HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Korištenje eliptičnih krivulja u kriptografiji

CCERT-PUBDOC-2006-09-169

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument, koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Kriptografija je danas sveprisutna u svim dijelovima računalnog svijeta. Na većini javnih poslužitelja ukida se korištenje Telnet i sličnih nekriptiranih protokola za udaljenu komunikaciju. Kako s rastom procesne snage i novih saznanja kriptografski algoritmi postaju sve češće razbijani, kriptografske se metode nastoje proširiti novima koje će omogućiti snažniju zaštitu informacija, ali i brz postupak enkripcije tj. dekripcije.

U ovom dokumentu opisane su osnove jedne relativno nepoznate metode kriptiranja podataka zasnovane na eliptičnim krivuljama (eng. ECC - *Eliptic Curve Cryptography*). Diskretni logaritamski problem eliptične krivulje (eng. ECDLP - *Elliptic Curve Discrete Log Problem*) predstavili su 1985. godine kriptografi Neal Koblitz iz IBM-a i Victor Miller s University of Washington. Na bazi ECDLP-a, a za potrebe elektroničkog kriptiranja nastao je ECDSA (eng. *Elliptic Curve Digital Signature Algorithm*), algoritam za digitalno potpisivanje zasnovan na eliptičnim krivuljama.

ECDSA algoritam je razvijen kao alternativa za RSA, poglavito zbog svojih prednosti kao što su kraći ključevi te brže generiranje ključeva. Također, korištenje eliptičnih krivulja osigurava i brže izračune matematičkih operacija, uštede u memoriji i iskorištenju mrežnog linka, a činjenica je da u svom radu više opterećuje poslužitelj nego klijenta te zato postaje idealno za upotrebu na malim uređajima.

U nastavku dokumenta objašnjene su osnove kriptografskih metoda s naglaskom na metode zasnovane na eliptičnim krivuljama. Također, opisane su i moguće primjene eliptičnih krivulja kao i njihova implementacija te neke značajnije specifičnosti.

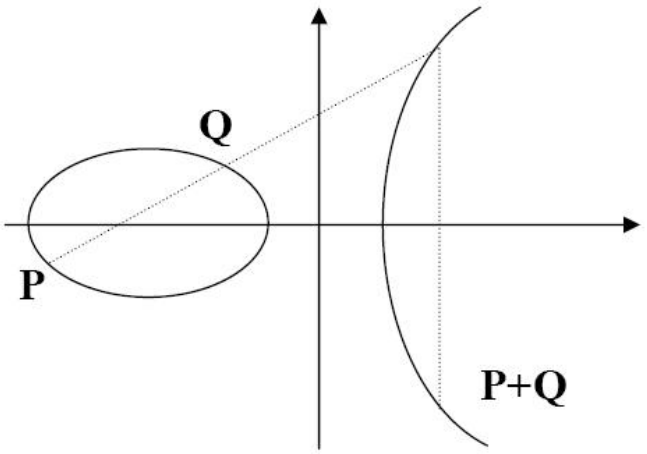
2. Metode kriptiranja

Danas se u primjeni nalazi velik broj kriptografskih metoda. Jedna od metoda koja posjeduje određene prednosti i nedostatke je i metoda zasnovana na korištenju eliptičnih krivulja. Uz eliptične krivulje, u ovom poglavlju obrađene su i trenutno aktualne metode kriptiranja pri čemu je detaljnije obrađena RSA metoda kojoj primjena eliptičnih krivulja treba biti alternativa.

2.1. Eliptične krivulje

Na bazi diskretnog logaritamskog problema za eliptične krivulje (ECDPL) nastao je Digitalni algoritam potpisa eliptične krivulje (ECDSA) kao alternativa RSA algoritmu. U prvom redu ECDSA se promatra kao alternativa za RSA zbog kraćih ključeva koji pružaju jednaku razinu sigurnosti pa tako 160 (210) bitni ECC ključ pruža jednaku zaštitu kao RSA 1024 (2048) bitni ključ, a ta prednost raste s povećanjem veličine ključeva.

Za primjer je uzeta krivulja iz sustava konačnih polja primarnih brojeva (F_p) jer ju je jednostavnije objasniti od polinomne krivulje. Ono što čini eliptičnu krivulju prikladnom za upotrebu u kriptografiji je njeno matematičko svojstvo koje kaže da ako se odaberu dvije različite točke na krivulji, tada pravac koji ih spaja presijeca krivulju u trećoj točki. Ako se ta točka reflektira na x-os dobiva se još jedna točka (krivulja je simetrična u odnosu na x-os). Pod uvjetom da su poznate točke P i Q, moguće je naći točku refleksije, koja je na slici označena kao $P + Q$. Pokazalo se da $P + Q$ zadovoljava najčešća matematička svojstva koja se pojavljuju uz cijele brojeve, pod uvjetom da je definirana točka beskonačnosti, što je u slučaju cijelih brojeva 0.



Slika 1: Eliptična krivulja

Moguće je definirati oblik matematičkih operacija nad točkama eliptične krivulje (plus točka beskonačnosti) što omogućava normalne matematičke operacije. Vezano uz to, ako se točke P i Q podudaraju, moguće je definirati točku refleksije $P + P$, koja se označava kao $2P$. Proširenjem te logike, moguće je odrediti kP, za bilo koji cijeli broj k.

Iz tog slijedi definicija ECDLP-a: ako postoji bazna točka P, te točka kP na krivulji, treba pronaći vrijednost k. Vjeruje se kako je za pogodne eliptične krivulje i bazne točke ovo jako težak problem.

2.1.1. Matematička analiza

Konačna polja F_q (eng. *finite fields*)

Sastoje se od ograničenog broja elemenata polja te operacija zbrajanja i množenja koja se mogu obaviti između neka dva elementa. Broj elemenata u polju odgovara vrijednosti q. F_q će se sastojati od q elemenata onda i samo onda kada je q primarni broj, te kada za svaki F_q postoji samo jedno polje.

Eliptične krivulje koje se koriste u kriptografiji, definirane su dvjema tipovima konačnih polja: poljem neparnih brojeva (F_p – konačna polja primarnih brojeva, gdje je p>3 veliki primarni broj) i poljem po

bazi 2 (F_{2^m} – konačna polja brojeva po bazi 2). Ako razlika između polja nije važna, oba polja se definiraju kao F_q, gdje $q=p$ ili $q=2^m$.

