**ПОЛНЫЙ КУРС КРИПТОГРАФИИ: ТЕОРИЯ, ПРАКТИКА, ЗАДАНИЯ**

**РАЗДЕЛ 1: МЕТОДЫ ПОДСТАНОВКИ (ЗАМЕЩЕНИЯ)**

**1.1 ТЕОРЕТИЧЕСКАЯ СУТЬ МЕТОДА**

**📚 СУТЬ МЕТОДА:**  
Каждая буква текста заменяется на другую букву по определенному правилу. Все буквы остаются теми же, но меняются их "имена". Представьте, что у каждой буквы есть секретное имя, и мы используем только эти секретные имена.

**🔑 Ключевые принципы:**

* **Сохранение позиций** - буквы остаются на своих местах
* **Изменение символов** - каждая буква заменяется на другую
* **Простота** - легко шифровать и расшифровывать
* **Уязвимость** - легко взломать, изучая частоты букв

**📖 Историческое развитие:**

* **Шифр Цезаря** (I в. до н.э.) - самый простой, сдвиг на 3 позиции
* **Аффинный шифр** (средневековье) - математическая формула
* **Шифр Виженера** (XVI век) - несколько алфавитов

**1.2 ПОДРОБНЫЕ ПРИМЕРЫ С ПОШАГОВЫМ ОБЪЯСНЕНИЕМ**

**Пример 1: Шифр Цезаря**

**🎯 СУТЬ:** Сдвигаем каждую букву на фиксированное число позиций вперед по алфавиту.

**🔢 РАЗБОР "ATTACK" с k=3:**

text

Алфавит: A=0, B=1, C=2, D=3, E=4, F=5, G=6, H=7, I=8, J=9, K=10, L=11, M=12,

N=13, O=14, P=15, Q=16, R=17, S=18, T=19, U=20, V=21, W=22, X=23, Y=24, Z=25

A(0) → (0+3) mod 26 = 3 → D

T(19) → (19+3) mod 26 = 22 → W

T(19) → (19+3) mod 26 = 22 → W

A(0) → (0+3) mod 26 = 3 → D

C(2) → (2+3) mod 26 = 5 → F

K(10) → (10+3) mod 26 = 13 → N

**✅ Результат:** DWWDFN

**🔍 ОТКУДА ЧИСЛА:**

* Буквы превращаем в номера (A=0, B=1, ... Z=25)
* Прибавляем ключ (3)
* Если результат >25, вычитаем 26 (mod 26)
* Превращаем обратно в буквы

**Пример 2: Аффинный шифр**

**🎯 СУТЬ:** Используем математическую формулу: число\_буквы × a + b

**🔢 РАЗБОР "HELP" с E(x) = (5x + 3) mod 26:**

text

H(7) → (5×7 + 3) = 35 + 3 = 38 → 38 mod 26 = 12 → M

E(4) → (5×4 + 3) = 20 + 3 = 23 → X

L(11) → (5×11 + 3) = 55 + 3 = 58 → 58 mod 26 = 6 → G

P(15) → (5×15 + 3) = 75 + 3 = 78 → 78 mod 26 = 0 → A

**✅ Результат:** MXGA

**🔍 ОТКУДА ЧИСЛА:**

* Умножаем номер буквы на 5
* Прибавляем 3
* Берем остаток от деления на 26
* Преобразуем в букву

**Пример 3: Шифр Виженера**

**🎯 СУТЬ:** Для каждой буквы используем свой сдвиг, который задается ключевым словом.

**🔢 РАЗБОР "CRYPTOGRAPHY" с ключом "KEY":**

text

Текст: C R Y P T O G R A P H Y

Ключ: K E Y K E Y K E Y K E Y

Сдвиги: 10 4 24 10 4 24 10 4 24 10 4 24

C(2) + 10 = 12 → M

R(17) + 4 = 21 → V

Y(24) + 24 = 48 → 48-26=22 → W

P(15) + 10 = 25 → Z

T(19) + 4 = 23 → X

O(14) + 24 = 38 → 38-26=12 → M

G(6) + 10 = 16 → Q

R(17) + 4 = 21 → V

A(0) + 24 = 24 → Y

P(15) + 10 = 25 → Z

H(7) + 4 = 11 → L

Y(24) + 24 = 48 → 48-26=22 → W

**✅ Результат:** MVWZXMQVYZLW

**🔍 ОТКУДА ЧИСЛА:**

* Ключ повторяем: KEYKEYKEYKEY
* Для каждой позиции: буква\_текста + буква\_ключа
* Если сумма >25, вычитаем 26

