$\theta\beta\ell\iota c\mathbb{Z}\varepsilon$ 2023/2024

Koło Naukowe Matematyków UAM

Spis treści

1 Patryk Doniec
Logarytmy dyskretne: Od podstaw do zastosowań **5**

Logarytmy dyskretne: Od podstaw do zastosowań

Patryk Doniec
Politechnika Krakowska
Wydział Informatyki i Telekomunikacji

1.1 Wprowadzenie

Artykuł ten poświęcony jest zagadnieniu logarytmu dyskretnego, jego zastosowaniom w kryptografii, oraz metodom rozwiązywania tego problemu.

Logarytmowanie w skończonych grupach multiplikatywnych napotyka na istotne problemy obliczeniowe, co odróżnia je od klasycznego logarytmowania. Obliczenie logarytmów dyskretnych w takich grupach jest powszechnie uważane za trudne, podczas gdy odwrotna operacja – potęgowanie – może być przeprowadzona wydajnie przy użyciu metod takich jak na przykład szybkie potęgowanie. Innymi słowy, potęgowanie w odpowiednich grupach jest funkcją jednokierunkową.

Ta właściwość jest kluczowa dla kryptografii, gdyż umożliwia tworzenie kryptosystemów opartych na trudności problemu logarytmu dyskretnego. W dalszej części artykułu zostaną omówione kryptosystemy, takie jak Protokół Wymiany Klucza Diffiego-Hellmana oraz System ElGamala, które wykorzystują tę jednokierunkowość do zapewnienia bezpieczeństwa komunikacji. Artykuł przedstawi także metody służące do obliczania logarytmów dyskretnych, przeanalizujemy Algorytm Pohliga-Hellmana, oraz pokażemy jak z jego pomocą obliczyć logarytm dyskretny.

W pierwszej kolejności, musimy jednak zapoznać się z kilkoma podstawowymi definicjami potrzebnymi do dalszego opisania logarytmu dyskretnego.

Definicja 1.1. Niech $n \in \mathbb{N} : n > 1$. Zbiór $\mathbb{Z}_n^*$ to zbiór wszystkich liczb naturalnych mniejszych niż n, które są względnie pierwsze z n. Innymi słowy:

$$\mathbb{Z}_n^* = \{a \in \mathbb{N} : 1 \leqslant a < n, (a, n) = 1\}.$$

Zbiór $\mathbb{Z}_n^*$ jest **grupą multiplikatywną** modulo n, co oznacza, że dla dowolnych dwóch elementów a i b należących do $\mathbb{Z}_n^*$, ich iloczyn $ab \pmod n$ również należy do $\mathbb{Z}_n^*$, inaczej:

$$\forall_{a,b \in \mathbb{Z}_n^*} : ab \pmod n \in \mathbb{Z}_n^*,$$

oraz że istnieje element odwrotny dla każdego elementu w tym zbiorze, czyli:

$$\forall_{a \in \mathbb{Z}_n^*} \exists_{a^{-1} \in \mathbb{Z}_n^*} : aa^{-1} \equiv 1 \pmod n.$$

Definicja 1.2. Jeśli a jest elementem $\mathbb{Z}_n^*$, to **rzędem** a nazywamy najmniejsze $k \in \mathbb{N} : k > 0$, takie że:

$$a^k \equiv 1 \pmod n.$$

Rząd a modulo n jest zwykle oznaczany $\mathrm{ord}_n(a)$.

Przykład 1.3. *Weźmy $n = 5$, oraz $a = 2$. Obliczymy $ord_5(2)$. Ponieważ dla $k \in \{1, 2, 3\}$ mamy: $2^k \not\equiv 1 \pmod 5$, natomiast dla $k = 4 : 2^k \equiv 1 \pmod 5$, to $ord_5(2) = 4$.*

Definicja 1.4. Niech $\varphi : \mathbb{N} \to \mathbb{N}$ będzie funkcją przypisującą liczbie n liczbę $k \in \{1, \ldots, n\}$ liczb względnie pierwszych z n:

$$\varphi(n) = \#\{k \in \{1, \ldots, n\} : (k, n) = 1\}.$$

Funkcję φ nazywamy **funkcją Eulera**, lub inaczej **tocjentem**.

