Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій (ІСТ)

Звіт до комп’ютерного практикуму №4

З дисципліни

«Алгоритми та математичні методи захисту інформації»

Підготував:

Студент групи ІК-02

Талько Ярослав

Перевірив:

Нестерук А.

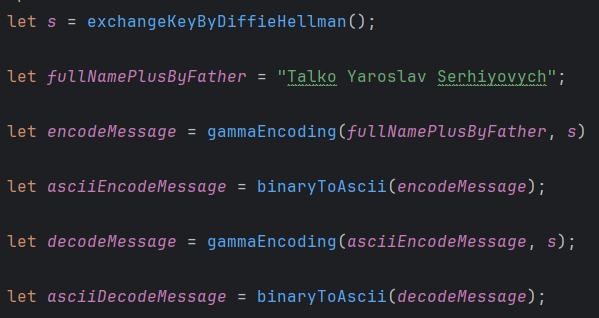
Київ 2024

**Тема:** Протокол узгодження ключів Діффі-Геллмена.

**Мета:** Навчитися застосовувати криптографічний протокол узгодження ключів Діффі-Геллмена.

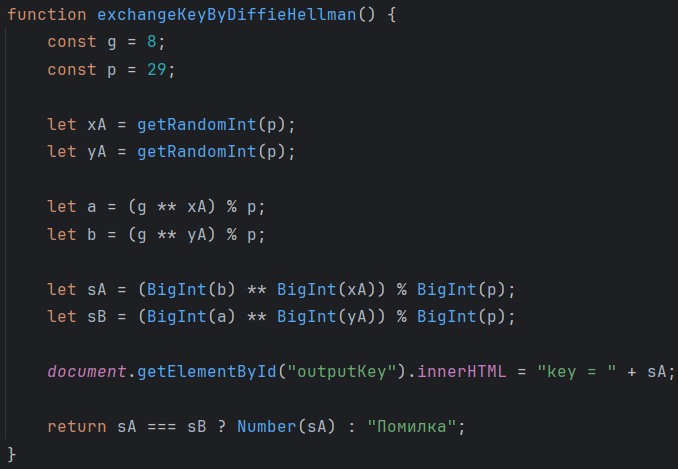
**Хід роботи**

Код програми написаний мовою JS. Спочатку розглянемо основну роботу програми:

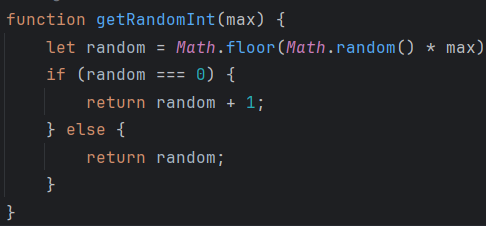


* s – приймає ключ, згенерований за протоколом Діффі-Геллмена.
* fullNamePlusByFather – відповідає за моє ПІБ, яке буде зашифроване.
* encodeMessage – приймає зашифроване повідомлення з допомогою ключа, який ми отримали з протоколу вище.
* asciiEncodeMessage – зберігає зашифроване повідомлення в ASCII коді.
* decodeMessage – приймає розшифроване повідомлення, яке було до цього зашифрованим
* asciiDecodeMessage – зберігає розшифроване повідомлення в ASCII коді.

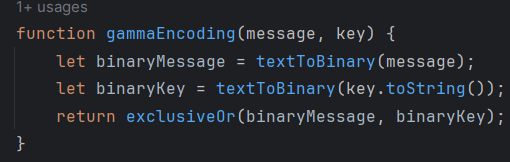
Розглянемо роботу протоколу Діффі-Геллмена:



* p та g – взяли з таблиці до завдання. Так, як у мене 41 номер, я взяв №11 з 30 доступних.
* xA та yA – числа, які рандомно генеруються від 1 до p не включно.
* a та b – g підноситься в степені до xA/yA і береться по модулю p;
* sA та sB – b/а підноситься в степені до xA/yA відповідно і береться по модулю p. sA та sB повинні вийти однакові
* Далі результат функції повертається

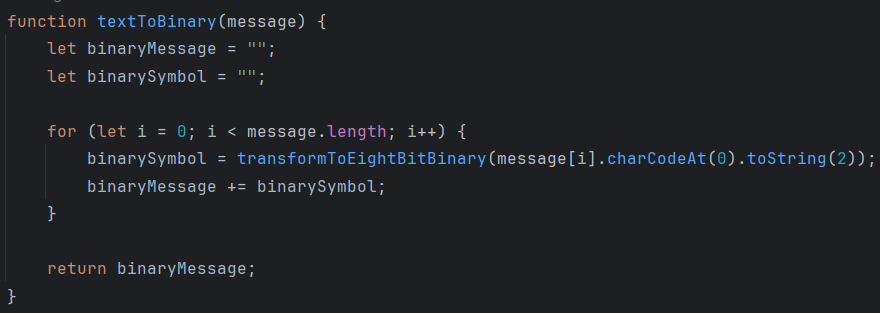


Дана функція забезпечує генерацію чисел від 1 до заданого максимального числа не включно.



