Jurnal Teknologi, 35(C) Dis. 2001: 61–70

KRIPTOSISTEM MULTI-RSA

HOW GUAN AUN[1], YAHYA ABU HASAN[2] & EDDIE SHAHRIL ISMAIL[3]

Abstrak. Dalam kertas ilmiah ini kami persembahkan kriptosistem multi-RSA. Sistem baru berasaskan RSA ini membenarkan sebarang kombinasi k+1 pelanggan menyahsulit tetapi kurang daripada k+1 pelanggan gagal berbuat demikian. Hal ini dapat dicapai dengan memperkenalkan fungsi aritmetik istimewa yang menjamin ketepatan algoritma penyahsulitan sistem kami.

Kata kunci: Kriptografi, kriptosistem multi-RSA, penyulitan dan penyahsulitan

Abstract. In this paper we present a multi-RSA cryptosystem. This new system based on RSA allows any k+1 parties to decrypt but less than k+1 parties cannot do so. All these happened by introducing a special arithmetic function, which ensures the exactness of our decryption algorithm.

Keywords: Cryptography, multi-RSA cryptosystem, encryption and decryption

1.0 PENGENALAN

Kriptosistem RSA ciptaan Rivest *et al*. [1] merupakan aplikasi terhebat yang terbit daripada cetusan idea kriptografi kunci awam. Hingga kini sistem tersohor ini yang keselamatannya dipercayai bergantung penuh pada pemfaktoran integer besar masih kekal selamat (walaupun hal ini masih dibahas). Namun dalam sistem ini, hanya seorang pelanggan sahaja cukup untuk melakukan penyahsulitan (*decryption*). Penyahsulitan adalah proses menukarkan maklumat rahsia (teks rahsia) kepada maklumat asal (teks asli). Justeru itu kami ingin mencipta sistem yang hanya membenarkan bilangan tertentu pelanggan melakukan penyahsulitan ke atas teks rahsia secara bersama. Lantaran itu sewajarnyalah multi-RSA diperkenalkan. Antara ciri-ciri utama sistem baru ini ialah ia membenarkan subset yang layak (mengandungi k+1 pelanggan) melakukan penyahsulitan secara bersama tetapi subset yang tidak layak (mengandungi kurang k+1 pelanggan) tidak dibenarkan. Sistem ini seterusnya memenuhi kehendak dan sifat-sifat skim mengongsi rahsia (*secret sharing*) yang ideanya telah lama diperkenalkan oleh Shamir [2].

Sebenarnya telah banyak sistem bervariasi RSA dicipta bertunjangkan skim mengongsi rahsia ini. Pembaca yang berminat boleh rujuk [3–5]. Bagaimanapun dalam sistem kami ini, seorang ketua yang merupakan satu pihak yang jujur (*a trusted party*) diperlukan untuk menjana kunci-kunci rahsia (digunakan semasa proses penyahsulitan) untuk diagihkan kepada pelanggan-pelanggan sistem. Kunci-kunci

[1,2&3] Pusat Pengajian Sains Matematik, Universiti Sains Malaysia, 11800 Minden, Pulau Pinang.

awam (digunakan semasa proses penyulitan (*encryption*)) juga dijana oleh ketua yang jujur. Penyulitan adalah proses bertentangan dengan penyahsulitan. Hari ini penggunaan ketua yang jujur adalah sesuatu yang tradisi, disebabkan sejak kebelakangan ini telah banyak sistem mengongsi rahsia dicipta tanpa memerlukan khidmat ketua yang jujur. Bagaimanapun sistem kami ini agak sedikit berbeza. Ketua yang jujur adalah salah seorang pelanggan dalam sistem dan wajib berada dalam sebarang kombinasi k+1 pelanggan tadi. Dengan kata lain sebarang proses penyahsulitan harus disertai oleh ketua yang jujur.

Dalam kertas ini, kami perkenalkan satu fungsi aritmetik istimewa. Fungsi ini amatlah penting terutama di dalam proses penyahsulitan dan merupakan nadi keseluruhan sistem multi-RSA.

