PROSTI BROJEVI U ASIMETRIČNOJ KRIPTOGRAFIJI

Mladen Veinović
Univerzitet Singidunum
Bulevar Zorana Đinđića 44, Novi Beograd
Telefon: +381 11 31 21 349
Fax: +381 11 31 21 563
e-mail: mveinovic@singidunum.ac.rs

Bogdan Brkić
Ministarstvo finansija Republike Srpske
Trg Republike Srpske I, 78000 Banja Luka
Telefon: +387 65 567 941
Fax: +387 51 339 686
e-mail: bbogdan@teol.net

Miroslav Ćajić
MikroByte doo
Vese Rackovića 16, 73220 Rogatica
Telefon: +387 65 631 675
Fax: +387 58 417 517
e-mail: miroslavcajic@gmail.com

Sažetak
Tajnost kriptosistema sa javnim ključem zasniva se na teško rješivim ili još uvijek nerješivim matematičkim problemima. Rezultat toga je da brojevi i njihova složenost imaju veoma veliku ulogu u održavanju sigurnosti sistema. Za asimetrične šifarske algoritme, kao što je RSA, po značaju se posebno izdvaja jedna klasa brojeva - prosti brojevi. Ako bi postojao efikasan algoritam za faktorizaciju prostih brojeva, sigurnost asimetričnih algoritama bila bi dovedena u pitanje. U većini asimetričnih algoritama za generisanje ključeva se koriste veliki prosti brojevi sa stotinu i više decimalnih cifara. Predmet ovog rada su metode generisanja prostih brojeva i testovi za ispitivanje da li je broj prost.

Ključne riječi: prosti brojevi, generisanje, testovi, kriptografija.

Abstract
Privacy of public key cryptosystems is based on the hardly solvable or still unsolvable mathematical problems. The result is that numbers and their complexity has important role in the mantenance of the security systems. For asymmetric cipher algorithms, such as RSA, the significance of particular stands out one class of numbers - prime numbers. If there would be an efficient algorithm for prime numbers factorisation, security of asymmetric algorithms would be called into question. In most asymmetric algorithms for key generating we using a large prime numbers with hunderd or more decimal digits. The subject of this work are methods of generating prime numbers and primality tests.

Keywords: prime numbers generating, primality tests, cryptography

1. UVOD

Asimetrični algoritmi svoju tajnost ne zasnivaju na nepoznavanju algoritma, već na upotrebi jednosmjernih funkcija tj. funkcija čije je pronalaženje inverznog postupka težak matematički problem, teško rješiv računarskim resursima u realnom vremenu. Uloga prostih brojeva u kriptosistemima sa javnim ključem je dvostruka. Prva je da se koriste za generisanje ključeva, a druga je da se neke njihove osobine koriste kao jednosmjerne funkcije. Za osrednji kriptografski sistem sa javnim ključevima potrebno je mnogo prostih brojeva koji se koriste za generisanje ključeva. Imajući ovo u vidu iznijećemo neke činjenice o prostim brojevima:

- prostih brojeva čija je dužina manja od n bita iznosi n/(ln n), pa tako prostih brojeva čija je dužina manja ili jednaka 512 bita ima oko 10^{151},
- vjerovatnoća da se izaberu dva ista prosta broja je veoma mala,
- kada bi mogla da se formira baza svih prostih brojeva ili samo prostih brojeva čija je dužina npr. manja ili jednaka od 1024 bita, dobila bi se tolika količina podataka koju bi bilo nemoguće pretražiti s ciljem da se provali algoritam sa javnim ključem

Pogrešan način pronalaženja prostih brojeva je da se oni generišu i da se vrši njihova faktorizacija. Mnogo lakše je utvrditi da li je broj prost, nego izvršiti njegovu faktorizaciju.

