

КРИПТОГРАФИЈА

- ЧЕТВРТИ ДЕО -

ДОЦ. ДР ДРАГАН ЂОКИЋ

Математички факултет, Универзитет у Београду

`dragan.djokic@matf.bg.ac.rs`

7. март 2024.

РСА КРИПТОСИСТЕМ (РИВЕСТ-ШАМИР-ЕЈДЛМАН)

Алиса жели да пошаље поруку или кључ Бобану

- ▶ Бобан тајно бира два велика проста броја p и q , множи их $n = pq$ и рачуна $\varphi(n) = (p - 1)(q - 1)$

РСА КРИПТОСИСТЕМ (РИВЕСТ-ШАМИР-ЕЈДЛМАН)

Алиса жели да пошаље поруку или кључ Бобану

- ▶ Бобан тајно бира два велика проста броја p и q , множи их $n = pq$ и рачуна $\varphi(n) = (p - 1)(q - 1)$
- ▶ Затим Бобан бира број $1 \leq e \leq \varphi(n)$ тд. $\text{НЗД}(e, \varphi(n)) = 1$ и рачуна $d = e^{-1} \bmod \varphi(n)$

РСА КРИПТОСИСТЕМ (РИВЕСТ-ШАМИР-ЕЈДЛМАН)

Алиса жели да пошаље поруку или кључ Бобану

- ▶ Бобан тајно бира два велика проста броја p и q , множи их $n = pq$ и рачуна $\varphi(n) = (p - 1)(q - 1)$
- ▶ Затим Бобан бира број $1 \leq e \leq \varphi(n)$ тд. $\text{НЗД}(e, \varphi(n)) = 1$ и рачуна $d = e^{-1} \bmod \varphi(n)$
- ▶ Бобан шаље Алиси јавни кључ (n, e) , а d чува као свој тајни кључ. У овом тренутку Бобан може да заборави p и q , али је битно да их не објављује

РСА КРИПТОСИСТЕМ (РИВЕСТ-ШАМИР-ЕЈДЛМАН)

Алиса жели да пошаље поруку или кључ Бобану

- ▶ Бобан тајно бира два велика проста броја p и q , множи их $n = pq$ и рачуна $\varphi(n) = (p - 1)(q - 1)$
- ▶ Затим Бобан бира број $1 \leq e \leq \varphi(n)$ тд. $\text{НЗД}(e, \varphi(n)) = 1$ и рачуна $d = e^{-1} \bmod \varphi(n)$
- ▶ Бобан шаље Алиси јавни кључ (n, e) , а d чува као свој тајни кључ. У овом тренутку Бобан може да заборави p и q , али је битно да их не објављује
- ▶ Ако је $M < n$ кодирана порука, Алиса рачуна $N = M^e \bmod n$ и шаље Бобану

РСА КРИПТОСИСТЕМ (РИВЕСТ-ШАМИР-ЕЈДЛМАН)

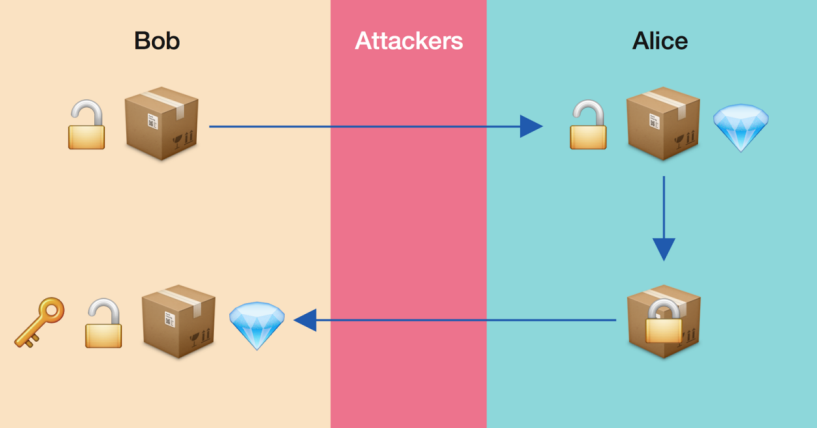
Алиса жели да пошаље поруку или кључ Бобану

- ▶ Бобан тајно бира два велика проста броја p и q , множи их $n = pq$ и рачуна $\varphi(n) = (p - 1)(q - 1)$
- ▶ Затим Бобан бира број $1 \leq e \leq \varphi(n)$ тд. $\text{НЗД}(e, \varphi(n)) = 1$ и рачуна $d = e^{-1} \bmod \varphi(n)$
- ▶ Бобан шаље Алиси јавни кључ (n, e) , а d чува као свој тајни кључ. У овом тренутку Бобан може да заборави p и q , али је битно да их не објављује
- ▶ Ако је $M < n$ кодирана порука, Алиса рачуна $N = M^e \bmod n$ и шаље Бобану
- ▶ Бобан има тајни кључ d помоћу кога лако рачуна $N^d \equiv M^{ed} \equiv M \pmod{n}$. Овде се користи $ed \equiv 1 \pmod{\varphi(n)}$ и Ојлерова теорема

