WPA-AES, WPA-TKIP Схема шифрования и дешифрования

Лекция 8

Как только Wi-Fi устройство получает все необходимые ключи и проходит процесс аутентификации, тогда оно может начинать передавать зашифрованный трафик. В стандарте 802.11i предусмотрено две схемы шифрования:

- Temporal Key Integrity Protocol (TKIP)
- Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP). CCMP использует блочное шифрование AES

В алгоритме TKIP решены три основные проблемы устаревшего шифрования WEP:

1. Добавлена функция генерации уникального ключа для каждого передаваемого кадра данных.
2. Добавлен счетчик кадров, для предотвращения атак типа «повторение».
3. Добавлен новый алгоритм проверки кадра на предмет его подделки или видоизменения. Алгоритм называется Message Integrity Code (MIC).

Message Integrity Code (MIC)

Основным нововведением алгоритма MIC стало то, что алгоритм использует часть ключа TK (неизвестного злоумышленнику) для вычисления проверочной суммы. Кроме того, проверочная сумма теперь включает проверку MAC-адреса отправителя и получателя, а так же метку QoS, т.е. по сути проверку важных полей заголовка.

- Схема шифрования WPA-AES, оно же Counter mode with cipher-block chaining message authentication code (CCMP) обеспечивает наивысший уровень безопасности, конфиденциальности и защиты от повторов, который может предложить стандарт 802.11. Схема шифрования основана на стандарте шифрования AES (advanced encryption standard), .

Если ТД обнаружит два неправильно вычисленных MIC в течении одной минуты, это будет расцениваться как атака и устройство, которое было источником ошибки будет заблокировано на 1 минуту. Кроме того, после разблокировки устройству будет необходимо снова пройти аутентификацию и обновить ключи.

Несмотря на небольшую вычислительную сложность алгоритм MIC является самой вычислительно-трудозатратной частью алгоритма TKIP. На ТД, использующих процессоры ARM7 и i486, пользователи чувствуют снижение скорости передачи данных даже на стандарте 802.11b (до 11 Мбит/с). Однако не существует другой альтернативы для эффективного обеспечения безопасности на канальном уровне, которая могла бы быть внедрена в старые устройства.

Счетчик кадров

- Для предотвращения атак типа «повторение» в алгоритм TKIP было внедрено использование счетчика кадров (англ. TKIP Sequence Number – TSC), чтобы и клиент и ТД вели строгий счет всем шифруемым кадрам. Такая строгость необходима для того, чтобы атакующий не смог вставить лишний кадр в процессе обмена.
- Счетчик TSC используется в алгоритме «смеси ключей», т.е. по сути, счетчик является частью ключа, который будет использоваться для шифрования. Если ТД или клиент «сбиваются» со счета, то это расценивается как результат атаки извне.

Алгоритм «смеси ключей»

Функция называется «смесь ключей» и нужна для предотвращения атаки вычисления ключа на основании статистики, так как в WEP для всех кадров использовался одинаковый ключ.

- Первая фаза смеси ключей использует 128-битный временный ключ (часть ключа PTK), MAC-адрес передающего устройства и 32 старших бита счетчика кадров TSC. В результате хеш-операции получается 80 битный вектор, называемый «TKIP-mixed Transmit Address and Key» (TTAK).
- Задача второй фазы смеси ключей - это генерация 128-битного сида, который будет использоваться RC4 как исходная точка для генерации потокового ключа. На входе этой хеш-операции будет TTAK, младшие 16 бит счетчика кадров и опять временный ключ.