2. ZNAČAJ PROSTIH BROJEVA ZA ASIMETRIČNU KRIPTOGRAFIJU

U literaturi koja se odnosi na RSA algoritme, često se predlaže da se u izboru para ključeva koriste jaki prostih brojevi p i q za generisanje n. Jaki prosti brojevi imaju određene osobine koje čine proizvod n teško faktorizovanim. Razlog za ovakav izbor je to što su neke od metoda faktorizacije, kao što su Polardove p-1 i p+1 metode, naročito pogodne za proste brojeve p za koje p-1 ili p+1 imaju samo male proste faktore. Jaki prosti brojevi su otporni na ovakve napade. Napredak u faktorizaciji tokom poslednjih deset godina pokazuje opravdanost upotrebe jakih prostih brojeva. Za generisanje ključeva u RSA algoritmu preporučuje se upotreba velikih prostih brojeva reda veličine 100 decimalnih cifara i više. Poznato nam je da se generisanje ključeva za RSA algoritam odvija na način da se izaberu dva velika broja p i q iz kojih računamo n=p*q, a zatim odaberemo e i d uz poštivanje određenih uslova. Postoji dobar razlog zbog čega n ima tačno dva prosta faktora. Prvi razlog je jednostavnost formule za računanje $\Phi(n)=(p-1)(q-1)$, a drugi je teža faktorizacija n, jer je poznato da je faktorizaciju teže izvršiti ukoliko broj ima manje prostih faktora. Da bi smo stekli jasniju predstavu o veličini trenutno najvećih poznatih prostih brojeva u Tabeli 1. razmotrićemo podatke objavljene na Web stranici [7] koja bilježi sve pronađene velike proste brojeve od 1994. godine. Podaci su ažurirani 25.10.2009. godine.

Tabela 1.

	Prost broj	**Broj cifara**	**God. pronalaska**
Najveći prost broj	$2^{43112609}-1$	**12.978.189**	**2008**
Najveći prost broj "blizanac"	**$65516468355\cdot2333333+1$**	**100.355**	**2009**
Najveći Marsenov prost broj	$2^{43112609}-1$	**12.978.189**	**2008**
Najveći Sophie-Germain prost broj	**$607095\cdot2^{176311}-1$**	**53.081**	**2008**

3. TESTOVI ZA ISPITIVANJE DA LI JE BROJ PROST

3.1. Generalni model odabira prostog broja provjerom djelilaca kandidata n

Za kriptografe je generisanje velikih prostih brojeva uvijek predstavljalo problem. Generalan metod za generisanje prostih brojeva je da se generiše broj n odgovarajuće dužine, a da se zatim provjerava da li je on prost. Najtrivijalnija provjera da li je broj prost bila bi da se provjeri da li je on djeljiv sa bilo kojim prostim brojem manjim od n. Uopšteni algoritam bi bio da se generiše neparan kandidat n, da se provjeri da li djeljiv bilo kojim prostim brojenjm manjim od n, pa ako jeste da se vrši se izbor novog kandidataje prost ispitivanjem njegovih djelilaca ji je kandidat za prost broj. Neznatna unapređenja mogu se postići načinom odabira sledećeg broja koji je kandidat za prost broj tokom povratka na prvi korak. Najprostiji način je provjera upotrebom sekvence n+2, n+4, n+6, ...

3.2. Testovi za ispitivanje da li je broj prost

Prethodan metod je neefikasan jer zahtijeva dosta vremena za provjeru i znatne računske resurse. Umjesto provjere da li su svi brojevi manji od posmatranog broja n njegovi djelioci, koriste se testovi za ispitivanje da li je broj prost. Postoje dvije grupe testova za ispitivanje da li je broj prost:

- "stvarni testovi" čiji je rezultat podatak da li je broj-kandidat prost ili nije prost i
- "probabilistički testovi" koji govore da je broj-kandidat ili "vjerovatno prost" ili je složen

Probabilistički testovi deklarišu cijeli broj n kao kandidata za prost broj sa nekom vjerovatnoćom, dok stvarni testovi obezbjeđuju matematički dokaz da je broj prost ili složen. Probabilistički testovi generalno zahtijevaju manje računskih resursa i izvršavaju se za kraće vrijeme od stvarnih testova.