2.0 PEMBINAAN MULTI-RSA

Misalkan sistem mengandungi n pelanggan dan tandakan mereka sebagai ahli dalam set $P = (P_1, \ldots, P_n)$. Tandakan $Q = (P_1, P_{i1}, \ldots, P_{ik})$ sebagai sebarang subset bagi P yang mengandungi kombinasi k+1 pelanggan dan P_1 kami tandakan sebagai ketua yang jujur itu. Sekarang kami perlihatkan kaedah penjanaan kunci-kunci awam dan rahsia yang perlu dilakukan oleh ketua yang jujur.

2.1 Menjana Kunci-Kunci Awam dan Rahsia

Ketua yang jujur memilih dua nombor perdana besar yang berbeza p dan q. Pemilihan ini diharapkan sama seperti pemilihan nombor perdana dalam RSA supaya keselamatannya terjamin. Pembaca boleh rujuk Koblitz [6] sebagai lanjutan.

Kemudian ketua hitung integer $N = pq$ dan seterusnya fungsi *phi* Euler $\phi(N) = (p-1)(q-1)$. Nombor $\phi(N)$ memberitahu bilangan integer antara 1 dan N yang perdana secara relatif terhadap N. Seterusnya ketua jana integer-integer positif r dan s dalam $Z_{\phi(N)}\backslash\{0\}$ dengan sifat:

$$kr + s \equiv 1 \bmod \phi(N) \tag{2.11}$$

dan selepas itu pilih satu lagi integer d dalam $Z_{\phi(N)}\backslash\{0\}$. Selepas itu ketua hitung kunci-kunci awam $e_i < \phi(N)$ yang perdana secara relatif terhadap $\phi(N)$ dan untuk setiap dua kunci awam yang berbeza, pembahagi sepunya terbesar haruslah bukan satu. Setelah itu $(e_1, \ldots, e_n, N)$ akan disenaraikan dalam fail awam yang boleh dicapai oleh setiap penghantar maklumat atau sesiapa sahaja. Setelah itu ketua mula hitung kunci-kunci rahsia pelanggan melalui kongruen (2.12) dan kunci rahsia ketua itu sendiri melalui kongruen (2.13):

$$e_i(d_i + d) \equiv r \bmod \phi(N) \tag{2.12}$$

$$e_i d_1 \equiv s \bmod \phi(N) \tag{2.13}$$

Kunci-kunci d_i yang telah dijana diagihkan kepada P_i masing-masing dan dua kunci rahsia d_i dan d akan dipegang oleh ketua yang jujur. Adalah diharapkan setiap pelanggan kecuali ketua yang jujur mengetahui satu dan hanya satu kunci rahsia sahaja. Untuk memahami dengan lebih jelas proses penjanaan kunci-kunci awam dan rahsia pelanggan, lihat Rajah 1 dalam Lampiran A.

2.2 Penyulitan dan Penyahsulitan

Misalkan seorang penghantar maklumat Hakim ingin menghantar teks asli yang telah dikod sebagai integer m antara 1 dan N. Integer m haruslah bukan perdana secara relatif terhadap N. Hal ini dituntut untuk mengelak pencuri rahsia, penyamar atau penyerang sistem memperoleh m dengan mudah.

Hakim seterusnya membuka fail awam untuk mendapatkan kunci awam setiap pelanggan. Untuk penyulitan, Hakim mengkuasakan m kepada e_i modulo N dan hasilnya adalah baki apabila m^{e_i} dibahagi oleh N. Berikut adalah algoritma penyulitan sistem kami:

$$E(m) = (m^{e_i}, \ldots, m^{e_n}) \bmod N = (c_1, \ \ldots, c_n) \tag{2.21}$$

untuk suatu teks asli m dan E menandakan fungsi penyulitan. Hakim kemudian hantar teks rahsia itu kepada setiap pelanggan dalam P. Untuk penyahsulitan, setiap pelanggan dalam Q mengkuasakan c_i kepada d_i modulo N dan ketua yang jujur mengkuasakan c_1 kepada d_1 modulo N dan $c_i\,(i \neq 1)$ kepada d modulo N. Berikut pula adalah algoritma penyahsulitan sistem kami:

$$D(c_1, c_{i1}, \ldots, c_{ik}) = (c_1^{d_1}) \prod_{t=1}^{k} (c_{it}^{d})(c_{it}^{d_{it}}) \bmod N \tag{2.22}$$

dan D menandakan fungsi penyahsulitan. Kongruen (2.22) jelas menunjukkan bahawa ketua perlu berada di dalam subset yang menyahsulit. Untuk memahami secara jelas perjalanan kedua-dua proses penyulitan dan penyahsulitan, sila lihat Rajah 2 dalam Lampiran A. Rajah ini menunjukkan gambaran piawai proses penyulitan dan penyahsulitan di dalam sebarang kriptosistem kunci awam.

