

# Криптография

## Лекция 1. Симметричные шифры.

*Дмитрий Яхонтов*

*“Кочерга”, 2019*

# Симметричное шифрование

(оно же шифрование с закрытым ключом)

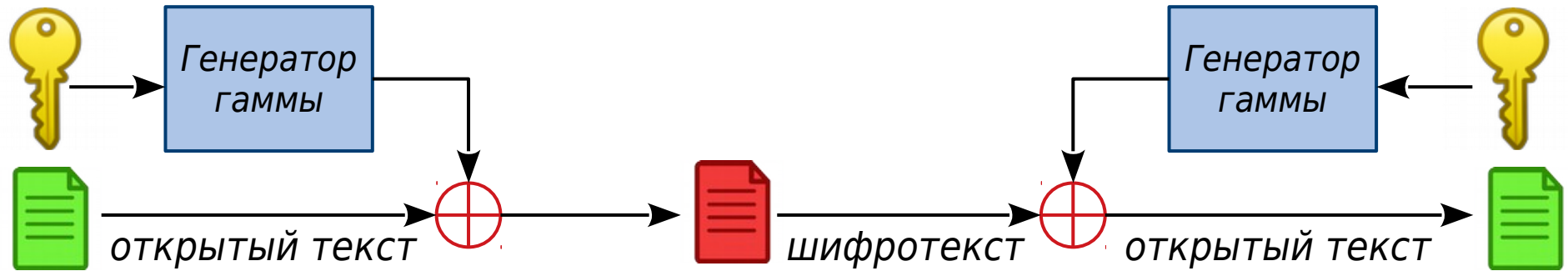


Для шифрования и дешифровки служит один и тот же ключ.

Ключ необходимо передать по защищённому каналу.

Обе стороны должны сохранять ключ в секрете.

# Поточные шифры

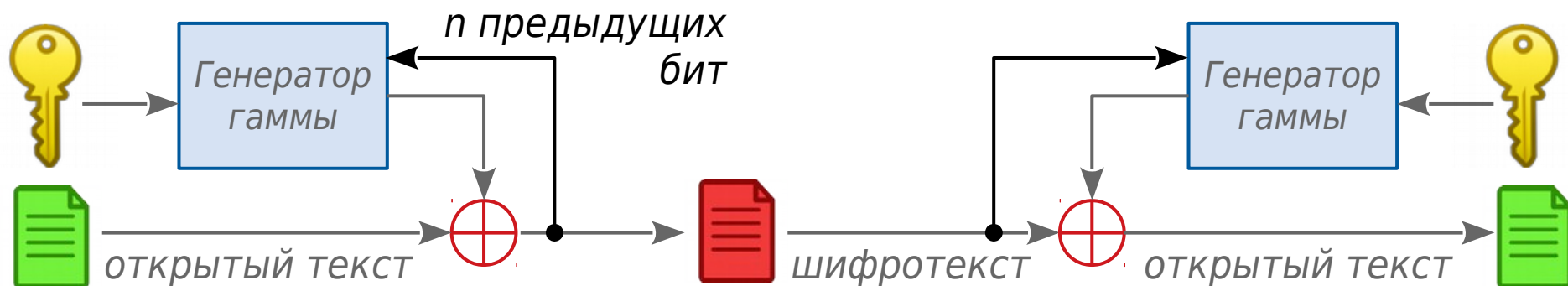


Биты открытого текста преобразуются в биты шифротекста путём наложения псевдослучайной последовательности (гаммы).

Гамма генерируется на основе ключа.

Шифрующий и дешифрующий генераторы должны работать синхронно.


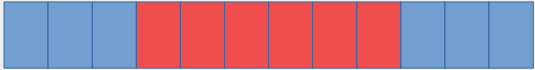
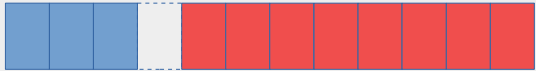
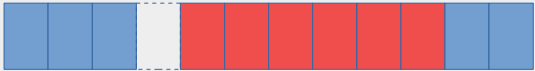
# Асинхронный поточный шифр (он же самосинхронизирующийся)



Внутреннее состояние генератора гаммы — функция от предыдущих  $n$  бит шифротекста.

Дешифрующий генератор синхронизируется с шифрующим автоматически после приёма  $n$  бит.

