ДСТУ 7624:2014

Распараллеливание блочного симметричного шифра

Докладчик: Мишина Изабелла, Национальный авиационный университет
Руководитель: Влад Ковтун, ООО Сайфер БИС

Содержание

- Актуальность
- Направления оптимизации
- Результаты сравнения
- Выводы

Актуальность

Достаточно новый шифр требует изучения своих потенциальных возможностей с точки зрения эффективности программной реализации на современных вычислительных системах.

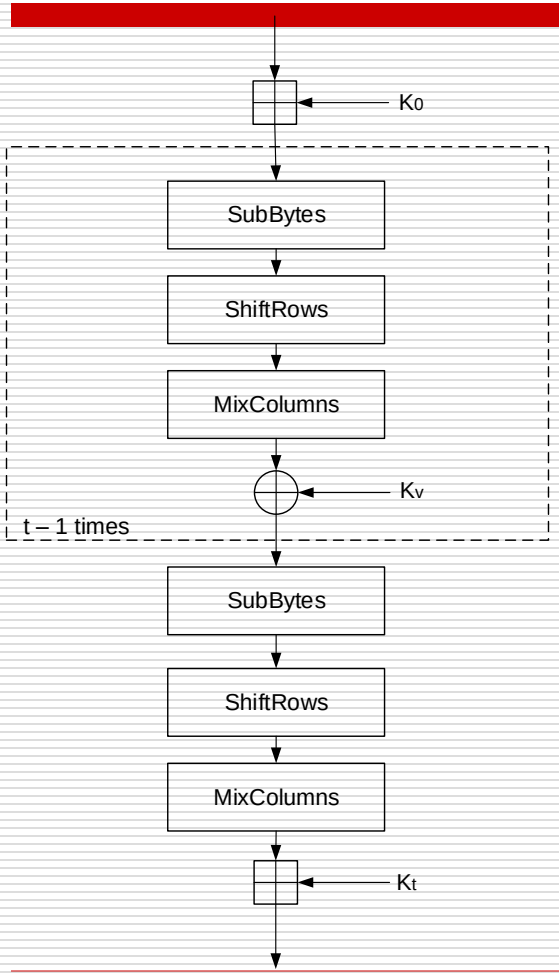
В связи с этим, **актуальной научно-технической задачей** является изучение возможности оптимизации работы шифра в различных его режимах при программной реализации.

Актуальность

- **Целью:** повышение быстродействия блочного симметричного шифра ДСТУ 7624:2014.
- **Объект:** процесс криптографической защиты информации с использованием шифра ДСТУ 7624:2014.
- **Предмет:** метод оптимизации процесса криптографической защиты информации с использованием шифра ДСТУ 7624:2014.

