RSA ALGORITAM I NJEGOVA PRAKTIČNA PRIMENA

Kuljanski R. *Sonja,*
EXECOM DOO, Novi Sad

UDC: 004.421

Sažetak:

RSA algoritam jeste algoritam sa javnim ključem koji uključuje tri koraka: generisanje ključa, enkripciju i dekripciju. RSA enkripciona šema je deterministička što znači da se osnovni tekst uvek enkriptuje u isti šifrovani tekst za unapred zadati javni ključ. Da bi se izbegao ovaj problem, praktična implementacija RSA algoritma obično koristi neke strukture, kao što je dodavanje slučajnog teksta u samu poruku pre enkripcije. Ovo dodavanje obezbeđuje da osnovna poruka bude sigurna i da se može enkriptovati u veliki broj različitih šifrovanih poruka. Standardi, kao što je PKCS #1, pažljivo su dizajnirani tako da dodaju tekst u osnovnu poruku pre RSA same enkripcije.

Ključne reči: *kriptografija, enkripcija, dekripcija, RSA, PKCS, OAEP, SAEP*.

Uvod

Kriptografija je poznavanje „tajnog pisanja“, odnosno poznavanje čuvanja informacija tako da one budu čitljive samo onima kojima su namenjene. Reč kriptografija potiče od grčkih reči kriptos (κριπτοσ) – tajna i grafien (γραφιεν) – pisati.

Kriptografija je doživela najznačajniji razvoj 1976. godine kada su Difie (*Whitfield Diffie*) i Helman (*Martin Hellman*) izdali [2]. U ovoj knjizi je uveden revolucionarni koncept kriptografije sa javnim ključem. Takođe, prikazan je i novi, genijalni metod za razmenu ključa, čija je sigurnost bazirana na nerešivosti problema diskretnog logaritma. Mada autori u to vreme nisu imali praktičnu realizaciju šeme enkripcije javnim ključem, ideja je bila jasna i dovela je do velike zainteresovanosti u svetu kriptografije. Rivest (*Ron Rivest*), Šamir (*Adi Shamir*) i Adleman (*Len Adleman*) 1978. godine otkrili su prvu praktičnu šemu za enkripciju sa javnim ključem, sada poznatu kao RSA šema.

Kriptografija se ujedno smatra i granom matematike i granom teorijskog računarstva. Enkriptovanje i digitalni potpis su kriptografske tehnike koje se koriste da bi se implementirali bezbednosni servisi. Osnovni ele-

sonjakuljanski@gmail.com

ment koji se koristi naziva se enkripcijski sistem ili algoritam enkripcije. Svaki enkripcijski sistem obuhvata par transformacija podataka koje se nazivaju enkripcija i dekripcija. U asimetričnim algoritmima, odnosno algoritmima sa javnim ključem, ključ za enkripciju se razlikuje od ključa za dekripciju. Štaviše, ključ za dekripciju se ne može (u razumnom vremenu) izračunati na osnovu ključa za enkripciju. Ključ za enkripciju naziva se „javni ključ“ i samim tim svako može uputiti šifrovanu poruku primaocu, ali je samo primalac može dešifrovati. Ključ za dekripciju se naziva „tajni ključ“.

Deterministička enkripciona šema je kriptosistem koji uvek produkuje isti šifrovani tekst za dati osnovni tekst i unapred zadati ključ, čak i pored nezavisnog izvršavanja enkripcionog algoritma. Zato deterministička enkripcija može odati informacije napadaču, koji može prepoznati od ranije poznati šifrovani tekst.

Verovatnosna enkripcija koristi slučajnost u algoritmima enkripcije, pa se dobija različita šifrovana poruka kada se više puta enkriptuje ista osnovna poruka. Da bi semantički bio siguran, odnosno da bi sakrio delimične informacije o osnovnoj poruci, enkripcioni algoritam mora biti verovatnosni. Verovatnosna enkripcija ima veoma značajnu ulogu u enkripciji sa javnim ključem. Pretpostavimo da napadač posmatra šifrovani tekst i pretpostavlja da je osnovna poruka DA ili NE. Kada se koristi deterministički enkripcioni algoritam napadač može jednostavno da pokuša da enkriptuje svako od njegovih nagađanja pomoću javnog ključa i da uporedi rezultat sa ranije posmatranim šifrovanim tekstom. Da bi se odbranili od ove vrste napada, enkripcione šeme sa javnim ključem moraju koristiti elemente slučajnosti i tako osigurati da će se jedna osnovna poruka enkriptovati u veliki broj mogućih šifrovanih poruka.

