ПОТОКОВОЕ ШИФРОВАНИЕ

КОМАНДА ГРУППЫ АА-22-08:
ГЕРАСИМОВА АНАСТАСИЯ
СКОРЖЕВСКАЯ МИЛЕНА
ТИТОВА АНАСТАСИЯ
ЕВГРАФОВ БОРИС

МОСКВА, 2023

ЧТО ЭТО ТАКОЕ?

ШИФРОВАНИЕ — ОБРАТИМОЕ ПРЕОБРАЗОВАНИЕ ИНФОРМАЦИИ В ЦЕЛЯХ СОХРАНЕНИЯ КОНФИДЕНЦИАЛЬНОСТИ ПЕРЕДАВАЕМОЙ ИНФОРМАЦИИ.

ПОТОКОВЫЙ ШИФР — СИММЕТРИЧНЫЙ ТИП ШИФРА, ГДЕ КАЖДЫЙ ЭЛЕМЕНТ ОТКРЫТОГО ТЕКСТА ПЕРЕВОДИТСЯ В ЗАШИФРОВАННЫЙ ВИД В ЗАВИСИМОСТИ ОТ ПРИМЕНЯЕМОГО КЛЮЧА И ПОЗИЦИИ КОНКРЕТНОГО ЭЛЕМЕНТА В ТЕКСТОВОМ ПОТОКЕ.

КОГДА ПРИМЕНЯЕТСЯ?

КОНФИДЕНЦИАЛЬНОСТЬ

ШИФРОВАНИЕ ИСПОЛЬЗУЕТСЯ ДЛЯ СКРЫТИЯ ИНФОРМАЦИИ ОТ НЕАВТОРИЗОВАННЫХ ПОЛЬЗОВАТЕЛЕЙ ПРИ ПЕРЕДАЧЕ ИЛИ ХРАНЕНИИ.

ЦЕЛОСТНОСТЬ

ШИФРОВАНИЕ ИСПОЛЬЗУЕТСЯ ДЛЯ ПРЕДОТВРАЩЕНИЯ ИЗМЕНЕНИЯ ИНФОРМАЦИИ ПРИ ПЕРЕДАЧЕ ИЛИ ПРИ ХРАНЕНИИ.

ИДЕНТИФИЦИРУЕМОСТЬ

ШИФРОВАНИЕ ИСПОЛЬЗУЕТСЯ ДЛЯ АУТЕНТИФИКАЦИИ ИСТОЧНИКА ИНФОРМАЦИИ.

ПОСТАНОВКА ЗАДАЧИ

- РЕАЛИЗОВАТЬ **АЛГОРИТМЫ ПОТОКОВОГО ШИФРОВАНИЯ**, ПОЛУЧАЮЩИЕ НА ВХОД СТРОКУ И ВЫВОДЯЩИЕ ЗАШИФРОВАННУЮ СТРОКУ.
- РЕАЛИЗОВАТЬ **ДЕШИФРАТОРЫ**, ПРИНЦИП РАБОТЫ КОТОРЫХ ОСНОВАН НА РЕАЛИЗУЕМОМ АЛГОРИТМЕ, КОТОРЫЕ ПРЕОБРАЗУЮТ ЗАШИФРОВАННУЮ СТРОКУ В ИСХОДНУЮ.

АЛГОРИТМЫ

RC4

Salsa20

A5

ИСТОРИЯ РАЗВИТИЯ RC4

- RC4 (Rivest Cipher 4) был разработан Роном Ривестом в 1987 году в США;
- В течение семи лет шифр являлся коммерческой тайной, и точное описание алгоритма предоставлялось только после подписания соглашения о неразглашении;
- В сентябре 1994 года его описание было анонимно отправлено в список рассылки и вскоре опубликовано в группе новостей «Usenet»;
- RC4 быстро стал популярным алгоритмом благодаря своей простоте и высокой скорости работы. Он был широко использован в различных протоколах и приложениях, таких как SSL/TLS, WEP (беспроводная защита), Bluetooth и т.д.