# Влияние ошибок при передаче

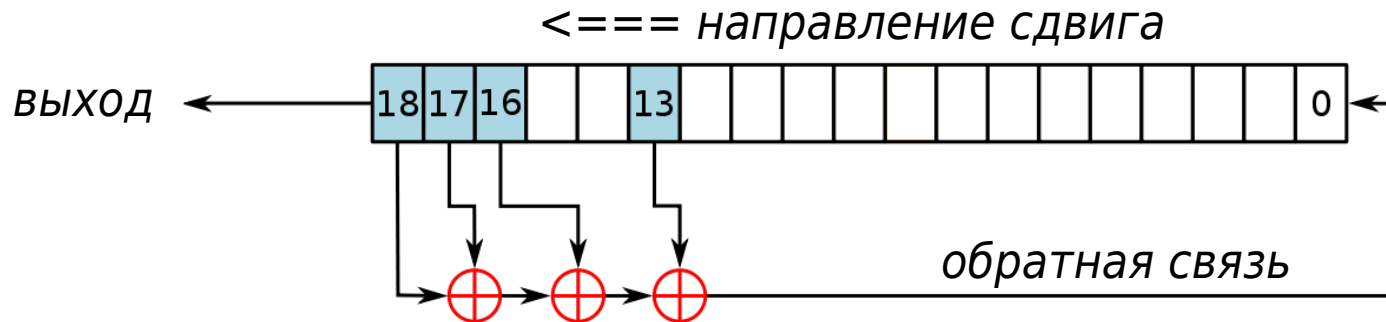
1 бит шифротекста...	синхронный шифр	асинхронный шифр
	после дешифровки будут ошибочными:	
...изменён	1 бит 	$n$ бит 
...потерян	весь дальнейший поток 	$n$ бит 

# Генераторы гаммы

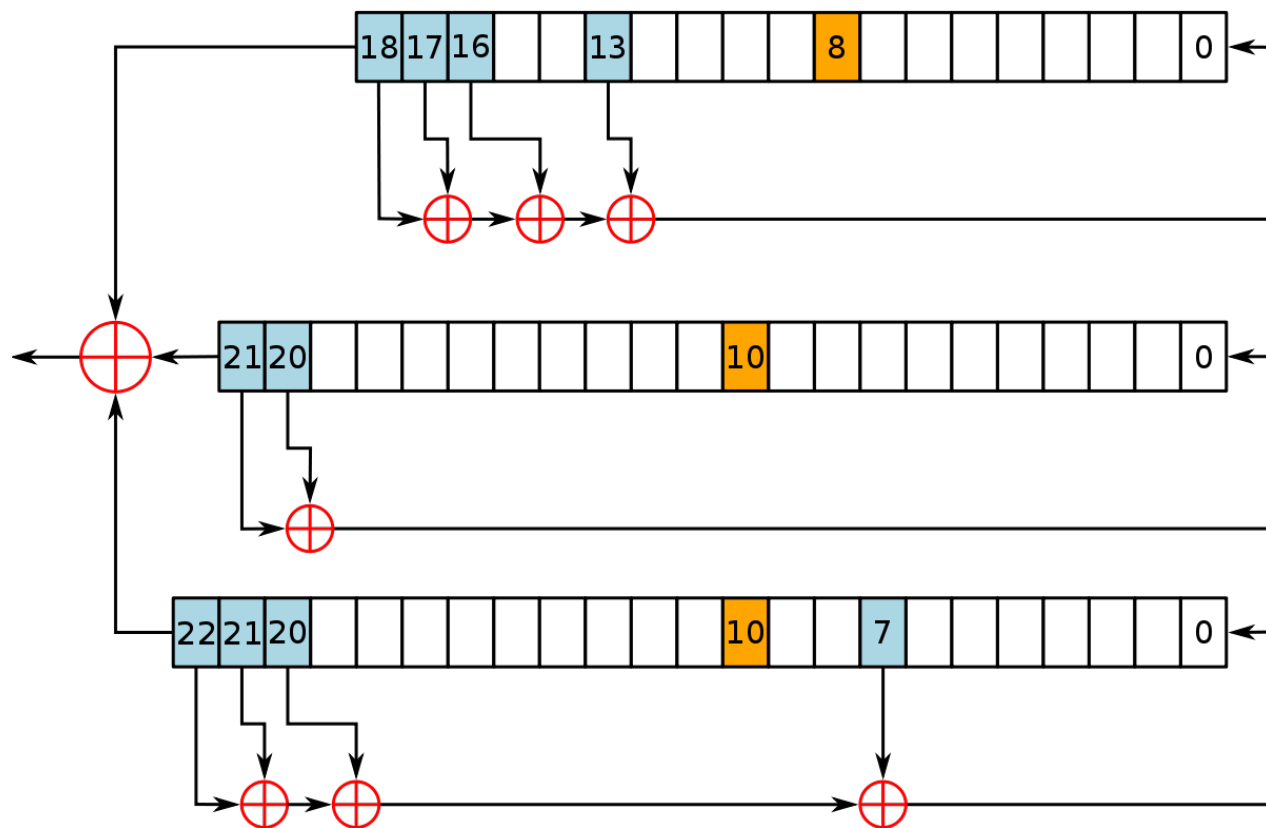
Регистр сдвига с линейной обратной связью  
(*Linear Feedback Shift Register, LFSR*)

Регистр длины N генерирует последовательность с периодом  $2^N - 1$ .

Начальное заполнение регистра определяется ключом шифрования.



# Пример: генератор гаммы шифра A5/1 (GSM)



Три сдвиговых регистра  
с длинами 19, 22 и 23.

Каждый регистр  
(биты 8, 10 и 10)  
управляет тактированием  
двух остальных регистров

Выход генератора —  
исключающее ИЛИ от  
выходов трех регистров

# Криптоанализ поточных шифров

- **Полный перебор** (*Bruteforce*)

Перебор всех возможных ключей.

Противодействие: увеличение длины ключа.

- **Статистическая атака**

Поиск статистических особенностей гаммы, которые позволят предсказать значение следующего бита по нескольким предыдущим.