Deterministička enkripciona šema može se konvertovati u verovatnosnu dodavanjem slučajnog stringa u osnovnu poruku pre enkripcije nekim determinističkim algoritmom. U tom slučaju dekripcija zahteva primenu determinističkog algoritma i ignorisanje slučajnog stringa koji je dodat. Ranije šeme koje su primenjivale ovaj pristup bile su razotkrivene na osnovu ograničenja u determinističkim enkripcionim šemama. OAEP integriše slučajan umetak tako da je sigurno korišćenje bilo koje *trapdoor* permutacije.

Osnovni RSA algoritam

Pre dvadeset godina Difie i Helman su izjavili: “We stand today on the bank of revolution in cryptography”. Danas se nalazimo na sredini te revolucije. U poslednje dve decenije došlo je do prave eksplozije istraživanja u oblasti kriptologije. Mnogi kriptosistemi bili su predlagani, a

mnogi od njih bili su razbijeni. Povezanost između kriptologije, teorije kompleksnosti i teorije brojeva postepeno je otkrila i obogatila sve tri grane istraživanja. U radu [3], kao i u radu [6], navedeni su osnovni koncepti kriptografije i kriptografski algoritmi.

Rivest, Šamir i Adleman su 1978. godine objavili kriptosistem sa javnim ključem koji zadovoljava sva tri zahteva koja su postavili Difie i Helman. U toj enkripcijskoj šemi svaki korisnik ima uređeni par celih brojeva (e,n), što predstavlja javni ključ, pri čemu je n proizvod dva velika prosta broja p i i važi $NZD(e,\varphi(n))=1$. Broj $\varphi(n)$ predstavlja red multiplikativne grupe Z^n. Algoritam za enkripciju je:

$$c = m^e \pmod n \tag{1}$$

Odgovarajući tajni ključ je d, pri čemu je $d \cdot e \equiv 1 (\mathrm{mod}\, \varphi(n))$ i algoritam za dekripciju je:

$$m = c^d \pmod n \tag{2}$$

RSA pretpostavka je pretpostavka da je RSA problem težak kada je modul n dovoljno velik i slučajno izabran i kada je osnovna poruka m (a samim tim i šifrovana poruka c) slučajno izabran ceo broj između 0 i $n-1$. Pretpostavka, u stvari, kaže da je RSA funkcija trapdoor jednosmerna funkcija (pri čemu je privatni ključ trapdoor).

Jasno je da RSA problem nije teži od problema faktorisanja celog broja, tako da napadač koji uspe da faktoriše modul n može da izračuna tajni ključ d ako mu je poznat javni ključ (e,n). Još uvek se ne zna da li važi i obrnuto, odnosno da li algoritam za faktorisanje celih brojeva može biti efikasno konstruisan iz algoritma za rešavanje RSA problema. Detaljno objašnjenje problema faktorizacije i RSA problema dato je u [4, poglavlja 3.2 i 3.3].

Algoritam 1 (Generisanje ključeva za RSA enkripciju javnim ključem)

Sažetak: svaki entitet kreira RSA javni ključ i odgovarajući privatni.

1. Generisati dva velika slučajna (i različita) prosta broja p i q, otprilike iste veličine.

2. Izračunati $n = pq$ i $\varphi = (p-1)(q-1)$.

3. Izabrati slučajan ceo broj *e*, $1 < d < \varphi$, takav da $NZD(e,\varphi)=1$.

4. Koristiti prošireni Euklidov algoritam za računanje jedinstvenog celog broja *d*, $1 < d < \varphi$, takvog da $ed \equiv 1 (\mathrm{mod}\, \varphi)$.

5. Javni ključ je (e*, n*), privatni ključ je *d*.

Definicija 1. Celi brojevi *e* i *d* u RSA generatoru ključa nazivaju se enkripcioni eksponent i dekripcioni eksponent, respektivno, dok se *n* naziva modul.

Algoritam 2 (RSA enkripcija javnim ključem)

Sažetak: entitet B enkriptuje poruku m za entitet A, koju će entitet A da dekriptuje.