2.3 Ketepatan Algoritma Penyahsulitan

Tulang belakang kepada sesuatu kriptosistem adalah ketepatan algoritma penyahsulitannya. Ketepatan algoritma ini membolehkan pelanggan-pelanggan dalam Q memahami teks rahsia hantaran Hakim. Untuk tujuan itu kami perkenalkan fungsi aritmetik istimewa yang sedikit berbeza daripada fungsi-fungsi aritmetik lain dalam teori nombor. Fungsi aritmetik istimewa ini kami berikan dalam dua bentuk yang berlainan.

Takrif 2.1. Misalkan $\omega \geq 3$ integer dan p perdana. Takrifkan fungsi β sebagai

$$\beta(\omega) = \prod_{p-1|\omega-1} p.$$

Simbol '|' di situ menandakan $p-1$ membahagi $\omega - 1$. Sekarang kami perlihatkan teorem berikut. Teorem ini banyak membantu kerja-kerja kami selepas ini. Pembuktian teorem ini boleh diperoleh dalam LAMPIRAN B.

Teorem 2.1. Misalkan $\omega \geq 3$ integer dan N set nombor asli. Maka

$$\beta(\omega) = maks\{x \in N \mid a^{\omega} \equiv a \bmod x, \forall a\}.$$

Sekarang kami tunjukkan ketepatan algoritma penyahsulitan. Daripada (2.22) diperoleh

$$D(c_1, c_{i1}, ..., c_{ik}) = \left(c_1^{d_1}\right)\prod_{t=1}^{k}\left(c_{it}^{d}\right)\left(c_{it}^{d_{it}}\right) \bmod N$$

$$= \left(m^{e_1 d_1}\right)\prod_{t=1}^{k}\left(m^{e_{it} d}\right)\left(m^{e_{it} d_{it}}\right) \bmod N = m^{e_1 d_1 + \sum_{t=1}^{k} e_{it} d + e_{it} d_{it}} \bmod N \quad (2.31)$$

Kami perlu tunjukkan bahawa (2.31) adalah kongruen m modulo N. Pertimbangkan sebarang subset Q yang mengandungi k+1 pelanggan. Dengan menggunakan operasi biasa penambahan modular terhadap (2.12) untuk setiap pelanggan dalam $Q\backslash\{P_1\}$ dengan (2.13) maka diperoleh

$$e_1 d_1 + \sum_{t=1}^{k} e_{it} d + e_{it} d_{it} \equiv kr + s \bmod \phi(N). \quad (2.32)$$

Seterusnya dari pada (2.11) diperoleh

$$e_1 d_1 + \sum_{t=1}^{k} e_{it} d + e_{it} d_{it} \equiv 1 \bmod \phi(N) \quad (2.33)$$

atau setaranya ditulis

$$e_1 d_1 + \sum_{t=1}^{k} e_{it} d + e_{it} d_{it} = v\phi(N) + 1 = v(p-1)(q-1) + 1 \quad (2.34)$$

untuk suatu integer v. Dengan mengambil $\omega = e_1 d_1 + \sum_{t=1}^{k} e_{it} d + e_{it} d_{it}$ serta berbekalkan

fakta yang $p-1$ dan $q-1$ membahagi $\omega-1$ dan melalui Takrif 2.1 maka diperoleh

$$\beta(\omega) = pqu = Nu \tag{2.35}$$

untuk suatu integer u. Oleh kerana $\beta(\omega) \in A_\omega$ dan N membahagi $\beta(\omega)$ dan sesuai dengan Teorem 2.1 (3) (lihat LAMPIRAN B) maka $N \in A_\omega$. Oleh itu,

$$a^\omega \equiv a \bmod N \tag{2.36}$$

untuk setiap a. Gantikan $a = m$ dan akhirnya diperoleh

$$m^\omega \equiv a \bmod N \tag{2.37}$$

3.0 CONTOH KECIL

Sejak diperkenalkan pada pertengahan 70-an, kriptosistem kunci awam telah banyak menyelesaikan masalah-masalah di dalam ketenteraan, perbankan, komputeran, tandatangan digital, kad pintar dan dalam setiap situasi yang melibatkan keselamatan terhadap sesuatu maklumat.

