МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ ЦИФРОВИХ ТЕХНОЛОГІЙ, ДИЗАЙНУ ТА ТРАНСПОРТУ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЄКТУВАННЯ ТА ДИЗАЙНУ

**МЕТОДИЧНІ ВКАЗІВКИ**

**до виконання розрахунково графічної роботи**

**з дисципліни**

**«****ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ»**

для студентів усіх форм навчання

**Затверджено на засіданні**

Кафедри ІТПД Протокол № 3 від 29.10.2024

**Укладачі:**

Коляда А.С. к.т.н., доц. кафедри

 Лопаков О.С., ст. викладач

 Космачевський В.В., ст. викладач

**Одеса – 2024**

**Зміст**

[1 Вступ 3](#__RefHeading___Toc13709_358955385)

[2 Мета та завдання 3](#__RefHeading___Toc13711_358955385)

[3 Тематика 4](#__RefHeading___Toc13713_358955385)

[3.1 Розробка стеганографічного шифру для передачі тексту 4](#__RefHeading___Toc14084_358955385)

[3.2 Генерація псевдовипадкових чисел для криптографії: принципи, алгоритми та реалізація 4](#__RefHeading___Toc1065_1917333554)

[3.3 Режими роботи блочного шифру та їх вплив на безпеку 4](#__RefHeading___Toc14232_358955385)

[3.4 Комбінований метод шифрування (гібридні криптосистеми): принципи, структура та застосування 4](#__RefHeading___Toc1067_1917333554)

[3.5 Triple DES: принципи роботи, безпека та застосування у сучасних інформаційних системах 5](#__RefHeading___Toc1069_1917333554)

[3.6 Еліптичні криві у цифрових підписах: алгоритм ECDSA та його застосування 5](#__RefHeading___Toc1071_1917333554)

[3.7 Еліптичні криві в обміні ключами: алгоритм ECDH та його застосування 5](#__RefHeading___Toc1073_1917333554)

[3.8 Алгоритм шифрування ChaCha20: принципи роботи, особливості та застосування 6](#__RefHeading___Toc1075_1917333554)

[3.9 Алгоритми хешування Blake2/Blake3: еволюція та ефективність у криптографії 6](#__RefHeading___Toc1077_1917333554)

[3.10 Алгоритм хешування паролів Argon2: структура, налаштування та застосування 6](#__RefHeading___Toc1079_1917333554)

[4 Критерії оцінювання 7](#__RefHeading___Toc13715_358955385)

[5 Рекомендації до оформлення 7](#__RefHeading___Toc12573_358955385)

[5.1 Загальні вимоги 7](#__RefHeading___Toc14549_358955385)

[5.2 Оформлення тексту 7](#__RefHeading___Toc14551_358955385)

[5.3 Таблиці, рисунки та формули 7](#__RefHeading___Toc14553_358955385)

[5.4 Оформлення списку літератури 7](#__RefHeading___Toc14555_358955385)

[5.5 Вимоги до змісту 8](#__RefHeading___Toc14557_358955385)

[6 Література 9](#__RefHeading___Toc13717_358955385)

# Вступ

Розрахунково-графічна робота (РГР) ‒ це самостійне дослідження студента, індивідуальне завдання, яке передбачає вирішення конкретної практичної, навчальної задачі з використанням відомого, а також (або) самостійно вивченого теоретичного матеріалу. Виконуючи РГР студент повинен продемонструвати вміння визначати мету, виділяти задачі, формулювати проблеми та знаходити, способи їх розв'язання з використанням знань та умінь, отриманих в процесі вивчення дисципліни.

Основну частину розрахункової роботи складають розрахунки, які можуть супроводжуватися ілюстративним матеріалом: графіками, векторними діаграмами, гістограмами тощо. Значну частину такої роботи складає графічний матеріал, який виконується відповідно до чинних нормативних вимог та з обов’язковим застосуванням комп’ютерної графіки, якщо це визначено завданням.