Twierdzenie 1.5 (Własności funkcji Eulera)**.** *Niech $p \in \mathbb{N}$ będzie liczbą pierwszą, oraz niech $m, n \in \mathbb{N}$ będą liczbami względnie pierwszymi, wtedy:*

$$\varphi(p) = p - 1,$$

$$\varphi(mn) = \varphi(m)\varphi(n).$$

Definicja 1.6. Jeśli $g \in \mathbb{Z}_n^*$ i jest spełniona równość:

$$\mathrm{ord}_n(g) = \varphi(n),$$

to mówimy, że g jest **pierwiastkiem pierwotnym** modulo n, lub inaczej **generatorem** $\mathbb{Z}_n^*$.

Przykład 1.7. *Weźmy $n = 5$, oraz $g = 2$. Sprawdzimy czy 2 jest pierwiastkiem pierwotnym modulo 5. W przykładzie 1.3 obliczyliśmy iż $ord_5(2) = 4$. Korzystając z twierdzenia 1.5 mamy iż $\varphi(5) = 4$. Mamy spełnioną równość:*

$$ord_5(2) = \varphi(5),$$

zatem 2 jest pierwiastkiem pierwotnym modulo 5.

Twierdzenie 1.8 (Warunek konieczny i dostateczny istnienia)**.** *Niech $k \in \mathbb{N}$. Pierwiastek pierwotny modulo n istnieje wtedy i tylko wtedy, gdy n jest jedną z następujących liczb:*

- *potęgą liczb pierwszych nieparzystych: $n = p^k$,*
- *podwojoną potęgą liczb pierwszych nieparzystych: $n = 2p^k$,*
- *liczbą 2 i 4.*

Twierdzenie 1.9. *Element g jest pierwiastkiem pierwotnym modulo n wtedy i tylko wtedy, gdy każdy element $\mathbb{Z}_n^*$ jest odpowiednią potęgą elementu g.*

1.2 Zagadnienie logarytmu dyskretnego

Definicja 1.10. Niech $a, b \in \mathbb{Z}_n^*$. Zagadnienie logarytmu dyskretnego polega na znalezieniu takiego $x \in \mathbb{N} : 0 \leqslant x < \varphi(n)$, które spełnia kongruencję:

$$b^x \equiv a \pmod{n}.$$

Wykładnik x nazywa się **logarytmem dyskretnym** a przy podstawie b modulo n i oznacza:

$$x = \log_b(a) \pmod{n}.$$

Aby mieć pewność że liczba a ma dobrze określony logarytm dyskretny, zakłada się że podstawa b jest pierwiastkiem pierwotnym modulo n.

Problem 1.11. *Weźmy $n = 557, b = 2$, oraz $a = 7$.*
Będziemy szukać takiego $x \in \mathbb{N} : 0 \leqslant x < \varphi(n)$, że:

$$x = \log_2(7) \pmod{557},$$

lub inaczej:

$$2^x \equiv 7 \pmod{557}.$$

Najprostszym rozwiązaniem, jest metoda przeliczania, czyli sprawdzenie czy dla kolejnych liczb $x = 0, 1, 2, \cdots$ zachodzi równość. Dla wybranych przez nas liczb, obliczylibyśmy w ten sposób że równość zachodzi dla $x = 458$:

$$2^{458} \equiv 7 \pmod{557}.$$

Oznacza to, iż z punktu widzenia kryptografii, dla małej liczby n musielibyśmy wykonać dużo obliczeń aby otrzmać wynik. Ogólnie w metodzie przeliczania należy wykonać $x - 1$ mnożeń modulo n aby znaleźć logarytm dyskretny.

Dochodzimy tutaj do sedna wykorzystywania logarytmów dyskrentych w kryptografii. Cecha o której powiedziliśmy sprawia że dla dużych liczb pierwszych metoda wyczerpująca jest po prostu nieprzydatna.