РСА КРИПТОСИСТЕМ (РИВЕСТ-ШАМИР-ЕЈДЛМАН)

Алиса жели да пошаље поруку или кључ Бобану

- ▶ Бобан тајно бира два велика проста броја p и q , множи их $n = pq$ и рачуна $\varphi(n) = (p - 1)(q - 1)$
- ▶ Затим Бобан бира број $1 \leq e \leq \varphi(n)$ тд. $\text{НЗД}(e, \varphi(n)) = 1$ и рачуна $d = e^{-1} \pmod{\varphi(n)}$
- ▶ Бобан шаље Алиси јавни кључ (n, e) , а d чува као свој тајни кључ. У овом тренутку Бобан може да заборави p и q , али је битно да их не објављује
- ▶ Ако је $M < n$ кодирана порука, Алиса рачуна $N = M^e \pmod{n}$ и шаље Бобану
- ▶ Бобан има тајни кључ d помоћу кога лако рачуна $N^d \equiv M^{ed} \equiv M \pmod{n}$. Овде се користи $ed \equiv 1 \pmod{\varphi(n)}$ и Ојлерова теорема
- ▶ Цица види n , e и N , али не може да дође до поруке M све док не одреди d тј. $\varphi(n)$




```
class RSA:

    def __init__(self):

        limit = 256
        p = get_prime(2**limit)
        q = get_prime(2**limit)
        self.p = p
        self.q = q

        self.n = self.p * self.q

        self.phi = (self.p - 1) * (self.q - 1)

        while True:
            e = random.randrange(2, self.phi - 1)
            gcd, m, n = extended_gcd(e, self.phi)

            if gcd == 1:
                self.e = e
                break

        self.d = mod_inv(self.e, self.phi)

    def encrypt(self, m, e):
        return mod_pow(m, e, self.n)

    def decrypt(self, me):
        return mod_pow(me, self.d, self.n)
```

```
def main():
    A = RSA()
    B = RSA()

    m1 = 207

    me = A.encrypt(m1, B.e)
    print("Sifrovano: ", me)

    med = B.decrypt(me)
    print("Desifrovano: ", med)
```

Променљива A је класе RSA да би имала имплементирану функцију криптовања, њени кључеви n , e и d се не користе

ДИГИТАЛНИ ПОТПИС ПОМОЋУ RSA

Како Алиса може да убеди Бобана у свој идентитет? Тј. како да Бобан буде сигуран да не добија поруке од Цице?

ДИГИТАЛНИ ПОТПИС ПОМОЋУ RSA

Како Алиса може да убеди Бобана у свој идентитет? Тј. како да Бобан буде сигуран да не добија поруке од Цице?

- ▶ Алиса генерише јавни кључ (n, e) и свој тајни кључ d као у RSA

ДИГИТАЛНИ ПОТПИС ПОМОЋУ RSA

Како Алиса може да убеди Бобана у свој идентитет? Тј. како да Бобан буде сигуран да не добија поруке од Цице?

- ▶ Алиса генерише јавни кључ (n, e) и свој тајни кључ d као у RSA
- ▶ Бобан генерише (неку насумичну) поруку M и шаље Алиси $M_1 = M^e \bmod n$

ДИГИТАЛНИ ПОТПИС ПОМОЋУ RSA

Како Алиса може да убеди Бобана у свој идентитет? Тј. како да Бобан буде сигуран да не добија поруке од Цице?

- ▶ Алиса генерише јавни кључ (n, e) и свој тајни кључ d као у RSA
- ▶ Бобан генерише (неку насумичну) поруку M и шаље Алиси $M_1 = M^e \bmod n$
- ▶ Алиса треба да декриптује M_1 помоћу свог кључа d и врати Бобану $M_2 = M_1^d \bmod n$

ДИГИТАЛНИ ПОТПИС ПОМОЋУ RSA

Како Алиса може да убеди Бобана у свој идентитет? Тј. како да Бобан буде сигуран да не добија поруке од Цице?

- ▶ Алиса генерише јавни кључ (n, e) и свој тајни кључ d као у RSA
- ▶ Бобан генерише (неку насумичну) поруку M и шаље Алиси $M_1 = M^e \bmod n$
- ▶ Алиса треба да декриптује M_1 помоћу свог кључа d и врати Бобану $M_2 = M_1^d \bmod n$
- ▶ Бобан пореди M_2 са оригиналном поруком M

ДИГИТАЛНИ ПОТПИС ПОМОЋУ RSA

Како Алиса може да убеди Бобана у свој идентитет? Тј. како да Бобан буде сигуран да не добија поруке од Цице?

