Sprawozdanie

Szyfry asymetryczne

Mikołaj Pluta 151827

RSA

W algorytmie RSA wykorzystuje się dwie duże liczby pierwsze. Im większe te liczby, tym większe bezpieczeństwo systemu, ponieważ łamane są one metodami faktoryzacji liczb dużych. Jednak zwiększanie rozmiaru liczb wykorzystanych w algorytmie może skutkować wydłużeniem czasu potrzebnego do obliczenia kluczy oraz szyfrowania i deszyfrowania danych.

Opis metod użytych do wyznaczania e i d:

Na początku wyznaczamy dwie duże liczby pierwsze, p i q oraz obliczamy phi = (p-1)*(q-1).

Następnie wyznaczamy e które jest liczbą względnie pierwszą z phi. Na podstawie tych liczb możemy wygenerować d takie, że iloczyn e * d modulo phi = 1.

Opis realizacji zadań (programu i jego składowych) i wartości uzyskane podczas ich realizacji:

Na realizację zadania składa się zestaw funkcji pomocniczych, odpowiadających za generowanie konkretnych liczb oraz fragment który te liczby generuje i używa ich do zaszyfrowania i odszyfrowania przykładowej wiadomości. Samo szyfrowanie wykonywane jest osobno dla każdego znaku tekstu początkowego a wynikiem szyfrowania jest tablica liczb, które po odszyfrowaniu odtwarzają wiadomość początkową.

Odpowiedzi na pytania:

1. Jakie elementy algorytmu są trudne w realizacji?

   W realizacji algorytmu RSA trudnością jest znajdowanie dużych liczb pierwszych.

2. Co stanowi o bezpieczeństwie i jakości tego algorytmu szyfrowania?

   Stanowi o tym wielkość wybranych liczb pierwszych, ze względu na sposób łamania tego szyfru, czyli faktoryzacja dużych liczb.

Wnioski:

Algorytm RSA jest potężnym narzędziem do szyfrowania danych, ale wymaga starannego wyboru parametrów, takich jak długość kluczy, aby zapewnić odpowiedni poziom bezpieczeństwa w zmieniającym się środowisku kryptograficznym.

Algorytm Diffiego-Helmana

Algorytm Diffiego-Hellmana to protokół wymiany klucza, który umożliwia dwóm stronom komunikującym się zaufane i bezpieczne ustalenie wspólnego tajnego klucza. W celu zapewnienia bezpieczeństwa, liczba pierwsza n powinna być odpowiedni duża, aby uniemożliwić jej faktoryzację.

Niestety, algorytm ten podatny jest na ataki ‘’man in the middle”, czyli sytuacje w których osoby niepożądane mogą uzyskać publiczne wartości generowane przez użytkowników oraz ich klucze publiczne, które musza między sobą wymienić. Na podstawie tych wartości podsłuchujący może odtworzyć wspólny dla stron klucz tajny, i np. przekierowywać przez siebie wszystkie wiadomości podszywając się pod jedną ze stron.