Працюючи над розрахунково-графічною роботою студент отримує вміння та навички, що будуть корисними в майбутньому — при виконанні більш складних завдань (дипломна робота, дисертація, наукове дослідження). РГР висвітлює відношення студента стосовно досліджуваного питання, його власні погляди на розв’язання поставленої задачі та досягнення визначеної мети. Метою роботи є систематизація, поглиблення і закріп­лення знань студентами з відповідних питань програми, розвиток на­вичок самостійної роботи, практичного застосування теоретичних знань. Виконання роботи потребує попереднього вивчення навчальної, науково-виробничої і довідкової літератури, систематизації отриманої інформації, що розвиває здатність до самостійного оновлення і придбання знань.

# Мета та завдання

 **Метою даної РГР** є формування глибоких теоретичних знань та практичних навичок у галузі сучасної криптографії, стеганографії, генерації псевдовипадкових чисел, цифрових підписів та алгоритмів хешування. Виконання роботи спрямоване на розуміння основних принципів безпеки інформаційних систем та реалізацію конкретних криптографічних методів і алгоритмів для забезпечення конфіденційності, цілісності та автентичності даних.

 **Задачі РГР**:

* **Ознайомлення з основами та сучасними технологіями в галузі інформаційної безпеки**
* **Визначення переваг і недоліків різних алгоритмів шифрування, цифрових підписів та хеш-функцій**
* **Проведення порівняльного аналізу ефективності та продуктивності алгоритмів шифрування**
* **Дослідження стійкості криптографічних методів**

 **Виконання РГР забезпечує**:

* Розуміння теоретичних основ криптографії, стеганографії та хешування.
* Практичні навички реалізації та тестування криптографічних алгоритмів.
* Здатність аналізувати стійкість алгоритмів до атак та порівнювати їх ефективність.
* Підготовку до реального застосування криптографічних методів у захисті інформації.

# Тематика

Студент отримує тему для РГР зі списка в цьому розділі, які представлені нище з коротким описом або рекомендаціями. Під моделюванням процесу шифрування мається на увазі формальне відтворення всіх етапів шифрування та розшифрування для конкретного алгоритму з метою його кращого розуміння, аналізу та тестування. Це може включати як теоретичні, так і практичні аспекти роботи алгоритму.

## Розробка стеганографічного шифру для передачі тексту

Ознайомитися з основами стеганографії як методу приховування інформації у несекретних даних. Розглянути алгоритми стеганографії для тексту, зображень чи аудіо (наприклад, методи Least Significant Bit). Проаналізувати переваги стеганографії у поєднанні з шифруванням для забезпечення конфіденційності та непомітності передачі даних. Реалізувати стеганографічний алгоритм для приховування тексту у зображеннях. Продемонструвати процес приховування, витягу тексту та порівняння змінених і початкових файлів. **Побудувати графік залежності часу приховування/витягу даних від розміру файлу.**

## ****Генерація псевдовипадкових чисел для криптографії: принципи, алгоритми та реалізація****

 Ознайомитися з основами генерації псевдовипадкових чисел (PRNG) та криптографічно безпечних генераторів (CSPRNG). Розглянути вимоги до CSPRNG, зокрема непередбачуваність, рівномірність розподілу та стійкість до атак. Дослідити приклади сучасних алгоритмів, таких як Fortuna, Yarrow та алгоритми на основі еліптичних кривих. Реалізувати криптографічно безпечний генератор псевдовипадкових чисел на вибраній мові програмування. Продемонструвати використання генератора для створення ключів шифрування або nonce. **Побудувати графік залежності часу генерації чисел від їхньої довжини.**

## Режими роботи блочного шифру та їх вплив на безпеку

Ознайомитися з режимами роботи блочних шифрів (ECB, CBC, CFB, OFB, CTR) та їхньою роллю у забезпеченні конфіденційності даних. Розглянути вплив вибору режиму на стійкість до атак (наприклад, на повторюваність патернів у даних). Проаналізувати переваги та недоліки кожного режиму, а також їхню продуктивність. Реалізувати різні режими роботи блочного шифру (наприклад, AES) на вибраній мові програмування. Продемонструвати, як вибір режиму впливає на результат шифрування даних із патернами. **Побудувати графік продуктивності (час шифрування) для різних режимів роботи блочного шифру.**