Enkripcija: entitet B treba da:

1. Dobije od entiteta A autentičan javni ključ (e,n).
2. Predstavi poruku koju želi da enkriptuje kao ceo broj m u intervalu $[0,n-1]$.
3. Izračuna $c = m^e \pmod n$.
4. Pošalje šifrovani tekst c entitetu A.

Dekripcija: za dešifrovanje teksta m iz šifrovanog teksta c entitet A treba da:

1. Koristi tajni ključ d za dešifrovanje $m = c^d \pmod n$.

RSA enkripcija u praksi

Enkripcioni RSA algoritam je deterministički enkripcioni algoritam (nema slučajnih komponenata) i napadač uspešno može izvršiti *chosen plaintext* napad na kriptosistem, enkriptujući mogući osnovni tekst, korišćenjem javnog ključa, i proveravajući da li je jednak šifrovanom tekstu koji želi da dekriptuje. Kriptosistem se naziva „semantički siguran“ ako napadač ne može razlikovati dva enkriptovana teksta, čak i ako mu je poznat odgovarajući osnovni tekst. RSA bez *padding* šeme nije semantički siguran.

Da bi se izbegao ovaj problem praktična RSA implementacija obično ubacuje neki dodatak (*padding*) u osnovnu poruku pre nego što se izvrši enkripcija. Ovaj dodatak omogućava da osnovna poruka ne upadne u opseg nesigurnog osnovnog teksta i da se data poruka enkriptuje u jednu od brojnih, različitih šifrovanih poruka.

Pravilno enkriptovanje RSA algoritmom

Belar (*Mihir Bellare*) i Rodevej (*Phillip Rogaway*) 1993. godine formalizovali su koncept *random orakla*, što predstavlja veoma važan deo teorije kompleksnosti u kriptografiji. Taj novi alat im je omogućio da predstave nekoliko asimetričnih enkripcionih šema koje su efikasne i dokazano sigurne (u random orakl modelu). *The Optimal Asymmetric Encryption Padding* (OAEP) najznačajnija je šema tog modela.

Određeno vreme naučnici su pokušavali da dođu do dokaza o sigurnosti kriptografskih protokola u redukcionom smislu. Da bi to postigli, predstavljali su algoritme koji koriste efektivan napad kao potprogram da bi razotkrili početnu tešku pretpostavku (kao što je RSA pretpostavka ili nemogućnost

faktorizacije celih brojeva). Takvi algoritmi nazivaju se „redukcioni“ i mogu biti uspešni, grubo govoreći, ako ne zahtevaju previše poziva potprograma.

Pre nekoliko godina započeo je novi pravac u istraživanjima koji je kombinovao dokazivanje sigurnosti i efikasnosti. Da bi postigli cilj Belar i Rodževej su formalizovali heuristiku koju su predložili Fiat (*Amos Fiat*) i Šamir. Ona se sastojala u izradi idealne pretpostavke o nekom objektu, kao što je heš funkcija, prema kojoj je moguće simulirati ponašanje zaista slučajne funkcije. Ova pretpostavka, poznata kao „random orakl model“, može izgledati strogo i bez mogućnosti praktične primene.

U random orakl modelu se pretpostavlja da napadač ne može da koristi bilo koji specifičan nedostatak heš funkcije koja se koristi u praksi.

Potrebno je naglasiti da čak i formalna analiza u random orakl modelu nije jak dokaz sigurnosti, jer je zasnovana na idealnoj pretpostavci. Medutim, ovaj model može obezbediti dovoljno sigurnosti i može se koristiti kao osnova za veoma efikasne šeme, videti [5, poglavlje 3.1].

Random orakl, RSA-PKCS[1] i OAEP šema

Random orakl u kriptografiji je orakl (crna kutija) koji na svaki upit odgovara slučajnim odgovorom, izabranim uniformno iz izlaznog domena. Sa druge strane, random orakl je matematička funkcija koja slika svaki mogući upit na slučajan odgovor iz izlaznog domena.

Random orakl predstavlja matematičku apstrakciju koja se koristi u kriptografskim dokazima, a koristi se kada nije poznata matematička funkcija koja dovodi do osobina traženih u dokazu. Sistem za koji je dokazano da je siguran, korišćenjem ovog načina dokazivanja, smatra se sigurnim u random orakl modelu, za razliku od sigurnosti u standardnom modelu 1. U praksi se random orakl obično koristi za modeliranje kriptografske heš funkcije u šemama u kojima je potrebna pretpostavka o strogoj slučajnosti. Ovakvi dokazi obično pokazuju da je sistem siguran, ukazujući na činjenicu da napadač mora da zahteva nemoguće ponašanje orakla ili da reši neki matematički problem za koji se veruje da je težak u cilju razotkrivanja sistema.