Sekarang, andaikan sebuah bank memiliki enam eksekutif. Bank itu saban hari menerima pelbagai maklumat sulit yang patut dilindungi. Maklumat sulit itu pula harus 'dibuka' untuk proses-proses seterusnya. Untuk tujuan ini, pengurus bank itu boleh memilih sistem multi-RSA dan ahli-ahli sistem pula terdiri daripada enam eksekutif dan pengurus itu sendiri. Di sini pengurus bertindak seperti ketua yang jujur. Andaikan pengurus itu memilih parameter-parameter sistem seperti berikut:

$$p = 359,\ q = 263,$$

$$N = 94417,\ \phi(94417) = 93796,$$

$$k = 3,\ r = 31200,\ s = 197 \text{ dan } d = 4532.$$

Berdasarkan pemilihan tersebut, sebarang kombinasi empat ahli boleh melakukan penyahsulitan tetapi kurang daripada empat ahli gagal melakukannya. Kunci-kunci awam dan rahsia yang telah dijana ditunjukkan dalam Jadual 1.

Jadual 1 Kunci-kunci Awam dan Rahsia Pengurus dan Eksekutif-eksekutif

i	00001	00002	00003	00004	00005	00006	00007
e_i	00003	00021	00027	00033	00045	00051	00063
d_i	31331	37152	69576	39048	58692	12632	40628

Misalkan Hakim ingin menghantar maklumat GUA kepada ahli dalam P. Sebelum melakukan penyulitan Hakim mesti kodkan dahulu GUA kepada integer. Beliau boleh menggunakan mana-mana kaedah pengkodan. Katakan sistem pengkodan

yang beliau pilih hanya mengandungi huruf-huruf biasa A hingga Z yang diumpukkan kepada integer antara 1 hingga N seperti yang ditunjukkan dalam Jadual 2.

Jadual 2 Pengkodan Abjad-abjad kepada Integer-integer Positif

A	B	...	G	...
(01)(263)	(02)(263)		(07)(263)	
...	U	...	Z	–
	(21)(263)		(26)(263)	–

Berdasarkan jadual itu, GUA dikodkan sebagai [1841][5523][0263]. Setiap nombor dalam kurungan akan melalui proses penyulitan. Kami hanya tunjukkan proses penyulitan untuk integer 1841 sahaja. Hakim melihat fail awam senarai kunci-kunci awam pengurus dan enam eksekutif tersebut. Setelah itu algoritma penyulitan dilakukan seperti berikut:

$$\begin{aligned} E(1841) &= (1841^3, 1841^{21}, 1841^{27}, 1841^{33}, 1841^{45}, 1841^{51}, 1841^{63}) \bmod 94417 \\ &= (24459, 87842, 16043, 32612, 79952, 54178, 74955) \end{aligned}$$

Teks rahsia ini seterusnya akan dihantar kepada semua ahli sistem itu. Untuk memahami teks rahsia itu, tidak perlu kesemua tujuh ahli itu bersatu, memadai hanya empat ahli sahaja. Misalkan pelanggan dalam $Q = (P_1, P_2, P_4, P_5)$ ingin mengetahui kandungan yang tersirat dalam teks rahsia itu. Jadi mereka haruslah melakukan algoritma penyahsulitan seperti berikut:

$$\begin{aligned} &D(24459, 87842, 32612, 79952) \\ &= (24459^{31331})(87842^{4532}32612^{4532}79952^{4532})(87842^{37152}32612^{39048}79952^{58692}) \\ &\quad \bmod 94417 \\ &= 1841 \bmod 94417 \end{aligned}$$

dan seterusnya diperoleh G. Dengan cara yang sama GUA akhirnya akan diperoleh.

4.0 PERBINCANGAN

Dalam sistem multi-RSA ini hanya p, q, r dan s harus dirahsiakan daripada pengetahuan umum. Ini dipercayai dapat menghalang individu luar mahupun pelanggan-pelanggan sistem daripada menyerang sistem dengan mudah. Namun dengan mengetahui dua daripada empat integer itu sudah memadai untuk sesiapa membongkar sistem ini.