Противодействие: использование шумоподобных генераторов гаммы.

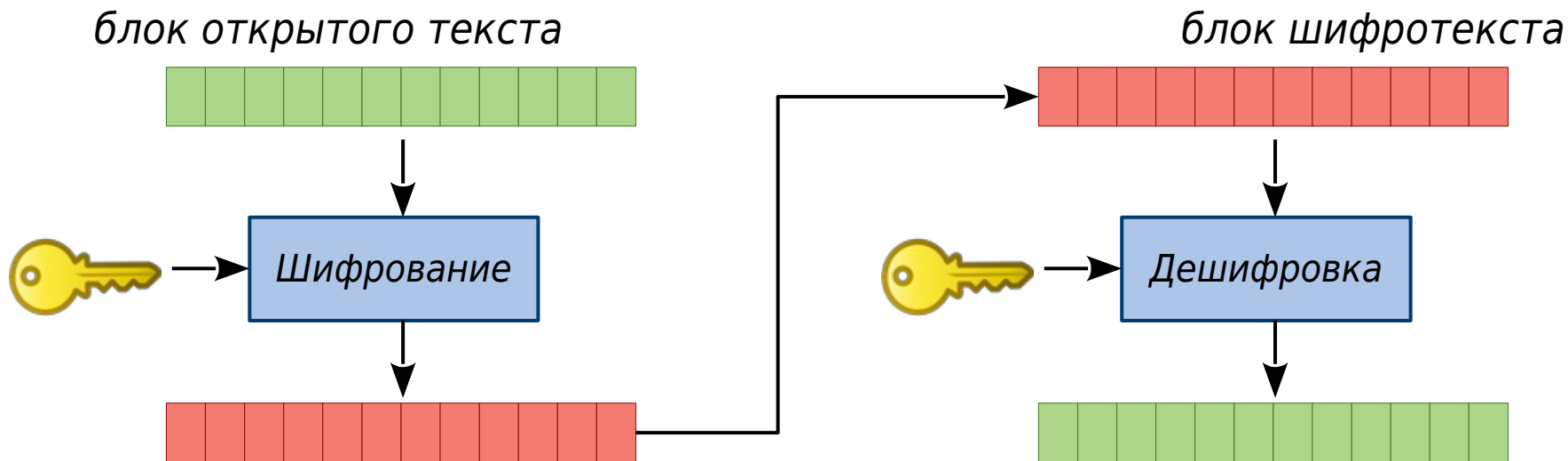
- **Корреляционная атака**

Поиск корреляции (частичного соответствия) между внутренним состоянием генератора гаммы и его выходом. Восстановив начальное внутреннее состояние генератора, получаем ключ.

Противодействие: использование нелинейной функции на выходе генератора гаммы.



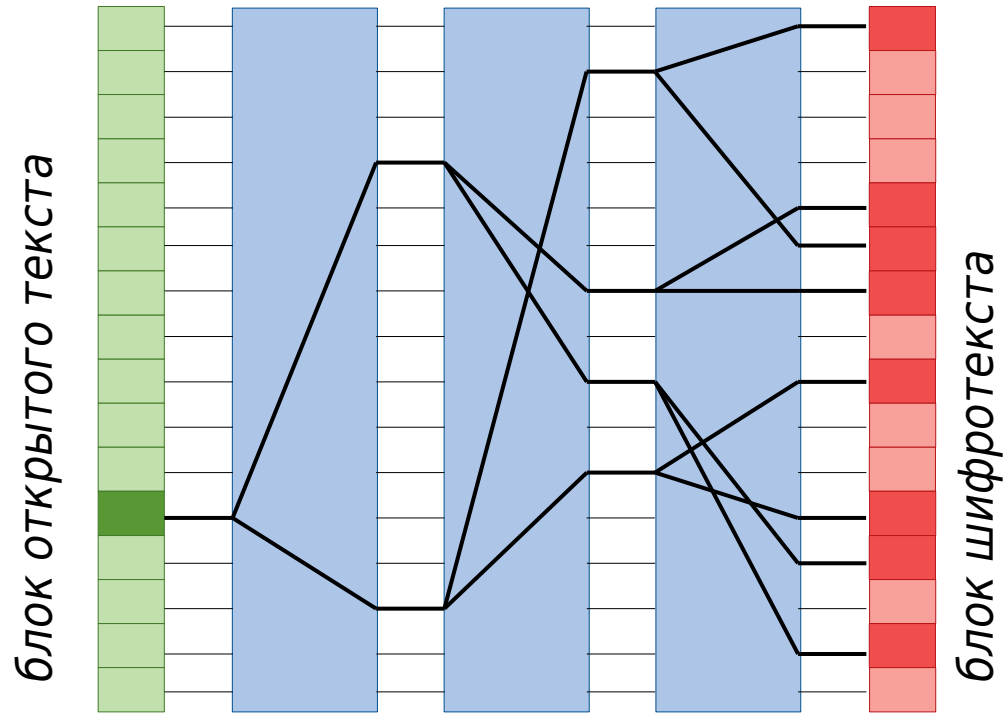
# Блочные шифры



Операции шифрования и дешифровки выполняются над блоками фиксированного размера.