**1.3 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ (20 заданий)**

**🎯 Уровень 1 (легкие):**

1. Зашифруйте "PYTHON" шифром Цезаря с k=7
2. Расшифруйте "MJQQT" (Цезарь, k=5)
3. Создайте шифр: A→N, B→O, C→P... Зашифруйте "TEST"
4. Зашифруйте "ABC" заменой A=01, B=02, C=03...
5. Расшифруйте "KHOOR" (Цезарь, k=3)
6. Зашифруйте "HELLO" сдвигом A→F, B→G, C→H...
7. Расшифруйте "D" при известном сдвиге +3
8. Создайте шифр с ключом "C"=+2
9. Зашифруйте "ABC" с k=25
10. Расшифруйте "BCD" с k=1

**🎯 Уровень 2 (средние):**  
11. Взломайте "DWWDFN" (Цезарь, английское слово)  
12. Зашифруйте "CODE" с E(x) = (3x + 7) mod 26  
13. Расшифруйте "ARMM" с D(x) = 9(y - 7) mod 26  
14. Найдите обратный элемент к 7 mod 26  
15. Взломайте аффинный шифр "ZQN"  
16. Зашифруйте "MESSAGE" с ключом Виженера "KEY"  
17. Расшифруйте "LXFOPV" с ключом Виженера "LEMON"  
18. Найдите все возможные ключи аффинного шифра  
19. Создайте шифр с ключевым словом "CRYPTO"  
20. Проанализируйте частоты в зашифрованном тексте

**РАЗДЕЛ 2: МЕТОДЫ ПЕРЕСТАНОВКИ**

**2.1 ТЕОРЕТИЧЕСКАЯ СУТЬ МЕТОДА**

**📚 СУТЬ МЕТОДА:**  
Буквы остаются теми же, но меняются их местами. Представьте, что вы переставляете мебель в комнате - все предметы те же, но стоят в другом порядке.

**🔑 Ключевые принципы:**

* **Сохранение букв** - все буквы те же самые
* **Изменение порядка** - буквы стоят в другом порядке
* **Устойчивость** - сложнее взломать, чем подстановку
* **Визуализация** - легко представить в виде таблицы

**2.2 ПОДРОБНЫЕ ПРИМЕРЫ С ПОШАГОВЫМ ОБЪЯСНЕНИЕМ**

**Пример 1: Одиночная перестановка**

**🎯 СУТЬ:** Записываем текст в таблицу по строкам, читаем по столбцам.

**🔢 РАЗБОР "CRYPTOGRAPHY" в таблицу 3×4:**

text

Записываем по строкам:

[Строка 1] C R Y P

[Строка 2] T O G R

[Строка 3] A P H Y

Читаем по столбцам:

[Столбец 1] C T A

[Столбец 2] R O P

[Столбец 3] Y G H

[Столбец 4] P R Y

**✅ Результат:** CTAROPYGHPHY

**🔍 КАК РАБОТАЕТ:**

* Делим текст на строки одинаковой длины
* Читаем сверху вниз по каждому столбцу
* Объединяем все столбцы

**Пример 2: Перестановка с ключом**

**🎯 СУТЬ:** Используем ключевое слово, чтобы определить порядок чтения столбцов.

**🔢 РАЗБОР "ATTACKATDAWN" с ключом "KEY":**

text

Ключ "KEY": K=11, E=5, Y=25 → сортируем: E(1), K(2), Y(3)

Записываем в таблицу 4×3:

[Строка 1] A T T

[Строка 2] A C K

[Строка 3] A T D

[Строка 4] A W N X (дополняем X)

Читаем в порядке 2-1-3:

[Столбец 2] T C T W → TCTW

[Столбец 1] A A A A → AAAA

[Столбец 3] T K D N → TKDN

**✅ Результат:** TCTWAAAATKDN

**🔍 КАК РАБОТАЕТ:**

* Ключ определяет порядок столбцов
* Буквы ключа нумеруем по алфавитному порядку
* Читаем столбцы в порядке этих номеров

**Пример 3: Двойная перестановка**

**🎯 СУТЬ:** Два раза переставляем текст с разными ключами.

**🔢 РАЗБОР "ENEMYATTACKS" с ключами "CODE", "KEY":**

text

ПЕРВАЯ ПЕРЕСТАНОВКА (ключ "CODE"):