3.2.1. Probabilistički testovi

Probabilistički testovi su najčešće korišteni za ispitivanje da li je broj prost. Pored testiranog broja n, koriste i slučajno odabrane brojeve a. Ovakvim testovima se za prost broj nikada ne može dobiti rezultat da je složen, ali je moguće da složen broj testom bude prepoznat kao prost. Redukovanje ove greške može se postići ukoliko se test ponavlja sa nekoliko slučajno odabranih vrijednosti a. Za dva najčešće korištena testa, Fermaov i Miler-Rabinov, za bilo koji složeni broj n najmanje polovina brojeva koje je moguće birati za a detektuju da je n složen. Ovo znači da k ponavljanja smanjuje grešku vjerovatnoće za najviše 2^{-k} i može se učiniti proizvoljno malom povećavanjem broja pokušaja k.

Osnovni algoritam svakog probabilističkog testa bio bi:

- slučajno biramo broj a
- provjeravamo određene jednakosti (u zavisnosti od odabranog testa) koje uključuju brojeve a i n
- ako jednakosti ne vrijede test se prekida i broj a je "svjedok složenosti" broja n; ako jednakosti vrijede vraćamo se na prvi korak i ponavljamo postupak dok ne postignemo potrebnu sigurnost

Najpoznatiji probabilistički testovi su: Fermaov test, Solovej-Štrasenov test, Lemanov test i Miler-Rabinov test koji je i najčešće korišten i naziva se još i "jaki pseudoprosti test" jer se zasniva na nekim osobinama jakih pseudoprostih brojeva. U pogledu korektnosti rezultata Miler-Rabinov test nikada nije lošiji od Solovej-Štrasenovog, a Solovej Štrasenov nikada nije lošiji od Fermaovog testa za bilo koji broj n. Među navedenim testovima najbolji je Miler-Rabinov, jer zahtijeva najmanji broj operacija i ima jednostavniju implementaciju i najmanju vjerovatnoću greške u odnosu na sve ostale testove.

3.2.2. Stvarni testovi

Za razliku od probabilističkih testova gdje rezultat testiranja može biti da je broj složen ili prost, ali sa određenom vjerovatnoćom, rezultat stvarnih testova je tvrdnja da je testirani broj n prost ili složen. Zbog toga se često nazivaju i "algoritmi dokazivanja da je broj prost". Ovi testovi su računski znatno složeniji od probabilističkih, pa se zato na kandidata n obično prvo primjenjuju probabilistički testovi, a zatim testovi dokazivanja da je broj prost. Cijeli broj n za koji se utvrdi da je prost nekim od algoritama dokazivanja da je broj prost nazivaju se "dokazano prosti". Značajniji stvarni testovi su:

- Lukas-Lemerov test za testiranje Mersenovih brojeva,
- testiranje faktorizacijom broja n-1,
- test Jakobijevih suma,
- testiranje upotrebom eliptičkih krivih,
- AKS (Agrawal–Kayal–Saxena) test,
- LLR (Lucas–Lehmer–Riesel) i BLS (Brillhart–Lehmer–Selfridge) testovi i
- Pepinov test

4. GENERISANJE PROSTIH BROJEVA

4.1. Generisanje slučajnih kandidata za prost broj

Jedna od taktika za generisanje prostih brojeva bila bi:

- generisati slučajan n-bitni prost broj p
- postaviti bit najveće težine na 1 čime se osigurava da je generisani broj tražene dužine
- postaviti bit najmanje težine na 1 čime se osigurava da je generisani broj neparan
- provjeriti da li je p djeljiv malim prostim brojevima: 3, 5, 7, 11, ... (najčešće se ispituje djeljivost sa svim prostim brojevima manjim od 256, a najefikasnije je testirati djeljivost sa prostim brojevima manjim od 2000)
- za pet slučajnih vrijednosti a izvršiti Miler-Rabinov test
- ako p ne prođe test vratiti se na prvi korak i generisati novo p

Provjera djeljivost sa malim prostim brojevima nije obavezna, ali se sa njom odbacuje velik postotak neparnih brojeva prije nego što se dođe do sledećeg koraka, Miler-Rabinovog testa. Ispitivanje djeljivosti sa 3,5,7 može da odbaci 54% neparnih kandidata, ispitivanje djeljivosti prostim brojevima manjim od 100 odbacuje se 76%, manjim od 256 odbacujemo 80%, ... Odbacivanjem prostih brojeva koji su djelioci p, a manji su od n, odbacuje se 1.12/(ln n) kandidata.

