(19) Евразийское патентное ведомство

(11) 021803 (13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2015.09.30

(21) Номер заявки
201200672

(22) Дата подачи заявки
2012.04.25

(51) Int. Cl. ***H04L 9/28*** (2006.01)
G09C 1/00 (2006.01)

(54) СПОСОБ ШИФРОВАНИЯ ДАННЫХ ДЛЯ ВЫЧИСЛИТЕЛЬНЫХ ПЛАТФОРМ С SIMD-АРХИТЕКТУРОЙ

(43) **2013.10.30**
(96) **2012000119 (RU) 2012.04.25**

(71)(73) Заявитель и патентовладелец:
ОТКРЫТОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО "ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И КОММУНИКАЦИОННЫЕ СИСТЕМЫ" (ОАО "ИНФОТЕКС") (RU)

(72) Изобретатель:
Тычина Леонид Анатольевич (RU)

(74) Представитель:
Медведев В.Н. (RU)

(56) RU-C2-2390949
US-A1-20040184602
US-A1-20090055458
US-A1-20070237324

(57) Изобретение относится к шифрованию данных. Предложена методика реализации цикла шифрования данных для вычислительных платформ с SIMD-архитектурой, позволяющая избежать вынужденного смешивания инструкций стандартной архитектуры и SIMD-инструкций за счет специализированной адаптации узлов замены, обеспечивающей возможность выполнения операции замены исключительно при помощи SIMD-инструкций. Технический результат заключается в приросте производительности.

021803 B1

021803 B1

Область техники, к которой относится изобретение

Изобретение, в общем, относится к шифрованию данных, в частности к шифрованию данных согласно ГОСТ 28147-89 для вычислительных платформ с SIMD-архитектурой.

Описание предшествующего уровня техники

В современной технике шифрование данных играет значительную роль и имеет весьма широкое применение. Как следствие, к разработкам в области шифрования данных приковано серьезное внимание на различных уровнях.

Национальный стандарт Российской Федерации ГОСТ 28147-89 устанавливает единый алгоритм криптографического преобразования для систем обработки информации в сетях ЭВМ и отдельных вычислительных комплексах. ГОСТ 28147-89, содержимое которого по ссылке полностью включено в настоящее описание, вводит алгоритм блочного шифрования с размером блока 64 двоичных разряда (далее - разряда) и 256-разрядным ключом. В криптографическом преобразовании используется блок подстановки, состоящий из восьми узлов замены. Каждый узел замены представляет собой таблицу из шестнадцати 4-разрядных чисел и обеспечивает взаимно однозначное отображение множества 4-разрядных чисел на себя. При этом следует отметить, что стандарт не специфицирует конкретного набора констант для блока подстановки. Это означает, что разработчик может по своему усмотрению выбрать его наполнение.

В ГОСТ 28147-89 описывается ряд режимов шифрования, а именно: простой замены, гаммирования, гаммирования с обратной связью и выработки имитовставки. Цикл шифрования составляет основу каждого из упомянутых режимов, которые отличаются друг от друга специфическим комбинированием входных параметров.

В настоящее время вычислительные платформы с SIMD-архитектурой широко применяются на практике. Такая платформа, в общем, допускает параллельную обработку N потоков данных, так как ее аппаратная компонента состоит из N независимых, но идентичных по ресурсоемкости процессорных элементов, при этом управление обработкой осуществляется единым потоком команд. Каждый процессорный элемент располагает локальной оперативной памятью. Процессорные элементы получают команды от одного центрального контроллера команд и работают синхронно, исполняя на каждом шаге одну и ту же команду. При этом каждый процессорный элемент оперирует с данными из локальной оперативной памяти, т.е. основной характеристикой SIMD-архитектуры является то, что все N процессорных элементов исполняют фиксированную команду, но над различными данными.

Известны процессоры с SIMD-архитектурой производства компаний Intel или AMD, которые помимо инструкций стандартной архитектуры включают также набор специализированных инструкций (так называемых SIMD-инструкций), в частности набор инструкций SSE-технологии (Streaming SIMD Extensions).

Принимая во внимание обеспечиваемые SIMD-архитектурой высокие рабочие характеристики, обуславливаемые вышеупомянутым распараллеливанием, перспективной представляется реализация на ней шифрования данных согласно ГОСТ 28147-89 с целью повышения производительности. Однако экспериментально показано, что при использовании исключительно SIMD-инструкций невозможно реализовать цикл шифрования ГОСТ 28147-89 в точном соответствии с описанным в стандарте алгоритмом. Суть проблемы заключается в том, что SIMD-инструкций не позволяют осуществить выполняемую при шифровании данных операцию замены так, как это указано в ГОСТ 28147-89. Очевидно, что для выполнения этой операции можно воспользоваться инструкциями стандартной архитектуры. Однако в итоге производительность не увеличивается, а наоборот снижается. Такое снижение обусловлено дополнительными накладными расходами, которые возникают в результате согласования данных при переключении между инструкциями SIMD- и стандартной архитектур.

Таким образом, в технике существует потребность в методике реализации цикла шифрования ГОСТ 28147-89, исключающей смешивание инструкций стандартной архитектуры и SIMD-инструкций.