Большинство блочных шифров являются итеративными: процедура шифрования состоит из нескольких раундов.

# Лавинный эффект



Text = "AAAAAAAAAAAAAAAA"  
0a72b5569fc8abaa014eae3ddbcbcd94

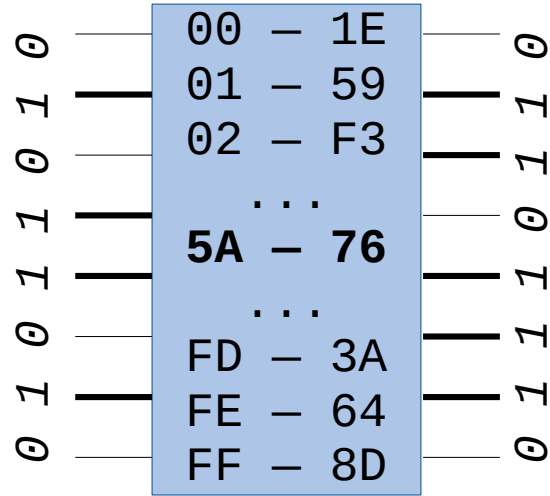
Text = "AAAAAAAAAAAAA**Q**AAA"  
b0338cd68e5c428d296fd3d3e7786fa2

Key = "CCCCCCCCCCCCCCCC"  
0a72b5569fc8abaa014eae3ddbcbcd94

Key = "CCC**G**CCCCCCCCCCCC"  
1481d79fcc645f5c427a82e6021796c1

Изменение одного бита входного блока ведёт к изменению в среднем половины бит выходного блока.

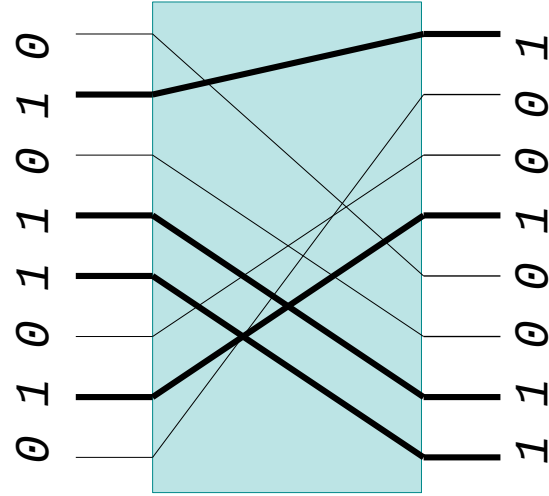
# SP-сеть



S-блок

(substitution)