Dalam aplikasi atau realiti dunia sebenar, perdana p dan q haruslah besar contohnya bersaiz 300 digit seperti yang disyorkan dalam RSA. Saiz perdana sebegini memerlukan masa yang tidak rasional untuk memfaktorkan N. Inilah antara penyebab mengapa RSA masih kekal selamat hingga kini. Dalam multi-RSA ini, kunci-kunci awam yang dijana haruslah bukan perdana. Tuntutan ini perlu kerana jika ianya perdana dikhuatiri individu luar dapat mencari kunci-kunci rahsia dengan hanya menggunakan algoritma Euklidan terpeluas yang terkenal itu.

Kami percaya bahawa tiada pelanggan dapat menjana kunci rahsia pelanggan lain dengan hanya mengetahui kunci awam dan N. Malahan untuk mengelak masalah yang sama, dalam mana-mana jurnal atau buku berkenaan serangan ke atas RSA seperti [7], mereka tidak menggalakkan penggunaan N yang sama untuk setiap pelanggan. Bagaimanapun hal ini dapat kami tangkis dengan meletakkan syarat bahawa setiap dua kunci awam yang berbeza haruslah tidak perdana secara relatif. Perhatikan juga bahawa kami tidak mengagihkan kunci rahsia pelanggan sebagai $d_i + d$ sebaliknya hanya mengagihkan mereka d_i sahaja. Ini dapat menghalang dua atau lebih pelanggan daripada menyerang sistem. Namun kaedahnya tidak akan ditunjukkan di sini.

Pertimbangkan pula situasi berikut: Suatu maklumat rahsia hendak dikongsi bersama oleh firma A dan firma B. Untuk itu firma A dan B masing-masing menghantar g dan h eksekutif-eksekutifnya dan seterusnya mengetahui rahsia itu. Situasi ini boleh direalisasikan dengan melakukan pembaikan ke atas kongruen (2.11). Ini merupakan salah satu perluasan yang dapat dilakukan ke atas sistem multi-RSA ini. Di samping itu seseorang mungkin terfikir untuk menggunakan hanya satu kunci awam sahaja untuk proses penyulitan. Jika ini dapat dilakukan maka aplikasi seperti tandatangan digital yang sedang hangat di perkatakan sekarang dapat diteroka. Dalam kertas ini, kami tidak buktikan secara terperinci keselamatan sistem kami mahupun algoritma-algoritma berkaitan penjanaan perdana dan kunci-kunci awam dan rahsia yang terlibat.

5.0 KESIMPULAN

Kami telah persembahkan variasi kepada RSA iaitu multi-RSA. Sistem ini mempunyai ciri-ciri seperti penyahsulitan dapat dilakukan oleh k+1 pelanggan dan adalah dipercayai (walaupun belum terbukti) bahawa keselamatan sistem baru ini bergantung penuh pada pemfaktoran integer besar dan pemilihan rawak r dan s. Sistem ini di harapkan seteguh dengan mana-mana sistem yang telah diketahui dari segi keselamatan dan kecekapannya.

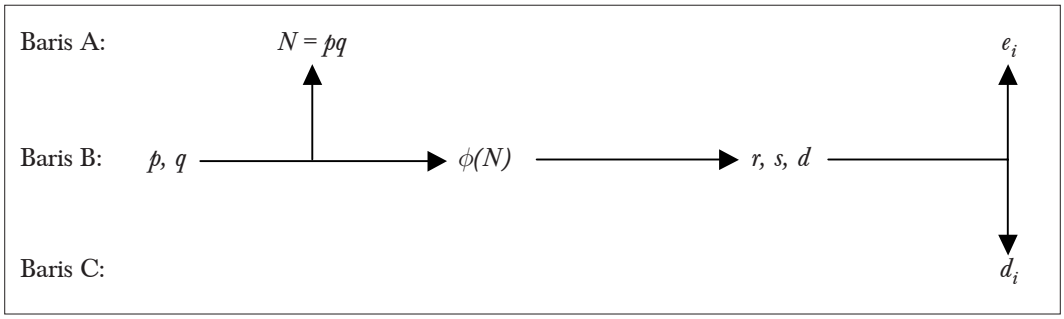
RUJUKAN

[1] Rivest, R. L., A. Shamir, dan L. Adleman. 1979. A Method for Obtaining Digital Signatures and Public Key Cryptosystems, *Comm. of the ACM*, 22(11): 120–126.