Для алгоритма блочного шифрования DES известна реализация с применением SIMD-инструкций (см. E. Biham, "A fast new DES implementation in software", Lecture Notes in Computer Science, vol. 1267, pp. 260-272, 1997), в которой предлагается использовать SIMD-инструкции над 64-разрядными регистрами для параллельного шифрования шестидесяти четырех независимых потоков данных. Однако использование описанного в этой статье подхода для реализации цикла шифрования по ГОСТ 28147-89 с конкретным наполнением узлов замены не применимо в силу существенных алгоритмических различий DES и ГОСТ 28147-89.

Возможный подход к реализации национального стандарта шифрования предлагается в публикации А. Винокурова "Алгоритм шифрования ГОСТ 28147-89, его использование и программная реализация для компьютеров платформы Intelx86", "Монитор", №1, с. 5, 1995, которая в переработанном и дополненном виде доступна по адресу http://www.enlight.ru/crypto/articles/vinokurov/gost_i.htm.

Данная публикация содержит описание реализации стандарта ГОСТ 28147-89 для платформ Intelx86, однако специфика SIMD-архитектуры при этом никак не учитывается.

Краткое описание сущности изобретения

Предлагается методика реализации цикла шифрования данных согласно ГОСТ 28147-89 на SIMD-

архитектуре, позволяющая избежать вынужденного смешивания инструкций стандартной архитектуры и SIMD-инструкций за счет специализированной адаптации узлов замены, обеспечивающей возможность выполнения операции замены исключительно при помощи SIMD-инструкций. При реализации на SIMD-архитектуре заданного типа предлагаемый подход обеспечивает двукратный прирост производительности по сравнению с реализацией исключительно на основе инструкций стандартной архитектуры.

Согласно настоящему изобретению предлагается реализация цикла шифрования данных по ГОСТ 28147-89 для компьютерных систем, имеющих SIMD-архитектуру. Предлагаемый способ содержит этапы, на которых инициализируют маски M_k (k=1, ..., 6), причем каждая маска представляет собой массив размерности 4, каждый элемент которого представляет собой уникальную для маски 32-разрядную константу; реорганизуют оговоренные в ГОСТ 28147-89 предварительно заполненные узлы замены K_m (m=1, ..., 8), каждый из которых представляет собой массив 4-разрядных чисел размерности 16, в массивы T_j (j=0, ..., 3) 4-разрядных чисел размерности 32 следующим образом:

$$\begin{aligned} &T_0[2(n)] = K_1[n], T_0[2(n)+1] = K_8[n], \\ &T_1[2(n)] = K_5[n], T_1[2(n)+1] = K_4[n], \\ &T_2[2(n)] = K_3[n], T_2[2(n)+1] = K_2[n], \\ &T_3[2(n)] = K_7[n], T_3[2(n)+1] = K_6[n] \end{aligned}, \quad n = 0, \ldots, 15; \text{ и}$$

выполняют цикл шифрования. Отметим, что количество выполнений упомянутого цикла шифрования определено в ГОСТ 28147-89.

Предпочтительно, SIMD-архитектура является SIMD-архитектурой компании Intel или аналогичной архитектурой компании AMD, образуемой процессором, имеющим четыре процессорных элемента.

При выполнении цикла шифрования формируют массивы 32-разрядных элементов размерности 4:

$$N_1 = \{N_{1,i}\}, \quad N_2 = \{N_{2,i}\}, \quad X = \{X_i\}, \quad i = 0, \ldots, 3,$$

где $N_{1,i}$, $N_{2,i}$ - подблоки, на которые предварительно разбит 64-разрядный блок шифруемых данных, а X_i - часть 256-разрядного ключа шифрования, выбираемая в соответствии с ГОСТ 28147-89. Выбор части ключа шифрования предпочтительно определяется режимом шифрования, предусмотренным ГОСТ 28147-89. С каждым из процессорных элементов связано отдельное локальное запоминающее устройство, в котором сохраняется соответственно один из наборов $N_{1,i}$, $N_{2,i}$, X_i (i=0, ..., 3) для независимой обработки этим процессорным элементом. В то же время узлы замены K_m (m=1, ..., 8) в совокупности образуют блок подстановки, который является единым для всех процессорных элементов.

При выполнении цикла шифрования далее формируют массив CM_1 путем выполнения операции суммирования по модулю 2^{32} в отношении массивов N_1 и X. Затем формируют первый вспомогательный массив Y_0 путем применения к массиву CM_1 операции поразрядного сдвига вправо на четыре разряда, формируют второй вспомогательный массив Y путем применения операции логического ИЛИ к результату операции логического И в отношении маски M_1 и массива CM_1 и результату операции логического И в отношении маски M_2 и первого вспомогательного массива Y_0 и формируют третий вспомогательный массив Z путем применения операции логического ИЛИ к результату операции логического И в отношении маски M_1 и первого вспомогательного массива Y_0 и результату операции логического И в отношении маски M_2 и массива CM_1. После этого формируют первый промежуточный массив R_0 путем применения операции логического И к маске M_2 и результату выполнения операции перемешивания в отношении массива T_0 и второго вспомогательного массива Y, формируют второй промежуточный массив R_1 путем применения операции логического И к маске M_5 и результату выполнения операции перемешивания в отношении массива T_2 и третьего вспомогательного массива Z, формируют третий промежуточный массив R_2 путем применения операции логического И к маске M_4 и результату выполнения операции перемешивания в отношении массива T_1 и второго вспомогательного массива Y, формируют четвертый промежуточный массив R_2 путем применения операции логического И к маске M_6 и результату выполнения операции перемешивания в отношении массива T_3 и третьего вспомогательного массива Z и формируют пятый промежуточный массив R, объединяющий ранее полученные промежуточные массивы с первого по четвертый, путем применения операции логического ИЛИ к результату операции логического ИЛИ в отношении первого промежуточного массива R_0 и третьего промежуточного массива R_2 и результату операции логического ИЛИ в отношении второго промежуточного массива R_1 и четвертого промежуточного массива R_3. Применением операции логического И к соответствующей маске и соответствующему результату операции перемешивания обнуляются незначащие значения в получающемся промежуточном массиве.

