dapat digunakan.