Konačna polja primarnih brojeva (F_p)

Sadrži p elemenata koji su cijeli brojevi {0, 1, ... , p-1} s operacijama zbrajanja i množenja koje su definirane na sljedeći način:

- Definicija zbrajanja: ako je $a,b \in F_p$, tada $a+b=r$ unutar F_p, gdje je $r \in [0, p-1]$ ostatak kada je cijeli broj $a+b$ podijeljen s p. Ovo je poznato kao $a+b=r \pmod p$.
- Definicija množenja**:** ako je $a,b \in F_p$, tada $a.b=s$ unutar F_p, gdje je $s \in [0, p-1]$ ostatak kada je cijeli broj $a*b$ podijeljen s p. Ovo je poznato kao $a.b=s \pmod p$.

Konačna polja po bazi 2 (F_{2^m})

Sadrži 2^m elemenata gdje je $m \geq 1$, a skup je poput polinoma razine m-1 ili niže $\{a_{m-1}x^{m-1}+a_{m-2}x^{m-2}+ ...+ a_1x+ a_0 : a_i \in \{0,1\}\}$ s operacijama zbrajanja i množenja koje su definirane na sljedeći način:

- Definicija zbrajanja: ako $a = a_{m-1}x^{m-1} +...+a_0$, $b = b_{m-1}x^{m-1} +...+b_0 \in F_{2^m}$, tada je $a+b=r$ unutar F_{2^m}, gdje je $r = r_{m-1}x^{m-1} +...+r_0$, pri čemu je $r_i = a_i + b_i \pmod 2$.
- Definicija množenja: ako $a = a_{m-1}x^{m-1} +...+a_0$, $b = b_{m-1}x^{m-1} +...+b_0 \in F_{2^m}$, tada je $a.b=s$ unutar F_{2^m}, gdje je $s = s_{m-1}x^{m-1} +...+s_0$, ostatak kada se polinom $a.b$ podijeli s f(x) uz modulo 2.

Analiza krivulje za konačno polje

Eliptična krivulja leži u dvodimenzionalnom polju i definirana je jednadžbom $y^2 = x^3 + ax + b$. Iz polja $F_q \times F_q$ odabiru se dvije točke $(x,y) \in F_q$, koje zadovoljavaju jednadžbu i točku beskonačnosti 0.

- Ako krivulja leži u polju F_p, definicija glasi: $y^2 = x^3 + ax + b$, gdje $a \in F_p$ i $b \in F_p$ su konstante tako da $4.a^3 + 27.b^2 \neq 0 \pmod p$.
- Ako krivulja leži u polju F_{2^m}, definicija glasi: $y^2 + xy = x^3 + ax^2 + b$, gdje $a \in F_{2^m}$ i $b \in F_{2^m}$ su konstante i $b \neq 0$.

Točka beskonačnosti 0 se, iako nema koordinate, koristi na način da ne zadovoljava jednadžbu $0=(0,0)$ ako je $b \neq 0$, ili u protivnom $0=(0,1)$.

Za svake dvije točke na krivulji, gdje $P \in F_q$ i $Q \in F_q$ moguće je naći treću točku $S = P + Q \in F_q$ tako da određena relacije vrijede za sve točke na krivulji:

- $(P + Q) + R = P + (Q + R)$ (asocijativnost)
- $P + 0 = 0 + P = P$ (neutralni element)
- postoji $(-P)$ tako da $(-P) + P = P + (-P) = 0$ (inverz)
- $P + Q = Q + P$ (komutativnost)

Negativni oblik točke $P = (x,y)$ definira se kao $-P = (x, -y)$ za $P \in F_p$, te $-P = (x,x+y)$ za $P \in F_{2^m}$.

Pravila zbrajanja:

- ako je $Q = 0$ tada je $P + Q = P$
- ako je $Q = -P$ tada je $P + Q = 0$
- ako je $Q = P$ tada je $P + Q = R$ gdje
  - slučaj F_p: $x_r = \lambda^2 - 2x_p$, $y_r = \lambda(x_p - x_r) - y_p$, $\lambda = (3x_p^2 + a) / (2y_p)$
  - slučaj F_{2^m}: $x_r = \lambda^2 + \lambda + a$, $y_r = x_r^2 + (\lambda + 1)x_r$, $\lambda = x_p + y_p/x_p$
- ako $Q \neq P$ tada $P + Q = R$ gdje
  - slučaj F_p: $x_r = \lambda^2 - x_p - x_q$, $y_r = \lambda(x_p - x_r) - y_p$, $\lambda=(y_q - y_p)/(x_q - x_p)$
  - slučaj F_{2^m}: $x_r = \lambda^2 + \lambda + x_p + x_q + a$, $y_r = \lambda(x_p + x_r) + x_r + y_p$, $\lambda=(y_p + y_q)/(x_p+x_q)$

Problem diskretnog logaritma (DLP)

Za zadanu konačnu Abelovu grupu F, koja sadrži veliku podgrupu prostog reda, i za $p, q \in F$, pri čemu je q generator velike podgrupe od F, potrebno je pronaći najmanji prirodni broj x takav da je: $p=q^x$. Ovaj

problem naziva se problemom diskretnog logaritma. Trenutno ne postoji efikasan algoritam za računanje diskretnog logaritma, osim u specijalnim slučajevima.
Uvođenjem eliptičnih krivulja, problem se mijenja. Ukoliko je zadana eliptična krivulja E i točka $P \in E$ reda n i ako je zadana točka $Q \in E$ formulom $Q=mP$, gdje je $m \in \{2, 3, \ldots, n\text{-}2\}$, potrebno je pronaći broj m za koji prethodna naredba vrijedi. Kada su E i P ispravno odabrani, rješavanje ECDPL-a se smatra nemogućim. Potrebno je naglasiti kako je jedan od uvjeta da je n, red točke P, toliko velik da je teško provjeriti sve mogućnosti od m.