- ▶ Алиса генерише јавни кључ (n, e) и свој тајни кључ d као у RSA
- ▶ Бобан генерише (неку насумичну) поруку M и шаље Алиси $M_1 = M^e \bmod n$
- ▶ Алиса треба да декриптује M_1 помоћу свог кључа d и врати Бобану $M_2 = M_1^d \bmod n$
- ▶ Бобан пореди M_2 са оригиналном поруком M
- ▶ Цица не може (пре Алисе) да одговори Бобану шта је M

ДИГИТАЛНИ ПОТПИС ПОМОЋУ RSA

Како Алиса може да убеди Бобана у свој идентитет? Тј. како да Бобан буде сигуран да не добија поруке од Цице?

- ▶ Алиса генерише јавни кључ (n, e) и свој тајни кључ d као у RSA
- ▶ Бобан генерише (неку насумичну) поруку M и шаље Алиси $M_1 = M^e \bmod n$
- ▶ Алиса треба да декриптује M_1 помоћу свог кључа d и врати Бобану $M_2 = M_1^d \bmod n$
- ▶ Бобан пореди M_2 са оригиналном поруком M
- ▶ Цица не може (пре Алисе) да одговори Бобану шта је M
- ▶ Цица ће на крају видети поруку M_2 , али не може да је употреби јер Бобан у следећој провери генерише ново M

ДИГИТАЛНИ ПОТПИС ПОМОЋУ RSA

Како Алиса може да убеди Бобана у свој идентитет? Тј. како да Бобан буде сигуран да не добија поруке од Цице?

- ▶ Алиса генерише јавни кључ (n, e) и свој тајни кључ d као у RSA
- ▶ Бобан генерише (неку насумичну) поруку M и шаље Алиси $M_1 = M^e \bmod n$
- ▶ Алиса треба да декриптује M_1 помоћу свог кључа d и врати Бобану $M_2 = M_1^d \bmod n$
- ▶ Бобан пореди M_2 са оригиналном поруком M
- ▶ Цица не може (пре Алисе) да одговори Бобану шта је M
- ▶ Цица ће на крају видети поруку M_2 , али не може да је употреби јер Бобан у следећој провери генерише ново M

Другачије улоге (у односу на претх.): Алиса \leftrightarrow Бобан

СЛАЊЕ ПОРУКЕ + ДИГИТАЛНИ ПОТПИС

Алиса треба да пошаље поруку Бобану, а да Бобан буде сигуран у њен идентитет

СЛАЊЕ ПОРУКЕ + ДИГИТАЛНИ ПОТПИС

Алиса треба да пошаље поруку Бобану, а да Бобан буде сигуран у њен идентитет

- ▶ Алиса генерише јавни кључ (n_A, e_A) и тајни кључ d_A као у RSA
- ▶ Бобан генерише јавни кључ (n_B, e_B) и тајни кључ d_B као у RSA

СЛАЊЕ ПОРУКЕ + ДИГИТАЛНИ ПОТПИС

Алиса треба да пошаље поруку Бобану, а да Бобан буде сигуран у њен идентитет

- ▶ Алиса генерише јавни кључ (n_A, e_A) и тајни кључ d_A као у RSA
- ▶ Бобан генерише јавни кључ (n_B, e_B) и тајни кључ d_B као у RSA
- ▶ $M < n_A, n_B$ кодирана Алисина порука

СЛАЊЕ ПОРУКЕ + ДИГИТАЛНИ ПОТПИС

Алиса треба да пошаље поруку Бобану, а да Бобан буде сигуран у њен идентитет

- ▶ Алиса генерише јавни кључ (n_A, e_A) и тајни кључ d_A као у RSA
- ▶ Бобан генерише јавни кључ (n_B, e_B) и тајни кључ d_B као у RSA
- ▶ $M < n_A, n_B$ кодирана Алисина порука
- ▶ Алиса рачуна $M_1 = M^{d_A} \bmod n_A$ и $M_2 = M_1^{e_B} \bmod n_B$ и шаље Бобану M_2

СЛАЊЕ ПОРУКЕ + ДИГИТАЛНИ ПОТПИС

Алиса треба да пошаље поруку Бобану, а да Бобан буде сигуран у њен идентитет

- ▶ Алиса генерише јавни кључ (n_A, e_A) и тајни кључ d_A као у RSA
- ▶ Бобан генерише јавни кључ (n_B, e_B) и тајни кључ d_B као у RSA
- ▶ $M < n_A, n_B$ кодирана Алисина порука
- ▶ Алиса рачуна $M_1 = M^{d_A} \bmod n_A$ и $M_2 = M_1^{e_B} \bmod n_B$ и шаље Бобану M_2
- ▶ Бобан рачуна $M_3 = M_2^{d_B} \bmod n_B$ и $M_4 = M_3^{e_A} \bmod n_A$, и управо M_4 ће бити Алисина порука

