МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ
КАФЕДРА КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

ДИПЛОМНА РОБОТА БАКАЛАВРА

«Аналіз стійкості сучасних симетричних алгоритмів шифрування (AES, ChaCha20) до нових типів атак».

Виконав студент групи КСМ-22: Халіфа Н.Г
Науковий Керівник : к.т.н., доцент Кондакова С. В.

Актуальність та загрозова модель

- Забезпечення надійності державних, фінансових та військових систем зв'язку України.
- Перехід глобальної інфраструктури на режими автентифікованого шифрування (AEAD).
- Поява нових типів кіберзагроз: квантові обчислення, атаки по побічних каналах на основі ШІ, помилки розгортання.

Мета та завдання дослідження

- Комплексний аналіз математичної і практичної стійкості алгоритмів AES та ChaCha20, а також розробка концепції їхньої адаптивної апаратної оптимізації.

Завдання:

- Дослідити математичні моделі SPN та ARX.
- Оцінити постквантову стійкість шифрів до алгоритму Гровера.
- Систематизувати практичні вразливості та атаки по побічних каналах.
- Створити програмний утилітний комплекс для бенчмаркінгу швидкодії.

Архітектура блочного шифру AES

- Базується на структурі **SPN** (підстановочно-перестановочна мережа).
- Обчислення в скінченних полях Галуа $GF(2^8)$.
- Нелінійне перетворення: **S-Box** (забезпечує властивість Confusion).
- Лінійні дифузійні трансформації: ShiftRows та MixColumns.

Класична схема одного раунду AES (SubBytes → ShiftRows→ MixColumns→ AddRoundKey).

Архітектура потокового шифру ChaCha20

- Базується на філософії **ARX** (Addition-Rotation-XOR).
- Повна відмова від таблиць замін та складних полів Галуа.
- Нелінійність: Додавання за модулем 2^{32}.
- Дифузія: Побітові циклічні зсуви на фіксовані константи та XOR.

- Матриця внутрішнього стану ChaCha20 (16 елементів по 32 біти: константи, ключ, лічильник, нонс). Схема функції Quarter Round

Оцінка стійкості до квантових атак

- Алгоритм Гровера забезпечує квадратичне прискорення перебору ключів ($\sqrt{N}$).
- Ефективна довжина симетричного ключа **скорочується вдвічі**.
- **AES-128 є квантово-вразливим** (стійкість деградує до критичних 64 біт).
- **AES-256 та ChaCha20 є постквантово-стійкими** (зберігають 128 біт безпеки).

Алгоритм	Довжина ключа	Класична стійкість	Квантова стійкість за Гровером	Статус безпеки
AES-128	128 біт	128 біт	**64 біт**	**Вразливий** (Недостатній рівень захисту)
AES-256	256 біт	256 біт	**128 біт**	**Постквантово-стійкий** (Абсолютний захист)
ChaCha20	256 біт	256 біт	**128 біт**	**Постквантово-стійкий** (Абсолютний захист)

Уразливості практичної реалізації шифрів

- **Проблема лукап-таблиць (T-tables) в AES:** Програмне прискорення полів Галуа створює коливання часу доступу до процесорного кешу (Cache Hits/Misses).
- Зловмисник може відновити секретний ключ через аналіз часу виконання операцій.
- **Рішення для AES:** Обов'язкова апаратна підтримка інструкцій процесора (Intel AES-NI).
- **Перевага ChaCha20:** Вроджений імунітет до таймінг-атак завдяки виконанню ARX-операцій за константний час

(Процесор → Кеш пам'яті → Вимірювання часу → Зловмисник).

Ризики повторного використання вектора ініціалізації

- Обидва шифри в AEAD режимах працюють як потокові (генерують гаму).
- При повторі Nonce на одному ключі гама повністю анулюється через XOR.
- Результат: миттєве дешифрування відкритого тексту.
- У режимі AES-GCM це додатково веде до повної компрометації ключа автентифікації (Forbidden Attack).