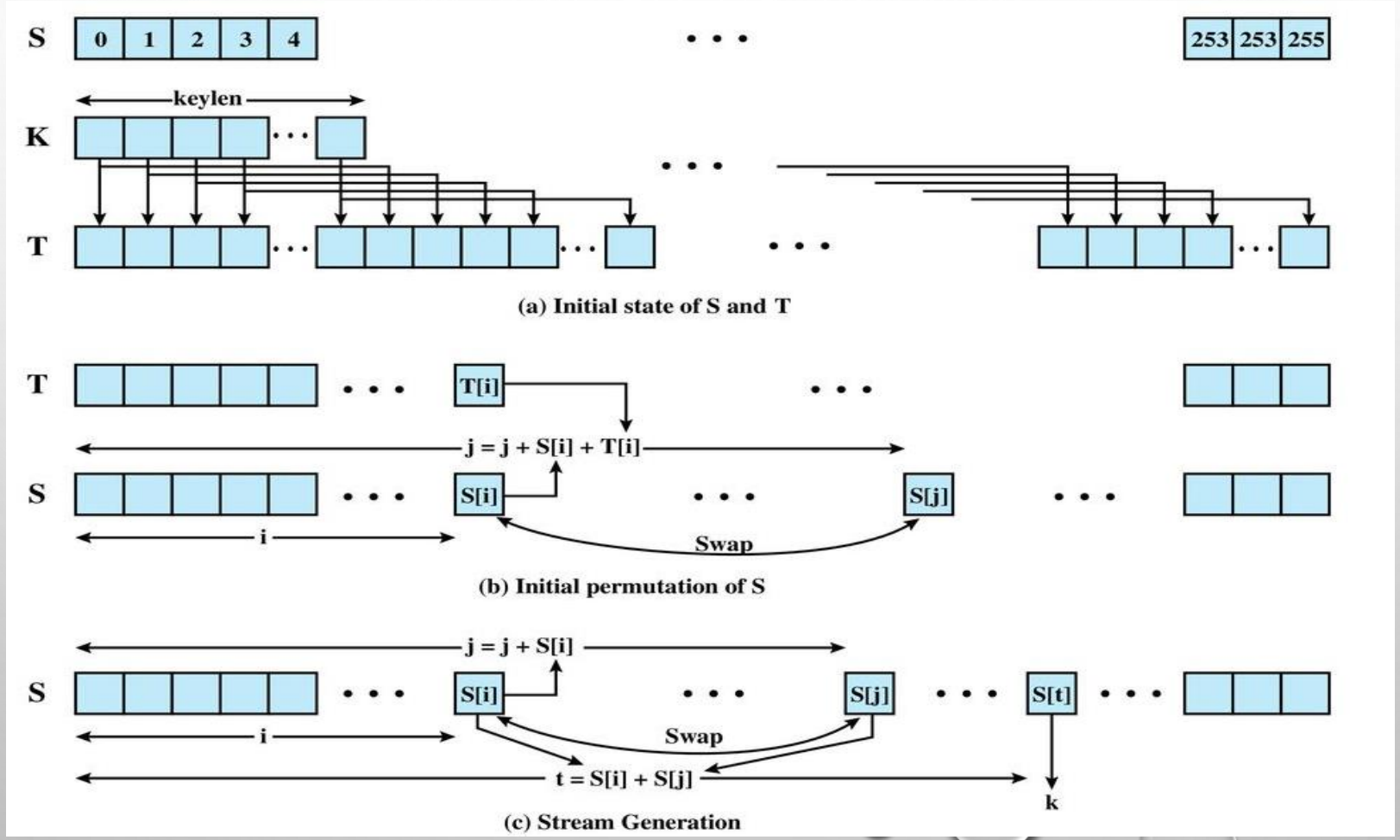
СУТЬ АЛГОРИТМА RC4

1. Генерация ключевой последовательности: алгоритм RC4 использует ключ фиксированной длины (обычно от 40 до 2048 бит). Сначала ключ преобразуется в начальное состояние алгоритма, которое представляет собой перестановку всех возможных значений байта;
2. Инициализация вектора состояния: вектор состояния является внутренним состоянием алгоритма и используется для генерации псевдослучайной ключевой последовательности. Вектор состояния инициализируется с использованием ключа путем применения операций перестановки к начальному состоянию;

СУТЬ АЛГОРИТМА RC4

3. Генерация псевдослучайной последовательности: после инициализации вектора состояния, алгоритм генерирует псевдослучайную ключевую последовательность, которая используется для шифрования данных. Генерация осуществляется путем применения операций перестановки и обмена значений вектора состояния;
4. Шифрование данных: полученная псевдослучайная ключевая последовательность комбинируется с исходными данными побитовым исключающим ИЛИ (XOR). Это преобразование делает данные непредсказуемыми и обеспечивает их конфиденциальность.