2.2. RSA (Rivest, Shamir, Adleman) algoritam

Algoritam za kriptiranje javnih ključeva idejno je predstavljen 1977. godine od strane Ron Rivesta, Adi Shamira i Len Adlemana s Tehnološkog Instituta Massachusetts (MIT), a 1983. godine je i patentiran.
RSA je prvi poznati algoritam pogodan za operacije digitalnog potpisa i kriptiranja te predstavlja veliki pomak u kriptografiji javnih ključeva. Zbog svoje sigurnosti vrlo je raširen u protokolima elektroničkog poslovanja.
U svom radu, RSA koristi dva ključa: javni i privatni ključ. Javni ključ može biti poznat svakome, a koristi se za kriptiranje poruke. Te poruke mogu biti dekriptirane samo korištenjem privatnog ključa što znači da bilo tko može kriptirati poruke, dok samo onaj koji posjeduje privatni ključ može otvoriti poruku i pročitati ju.

Generiranje ključeva
Koraci koje pošiljatelj poruke mora provesti kako bi dobio javni i privatni ključ su sljedeći:

- slučajni odabir dva velika primarna broja p i q tako da je $p \neq q$,
- izračun $n=p*q$,
- izračun $\varphi(n)$ (ukupan broj pozitivnih cijelih broja koji su primarni i nalaze se unutar broja n) gdje $\varphi(n)=(p\text{-}1)(q\text{-}1)$,
- odabir cijelog broja e tako da $1<e<\varphi(n)$,
- izračun privatnog ključa d tako da je $d*e \equiv 1 \pmod{\varphi(n)}$.

Javni ključ sastoji se od modula n i javnog eksponenta e, a privatni od modula n i privatnog eksponenta d. Pošiljatelj šalje javni ključ primatelju dok privatni zadržava za potrebe dekripcije.

Shema pripreme poruka (eng. *padding sheme*)
RSA u praksi koristi jedan od oblika pripreme poruka kako niti jedna vrijednost poruke ne bi rezultirala u nesigurno šifriranom tekstu. Bez korištenja pripreme, RSA kriptirana poruka može rezultirati sa sigurnosnim problemima kao što su:

1) Vrijednosti $m=0$ i $m=1$ uvijek proizvode šifrirani tekst 0 ili 1 zbog eksponenta.
2) Kada se koristi mali eksponent i male vrijednosti m, ne-modularni rezultat je manji od modula n što znači da se šifrirani tekst može lako probiti tako što se uzme e-ti korijen šifriranog teksta bez obzira na modul.
3) S obzirom da je RSA deterministično kriptiranje, napadač može uspješno izvesti napad na kripto-sustav kriptirajući vjerojatne oblike teksta preko javnog ključa u šifrirani tekst te ih spremiti u bazu podataka (rječnik). U slučaju da se šifrirani tekstovi podudaraju, napadač može pomoću rječnika pročitati poruku.

Prva dva problema mogu nastati kada se šalje kratka ASCII poruka, npr. vrijednost znaka je 0 i tada je šifrirani tekst 0. Kako bi se izbjegli ovi problemi koristi se slučajna priprema poruke u vrijednost m prije samog kriptiranja, čime se osigurava da poruka ne dospije u područje gdje će biti ranjiva. Manja poruka će biti kriptirana u veliki broj mogućih šifriranih tekstova te će se time povećati sigurnost.
Standardi poput Kriptografskog standarda javnog ključa (eng. *Public Key Cryptography Standards-PKCS*) su dizajnirani da na siguran način pripreme poruke prije nego ih kodiraju pomoću RSA. Iako pripreme poruka malim porukama dodaju određen broj bitova te ih tako povećavaju, originalna poruka je i dalje jako mala te može podlijeći sofisticiranim napadima koji predviđaju strukturu poruke. U svrhu toga koristi se sigurnosna tehnika Optimalno asimetrično kriptiranje pripeme poruka (eng. OAEP - *Optimal Asymmetric Encryption Padding*) te PKCS-ova tehnika RSA-PPS (eng. *Probabilistic Signature Scheme*).

2.2.1. Problemi i praktična razmatranja

Sigurnost

RSA koristi 1024-2048 bitne ključeve koji još nikada nisu razbijeni no vjeruje se kako bi se to moglo dogoditi u bliskoj budućnosti. 512 bitni modul može biti razbijen za nekoliko sati korištenjem osobnog računala, a teoretsko rješenje TWIRL (eng. *The Weizmann Institute Relation Locator*) dovelo je u pitanje sigurnost 1024 bitnih modula. Danas se preporuča duljina ključa od najmanje 2048 bita.

Generiranje ključeva

Traženje velikih primarnih brojeva p i q najčešće se radi testiranjem slučajnih brojeva odabrane veličine korištenjem testova koji vrlo brzo eliminiraju ne-primarne brojeve. Da bi modul n bio valjan, brojevi p i q ne smiju biti blizu jedan drugome. Također ako (p-1) ili (q-1) imaju samo male primarne brojeve, modul n se može izračunati vrlo brzo te takve p i q treba odbaciti.
Metoda generiranja primarnih brojeva p i q mora biti slučajna i nepredvidiva, tako da uvelike poveća sigurnost. Ako je i parcijalno predvidiva, napadač može pronaći polovicu brojeva, a nakon toga vrlo brzo i ostalu polovicu. Vrlo je važno da je tajni ključ d dovoljno velik. Pokazalo se da ako se broj p nalazi između q i $2q$ te ako je $d < (n^{1/4})/3$, tada se d može izračunati iz n i e. NIST (eng. *National Institute of Standards and Technology*) je zato propisao da radi sigurnosti javni eksponent e ne smije biti manji od 65537.

Brzina

RSA je mnogo sporiji od DES-a i ostalih simetričnih kriptosustava. U praksi pošiljatelj mora kriptirati poruku simetričnim algoritmom, kriptirati simetrični ključ putem RSA te poslati oboje primatelju. Ovakav način rada donosi određene sigurnosne probleme te je jako važno koristiti vrlo jaki generator slučajnih brojeva za simetrične ključeve jer bi u protivnom napadač mogao premostiti RSA pogađanjem simetričnog ključa.