1.3 Protokół Diffiego-Hellmana

Pierwszym sztandarowym zastosowaniem zagadnienia logarytmu dyskretnego jest Protokół Wymiany Klucza Diffiego-Hellmana. Jego siła oparta jest na trudności obliczenia logarytmów dyskretnych w skończonych grupach multiplikatywnych.

Uzgadnianie klucza Diffiego-Hellmana jest kryptosystemem klucza publicznego. Oznacza to iż wykorzystuje niezabezpieczony kanał do bezpiecznego przekazywania informacji. Będziemy zakładać sytuację gdy Alicja(**A**) i Bob(**B**) chcą uzgodnić wspólny, tajny klucz korzystając właśnie z takiego kanału. Spójrzymy zatem na protokół wymiany klucza.

1. Użytkownicy **A** i **B** uzgadniają dużą liczbę pierwszą p, oraz liczbę g, która jest pierwiastkiem pierwotnym modulo p. Liczby p, g są znane publicznie.
2. **A** wybiera tajną liczbę $a \in \mathbb{N}$, oraz przesyła **B** wartość:
$$x \equiv g^a \pmod{p}.$$
3. **B** wybiera tajną liczbę $b \in \mathbb{N}$, oraz przesyła **A** wartość:
$$y \equiv g^b \pmod{p}.$$
4. Wspólnym tajnym kluczem jest teraz:
$$K \equiv g^{ab} \pmod{p}.$$

Jeżeli Cezary podsłuchiwał konwersację i zna p, g, x, y, będzie chciał obliczyć K. Aby to zrobić, musi obliczyć jeden z wykładników a lub b. Jest to równoważne z obliczeniem logarytmów dyskretnych:

$$a = \log_g(x) \pmod{p} \quad , \quad b = \log_g(y) \pmod{p}.$$

Kryptosystem opiera się zatem na własności, iż w grupie $\mathbb{Z}_p^*$:

$$K \equiv g^{ab} \equiv (g^a)^b \equiv (g^b)^a \equiv x^b \equiv y^a \pmod{p},$$

oraz, że jeżeli Cezary słucha, zna g^a, g^b, g^{a+b}, ale nie zna g^{ab}.

1.4 System ElGamala

System ElGamala jest kolejnym kryptosystemem opartym na trudności problemu logarytmu dyskretnego w skończonych grupach multiplikatywnych. Algorytm w połowie lat 80 XX wieku zaproponował Egipcjanin Taher ElGamal. Podobnie jak metoda Diffiego-Hellmana korzysta on z bezpiecznego przesyłania wiadomośći niezabezpieczonym kanałem.

System wymaga aby użytkownik posiadał zarówno klucz prywatny jak i klucz publiczny. Zaszyfrowane wiadomośći są ogólnie dostępne, natomiast deszyfrowanie jest możliwe tylko poprzez powołane osoby, które posiadają klucz prywatny. Krótko mówiąc, System ElGamala jest kryptosystemem asymetrycznym. Posiadanie klucza publicznego danego użytkownika, oraz znajomość metody szyfrowania, nie są wystarczające do odkrycia jego klucza prywatnego.

Pokażemy teraz algorytm generowania klucza publicznego, algorytm szyfrowania, oraz deszyfrowania w systemie ElGamala.

1.4.1 Algorytm generowania klucza w systemie ElGamala

1. Użytkownik **A** wybiera liczbę pierwszą p, oraz liczbę g, która jest pierwiastkiem pierwotnym modulo p.

2. **A** wybiera również liczbę $k \in \mathbb{N} : 0 \leqslant k < p - 1$, służącą za klucz prywatny. Obliczana jest liczba:

$$x \equiv g^k \pmod{p}.$$

3. Trójka (p, g, x) jest kluczem publicznym użytkownika **A**. Jest on dostępny dla wszystkich innych użytkowników.

Odkrycie liczby k na podstawie znajomości jawnych liczb $(p, g, x,)$ wymaga rozwiązania zagadnienia logarytmu dyskretnego. Analogicznie jak w przypadku metody Diffiego-Hellmana:

$$k = \log_g(x) \pmod{p}.$$

Zobaczyliśmy jak Alicja genereuje klucz publiczny. Teraz po jego publikacji Bob chce przesłać jej wiadomość. Pierwsze co Bob musi zrobić, to przekształcić ją na odpowiednik liczbowy, oraz ewentualnie może podzielić wiadomość na bloki. Spójrzmy zatem co z tak przygotowaną wiadomością należy dalej zrobić.