Ne postoji realna funkcija koja predstavlja random orakl. U suštini, određene enkripcione šeme su dokazano sigurne u random orakl modelu, ali su trivijalno nesigurne kada bilo koja realna funkcija zameni random orakl. Bez obzira na to, dokaz o sigurnosti u random orakl modelu obično daje veoma jak dokaz da napad koji ne razbije druge pretpostavke dokaza (kao što je faktorizacija celih brojeva) mora otkriti neke nepoznate osobine heš funkcije koja se koristi. Među šemama koje su dokazano sigurne u random orakl modelu jedna od najznačajnijih je OAEP šema.

[1] Public-Key Cryptography Standards.

Posle Blaihenbaherovog (*Daniel Bleichenbacher)* razarajućeg napada na RSA-PKCS #1 v1.5 1998. godine, RSA-OAEP (RSA-PKCS #1 v2.0) postao je naslednik standarda 2, a samim tim i internacionalni standard. Interesantno je da je ˇ Soup (*Victor Shoup*) nedavno pokazao da originalni dokaz sigurnosti OAEP-a nije korektan. Srećom, ubrzo posle toga otkrio je formalan i kompletan dokaz koji garantuje visok nivo sigurnosti RSA-OAEP šeme. Međutim, ovaj novi dokaz sigurnosti još uvek ne garantuje sigurnost za veličinu ključa koji se koristi u praksi. Alternative OAEP šeme, kao što su OAEP$^+$ i SAEP$^+$ omogućavaju efikasnije dokazivanje i zbog toga obezbeđuju adekvatan nivo sigurnosti za veličinu ključa koja se koristi u praksi. Sve tri šeme su navedene u [5, poglavlje 3].

RSA-PKCS #1 v1.5 enkripcija

Široko rasprostranjen *padding* za RSA enkripciju definisan je u PKCS #1 v1.5 standardu: za bilo koji modul $2^{8(k-1)} \le n \le 2^{8k}$, kako bi se enkriptovala l bitova dugačka poruka m $(l \le k-11)$ potrebno je slučajno izabrati string r dužine $k-3-l$ bitova. Tada se definiše k bitova duga poruka $M = 02 \,||\, * \,||\, 0 \,||\, *$. Ako je šifrovana poruka ovog formata, dekriptor dešifruje osnovni tekst, a ako nije onda se šifrovani tekst odbacuje.

Tabela 1
Table 1

0	2	ne nula bitovi (više od 8 bitova)	0	m

Intuitivno, ovaj *padding* izgleda dovoljan da otkloni nedostatke obične RSA enkripcije, ali ne postoji formalan dokaz koji to i garantuje. Blaihenbaher je neočekivano pokazao da jednostavan aktivan napad može kompletno slomiti PKCS #1. Taj napad je primenjen na realan sistem kao što je Web server koji koristi SSL v3.03. Ovi serveri često produkuju specifičnu poruku o grešci ako šifrovani tekst nije korektan. Ta osobina servera omogućava napadaču da testira da li su dva najznačajnija bajta šifrovanog teksta c baš 02. Ako jesu, napadač saznaje sledeće ograničenje za dekriptovanje šifrovanog teksta c:

$$2 \cdot 2^{8(k-2)} \le c^d \bmod n < 3 \cdot 2^{8(k-2)} \tag{3}$$

Zahvaljujući samoreducibilnosti RSA permutacije, naročito homomorfizmu $cs^e = m^e s^e = (ms)^e \bmod n$, kompletna dekripcija šifrovanog teksta c može biti razotkrivena posle relativno malog broja upita. Samo nekoliko miliona upita potrebno je za 1024-bitni modul.

Blaihenbaherov napad imao je uticaj na mnoge praktične sisteme i odjednom je postalo jasno koliko je veliki značaj formalnog dokaza sigurnosti, videti [5, poglavlje 2.2].