СЛАЊЕ ПОРУКЕ + ДИГИТАЛНИ ПОТПИС

Алиса треба да пошаље поруку Бобану, а да Бобан буде сигуран у њен идентитет

- ▶ Алиса генерише јавни кључ (n_A, e_A) и тајни кључ d_A као у RSA
- ▶ Бобан генерише јавни кључ (n_B, e_B) и тајни кључ d_B као у RSA
- ▶ $M < n_A, n_B$ кодирана Алисина порука
- ▶ Алиса рачуна $M_1 = M^{d_A} \bmod n_A$ и $M_2 = M_1^{e_B} \bmod n_B$ и шаље Бобану M_2
- ▶ Бобан рачуна $M_3 = M_2^{d_B} \bmod n_B$ и $M_4 = M_3^{e_A} \bmod n_A$, и управо M_4 ће бити Алисина порука

Битан је редослед операција:

- ▶ $M_3 \equiv_{n_B} M_2^{d_B} \equiv_{n_B} M_1^{e_B d_B} \equiv_{n_B} M_1 \implies M_3 = M_1$
- ▶ И слично ће бити $M_4 = M$


```
def main():  
    A = RSA()  
    B = RSA()  
  
    m = 207  
    m1 = A.decrypt(m)  
    m2 = A.encrypt(m1, B.e)  
    print("Sifrovano: ", m2)  
    m3 = B.decrypt(m2)  
    m4 = B.encrypt(m3, A.e)  
    print("Desifrovano: ", m4)
```

- ▶ Одредити тајни кључ \iff наћи инверз за множење $\cdot \varphi(n)$
 $\iff \varphi(n) = ? \iff$ раставити $n \iff$ наћи прави
делилац од n

КРИПТОАНАЛИЗА RSA

- ▶ Одредити тајни кључ \iff наћи инверз за множење $\cdot \varphi(n)$
 $\iff \varphi(n) = ? \iff$ раставити $n \iff$ наћи прави
делилац од n
- ▶ Елементарно решето (провера да ли $p|n$ за $p \leq \sqrt{n}$) је
преспоро - време криптовања је $O(\log^3 n)$

КРИПТОАНАЛИЗА RSA

- ▶ Одредити тајни кључ \iff наћи инверз за множење $\cdot \varphi(n)$
 $\iff \varphi(n) = ? \iff$ раставити $n \iff$ наћи прави
делилац од n
- ▶ Елементарно решето (провера да ли $p|n$ за $p \leq \sqrt{n}$) је
преспоро - време криптовања је $O(\log^3 n)$
- ▶ Основна претпоставка RSA је да не постоји ефикасан
начин да се реши један од претходних проблема (а самим
тим и сви)

КРИПТОАНАЛИЗА RSA

- ▶ Одредити тајни кључ \iff наћи инверз за множење $\cdot \varphi(n)$
 $\iff \varphi(n) = ? \iff$ раставити $n \iff$ наћи прави
делилац од n
- ▶ Елементарно решето (провера да ли $p|n$ за $p \leq \sqrt{n}$) је
преспоро - време криптовања је $O(\log^3 n)$
- ▶ Основна претпоставка RSA је да не постоји ефикасан
начин да се реши један од претходних проблема (а самим
тим и сви)
- ▶ Показаћемо два алгорита који раде ефикасно за неке
специфичне n

- ▶ Претпоставља да је број n производ два проста броја
сличне величине

- ▶ Претпоставља да је број n производ два проста броја сличне величине
- ▶ Напада случај који делује најоптималније за избор кључа n у RSA - тада би елементарним решетом трагали до \sqrt{n}

ФЕРМАНОВ МЕТОД

- ▶ Претпоставља да је број n производ два проста броја сличне величине
- ▶ Напада случај који делује најоптималније за избор кључа n у РСА - тада би елементарним решетом трагали до \sqrt{n}
- ▶ $n = pq = s^2 - t^2$, где су $s = \frac{p+q}{2}$ и $t = \frac{p-q}{2}$ природни бројеви

ФЕРМАОВ МЕТОД

- ▶ Претпоставља да је број n производ два проста броја сличне величине
- ▶ Напада случај који делује најоптималније за избор кључа n у RSA - тада би елементарним решетом трагали до \sqrt{n}
- ▶ $n = pq = s^2 - t^2$, где су $s = \frac{p+q}{2}$ и $t = \frac{p-q}{2}$ природни бројеви
- ▶ Проблем се своди на налажење s и t , при чему је $s > \sqrt{n}$ и t мало

