Kryptografia - lista 3

Oskar Makowski 236554

24 Maja 2020

1 Merkle-Hellman cryptosystem

System kryptograficzny Merkle-Hellman'a opiera się na kryptografii asymetrycznej, używany jest klucz prywatny i publiczny. Klucz publiczny służy tylko do szyfrowania, a klucz prywatny tylko do deszyfrowania. System bazuje na problemie sumy podzbioru, będącej szczególnym przypadkiem problemu plecakowego, gdzie mając dany zbiór A i liczbę b, należy znaleźć taki podzbiór $B \subseteq A$, że zawarte w nim liczby sumują się do b: $\sum_{x \in B} x = b$. W systemie wykorzystywane jest też pojęcie ciąg superrosnącego, tzn. takiego, w którym każdy wyraz jest większy od sumy wszystkich wcześniejszych wyrazów ciągu: $a_k > \sum_{i=0}^{k-1} a_i$

2 Generowanie klucza

Klucze są dwoma "plecakami" (ang. *knapsacks*). Klucz publiczny jest plecakiem "trudnym" A, a klucz prywatny jest "łatwym" plecakiem B, z dodatkowymi liczbami: mnożnikiem i modułem. Te liczby pozwalają w wielomianowym czasie wykonać transformację z plecaka "trudnego" do "łatwego".
Ciąg superrosnący w może być prosto wygenerowany. Wystarczy zacząć od losowej liczby, a następnie do posiadanej już sumy dodawać następne losowe liczby większe od 0, a kolejno uzyskiwane sumy utworzą żądany ciąg. Mnożnik q musi być większy niż suma wszystkich wyrazów ciągu w, niech więc $q = (\sum_{i=1}^{n} w_i)+1$. Moduł r musi być względnie pierwszy w stosunku do q, niech więc $r = q - 1$, bowiem dwie kolejne liczby naturalne $a, a+1$ są zawsze względnie pierwsze względem siebie. Klucz prywatny B tworzy trójka (w, q, r). Ciąg klucza publicznego A powstaje następująco: $b_i = w_i * r \mod q$.

3 Szyfrowanie

Żeby utworzyć szyfrogram n-bitowej wiadomości postaci $\alpha = (\alpha_1, \ldots, \alpha_n)$, gdzie α_i jest i-tym bitem wiadomości, należy obliczyć

$$c = \sum_{i=1}^{n} \alpha_i b_i$$

gdzie c jest zaszyfrowaną wiadomością.

4 Deszyfrowanie

Wartości b_i są wybrane w taki sposób, żeby deszyfracja była łatwa, dla osoby znającej klucz prywatny. Na początku należy wyznaczyć takie s, że $sr \mod q = 1$. r, q znane są z klucza prywatnego, toteż można użyć rozszerzonego algorytmu Euklidesa do wyznaczenia s.

$$c' = cs = \sum_{i=1}^{n} \alpha_i b_i s(\mod q)$$

ponieważ $rs \mod q = 1$ i $b_i = rw_i \mod q$, to

$$b_i s = w_i rs = w_i(\mod q)$$

więc

$$c' = \sum_{i=1}^{n} \alpha_i w_i(\mod q)$$

Ponieważ w jest superrosnącym ciągiem wystarczy wybrać największy element w, niech będzie on oznaczony przez w_k. Jeśli $w_k > c'$ wtedy $\alpha_k = 0$, w przeciwnym przypadku $\alpha_k = 1$. Pozostaje jeszcze wykonać $c' - w_k a_k$.

5 Wynik

```
Enter plaintext: ala ma kota
Number of plaintext bytes = 11
ala ma kota is encrypted as: 400962303361253691565636296179906586471
Result of decryption: ala ma kota

Process finished with exit code 0
```

Rysunek 1: Rezultat wykonania programu