## ****Комбінований метод шифрування (гібридні криптосистеми): принципи, структура та застосування****

 Ознайомитися з принципами гібридних криптосистем, які поєднують переваги симетричного та асиметричного шифрування. Розглянути основні етапи роботи, зокрема генерацію симетричного ключа, його шифрування асиметричним методом та передачу зашифрованих даних. Проаналізувати приклади сучасних протоколів, які використовують гібридні криптосистеми (наприклад, TLS). Порівняти продуктивність та безпеку гібридних підходів із виключно симетричними або асиметричними методами. Реалізувати гібридну криптосистему, що поєднує AES та RSA. Продемонструвати шифрування тексту або файлів, використовуючи реалізовану систему. **Побудувати графік залежності часу виконання шифрування/дешифрування від розміру даних.**

## ****Triple DES: принципи роботи, безпека та застосування у сучасних інформаційних системах****

 Ознайомитися з принципами роботи алгоритму Triple DES (3DES), який є вдосконаленням DES (Data Encryption Standard). Розглянути його структуру, включаючи потрійне шифрування з використанням трьох ключів, режими роботи (наприклад, ECB, CBC) та способи забезпечення стійкості до атак. Проаналізувати переваги та недоліки Triple DES у порівнянні з сучасними симетричними алгоритмами (наприклад, AES), зокрема з точки зору продуктивності, безпеки та стійкості до криптографічних атак. Реалізувати алгоритм Triple DES програмно на вибраній мові програмування. Продемонструвати процес шифрування та дешифрування текстових або бінарних даних за допомогою Triple DES. **Побудувати графік залежності швидкості шифрування/дешифрування від розміру даних або порівняти продуктивність Triple DES із AES.**

## ****Еліптичні криві у цифрових підписах: алгоритм ECDSA та його застосування****

 **Ознайомитися з принципами роботи алгоритму цифрового підпису на еліптичних кривих (ECDSA). Розглянути математичну основу, включаючи використання еліптичних кривих для створення ключів, операції підпису та перевірки. Вивчити переваги ECDSA у порівнянні з іншими алгоритмами цифрового підпису (RSA, DSA), зокрема ефективність, компактність ключів та швидкість виконання. Дослідити сучасні стандарти еліптичних кривих (наприклад, secp256k1, Curve25519) та їх використання у таких сферах, як блокчейн, електронна пошта та цифрові сертифікати. Реалізувати програмно алгоритм ECDSA на вибраній мові програмування. Продемонструвати створення та перевірку цифрового підпису для текстових даних чи файлів. Побудувати графік залежності часу створення та перевірки підпису від розміру кривої.**

## ****Еліптичні криві в обміні ключами: алгоритм ECDH та його застосування****

 Ознайомитися з принципами роботи алгоритму обміну ключами на основі еліптичних кривих (ECDH). Розглянути математичну основу, зокрема операції з використанням публічних та приватних ключів для спільного створення секретного ключа. Пояснити, як ECDH забезпечує конфіденційність та захист даних у комунікаційних протоколах (TLS, Signal, WireGuard). Порівняти ефективність ECDH з іншими алгоритмами обміну ключами, такими як Diffie-Hellman та RSA. Реалізувати програмно алгоритм ECDH на вибраній мові програмування. Продемонструвати процес узгодження ключа між двома сторонами та його використання для шифрування даних. **Побудувати графік залежності часу узгодження ключа від розміру кривої або порівняти швидкість ECDH з традиційним Diffie-Hellman.**

## Алгоритм шифрування ChaCha20: принципи роботи, особливості та застосування

 Ознайомитися з принципами роботи алгоритму ChaCha20, який є удосконаленою версією алгоритму Salsa20. Розглянути структуру ChaCha20, включаючи побудову блоку шифрування, використання ключа та nonce для забезпечення унікальності шифру. Вивчити переваги алгоритму, такі як висока швидкість, простота реалізації та захист від атак на поточне шифрування. Дослідити використання ChaCha20 у сучасних протоколах (TLS, WireGuard) та інших додатках. Реалізувати алгоритм ChaCha20 програмно, продемонструвати процес шифрування та дешифрування текстових даних. Побудувати графік залежності швидкості шифрування/дешифрування від розміру даних.