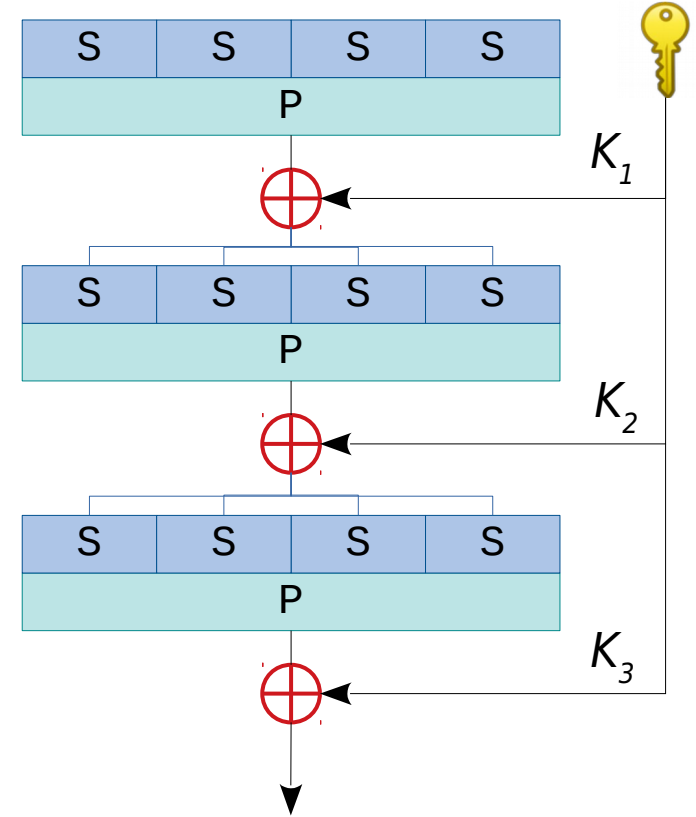
таблица замены



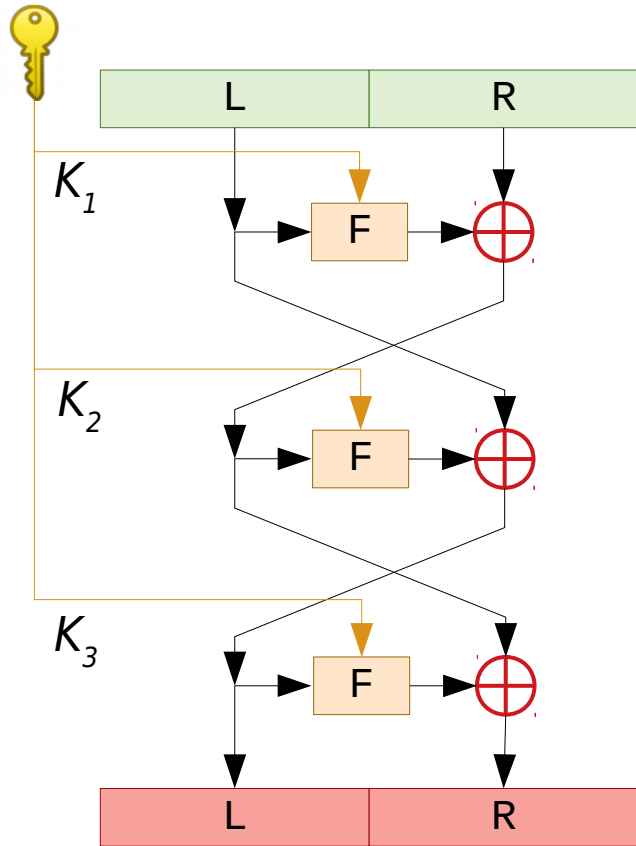
P-блок

(permutation)

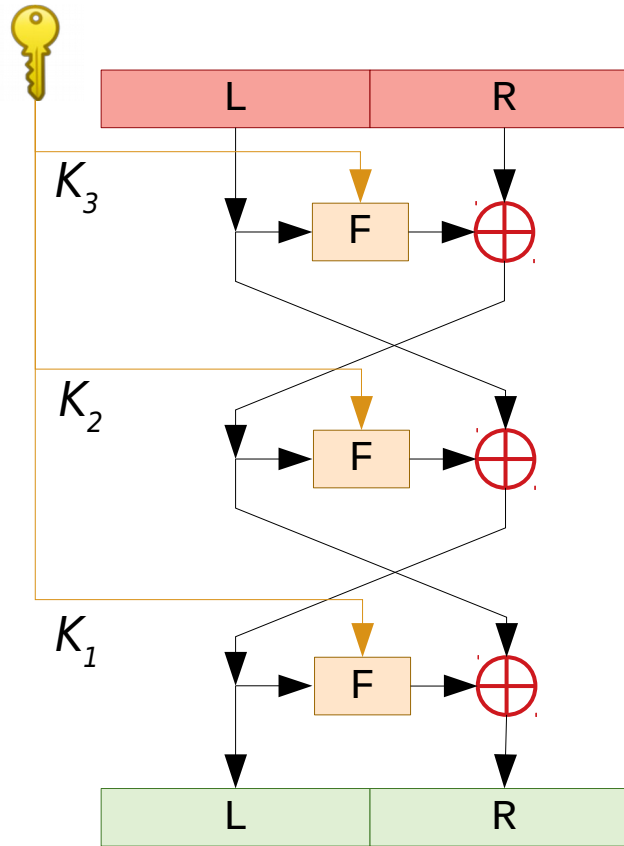
перестановка бит



# Сеть Фейстеля



Шифрование



Дешифровка

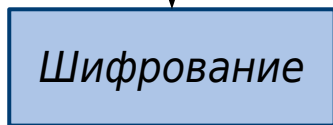
Варианты функции F:

- S-блок
- P-блок
- Циклический сдвиг
- Сложение по модулю  $n$
- Умножение по модулю  $n$
- Комбинация всего вышеперечисленного

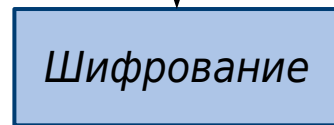
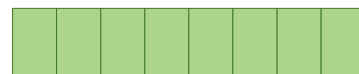
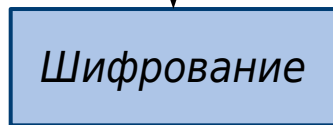
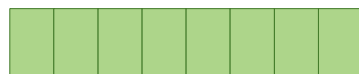
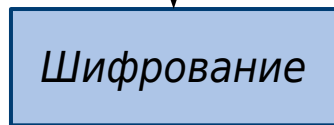
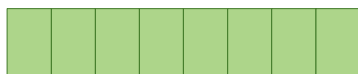
название шифра	год	тип шифра	число раундов	размер блока	размер ключа
DES	1977	Фейст.	16	64	56
Triple DES	1978	Фейст.	48	64	168
Магма (ГОСТ 28147-89)	1989	Фейст.	16 / 32	64	256
Blowfish	1993	Фейст.	16	64	32-448
AES (Rijndael)	1998	SP	10-14	128	128 / 192 / 256
Serpent	1998	SP	32	128	128 / 192 / 256
Twofish	1998	Фейст.	16	128	128 / 192 / 256
Threefish	2008	SP	72 / 80	256 / 512 / 1024	256 / 512 / 1024
Кузнечик (ГОСТ Р 34.12-2015)	2015	SP	10	128	256