OAEP šema

U vreme kada je Blaihenbaher objavio svoj napad na RSA-PKCS #1 v1.5 jedina efikasna i „dokazano sigurna“ enkripciona šema, zasnovana na RSA problemu, bila je OAEP šema, koju su predložili Belar i Rodževej. OAEP se može koristiti sa bilo kojom *trapdoor* permutacijom f.

Belar i Rodževej su dokazali da OAEP *padding* korišćen bilo kojom *trapdoor* – jednosmernom permutacijom f semantički obezbeđuje sigurnost enkripcionoj šemi. Dopunjavanjem osnovne poruke oni su, pored sigurnosti, dokazali i da je OAEP *padding* slabo svestan osnovne poruke. Kriptosistem je slabo svestan osnovne poruke ako bilo koji algoritam teško može da dode do šifrovanog teksta kad mu nije poznat odgovarajući osnovni tekst. Nažalost, *adaptive chosen-ciphertext* napad dozvoljava napadaču da proizvoljno dugo pristupa dekripcionom oraklu, čak i posle primanja spornog šifrovanog teksta o kojem napadač želi da dobije neke informacije. Zbog toga, semantička sigurnost, zajedno sa slabom svešću o osnovnom tekstu, samo implicira semantičku sigurnost na *non-adaptive chosen-ciphertext* napad (*lunchtime* napad ili indiferentan*chosen-ciphertext* napad), gde je pristup dekripcionom oraklu ograničen, dok napadač ne primi testirani šifrovani tekst.

Činjenica je da je jedini formalan dokaz sigurnosti OAEP šeme dokaz da je ona semantički sigurna na *lunchtime attacks,* pod pretpostavkom da je osnovna permutacija jednosmerna. Međutim, veruje se da je OAEP takođe semantički sigurna šema na *chosen-ciphertext* napad.

OAEP šema je vrsta Feistelove mreže koja koristi par random orakla G i H da bi obradila osnovni tekst asimetričnom enkripcijom. Kada se kombinuje sa bilo kojom jednosmernom *trapdoor* permutacijom f ova obrada će dokazano, u smislu random orakl modela, rezultirati kombinovanom šemom koja je semantički sigurna na *chosen plaintext* napad (IND-CPA). Kada je implementirana sa pouzdanom *trapdooor* permutacijom (na primer RSA), OAEP će biti dokazano siguran i na *chosen ciphertext* napad (IND-CCA).

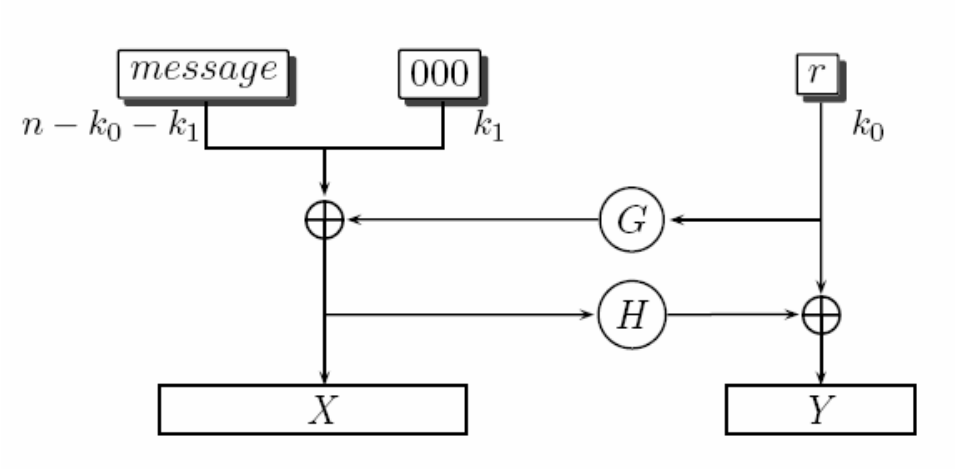
OAEP šema zadovoljava sledeća dva uslova:

– dodaje elemente slučajnosti koji mogu biti korišćeni za konvertovanje determinističke enkripcione šeme (kakva je RSA šema) u verovatnosnu šemu,

– štiti delimičnu dekripciju šifrovanog teksta, onemogućavajući napadača da otkrije bilo koji deo osnovnog teksta ako nije u mogućnosti da invertuje *trapdoor* jednosmernu permutaciju f.

