Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

НН Фізико-технічний інститут

Дисципліна: “Методи реалізації криптографічних механізмів”

**ЛАБОРАТОРНА РОБОТА №2**

**Виконали:**

Студенти групи ФБ-21мн

ЩЕРБАКОВ О.К.

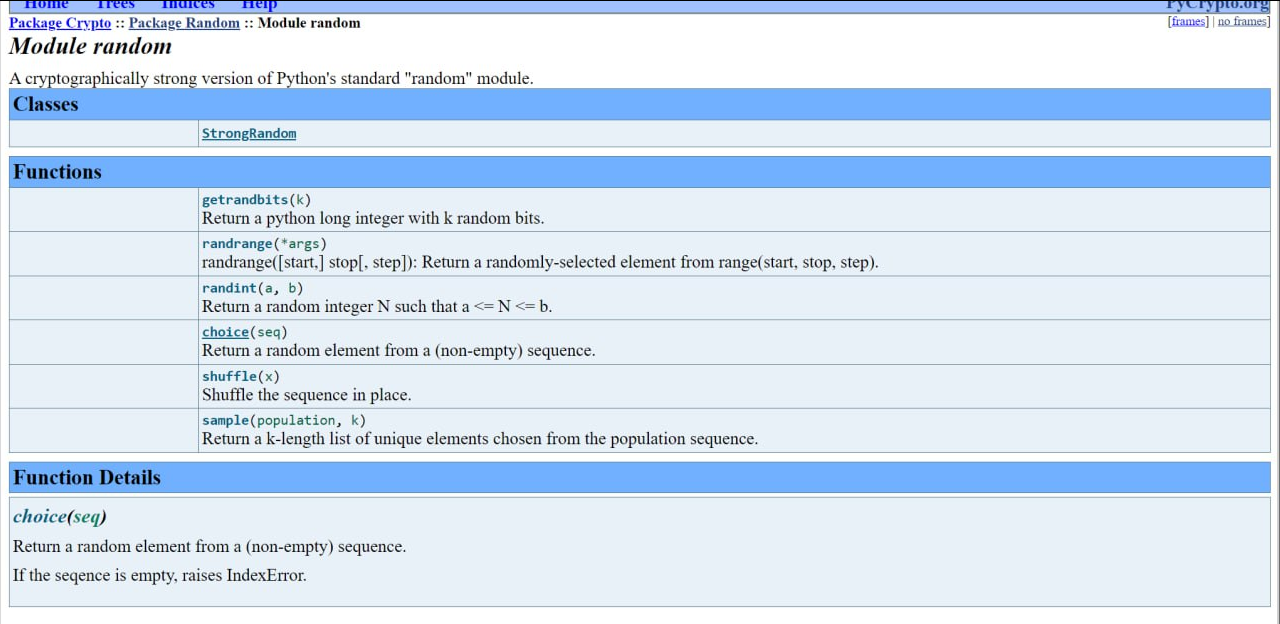
КАЗМІДІ І.Д.

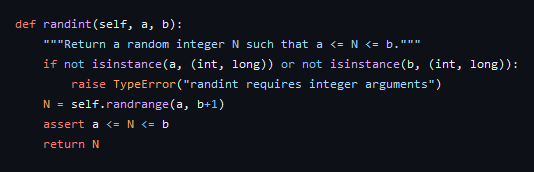
Київ – 2023

**Завдання**

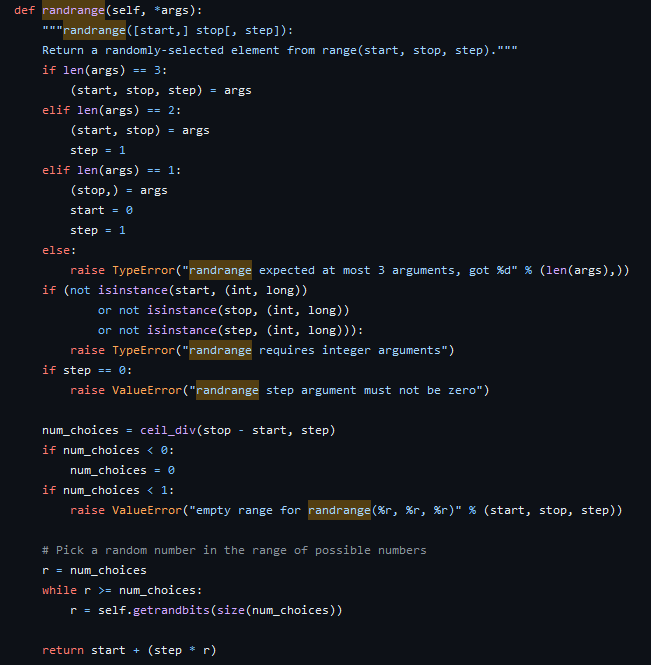
1. Виконати аналіз стійкості реалізацій ПВЧ та генераторів ключів для обраної бібліотеки. Бібліотека PyCrypto під Linux платформу