## ****Алгоритми хешування Blake2/Blake3: еволюція та ефективність у криптографії****

 Ознайомитися з історією створення Blake2 та його наступника Blake3, їхньою структурою та принципами роботи. Розглянути основні переваги Blake2 у порівнянні з SHA-2 та SHA-3, зокрема швидкість, енергоефективність і криптографічну стійкість. Дослідити, як Blake3 покращує масштабованість і продуктивність через підтримку паралельного виконання. Вивчити сфери застосування, включаючи цифрові підписи, хмарні сховища та блокчейн. Реалізувати алгоритм Blake2 або Blake3 програмно та продемонструвати його використання для гешування файлів чи перевірки цілісності даних. **Побудувати графік порівняння продуктивності (час обчислення) між Blake2, Blake3 та SHA-256 при різних розмірах даних.**

## ****Алгоритм хешування паролів Argon2: структура, налаштування та застосування****

 Ознайомитися з принципами роботи Argon2 — алгоритму хешування паролів, який став переможцем Password Hashing Competition (PHC). Розглянути основні різновиди Argon2 (Argon2i, Argon2d, Argon2id), їхні особливості та налаштування параметрів (кількість ітерацій, обсяг пам’яті, кількість потоків). Дослідити стійкість алгоритму до атак грубої сили та інші аспекти криптографічної безпеки. Порівняти Argon2 із іншими алгоритмами для хешування паролів (наприклад, PBKDF2, bcrypt). Реалізувати Argon2 програмно, продемонструвати створення гешу для паролів та перевірку автентифікації. **Побудувати графік залежності часу обчислення гешу від кількості ітерацій або обсягу пам’яті.**

# Критерії оцінювання

* Змістовність (30%) – відповідність роботи темі, повнота викладення матеріалу;
* Практична частина (30%) – коректність реалізації завдань та обґрунтованість результатів;
* Оформлення (20%) – відповідність вимогам оформлення;
* Своєчасність здачі (10%) – робота, виконана із запізненням, може оцінюватися зниженою оцінкою;
* Творчий підхід та оригінальність (10%).

# Рекомендації до оформлення

## Загальні вимоги

* + - Робота повинна бути виконана українською мовою.
		- Обсяг роботи: 10-15 сторінок (без урахування додатків).
		- Виклад матеріалу повинен бути логічним, структурованим і послідовним.
		- У роботі не допускаються граматичні та орфографічні помилки.

## Оформлення тексту

* Шрифт: Times New Roman, 12-14 pt.
* Міжрядковий інтервал: 1.5.
* Вирівнювання тексту: по ширині.
* Абзацний відступ: 1.25 см.
* Поля: з усіх боків по 2 см.
* Нумерація сторінок: у правому нижньому куті, починаючи з другої сторінки (зміст).
* Розділи та підрозділи: нумеруються арабськими цифрами (наприклад, 1, 1.1, 1.2).
* Заголовки: виділяються жирним шрифтом, без крапок у кінці.

## Таблиці, рисунки та формули

* Таблиці:
	+ Вирівнюються по центру сторінки, нумеруються (наприклад, "Таблиця 1").
	+ Назва таблиці подається зверху.
* Рисунки:
	+ Вирівнюються по центру, нумеруються (наприклад, "Рисунок 1").
	+ Підпис під рисунком із зазначенням його номера та назви.
* Формули:
	+ Нумерація у круглих дужках праворуч (наприклад, (1)).
	+ Формули вирівнюються по центру.