# Режим простой замены (ECB — Electronic Code Book)

открытый текст

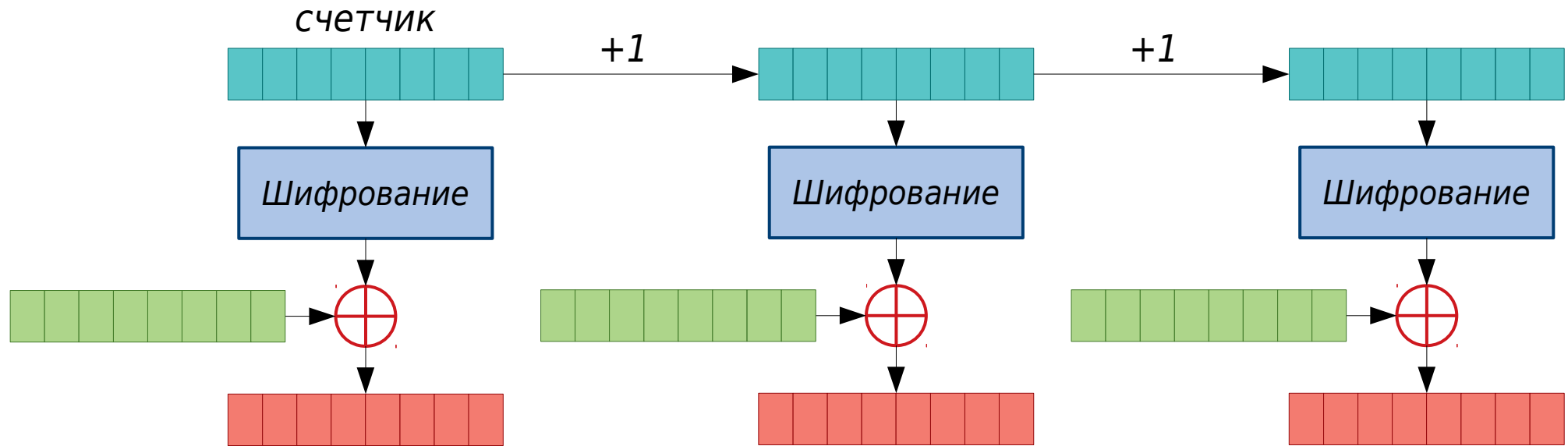


шифротекст



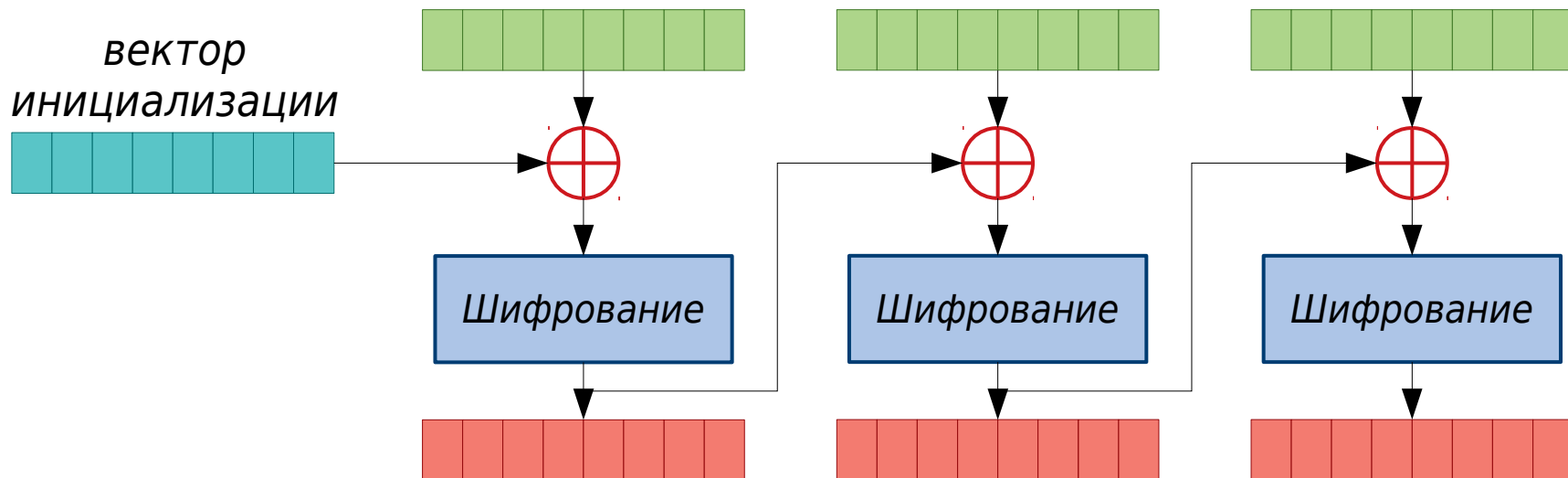
Все блоки шифруются независимо. Одинаковые блоки открытого текста дают одинаковые блоки шифротекста.

# Режим счетчика (CTR – Counter)



Значение счетчика должно быть уникальным для каждого блока.