НАПРАВЛЕНИЯ ОПТИМИЗАЦИИ

Базовое преобразование шифра

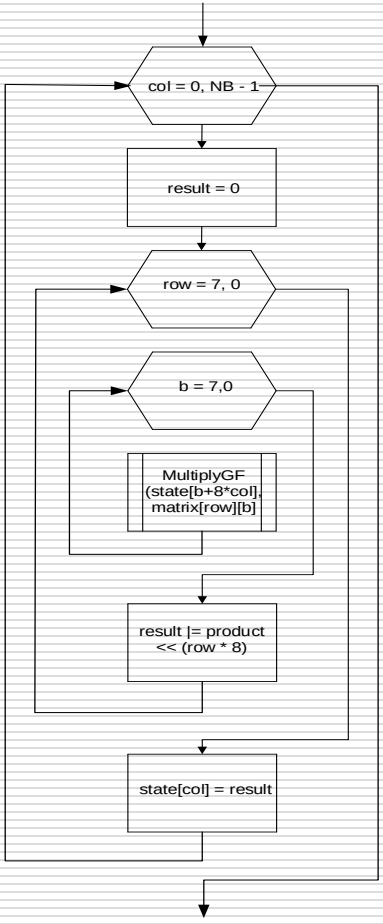


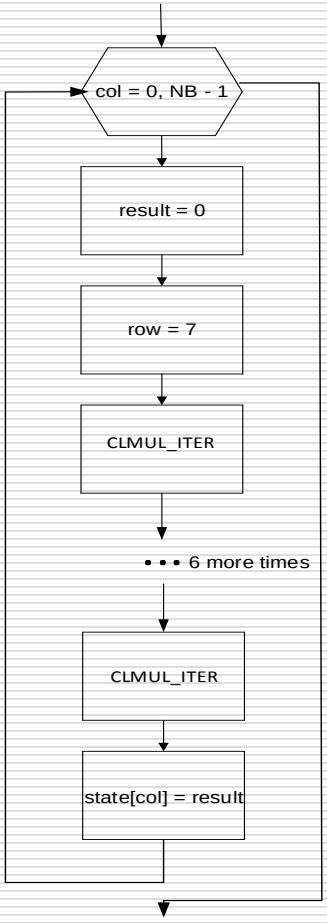
$$T_{l,k}^{(K)} = \eta_l^{(K_t)} \circ \psi_l \circ \tau_l \circ \pi_l' \circ \left(\prod_{v=1}^{t-1} \left(\kappa_l^{(K_v)} \circ \psi_l \circ \tau_l \circ \pi_l' \right) \right) \circ \eta_l^{(K_0)}$$

Направления оптимизации

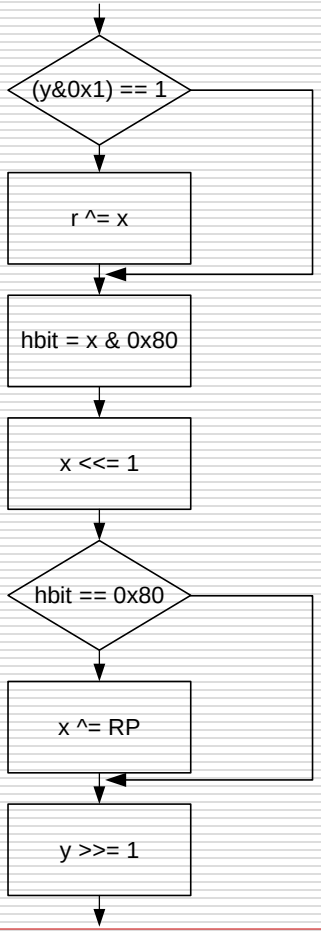
Наличие внутреннего параллелизма	Распараллеливание процедуры умножения матрицы внутреннего состояния на циркулянтную матрицу МДР-кода с использованием OpenMP v2.0.
Наличие циклов с константным небольшим числом итераций	Развертывание циклов операций SubBytes, ShiftRow, RotateWords, AddRoundKey, XorRaundKey, SubRoundKey, внутренних циклов MixColumns.
Возможность упростить операцию приведения в поле	Использование вместо двух байт 0x011D одного байта 0x1D, что не требует дополнительных расходов на выравнивание в MixColumns.
Дополнительные возможности при использовании C++	Использование шаблонов, что позволяет отказаться от лишних ветвлений в циклах.

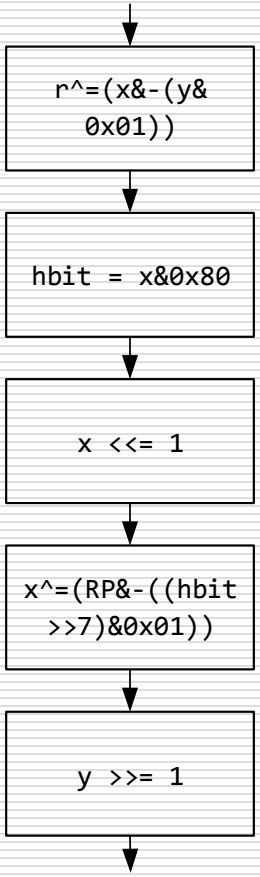
MixColumns. Разворачивание внутренних циклов