ПРИНЦИП РАБОТЫ RC4

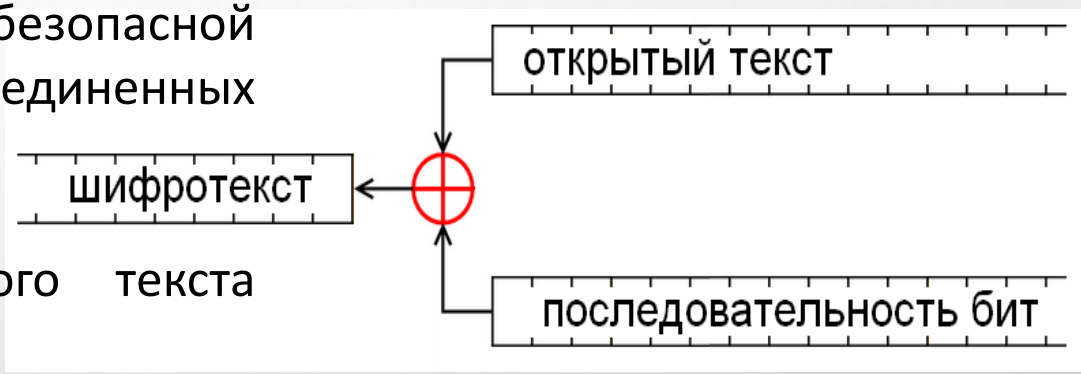


СВОЙСТВА АЛГОРИТМА RC4

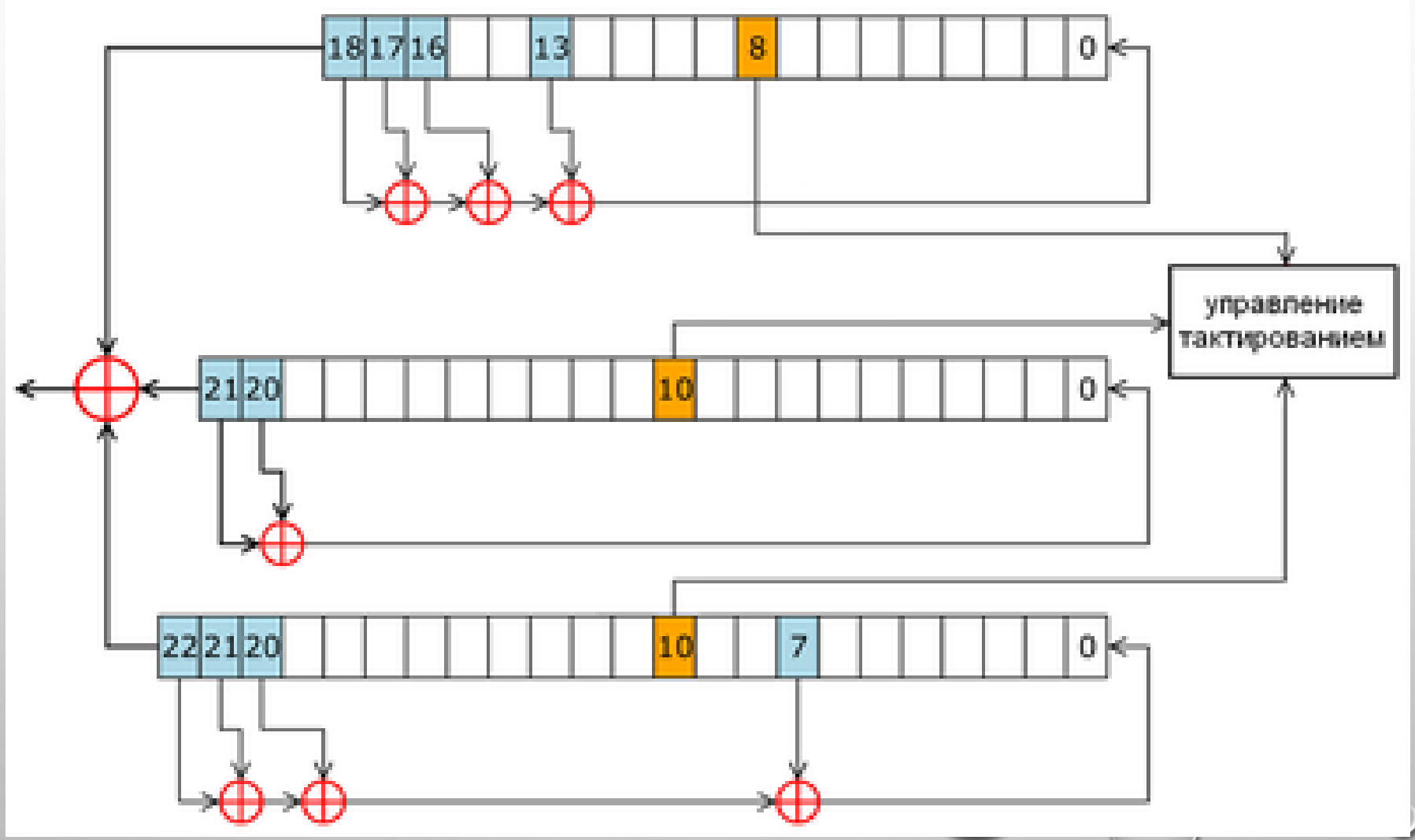
Свойство	Описание
Псевдослучайность	Алгоритм RC4 генерирует псевдослучайную последовательность байтов, которая используется для шифрования данных.
Простота реализации	RC4 является относительно простым алгоритмом, что делает его легко реализуемым на различных платформах и устройствах.
Быстрота	RC4 работает очень быстро и эффективно, что делает его привлекательным для использования в реальном времени.
Ключевой размер	RC4 поддерживает ключи различной длины, от 1 до 256 бит.
Уязвимости	RC4 имеет некоторые известные уязвимости, такие как утечка информации о ключе и возможность атаки на слабые ключи.