Distribucija ključeva

Za svaki kriptografski sustav važna je sigurna distribucija ključeva tako da se ne može presresti od strane napadača koji bi ih mogao pročitati i promijeniti. Kada bi napadač mogao osobi A poslati svoj ključ i uvjeriti ju da je on osoba B te da može presresti poruku između osobe B i A, tada bi mogao čitati njihove poruke i slati im izmijenjene poruke bez da oni znaju za njegovo postojanje. Obrana protiv ovakvog napada se bazira na digitalnim certifikatima ili drugim komponentama infrastrukture javnog ključa.

Vremenski napadi

Kada bi napadač znao dovoljno detalja o sklopovskoj (eng. *hardware*) strukturi računala osobe A te vremenu potrebnom za dekriptiranje nekoliko poznatih šifriranih tekstova, mogao bi izračunati dekripcijski ključ d relativno brzo. Napad se može izvesti i protiv RSA sheme potpisa, a moguća obrana sastoji se od korištenja konstantnog vremena dekripcije za svaki šifrirani tekst, što uvelike umanjuje performanse. Zato većina RSA implementacija koristi alternativu poznatu kao kriptografsko zasljepljivanje (eng. *cryptographic blinding*) koje umjesto da prvo računa $c^d \bmod(n)$, najprije odabire slučajnu vrijednost r i izračunava $(r^e c)^d \bmod(n)$. Rezultat je $rm \bmod(n)$ koji se množi s recipročnom vrijednosti r broja kako bi se isti maknuo. Za svaki šifrirani tekst koristi se novi r.

2.3. Ostale metode kriptiranja

DSA (eng. *Digital Signature Algorithm*)
DSA je još jedna metoda za generiranje digitalnih potpisa uz RSA, koja je zasnovana na problemu diskretnog logaritma u multiplikativnoj grupi konačnog polja. DSA je predložio 1991. godine NIST, specificiran je od U.S. Government Federal Information, a patentiran je 1993. godine.

DES (eng. *Data Encryption Standard*)
Metoda simetričnog kriptiranja podataka razvijena od strane američkog NIST-a 1970. Radi na principu upotrebe nizova znakova fiksne duljine od bitova običnog teksta te ih kroz mnogo kompliciranih

operacija transformira u bitni niz znakova šifriranog teksta jednake duljine. U ovom slučaju to je 64 bita, gdje se 56 bita koristi u samom algoritmu dok ostalih 8 bitova služi za provjeru parnosti te kasnije budu odbačeni.
Danas se smatra kako je DES nesiguran za mnoge aplikacije ponajprije zbog male veličine ključeva (56-bita), što dovodi do probijanja zaštite u manje od 24 sata ako se koristi tzv. *brute force* napad zasnovan na procesorskoj moći računala. Nekada se vjerovalo kako se povećana sigurnost može postići korištenjem metode trostrukog DES-a, ali proteklih godina izbačen je iz upotrebe od strane svog nasljednika AES-a (eng. *Advanced Encryption Standard*).

AES (eng. *Advanced Encryption Standard*)
Poznat još kao i Rijndael, razvijen je od strane dva belgijska programera-matematičara Joan Daemena i Vincenta Rijmena. 2001. godine nakon 5 godina standardizacije prihvaćen je od NIST-a te predstavlja zamjenu za DES. Posjeduje veliku brzinu rada kod sklopovlja i programa, relativno laganu implementaciju te zahtjeva malo memorije.
AES koristi ključeve veličine 128, 192 ili 256 bita, a većina kalkulacija se odvija u specijalnom konačnom polju brojeva. Radi u poljima 4x4 okteta (eng. *bytes*), a svaka runda kriptiranja sastoji se o 4 koraka (dodaj-modificirani-ključ, zamjena okteta, pomicanje redova, pomicanje stupaca).
Od 2006. godine AES je podložan tzv. *side channel* napadima (napadi koji nisu bazirani na propustima u algoritmu već na informacijama o fizičkoj implementaciji kripto-sustava: trošenju električne energije, vremenima izračuna podataka, i sl.). Najčešće su napadani AES podaci s reduciranim brojem rundi kriptiranja, jer budući da AES koristi 10 rundi za 128-bitne ključeve, 12 za 192-bitne, 14 za 256-bitne, napadi se obavljaju s reduciranim brojem rundi: 7 za 128-bitne, 8 za 192 bitne i 9 za 256-bitne ključeve. Zbog toga kriptografi sumnjaju u sigurnost AES-a.

IDEA (eng. *International Dana Encryption Standard*)
IDEA je kriptosustav koji su razvili švicarski kriptografi Xuejia Lai i James Massey. Prvu verziju zvanu PES (eng. *Proposed Encryption Standard*) su objavili 1990. godine, ali taj kriptosustav nije bio otporan na diferencijalnu kriptoanalizu (za 128-bitni ključ je trebalo 264 operacija), pa su nakon Biham-Shamirovog otkrića, autori 1992. godine prepravili algoritam i nazvali ga IDEA. IDEA koristi 128-bitni ključ za šifriranje 64-bitnih blokova otvorenog teksta. Koristi tri operacije na 16-bitnim podblokovima (XOR, zbrajanje modulo 216, množenje modulo 216). Zahvaljujući kompleksnosti svojih operacija, IDEA algoritam stručnjaci smatraju jednim od najsigurnijih iz područja simetričnih algoritama.

3. Primjena eliptičnih krivulja

3.1. Kriptosustavi koji koriste eliptične krivulje

Asimetrični kriptosustav je uređena petorka (***P***, ***C***, ***U***, ***E*** , ***D***) za koju vrijedi:

- ***P*** je konačan skup svih jasnih tekstova,
- ***C*** je konačan skup svih mogućih šifrata,
- ***U*** je konačan skup svih mogućih korisnika,
- svaki korisnik $U \in \boldsymbol{U}$ posjeduje par $(E_U, D_U) \in \boldsymbol{E} \times \boldsymbol{D}$, gdje su $E_U : \boldsymbol{P} \rightarrow \boldsymbol{C}$ i $D_U : \boldsymbol{C} \rightarrow \boldsymbol{P}$ funkcije enkripcije i dekripcije takve da vrijedi $D_U(E_U(x)) = x$ za svaki jasan teskt $x \in \boldsymbol{P}$.