1.4.2 Algorytm szyfrowania w systemie ElGamala

Użytkownik **B** chce przesłać użytkownikowi **A** wiadmość którą przekształcił na odpowiednik liczbowy M.

1. Użytkownik **B** wybiera liczbę $j \in \mathbb{N} : 0 \leqslant j < p-1$, oraz oblicza:
$$C_1 \equiv g^j \pmod{p} \quad , \quad C_2 \equiv Mx^j \pmod{p}.$$
2. Szyfrogramem wiadomości M jest: $C = (C_1, C_2)$.

Bob zaszyfrował wiadomość M, oraz przesłał Alicji szyfrogram C. Zobaczmy teraz co należy zrobić aby odzyskać oryginalną wiadomość.

1.4.3 Algorytm deszyfrowania w systemie ElGamala

Użytkownik **A** może odszyfrować otrzymaną wiadomość używając swojego klucza prywatnego k.

$$C_2 C_1^{-k} \equiv (Mx^j)(g^{-jk}) \equiv (Mg^{jk})(g^{-jk}) \equiv M \pmod{p}.$$

Przykład 1.12. *Użytkownik **B** chce przesłać wiadomość SZYFR, osobie z kluczem publicznym $(p, g, x) = (43, 3, 22)$, oraz kluczem prywatnym $k = 15$.*

1. *Odpowiednikiem SZYFR jest $M = [18, 25, 24, 5, 17]$.*
2. ***B** wybiera $j = 23$ i oblicza:*
$$C_1 = g^j = 3^{23} \equiv 34 \pmod{43},$$
$$C_2 = Mx^j = M \cdot 22^{23} \equiv [17, 26, 37, 31, 28] \pmod{43}.$$

3. Zaszyfrowana wiadomość ma postać:

$$(C_1, C_2) = (34, [17, 26, 37, 31, 28]).$$

Teraz popatrzymy jak użytkownik ***A*** *poradzi sobie z otrzymanym szyfrogramem* (C_1, C_2).

4. ***A*** *używa klucza prywatnego* $k = 15$ *otrzymując:*

$$C_2 C_1^{-k} = C_2 \cdot 34^{-15} \equiv C_2 \cdot 39 \equiv [18, 25, 24, 5, 17] \pmod{43}.$$

Użytkownik ***A*** *otrzymał oryginalną wiadomość* M*, której odpowiednikiem jest wiadomość SZYFR.*

We wcześniejszej części artykułu rozważaliśmy problem 1.11, z którego wnioskowaliśmy iż szukanie logarytmów dyskretnych metodą przeliczania, jest nieefektywne dla dużych liczb pierwszych. Konsekwencją są protokoły kryptograficzne bazujące na tej własności. Istnieją jednak inne metody, które pozwalają przyśpieszyć proces obliczania logarytmów dyskretnych.

Przedstawimy teraz Algorytm Pohliga-Hellmana, oraz pokażemy jak korzystając z niego możemy obliczyć przykładowy logarytm.

1.5 Algorytm Pohliga-Hellmana

Algorytm Pohliga-Hellmana polega na zredukowaniu zagadnienia logarytmu dyskretnego do analogicznego problemu w mniejszych grupach cyklicznych.

1. Zakładamy że szukamy $x : b^x \equiv a \pmod{p}$, w grupie $\mathbb{Z}_p^*$ rzędu n, oraz że liczba n ma rozkład postaci:

$$n = p_1^{\alpha_1} \dots p_k^{\alpha_k}.$$

2. Dla każdej liczby p_i występującej w rozkładzie n obliczamy trzy liczby:

$$n_{p_i} = \frac{n}{p_i^{\alpha_i}}, \quad b_{p_i} \equiv b^{n_{p_i}} \pmod{p}, \quad a_{p_i} \equiv a^{n_{p_i}} \pmod{p}.$$