Ключ "CODE": C=3, O=15, D=4, E=5 → сортируем: C(1), D(2), E(3), O(4)

Таблица 3×4:

E N E M

Y A T T

A C K S

Читаем столбцы 1-3-4-2:

E Y A

E T K

M T S

N A C

ВТОРАЯ ПЕРЕСТАНОВКА (ключ "KEY"):

Ключ "KEY": K=11, E=5, Y=25 → сортируем: E(1), K(2), Y(3)

Таблица 4×3:

E Y A

E T K

M T S

N A C

Читаем строки 2-1-3:

Y E A

E Y A

T T S

**✅ Результат:** YEAEYATTS

**2.3 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ (20 заданий)**

**🎯 Уровень 1 (легкие):**

1. Переставьте "HELLO" в обратном порядке
2. Примените перестановку 3-1-4-2 к "WORD"
3. Зашифруйте "MATRIX" в таблицу 2×3, читая по столбцам
4. Расшифруйте "HLOEL" (парная перестановка)
5. Запишите "CRYPTO" в таблицу 2×3 по строкам
6. Прочитайте "ABCDEF" из таблицы 2×3 по столбцам
7. Примените перестановку 2-4-1-3 к "TEXT"
8. Восстановите исходный текст из "AEIBFJCGKDHL"
9. Зашифруйте "COMPUTER" в таблицу 4×2
10. Расшифруйте "ADGJBEH KCFIL" (таблица 3×4)

**🎯 Уровень 2 (средние):**  
11. Зашифруйте "CRYPTOGRAPHY" в таблицу 4×3 зигзагом  
12. Расшифруйте "AEIMBDFHJNCGKO" (таблица 3×5 по столбцам)  
13. Примените перестановку с ключом "CODE" к "SECRET"  
14. Зашифруйте "COMPUTER" вертикально с ключом "KEY"  
15. Расшифруйте "HLCERAEOPTM" (исходное "TELEPHONE")  
16. Зашифруйте "INFORMATION" с ключом "SECRET"  
17. Восстановите таблицу для "TSOXETNMAE"  
18. Примените двойную перестановку к "MESSAGE"  
19. Найдите размер таблицы для зашифрованного текста  
20. Создайте перестановку с динамическим ключом

**РАЗДЕЛ 3: БЛОЧНЫЕ ШИФРЫ**

**3.1 ТЕОРЕТИЧЕСКАЯ СУТЬ МЕТОДА**

**📚 СУТЬ МЕТОДА:**  
Делим текст на блоки фиксированного размера и каждый блок шифруем отдельно. Представьте, что режете торт на куски и каждый кусок украшаете отдельно.

**🔑 Ключевые принципы:**

* **Блочная обработка** - работа с частями текста
* **Раундовые преобразования** - многократное применение операций
* **Диффузия** - изменение одного бита влияет на многие
* **Перемешивание** - сложная зависимость выхода от входа

**3.2 ПОДРОБНЫЕ ПРИМЕРЫ С ПОШАГОВЫМ ОБЪЯСНЕНИЕМ**

**Пример 1: Сеть Фейстеля**

**🎯 СУТЬ:** Делим блок на две половинки и на каждом раунде меняем их местами, "перемешивая" с помощью функции.

**🔢 РАЗБОР 2 раунда для "DATA", ключ "K", F(R,K)=R⊕K:**

text

ИСХОДНЫЕ ДАННЫЕ:

Текст: "DATA" = D(3) A(0) T(19) A(0)

Делим: ЛЕВАЯ = "DA" = (3,0), ПРАВАЯ = "TA" = (19,0)

Ключ: "K" = 10

РАУНД 1:

ЛЕВАЯ₁ = ПРАВАЯ₀ = "TA" = (19,0)

F(ПРАВАЯ₀, K) = "TA" ⊕ "KK" = (19⊕10, 0⊕10) = (25,10) = "ZK"

ПРАВАЯ₁ = ЛЕВАЯ₀ ⊕ F(ПРАВАЯ₀, K) = "DA" ⊕ "ZK" = (3⊕25, 0⊕10) = (22,10) = "WK"

РАУНД 2:

ЛЕВАЯ₂ = ПРАВАЯ₁ = "WK" = (22,10)

F(ПРАВАЯ₁, K) = "WK" ⊕ "KK" = (22⊕10, 10⊕10) = (32,20) → mod 26 = (6,20) = "GU"