Kriptosustavi koji koriste eliptične krivulje su sljedeći:

- EC ElGamalov kriptosustav (eng. *Elliptic Curve ElGamal encription*) – originalni sustav je ElGamal predstavio 1985. godine, a temelji se na problemu diskretnog logaritma. EC ElGamalov kriptosustav služi samo za kriptiranje i dekriptiranje, dok autentikacija nije moguća. Kriptosustav nikad nije standardiziran jer nije jednostavno pretvoriti niz bitova u točku krivulje.
- Menzes – Vanstoneov kriptosustav i ECES (eng. *Elliptic Curve Encryption System*) koji u osnovi imaju istu metodologiju enkripcije i dekripcije pri čemu se samo poruka razlikuje.
- Demytkov kriptosustav – 1993. godine Demytko je predstavio kriptosustav temeljen na eliptičnim krivuljama analogan RSA algoritmu, koji koristi teoriju komplementarnih grupa

eliptičnih krivulja, a temelji se na traženju brojeva p i q, $n = pq$, pri čemu su p i q veliki prosti brojevi.

- KMOV kriptosustav – Koyama, Maurer, Okamoto i Vanstone predstavili su 1991. godine kriptosustave zasnovane na eliptičnim krivuljama nad prstenom Z_n, gdje n nije prost broj, i na problemu faktoriziranja velikih brojeva. Predstavili su tri nove sheme od kojih je najvažnija KMOV shema kod koje postoje određena ograničenja vezana za brojeve p i q, ($n = pq$).
- Kuwokado – Koyama kriptosustav – temelji se na odabiru prostih brojeva p i q takvih da je faktoriziranje $n = pq$ nemoguće i u ovisnosti o brojevima p i q računaju se i ostali parametri sustava.

3.2. Digitalni potpis

Digitalni potpis je takva metoda koja se koristi za provjeru porijekla i utvrđivanje besprijekornosti informacije. Pri tome je potrebno da zadovoljava određene zahtjeve: vjerodostojnost potpisanog dokumenta (nepromjenjivost dokumenta), nemogućnost ponovnog korištenja jednom generiranog potpisa na drugom mjestu, nemogućnost krivotvorenja potpisa, nemogućnost izbjegavanja odgovornosti za potpisani dokument.

Metode izračuna digitalnih potpisa baziranih na eliptičnim krivuljama:

- ECDSA (eng. *Elliptic Curve Digital Signature Algorithm*) - varijanta DSA algoritma koja u svom radu koristi eliptične krivulje. Ova varijanta s eliptičnim krivuljama pruža manju veličinu ključa s približno jednakim razinama sigurnosti i vremenom obrade te identičnom duljinom generiranog sažetka, kao i DSA. Konkretno, DSA s 1024-bitnim p i 160-bitnim q, i ECDSA nad skupom 160-bitnog primitivnog polja, generiraju za obje metode 320-bitni potpis u samo nekoliko milisekundi. ECDSA je prihvaćen kao ANSI standard 1999. godine, 1998. godine kao ISO standard, a 2000. godine i kao IEEE i NIST standardi.
- ECSS (eng. *Elliptic Curves Signature Scheme*) - još jedan mehanizam digitalnog potpisa koji nije toliko poznat kao prethodni.
- EC Nyberg-Rueppelova shema digitalnog potpisa.
- OFF shema digitalnog potpisa – Okamoto, Fujioka i Fujisaki su 1992. godine predstavili shemu digitalnog potpisa utemeljenu na eliptičnim krivuljama nad Z_n, gdje je $n=p^2q$, a brojevi p i q su prosti.

3.2.1. ECDSA algoritam

Pošto je ECDSA najpopularniji od svih drugih primjena eliptičnih krivulja u kriptografiji, u ovom poglavlju je opisan njegov rad. ECDSA kriptosustav $\boldsymbol{D}=(q, a, b, P, n, h)$ definiran je sa sljedećih šest elemenata:

- $q=p$ ili $q=2^m$, p je prosti broj,
- a, b su elementi polja koji određuju jednadžbu eliptične krivulje,
- P je točka na krivulji $E(F_q)$, $P=(x_p, y_p)$,
- n je red točke P, najmanji pozitivni cijeli broj takav da je $nP=0$,
- $h = \#E(F_q)/n$.

Generiranje ključeva

Prilikom generiranja ključeva potrebno je učiniti sljedeće:

1. Izabrati slučajni broj d iz intervala [1, n-1].
2. Izračunati $Q = d^*P$.

Javni ključ je točka Q, a privatni ključ je broj d. Naknadno se ispituje i ispravnost javnog ključa $Q=(x_q, y_q)$ na sljedeći način:

- provjeriti da je $Q = 0$,
- provjeriti da su x_q i y_q na ispravan način predstavljeni elementi konačnog polja Fq,
- provjeriti da se točka Q nalazi na eliptičnoj krivulji određenoj brojevima a i b,
- provjeriti da je $nQ = 0$.

Generiranje potpisa

Da bi se potpisala poruka m, potrebno je sljedeće:

1. Izabrati slučajni broj k iz intervala [1, n-1].
2. Izabrati $kP=(x,y)$ i $r = x \bmod n$ (ako je $r = 0$, slijedi povratak na prvi korak).
3. Izračunati $t = k^{-1} \bmod n$.
4. Izračunati $e = \text{SHA-1}(m)$, gdje SHA-1 predstavlja 160 bitnu *hash* funkciju.
5. Koristeći privatni ključ d izračunati $s = k^{-1} (e + rd) \bmod n$ (ako je $s = 0$, slijedi povratak na prvi korak).

Provjera potpisa

Primatelj provjerava primljeni potpis (r, s) poslane poruke m. Pri tome su mu poznati parametri kriptosustava $\boldsymbol{D}$ kao i pošiljateljev javni ključ Q, i kako bi provjerio potpis potrebne su sljedeće akcije:

1. Provjeriti da su r i s brojevi iz intervala [1, n-1].
2. Izračunati $e = \text{SHA-1}(m)$.
3. Izračunati $w = s^{-1} \bmod n$.
4. Izračunati $u_1 = e\,s^{-1} \bmod n$ i $u_2 = r\,s^{-1} \bmod n$.
5. Izračunati točku $X = (x_1, y_1) = u_1 P + u_2 Q$.
6. Ako je $X = 0$, potrebno je odbiti potpis, inače izračunati $v = x_1 \bmod n$.
7. Prihvatiti potpis za poruku m, ako i samo ako je $v = r$.

