**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

**1. Общее описание алгоритма**

В рамках данной лабораторной работы реализован генератор псевдослучайных битовых последовательностей на базе HMAC (HMAC-based Deterministic Random Bit Generator), соответствующий спецификации NIST SP 800-90A Rev. 1.

**Ключевые свойства:**

* **Используемый механизм:** HMAC с SHA-256 обеспечивает высокую степень криптографической стойкости и устойчивость к коллизиям.
* **Источник начальной энтропии:** Получается через вызов RAND\_bytes из OpenSSL, который задействует внутренние аппаратные и системные источники.
* **Обновление состояния:** После каждого запроса на генерацию состояние DRBG модифицируется, обеспечивая безопасность и непредсказуемость следующего блока.

**Принцип работы:**

* **Начальная инициализация:** Значения ключа (Key) и вектора (V) задаются стандартными значениями (нулевые и единичные байты).
* **Процесс обновления:** Алгоритм вызывает HMAC дважды — сначала без дополнительной информации, затем с ней (если она задана), чтобы пересчитать Key и V.
* **Генерация данных:** Каждые 32 байта выходной последовательности получаются через последовательные применения HMAC к текущему V, количество которых определяется длиной требуемой выходной последовательности.

**Итерационность:**

* Для каждого вызова hmac\_drbg\_update выполняются **2 раунда HMAC**.
* При генерации данных производится ceil(output\_length / 32) итераций HMAC-SHA256.

**2. Вычислительная сложность**

**Временная сложность:**

* Алгоритм работает пропорционально объёму требуемых данных, то есть **O(n)**, где n — объём в байтах.
* Каждый блок из 32 байт генерируется за константное время.

**Память:**

* **Постоянная:** структура HMAC\_DRBG занимает порядка 64 байт.
* **Дополнительная:** основной буфер под выходной массив переменного размера (в зависимости от задания — от 1 МБ до 1 ГБ).

**3. Оценки качества выходных данных (по открытым источникам)**

* **Соответствие международным требованиям:** HMAC-DRBG полностью отвечает требованиям NIST SP 800-90A, включая устойчивость к внутренним и внешним атакам.
* **Верификация:** Алгоритм получил признание в профессиональном сообществе и используется в TLS, сертификатной инфраструктуре и других критически важных системах.
* **Доказанная стойкость:** Исследования показывают, что при правильной инициализации и регулярном обновлении состояния HMAC-DRBG не демонстрирует признаков статистической слабости.
* **Тестирование NIST STS:** Генератор прошёл проверку качества генерации через стандартную батарею тестов, продемонстрировав соответствие нормам случайности.

**4. Обоснование выбора алгоритма**

Выбор в пользу HMAC-DRBG был сделан по следующим причинам:

* **Соответствие требованиям стандартизации:** Реализация полностью базируется на официальных рекомендациях NIST SP 800-90A.
* **Интеграция с OpenSSL:** Позволяет безопасно и эффективно получать начальную энтропию.
* **Гибкость:** Алгоритм поддерживает персонализацию, что позволяет получать уникальные выходные последовательности.
* **Надёжность:** Надёжная криптографическая основа (HMAC с SHA-256) гарантирует стойкость ко всем типам атак на генератор.

**5. Описание алгоритма генерации**

**Функциональные блоки:**

* **hmac\_drbg\_init** — настройка генератора, установка начального значения Key, V, а также интеграция энтропии и параметра персонализации.
* **hmac\_drbg\_update** — обновление внутреннего состояния генератора на основе новых входных данных (при необходимости).
* **hmac\_drbg\_generate** — непосредственная генерация выходной битовой последовательности заданной длины.

**Особенности реализации:**

* Используется SHA-256 как хеш-функция в HMAC.
* Начальная энтропия формируется с помощью OpenSSL (RAND\_bytes), объём которой составляет 48 байт.
* Каждая итерация hmac\_drbg\_generate использует актуальное состояние генератора и обновляет его после завершения.

**6. Скорость выполнения тестовых заданий**

**6.1 Генерация файлов**

| **Объём данных** | **Команда запуска** | **Время CPU (сек)** | **Общее время (сек)** |
| --- | --- | --- | --- |
| 1 МБ | ./hmac\_drbg random1.bin 1 481290234 | 0.01 | 0.01 |
| 100 МБ | ./hmac\_drbg random100.bin 100 481290234 | 1.15 | 1.39 |
| 1 ГБ | ./hmac\_drbg random1000.bin 1000 481290234 | 11.04 | 13.83 |

**6.2 Генерация ключей**

| **Кол-во ключей** | **Время (сек)** |
| --- | --- |
| 1 000 | 0.03 |
| 10 000 | 0.29 |

**7. Архитектура программной реализации**

**Язык и технологии:**

* **Язык программирования:** C (соответствие стандарту C11).
* **Используемые библиотеки:** OpenSSL 3.0 для реализации HMAC и получения энтропии.

**Структура:**

* Код разделён на отдельные модули для инициализации, генерации и обновления состояния генератора.
* Данные записываются в бинарные файлы, каждый из которых содержит псевдослучайную последовательность заданного объёма.

**8. Конфигурация тестового стенда**

* **ОС:** Ubuntu 22.04.4 LTS (WSL2 на Windows 11)
* **Архитектура:** x86\_64
* **Процессор:** Intel Core i5-12400H
* **ОЗУ:** 32 ГБ
* **Компилятор:** g++ 11.4

**9. Нагрузка на ресурсы (ОЗУ и ЦП)**

**1 МБ:**

* CPU: ~66%
* Пиковое использование памяти: 3 924 КБ

**100 МБ:**

* CPU: ~83%
* Пиковое использование памяти: 103 258 КБ

**1 ГБ:**

* CPU: ~82%
* Пиковое использование памяти: 1 047 784 КБ