$C_1 \oplus C_2 = P_{1\oplus}P_2$ - ця формула демонструє, як при однаковій гамі зловмисник отримує зв'язок між відкритими текстами

Розробка програмного комплексу тестування

- Мова програмування: **C++** (стандарт C++17).
- Середовище розробки: Visual Studio Code / GCC Compiler.
- Основний інструментарій: Криптографічні бібліотеки **OpenSSL API**.
- Функціонал: Модулі генерації ключів, автентифікованого шифрування великих масивів даних та швидкісного гешування

```
22      EVP_EncryptInit_ex(ctx, cipher, nullptr, key.data(), iv.data());
23      EVP_CIPHER_CTX_set_padding(ctx, 0);
24
25      int len;
26      ciphertext.resize(plaintext.size() + 32);
27
28      EVP_EncryptUpdate(ctx, ciphertext.data(), &len, plaintext.data(), plaintext.size());
29      int ciphertext_len = len;
30
31      EVP_EncryptFinal_ex(ctx, ciphertext.data() + len, &len);
32      ciphertext_len += len;
33
34      ciphertext.resize(plaintext.size()); // залишаємо строго розмір вхідних даних
35      EVP_CIPHER_CTX_free(ctx);
36  }
```

Експериментальна оцінка пропускної здатності

- Тестування проводилося на архітектурах x86_64 (з AES-NI) та ARM (без крипторозширень). І результати з консолі зберігаються також у текстовому файлі (.txt)
- **Архітектура x86_64:** AES-GCM демонструє максимальну швидкість завдяки апаратній обробці на рівні транзисторів.
- **Архітектура ARM / IoT:** Програмна емуляція AES викликає деградацію швидкості. Алгоритм **ChaCha20 виявляється у 3-4 рази швидшим.**

```
~ $ cd "D:/diploma code project" && g++ -std=c++20 main.cpp -I"C:/Program Files/OpenSSL-Win64/include" -L"C:/Program Files/OpenSSL-Win64/lib/VC/x64/MD" -lssl -lcrypto -o main.exe
D:/diploma code project $ ./main.exe
========================================================
КРИПТОАНАЛІТИЧНИЙ КОМПЛЕКС ОЦІНКИ ЗАВАДОСТІЙКОСТІ ТА ШВИДКОДІЇ
========================================================

[ЕТАП 1] Запуск навантажувального тестування (100 МБ)...
AES-256:       45.934000 ms (Швидкість: 2177.036618 MB/s)
ChaCha20:      29.736000 ms (Швидкість: 3362.927092 MB/s)

[ЕТАП 2] Аналіз строгого лавинного ефекту (128 біт)...
AES-256 Лавинний ефект:       66 бітів змінено (51.562500%)
ChaCha20 Лавинний ефект:      1 бітів змінено (0.781250%)

[ЕТАП 3] Симуляція спотворення пакетів у нестабільному каналі (4G / РЕБ)...
Дешифрація AES-256 після завади:   ЗРУЙНОВАНО 65 бітів (весь блок даних втрачено).
Дешифрація ChaCha20 после завади:  ЗРУЙНОВАНО 1 біт (помилка локалізована, дані вціліли).
========================================================
```

Концепція адаптивного вибору шифру (Cipher-Agility)

- Динамічний вибір криптографічного примітиву на основі аналізу апаратних можливостей процесора в реальному часі.
- Автоматична інспекція прапорців CPU (наявність інструкцій AES-NI або ARM CE).
- **Практична цінність:** Оптимізація VPN-серверів та гетерогенних мереж (на базі WireGuard / IPsec). Сервери використовують апаратний AES, мобільні клієнти та IoT — енергоефективний ChaCha20.

[Клієнт підключається]→ [Перевірка CPU: Є апаратний AES?] →Так (Ввімкнути AES-GCM) / Ні (Ввімкнути ChaCha20-Poly1305)

Висновки

- **Проведено** порівняльний аналіз математичних моделей симетричного шифрування та обґрунтовано безальтернативність переходу на **256-бітні ключі** в умовах постквантових загроз.
- **Доведено**, що класична програмна реалізація AES без апаратної підтримки є критично вразливою до атак по побічних каналах через коливання часу доступу до процесорного кешу.
- **Розроблено** програмний комплекс на C++ та **експериментально оцінено** лавинний ефект шифрів в умовах завад у каналі зв'язку.
- **Обґрунтовано** ефективність концепції **Cipher-Agility** для оптимізації пропускної здатності та енергоефективності гетерогенних комп'ютерних мереж.

ДЯКУЮ ЗА УВАГУ !