**Аналіз генератора ПВЧ**

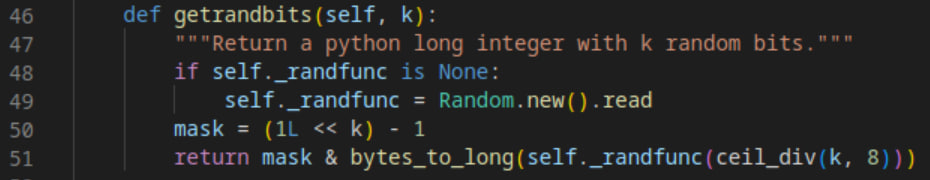
Проаналізуємо функції роботи з випадковими даними у PyCryptoDome:  




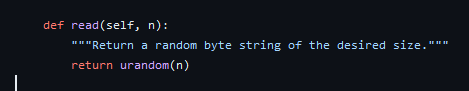
Функція randint призначена для отримання випадкового числа, що знаходиться між аргументами “a” та “b”. Під час роботи функція виконує стандартну перевірку на тип отриманих даних, після чого передає керування на функцію randrange, яка обирає випадкове число.  
Після відпрацювання randrange виконується перевірка, що отримане число знаходиться в потрібно проміжку. Якщо перевірка успішно - функція повертає обране число.



Randrange - функція, що повертає випадкове число з заданого проміжку, з можливістю вказати крок, через який враховуються числа у проміжку. Після перевірок аргументів, у змінну num\_choices присвоюється кількість можливих варіантів відповіді, яка обраховується через ceil\_div, також оброблюється виключення, коли можливих варіантів для вибору нема.  
Далі у while викликається getrandombits, яка повертає випадкові біти кількістю, яка вказана в аргументі. Оскільки розмір цілого числа в Python залежить від величини числа, в якості бажаної кількості бітів передається розмір числа кількості варіантів. Цикл повторюється до тих пір, поки не отримаємо число, яке менше ніж кількість можливих варіантів num\_choices. Після отримання відповідного числа, випадкове число “збирається” назад за формулою start+(step\*r).   
Фактично вибір випадкового числа із заданого проміжку зводиться до вибору кількості кроків, які потім додаються до початку проміжку.



Getrandbits - повертає K-випадкових бітів .Після підрахунку mask відбувається бінарна операція “AND” над бітами отриманого числа.  
Під час виклику \_randfunc (Random.new().read) викликається функція urandom() з модуля “os”.



Таким чином, генератор ПВЧ в PyCryptoDome з невеликими змінами, спирається на стандартну функцію Python.

Urandom - функція з модуля “OS”, після виклику визначає тип ОС хоста і у випадку Unix звертається до /dev/urandom, а для Windows - функція WinAPI CryptGenRandom().   
У випадку Unix /dev/urandom визначається як достатньо надійний генератор ПВЧ. Після використання всіх наявних випадкових бітів urandom почне генерувати нові біти, які теоретично вразливі до де-яких криптографічних атак. Розробники рекомендують користуватися /dev/random, який у випадку закінчення бітів очікує створення нових і використовує апаратні джерела ентропії, вбудовані в ЦП.  
CryptoGenRandom() - бере ентропію з даних програми на хост (наприклад ID процеса, що викликає цю функцію). Даний генератор має необхідну сертифікацію для використання у криптографії і вважається стійким, хоча у ранніх версіях Windows спостерігалися значні проблеми з безпекою функції через використання алгоритму RC4, який дозволяє прорахувати усі минулі і майбутні стани бітів генератора за умови, що хоча б один стан вже відомий.

**Висновки**

Генератор ПВЧ модуля PyCryptoDome має достатній рівень захищеності для використання у криптографічних системах ОС Linux. У критичних криптографічних системах (наприклад у державних установах) рекомендується користуватися апаратними джерелами ентропії для генераторів ПВЧ.