ПРАВАЯ₂ = ЛЕВАЯ₁ ⊕ F(ПРАВАЯ₁, K) = "TA" ⊕ "GU" = (19⊕6, 0⊕20) = (21,20) = "VU"

**✅ Результат:** "WK" + "VU" = "WKVU"

**🔍 КАК РАБОТАЕТ:**

* Делим блок пополам
* На каждом раунде: левая = предыдущая правая, правая = предыдущая левая XOR F(правая, ключ)
* Функция F может быть любой

**Пример 2: Режим CBC**

**🎯 СУТЬ:** Каждый блок перед шифрованием смешиваем с предыдущим зашифрованным блоком.

**🔢 РАЗБОР "MESSAGE" с IV="INIT", шифр Цезаря k=3:**

text

ПОДГОТОВКА:

IV: "INIT" = I(8) N(13) I(8) T(19)

Текст: "MESSAGE" → блоки: "MESS" = (12,4,18,18), "AGEX" = (0,6,4,23)

БЛОК 1 "MESS":

Смешиваем с IV: (12⊕8, 4⊕13, 18⊕8, 18⊕19) = (4,17,26,1) → mod 26 = (4,17,0,1) = "ERAB"

Шифруем Цезарем: (4+3,17+3,0+3,1+3) = (7,20,3,4) = "HUDE"

БЛОК 2 "AGEX":

Смешиваем с "HUDE": (0⊕7,6⊕20,4⊕3,23⊕4) = (7,26,7,27) → mod 26 = (7,0,7,1) = "HAB"

Шифруем Цезарем: (7+3,0+3,7+3,1+3) = (10,3,10,4) = "KDKE"

**✅ Результат:** "HUDEKDKE"

**🔍 КАК РАБОТАЕТ:**

* Первый блок XOR с IV
* Каждый следующий блок XOR с предыдущим шифрблоком
* Результат шифруем

**3.3 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ (20 заданий)**

**🎯 Уровень 1 (легкие):**

1. Разделите "COMPUTER" на блоки по 4 символа
2. Примените XOR с "CODE" к блоку "TEXT"
3. Зашифруйте "HELL" в режиме ECB с XOR-шифрованием
4. Выполните 1 раунд Фейстеля для "AB", ключ "K", F(R,K)=R
5. Разделите "INFORMATION" на блоки по 5 символов
6. Примените операцию AND с "1111" к блоку "DATA"
7. Зашифруйте "TEST" в режиме ECB с шифром Цезаря k=5
8. Выполните начальную перестановку IP для блока "ABCD"
9. Разделите "CRYPTOGRAPHY" на блоки по 3 символа
10. Примените операцию OR с "0101" к блоку "WORD"

**🎯 Уровень 2 (средние):**  
11. Реализуйте 2 раунда Фейстеля для "DATA", ключ "SECRET", F(R,K)=R⊕K  
12. Зашифруйте "MESSAGE" в режиме CBC с IV="INIT", базовый шифр - Цезарь k=3  
13. Примените S-блок: 1010 → 1100, 1100 → 1010 к "A"  
14. Реализуйте P-блок перестановки 1→3, 2→1, 3→4, 4→2 для "WORD"  
15. Зашифруйте "BLOCKCIPHER" в режиме CFB с IV="START"  
16. Реализуйте 3 раунда Фейстеля для "HELLO", ключ "KEY", F(R,K)=R+K mod 26  
17. Примените компрессирующую функцию f(x)=x² mod 100 к блоку "12"  
18. Реализуйте режим OFB для "SECRET", IV="VECTOR"  
19. Создайте раундовый ключ из мастер-ключа "SECRETKEY"  
20. Реализуйте операцию MixColumns для упрощенного AES

**РАЗДЕЛ 4: ПОТОКОВЫЕ ШИФРЫ**

**4.1 ТЕОРЕТИЧЕСКАЯ СУТЬ МЕТОДА**

**📚 СУТЬ МЕТОДА:**  
Шифруем текст побитно, смешивая с псевдослучайной последовательностью (гаммой). Представьте, что у вас есть бесконечная лента со случайными числами, которую вы накладываете на текст.