Затем при выполнении цикла шифрования в отношении пятого промежуточного массива R выполняют циклический сдвиг путем применения операции логического ИЛИ к результату операции поразрядного сдвига влево на одиннадцать разрядов в отношении массива R и результату операции поразрядного сдвига вправо на двадцать один разряд в отношении массива R. После этого формируют представляющий шифрованные данные массив CM_2 путем выполнения операции поразрядного суммирования по модулю 2 в отношении пятого промежуточного массива, прошедшего циклический сдвиг, и массива N_2.

Упомянутые выше операции логического И, логического ИЛИ, сдвига вправо/влево, суммирования

по модулю 2, суммирования по модулю 2^{32} и перемешивания непосредственно предусмотрены SIMD-архитектурой.

Согласно настоящему изобретению также предлагается машиночитаемый носитель, на котором сохранены машиноисполняемые команды, которые при их исполнении компьютерной системой, имеющей SIMD-архитектуру, предписывают этой компьютерной системе выполнять заявляемый способ шифрования данных согласно ГОСТ 28147-89.

Перечень фигур чертежей

На фиг. 1 показана логическая блок-схема цикла шифрования; на фиг. 2 - логическая блок-схема операции замены согласно настоящему изобретению.

Подробное описание изобретения

При раскрытии заявляемого предложения для удобства воспользуемся специализированным псевдоязыком для описания процедур, каждая из которых соответствует конкретной используемой SIMD-инструкции (например, из набора SSE2, SSE3, SSSE3).

При описании процедур будем интерпретировать переменную как массив. Предположим, A - это 128-разрядная переменная, тогда старшие 32 разряда будем обозначать как A[0], следующие в порядке убывания 32 разряда - как A[1] и так далее до A[3]. Аналогично поступим при разбиении переменной на более мелкие части, например по восемь разрядов, тогда массив будет иметь размерность шестнадцать. Способ разбиения переменной однозначно следует из диапазона значений, который принимает индекс при адресации в массиве.

Для поразрядных логических операций И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ, СДВИГ ВПРАВО, СДВИГ ВЛЕВО воспользуемся обозначениями or, and, xor, shr, shl соответственно.

Итак, в предлагаемом подходе используется следующий набор процедур, где в угловых скобках приводятся соответствующие SIMD-инструкции.

A = **загрузить**(B) **<_mm_set1_epi32>**

Вход: B - 32-разрядное беззнаковое целое число.

Выход: A - массив 32-разрядных беззнаковых чисел размерности четыре и $A[i]=B$, где $0 \le i \le 3$.

A = **сумма**(B,C) **<_mm_add_epi32>**

Вход: B и C - массивы 32-разрядных беззнаковых целых чисел размерности четыре.

Выход: A - массив 32-разрядных беззнаковых целых чисел размерности четыре и

$A[i]=B[i]+C[i]$, где $0 \le i \le 3$.

Иными словами, данная операция является операцией суммирования по модулю 2^{32}, суть которой заключается в том, что при переполнении, которое возникает в результате суммирования, значения двоичных разрядов, вышедших за пределы разрядной сетки, игнорируются.

A = **сдвиг_влево**(B,C) **<_mm_slli_epi>**

Вход: B - массив 32-разрядных беззнаковых целых чисел размерности четыре, C - целое положительное число, $0 \le C \le 32$.

Выход: A - массив 32-разрядных беззнаковых целых чисел размерности четыре и

$A[i]=B[i]$ **shl** C, где $0 \le i \le 3$.

При сдвиге влево на один двоичный разряд происходит смещение двоичной последовательности в направлении старшего значащего разряда, при этом само значение старшего значащего разряда выходит за пределы разрядной сетки и игнорируется. На позицию младшего значащего разряда заносится значение двоичного нуля.

A = **сдвиг_вправо**(B,C) **<_mm_srli_epi32>**

Вход: B - массив 32-разрядных беззнаковых целых чисел размерности четыре, C - целое положительное число, $0 \le C \le 32$.

Выход: A - массив 32-разрядных беззнаковых целых чисел размерности четыре и

$A[i]=B[i]$ **shr** C, где $0 \le i \le 3$.

Аналогично, при сдвиге вправо на один двоичный разряд младший значащий разряд сдвигаемой двоичной последовательности выходит за пределы разрядной сетки и игнорируется, а на позицию старшего значащего разряда заносится значение двоичного нуля.

A = **или**(B, C) <_mm_or_si128>

Вход: B и C - массивы 32-разрядных беззнаковых целых чисел размерности четыре.

Выход: A - массив 32-разрядных беззнаковых целых чисел размерности четыре и

$A[i] = B[i]$ **or** $C[i]$, где $0 \le i \le 3$.

A = **и**(B, C) <_mm_and_si128>

Вход: B и C - массивы 32-разрядных беззнаковых целых чисел размерности четыре.

Выход: A - массив 32-разрядных беззнаковых целых чисел размерности четыре и

$A[i] = B[i]$ **and** $C[i]$, где $0 \le i \le 3$.

