1 Grupa mnożenia modulo - skrótowe przypomnienie

1.1 Definicja

Grupa mnożenia modulo jest strukurą algebraiczną $G_N = (A, \mod N)$, gdzie:

- A jest (skończonym) zbiórem wszystkich liczb naturalnych mniejszych od N, które nie mają wspólnych podzielników z N. Do zbioru tego włączamy też liczbę 1.
- działaniem wewnętrznym jest mnożenie modulo zdefiniowane dla $a, b \in A$ $(a \cdot b) \mod N$

1.2 Własności

Można pokazać, że G_N spełnia własności grupy. W szczególności:

- grupa jest zamknięta ze względu na działanie mnożenia modulo
- elementem neutralnym jest 1
- dla każdego $a \in G_N$ istnieje odwrotność b, taka że

$$(a \cdot b) \mod N = 1$$

1.3 Rząd grupy

Liczbę elementów grupy G_N nazywamy jej rzędem. W przypadku, kiedy $N = p \cdot q$, gdzie p, q są pierwsze, to liczba elementów w G_N (po wyrzuceniu wszystkich wielokrotności p lub q mniejszych od N) wynosi:

$$\underbrace{pq - 1}_{\substack{\text{wszystkie liczby} \\ \text{mniejsze od N}}} \quad - \quad \underbrace{(q-1)}_{\substack{\text{liczba wielokrotności p} \\ \text{mniejszych od N}}} \quad - \quad \underbrace{(p-1)}_{\substack{\text{liczba wielokrotności q} \\ \text{mniejszych od N}}}$$
$$= (p-1) \cdot (q-1)$$

1.4 Rząd elementu grupy

Każdy element $a \in G_N$ jest również charakteryzowany przez swój rząd – jest to najmniejsza liczba r taka, że

$$a^r \equiv 1(\mod N)$$

Można udowodnić (Mermin, rozdział 3, appendix A1), że **rząd każdego elementu grupy jest podzielnikiem rzędu całej grupy**

1.5 Rząd elementu grupy a okres funkcji

Można zauważyć, że rząd dla liczby a w G_N jest jednocześnie okresem funkcji

$$f(x) = a^x \mod N$$

zdefiniowanej dla $x \in N$.
Uzasadnienie:
Jeśli r jest okresem to dla każdego x

$$f(x+r) = f(x)$$

Czyli:

$$a^{(x+r)} \mod N = a^x \mod N$$

Stosując wzór na iloczyn potęg o tych samych podstawach mamy:

$$(a^x \mod N) \cdot (a^r \mod N) = a^x \mod N$$

Dzieląc przez $a^x \mod N$ otrzymamy:

$$a^r \mod N = 1$$

Czyli:

$$a^r \equiv 1 (\mod N)$$

2 Zasada działania RSA

2.1 Klucze

Mamy dane

- Klucz publiczny - para liczb N oraz c, gdzie N jest iloczynem dwóch liczb pierwszych $N = p \cdot q$, a $c \in G_{(p-1)(q-1)}$ czyli nie ma wspólnych podzielników z $(p-1)(q-1)$
- Klucz prywatny - liczba d

Klucz publiczny oraz klucz prywatny łączy zależność:

$$c \cdot d \equiv 1 \mod (p-1)(q-1)$$

Czyli są one odwrotnościami w $G_{(p-1)(q-1)}$

2.2 Kodowanie liczb

Każdą wiadomość a kodujemy do b

$$b = a^c \mod N$$

Odkodowywanie:

$$a = b^d \mod N$$

2.3 Uzasadnienie kodowania

Ponieważ
$$b^d = (a^c)^d = a^{(c \cdot d)}$$

Oraz
$$c \cdot d \equiv 1 \mod (p-1)(q-1)$$

to istnieje takie w, że
$$c \cdot d = (p-1)(q-1) \cdot w + 1$$

i mamy
$$(a^c)^d = a^{c \cdot d} = a^{(p-1)(q-1)w+1} = a^{(p-1)(q-1)w} \cdot a$$

Ponieważ rząd r ($a^r \equiv 1 \mod N$) dla a jest podzielnikiem $(p-1)(q-1)$ czyli istnieje takie u, że $(p-1)(q-1) = u \cdot r$. Wtedy:
$$a^{(p-1)(q-1)w} \cdot a \equiv a^{r \cdot u \cdot w} \cdot a \equiv (a^r)^{u \cdot w} \cdot a \equiv a \mod N$$

Istotny jest fakt, że $a^r \equiv 1 \mod N$, kolejne potęgowania przez u czy w już nic nie zmienią!

Wniosek: **aby odkodować wiadomość zamiast** $(p-1)(q-1)$ **można użyć** r **!**

2.4 Dwa sposoby odkodowywania

Mamy dane: zakodowaną wiadomość b oraz klucz publiczny czyli liczby N, c.

Najpierw należy sprawdzić (algorytm Euklidesa), czy najwiekszy wspólny dzielnik (NWD, ang. GCD) dla N i b nie jest czasem większy od 1 - jesli tak jest - jest on równy p, automatycznie mamy też q. Dla dużych p i q szansa trafienia na takie a jest bardzo mała.