3.3. Protokoli za razmjenu tajnog ključa

Protokoli za uspostavu simetričnog kriptosustava (eng. *key establishment protocol*) omogućavaju i sigurnu razmjenu ključeva od čega su poznata dva oblika:

- Protokoli za slanje ključeva (eng. *key transfer*) – kod ovih protokola jedna strana sigurnim putem šalje tajni ključ drugoj strani. Pri tome pošiljatelj koristi javni ključ primatelja kako bi sakrio odabrani tajni ključ, a primatelj sa svojim privatnim ključem može dekriptirati poruku.
- Protokoli za dogovor oko ključeva (eng. *key agreement*) – protokoli kod kojih oba sudionika na jednak način sudjeluju u izračunavanju ključeva.

Protokoli za razmjenu tajnog ključa zasnovani na eliptičnim krivuljama:

- ECDH (eng. *Elliptic Curve Diffie-Hellman protocol*) - verzija Diffie-Hellmanovog protokola za razmjenu ključeva u kojoj se grupe temelje na eliptičnim krivuljama. Diffie-Hellmanov protokol predstavili su 1976. godine Whitfield Diffie i Martin Hellman s glavnom tezom kako je u nekim matematičkim strukturama potenciranje puno jednostavnije nego logaritmiranje.
- EC Nyberg-Rueppelov protokol za razmjenu ključeva - izveden je iz prethodno spomenute Nyberg-Rueppelove sheme digitalnog potpisa.

3.4. Sustavi za raspodjelu ključeva

Kad dva sudionika žele uspostaviti simetrični sustav, trebaju prvo razmijeniti tajni ključ nekim od postojećih protokola za razmjenu ključeva. Ali kad je sigurnu komunikaciju potrebno uspostaviti između određene grupe korisnika (N), javlja se problem pohrane tajnih ključeva jer bi svaki sudionik morao čuvati N-1 ključeva. Taj problem je moguće riješiti uspostavom centra za raspodjelu ključeva (eng. KDC – *Key Distribution Center*). KDC predstavlja pouzdani poslužitelj kojem svi sudionici vjeruju i koji je zaštićen od vanjskih opasnosti. Centar svakom sudioniku pridjeljuje njegov identifikator i tajni ključ.

Sakazaki-Okamoto-Mamba je sustav za raspodjelu ključeva temeljen je na identifikatoru koji koristi eliptične krivulje nad poljem Z_n (n je produkt dva različita prosta broja p i q, oba veća od 3). Predstavili su ga H. Sakazaki, E. Okamoto i M. Mamba 1997. godine.

4. Implementacija

Osnovna operacija na kojoj se zasniva ECC je množenje točaka (eng. *point multiplication*), definirana preko operacija nad ograničenim poljima. Slijedeći dio opisuje množenje točaka samo na poljima primarnih cijelih brojeva (F_p) budući da binarna polja polinoma (F_{2^m}) nisu dovoljno dobro podržana današnjim procesorima pa bi uzrokovale slabije performanse. Multiplikacija kP cijelog broja k i točke P

na eliptičnoj krivulji $y^2=x^3+ax+b$ u polju F_p gdje su $a,b \in F_p$, može biti razdvojena na sekvence zbrajanja i dupliranja točaka. Sljedeće optimizacijske tehnike su najvažnije za množenje točaka, a standardizirane su od strane NIST organizacije:

- Sustav projiciranih koordinata (eng. *projective coordinate system*) - smatra se kako sustav koji se dobiva kombiniranjem modificiranog Jacobiana i sličnih koordinata pruža najbolje performanse.
- Ne-susjedne forme (eng. *non adjacent forms*) - metoda snimanja skalara k u množenju točaka kP kako bi se smanjio broj bitova koji nisu nula te reducirao broj množenja.
- Specifična optimizacija krivulja (eng. *curve specific optimizations*) - NIST i SECG specificirali su skup eliptičkih krivulja s provjerenim sigurnosnim postavkama koje omogućuju velike optimizacije performansi.

Modularno množenje i kvadriranje velikih cijelih brojeva su kritično važne operacije za RSA i ECC. Zbog toga se mora posvetiti posebna pažnja optimizaciji tih operacija. Na slabijim procesorima mnogostruka množenja velikih cijelih brojeva zahtijeva ne samo aritmetičke operacije nego i znatno velike prijenose podataka prema i iz memorije zbog limitiranog prostora registara. Vrijeme računanja se može optimizirati tako da se smanji broj ne-aritmetičkih operacija, osobito memorijskih operacija, pri čemu se za optimiranje množenja koriste sljedeće metode:

- Množenje redaka **-** strategija ostavlja množitelj b_i konstantnim i množi ga sa cijelom grupom mutiplikanata iz jednog retka (a_{n-1}, ..., a_1, a_0) prije nego prijeđe na drugi množitelj b_{i+1}. Parcijalni produkti se zbrajaju u akumulator koji se sastoji od n registara (n = broj redaka) (r_{n-1}, ...,r_1, r_0). Kada se završi s prvim retkom, suma se pohranjuje u registar (r_0 za prvi red) i kreće se na slijedeći red. Pri ovakvom obliku množenja, koristi se $2n+1$ registara te se obavlja $4n$ memorijskih operacija.
- Množenje stupaca **-** zbraja stupce parcijalnih produkta $a_j * b_i$, gdje je $i+j=l$ za stupac l. Na kraju svakog stupca jedan k-bitni rezultat se sprema kao dio krajnjeg rezultata množenja. Ova procedura zahtijeva $4+[\log_2(n)/k]$ registra što je najmanje od sva tri algoritma. Pošto s povećanjem operanda n broj registra raste neznatno, ova metoda množenja pogodna je za arhitekture s ograničenim brojem registara. Ipak, pošto se tijekom rada izvodi $2n^2+2n$ memorijskih operacija, to rezultira s dvostrukim brojem memorijskih operacija po $k \times \underline{k}$ množenju.
- Hibridno množenje **-** strategija koja kombinira prednosti prethodne dvije tehnike. Cilj ove metode množenja je optimiziranje broja registara i broja memorijskih operacija.