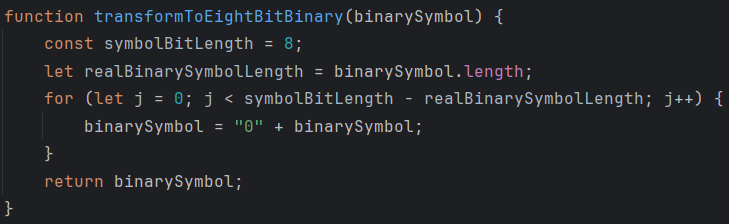
Саме ґамма щифрування розділено на декілька етапів:

* Перетворення повідомлення з ASCII в двійковий код
* Перетворення ключа з ASCII в двійковий код
* Повертання результату функції XOR між повідомленням та ключем.



Функція переведення у двійковий код:

* Кожен елемент повідомлення перетворюється у восьмибітний двійковий код.
* Після чого записується покроково в одну змінну, яка потім повертається.



Функція перетворення у восьмибітний двійковий код:

* Елемент повідомлення, яке перетворилося у двійковий код, заповнюється нулями спереду до тих пір, поки не матиме довжину у 8 бітів.



XOR операція:

* Повідомлення ділиться по 8 бітів. Кожна з цих частин проходить XOR операцію з ключем. Так, як ключ набагато менший за повідомлення, то після закінчення його елементів, він починає проходити ключ наново.
* Сама XOR операція: збіг – 0, не збіг – 1.
* Після чого результат повертається.

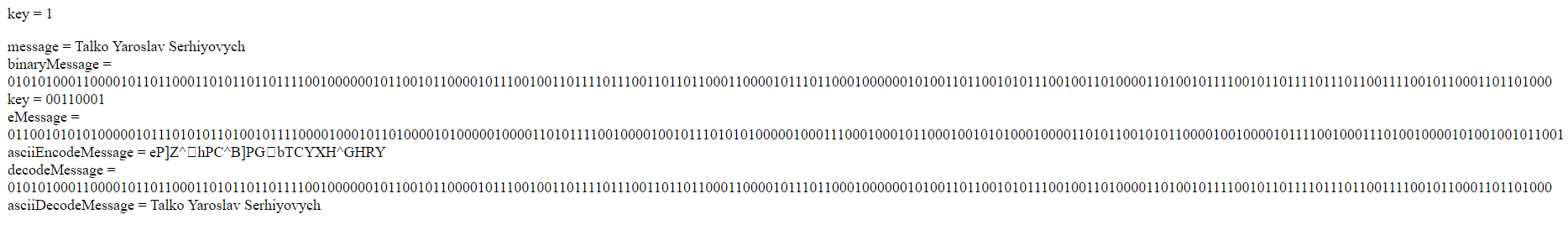


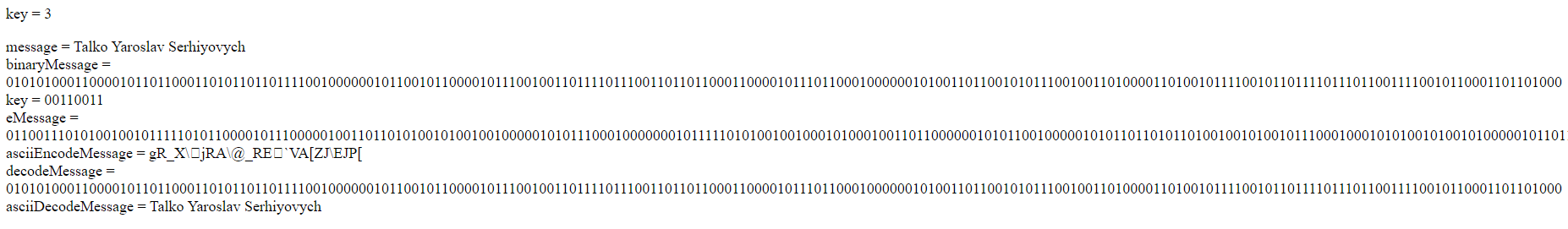
Переведення з двійкового в ASCII код:

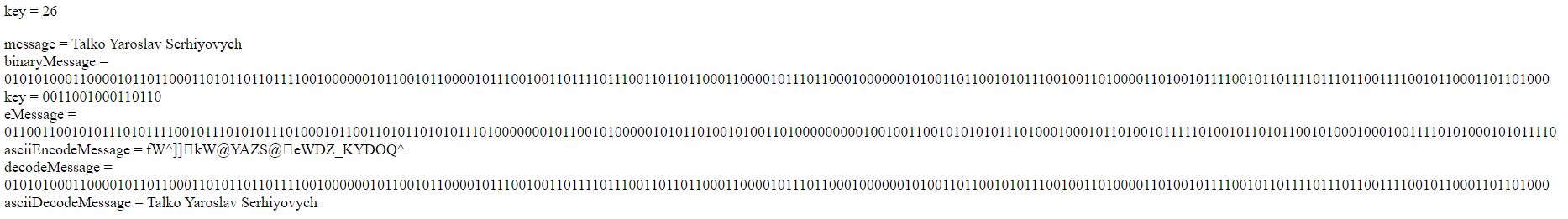
* 8 бітів зберігаються в одну тимчасову змінну і переводяться у ASCII код з допомогою маніпуляцій вбудованих функцій JS.
* Після чого результат повертається.

Результати програми:

Я зробив кілька скриншотів результатів, аби засвідчити, що при різних значеннях ключа, шифрування та розшифрування працює коректно:







**Висновок:** На цій лабораторій роботі я навчився застосовувати криптографічний протокол узгодження ключів Діффі-Геллмена.