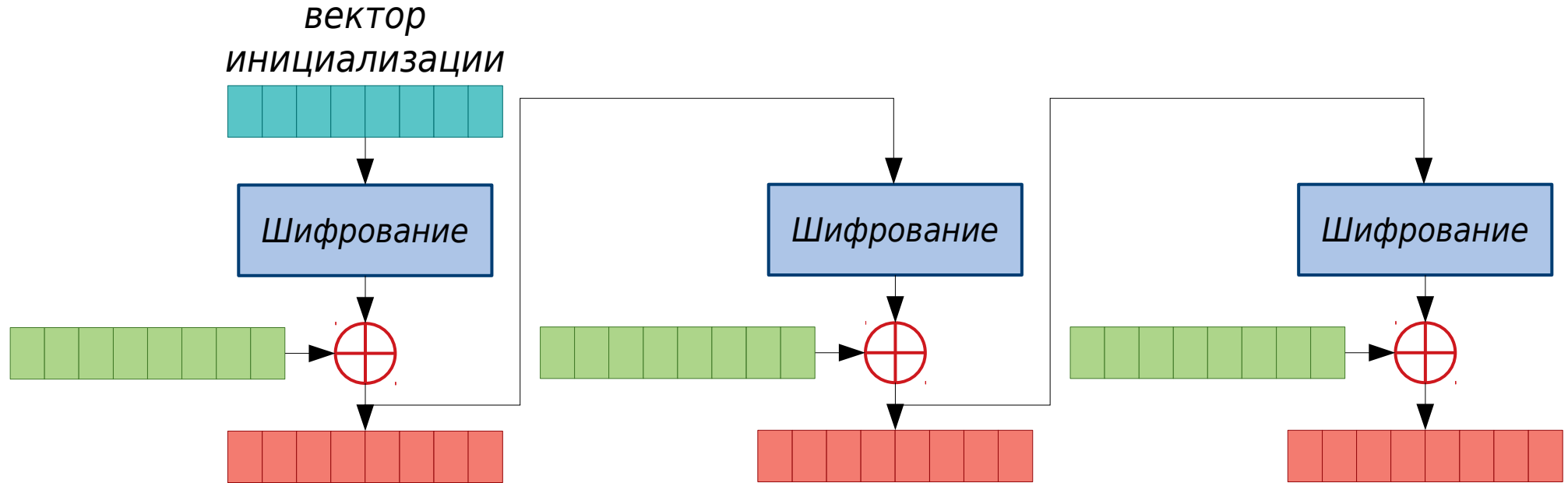
# Режим сцепления блоков шифротекста (CBC – Cipher Block Chaining)



Вектор инициализации не обязан быть секретным, но должен быть уникальным для каждого сообщения.

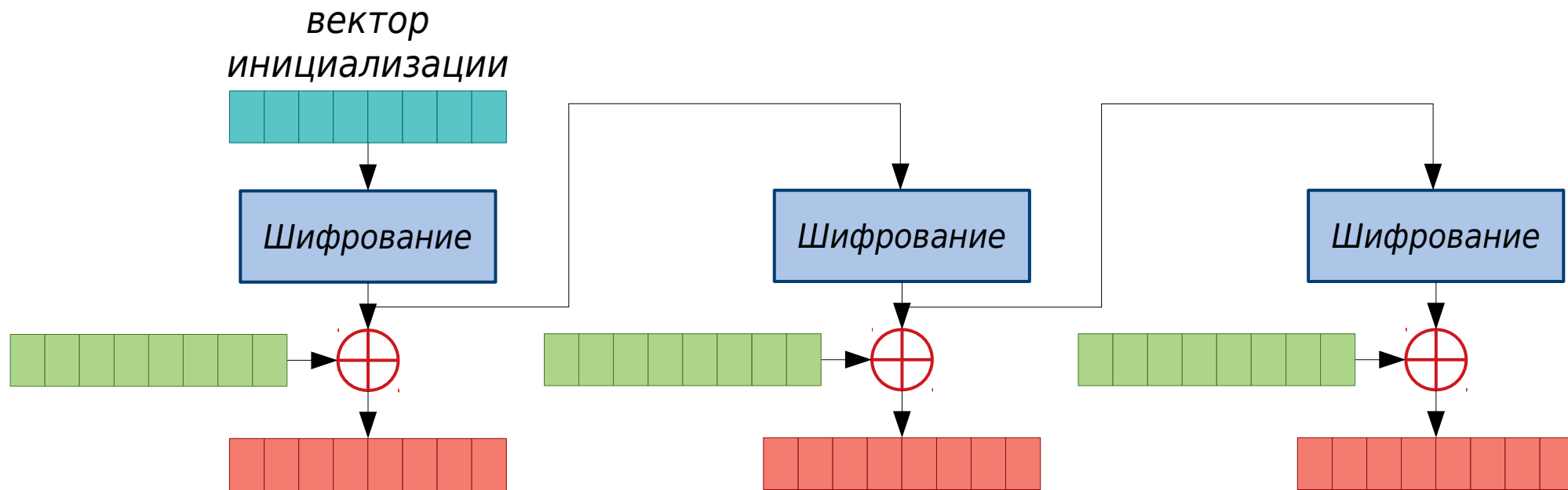


# Режим обратной связи по шифротексту (CFB — Cipher Feedback )



Вектор инициализации не обязан быть секретным, но должен быть уникальным для каждого сообщения.

# Режим обратной связи по выходу (OFB — Output Feedback)



Вектор инициализации не обязан быть секретным, но должен быть уникальным для каждого сообщения.