5. Specifičnosti eliptičnih krivulja

Performanse i sigurnost

U sljedećoj tablici prikazane su veličine ključeva različitih algoritama u bitovima koje pružaju jednaku razinu sigurnosti (Izvor [13]). Npr. za zaštitu 128 bitnog simetričnog AES ključa potrebno je koristiti 3072 bitni RSA. Ako se za isti ključ koriste algoritmi eliptičnih krivulja, potrebno je koristiti 256 bitni ključ.

Simetrični algoritmi	DSA/RSA/Diffie-Hellman	Eliptične krivulje
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Tablica 1: Duljine kriptografskih ključeva u bitovima prema preporuci NIST organizacije

Sigurnost koju algoritmi eliptičnih krivulja pružaju nije jedina prednost eliptičnih krivulja u odnosu na RSA i Diffie-Hellman algoritme. Naime, algoritmi eliptičnih krivulja su i manje računski zahtjevni od spomenutih algoritama iako imaju složenije operacije po bitu ključa. Na sljedećoj tablici prikazan je odnos broja računskih operacija za Diffie-Hellman algoritam te za algoritme bazirane na eliptičnim krivuljama (Izvor [13]).

Sigurnosna razina (bitovi)	Omjer složenosti operacija Diffie-Hellman algoritma i eliptičnih krivulja
80	3:1
112	6:1
128	10:1
192	32:1
256	64:1

Tablica 2: Odnos složenosti operacija Diffie-Hellman algoritma i eliptičnih krivulja prema različitim razinama sigurnosti

Kompleksnost diskretnog logaritamskog problema

Veliki problem predstavlja to što prava kompleksnost diskretnog logaritamskog problema za eliptične krivulje nije u potpunosti jasna. Istraživanja su pokazala kako neke krivulje za koje se vjerovalo da su pogodne za ECC, to ipak nisu bile. U slučaju anomalija u krivuljama, kada je bazna točka P jednaka primarnom broju p, ECDLP se može jednostavno riješiti. Dodatna istraživanja se rade po tom pitanju.

Generiranje krivulja

Kada se definira sustav eliptične krivulje, potrebo je definirati krivulju i baznu točku P (ta dva elementa nisu tajni i mogu biti isti za sve korisnike). Za danu krivulju i baznu točku relativno je jednostavno generirati privatni ključ (slučajni cijeli broj k) i javni ključ (točka kP na krivulji). Pri tome je jako teško generirati odgovarajuću krivulju i bazni P.

Glavni problem je kako pronaći ukupan broj točaka na krivulji. Čak i kada se to napravi treba odrediti odgovarajuću baznu točku P koja mora imati veliki slijed kojim će osigurati kompleksnost ECDLP-a. Pri tome P mora podijeliti broj točaka na krivulji. Sve to pokazuje da je pronalaženje odgovarajućeg baznog P veoma složen proces.

Nekompatibilni sistemi

Implementacija „parnih" i „neparnih" krivulja je slična, ali dovoljno različita da osigura njihovu nekompatibilnost. U neparnim krivuljama dolazi do problema uzrokovanog razlikama u prezentaciji krivulje i baznog P, što dovodi do pogrešaka u komunikaciji između korisnika ako oni koriste različite prezentacije. To je u suprotnosti s RSA algoritmom kod kojeg su (u teoriji) sve korisničke implementacije kompatibilne.

Procesiranje

Sistem eliptičnih krivulja koristi mnogo manje ključeve nego ostali sistemi kriptiranja danas u upotrebi. Kako to u praksi izgleda pokazuje tablica koja uspoređuje brzinu generiranja i verificiranja potpisa između RSA i ECDSA. Test je rađen na dva paralelno povezana Motorola 56303 procesora s brzinom takta 66Mhz. RSA ključ koristi javni eksponent e=65537

Algoritam	Generiranje potpisa	Provjera potpisa
RSA (1024 bit)	25 ms	< 2 ms
ECDSA (160 bit)	32 ms	33 ms
RSA (2048 bit)	120 ms	5 ms
ECDSA (216 bit)	68 ms	70 ms

Tablica 3: Vremenski utrošak generiranja i provjere potpisa RSA i ECDSA

Iz prethodne tablice vidljivo je kako s povećanjem veličine ključa, generiranje potpisa ECDSA algoritma postaje uvelike brže u odnosu na RSA algoritam (Izvor [7]). S druge strane, verifikacija potpisa ECDSA algoritma je uvelike sporija od verifikacije potpisa RSA algoritma. Razlog zašto je RSA znatno sporiji kod generiranja ključeva je u tome što RSA računa velike primarne brojeve. S druge

strane, ECDSA za generiranje ključa treba samo stvoriti slučajni broj koji postaje korisnikov tajni ključ te provesti operacije za izračun javnog ključa.
Vrijeme koje je potrebno za provjeru potpisa korištenjem ECDSA algoritma može imati značajne negativne učinke na performanse sustava. Mnogu sustavi imaju velik broj udaljenih uređaja koji komuniciraju sa središnjim poslužiteljem. Vrijeme koje je potrebno da bi udaljeni uređaji mogli generirati potpise ne mora biti nužno bitno, ali poslužitelj mora biti sposoban brzo provjeriti primljene potpise. Stoga su RSA sustavi u nekim slučajevima prikladniji od algoritama baziranih na eliptičnim krivuljama čak i kad su performanse u pitanju.

Potrošak energije na bežičnim mrežama
Eksperiment je izveden na Berkley/Crossbow Mica2dots platformi popularnoj za istraživanja o bežičnim mrežama. Korišteni su Atmel Atmega128L 8-bitni mikrokontroler frekvencije 4MHz i Chipcon CC1000 bežični prijemnik male snage.

