ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

DOI 10.17223/20710410/13/9

УДК 519.6

О РЕАЛИЗАЦИИ МЕТОДА СОГЛАСОВАНИЯ В КРИПТОАНАЛИЗЕ С ПОМОЩЬЮ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ[1]

В. М. Фомичев

Институт проблем информатики РАН, г. Москва, Россия

E-mail: fomichev@nm.ru

Оценено время реализации метода согласования применительно к анализу итеративных симметричных блочных шифров с использованием кластерных и распределенных вычислений. Показано, что по сравнению с однопроцессорной системой коэффициент сокращения времени может достигать числа используемых процессоров.

Ключевые слова: *метод согласования, кластер, распределенные вычисления.*

Введение

Метод согласования [1], или атака «встреча посередине» (meet-in-the-middle attack) [2 – 5], применяется для определения ключа шифра, как правило, по известным открытому и шифрованному текстам. Он менее трудоемок по сравнению с полным опробованием ключей, если функция шифрования $E(q, x)$ открытого текста x по ключу $q \in V_n = \{0, 1\}^n$ допускает декомпозицию на две функции как $E(q, x) = g(q, g'(q, x))$, где для множеств существенных ключевых переменных K и K' соответственно функций g и g' выполнено $K \setminus K' \neq \varnothing$ и $K' \setminus K \neq \varnothing$. Наибольший эффект от применения метода достигается, если множества K и K' равномощны и $K \cap K' = \varnothing$. При этом опробование ключа выполняется как независимое опробование переменных из множеств K и K' и ключ q определяется с вычислительной сложностью порядка $O(2^{n/2})$ операций типа зашифрования-расшифрования при использовании памяти, достаточной для хранения порядка $O(2^{n/2})$ ключей.

Параллельные вычисления с использованием N процессоров позволяют сократить в N раз время решения некоторых задач, например полного опробования ключей. Вместе с тем применение параллельных вычислений для решения других задач не столь эффективно. Оценим эффективность различных моделей параллельных вычислений для реализации метода согласования в решении следующей задачи.

Для r-раундового симметричного блочного шифра требуется вычислить n-битовый ключ q по известным t-битовым блокам x и y открытого и шифрованного текстов, где $t \geqslant n$ и ключ q по блокам x и y определяется однозначно, при следующих предположениях:

[1]Работа выполнена в рамках мероприятия 1.2.1 ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. по направлению «Распределенные вычислительные системы».

1) ключ q есть конкатенация независимых ключей: $q = v \cdot w$, где $v \in V_m$, $w \in V_{n-m}$ и $m \leqslant n/2$;
2) функции шифрования первых $l < r$ раундов и остальных $r - l$ раундов шифрования суть подстановки соответственно g_v и z_w, определяемые бинарными ключами v и w.

Тогда зашифрование блока x в блок y с помощью подстановки E_q имеет вид

$$y = E_q(x) = g_v z_w(x) = z_w(g_v(x)). \tag{1}$$

Корректность предложенных ниже алгоритмов следует из (1).

Модель вычислительной системы предполагает использование N идентичных вычислителей с неограниченной памятью (размер памяти далее уточняется), где у вычислителей одинаковы производительность вычислений, скорости чтения/записи данных и др. Различаются два случая: кластерные вычисления (КВ) и распределенные вычисления (РВ). Преимущество модели КВ заключается в возможности достаточно активного обмена данными между вычислителями. Преимущество модели РВ, где координатор распределяет задания между процессорами — участниками вычислений и объединяет результаты вычислений, в том, что число участников РВ может заметно превышать число процессоров в кластере. При этом участник обменивается данными только с координатором.

1. Кластерные вычисления

Пусть кластерная система имеет 2^k вычислителей, снабженных блоками памяти, $k \leqslant m$. Каждый вычислитель имеет номер, являющийся его адресом (число от 0 до $2^k - 1$, или в двоичной записи — вектор из V_k). Между блоками памяти вычислителей может выполняться интенсивный обмен данными. Для реализации алгоритма каждый вычислитель использует адресную память размера 2^{t-k} ячеек, в ячейке могут быть записаны несколько вариантов ключей, то есть элементов V_n. Адреса ячеек суть элементы V_{t-k}, являющиеся значениями некоторой хеш-функции: $V_t \to V_{t-k}$. Хеш-функция может быть весьма простой, например выделение первых $t - k$ битов двоичного t-битового вектора.