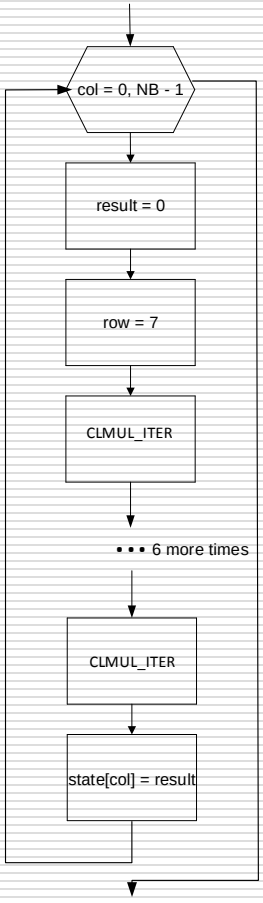


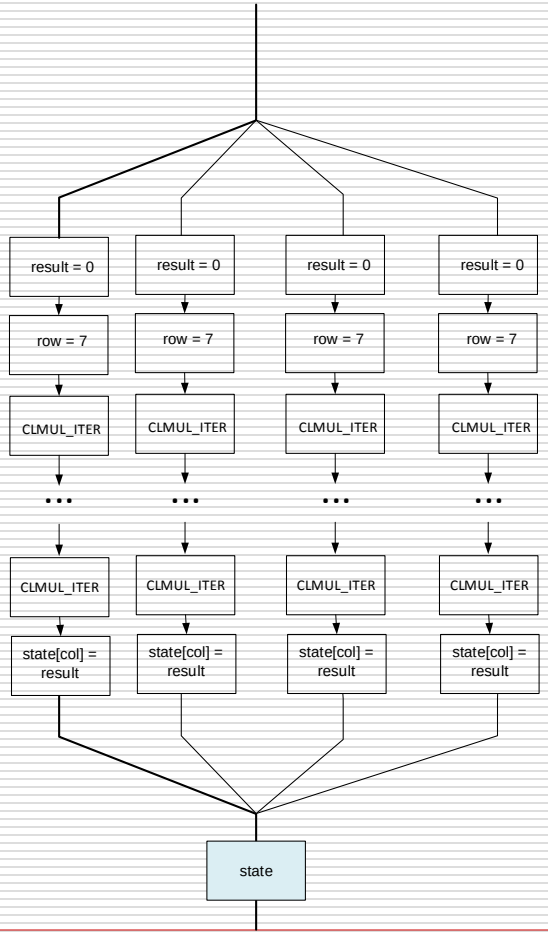
MixColumns. Умножение в поле GF(2^8). Уход от сравнений





MixColumns. Распараллеливание внешнего цикла





Условия эксперимента

- Язык: C++
- Компилятор: Microsoft Visual Studio 2015
- ОС: Microsoft Windows 8.1 x86-64
- CPU: Intel Core i7-4720 HQ 2,6 Ггц (4 core with HT)
- Turbo-Boost: Отключен
- Количество итераций: 1000
- Объем данных для шифрования: 100 Мб
- Не рассматривались режим 128/256 и 256/512
- Эталонная программная реализация Романа Олейникова: https://github.com/Roman-Oliynykov/Kalyna-reference