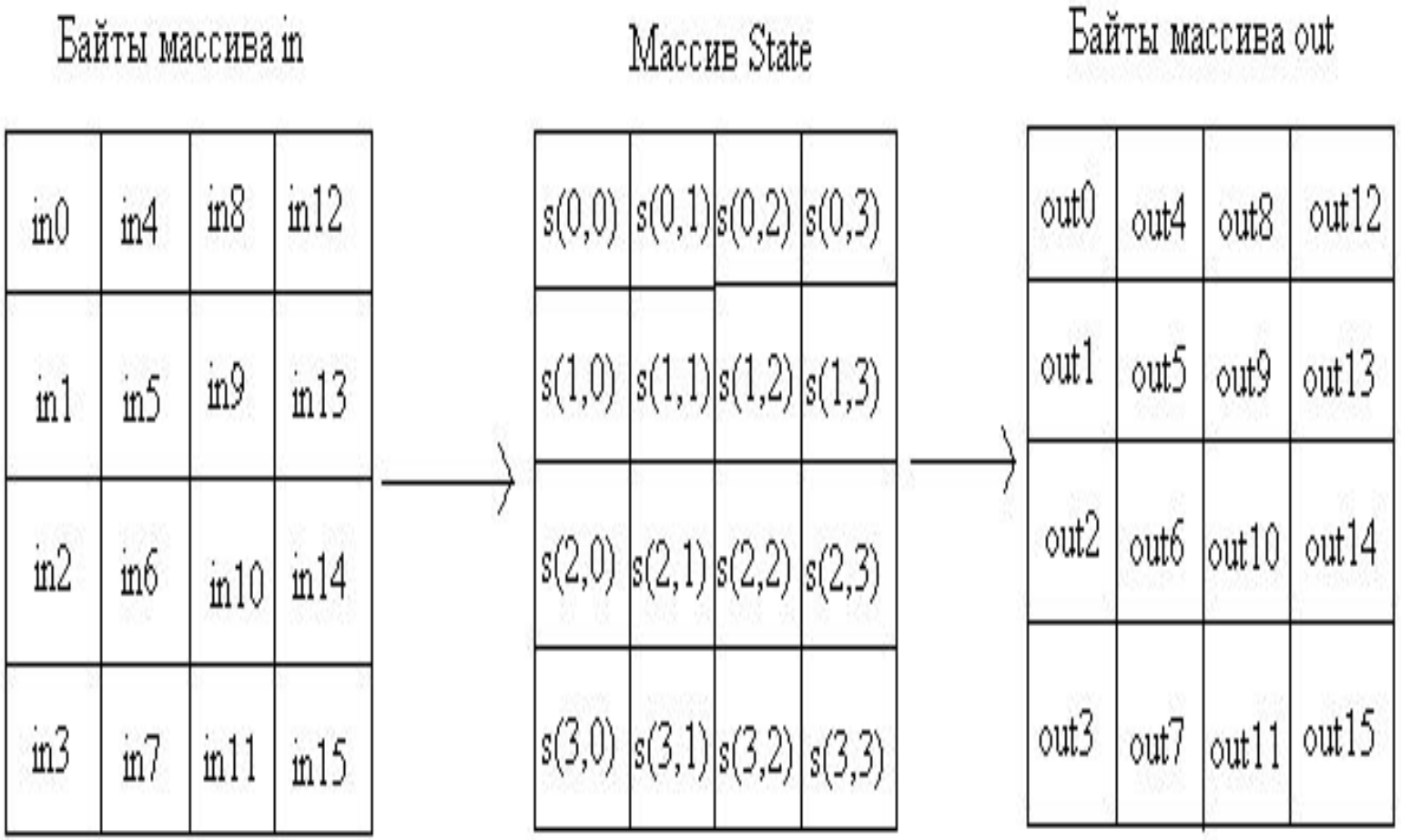
- Так как в алгоритме смеси ключей используется счетчик кадров, это позволяет создавать новый «сид» для каждого кадра. Такой подход позволяет устранить главную уязвимость старого алгоритма WEP.

- **Алгоритм AES**
- Алгоритм AES относится к симметричным методам шифрования, т.е. для шифрования и дешифрования используется один и тот же секретный ключ. Шифрование осуществляется поблочно. Длина блока данных - 128 бит. Из этого блока данных формируется массив State.

- Процесс шифрования данных в алгоритме AES можно интерпретировать как выполнение операций алгоритма над двумерным массивом байтов State.

- В массиве State, обозначаемом s, каждый отдельный байт имеет два индекса r и c, где r – номер его строки в диапазоне $0 \leq r \leq 3$; c – номер его столбца в диапазоне $0 \leq c \leq N_b-1$. Эта индексация позволяет ссылаться на конкретный байт массива State как на s [r,c]. N_b – это число 32-битных слов, составляющих State. Для стандарта AES число $N_b=4$, то есть $0 \leq c \leq 3$.

В самом начале процессов шифрования и дешифрования входной массив in – массив байтов in0, in1,…in15 – копируется в массив State, как показано. Затем в массиве State выполняются необходимые операции шифрования или дешифрования, после чего окончательное значение элементов массива State копируется в выходной массив out – массив байтов out0, out1,…out15.