A = **искл_или**(B, C) <_mm_xor_si128>

Вход: B и C- массивы 32-разрядных беззнаковых целых чисел размерности четыре.

Выход: A - массив 32-разрядных беззнаковых целых чисел размерности четыре и

$A[i] = B[i]$ **xor** $C[i]$, где $0 \le i \le 3$.

A = **перемешать**(B, C) <_mm_shuffle_epi8>

Вход: B и C - массивы 8-разрядных беззнаковых целых чисел размерности шестнадцать.

Выход: A - массив 8-разрядных беззнаковых целых чисел размерности шестнадцать.

Если $C[j]$ **and** $0x80 \neq 0x00$, то $A[j] = 0x00$, иначе $A[j] = B[C[j]$ **and** $0x0F]$, где $0 \le j \le 15$.

Операция позволяет сформировать массив A из значений массива B. Значения младших четырех разрядов некоторого элемента массива C определяют позицию элемента из массива B для записи в позицию массива A, соответствующую позиции данного элемента массива C. Если старший разряд некоторого элемента C имеет значение двоичной единицы, то на соответствующую позицию A заносится нулевое значение.

Далее, при раскрытии заявляемого предложения используются следующие обозначения, соответствующие ГОСТ 28147-89:

$X_j, 0 \le j \le 7$ - ключевое хранилище на 256 разрядов, состоящее из восьми 32-разрядных накопителей. Здесь под накопителем понимается ячейка памяти;

N_1, N_2 - 32-разрядные накопители, в которые перед началом исполнения цикла шифрования заносится 64-разрядный блок входных данных, предварительно разбитый на два 32-разрядных подблока;

CM_1 - 32-разрядный накопитель, в котором сохраняется результат суммирования по модулю 2^{32};

CM_2 - 32-разрядный накопитель, в котором сохраняется результат поразрядного суммирования по модулю 2 (иными словами, поразрядной логической операции ИСКЛЮЧАЮЩЕЕ ИЛИ). По окончании исполнения цикла шифрования в этом же накопителе сохраняется итоговый результат;

$R_m, 1 \le m \le 8$ - восемь 32-разрядных накопителей, которые используются в промежуточных вычислениях;

K_m, $1 \le m \le 8$ - восемь узлов замены, в совокупности составляющих блок подстановки.

Как отмечалось ранее, каждый узел замены представляет собой таблицу из шестнадцати 4-разрядных чисел. Соответственно для хранения одного узла замены необходимо зарезервировать 64 разряда. Обобщенно говоря, замена выполняется следующим образом: число L задает номер позиции в узле замены K_m, затем выполняется считывание соответствующего 4-разрядного числа с последующей его записью в накопитель R_m.

Далее приводится описание отвечающей настоящему изобретению реализации цикла шифрования данных согласно ГОСТ 28147-89 для платформы с SIMD-архитектурой.

Пусть имеется компьютерная система с SIMD-архитектурой компании Intel (или схожей архитектурой компании AMD), образуемой процессором, имеющим четыре процессорных элемента. Как было сказано ранее, с каждым из процессорных элементов связана собственная локальная оперативная память.

С учетом вышеприведенных обозначений вводятся переменные $N_{1,i}$, $N_{2,i}$, X_i, $0 \le i \le 3$. Так, $N_{1,i}$, $N_{2,i}$ обозначают 32-разрядные подблоки данных, над которым необходимо выполнить один цикл шифрования, X_i обозначает 32-разрядную часть 256-разрядного ключа, используемую циклом шифрования. При

этом, порядок выбора части ключа описан в ГОСТ 28147-89.

Согласно стандарту, перед исполнением цикла шифрования в накопители N_1, N_2, X_j, $0 \le j \le 7$ необходимо загрузить определенные значения. Тогда содержимое переменных $N_{1,0}$, $N_{2,0}$, X_0, соответствует значениям накопителей N_1, N_2, X_j первого процессорного элемента, а содержимое $N_{1,1}$, $N_{2,1}$, X_1 - значениям накопителей N_1, N_2, X_j второго процессорного элемента и так далее до четвертого процессорного элемента с переменными $N_{1,3}$, $N_{2,3}$, X_3.

Основная особенность предлагаемого подхода для вычислительной платформы с SIMD-архитектурой заключается в обеспечении возможности выполнения цикла шифрования ГОСТ 28147-89 четырьмя процессорными элементами одновременно в отношении четырех наборов переменных $N_{1,i}$, $N_{2,i}$, X_i с использованием одних и тех же узлов замены K_m, $1 \le m \le 8$. При этом, компонент ключа в каждом наборе переменных может быть свой. Также следует отметить, что содержимое каждого из таких наборов никак не коррелирует с содержимым любого другого из них.

Выполнение цикла шифрования согласно настоящему изобретению предваряется следующими действиями.

Во-первых, константы M_1-M_6, выполняющие роль масок, задаются следующим образом:

$$M_1 = \text{загрузить}(0x000F000F);$$

$$M_2 = \text{загрузить}(0x0F000F00);$$

$$M_3 = \text{загрузить}(0xF000000F);$$

$$M_4 = \text{загрузить}(0x000FF000);$$

$$M_5 = \text{загрузить}(0x0FF00000);$$

$$M_6 = \text{загрузить}(0x00000FF0).$$

Выбор этих констант определяется особенностями используемой SIMD-архитектуры, а именно набором SIMD-инструкций (по сути, вопрос о выборе конкретных констант эквивалентен вопросу об использовании конкретных инструкций). Функциональное назначение этих констант объясняется ниже.