Dijagram OAEP šeme

Na prikazanom dijagramu:
- n predstavlja broj bitova RSA modula;
- k_0 i k_1 su celi brojevi određeni protokolom;
- $message$ je osnovna poruka, čija je dužina $n - k_0 - k_1$ bitova;
- G i H su heš funkcije utvrđene protokolom.



Slika 1 – OAEP šema[2]
Figure 1 – OAEP diagram

Enkripcija se vrši na sledeći način:

1. osnovna poruka se proširi sa k_1 nula i na taj način se dobija poruka dužine $n - k_0$ bitova;
2. r je proizvoljan string dužine k_0 bitova;
3. G je heš funkcija koja konvertuje k_0 bitova stringa r u $n - k_0$ bitova;
4. $X = message0^{k_1} \oplus G(r)$;
5. H je heš funkcija koja konvertuje $n - k_0$ bitova od X u k_0 bitova;
6. $Y = r \oplus H(X)$;
7. Izlaz je $X \,||\, Y$.

Dekripcija se vrši na sledeći način:

1. izračunava se proizvoljan string $r = Y \oplus H(X)$;
2. izračunava se osnovna poruka $message0^{k_1} = X \oplus G(r)$.

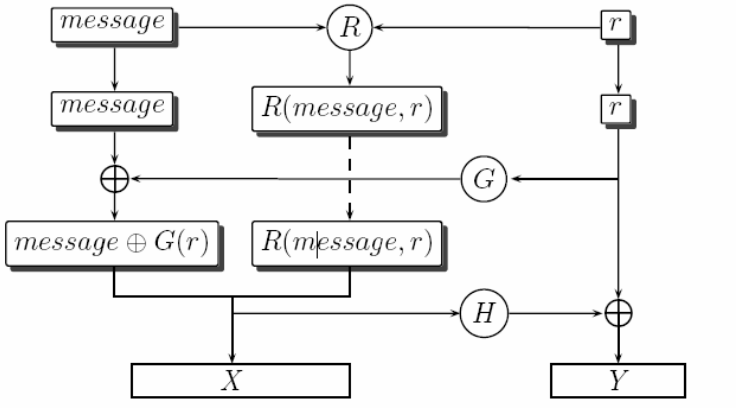
[2] Slika preuzeta iz [5].

Sigurnosna redukcija RSA inversa u napad je neefikasna u praktičnoj primeni. Samim tim, redukcija je besmislena ako se neće koristiti dovoljno veliki modul, pri čemu bi RSA invers (ili faktorizacija) zahtevao mnogo više od 2^{150} poziva. Koristeći postojeće tehnike faktorizacije potrebno je koristiti modul veći od 4096 bitova kako bi redukcija imala smisla. Sa druge strane, redukcija pokazuje da 1024-bitni modul obezbeđuje dokazani nivo sigurnosti od 2^{40}, što je neadekvatna zaštita, imajući u vidu trenutnu kompjutersku moć.

Alternative OAEP šeme

OAEP⁺ Padding

Šoup je predložio formalan dokaz sigurnosti RSA-OAEP šeme sa mnogo uspešnijom sigurnosnom redukcijom, ali u praktičnoj primeni to znači da bi enkripcioni eksponent trebao da bude 3. Međutim, mnogi naučnici veruju da je RSA *trapdoor* permutacija sa eksponentom $e = 3$ slabija od permutacije sa većim eksponentom. Zato je predložio modifikovanu verziju OAEP šeme, koja se naziva OAEP⁺. Ova šema koristi redundantnost promenljivih $R(message, r)$ umesto konstantnog broja 0 (k^{k_1}), pa je samim tim OAEP⁺ šema malo kompleksnija od OAEP šeme, [5, poglavlje 4.1]. Fujisaki je u [4] pokazao da isto važi za RSA permutaciju sa bilo kojim javnim eksponentom e.