Байты массива in
in0 in4 in8 in12
in1 in5 in9 in13
in2 in6 in10 in14
in3 in7 in11 in15
Массив State
s(0,0) s(0,1) s(0,2) s(0,3)
s(1,0) s(1,1) s(1,2) s(1,3)
s(2,0) s(2,1) s(2,2) s(2,3)
s(3,0) s(3,1) s(3,2) s(3,3)
Байты массива out
out0 out4 out8 out12
out1 out5 out9 out13
out2 out6 out10 out14
out3 out7 out11 out15

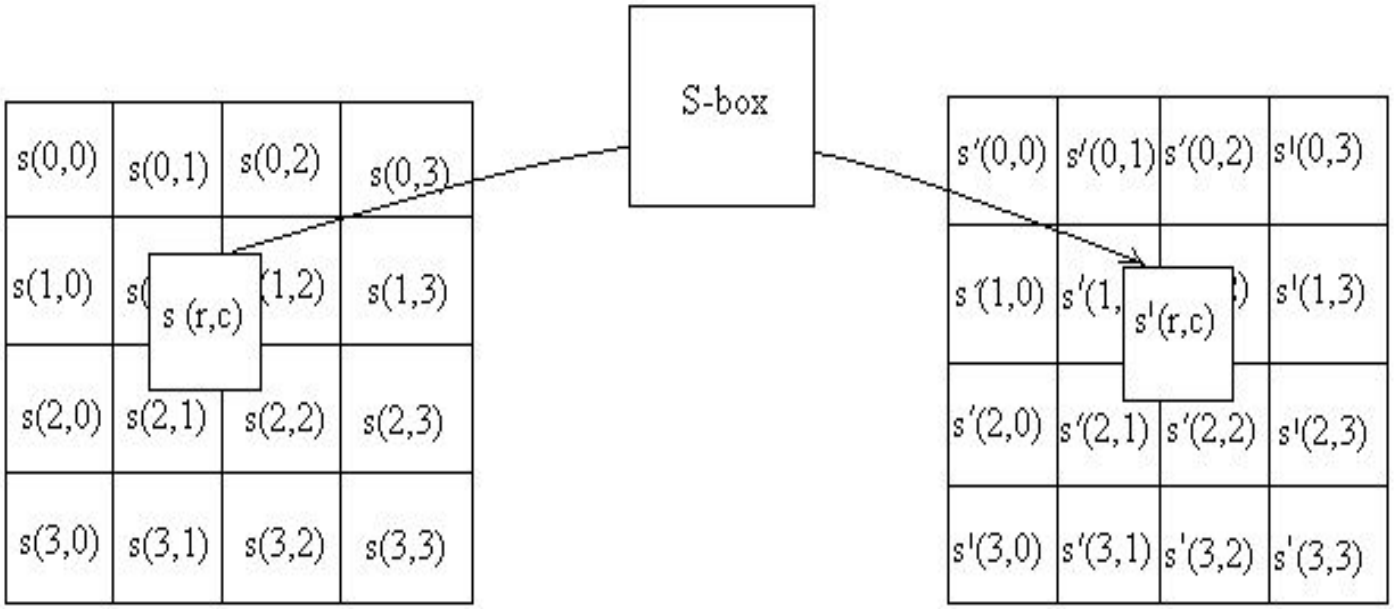
- В алгоритме AES длина входного блока, длина выходного блока и массива State (текущее состояние шифра) равны 128 бит. Длина ключа шифрования в алгоритме AES может быть равна 128, 192 или 256 бит.

- В режимах как шифрования (Cipher), так и дешифрования (Inverse Cipher) алгоритм AES использует раунд-функцию, которая включает в себя следующие четыре байт-ориентированных преобразования:

- 1) подстановку байтов (SubBytes/InvSubBytes), использующую таблицу подстановок (S-box/Inverse S-box);
- 2) сдвиги строк массива State на различные значения смещений (ShiftRows/InvShiftRows);
- 3) смешивание данных в пределах каждого столбца массива State (MixColumns/InvMixColumns);
- 4) прибавление ключа раунда Round Key к массиву State (AddRoundKey).

- В самом начале процедуры шифрования входная последовательность in копируется в массив State. После начального сложения ключом раунда Round Key массив State подвергается преобразованию с использованием раунд-функции в течение N_r раундов, причем завершающий раунд отличается от предыдущих N_r–1 раундов отсутствием процедуры MixColumns. Число раундов N_r выбирается в зависимости от длины ключа и может быть равно 10, 12 или 14 для ключа длиной 128, 192 или 256 бит соответственно. По окончании последнего раунда конечное состояние массива State копируется в выходной массив.