Во-вторых, формируются массивы T_i беззнаковых 4-разрядных чисел размерности 32, в которых размещаются предварительно заполненные узлы замены. Тогда в ячейках массивов T_i будут сохраняться следующие значения:

$$\begin{aligned} &T_0[2(n)] = K_1[n], T_0[2(n)+1] = K_8[n], \\ &T_1[2(n)] = K_5[n], T_1[2(n)+1] = K_4[n], \\ &T_2[2(n)] = K_3[n], T_2[2(n)+1] = K_2[n], \\ &T_3[2(n)] = K_7[n], T_3[2(n)+1] = K_6[n] \end{aligned}, \quad n = 0, \ldots, 15.$$

Например, $T_0 = (K_1[0], K_8[0], K_1[1], K_8[1], \ldots, K_1[15], K_8[15])$. Следует при этом отметить, что, как сказано ранее, ГОСТ 28147-89 не специфицирует конкретный набор констант для блока подстановки, что означает, что разработчик может по своему усмотрению выбрать его наполнение.

Далее со ссылкой на фиг. 1 непосредственно описывается выполнение одного цикла шифрования данных согласно ГОСТ 2814 7-89. Как было сказано выше, количество циклов шифрования определяется ГОСТ 28147-89 в зависимости от режима шифрования.

На входе цикла - три массива данных N_1, N_2 и X:

$$\begin{aligned} N_1 &= (N_{1,0}, N_{1,1}, N_{1,2}, N_{1,3}), \\ N_2 &= (N_{2,0}, N_{2,1}, N_{2,2}, N_{2,3}), \\ X &= (X_0, X_1, X_2, X_3). \end{aligned}$$

Пояснения, касающиеся 32-разрядных элементов этих трех массивов, приведены выше.

На его выходе формируется массив CM_2, состоящий из четырех 32-разрядных элементов, где $CM_2[i]$ - результат выполнения одного цикла шифрования для набора переменных $N_{1,i}$, $N_{2,i}$, X_i, i = 0, ..., 3.

На этапе 101 выполняется суммирование $N_{1,i}$ и X_i по модулю 2^{32}. Результат заносится в 32-разрядный элемент массива CM_1, т.е. $CM_1[i] = (N_{1,i} + X_i) \bmod 2^{32}$. На используемом здесь псевдоязыке этап 101 обобщенно представляется как

$$CM_1 = \text{сумма}(N_1, X)$$

Далее на этапе 102 выполняется отвечающая настоящему изобретению операция замены, которая подробно описывается ниже.

Затем на этапе 103 в отношении результата R замены выполняется циклический сдвиг

$$R = \text{или}(\text{сдвиг\_влево}(R, 11), \text{сдвиг\_вправо}(R, 21))$$

после чего на этапе 104 получают выходные зашифрованные данные в массиве CM_2 путем выполнения поразрядного суммирования по модулю 2 следующим образом

$$CM_2 = \text{искл\_или}(R, N_2)$$

Следует отметить, что операции по этапам 101, 103, 104 соответствуют спецификации ГОСТ 28147-89 без какой-либо SIMD-ориентированной адаптации. Иными словами, эти операции напрямую проецируются на SIMD-инструкции.

Как было сказано выше, суть решаемой настоящим изобретением задачи заключается в том, что операция замены, как она специфицирована в ГОСТ 28147-89, не может быть напрямую спроецирована на SIMD-инструкции, подобно вышеуказанным операциям по этапам 101, 103, 104, и, следовательно, необходима соответствующая ее адаптация с учетом специфики SIMD-архитектуры, которая описывается ниже со ссылкой на фиг. 2.

На вход операции замены по этапу 102 подается массив CM_1, полученный на этапе 101. Результат данной операции помещается в переменную R. При выполнении операции замены по сути осуществляется замена $CM_1[i]$ в соответствии с содержимым узлов замены K_m.

На этапе 201 при помощи масок M_1, M_2 (см. выше) из массива CM_1 формируются два вспомогательных массива Y и Z следующим образом:

$$Y' = \text{сдвиг\_вправо}(CM_1, 4),$$

$$Y = \text{или}(\text{и}(M_1, CM_1), \text{и}(M_2, Y')),$$

$$Z = \text{или}(\text{и}(M_1, Y'), \text{и}(M_2, CM_1)).$$

Далее, каждый 32-разрядный элемент массивов CM_1, Y и Z будет естественным образом рассматриваться как массив 4-разрядных элементов размерности восемь. В результате такой интерпретации массивы CM_1, Y и Z будут иметь размерность 32. Предположим, что массив CM_1 задан в виде последовательности

$$L_{1,1}, L_{1,2}, \ldots, L_{1,8}, L_{2,1}, L_{2,2}, \ldots, L_{2,8}, \ldots, L_{4,1}, L_{4,2}, \ldots, L_{4,8}.$$

Тогда

$$Y = (0, L_{1,1}, 0, L_{1,4}, 0, L_{1,5}, 0, L_{1,8}, 0, L_{2,1}, 0, L_{2,4}, 0, L_{2,5}, 0, L_{2,8}, \ldots, 0, L_{4,1}, 0, L_{4,4}, 0, L_{4,5}, 0, L_{4,8}) \quad \text{и}$$

$$Z = (0, L_{1,2}, 0, L_{1,3}, 0, L_{1,6}, 0, L_{1,7}, 0, L_{2,2}, 0, L_{2,3}, 0, L_{2,6}, 0, L_{2,7}, \ldots, 0, L_{4,2}, 0, L_{4,3}, 0, L_{4,6}, 0, L_{4,7}).$$