ФЕРМАОВ МЕТОД

- ▶ Претпоставља да је број n производ два проста броја сличне величине
- ▶ Напада случај који делује најоптималније за избор кључа n у РСА - тада би елементарним решетом трагали до \sqrt{n}
- ▶ $n = pq = s^2 - t^2$, где су $s = \frac{p+q}{2}$ и $t = \frac{p-q}{2}$ природни бројеви
- ▶ Проблем се своди на налажење s и t , при чему је $s > \sqrt{n}$ и t мало
- ▶ У низу $s_1 = [\sqrt{n}] + 1, s_2 = [\sqrt{n}] + 2, \dots, s_i = [\sqrt{n}] + i, \dots$ тражимо најмањи s_i тд. је $s_i^2 - n$ потпун квадрат тј.
 $t_i = \sqrt{s_i^2 - n}$ цео број
(где је $[\cdot]$ цео део)

ФЕРМАОВ МЕТОД

- ▶ Претпоставља да је број n производ два проста броја сличне величине
- ▶ Напада случај који делује најоптималније за избор кључа n у RSA - тада би елементарним решетом трагали до \sqrt{n}
- ▶ $n = pq = s^2 - t^2$, где су $s = \frac{p+q}{2}$ и $t = \frac{p-q}{2}$ природни бројеви
- ▶ Проблем се своди на налажење s и t , при чему је $s > \sqrt{n}$ и t мало
- ▶ У низу $s_1 = [\sqrt{n}] + 1, s_2 = [\sqrt{n}] + 2, \dots, s_i = [\sqrt{n}] + i, \dots$ тражимо најмањи s_i тд. је $s_i^2 - n$ потпун квадрат тј.
 $t_i = \sqrt{s_i^2 - n}$ цео број
(где је $[\cdot]$ цео део)
- ▶ Тада је $p = s_i + t_i$ и $q = s_i - t_i$

Пример: разложить 3229799

$$\sqrt{3229799} = 1797,164\dots$$

$$\sqrt{1798^2 - 3229799} = \sqrt{3232804 - 3229799} = \sqrt{3005} \notin \mathbb{Z}$$

$$\sqrt{1799^2 - 3229799} = \sqrt{3236401 - 3229799} = \sqrt{6602} \notin \mathbb{Z}$$

$$\sqrt{1800^2 - 3229799} = \sqrt{3240000 - 3229799} = \sqrt{10201} = 101$$

$$3229799 = (1800 - 101)(1800 + 101) = 1699 \cdot 1901$$

Пример: разложить 3229799

$$\sqrt{3229799} = 1797,164\dots$$

$$\sqrt{1798^2 - 3229799} = \sqrt{3232804 - 3229799} = \sqrt{3005} \notin \mathbb{Z}$$

$$\sqrt{1799^2 - 3229799} = \sqrt{3236401 - 3229799} = \sqrt{6602} \notin \mathbb{Z}$$

$$\sqrt{1800^2 - 3229799} = \sqrt{3240000 - 3229799} = \sqrt{10201} = 101$$

$$3229799 = (1800 - 101)(1800 + 101) = 1699 \cdot 1901$$

Пример: разложить 1357

$$\sqrt{1357} = 36,837\dots$$

$$\sqrt{37^2 - 1357} = \sqrt{12} \notin \mathbb{Z} \quad \sqrt{40^2 - 1357} = \sqrt{243} \notin \mathbb{Z}$$

$$\sqrt{38^2 - 1357} = \sqrt{87} \notin \mathbb{Z} \quad \sqrt{41^2 - 1357} = \sqrt{324} = 18$$

$$\sqrt{39^2 - 1357} = \sqrt{164} \notin \mathbb{Z} \quad 1357 = (41 - 18)(41 + 18) = 23 \cdot 59$$

За имплементацију користимо функцију `isqrt(n)` која рачуна $\lfloor \sqrt{n} \rfloor$ (и која постоји у `python-y`):

```
def fermat(n):
    s=isqrt(n)+1
    t=isqrt((t*t)-n)
    while((t*t)!=((s*s)-n)):
        s+=1
        t=isqrt((t*t)-n)
    p=s+t
    q=s-t
    return p,q
```

ДЕФИНИЦИЈА

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B -гладак ако важи $p_i^{\alpha_i} \leq B$, за све $1 \leq i \leq k$

ДЕФИНИЦИЈА

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B -гладак ако важи $p_i^{\alpha_i} \leq B$, за све $1 \leq i \leq k$

- ▶ Пример: $70 = 2 \cdot 5 \cdot 7$ је 7-гладак, али $150 = 2 \cdot 3 \cdot 5^2$ није 7-гладак

ДЕФИНИЦИЈА

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B -гладак ако важи $p_i^{\alpha_i} \leq B$, за све $1 \leq i \leq k$

- ▶ Пример: $70 = 2 \cdot 5 \cdot 7$ је 7-гладак, али $150 = 2 \cdot 3 \cdot 5^2$ није 7-гладак
- ▶ Полардов метод омогућава да (брзо) факторишемо природан број n . Претпоставке
 - ▶ $B \ll n$ нпр. $B \sim \log n$
 - ▶ n има прост чинилац p тд. $p - 1$ је B -гладак
(не знамо p , само претпостављамо да постоји)