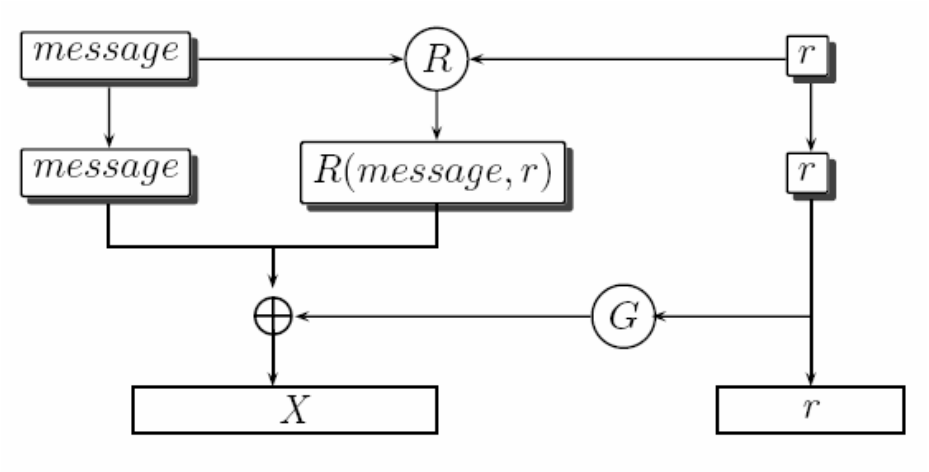


Slika 2 – OAEP⁺ šema[3]
Figure 2 – OAEP⁺ diagram

[3] Slika preuzeta iz [5].

SAEP⁺ Padding

Bonei je nedavno objavio novu *padding* šemu, SAEP⁺. Ona je jednostavnija od OAEP šeme, pa je nazvana *Simplified Asymmetric Encryption Padding*. Dok je OAEP šema dvostruka Feistelova mreža, SAEP⁺ je jednostruka. Međutim, za velike eksponente $(e > 2)$, SAEP⁺ ne garantuje sigurnost u praktičnoj primeni [5, poglavlje 4.2].



Slika 3 – SAEP⁺ šema[4]
Figure 3 – SAEP⁺ diagram

Zaključak

Među algoritmima sa javnim ključem postoji ogroman jaz između praktičnih šema i šema za koje je dokazano da su sigurne: praktični metodi su efikasni, ali nemaju dovoljan nivo dokazane sigurnosti, dok su dokazano sigurne šeme sigurne, ali ni blizu toliko efikasne. U ovom radu prikazane su šeme koje su dokazano sigurne, ali i zadovoljavajuće efikasne.

Pri dokazivanju sigurnosti OAEP šeme zahteva se da G i H budu random funkcije, međutim, prilikom konkretne implementacije koriste se kriptografske heš funkcije. Paradigma [1] tvrdi da rezultati koji se zasnivaju na idealnoj heš funkciju i koji dokazuju sigurnost imaju veći značaj od protokola koji su dizajnirani *ad. hoc*.

Pri dokazivanju da je OAEP sistem siguran garantuje se da napadač, koji poseduje šifrovani tekst, mora otkriti $message0^{k_1} \oplus G(r)$ ako želi da otkrije bilo šta smisleno o osnovnoj poruci $message$. Pokazano je da postoji

[4] Slika preuzeta iz [5].

jednostavnija *padding* šema za konvertovanje RSA enkripcione šeme u verovatnosnu, koja je, takođe, sigurna u random orakl modelu. To je takozvana SAEP$^+$. Primećeno je da jednostavnija *padding* šema čini sistem lakšim za opisivanje i lakšim za implementaciju, a samim tim mnogo elegantnijim. Pojednostavljenje *padding* šeme ima malo uticaja na performanse, jer je vreme za njeno izvršavanje zanemarljivo u odnosu na samu enkripciju.

Mada je SAEP$^+$ *padding* šema jednostavnija od OAEP šeme, ona je i restriktivnija. Korišćenjem OAEP i OAEP$^+$ šeme može se enkriptovati poruka koja je dugačka skoro kao I modul. Na primer, za 1024-ro bitni modul bezbedno je enkriptovati poruku koja je dugačka 768 bitova. Nasuprot tome, korišćenjem modula iste veličine, SAEP$^+$ šema može enkriptovati poruku od najviše 384 bita. Ova razlika je nije toliko bitna u svakodnevnoj upotrebi (za transport ključa), ali je ipak vredna pomena.

Literatura

[1] Bellare, M. and Rogaway, P., *Random oracles are practical: a paradigm for designing efficient protocols*," Proceedings of the First Annual Conference on Computer and Communications Security, ACM, 1993.

[2] Diffie, W. and Hellman, M., *New Directions in Cryptography*, IEEE Transactions on Information Theory, IT-22, no. 6, November 1976, pages 644–654.