Затем на этапе 202 в отношении вспомогательных массивов Y и Z выполняется замена при помощи операции "перемешать" в соответствии с представлением блока подстановки T_i, i = 0, ..., 3 (см. выше), с получением промежуточных массивов данных R_0-R_3:

$$R_0 = \text{и}(M_3, \text{перемешать}(T_0, Y)),$$

$$R_1 = \text{и}(M_5, \text{перемешать}(T_2, Z)),$$

$$R_2 = \text{и}(M_4, \text{перемешать}(T_1, Y)),$$

$$R_3 = \text{и}(M_6, \text{перемешать}(T_3, Z)).$$

При этом получаемые в результате операции "перемешать" массивы данных, в силу специфики этой операции, содержат как значимые с точки зрения конечного результата данные, так и незначимые. Для дальнейших вычислений выполняется фильтрация данных из этих массивов данных путем выделения значимых данных при помощи масок M_3-M_6.

При этом следует отметить следующее. Хотя в вышеприведенном описании набора процедур сказано, что на вход оператора "перемешать" подаются массивы 8-разрядных целых чисел размерности 16, в то время как в описании по фиг. 2 на его вход подаются массивы 4-разрядных целых чисел размерности 32, никакого противоречия нет, поскольку отвечающий SIMD-архитектуре оператор "перемешать" оперирует входными массивами как 128-разрядными переменными, в то время как задействуемая здесь конкретная их разбивка используется для удобства иллюстрирования того или иного аспекта.

Рассмотрим подробнее операцию замены на примере T_0 и Y. При выполнении операции будут получены $T_0[Y[k]]$, $0 \le k \le 15$. Однако значимыми результатами операции являются значения старших четырех разрядов для k=0,4,8,12 и младших четырех разрядов для k=3,7,11,15, значения же остальных разрядов следует игнорировать в силу их бесполезности. Запишем промежуточный результат в виде развернутого массива $R_{i,m} = K_m[L_{i,m}]$. При помощи x пометим значения, которые не являются значимыми. В результате получим

$$R_{0,1}, x, x, x, x, x, x, R_{0,8}, R_{1,1}, x, x, x, x, x, x, R_{1,8}, \ldots, R_{3,1}, x, x, x, x, x, x, R_{3,8}$$

Для того чтобы значения на позициях с пометкой x не влияли на результаты вычислений, в эти позиции следует записать нулевое значение. Это и делается при помощи маски M_3.

Замена посредством остальных узлов выполняется аналогичным образом.

Наконец, на этапе 203 объединением полученных ранее промежуточных массивов R_0-R_3 получают выдаваемый на этап 103 результат операции замены в виде массива R следующим образом:

$$R = \text{или}(\text{или}(R_0, R_2), \text{или}(R_1, R_3))$$

Для иллюстрации приведем вычисленные значения на этапах 101, 102 (т.е. 201-203):

$$CM_1 = (L_{1,1}, L_{1,2}, \ldots, L_{1,8}, L_{2,1}, L_{2,2}, \ldots, L_{2,8}, \ldots, L_{4,1}, L_{4,2}, \ldots, L_{4,8}),$$

$$Y = (0, L_{1,1}, 0, L_{1,4}, 0, L_{1,5}, 0, L_{1,8}, 0, L_{2,1}, 0, L_{2,4}, 0, L_{2,5}, 0, L_{2,8}, \ldots, 0, L_{4,1}, 0, L_{4,4}, 0, L_{4,5}, 0, L_{4,8}),$$

$$Z = (0, L_{1,2}, 0, L_{1,3}, 0, L_{1,6}, 0, L_{1,7}, 0, L_{2,2}, 0, L_{2,3}, 0, L_{2,6}, 0, L_{2,7}, \ldots, 0, L_{4,2}, 0, L_{4,3}, 0, L_{4,6}, 0, L_{4,7}),$$

$$R_0 = (R_{0,1}, 0, 0, 0, 0, 0, 0, R_{0,8}, R_{1,1}, 0, 0, 0, 0, 0, 0, R_{1,8}, R_{2,1}, 0, 0, 0, 0, 0, 0, R_{2,8}, R_{3,1}, 0, 0, 0, 0, 0, 0, R_{3,8}),$$

$$R_1 = (0, R_{0,2}, R_{0,3}, 0, 0, 0, 0, 0, 0, R_{1,2}, R_{1,3}, 0, 0, 0, 0, 0, 0, R_{2,2}, R_{2,3}, 0, 0, 0, 0, 0, 0, R_{3,2}, R_{3,3}, 0, 0, 0, 0, 0,),$$

$$R_2 = (0, 0, 0, R_{0,4}, R_{0,5}, 0, 0, 0, 0, 0, 0, R_{1,4}, R_{1,5}, 0, 0, 0, 0, 0, 0, R_{2,4}, R_{2,5}, 0, 0, 0, 0, 0, 0, R_{3,4}, R_{3,5}, 0, 0, 0),$$

$$R_3 = (0, 0, 0, 0, 0, R_{0,6}, R_{0,7}, 0, 0, 0, 0, 0, 0, R_{1,6}, R_{1,7}, 0, 0, 0, 0, 0, 0, R_{2,6}, R_{2,7}, 0, 0, 0, 0, 0, 0, R_{3,6}, R_{3,7}, 0),$$

$$R = (R_{0,1}, R_{0,2}, \ldots, R_{0,8}, R_{1,1}, R_{1,2}, \ldots, R_{1,8}, R_{2,1}, R_{2,2}, \ldots, R_{2,8}, R_{3,1}, R_{3,2}, \ldots, R_{3,8}).$$

