

Національний технічний університет України

Київський політехнічний інститут

Фізико-технічний інститут

**Комп’ютерний практикум №**

**з дисципліни «»**

**Виконав**:

**Перевірив**:

Київ 2017

1. **Схема шифрування Ель-Гамаля**

Схема шифрування Ель-Гамаля була запропонована у 1985 році. Стійкість даної схеми грунтується на складності розв’язання задачі дискретного логарифма у групі . Головним недоліком даної схеми є те, що отриманий із її застосуванням шифртекст є вдвічі довшим за початкове повідомлення.

**Побудова схеми шифрування Ель-Гамаля абонентом .**

Для цього абонент виконує наступні кроки.

1. Обирає велике просте число та – примітивний елемент поля ( і є відкритими параметрами).
2. Обирає випадкове число – секретний ключ , який використовується для розшифрування.
3. Обчислює . Параметри складають відкритий ключ, що застосовується для шифрування повідомлень, призначених для абонента .
4. Забезпечує доступність відкритого ключа для всіх абонентів, які бажають відіслати зашифроване повідомлення.

Умовно кажучи, відкритий ключ може бути опублікований.

**Шифрування повідомлення , абонентом для абонента**

Абонент , який знає алгоритм шифрування і відкритий ключ хоче переслати абонентові зашифроване повідомлення, що кодується і є відповідним цілим числом ,. Для цього він виконує наступні дії.

1. Вибирає випадкове число , .
2. Обчислює та .
3. Формує шифрований текст, що є впорядкованою парою чисел (складається з двох частин): .

**Розшифрування зашифрованого повідомлення абонентом**

Абонент обчислює .

1. **Генерація параметрів для схеми Ель-Гамаля**

Для роботи дана криптосистема потребує генерації двох основних параметрів великого простого числа та - генератора групи . Для унеможливлення деяких атак, а також для простоти пошуку генератора , рекомендовано обирати модуль вигляду , де – велике просте число. Найбільш зручним є випадок , тоді . Для генерації числа довжиною біт використовується наступний алгоритм:

1. Генеруємо просте число довжиною біт (ймовірнісними або детермінованими методами).
2. Обчислюємо і перевіряємо його на простоту. Якщо тест встановив, що просте, то завершуємо роботу алгоритму. Інакше повертаємось на крок 1.