**🔑 Ключевые принципы:**

* **Побитная обработка** - работа с отдельными битами
* **Генерация гаммы** - создание ключевого потока
* **Синхронизация** - одинаковые гаммы у отправителя и получателя
* **Простота** - быстрее блочных шифров

**4.2 ПОДРОБНЫЕ ПРИМЕРЫ С ПОШАГОВЫМ ОБЪЯСНЕНИЕМ**

**Пример 1: Гаммирование**

**🎯 СУТЬ:** Смешиваем текст с гаммой с помощью операции XOR.

**🔢 РАЗБОР "HI" с гаммой "KEY":**

text

Текст: "HI" = H(7) I(8)

Гамма: "KEY" берем первые 2 символа: K(10) E(4)

H(7) ⊕ K(10):

7 = 00000111

10 = 00001010

XOR = 00001101 = 13 = N

I(8) ⊕ E(4):

8 = 00001000

4 = 00000100

XOR = 00001100 = 12 = M

**✅ Результат:** "NM"

**🔍 КАК РАБОТАЕТ:**

* Преобразуем буквы в числа
* Выполняем побитный XOR
* Преобразуем обратно в буквы

**Пример 2: LFSR (Linear Feedback Shift Register)**

**🎯 СУТЬ:** Генерируем псевдослучайную последовательность с помощью сдвигового регистра.

**🔢 РАЗБОР LFSR с полиномом x⁴ + x + 1, начальное 1101:**

text

Полином x⁴ + x + 1 → обратные связи: бит3 и бит1

Начальное состояние: 1 1 0 1

Такт 1: новый бит = 1⊕0 = 1 → состояние: 1 1 1 0, выход: 1

Такт 2: новый бит = 1⊕1 = 0 → состояние: 0 1 1 1, выход: 0

Такт 3: новый бит = 0⊕1 = 1 → состояние: 1 0 1 1, выход: 1

Такт 4: новый бит = 1⊕1 = 0 → состояние: 0 1 0 1, выход: 1

**✅ Результат:** выходная последовательность: 1, 0, 1, 1, ...

**🔍 КАК РАБОТАЕТ:**

* Регистр хранит несколько битов
* Новый бит = XOR определенных битов (тапов)
* Все биты сдвигаются, новый бит занимает первую позицию

**Пример 3: RC4**

**🎯 СУТЬ:** Используем таблицу (S-блок), которую перемешиваем на основе ключа.

**🔢 РАЗБОР упрощенного RC4 с ключом "KEY":**

text

ИНИЦИАЛИЗАЦИЯ (KSA):

S = [0,1,2], ключ K = [10,4,24]

i=0: j = (0+0+10)%3 = 1 → swap S[0]↔S[1] → [1,0,2]

i=1: j = (1+0+4)%3 = 2 → swap S[1]↔S[2] → [1,2,0]

i=2: j = (2+0+24)%3 = 2 → без изменений

ГЕНЕРАЦИЯ (PRGA):

i=0,j=0 → i=(0+1)%3=1, j=(0+S[1])%3=2 → swap S[1]↔S[2] → [1,0,2]

t=(S[1]+S[2])%3=(0+2)%3=2 → выход S[2]=2

**✅ Результат:** первый байт гаммы = 2

**4.3 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ (20 заданий)**

**🎯 Уровень 1 (легкие):**

1. Зашифруйте "A" XOR с гаммой "1"
2. Сгенерируйте 5 бит LFSR x³+x+1, нач. 101
3. Примените гаммирование к "TEXT" с "KEYKEY"
4. Восстановите текст из "U" с гаммой "A"
5. Зашифруйте "HELLO" с гаммой "WORLD"
6. Сгенерируйте 4 бита LFSR x³+x²+1, нач. 1100
7. Примените гаммирование к "DATA" с "STREAM"
8. Восстановите текст из "X" с гаммой "B"
9. Зашифруйте "TEST" с гаммой "CIPHER"
10. Сгенерируйте 6 бит LFSR x⁴+x+1, нач. 1010

**🎯 Уровень 2 (средние):**  
11. Реализуйте RC4 для 3 байт с ключом "KEY"  
12. Сгенерируйте 10 бит LFSR x⁴+x³+1, нач. 1110  
13. Зашифруйте "MESSAGE" самосинхронизирующимся шифром  
14. Найдите период LFSR x⁴+x+1, нач. 1001  
15. Реализуйте гаммирование с нелинейным генератором  
16. Сгенерируйте 8 бит LFSR x⁵+x²+1, нач. 11011  
17. Зашифруйте "STREAMCIPHER" с динамической гаммой  
18. Найдите начальное состояние LFSR по выходу 10101  
19. Реализуйте комбинированный генератор LFSR  
20. Проанализируйте статистические свойства гаммы

**РАЗДЕЛ 5: ХЭШ-ФУНКЦИИ**

**5.1 ТЕОРЕТИЧЕСКАЯ СУТЬ МЕТОДА**

**📚 СУТЬ МЕТОДА:**  
Преобразуем любой текст в строку фиксированной длины ("отпечаток пальца"). Даже маленькое изменение в тексте должно полностью менять хэш.

