Зміст

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 2](#_Toc358845423)

[ВСТУП 3](#_Toc358845424)

[1 ОЗНАЧЕННЯ НЕОБХІДНИХ В РОБОТІ ТЕРМІНІВ ТА ПОНЯТЬ 5](#_Toc358845425)

[2 ОГЛЯД ТА АНАЛІЗ ШИФРІВ 6](#_Toc358845426)

[2.1 Шифр Moustique 6](#_Toc358845427)

[ПЕРЕЛІК ПОСИЛАНЬ 11](#_Toc358845428)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

# ВСТУП

Потокові шифри на базі [зсувних регістрів](https://uk.wikipedia.org/w/index.php?title=%D0%97%D1%81%D1%83%D0%B2%D0%BD%D0%B8%D0%B9_%D1%80%D0%B5%D0%B3%D1%96%D1%81%D1%82%D1%80&action=edit&redlink=1) активно використовувалися в роки війни, ще задовго до появи електроніки. Вони були прості в проектуванні та реалізації.

1965 Ернст Селмер, головний криптограф норвезького уряду, розробив теорію послідовності зсувних регістрів. Пізніше [Соломон Голомб](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D0%BE%D0%BB%D0%BE%D0%BC%D0%BE%D0%BD_%D0%92%D0%BE%D0%BB%D1%8C%D1%84_%D0%93%D0%BE%D0%BB%D0%BE%D0%BC%D0%B1&action=edit&redlink=1), [математик](https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA) [Агентства Національної Безпеки США](https://uk.wikipedia.org/wiki/%D0%90%D0%B3%D0%B5%D0%BD%D1%82%D1%81%D1%82%D0%B2%D0%BE_%D0%BD%D0%B0%D1%86%D1%96%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%97_%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D0%BA%D0%B8), написав книгу під назвою «Shift Register Sequences» («Послідовності зсувних регістрів»), в якій виклав свої основні досягнення в цій галузі, а також досягнення Селмер.

Велику популярність потоковим шифрів принесла робота [Клода Шеннона](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D0%BE%D0%B4_%D0%A8%D0%B5%D0%BD%D0%BD%D0%BE%D0%BD), опублікована в 1949 році, в якій Шеннон довів абсолютну стійкість [шифру Вернама](https://uk.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%92%D0%B5%D1%80%D0%BD%D0%B0%D0%BC%D0%B0). У шифрі Вернама [ключ](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D1%96%D1%8F)) має довжину, рівну довжині самого переданого повідомлення. Якщо кожен [біт](https://uk.wikipedia.org/wiki/%D0%91%D1%96%D1%82) ключа вибирається випадково, то розкрити шифр неможливо (тому що всі можливі відкриті тексти будуть рівноймовірними). Шифри, в яких довжина ключа менша від довжини тексту, згідно з Шенноном, не можуть бути «ідеально безпечними» [1].

З лютого 2000р. по лютий 2003р. проходив європейський дослідницький проект NESSI (New European Schemes of Signatures, Integrity and Encryption) для визначення безпечних алгоритмів шифрування. На цьому проекті було представлено лише 6 потокових шифрів, причому наприкінці конкурсу жоден з них не був схвалений як такий, котрий міг би задовольнити всім вимогам. Саме цей проект став поштовхом до оголошення нового європейського конкурсу, присвяченого виключно потоковим шифрам. Цим конкурсом став проект eSTREAM. На конкурс eSTREAM було пред’явлено 34 шифри, що порівняно з конкурсом NESSI було великим кроком уперед.

Головною метою конкурсу було отримання шифру широкого використання, котрий працював би швидше за AES (у режимі лічильника), але був би вільним від його.

**Актуальність роботи**. В процесі створення систем шифрування гостро постає проблема знаходження компромісу між високою швидкістю роботи та низькими вимогами до використання обчислювальних ресурсів. Саме потокові шифри виступають таким компромісом – питання побудови стійких і водночас ефективних потокових систем шифрування знаходиться в центрі уваги сучасної криптографічної спільноти. Підтвердженням цього є проведення у 2005-2008р.р. у рамках проекту ECRYPT загальноєвропейського конкурсу eSTREAM, присвяченого виключно потоковим шифрам. Цей конкурс став найвизначнішою подією у розвитку теоретичної та прикладної криптографії останнього десятиліття. eSTREAM сприяв пожвавленню науково-практичної роботи в галузі розробки та криптоаналізу потокових шифрів. Вивчення, дослідження, осмислення результатів цього проекту – одна з актуальних задач сучасної криптографії.