A5/1

- Был разработан для защиты передачи данных и голоса в сотовых сетях GSM в конце 80-х годов;
- A5/1 является первоначальной и более безопасной версией, используемой в Европе и Соединенных Штатах;
- В алгоритме каждому символу открытого текста соответствует символ шифротекста;
- Текст не делится на блоки и не изменяется в размере;
- Сложение по модулю 2 (XOR) и сдвиг регистра являются двумя основными операциями.



ПРИНЦИП РАБОТЫ A5/1



СТРУКТУРА АЛГОРИТМА A5/1

Структура A5 выглядит следующим образом:

- три регистра (R1, R2, R3) имеют длины 19, 22 и 23 бита,
- многочлены обратных связей:
  $X^{19} + X^{18} + X^{17} + X^{14} + 1$ для R1,
  $X^{22} + X^{21} + 1$ для R2 и
  $X^{23} + X^{22} + X^{21} + X^{8} + 1$ для R3,
- управление тактированием осуществляется специальным механизмом:
  - в каждом регистре есть биты синхронизации: 8 (R1), 10 (R2), 10 (R3),
  - вычисляется функция F = x&y|x&z|y&z, где & — булево AND, | - булево OR, а x, y и z — биты синхронизации R1, R2 и R3 соответственно,
  - сдвигаются только те регистры, у которых бит синхронизации равен F,
  - фактически, сдвигаются регистры, синхробит которых принадлежит большинству,
- выходной бит системы — результат операции XOR над выходными битами регистров.

ПРЕИМУЩЕСТВА И НЕДОСТАТКИ A5/1

Преимущества A5/1:

1.Простота реализации: A5/1 относительно прост в реализации, особенно в аппаратном обеспечении.

2.Низкие вычислительные требования: Алгоритм не требует большой вычислительной мощности.

3.Подходит для потокового шифрования: Данные передаются и шифруются по мере их получения.

Недостатки A5/1:

1.Уязвимость к атакам: A5/1 подвержен атакам с использованием метода временных корреляций и с перебором.

2. Ограниченная длина ключа: 64-битный ключ.

3. Проблемы с безопасностью.

4. Отсутствие гибкости: A5/1 был разработан специально для GSM и не обладает гибкостью в плане применения в других сферах или адаптации под новые требования безопасности.

Salsa20

Salsa20 — система поточного шифрования, разработанная Даниэлем Бернштейном в 2005 году.