Для любого двоичного вектора $(\alpha_1, \alpha_2, \ldots)$ размерности больше k обозначим

$$\delta(\alpha_1, \alpha_2, \ldots) = (\alpha_1, \ldots, \alpha_k), \quad \bar{\delta}(\alpha_1, \alpha_2, \ldots) = (\alpha_{k+1}, \alpha_{k+2}, \ldots).$$

Для вектора $\alpha = (\alpha_1, \ldots, \alpha_k) \in V_k$ и пространства векторов V_s, где $s \geqslant k$, обозначим

$$V_s(\alpha) = \{\xi \in V_s : \delta(\xi) = \alpha\}.$$

Алгоритм состоит из предварительного и оперативного этапов.

Предварительный этап (*заполнение блоков памяти вычислителей*)

Вычислитель с номером $\alpha \in V_k$ последовательно опробует ключи v из $V_m(\alpha)$ и вычисляет $g_v(x)$ для блока x. Затем пара $(v, g_v(x))$ направляется вычислителю с номером $\delta(g_v(x))$, который записывает ключ v в свою память по адресу $\bar{\delta}(g_v(x))$.

По завершении этапа множество ключей из V_m распределено по ячейкам памяти всех вычислителей. Обозначим через $Q(\alpha, \beta)$ множество ключей из V_m, записанных в памяти вычислителя с номером α по адресу β.

Оперативный этап (*определение ключа*)

1) Вычислитель с номером $\alpha \in V_k$ последовательно опробует ключи w из $V_{n-m}(\alpha)$ и вычисляет $(z_w)^{-1}(y)$ для блока y, затем пара $(w, (z_w^{-1})(y))$ направляется вычислителю с номером $\delta((z_w^{-1})(y))$.

2) Вычислитель с номером $\delta((z_w^{-1})(y))$ обращается в свою память по адресу $\bar{\delta}((z_w^{-1})(y))$. Конкатенация $v \cdot w$ для каждого ключа v из множества $Q = Q(\delta((z_w^{-1})(y)), \bar{\delta}((z_w^{-1})(y)))$ есть кандидат на значение искомого ключа $q = v \cdot w$. Если $Q \neq \varnothing$, то выполняется отбраковка всех ключей вида $v \cdot w$ (например, по критерию соответствия известным парам открытого и шифрованного текстов).

Характеристики метода

Оценим (в предположении, что ключ q выбирается случайно равновероятно из множества V_n) среднее время $T(m)$ описанной реализации метода согласования через время реализации операций зашифрования, расшифрования, пересылки и обращения в память, обозначаемых соответственно $\tau_з$, $\tau_р$, $\tau_п$, $\tau_о$. Положим, что работа алгоритма происходит в дискретные моменты времени (такты) и в каждый такт на первом этапе в любую ячейку памяти записывается не более одного варианта ключа v, на втором этапе из любой ячейки памяти извлекается не более одного варианта ключа v, т. е. замедления «из-за очередей» в работе вычислителей не происходит.

На первом этапе для ключа v из $V_m(\alpha)$ реализуется однократное зашифрование, пересылка и запись в память, отсюда среднее время $T_1(m)$ работы вычислителя равно $2^{m-k}(\tau_з + \tau_п + \tau_о)$. На втором этапе для ключа w из $V_{n-m}(\alpha)$ реализуется однократное расшифрование, пересылка и обращение в память. Следовательно, среднее время $T_2(m)$ выполнения вычислителем второго этапа равно $2^{n-m-k}(\tau_р + \tau_п + \tau_о) + T_{бр}$, где $T_{бр}$ — среднее время отбраковки кандидатов на значение ключа.