4.2. Inkrementalno pretraživanje

Druga moguća realizacija bila bi da se nakon prvog generisanja kandidata p i njegovog eventualnog odbacivanja ne generiše novi slučajni kandidat, već da se izbor novih kandidata vrši u okolini broja p. Za svakog novog kandidata se primjenjuje prethodni test, a pokazuje se da je ispitivanje djeljivosti sa malim prostim brojevima nešto jednostavnije ako je ispitivanje već rađeno za prethodne kandidate koji su od p_k (p_{k-2}, p_{k-4}, ...).

4.3. Generisanje jakih prostih brojeva

Za generisanje jakih prostih brojeva može se koristiti Gordonov algoritam. Prost broj p je jak prost broj ako postoje cijeli brojevi r, s i t takvi da vrijedi:

- p-1 ima veliki prost faktor, označen sa r
- p+1 ima veliki prost faktor, označen sa s i
- r-1 ima veliki prost faktor, označen sa t.

4.4. Maurerov algoritam za generisanje "sigurno" prostih brojeva

Maurerov algoritam generiše "sigurno" proste brojeve, odnosno brojeve za koje se nekim od stvarnih testova može pokazati da su prosti. Ovaj podskup prostih brojeva je gotovo uniformno raspoređen po skupu svih prostih brojeva. Očekivano vrijeme za izvršavanje ovakvog algoritma je neznatno veće od vremena potrebnog za algoritam 4.1. sa jednim ciklusom u Miler-Rabinovom testu.

6. ZAKLJUČAK

Prosti brojevi imaju veoma važnu ulogu u kriptografiji. U ovom radu navedene su metode generisanja prostih brojeva i testovi za ispitivanje da li je broj prost. Analiza je pokazala da su stvarni testovi tačniji, ali manje efikasni u pogledu vremena ispitivanja i upotrebe računskih resursa. S druge strane, probabilistički testovi su znatno brži i lakši za implementaciju, ali daju rezultat sa određenom vjerovatnoćom odnosno sa manjom tačnošću. Pri određivanju parametara za generisanje ključeva RSA algoritma, prostih brojeva p i q, treba voditi računa o sledećem:

- p i q moraju biti dovoljno veliki kako bi faktorizacija njihovog proizvoda bila računski teško izvodljiva
- p i q bi trebali biti slučajni prosti brojevi takvi da je napad njihovog otrivanja "grubom silom" neisplativ
- p i q bi trebalo da imaju predefinisanu bitsku dužinu kako bi zadovoljavali specifikaciju algoritma

Otkriće RSA algoritma dovelo je do razmatranja nekoliko dodatnih ograničenja pri izboru p i q koja bi osigurala da rezultujući RSA sistem bude bezbjedan od napada kriptoanalitičara, te je uvedena definicija jakih prostih brojeva.

LITERATURA

[1.] A.Menzes, P.van Oorschot, S. Vanstone: *Handbook of Applied Cryptography*, CRC Press, 1996
[2.] B. Schneier: *Applied Cryptography*, John Wiley & Sons, 1996
[3.] R.L.Rivest, A.Shamir, L.Adleman: *A Method for Obtaining digital Signatures and Public-KeyCryptosystems*, Communications of ACM, 1983
[4.] U.M. Maurer: *Fast Generation of Prime Numbers and Secure Public-Key Cryptographic Parameters*, Journal of Cryptology, vol. 8, pp. 123-155, 1995
[5.] M. Masud, H. Galzie, K.A. Hossain, M.U. Islam: *Aspect of Prime Numbers in Public Key Cryptosystem*, International Journal of The Computer, the Internet and Management Vol. 13 No.2 (May-August, 2005)
[6.] E. Kranakis: *Primality and Cryptography*, John Willey & Sons, 1991
[7.] The Univeristy of Tennessee: "*The Largest Known Primes*", (pristup 31.1.2010)