ДЕФИНИЦИЈА

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B -гладак ако важи $p_i^{\alpha_i} \leq B$, за све $1 \leq i \leq k$

- ▶ Пример: $70 = 2 \cdot 5 \cdot 7$ је 7-гладак, али $150 = 2 \cdot 3 \cdot 5^2$ није 7-гладак
- ▶ Полардов метод омогућава да (брзо) факторишемо природан број n . Претпоставке
 - ▶ $B \ll n$ нпр. $B \sim \log n$
 - ▶ n има прост чинилац p тд. $p - 1$ је B -гладак
(не знамо p , само претпостављамо да постоји)
- ▶ Пре примене Полардовога алгоритма треба израчунати $m = \text{НЗС}(1, 2, \dots, B)$

ДЕФИНИЦИЈА

Нека су $N, B \in \mathbb{N}$. За број $N = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ кажемо да је B -гладак ако важи $p_i^{\alpha_i} \leq B$, за све $1 \leq i \leq k$

- ▶ Пример: $70 = 2 \cdot 5 \cdot 7$ је 7-гладак, али $150 = 2 \cdot 3 \cdot 5^2$ није 7-гладак
- ▶ Полардов метод омогућава да (брзо) факторишемо природан број n . Претпоставке
 - ▶ $B \ll n$ нпр. $B \sim \log n$
 - ▶ n има прост чинилац p тд. $p - 1$ је B -гладак
(не знамо p , само претпостављамо да постоји)
- ▶ Пре примене Полардовога алгоритма треба израчунати $m = \text{НЗС}(1, 2, \dots, B)$
 - ▶ Уместо m може се користити и $B!$

Како се рачуна $m = \text{НЗС}(1, 2, \dots, B)$?

$$m = \prod_{\substack{p \text{ ПРОСТ} \\ p \leq B}} p^{\lfloor \log_p B \rfloor}$$

У канонској факторизацији m учествују само прости бројеви p који деле неки од $1, 2, \dots, B$, па је $p \leq B$. Ако је $\alpha = \alpha_p$ највећи степен тд. $p^\alpha \mid m$ онда p^α дели неки од $1, 2, \dots, B$. Зато је $p^\alpha \leq B$ тј. $\alpha \leq \log_p B$.

Како се рачуна $m = \text{НЗС}(1, 2, \dots, B)$?

$$m = \prod_{\substack{p \text{ ПРОСТ} \\ p \leq B}} p^{\lfloor \log_p B \rfloor}$$

У канонској факторизацији m учествују само прости бројеви p који деле неки од $1, 2, \dots, B$, па је $p \leq B$. Ако је $\alpha = \alpha_p$ највећи степен тд. $p^\alpha \mid m$ онда p^α дели неки од $1, 2, \dots, B$. Зато је $p^\alpha \leq B$ тј. $\alpha \leq \log_p B$.

Овде проналажење простих чинилаца p захтева $O(B)$ операција, остало се брзо извршава

Како се рачуна $m = \text{НЗС}(1, 2, \dots, B)$?

$$m = \prod_{\substack{p \text{ ПРОСТ} \\ p \leq B}} p^{\lfloor \log_p B \rfloor}$$

У канонској факторизацији m учествују само прости бројеви p који деле неки од $1, 2, \dots, B$, па је $p \leq B$. Ако је $\alpha = \alpha_p$ највећи степен тд. $p^\alpha \mid m$ онда p^α дели неки од $1, 2, \dots, B$. Зато је $p^\alpha \leq B$ тј. $\alpha \leq \log_p B$.

Овде проналажење простих чинилаца p захтева $O(B)$ операција, остало се брзо извршава

Сви B -глатки бројеви деле m

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
- ▶ $p - 1 | m$ и МФТ $a^{p-1} \equiv 1 \pmod{p}$ повлаче $a^m \equiv 1 \pmod{p}$

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
- ▶ $p - 1 | m$ и МФТ $a^{p-1} \equiv 1 \pmod{p}$ повлаче $a^m \equiv 1 \pmod{p}$
- ▶ $g = \text{НЗД}(n, a^m - 1)$ је дељив са p , па g не може бити један

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
- ▶ $p - 1 | m$ и МФТ $a^{p-1} \equiv 1 \pmod{p}$ повлаче $a^m \equiv 1 \pmod{p}$
- ▶ $g = \text{НЗД}(n, a^m - 1)$ је дељив са p , па g не може бити један
- ▶ Ако $g \neq n$ онда ће g бити прави делилац од n који тражимо.