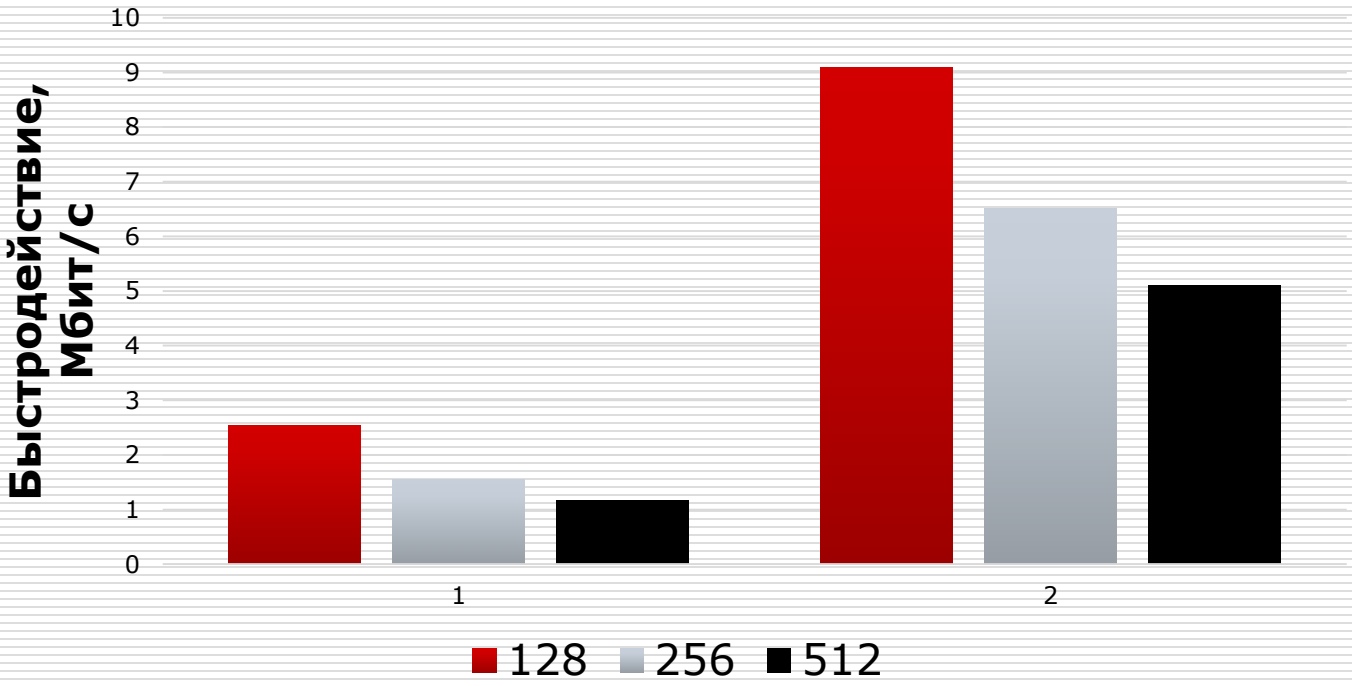
Сравнение быстродействия до и после упрощения операции умножения в GF(2^8)

Длина блока, бит	Быстродействие, Мбит/с	
	1	2
128	2,5	9,1
256	1,5	6,6
512	1,2	5,2

1 - Использование шаблонов и разворачивание циклов основных преобразований, помимо MixColumns

2 - Использование шаблонов и разворачивание циклов основных преобразований, помимо MixColumns, с упрощением операции умножения в GF(2^8)

Сравнение быстродействия алгоритма до и после упрощения операции умножения в GF(2^8)