Оценим величину $T_{бр}$. В каждой ячейке памяти записано в среднем 2^{m-t} вариантов ключа v. Среднее число обращений в любую ячейку памяти на втором этапе равно 2^{n-m-t}. Отсюда каждым вычислителем отбраковывается в среднем 2^{n-t-k} кандидатов на значение ключа. Значит, $T_{бр} = 2^{n-t-k}\tau_з$, и отбраковка кандидатов на значение ключа вносит несущественный вклад в общую трудоемкость. Следовательно,

$$T(m) = T_1(m) + T_2(m) \approx 2^{m-k}(\tau_з + \tau_п + \tau_о) + 2^{n-m-k}(\tau_р + \tau_п + \tau_о).$$

Отсюда, если $\tau_з \approx \tau_р$, то минимум трудоёмкости $T(m)$ достигается при $m = \lfloor n/2 \rfloor$:

$$T = T(\lfloor n/2 \rfloor) \approx 2^{n/2-k+1}(\tau_з + \tau_п + \tau_о). \tag{2}$$

Следовательно, среднее время T оценивается величиной порядка $O(\tau 2^{n/2-k})$, где $\tau = \max\{\tau_з, \tau_п, \tau_о\}$.

Надёжность метода равна 1. В связи с минимизацией по m трудоемкости $T(m)$ уточним размер требуемой памяти: вычислителю достаточно иметь $2^{n/2-k}$ ячеек, в которые записываются элементы $V_{n/2}$. Адресами ячеек являются элементы $V_{n/2-k}$.

При выборе оптимального (по времени реализации алгоритма) размера памяти следует учесть, что реализация алгоритма может замедляться «из-за очередей», когда в одну ячейку одновременно поступает несколько запросов в связи с необходимостью записи или извлечения информации. Это замедление тем несущественней, чем меньше соотношение $\tau_о/\tau_з$.

Таким образом, время определения ключа блочного шифра методом согласования с использованием кластерных вычислений с числом процессоров 2^k может быть сокращено до 2^k раз по сравнению с однопроцессорной вычислительной системой, если время пересылки данных между вычислителями не слишком велико. Важно также, что совокупный объем требуемой памяти $2^{n/2}$ ячеек также распределяется между 2^k процессорами.

2. Распределенные вычисления

В системе РВ с 2^p участниками (вычислителями), $p \leqslant m$, каждый участник имеет номер, являющийся его адресом (число от 0 до 2^{p-1}, или в двоичной записи — вектор из V_p). Алгоритм использует 2^t ячеек адресной памяти координатора (адрес ячейки есть элемент V_t), в каждую из них могут быть записаны несколько вариантов ключей — элементов V_n. Участники могут отправлять данные координатору, но не могут обращаться к его памяти.

Алгоритм состоит из предварительного и оперативного этапов.

Предварительный этап (*заполнение памяти координатора*)

Вычислитель с номером $\alpha \in V_p$ последовательно при каждом ключе v из $V_m(\alpha)$ вычисляет $g_v(x)$ для блока x и направляет пару $(v, g_v(x))$ координатору, где ключ v записывается в память координатора по адресу $g_v(x)$. По завершении этапа множество ключей из V_m распределено по ячейкам памяти координатора. Обозначим через $Q(\beta)$ множество ключей из V_m, записанных в памяти координатора по адресу β.

Оперативный этап (*определение ключа*)

1) Вычислитель с номером $\alpha \in V_p$ последовательно при каждом ключе w из $V_{n-m}(\alpha)$ вычисляет $(z_w^{-1})(y)$ для блока y и направляет пару $(w, (z_w^{-1})(y))$ координатору.
2) Координатор обращается в память по адресу $(z_w^{-1})(y)$. Конкатенация каждого ключа v из $Q((z_w^{-1})(y))$ с ключом w есть кандидат на значение искомого ключа $q = v{\cdot}w$. Если $Q((z_w^{-1})(y)) \neq \varnothing$, то координатор подвергает отбраковке все ключи вида $v \cdot w$ (например, по критерию соответствия известным парам открытого и шифрованного текстов).

Характеристики метода

Оценим (ключ q выбирается случайно равновероятно из V_n) среднее время $T(m)$ описанной реализации метода согласования через время реализации операций зашифрования, расшифрования, пересылки и обращения в память. Положим, что в каждый такт на первом этапе в любую ячейку памяти записывается не более одного варианта ключа v, на втором этапе из любой ячейки памяти извлекается не более одного варианта ключа v, т. е. замедления «из-за очередей» в работе вычислителей не происходит.

Среднее время $T_{1_\text{у}}(m)$ выполнения первого этапа участниками равно $2^{m-p}(\tau_\text{з} + \tau_\text{п})$, так как для каждого ключа v из $V_m(\alpha)$ реализуется по одной операции зашифрования и пересылки. Среднее время $T_{1_\text{к}}(m)$ выполнения первого этапа координатором (записи в память) равно $2^m\tau_\text{о}$.