**🔑 Ключевые принципы:**

* **Фиксированная длина** - любой текст → хэш одинаковой длины
* **Детерминированность** - одинаковый текст → одинаковый хэш
* **Необратимость** - по хэшу нельзя восстановить текст
* **Устойчивость к коллизиям** - сложно найти два текста с одинаковым хэшем

**5.2 ПОДРОБНЫЕ ПРИМЕРЫ С ПОШАГОВЫМ ОБЪЯСНЕНИЕМ**

**Пример 1: Конструкция Меркла-Дамгарда**

**🎯 СУТЬ:** Разбиваем длинный текст на блоки и последовательно их "сжимаем".

**🔢 РАЗБОР "DATA" с f(x,y)=(x+y) mod 100, IV=0:**

text

Блоки: "DA" = 0300 = 300, "TA" = 1900 = 1900

h₁ = f(IV, "DA") = (0 + 300) mod 100 = 0

h₂ = f(h₁, "TA") = (0 + 1900) mod 100 = 0

**✅ Результат:** 0

**🔍 КАК РАБОТАЕТ:**

* Начинаем с начального значения (IV)
* Для каждого блока: новое\_состояние = f(старое\_состояние, блок)
* Последнее состояние - это хэш

**Пример 2: Упрощенный MD5**

**🎯 СУТЬ:** Сложные преобразования в несколько раундов с разными функциями.

**🔢 РАЗБОР для "A":**

text

1. ДОПОЛНЕНИЕ:

"A" = 65 = 01000001

Добавляем 1: 01000001 1

Добавляем 439 нулей

Добавляем 64 бита длины (8)

2. ФУНКЦИИ:

F(X,Y,Z) = (X AND Y) OR (NOT X AND Z)

G(X,Y,Z) = (X AND Z) OR (Y AND NOT Z)

H(X,Y,Z) = X XOR Y XOR Z

I(X,Y,Z) = Y XOR (X OR NOT Z)

3. 4 РАУНДА преобразований

**✅ Результат:** 128-битный хэш

**Пример 3: HMAC**

**🎯 СУТЬ:** Добавляем ключ к хэш-функции для аутентификации.

**🔢 РАЗБОР с ключом "SECRET" для "HELLO":**

text

1. ВНУТРЕННИЙ ХЭШ:

inner = H((K ⊕ ipad) || M)

ipad = 0x36 (повторяющийся байт)

2. ВНЕШНИЙ ХЭШ:

HMAC = H((K ⊕ opad) || inner)

opad = 0x5C (другой повторяющийся байт)

**✅ Результат:** ключевой хэш

**5.3 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ (20 заданий)**

**🎯 Уровень 1 (легкие):**

1. Вычислите XOR-хэш "ABC" (A⊕B⊕C)
2. Создайте контрольную сумму "TEXT" (сумма кодов mod 256)
3. Вычислите CRC8 для байта 0x12
4. Найдите коллизию для хэша "сумма кодов mod 10"
5. Вычислите XOR-хэш "HELLO"
6. Создайте контрольную сумму "MESSAGE"
7. Вычислите CRC16 для "DATA"
8. Найдите коллизию для хэша "произведение mod 15"
9. Вычислите сумму ASCII "TEST"
10. Создайте упрощенный хэш "BLOCK"

**🎯 Уровень 2 (средние):**  
11. Реализуйте хэш на основе SHA-1 (упрощенный)  
12. Вычислите Merkle-Damgard конструкцию для "DATA"  
13. Создайте HMAC с ключом "SECRET" для "HELLO"  
14. Найдите прообраз для хэша (атака грубой силой)  
15. Реализуйте хэш на основе MD4 (упрощенный)  
16. Вычислите каскадный хэш для "LONGTEXT"  
17. Создайте HMAC с ключом "KEY" для "MESSAGE"  
18. Найдите второе предображение для хэша  
19. Реализуйте хэш с солью для "PASSWORD"  
20. Вычислите инкрементальный хэш