[2] Shamir, A. 1979. How to share a secret, *Comm. of the ACM*, 22(11): 612–613.
[3] Boneh, D., dan M. Franklin. 1997. Efficient generation of shared RSA keys. Dalam *Advances in Cryptology-CRYPTO '97*. Lecture Notes in Computer Science. Springer-Verlag. 424-439.
[4] Cocks, C. 1998. *Split Generation of Rsa Parameters with Multiple Participants*. Muncul dalam laman web www.cesg.gov.uk/downlds/math/rsa2.pdf.
[5] Gennaro, R., S. Jarecki, H. Krawcyzk, dan T. Rabin. 1996. Robust and Efficient Sharing of RSA Functions. Dalam *Advances in Cryptology-CRYPTO '96*. Lecture notes in Computer Science. Springer-Verlag. 157–172.
[6] Koblitz, N. 1987. *A course in Number Theory and Cryptography*. Springer-Verlag.
[7] Schneier, B. 1996. *Applied Cryptography*. John Wiley & Sons, Inc.

LAMPIRAN A

Rajah 1 Aliran ringkas penjanaan parameter-parameter dan kunci-kunci sistem

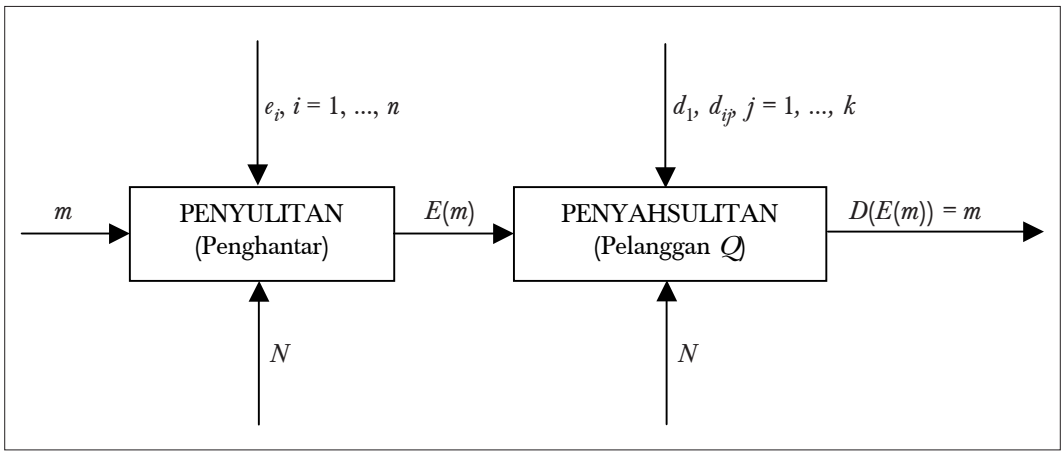


Baris A: Kunci-kunci awam (disenaraikan dalam fail awam)

Baris B: Parameter-parameter sistem (dirahsiakan)

Baris C: Kunci-kunci rahsia (dirahsiakan)

Rajah 2 Ringkasan perjalanan proses-proses penyulitan dan penyahsulitan



LAMPIRAN B

Pembuktian Teorem 2.1

Misalkan $A_\omega = \{x \in N \mid a^\omega \equiv a \bmod x, \forall a\}$. Kami akan buktikan yang berikut:

1. $A_\omega \neq \varnothing$
2. Wujud maksimum A_ω.
3. Jika $x \in A_\omega$ dan y membahagi x maka $y \in A_\omega$.
4. Jika dua perdana berbeza masing-masing unsur dalam A_ω maka hasildarabnya juga unsur dalam A_ω.
5. Setiap unsur A_ω adalah bebas kuasa dua.
6. Untuk p perdana, $p \in A_\omega$ jika dan hanya jika $p - 1$ membahagi $\omega - 1$.