Twierdzenie 1.13. *Załóżmy że dla każdego czynnika pierwszego p_i rozkładu liczby n, liczba $x(p_i)$ jest rozwiązaniem logarytmu dyskretnego:*

$$b_{p_i}^{x(p_i)} \equiv a_{p_i} \pmod{p}.$$

Niech x będzie rozwiązaniem układu kongruencji:

$$x \equiv x(p_i) \pmod{p_i^{\alpha_i}},$$

dla wszystkich liczb pierwszych p_i w rozkładzie n. Wówczas x jest również rozwiązaniem zagadnienia logarytmu dyskretnego:

$$b^x \equiv a \pmod{p}.$$

Przykład 1.14. *Spróbujmy znaleźć $x : 2^x \equiv 7 \pmod{181}$.*

1. *Dla grupy $\mathbb{Z}_{181}^*$, której rząd wynosi 180, mamy rozkład:*

$$180 = 2^2 \cdot 3^2 \cdot 5^1.$$

2. *Obliczamy kolejno:*

$$\begin{aligned} p_1 &= 2^2, \quad n_{p_1} = 45, \quad b_{p_1} = 162, \quad a_{p_1} = 19, \\ p_2 &= 3^2, \quad n_{p_2} = 20, \quad b_{p_2} = 43, \quad a_{p_2} = 132, \\ p_3 &= 5^1, \quad n_{p_3} = 36, \quad b_{p_3} = 59, \quad a_{p_3} = 1. \end{aligned}$$

3. *Wyznaczenie logarytmu dyskretnego $x : 2^x \equiv 7 \pmod{181}$, możemy sprowadzić do rozwiązania trzech zagadnień:*

$$\begin{aligned} 162^{x(2)} &\equiv 19 \pmod{181}, \\ 43^{x(3)} &\equiv 132 \pmod{181}, \\ 59^{x(5)} &\equiv 1 \pmod{181}. \end{aligned}$$

4. *Obliczając logarytmy związane z kolejnymi p_i otrzymujemy:*

$$x(2) = 3, \quad x(3) = 6, \quad x(5) = 0.$$

5. *Zgodnie z twierdzeniem 1.13, szukany logarytm dyskretny x jest rozwiązaniem układu kongruencji:*

$$x \equiv 3 \pmod{2^2},$$
$$x \equiv 6 \pmod{3^2},$$
$$x \equiv 0 \pmod{5^1}.$$

6. *Rozwiązując układ otrzymamy:* $x = 15$.

Algorytm Pohliga-Hellmana nie jest oczywiście jedyną metodą przyśpieszającą obliczanie logarytmów dyskretnych. Innym algorytmem, który warto wymienić jest Algorytm Małych i Wielkich Kroków. Metoda ta jest zdecydowanie szybsza niż przeliczanie, jednak mniej wydajna niż Algorytm Pohliga-Hellmana. Oprócz kwestii czasu, wymaga ona dużo więcej pamięci niż pozostałe dwie metody. Dla dużych liczb pierwszych obliczenie logarytmu będzie wymagało przechowywania ogromnych tablic.

Kolejną interesującą metodą jest Algorytm ρ Pollarda. Mimo że jest on bardziej efektywny pod względem czasu obliczeń niż Algorytm Małych i Wielkich Kroków, także wymaga przechowywania dużych ilości danych. Warto również wspomnieć iż istnieją metody obliczania logarytmów dyskretnych na krzywych eliptycznych. Krzywe eliptyczne stanowią zaawansowaną dziedzinę kryptografii, oferującą zwiększoną bezpieczeństwo i wydajność w porównaniu do tradycyjnych metod.

Bibliografia

[1] Stinson D.R., Paterson M.B., *Kryptografia. W teroii i praktyce*, PWN, Wydanie IV, 2021.

[2] Dummit E., *Discrete Logarithms in Cryptography*, 2016.

[3] Chrzęszczyk A., *Algorytmy teorii liczb i kryptografii w przykładach*, BTC, 2010.

[4] Aumasson J.P., *Nowoczesna Kryptografia. Praktyczne wprowadzenie do szyfrowania*, PWN, 2018.