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
- ▶ $p - 1 | m$ и МФТ $a^{p-1} \equiv 1 \pmod{p}$ повлаче $a^m \equiv 1 \pmod{p}$
- ▶ $g = \text{НЗД}(n, a^m - 1)$ је дељив са p , па g не може бити један
- ▶ Ако $g \neq n$ онда ће g бити прави делилац од n који тражимо.
- ▶ важно: g не зависи од p , зависи само од n и a (које можемо изабрати произвољно)

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
- ▶ $p - 1 | m$ и МФТ $a^{p-1} \equiv 1 \pmod{p}$ повлаче $a^m \equiv 1 \pmod{p}$
- ▶ $g = \text{НЗД}(n, a^m - 1)$ је дељив са p , па g не може бити један
- ▶ Ако $g \neq n$ онда ће g бити прави делилац од n који тражимо.
- ▶ важно: g не зависи од p , зависи само од n и a (које можемо изабрати произвољно)
- ▶ Ако је $g = n$ треба пробати са другим a

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
 - ▶ Одабрати $2 \leq a \leq n - 1$ тд. $\text{НЗД}(a, n) = 1$ нпр. $a = 2$

ПОЛАРДОВ АЛГОРИТАМ

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
 - ▶ Одабрати $2 \leq a \leq n - 1$ тд. $\text{НЗД}(a, n) = 1$ нпр. $a = 2$
 - ▶ Израчунати $x = a^m - 1 \pmod n$

ПОЛАРДОВ АЛГОРИТАМ

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
 - ▶ Одабрати $2 \leq a \leq n - 1$ тд. $\text{НЗД}(a, n) = 1$ нпр. $a = 2$
 - ▶ Израчунати $x = a^m - 1 \pmod n$
 - ▶ Ако је $x = 0$ променити a (нпр. $a + 1$) и вратити се на први корак

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
 - ▶ Одабрати $2 \leq a \leq n - 1$ тд. $\text{НЗД}(a, n) = 1$ нпр. $a = 2$
 - ▶ Израчунати $x = a^m - 1 \pmod n$
 - ▶ Ако је $x = 0$ променити a (нпр. $a + 1$) и вратити се на први корак
 - ▶ Ако је $x \neq 0$ израчунати $g = \text{НЗД}(n, x)$

ПОЛАРДОВ АЛГОРИТАМ

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
 - ▶ Одабрати $2 \leq a \leq n - 1$ тд. $\text{НЗД}(a, n) = 1$ нпр. $a = 2$
 - ▶ Израчунати $x = a^m - 1 \pmod n$
 - ▶ Ако је $x = 0$ променити a (нпр. $a + 1$) и вратити се на први корак
 - ▶ Ако је $x \neq 0$ израчунати $g = \text{НЗД}(n, x)$
- ▶ Ако је претпоставка о B -глаткости испуњена, знамо да ће алгоритам дати $g \in \{2, 3, \dots, n - 1\}$, $g|n$

ПОЛАРДОВ АЛГОРИТАМ

- ▶ Претпоставке: $B \ll n$ и n има прост чинилац p тд. $p - 1$ је B -гладак (па дели $m = \text{НЗС}(1, 2, \dots, B)$)
 - ▶ Одабрати $2 \leq a \leq n - 1$ тд. $\text{НЗД}(a, n) = 1$ нпр. $a = 2$
 - ▶ Израчунати $x = a^m - 1 \pmod n$
 - ▶ Ако је $x = 0$ променити a (нпр. $a + 1$) и вратити се на први корак
 - ▶ Ако је $x \neq 0$ израчунати $g = \text{НЗД}(n, x)$
- ▶ Ако је претпоставка о B -глаткости испуњена, знамо да ће алгоритам дати $g \in \{2, 3, \dots, n - 1\}$, $g|n$
- ▶ Уколико се појави $x = 1$ то значи да претпоставка о B -глаткости није испуњена. Тада евентуално може да се покуша са већим B

ОДРЕДИТИ ДЕЛИЛАЦ БРОЈА 5197

Нека је $B = 5$ и $m = \text{НЗС}(1, 2, 3, 4, 5) = 60$.

$$2^{60} - 1 \equiv 3416 \pmod{5197} \quad \text{и} \quad \text{НЗД}(3416, 5197) = 61$$

па је 61 делилац 5197

ОДРЕДИТИ ДЕЛИЛАЦ БРОЈА 5197

Нека је $B = 5$ и $m = \text{НЗС}(1, 2, 3, 4, 5) = 60$.

$$2^{60} - 1 \equiv 3416 \pmod{5197} \quad \text{и} \quad \text{НЗД}(3416, 5197) = 61$$

па је 61 делилац 5197

ОДРЕДИТИ ДЕЛИЛАЦ БРОЈА 187

Нека је $B = 15$ и $m = \text{НЗС}(1, 2, \dots, 15) = 360360$.

$$2^{360360} - 1 \equiv 0 \pmod{187}$$

$$3^{360360} - 1 \equiv 66 \pmod{187} \quad \text{и} \quad \text{НЗД}(66, 187) = 11$$