Для практической реализации описанного цикла шифрования ГОСТ 28147-89 необходима компьютерная система с SIMD-архитектурой, например компьютерная система с процессором производства компаний Intel или AMD с набором инструкций SSE-технологии. В частности, вышеизложенный подход может быть реализован в виде программного обеспечения, сохраненного на известном машиночитаемом носителе(ях) из состава такой компьютерной системы, которое при его исполнении упомянутым процессором будет предписывать ему осуществлять шифрование данных согласно настоящему изобретению. Такое программное обеспечение может быть создано путем компиляции исходного кода, представляющего описанные выше этапы, операции и действия, а также переменные и константы, на соответствующем языке программирования (например, на языке программирования C++) с помощью компилятора, поддерживающего инструкции SSE-технологии, такого как, например, компилятор Visual C++ 2008 компании Microsoft или компилятор компании Intel.

Изобретение было раскрыто выше со ссылкой на конкретный вариант его осуществления. Для специалистов могут быть очевидны и иные варианты осуществления изобретения, не меняющие его сущности, как она раскрыта в настоящем описании. Соответственно, изобретение следует считать ограниченным по объему только нижеследующей формулой изобретения.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ шифрования данных, реализуемый посредством вычислительной платформы, имеющей SIMD-архитектуру, содержащий этапы, на которых

инициализируют маски M_k (k=1, ..., 6), причем каждая маска представляет собой массив размерности 4, каждый элемент которого представляет собой уникальную для маски 32-разрядную константу;

реорганизуют оговоренные в ГОСТ 28147-89 предварительно заполненные узлы замены K_m (m=1, ..., 8), каждый из которых представляет собой массив 4-разрядных чисел размерности 16, в массивы T_j (j=0, ..., 3) 4-разрядных чисел размерности 32 следующим образом:

$$\left.\begin{array}{l} T_0[2(n)] = K_1[n], T_0[2(n)+1] = K_8[n], \\ T_1[2(n)] = K_5[n], T_1[2(n)+1] = K_4[n], \\ T_2[2(n)] = K_3[n], T_2[2(n)+1] = K_2[n], \\ T_3[2(n)] = K_7[n], T_3[2(n)+1] = K_6[n] \end{array}\right\}, \quad n = 0, \quad , 15; \; n$$

выполняют по меньшей мере один цикл шифрования, в котором формируют массивы 32-разрядных элементов размерности 4

$$N_1 = \{N_{1,i}\}, \; N_2 = \{N_{2,i}\}, \; X = \{X_i\}, \; i = 0, \quad , 3,$$

где $N_{1,i}$, $N_{2,i}$ - подблоки, на которые предварительно разбит 64-разрядный блок шифруемых данных, а X_i - часть 256-разрядного ключа шифрования, выбираемая в соответствии с ГОСТ 28147-89;

формируют массив CM_1 путем выполнения операции суммирования по модулю 2^{32} в отношении массивов N_1 и X;

формируют первый вспомогательный массив Y_0 путем применения к массиву CM_1 операции поразрядного сдвига вправо на четыре разряда, формируют второй вспомогательный массив Y путем применения операции логического ИЛИ к результату операции логического И в отношении маски M_1 и массива CM_1 и результату операции логического И в отношении маски M_2 и первого вспомогательного массива Y_0 и формируют третий вспомогательный массив Z путем применения операции логического ИЛИ к результату операции логического И в отношении маски M_1 и первого вспомогательного массива Y_0 и результату операции логического И в отношении маски M_2 и массива CM_1;

формируют первый промежуточный массив R_0 путем применения операции логического И к маске M_3 и результату выполнения операции перемешивания в отношении массива T_0 и второго вспомогательного массива Y, формируют второй промежуточный массив R_1 путем применения операции логического И к маске M_5 и результату выполнения операции перемешивания в отношении массива T_2 и третьего вспомогательного массива Z, формируют третий промежуточный массив R_2 путем применения операции логического И к маске M_4 и результату выполнения операции перемешивания в отношении массива T_1 и второго вспомогательного массива Y, формируют четвертый промежуточный массив R_3 путем применения операции логического И к маске M_6 и результату выполнения операции перемешивания в отношении

массива T_3 и третьего вспомогательного массива Z и формируют пятый промежуточный массив R путем применения операции логического ИЛИ к результату операции логического ИЛИ в отношении первого промежуточного массива R_0 и третьего промежуточного массива R_2 и результату операции логического ИЛИ в отношении второго промежуточного массива R_1 и четвертого промежуточного массива R_3;

выполняют в отношении пятого промежуточного массива R циклический сдвиг и

формируют представляющий шифрованные данные массив CM_2 путем выполнения операции поразрядного суммирования по модулю 2 в отношении пятого промежуточного массива, прошедшего циклический сдвиг, и массива N_2;

при этом вышеупомянутые суммирования, логические операции, сдвиги и операция перемешивания непосредственно предусмотрены SIMD-архитектурой.