- **Шифрование**
- *Преобразование SubBytes* является нелинейной байтовой подстановкой, которая воздействует на каждый байт массива State, используя таблицу подстановок S-box.



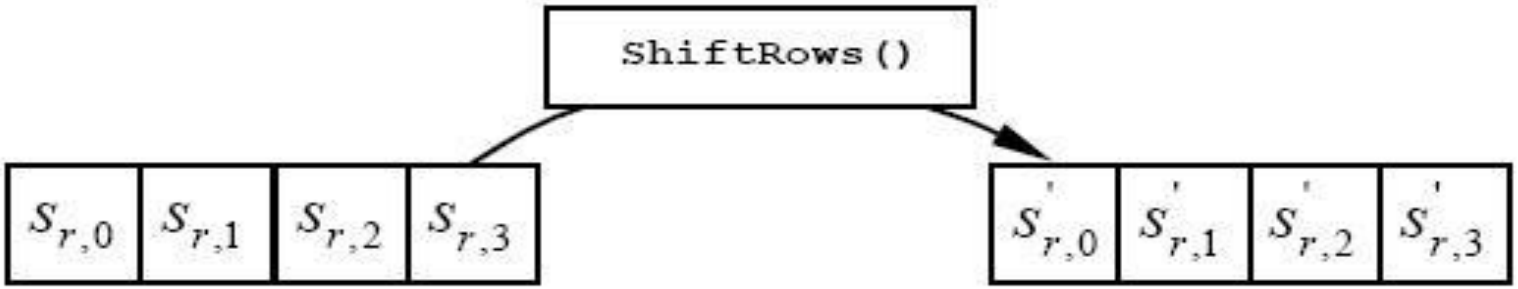
Преобразование SubBytes использует таблицу подстановок

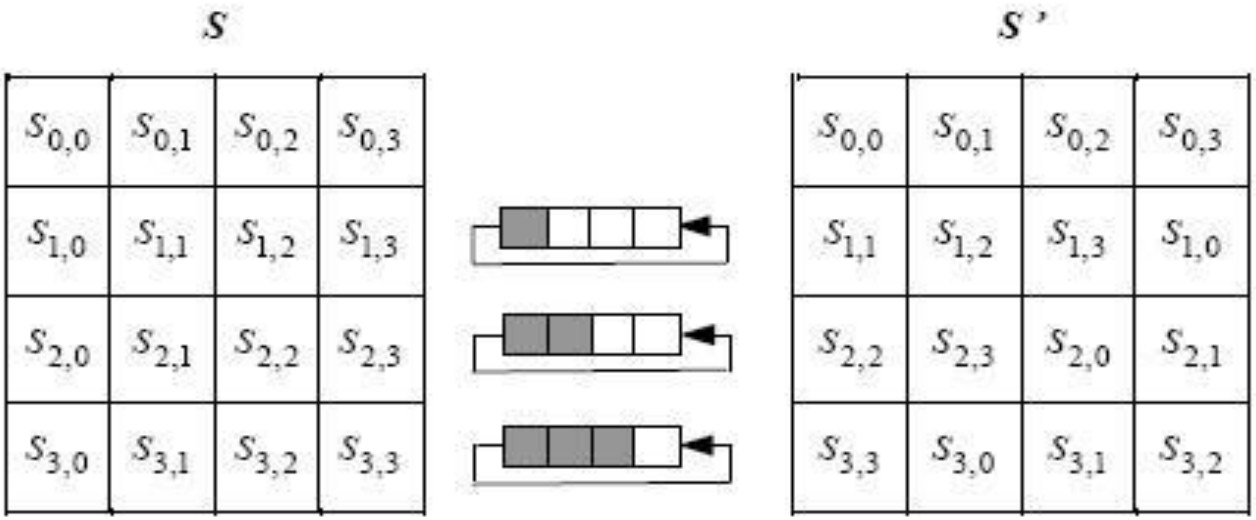
Таблица 1 – S-box

		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	0	63	7c	77	7b	f2	6b		c5	30	1	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	Fd	93	26	36		f7	cc	34	a5	e5	f1	71	d8	31	15
	3	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
	4	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3		84
	5	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	2		50	3c		a8
	7	51	a3	40		92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec		97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81		dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	6	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	8
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74		4b	bd	8b	8a
	d	70	3e	b5	66	48	3	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
X	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d		b0	54	bb	16