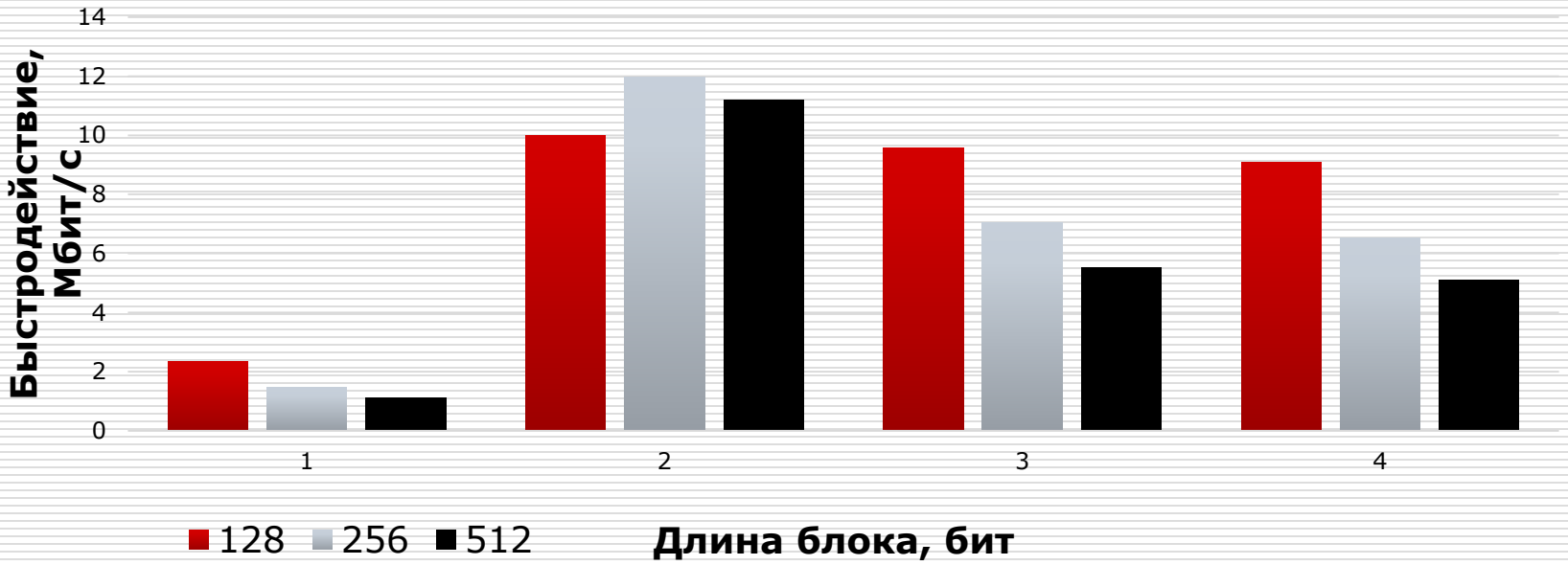
1 - Использование шаблонов и разворачивание циклов основных преобразований, помимо MixColumns
2 - Использование шаблонов и разворачивание циклов основных преобразований, помимо MixColumns, с упрощением операции умножения в GF(2^8)

Сравнение реализаций

Длина блока, бит	Быстродействие, Мбит/с			
	1	2	3	4
128	2,35	10,01 (2)	9,1	9,09
256	1,47	11,97 (4)	7,03	6,52
512	1,12	11,20 (4)	5,5	5,10

1 – Эталонная реализация
2 - Разворачивание циклов, упрощение операции умножения на байт в GF(2^8) и распараллеливание
3 - Разворачивание циклов, упрощение операции умножения на байт в GF(2^8)
4 - Использование шаблонов и разворачивание циклов базовых операций, помимо матричного умножения, упрощение операции умножения на байт в GF(2^8)

Сравнение реализаций



1 – Эталонная реализация

2 - Разворачивание циклов, упрощение операции умножения на байт в GF(2^8) и распараллеливание

3 - Разворачивание циклов, упрощение операции умножения на байт в GF(2^8)

4 - Использование шаблонов и разворачивание циклов базовых операций, помимо матричного умножения, упрощение операции умножения на байт в GF(2^8)

Выводы

Были получены следующие **результаты**:

1. Исследован блочный симметричный шифр ДСТУ 7624:2014, что позволило:

- Выявить возможность распараллеливания преобразования MixColumn.
- Развернуть циклы в преобразованиях ShiftRow, RotateWords, AddRoundKey, XorRaundKey, SubRoundKey, внутренние циклов MixColumns.
- Отказаться от операций сравнения при умножении в поле GF(2^8)
- Отказаться от сравнений за счет использования шаблонов
- Упростить неприводимый полином 0x011D до 0x1D.

Выводы

2. В результате сравнения быстродействия разработанной программной реализации блочного симметричного шифра ДСТУ 7624:2014 с эталонной выяснилось, что быстродействие увеличилось в 4 раза для 128 битного блока, в 8 раз для 256 битного и в 10 раз для 512 битного блока. В сравнении с однопоточной реализацией повышения быстродействия удалось достичь для 128 битного блока на 10%, для 256 битного на 40% и для 512 битного в два раза.

Спасибо за внимание

Вопросы?