па је 11 делилац 187

- ▶ RSA је рањив ако је $n = pq$ тд. један од $p - 1$ или $q - 1$ је B -гладак

- ▶ RSA је рањив ако је $n = pq$ тд. један од $p - 1$ или $q - 1$ је B -гладак
- ▶ Не можемо да проверимо B -глаткост $p - 1$ јер бисмо се опет вратили на факторизацију

- ▶ RSA је рањив ако је $n = pq$ тд. један од $p - 1$ или $q - 1$ је B -гладак
- ▶ Не можемо да проверимо B -глаткост $p - 1$ јер бисмо се опет вратили на факторизацију
- ▶ Слично, алгоритми засновани на проблему дискретног логаритма могу бити рањиви Полиг-Хелмановим алгоритмом

ПОЛАРДОВ МЕТОД ПРИМЕЊЕН НА $n = 5959$, СА $B = 20$

$n = 5959 = 59 \cdot 101$, а ни $58 = 2 \cdot 29$ ни $100 = 2^2 \cdot 5^2$ нису
20-глатки.

Са друге стране, $59 - 2 = 57 = 3 \cdot 19$ јесте 20-гладак

ПОЛАРДОВ МЕТОД ПРИМЕЊЕН НА $n = 5959$, СА $B = 20$

$n = 5959 = 59 \cdot 101$, а ни $58 = 2 \cdot 29$ ни $100 = 2^2 \cdot 5^2$ нису 20-глатки.

Са друге стране, $59 - 2 = 57 = 3 \cdot 19$ јесте 20-гладак

- ▶ Да ли можемо да радимо нешто слично Полардовом методу са $p - 2$ или генерално са $p + s$, где је $s \ll p$

ПОЛАРДОВ МЕТОД ПРИМЕЊЕН НА $n = 5959$, СА $B = 20$

$n = 5959 = 59 \cdot 101$, а ни $58 = 2 \cdot 29$ ни $100 = 2^2 \cdot 5^2$ нису 20-глатки.

Са друге стране, $59 - 2 = 57 = 3 \cdot 19$ јесте 20-гладак

- ▶ Да ли можемо да радимо нешто слично Полардовом методу са $p - 2$ или генерално са $p + s$, где је $s \ll p$
- ▶ Како се појавило $p - 1$ у нашој причи?

ПОЛАРДОВ МЕТОД ПРИМЕЊЕН НА $n = 5959$, СА $B = 20$

$n = 5959 = 59 \cdot 101$, а ни $58 = 2 \cdot 29$ ни $100 = 2^2 \cdot 5^2$ нису 20-глатки.

Са друге стране, $59 - 2 = 57 = 3 \cdot 19$ јесте 20-гладак

- ▶ Да ли можемо да радимо нешто слично Полардовом методу са $p - 2$ или генерално са $p + s$, где је $s \ll p$
- ▶ Како се појавило $p - 1$ у нашој причи?
- ▶ $p - 1$ је кардиналност групе $(\mathbb{Z}_p^*, \cdot_p)$ и зато се појављује у експоненту у МФТ. Зато је тајни кључ d за RSA прављен по модулу $p - 1$

ПОЛАРДОВ МЕТОД ПРИМЕЊЕН НА $n = 5959$, СА $B = 20$

$n = 5959 = 59 \cdot 101$, а ни $58 = 2 \cdot 29$ ни $100 = 2^2 \cdot 5^2$ нису 20-глатки.

Са друге стране, $59 - 2 = 57 = 3 \cdot 19$ јесте 20-гладак

- ▶ Да ли можемо да радимо нешто слично Полардовом методу са $p - 2$ или генерално са $p + s$, где је $s \ll p$
- ▶ Како се појавило $p - 1$ у нашој причи?
- ▶ $p - 1$ је кардиналност групе $(\mathbb{Z}_p^*, \cdot_p)$ и зато се појављује у експоненту у МФТ. Зато је тајни кључ d за РСА прављен по модулу $p - 1$
- ▶ Било би добро кад бисмо имали групу кардиналности $p + s$

ПОЛАРДОВ МЕТОД ПРИМЕЊЕН НА $n = 5959$, СА $B = 20$

$n = 5959 = 59 \cdot 101$, а ни $58 = 2 \cdot 29$ ни $100 = 2^2 \cdot 5^2$ нису 20-глатки.

Са друге стране, $59 - 2 = 57 = 3 \cdot 19$ јесте 20-гладак

- ▶ Да ли можемо да радимо нешто слично Полардовом методу са $p - 2$ или генерално са $p + s$, где је $s \ll p$
- ▶ Како се појавило $p - 1$ у нашој причи?
- ▶ $p - 1$ је кардиналност групе $(\mathbb{Z}_p^*, \cdot_p)$ и зато се појављује у експоненту у МФТ. Зато је тајни кључ d за РСА прављен по модулу $p - 1$
- ▶ Било би добро кад бисмо имали групу кардиналности $p + s$
- ▶ Такве групе ће се појавити на елиптичким кривама