В преобразовании ShiftRows байты массива State циклически сдвигаются влево на расстояние, равное номеру строки (для нулевой строки величина сдвига равна нулю, т.е. байты сдвигаются только в последних трех строках). Преобразование ShiftRows выполняется следующим образом

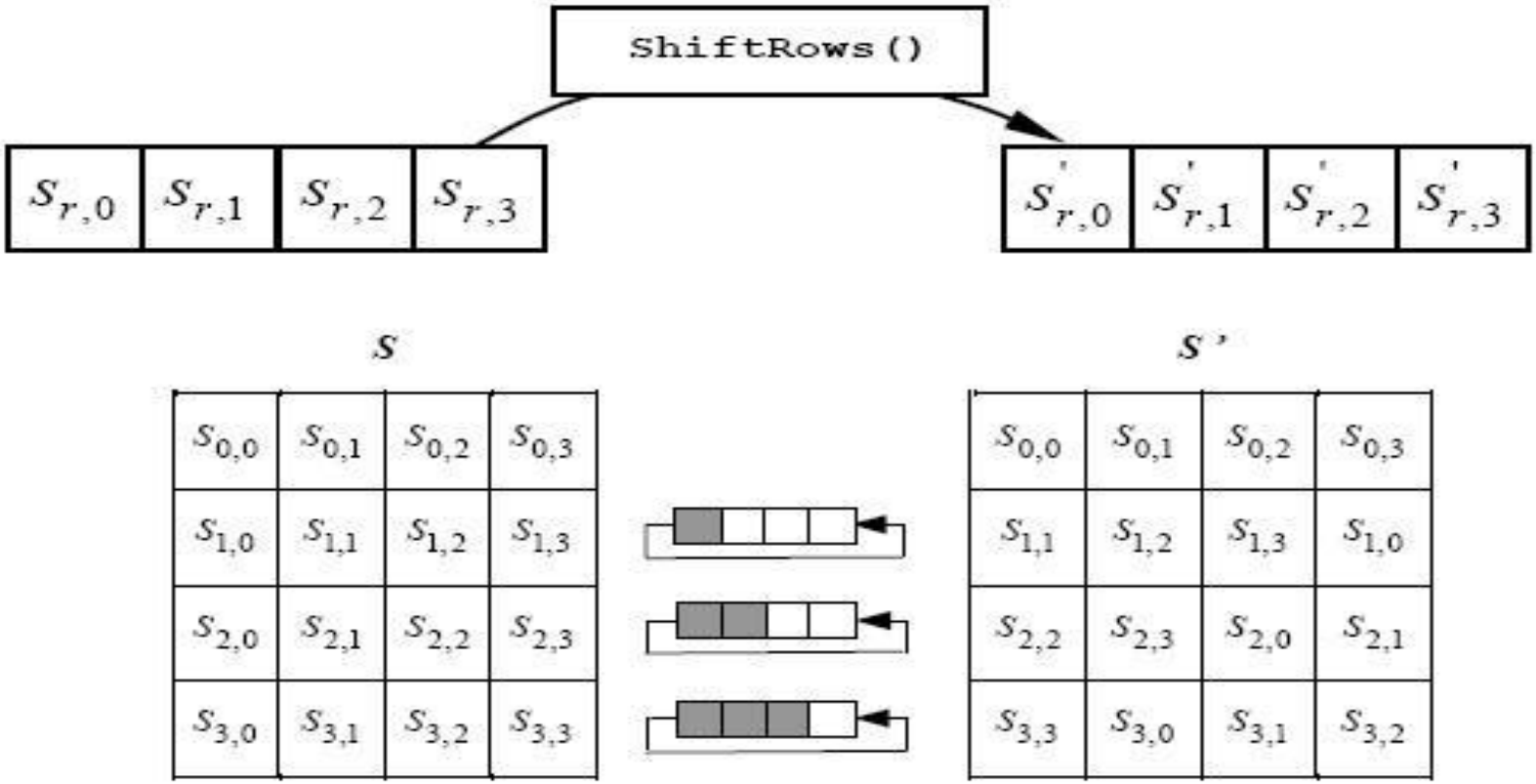
Преобразование ShiftRows циклически сдвигает три последних строки в массиве State