[3] Menezes, A., Oorschot, P. and Vanstone, S., *Handbook of Applied Cryptography*, CRC Press, Boca Raton, October 1996.

[4] Menezes, A., *Evaluation of Security Level of Cryptography: RSA-OAEP, RSAPSS, RSA Signature*, CRYPTREC, December 2001.

[5] Pointcheval, P., *How to Encrypt Properly with RSA*, RSA Laboratories' CryptoBytes. Volume 5, No. 1 – Winter/Spring 2002, pages 9–19.

[6] Rivest, R., *Cryptology*, MIT Laboratory for Computer Science, 1990.

RSA ALGORITHM

Summary:

Introduction

RSA is an algorithm for public-key encryption. It is the first algorithm known to be suitable for encryption as well as digital signing.

The RSA encryption scheme is deterministic in the sense that under a fixed public key, a particular plaintext is always encrypted to the same ciphertext. A deterministic encryption scheme (as opposed to a probabilistic encryption scheme) is a cryptosystem which always produces the same ciphertext for a given plaintext and key, even over separate executions of the encryption algorithm. Probabilistic encryption uses randomness in an encryption algorithm, so that when encrypting the same message several times it will, in general, yield different ciphertexts.

Basic RSA algorithm

The RSA algorithm involves three steps: key generation, encryption and decryption.

The key generation algorithm:
1. Generate two large random primes, p and q.
2. Compute n = pq and φ = (p-1)(q-1).
3. Choose an integer e, 1 < e < φ, such that gcd(e, phi) = 1.
4. Compute the secret exponent d, 1 < d < φ, such that ed ≡ 1 (mod φ).
5. The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and φ secret.

• n is known as the modulus.
• e is known as the public exponent or encryption exponent or just the exponent.
• d is known as the secret exponent or decryption exponent.

Encryption:

Sender A does the following:
1. Obtains the recipient B's public key (n, e).
2. Represents the plaintext message as a positive integer m
3. Computes the ciphertext $c = m^e \bmod n$.
4. Sends the ciphertext c to B.

Decryption:

Recipient B does the following:
1. Uses his private key (n, d) to compute $m = c^d \bmod n$.

RSA encryption in practice

To solve a deterministic problem, practical RSA implementations typically embed some form of structured, randomized padding into the plaintext before encrypting it. This padding ensures that the plaintext does not fall into the range of insecure plaintexts, and that a given message, once padded, will encrypt to one of a large number of different possible ciphertexts. Standards, such as PKCS #1, have been carefully designed to securely pad messages prior to RSA encryption.

How regularly encrypt RSA algorithm

A few years ago, a new line of research started with the goal of combining provable security with efficiency. To achieve this goal, Bellare and Rogaway formalized a heuristic suggested by Fiat and Shamir. This heuristic consisted in making an idealized assumption about some objects, such as hash functions, according to which they were assumed to behave like truly random functions. This assumption, known as the „random oracle model", may seem strong, and lacking in practical embodiments.

Random oracle, RSA-PKCS and OAEP scheme

No real function can implement a true random oracle. In fact, certain encryption schemes are proven secure in the random oracle model, but are trivially insecure when any real function is substituted for the random oracle. Nonetheless, a proof of security in the random oracle model gives very strong evidence that an attack which does not break the other assumptions of the proof, if any (such as the hardness of integer factorization) must discover some unknown and undesirable property of the hash function used in the protocol to work. Many schemes have been proven secure in the random oracle model, for example the OAEP scheme.

Shoup also proposed a formal security proof of RSA-OAEP with a much more efficient security reduction, but in the particular case where the encryption exponent e is equal to 3. However, many people believe that the RSA trapdoor permutation with exponent 3 may be weaker than with greater exponents. Therefore, he also proposed a slightly modified version of OAEP, called OAEP+.

Boneh recently proposed a new padding scheme, SAEP+, to be used with RSA. It is simpler than OAEP, hence the name Simplified Asymmetric Encryption Padding: whereas OAEP is a two-round Feistel network, SAEP+ is a single round.

Key words: *cryptography, encryption, decryption, RSA, PKCS, OAEP, SAEP*

Datum prijema članka: 16. 01. 2010.

Datum dostavljanja ispravki rukopisa: 01. 02. 2010.

Datum konačnog prihvatanja članka za objavljivanje: 02. 02. 2010.