2.4.1 Algorytm "Boba"

1. Rozkładamy N na czynniki pierwsze p i q (trudne)
2. Znajdujemy rząd G_N czyli $(p-1)(q-1)$ (łatwe)
3. Znajdujemy klucz prywatny d, taki, że $c \cdot d \equiv 1 \mod (p-1)(q-1)$. Czyli c oraz d to odwrotności w $G_{(p-1)(q-1)}$ (łatwe - np. rozszerzony algorytm Euklidesa)
4. Odkodowywujemy $a = b^d \mod N$ (łatwe - np. algorytmem szybkiego potęgowania)

2.4.2 Algorytm "Ewy"

1. Znajdujemy rząd zakodowanej wiadomości b w G_N czyli taką liczbę r, ze $b^r \equiv 1 \mod N$. Czyli inaczej znajdujemy okres funkcji $f(x) = b^x \mod N$ (trudne)

2. Znajdujemy klucz prywatny d', taki, że $c \cdot d' \equiv 1 \mod r$. Czyli c oraz d' to odwrotności w G_r (łatwe - np. rozszerzony algorytm Euklidesa)

3. Odkodowywujemy $a = b^{d'} \mod N$ (łatwe - np. algorytmem szybkiego potęgowania)

3 Co tak naprawdę robi algorytm Shora?

Algorytm Shora używa komputera kwantowego do znalezienia okresu funkcji $f(x) = b^x \mod N$. Do tego celu oblicza funkcję $f(x) = b^x \mod N$ dla superpozycji wielu zmiennych x, a następnie dokonuje kwantowej transformaty Fouriera do wyciągnięcia okresu. [tutaj demo funkcji FindPeriod].

3.1 Implementacje

- W przypadku QUIDE mamy w pełni zaimplementowaną funkcję *FindPeriod(int N, int a)*

- w przypadku Qiskita mamy przykład implementacji algorytmu Shora dla a=7 N=15 [https://qiskit.org/textbook/ch-algorithms/shor.html]

3.2 Jak to się ma do algorytmu Boba i Ewy ?

- W przypadku algorytmu Ewy wystarczy w punkcie (1) użyć komputera kwantowego do znalezienia okresu wiadomości *(FindPeriod)*.

- W przypadku algorytmu Boba w punkcie (1) należy użyć komputera kwantowego oraz klasycznego - za pomocą kwantowej funkcji znajdującej okres dodając pewne klasyczne obliczenia można rozłożyć N na p i q (patrz sekcja 4)

4 Jak za pomocą znajdowania okresu funkcji $f(x) = b^x \mod N$ rozłożyć N na czynniki pierwsze?

Poniżej podeję sposób - jest on również opisany w [Mermin, rozdział 3, sekcja H]

1. Wybieramy losową liczbę a

2. Sprawdzamy (np. algorymem Euklidesa) czy $1 < NWD(a, N) < N$. Jeśli tak - mamy super szczęście

$$p = NWD(a, N)$$

$$q = N/p$$

STOP

3. używamy komputera kwantowego do znalezienia okresu funkcji $f(x) = a^x \mod N$ czyli takiej najmniejszej liczby r, że $a^r \equiv 1 \mod N$

4. jeśli r jest nieparzyste wróć do p.1

5. jeśli r jest parzyste możemy zapisać

$$a^r \equiv 1 \mod N$$

$$(a^{\frac{r}{2}})^2 \equiv 1 \mod N$$

$$(a^{\frac{r}{2}})^2 - 1 \equiv 0 \mod N$$

Ze wzoru skróconego mnożenia $m^2 - n^2 = (m - n)(m + n)$ mamy:

$$(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) \equiv 0 \mod N$$

$$(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) \equiv 0 \mod (p \cdot q)$$

czyli istnieje pewne s

$$(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) = p \cdot q \cdot s$$

Istnieje duża szansa, że p jest podzielnikiem $(a^{\frac{r}{2}} - 1)$, natomiast q jest podzielnikiem $(a^{\frac{r}{2}} + 1)$

- Pech (*): nie zawsze ! - może się zdarzyć, że zarówno p, jak i q będą podzielnikami $(a^{\frac{r}{2}} + 1)$
- Pech (**) teoretycznie nigdy nie będzie tak, że zarówno p, jak i q będą podzielnikami $(a^{\frac{r}{2}} - 1)$, bo wtedy byłoby:

$$(a^{\frac{r}{2}} - 1) \equiv 0 \mod (p \cdot q)$$

Wtedy $\frac{r}{2}$ byłoby okresem, tymczasem okresem (najmniejszą liczbą spełniającą ten warunek) jest r. W praktyce jednak implementacja algorytmu Shora znajduje czasem wielokrotność okresu, więc czasem może się to zdarzyć.

6. Policz NWD dla $(a^{\frac{r}{2}} + 1)$ oraz N (np. algorytmem Euklidesa). Jeśli jest on równy N (mamy pecha (*)) lub 1 (mamy pecha (**)) wróć do p.1

7. Jeśli nie mamy żadnego pecha (jest na to $\approx 50\%$ szans) to:
p=NWD($(a^{\frac{r}{2}} + 1)$, N)
q=NWD($(a^{\frac{r}{2}} - 1)$, N) albo q=N/p
STOP