Следовательно, среднее время $T_1(m)$ выполнения первого этапа равно

$$T_1(m) = T_{1_\text{у}}(m) + T_{1_\text{к}}(m) = 2^{m-p}(\tau_\text{з} + \tau_\text{п} + 2^p\tau_\text{о}).$$

Для ключа w из $V_{n-m}(\alpha)$ участником реализуется по одной операции расшифрования и пересылки. Следовательно, среднее время $T_{2_\text{у}}(m)$ выполнения второго этапа каждым участником равно $2^{n-m-p}(\tau_\text{р} + \tau_\text{п})$. Среднее время $T_{2_\text{к}}(m)$ выполнения второго этапа координатором (записи в память и отбраковки) равно $2^{n-m}\tau_\text{о} + T_\text{бр}$, где $T_\text{бр}$ — среднее время отбраковки кандидатов на значение ключа. Отсюда среднее время $T_2(m)$ выполнения второго этапа равно $2^{n-m-p}(\tau_\text{р} + \tau_\text{п} + 2^p\tau_\text{о}) + T_\text{бр}$.

Оценим величину $T_\text{бр}$. В каждой ячейке памяти записано в среднем 2^{m-t} вариантов ключа v. Среднее число обращений в память на втором этапе равно 2^{n-m}. Отсюда координатором отбраковывается в среднем 2^{n-t} кандидатов на значение ключа. Значит, $T_\text{бр} = 2^{n-t}\tau_\text{з}$, и отбраковка кандидатов в ключи вносит несущественный вклад в общую трудоемкость. Следовательно,

$$T(m) = T_1(m) + T_2(m) \approx 2^{m-p}(\tau_{\text{з}} + \tau_{\text{п}} + 2^p\tau_{\text{о}}) + 2^{n-m-p}(\tau_{\text{р}} + \tau_{\text{п}} + 2^p\tau_{\text{о}}).$$

Отсюда, если $\tau_{\text{з}} \approx \tau_{\text{р}}$, то минимум среднего времени $T(m)$ достигается при $m = \lfloor n/2 \rfloor$:

$$T = T(\lfloor n/2 \rfloor) \approx 2^{n/2-p}(\tau_{\text{з}} + \tau_{\text{п}} + 2^p\tau_{\text{о}}). \tag{3}$$

В связи с минимизацией $T(m)$ по m уточним размер требуемой памяти: координатору достаточно иметь $2^{n/2}$ ячеек, в которые записываются элементы $V_{n/2}$. Адресами ячеек являются элементы $V_{n/2}$. Следовательно, среднее время работы алгоритма согласования по сравнению с полным опробованием ключей сокращается не более чем в 2^p раз, и сокращение зависит от соотношения величин $\tau_{\text{о}}$ и $\max\{\tau_{\text{з}}, \tau_{\text{п}}\}$. Надёжность метода равна 1.

3. Комбинирование кластерных и распределенных вычислений

В данной модели РВ система использует 2^p участников, $p \leqslant m$. Каждый участник имеет номер, являющийся его адресом (число от 0 до 2^{p-1}, или в двоичной записи — вектор из V_p). Координатор располагает кластерной подсистемой 2^k вычислителей, $k \leqslant p$, каждый вычислитель кластерной системы имеет номер, являющийся его адресом (число от 0 до $2^k - 1$, или в двоичной записи — вектор из V_k), и имеет блок памяти размера 2^{t-k} ячеек (адрес ячейки есть элемент V_{t-k}). В каждую ячейку могут быть записаны несколько вариантов ключей — элементов V_n. Участники РВ могут отправлять данные кластерным вычислителям, но не могут обращаться в память кластерных вычислителей.

Предварительный этап (*заполнение памяти координатора*)

Участник с номером $\alpha \in V_p$ последовательно при каждом ключе v из $V_m(\alpha)$ вычисляет $g_v(x)$ для блока x и направляет пару $(v, g_v(x))$ кластерному вычислителю с номером $\delta(g_v(x))$, который записывает в память ключ v по адресу $\bar{\delta}(g_v(x))$. По завершении этапа множество ключей из V_m распределено по блокам памяти кластерных вычислителей координатора. Обозначим через $Q(\alpha, \beta)$ множество ключей из V_m, записанных в блоке памяти вычислителя с номером α по адресу β.