Преобразование MixColumns

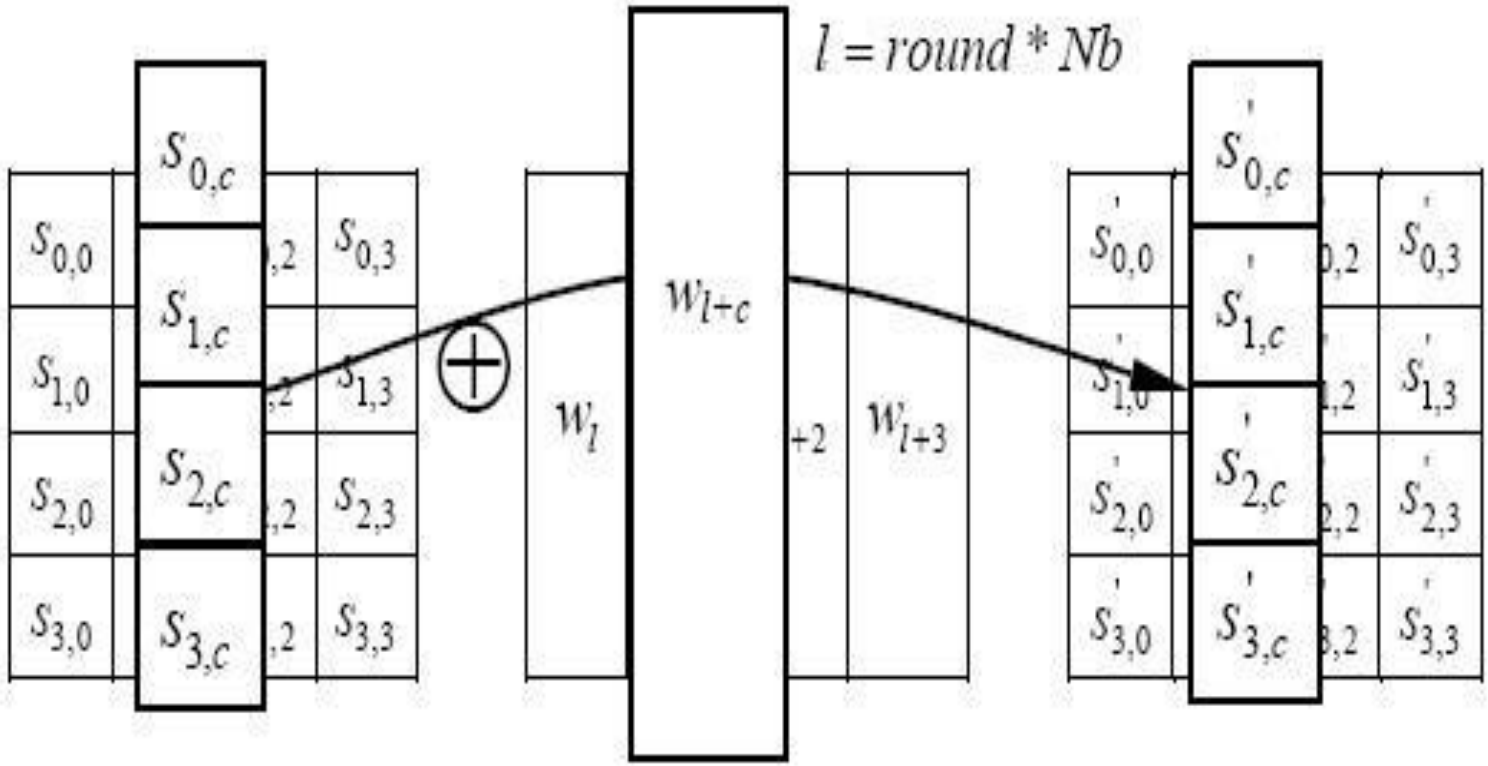
- Процедура MixColumns обрабатывает столбцы массива state. При этом преобразовании столбцы массива рассматриваются как многочлены GF в 8 степени и умножаются на многочлен



Преобразование MixColumns

В преобразовании AddRoundKey ключ раунда Round Key прибавляется к массиву State с помощью операции простого побитового сложения XOR (сложения по модулю 2). Каждый ключ раунда Round Key состоит из слов, взятых из набора ключей (key schedule), содержащихся в массиве w. Эти слов суммируются со столбцами массива State

преобразование AddRoundKey



Дешифрование

Для осуществления процедуры дешифрования инвертируются и затем выполняются в обратном порядке преобразования шифрования, описанные выше. При дешифровании массив State обрабатывается совокупностью преобразований InvShiftRows, InvSubBytes, InvMixColumns и AddRoundKey.