2. Способ по п.1, в котором SIMD-архитектура является SIMD-архитектурой компании Intel или аналогичной архитектурой компании AMD, образуемой процессором, имеющим четыре процессорных элемента.

3. Способ по п.2, в котором с каждым из процессорных элементов связано отдельное локальное запоминающее устройство, в котором сохраняется соответственно один из наборов $N_{1,i}$, $N_{2,i}$, X_i (i = 0, ..., 3) для независимой обработки этим процессорным элементом.

4. Способ по п.3, в котором узлы замены K_m (m=1, ..., 8) в совокупности образуют блок подстановки, который является единым для всех процессорных элементов.

5. Способ по п.1, в котором количество циклов шифрования определяется ГОСТ 28147-89.

6. Способ по п.1, в котором выбор части ключа шифрования определяется режимом шифрования, предусмотренным ГОСТ 28147-89.

7. Способ по п.1, в котором при формировании массива CM_1 путем выполнения операции суммирования по модулю 2^{32} в отношении массивов N_1 и X i-й элемент массива CM_1 получают суммированием $N_{1,i}$ и X_i по модулю 2^{32}, i=0, ..., 3.

8. Способ по п.1, в котором перед формированием промежуточных массивов каждый из вспомогательных массивов Y и Z реорганизуют как массив 4-разрядных чисел размерности 32 путем реорганизации каждого его 32-разрядного элемента в массив 4-рязрядных чисел размерности 8.

9. Способ по п.8, в котором при операции перемешивания, выполняемой в отношении двух массивов, результирующий массив заполняется элементами первого из этих двух массивов так, что в элемент результирующего массива, позиция которого соответствует позиции элемента второго из упомянутых двух массивов, записывается либо 0, либо элемент первого из упомянутых двух массивов, позиция которого указывается младшими четырьмя разрядами упомянутого элемента второго из этих двух массивов.

10. Способ по п.9, в котором применением операции логического И к соответствующей маске и соответствующему результату операции перемешивания обнуляются незначащие значения в получающемся промежуточном массиве.

11. Способ по п.1, в котором циклический сдвиг в отношении пятого промежуточного массива R выполняют путем применения операции логического ИЛИ к результату операции поразрядного сдвига влево на одиннадцать разрядов в отношении массива R и результату операции поразрядного сдвига вправо на двадцать один разряд в отношении массива R.

12. Способ по п.7 или 11, в котором при применении к массиву 32-разрядных чисел размерности 4 операции поразрядного сдвига влево/вправо операцию поразрядного сдвига влево/вправо выполняют в отношении каждого элемента данного массива.

13. Способ по п.1, в котором при формировании массива CM_2 путем выполнения операции поразрядного суммирования по модулю 2 в отношении массивов R и N_2 i-й элемент массива CM_2 получают поразрядным суммированием 1-го элемента массива R и $N_{2,i}$ по модулю 2, i=0, ..., 3.

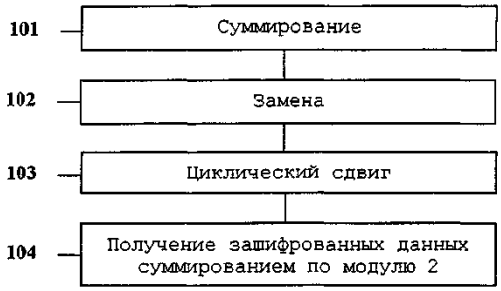
14. Способ по любому из пп.1-11, в котором при применении к массивам 32-разрядных чисел размерности 4 операции логического ИЛИ i-й элемент результирующего массива получается соответственным применением операции логического ИЛИ в отношении i-х элементов упомянутых массивов, i=0, ..., 3.

15. Способ по любому из пп.1-11, в котором при применении к массивам 32-разрядных чисел размерности 4 операции логического И i-й элемент результирующего массива получается соответственным применением операции логического И в отношении i-х элементов упомянутых массивов, i=0, ..., 3.

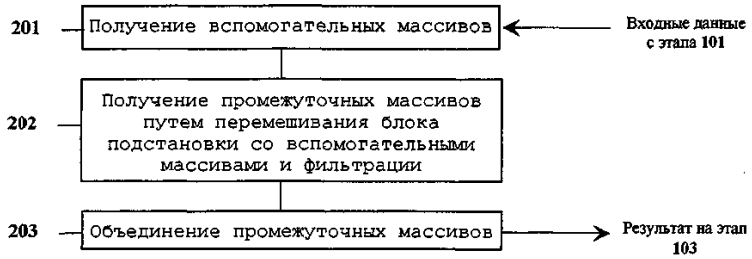
16. Способ по п.1, в котором маски M_k (k=1, ..., 6) инициализируют следующими значениями:

$M_1[i]$ = 0x000F000F, $M_2[i]$ = 0x0F000F00, $M_3[i]$ = 0xF000000F, $M_4[i]$ = 0x000FF000, $M_5[i]$ = 0x000FF000, $M_5[i]$ = 0x0FF00000, $M_6[i]$ = 0x00000FF0, i = 0, , 3.

17. Машиночитаемый носитель, на котором сохранены машиноисполняемые команды, которые при их исполнении вычислительной платформой, имеющей SIMD-архитектуру, предписывают этой вычислительной платформе выполнять способ шифрования данных по п.1.



Фиг. 1



Фиг. 2

Евразийская патентная организация, ЕАПВ
Россия, 109012, Москва, Малый Черкасский пер., 2