**Мета і завдання дослідження**. Метою роботи є опис та аналіз деяких сучасних потокових шифрів, огляд та імплементація існуючих криптоатак на них.

*Об’єкт дослідження*: алгоритми потокового шифрування, що були представлені на проекті eSTREAM.

*Предмет дослідження*: опис та аналіз деяких шифрів-учасників проекту eSTREAM за їх характеристиками, особливостями будови, швидкодією, огляд та імплементація криптоатак на ці шифри; виявлення на цій основі сучасних тенденцій розвитку потокового шифрування.

**Практичне значення одержаних результатів.** Отримані результати можна використовувати, як аналіз та інструкцію по реалізації деяких потокових шифрів із самосинхронізацією та криптоатак для них.

# 1 ОЗНАЧЕННЯ НЕОБХІДНИХ В РОБОТІ ТЕРМІНІВ ТА ПОНЯТЬ

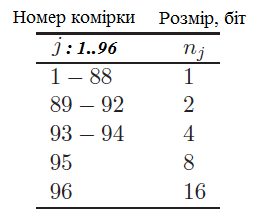
# 2 ОГЛЯД ТА АНАЛІЗ ШИФРІВ

## 2.1 Шифр Moustique



Рис. 1 Схема за шифрування-розшифрування шифру Moustique

Ключ K – 96 бітів подається на вхід спеціального регістру зсуву CCSR, що складається із 96 комірок, розмір кожної з комірок в бітах задано таблицею:



Пояснення позначень, що будуть використовуватись далі:



Розглянемо схему самого CCSR:



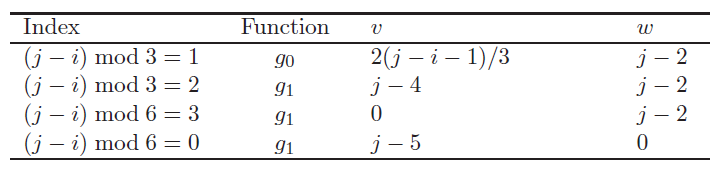
Розглянемо як заповнюється регістр на прикладі кожної комірки та кожного її біта.

Для :

Для є загальна формула:

*,*

де , а значення та всіх комбінацій визначені в таблиці:



Для прикладу розглянемо, як отримують значення деякі біти цих комірок:

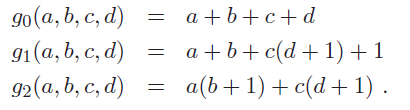
…

…

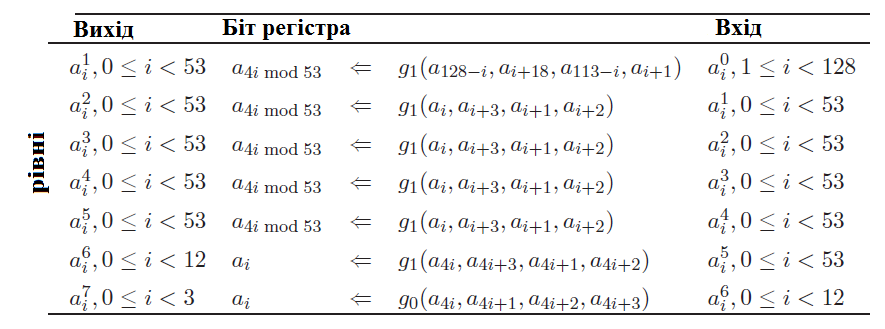
Для є окрема формула:

Для прикладу, обрахуємо випадок, коли :

Що ми забули визначити – це функції :



Далі йдуть інші рівні шифру – інші регістри. Їх заповнення теж визначено таблицею:



, де *i* – номер біту регістра *k-го –* рівня.

Приклад: 11-й біт, регістру 1-го рівня, знаходимо, коли

,

де – це біти CCSR

Аналогічно рахуємо за схемою, зображеній в таблиці вище, для всіх рівнів. Останній 7-й рівень має лише 3 біти.

Біт шифр тексту , відповідно

Коментар:

На даному етапі, цілком зрозуміло, як працює шифр і як зашифровує. Я не дуже розумію, як він розшифровує і як саме використовується вектор ініціалізації.

Я вивчаю код, представлений на конкурс і надіюсь, що коли зможу відтворити алгоритм на зручній мені мові програмування, то зрозумію. В мене є тестові приклади всіх вхідних та вихідних даних, щоб перевірити правильність власної імплементації.

# ПЕРЕЛІК ПОСИЛАНЬ

1. https://uk.wikipedia.org/wiki/Потоковий\_шифр