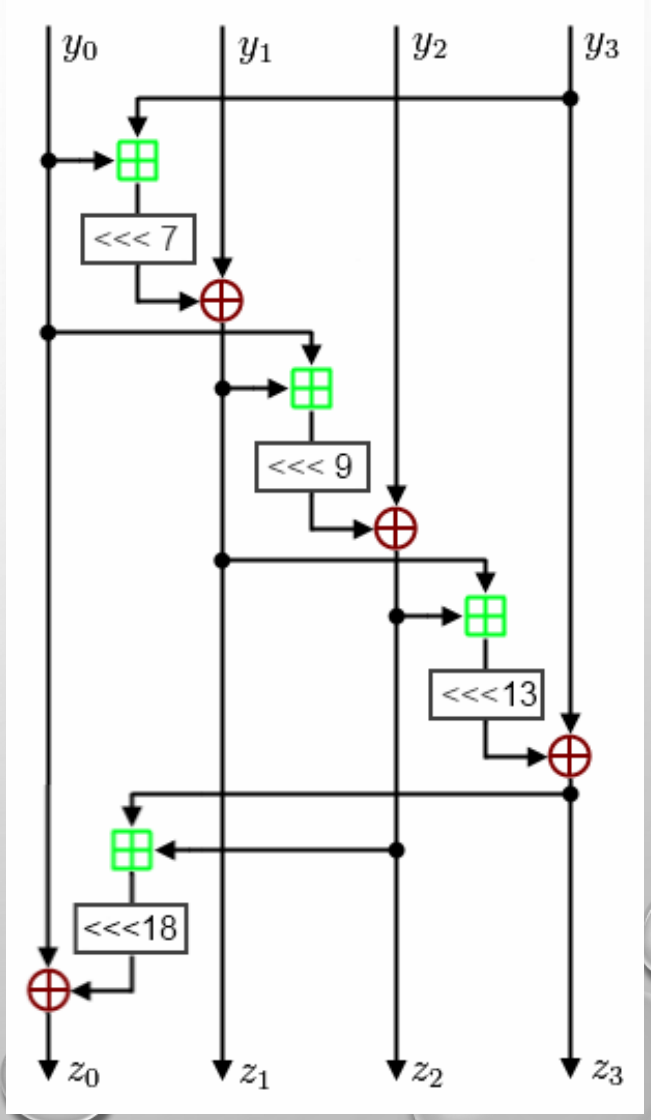
Алгоритм был представлен на конкурсе eSTREAM, целью которого было создание европейских стандартов для поточных систем шифрования, и стал победителем конкурса в первом профиле (поточные шифры для программного применения с большой пропускной способностью).

Шифр Salsa20 использует следующие операции:

1. сложение 32-битных чисел;
2. побитовое сложение по модулю 2 (xor);
3. сдвиги битов.

Суть алгоритма

1. Алгоритм принимает ключ шириной от 128 до 256 бит и некоторый инициализационный вектор (IV), чтобы генерировать псевдослучайный поток бит.
2. Происходит объединение 4-битных чисел ключа и вектора с помощью операции XOR.
3. Перемещение битов внутри каждого из 16-битных слов ключа и вектора.
4. Полученный поток битов затем применяется к открытому тексту или шифротексту посредством операции XOR для получения зашифрованного или расшифрованного сообщения.



Преимущества и недостатки

Преимущества Salsa20:

1. **Эффективность:** Salsa20 является очень быстрым алгоритмом шифрования, что делает его привлекательным для применения в реальном времени или в вычислительно интенсивных приложениях.
2. **Производительность:** Salsa20 разработан с учетом высокой производительности и эффективности. Алгоритм легко параллелизуется, что позволяет использовать его для шифрования больших объемов данных.

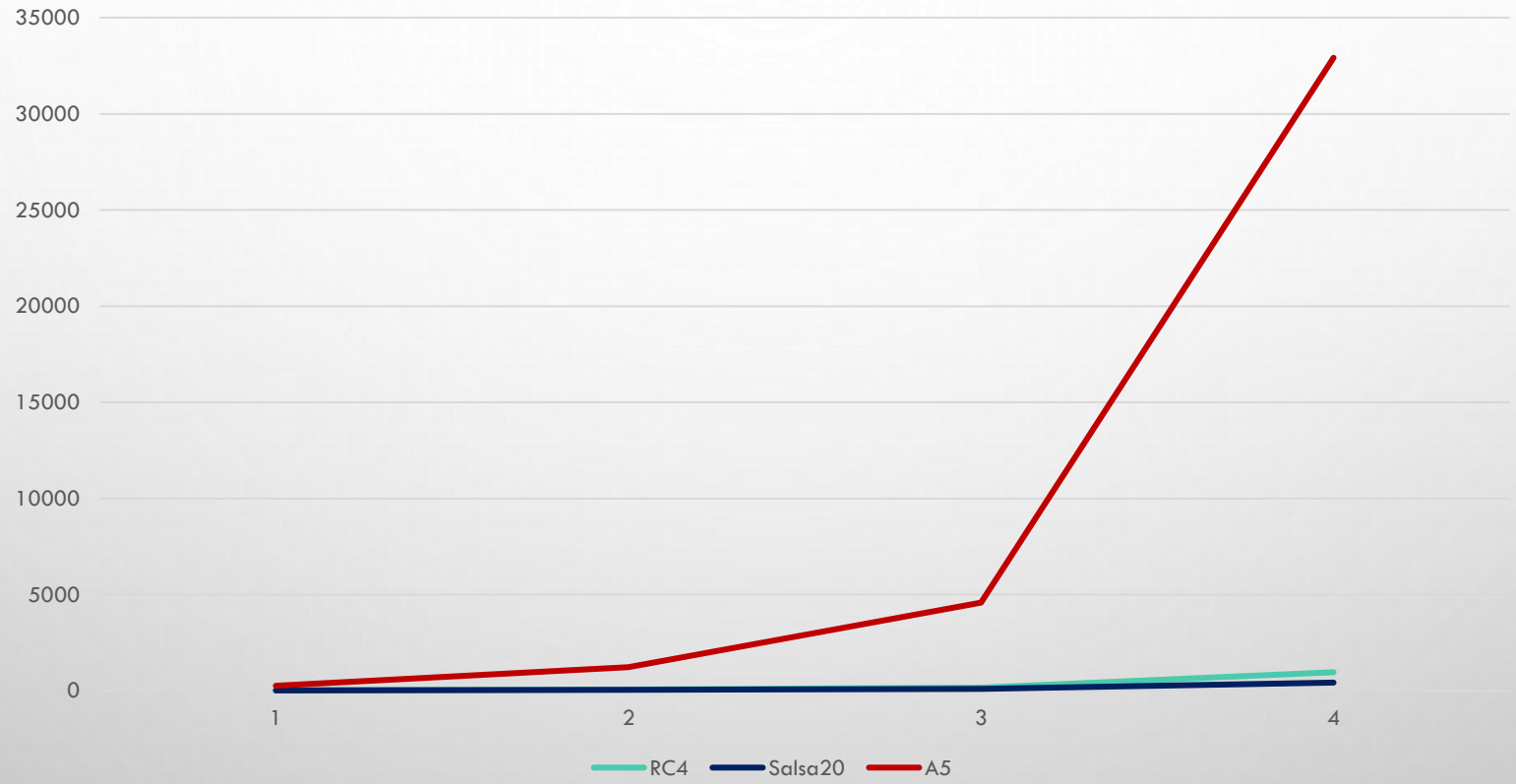
Недостатки Salsa20:

1. **Ограничение длины ключа:** Salsa20 поддерживает только ключи фиксированной длины (128 или 256 бит), что ограничивает использование других длин ключей.
2. **Относительная новизна:** Алгоритм Salsa20 появился в 2005 году, что делает его менее изученным и освоенным, чем некоторые более старые алгоритмы.

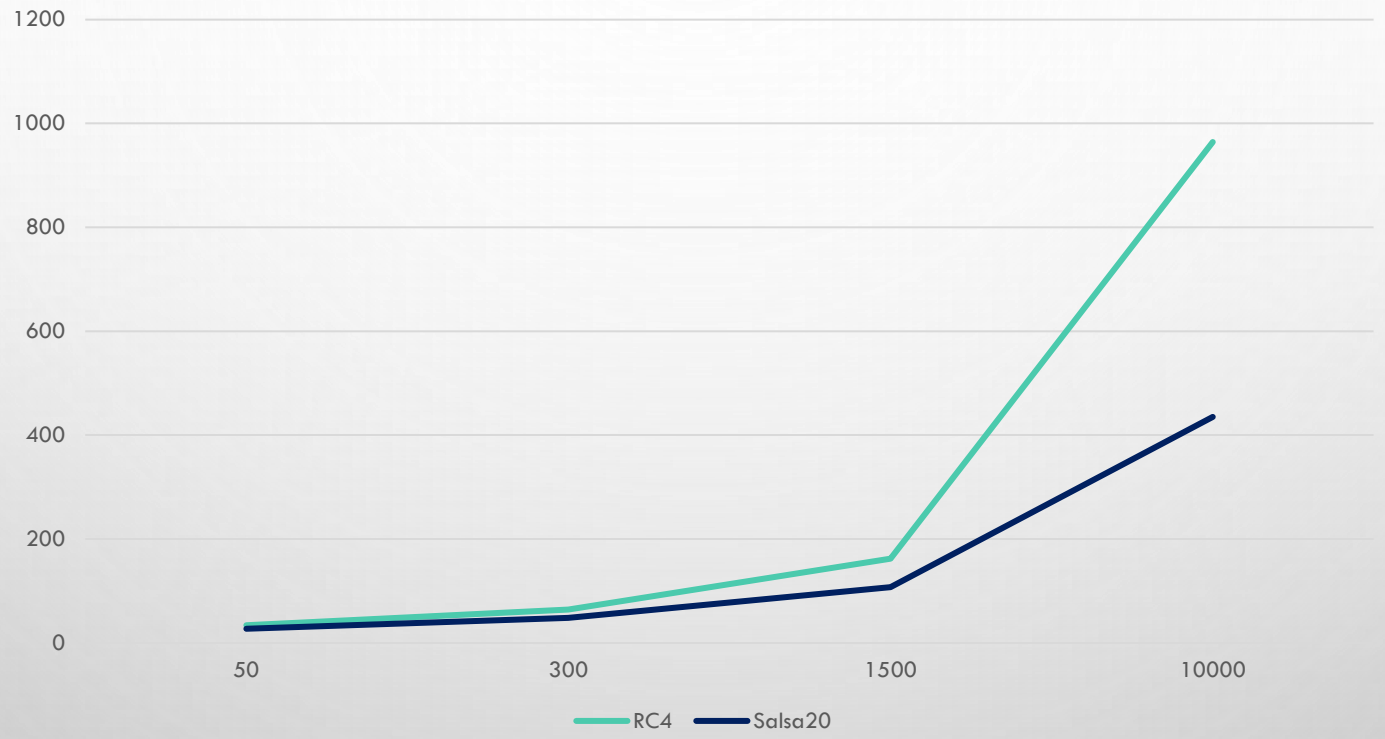
РЕЗУЛЬТАТЫ

Шифрование			
Время в микросекундах (мкс)			
Объём текста (символы)	**RC4**	**Salsa20**	**A5**
50	34	27	262
300	64	48	1226
1500	162	107	4591
10000	964	435	32913