# Сравнение режимов шифрования

Режим	Параллельное шифрование	Параллельная дешифровка	При ошибке в 1 бите неверно дешифруется...
ECB прямая замена	да	да	1 блок
CTR счётчик	да	да	1 бит
CBC сцепление блоков	нет	да	2 блока
CFB обратная связь по шифротексту	нет	да	2 блока
OFB обратная связь по выходу	нет	нет	1 бит

# Криптоанализ блочных шифров

- **Полный перебор** (*Bruteforce*)

Работает с любыми шифрами. Только очень медленно.

Противодействие: увеличение длины ключа.

- **Линейный криптоанализ**

1. Поиск соотношений между битами открытого текста, шифротекста и ключа, которые верны с вероятностью больше  $\frac{1}{2}$ .

$$(P_{i1} \oplus P_{i2} \oplus \dots \oplus P_{ia}) \oplus (C_{j1} \oplus C_{j2} \oplus \dots \oplus C_{jb}) = K_{k1} \oplus K_{k2} \oplus \dots \oplus K_{kc}$$

2. Использование этих соотношений вместе с известными парами “открытый текст — шифротекст” для получения битов ключа.

Противодействие: хороший лавинный эффект: каждый входной бит влияет на любой выходной с вероятностью  $\frac{1}{2}$ .

- **Дифференциальный криптоанализ**

Метод рассматривает разность между двумя подобранными открытыми текстами на различных раундах шифрования.

1. Наложение раундового ключа линейно. Нелинейные элементы (S-блоки) известны, для них можно составить таблицы соответствия “разность на входе — разность на выходе”.
2. Подбирая пары открытых текстов, можно добиться нужной разности на входе последнего раунда, и определить раундовый ключ.
3. После нахождения ключа последнего раунда те же действия производятся для предпоследнего раунда и т.д. Последовательно находим все раундовые ключи.

- **Метод бумеранга**

Модификация дифференциального криптоанализа, в которой используются не пары, а четвёрки открытых текстов.

Противодействие: увеличение числа раундов, выбор стойких S-блоков.

- **Метод встречи посередине**

Если раундовые ключи независимы, можно разделить последовательность шифрования на части и вскрывать каждую часть по отдельности.

Пусть блочный шифр состоит из  $k$  раундов и использует ключ длины  $n$ .

1. Разобьём шифр на две половины по  $k/2$  раундов.  
Каждая из них использует раундовые ключи суммарной длиной  $n/2$ .
2. Для первой половины зашифруем открытый текст для всех вариантов её полуключа. Результаты запишем в таблицу.
3. Для второй половины проведём расшифровку соответствующего шифротекста для всех вариантов её полуключа. Результаты запишем в другую таблицу.
4. Находим между таблицами совпадение выхода первой половины и входа второй. Получаем соответствующую пару полуключей.

Вместо полного перебора со сложностью  $2^n$  перебор двух половин ключа со сложностью  $2^{(n/2+1)}$  и использование памяти объемом  $2^{(n/2+1)}$

Противодействие: не допускать линейно-независимых раундовых ключей.

# Задачи

1. Имеется шифратор, который выполняет блочное шифрование в одном из режимов:

прямой замены (ECB),

сцепления блоков (CBC),

обратной связи по шифротексту (CFB)

или обратной связи по выходу (OFB).

Ключ неизвестен.

Можно подавать на вход шифратора произвольные сообщения и получать соответствующие шифротексты.

Требуется определить режим шифрования.

Какое минимальное число сообщений для этого необходимо?

2. Алиса и Боб проводят взаимную аутентификацию, которая состоит в проверке знания общего секретного ключа. Стороны шифруют пары блоков, используя блочный алгоритм в режиме сцепления блоков (CBC). Вектор инициализации нулевой. Аутентификация проходит по следующему протоколу:

- Боб выбирает случайный блок  $B$  и посылает его Алисе.
- Алиса выбирает случайный блок  $A$  и посылает Бобу зашифрованную пару  $\{A, B\}$ .
- Боб расшифровывает сообщение и сравнивает блок  $B$  со своим. Если блоки отличаются, то Боб завершает протокол с ошибкой. Если совпадают, Боб признает подлинность Алисы и отправляет ей зашифрованную пару  $\{B, A\}$ .
- Алиса расшифровывает сообщение и сравнивает блок  $A$  со своим. Если блоки отличаются, то Алиса завершает протокол с ошибкой. Если совпадают, Алиса признает подлинность Боба.

Как Мэллори, не зная ключа, может выдать себя за Боба?

Какие изменения нужно внести в протокол, чтобы закрыть уязвимость?



# Ссылки

- Обратная связь:

 [android.ruberoid@gmail.com](mailto:android.ruberoid@gmail.com)

 [@androidruberoid](https://t.me/androidruberoid)

- Анонсы:

 [facebook.com/kocherga.club](https://facebook.com/kocherga.club)

 [vk.com/kocherga\\_club](https://vk.com/kocherga_club)

 [vk.com/kocherga\\_prog](https://vk.com/kocherga_prog)

- Материалы лекций:

 [github.com/notOcelot/Kocherga\\_crypto](https://github.com/notOcelot/Kocherga_crypto)

- Видео:

 [youtube.com/channel/UCeLSDFOndl4eKFutg3oowHg](https://youtube.com/channel/UCeLSDFOndl4eKFutg3oowHg)