Algoritam	Potpisivanje		Razmjena ključa	
	Potpis	Provjera	Klijent	Poslužitelj
RSA (1024 bit)	304	11,9	15,4	304
ECDSA (160 bit)	22,82	45,09	22,3	22,3
RSA (2048 bit)	2302,7	53,7	57,2	2302,7
ECDSA (224 bit)	61,54	121,98	60,4	60,4

Tablica 4: Utrošak energije digitalnog potpisa i razmjene ključa u [mJ]

Tablica 4 uspoređuje potrošenu energiju RSA i ECDSA algoritama za generiranje i provjeru potpisa te trošak energije razmjene ključeva ne uključujući autentikaciju i provjeru certifikata (Izvor [4]). Dok je potrošnja energije za provjeru potpisa kod RSA jako malen, potrošnja energije kod potpisa je mnogostruko veća. Za usporedbu, ECDSA potpisi imaju znatno manju potrošnju energije u usporedbi s RSA. Prelaskom s 1024 bitnog na 2048 bitni RSA, trošak energije potpisa se povećava gotovo 7 puta, dok je ECDSA 224 bitni potpis samo 3 puta veći od 160 bitnog ECDSA. Također kod RSA algoritma, razmjena ključa ovisi o poslužitelju koji treba kriptirati slučajno generirani tajni ključ s javnim ključem korisnika koji potom mora dekriptirati ključ pomoću svog privatnog ključa. Stoga proces razmjene ključa na poslužitelju troši znatno više energije nego kod klijenta. Nasuprot tome, kod eliptičnih krivulja oba korisnika moraju obaviti jednu ECDH operaciju kako bi dobili tajni ključ pa je potrošnja energije jednaka.

Intelektualno vlasništvo
Usprkos brojnim prednostima eliptičnih krivulja pa i prihvaćenosti od brojnih korisnika, smatra se da eliptične krivulje nisu dovoljno implementirane u praksi zbog velikog broja prijavljenih patenata nad eliptičnim krivuljama. Samo Certicom Inc, kanadska tvrtka, posjeduje pravo na preko 300 patenata povezanih s eliptičnim krivuljama i javnom kriptografijom.
Kako bi omogućio korištenje eliptičnih krivulja u svrhu zaštite tajnih vladinih i američkih informacija, NSA (National Security Agency) kupila je od Certicom organizacije licencu koja pokriva sve njihovo intelektualno vlasništvo s ograničenom mogućnošću korištenja. Licenca pokriva implementacije od državnog sigurnosnog značaja koje su certificirane pod FIPS 140-1 ili su odobrene od NSA. Nadalje, licenca je ograničena samo na polja primarnih brojeva gdje je primarni broj veći od 2255. Na NIST-ovoj listi krivulja, 3 od 15 eliptičnih krivulja odgovaraju spomenutoj svrsi primjene: primarne krivulje s prostim brojevima od 256, 384 i 521 bitova. Certicom je izdao 26 patenata koji pokrivaju ovo područje, a NSA ima pravo na ograničeno korištenje. Komercijalne tvrtke mogu dobiti pravo na korištenje od NSA organizacije pod uvjetima njihove licence ili mogu kupiti posebnu licencu od Certicom organizacije.

6. Zaključak

Kriptografski sustavi zasnovani na eliptičnim krivuljama smatraju se alternativom za RSA algoritam, a tek nakon toga i zamjenom. Zahvaljujući svojim performansama i malenoj potrošnji električne energije, eliptične krivulje najveću prednost iskazuju u primjeni malih uređaja s ograničenom procesorskom snagom i memorijom te će u tom segmentu sigurno brzo naći svoje mjesto. Te aplikacije uključuju mobilne uređaje, pametne kartice, bankarske aplikacije, aplikacije za elektroničko poslovanje, itd.

Algoritmi bazirani na eliptičnim krivuljama posjeduju snažnije sigurnosne parametre od prve generacije javnih ključeva (RSA, Diffie-Hellman) koje su trenutno u upotrebi. Unatoč određenim parametrima koji narušavaju sigurnost algoritama baziranih na eliptičnih krivulja i koje je potrebno izbjegavati, algoritmi bazirani na eliptičnim krivuljama posjeduju jednaku neprobojnost kao i kad su predstavljeni 1985. godine. Ipak, uz te probleme vezane uz sigurnost, postoje i određeni problemi poput nekompatibilnosti implementacije, teškoća pri generiranju prikladnih krivulja, relativno spore potvrde potpisa, patentnih prava i sl.

Eliptične krivulje tek moraju izaći na tržište, a vrijeme će pokazati kolika je njihova vrijednost.

7. Reference

[1] Standards for efficient cryptography - SEC 1: Elliptic Curve Cryptography, Certicom research, 2000,

[2] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, Sheueling Chang: Comparing Eliptic Curve Cryptography and RSA on 8-bit CPU, Sun Microsistem Laboratories, 2004.

[3] Don Johnson, Alfred Menezes, Scott Vanstone: The Elliptic Curve Digital Signature Algorithm (ECDSA), Certicom Research, 2001

[4] Arvinderpal S. Wander†, Nils Gura, Hans Eberle, Vipul Gupta, Sheueling Chang Shantz: Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks, University of California, Santa Cruz, Sun Microsystems Laboratories, 2004.

[5] Hans Eberle, Nils Gura, Sheueling Chang Shantz, Vipul Gupta, Leonard Rarick: A Public-key Cryptographic Processor for RSA and ECC, Sun Microsystems Laboratories, 2004.

[6] Elliptic curves – A new generation of public key tehniques, http://www.cryptomathic.com/company/elliptic.html, rujan 2006.

[7] Thales e-Security: Eliptic Curve Cryptography, 2000.

[8] Marcel Maretić: Eliptične krivulje u kriptografiji - diplomski rad, Sveučilište u Zagrebu, Prirodoslovno matematički fakultet, svibanj 2002.

[9] Vlatka Krivačić: Kriptiranje eliptičnim krivuljama, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, rujan 2004.

[10] Julio Lopez, Ricardo Dahab: Performance of Elliptic Curve Cryptosystem, Institute of Computing, State University of Campinas, 2000.

[11] Julio Lopez, Ricardo Dahab: An Overview of Elliptic Curve Cryptography, Institute of Computing, State University of Campinas, 2000.

[12] DI Management Services Pty Limited: RSA algorithm, http://www.di-mgt.com.au/rsa_alg.html, rujan 2006.

[13] The Case for Elliptic Curve Cryptography, http://www.nsa.gov/ia/industry/crypto_elliptic_curve.cfm, rujan 2006.

[14] Kristin Lauter: The advantages of Elliptic Curve Cryptography for Wireless Security, 2004.

[15] Certicom Inc., Certicom Research: http://www.certicom.com/index.php?action=ecc,res_home, rujan 2006.