Шифрование


35000
30000
25000
20000
15000
10000
5000
0
1
2
3
4
RC4
Salsa20
A5

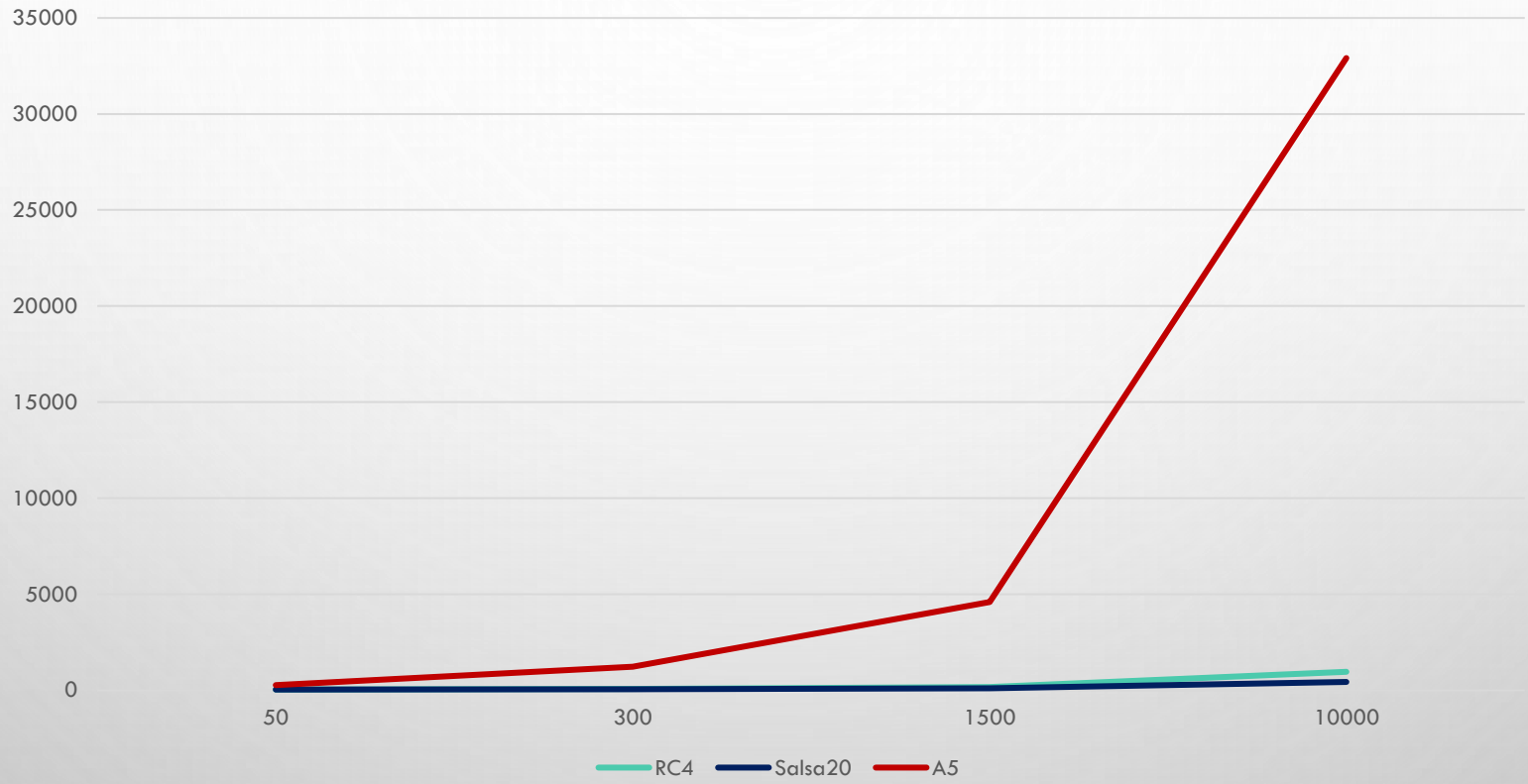
Шифрование



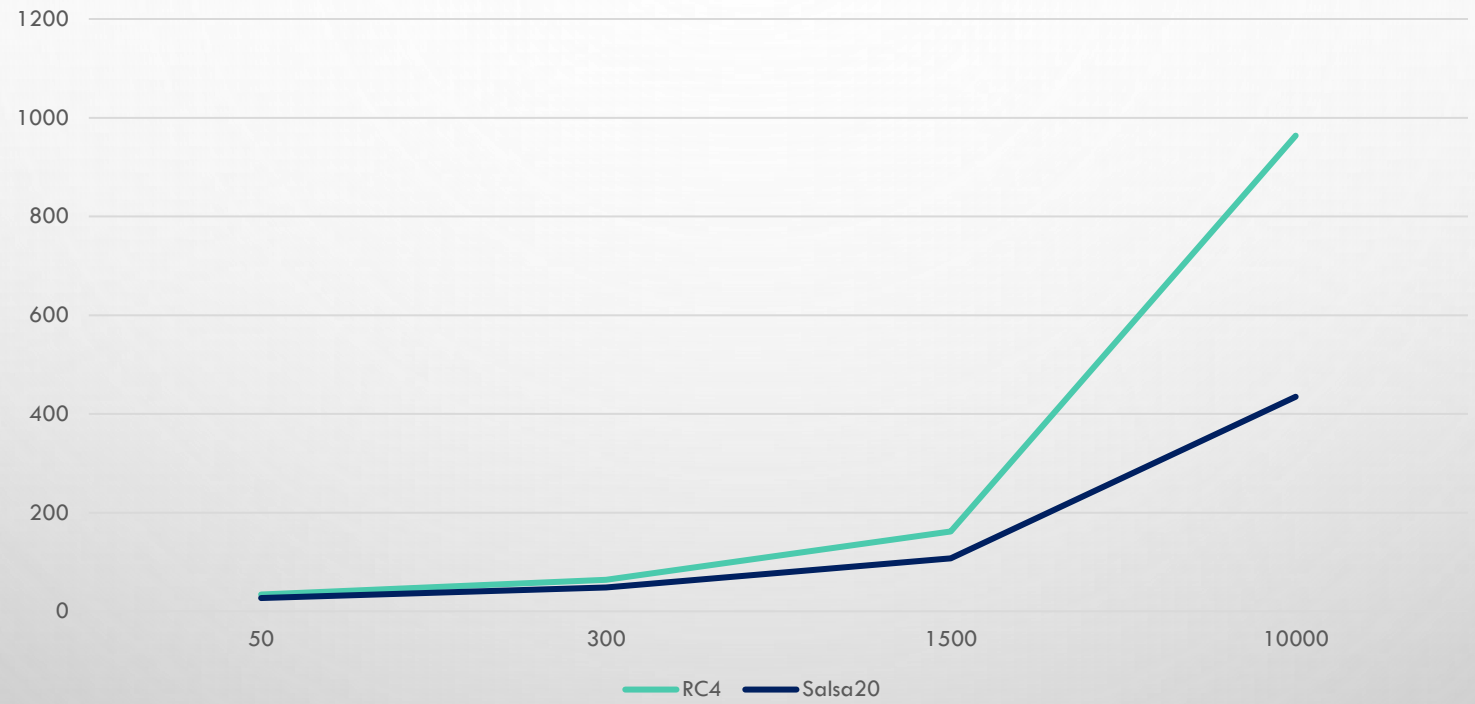
РЕЗУЛЬТАТЫ

Дешифрование			
Время в микросекундах (мкс)			
Объём текста (символы)	**RC4**	**Salsa20**	**A5**
50	30	8	124
300	56	13	539
1500	157	51	2713
10000	959	176	15985

Дешифрование



Дешифрование



ССЫЛКА НА РЕПОЗИТОРИЙ