

## Zajęcia 11 Łamanie Diffiego-Hellmana, czyli odwracanie potęgowania modulo

Bezpieczeństwo protokołu D-H opiera się na tym że nie jest łatwo odwrócić potęgowanie modulo, tj. wyznaczyć  $x$ , taki że  $a^x \bmod n = d$ .

### Pierwsza metoda: próbne mnożenie

Najprostszą metodą jest próbowanie jednej po drugiej wykładników  $x$  i sprawdzanie kiedy powyższe równanie będzie spełnione.

Przykład  $3^x \bmod 17 = 2$

zaczynamy od  $3^1 \bmod 17 = 3$  (nie spełnione)

w kolejnych krokach wykorzystujemy poprzednie:

$$3^2 \bmod 17 = (3^1 * 3) \bmod 17 = (3 * 3) \bmod 17 = 9$$

$$3^3 \bmod 17 = (3^2 * 3) \bmod 17 = (9 * 3) \bmod 17 = 27 \bmod 17 = 10$$

$$3^4 \bmod 17 = (3^3 * 3) \bmod 17 = (10 * 3) \bmod 17 = 30 \bmod 17 = 13$$

$$3^5 \bmod 17 = 5$$

...

$$3^{14} \bmod 17 = 2, \text{ czyli } x = 14$$

Metoda ta jest bardzo wolna, bo musimy sprawdzić wszystkie liczby aż do  $x$  (które jest rzędu  $n$ ) – złożoność liniowa

### Metoda baby step – giant step

W tej metodzie mamy mniejszą złożoność obliczeniową kosztem zapotrzebowania na pamięć (obie są teoretycznie  $\sqrt{x}$ ).

równanie  $a^x \bmod n = d$  możemy zapisać jako  $a^{i \cdot m + j} \bmod n = d$  (czyli  $x = i \cdot m + j$ ), gdzie  $m$  jest  $\sqrt{n}$ , zaokrąglony w górę, a 'i' i 'j' są z zakresu 0 do  $m-1$ .

Równanie możemy zapisać w postaci

$$a^j \bmod n = (d * (a^{-m})^i) \bmod n$$

Ujemnej potęgi możemy się pozbyć zastępując  $a^{-m}$  przez  $v^m$ , gdzie  $(a * v) \bmod n = 1$ . Ostatni warunek to odwracanie modulo które już przerabialiśmy.

Istotą algorytmu jest, że liczymy osobno lewą stronę, zapisujemy w tablicy dla  $i$  od 0 do  $m-1$  i potem liczymy prawą stronę za każdym razem sprawdzając czy zgodzi nam się z którymś z elementów tablicy.

Przykład  $3^x \bmod 17 = 2$ ,  $m = \sqrt{17}$  czyli zaokrąglone w górę  $m=5$

liczymy  $3^j \bmod 17$  dla  $j=0$  do 4 (najlepiej tak jak w metodzie próbnego mnożenia)

$$j=0: 3^0 \bmod 17 = 1$$

$$j=1: 3^1 \bmod 17 = 3$$

$$j=2: 3^2 \bmod 17 = 9$$

$$j=3: 3^3 \bmod 17 = 10$$

$$j=4: 3^4 \bmod 17 = 13$$

teraz liczymy prawą stronę, ale zanim to zrobimy wyznaczmy  $v$  z równania  $(3*v) \bmod 17 = 1$  (w ten sam sposób jak na poprzednich zajęciach), no i dostajemy  $v = 6$ , czyli  $w = v^m \bmod n = 6^5 \bmod 17 = 7$ .

$i=0$ :  $d \bmod n = 2 \bmod 17 = 2$ : nie zgadza się z żadną wartością  $a^j \bmod n$

$i=1$ :  $d * w \bmod n = (2*7) \bmod 17 = 14$ : nie zgadza się

$i=2$ :  $(d * w^2) \bmod n = ((d * w \bmod n) * w) \bmod n = (14 * 7) \bmod 17 = 13$  – zgadza się z  $j=4$

Ponieważ zgodziły nam się wartości dla  $j=4$  z  $i=2$  to nasze  $x = 2 * 5 + 4 = 14$ .

No i faktycznie po sprawdzeniu  $3^{14} \bmod 17 = 2$

Uwaga – złożoność obliczeniowa jest rzędu pierwiastka z  $n$ , bo i po lewej i po prawej stronie liczymy do  $m \sim \sqrt{n}$ . Ale trzeba zauważyć, że mamy też krok sprawdzania czy lewa strona równa się prawej. W najprostszym przypadku jeżeli  $a^j \bmod n$  są zapisane w tablicy (albo słowniku!) i mamy przeszukiwanie w tablicy to jego złożoność jest znowu  $m \sim \sqrt{n}$ , czyli łączna złożoność wynosi ponownie  $n$  i jest niewiele lepsza od pierwszej metody.

Żeby wykorzystać pełną moc tej metody musimy mieć sposób szybkiego przeszukiwania  $a_j$ . Najprościej to zrobić wykorzystując prostą funkcję mieszającą.

Oznaczmy  $a_j = a^j \bmod n$  i wybierzmy liczbę pierwszą  $p$  która jest bliska  $m$ .

Funkcję mieszającą będziemy liczyć przez:

$\text{hash} = a_j \bmod p$ .

Tworzymy wtedy tablicę rozmiaru  $p$  w której każdy element będzie dynamiczną tablicą wszystkich  $a_j$  które mają taką samą funkcję mieszającą. Wtedy zamiast szukać w pełnej tablicy  $a_j$  wystarczy policzyć funkcję mieszającą tego co chcemy znaleźć i przeszukać tylko małą dynamiczną tablicę znajdującą się na pozycji tej funkcji mieszającej. Reszta algorytmu wygląda tak samo

Dla przykładu powyżej:

$m=5$  więc możemy wziąć również  $p=5$ ,

$a_0=1$ ;  $a_0 \bmod 5=1$

$a_1=3$ ;  $a_1 \bmod 5=3$

$a_2=9$ ;  $a_2 \bmod 5=4$

$a_3=10$ ;  $a_3 \bmod 5=0$

$a_4=13$ ;  $a_4 \bmod 5=3$

czyli nasza tablica funkcji mieszających wygląda tak:

hash = 0:  $a_3=10$

hash = 1:  $a_0=1$

hash = 2: <puste>

hash = 3:  $a_1=3$ ,  $a_4=13$

hash = 4:  $a_2=9$

liczymy teraz prawą stronę (tak jak poprzednio)

$i=0$ :  $d \bmod n = 2 \bmod 17 = 2$ ;  $2 \bmod 5 = 2$ ; tablica dla hash = 2 jest pusta – brak rozwiązań

$i=1$ :  $d * w \bmod n = (2*7) \bmod 17 = 14$ ;  $14 \bmod 5 = 4$ , dla hash =4 mamy jedynie  $a_2=9$ , ale nie ma 14, więc ciągle nie ma rozwiązań

$i=2$ :  $(d * w^2) \bmod n = (14 * 7) \bmod 17 = 13$ ;  $13 \bmod 5 = 3$ , dla hash = 3 mamy  $a_1=3$ ,  $a_4=13$  i druga pozycja z listy się zgadza, więc ponownie dostajemy  $j=4$

UWAGA: nie należy mylić odwracania potęgowania modulo (które jak wyjaśniam wyżej jest skomplikowane obliczeniowo), czyli  $a^x \bmod n = c$ ,  $x = ?$  z odwracaniem modulo (które wykorzystywaliśmy np. w RSA i jest szybkie obliczeniowo), czyli  $a * x \bmod n = 1$ ,  $x = ?$ .