Bukti:

1. Tidak kira sama ada a integer genap atau ganjil, integer $a^\omega - a$ sentiasa genap. Oleh yang demikian ianya sentiasa terbahagi oleh 2, sehinggakan $a^\omega \equiv a$. Jadi $2 \in A_\omega$.
2. Oleh kerana a sebarang integer, ambil $a = 2$. Seseorang beroleh $2^\omega = 2 \bmod x$ yang bermaksud x membahagi $2^\omega - 2$. Jelaslah $2^\omega - 2$ adalah salah satu batas atas bagi yang seterusnya menjamin kewujudan maksimum A_ω.
3. Misalkan $x \in A_\omega$ dan y membahagi x. Oleh yang demikian $a^\omega \equiv a \bmod x$ atau setaranya ditulis $a^\omega - a = xn$ untuk suatu integer n. Akibat daripada y membahagi x, kami peroleh $x = yr$ untuk suatu integer r sehinggakan $a^\omega - a = xn = yrn = ym$ yang m adalah integer. Jelaslah y membahagi $a^\omega - a$ atau setaranya ditulis $a^\omega \equiv a$. Ini seterusnya memberikan $y \in A_\omega$.
4. Misalkan dua perdana berbeza p dan q masing-masing unsur dalam A_ω. Dengan itu kami $a^\omega \equiv a$ peroleh p dan $a^\omega \equiv a \bmod q$ yang masing-masing boleh ditulis sebagai $a^\omega - a = pn = qm$ untuk suatu integer n dan m. Oleh kerana p dan q perdana serta dengan menggunakan Teorem Asas Aritmetik pastilah n mengandungi sebutan q dan m mengandungi sebutan p. Natijahnya diperoleh $a^\omega - a = pqr$ untuk suatu integer r. Jelaslah pq membahagi $a^\omega - a$ sehinggakan $a^\omega \equiv a \bmod pq$ dan seterusnya hasil darab tersebut juga unsur dalam A_ω.
5. Misalkan $x \in A_\omega$. Hendak tunjukkan yang x bebas kuasa dua. Misalkan p^2 membahagi x yang p perdana. Jadi daripada keputusan (3) diperoleh $p^2 \in A_\omega$. Oleh itu $a^\omega \equiv a \bmod p^2$ untuk sebarang integer a. Ambillah $a = p$ sehinggakan $p^\omega \equiv p \bmod p^2$. Oleh kerana $\omega \geq 3$ pastilah $p^\omega \equiv 0 \bmod p^2$ sehinggakan $0 \equiv p \bmod p^2$ yang bermaksud p^2 membahagi p. Hasil ini jelas sesuatu yang mustahil. Oleh yang demikian benarlah setiap unsur dalam A_ω adalah bebas kuasa dua.

   Daripada lima keputusan di atas kami simpul bahawa maksimum A_ω adalah $\prod_{p \in A_\omega} p$ yang p perdana. Bandingkan keputusan ini dengan Takrif 2.1.

Untuk melengkapkan pembuktian Teorem 2.1, kami perlu buktikan (6).

6. Misalkan $p-1$ membahagi $\omega-1$. Oleh itu wujud suatu integer r sehinggakan $\omega-1 = (p-1)r$. Sekarang ambillah sebarang integer a yang perdana secara relatif dengan p. Daripada Teorem Fermat diperoleh $a^{p-1} \equiv 1 \bmod p$. Fakta di atas mencukupi untuk menunjukkan $a^{\omega} \equiv (a^{p-1})^r a \bmod p \equiv (1)^r a \bmod p \equiv a \bmod p$. Natijahnya $p \in A_{\omega}$. Sekarang misalkan pula $p \in A_{\omega}$. Oleh itu p untuk sebarang integer a. Pertimbangkan integer-integer a yang perdana secara relatif dengan p yang menjamin kewujudan sonsangan a. Oleh itu kami mendapat $a^{\omega-1} \equiv 1 \bmod p$. Daripada Teorem Fermat, kami juga mendapat $a^{p-1} \equiv 1 \bmod p$. Pertimbangkan set $Z_p = \{0, 1, ..., p-1\}$. Set $Z_p\backslash\{0\}$ beserta operasi pendaraban adalah suatu kumpulan dengan peringkat $p-1$. Sebenarnya kumpulan itu juga adalah kumpulan kitaran. Pertimbangkan integer-integer $a \in Z_p\backslash\{0\}$. Peringkat setiap unsur dalam $Z_p\backslash\{0\}$ membahagi $p-1$ malahan terdapat unsur yang berperingkat $p-1$. Oleh kerana unsur itu juga memenuhi $a^{\omega-1} \equiv 1 \bmod p$ maka diperoleh $p-1$ membahagi $\omega-1$.