## Оформлення списку літератури

* Оформлюється за стандартом ДСТУ 8302:2015 «Бібліографічне посилання. Загальні положення та правила складання»
* Приклади:
	+ Книга: Автор(и). Назва. Місце видання: Видавництво, рік. Кількість сторінок.
	+ Стаття: Автор(и). Назва статті. Назва журналу. Рік. Номер. Сторінки.
	+ Інтернет-ресурс: Автор (якщо є). Назва. URL-адреса. Дата звернення.

## Вимоги до змісту

* + - Титульна сторінка
			* Назва вищого закладу освіти, інституту, кафедри;
			* Назва дисципліни і теми;
			* Прізвище, ім'я та по-батькові студента, група;
			* Прізвище, ініціали викладача;
			* Місто та рік.
		- Зміст
			* Перелік розділів з вказанням сторінок.
		- Вступ
			* Актуальність теми;
			* Мета та завдання роботи;
			* Опис методів, які будуть використані для досягнення мети;
			* Структура роботи (короткий опис розділів).
		- Теоретична частина
			* Докладний виклад теоретичних основ, що стосуються теми;
			* Опис методів, алгоритмів чи моделей, які використовуються у роботі;
			* Обґрунтування вибору технологій, протоколів чи алгоритмів (у разі потреби);
			* Короткий огляд сучасного стану питання (включаючи можливі вразливості або перспективи вдосконалення).
		- Практична частина
			* Опис розробленого програмного або математичного рішення;
			* Псевдокод з коментарями або витяги з програмного коду (повний код програми має бути представлений у додатку);
			* Інструкції щодо виконання експериментів або тестів;
			* Результати виконання (тестові приклади, таблиці, графіки, візуалізації);
			* Аналіз коректності отриманих результатів.
		- Висновки
			* Узагальнення отриманих результатів.
			* Оцінка досягнення поставленої мети.
			* Рекомендації щодо застосування результатів.
		- Список використаних джерел
			* Використання актуальних наукових джерел та нормативних документів;
			* Включення посилань на стандарти, статті, книги, ресурси з офіційних сайтів;
			* Мінімальна кількість джерел: 5-7.
		- Додатки
			* Повний код програми;
			* Інструкції для запуску програмного забезпечення (якщо передбачено);
			* Графіки, схеми, таблиці, інші допоміжні матеріали.

# Література

* + Брюс Шнайдер. Криптографія: принципи і практика. – Київ: ВД "Професіонал", 2017. – 432 с.
	+ Мороз, В. В. Основи криптографії та захисту інформації. – Київ: Кондор, 2012. – 200 с.
	+ Шевченко, В. І. Криптографія та інформаційна безпека. – Харків: ХНУРЕ, 2016. – 300 с.
	+ Камінський, О. О. Захист інформації: теорія та практика. – Київ: Либідь, 2015. – 400 с.
	+ Лисенко, В. В. Методи і засоби криптографії. – Одеса: Одеська національна академія харчових технологій, 2018. – 250 с.
	+ Романенко, А. В. Теоретичні основи криптографії. – Львів: Львівська політехніка, 2020. – 220 с.
	+ Stallings, W. Cryptography and Network Security: Principles and Practice. – 7th ed. – Boston: Pearson, 2017. – 672 p.
	+ Paar, C., & Pelzl, J. Understanding Cryptography: A Textbook for Students and Practitioners. – Berlin: Springer, 2010. – 362 p.
	+ Stinson, D. R. Cryptography: Theory and Practice. – 3rd ed. – Boca Raton: Chapman & Hall/CRC, 2006. – 688 p.
	+ NIST Special Publication 800-131A. Transitioning the Use of Cryptographic Algorithms and Key Lengths. – National Institute of Standards and Technology, 2019. – 31 p.
	+ FIPS PUB 197. Announcing the Advanced Encryption Standard (AES). – National Institute of Standards and Technology, 2001. – 60 p.