Оперативный этап (*определение ключа*)

1) Вычислитель с номером $\alpha \in V_p$ последовательно при каждом ключе w из $V_{n-m}(\alpha)$ вычисляет $(z_w^{-1})(y)$ для блока y и направляет пару $(w, (z_w^{-1})(y))$ кластерному вычислителю с номером $\delta((z_w^{-1})(y))$.
2) Кластерный вычислитель с номером $\delta((z_w^{-1})(y))$ вычисляет адрес $\bar{\delta}((z_w^{-1})(y))$ и обращается к своему блоку памяти по этому адресу. Конкатенация вида $v \cdot w$ для каждого ключа v из множества $Q = Q(\delta((z_w^{-1})(y)), \bar{\delta}((z_w^{-1})(y))$ есть кандидат на значение ключа. Если $Q \neq \varnothing$, то вычислитель с номером $\delta((z_w^{-1})(y))$ все ключи вида $v \cdot w$ подвергает отбраковке (например, по критерию соответствия известным парам открытого и шифрованного текстов).

Характеристики метода

Оценим (ключ q выбирается случайно равновероятно из V_n) среднее время $T(m)$ описанной реализации метода согласования через время реализации операций зашифрования, расшифрования, пересылки и обращения в память.

Среднее время $T_{1_{\text{у}}}(m)$ выполнения первого этапа участником равно $2^{m-p}(\tau_{\text{з}} + \tau_{\text{п}})$, так как для каждого ключа v из $V_m(\alpha)$ реализуется по одной операции зашифрования и пересылки. Среднее время $T_{1_{\text{к}}}(m)$ выполнения первого этапа кластерным вычислителем оценивается величиной $2^{m-k}\tau_{\text{о}}$, так как запись в память 2^m ключей v выполняется

2^k вычислителями. Отсюда среднее время выполнения первого этапа алгоритма определяется величиной $\max\{2^{m-p}(\tau_з + \tau_п), 2^{m-k}\tau_о\}$.

Для ключа w из $V_{n-m}(\alpha)$ участником реализуется одно расшифрование и пересылка. Следовательно, среднее время $T_{2_у}(m)$ выполнения второго этапа участником равно $2^{n-m-p}(\tau_р + \tau_п)$. Среднее время $T_{2_к}(m)$ выполнения второго этапа кластерным вычислителем оценивается величиной $2^{n-m-k}\tau_о + T_{бр}$, так как запись в память 2^{n-m} ключей w выполняется 2^k вычислителями. Среднее время $T_{бр}$ отбраковки кандидатов на значение искомого ключа вносит несущественный вклад в общую трудоемкость. Отсюда среднее время выполнения второго этапа определяется величиной $\max\{2^{n-m-p}(\tau_р + \tau_п), 2^{n-m-k}\tau_о\}$. Следовательно, время $T(m)$ определяется величиной порядка

$$\max\{2^{m-p}(\tau_з + \tau_п), 2^{n-m-p}(\tau_р + \tau_п), 2^{m-k}\tau_о, 2^{n-m-k}\tau_о\}. \tag{4}$$

Тогда минимум $T(m)$ достигается при $m = \lfloor n/2 \rfloor$ и при $\tau_з = \tau_р$ верны оценки

$$O((\tau_з + \tau_п)2^{n/2-p}) \leqslant \min T(m) \leqslant O(\tau_о 2^{n/2-k}).$$

Следовательно, $\min T(m)$ может быть сокращен в несколько раз по сравнению с КВ и РВ (ср. с формулами (2), (3)). Коэффициент сокращения определяется соотношением скоростей шифрования, пересылки данных и обращения к памяти. Надёжность метода равна 1.

Уточним размер требуемой памяти: кластерному вычислителю достаточно иметь $2^{n/2-k}$ ячеек, в которые записываются элементы $V_{n/2}$. Адресами ячеек являются элементы $V_{n/2-k}$.

Выводы

Время определения ключа блочного шифра методом согласования может быть существенно сокращено по сравнению с однопроцессорной вычислительной системой:

1) при использовании КВ с числом процессоров 2^k — примерно в 2^k раз;
2) при использовании РВ с 2^p участниками — до 2^p раз, сокращение определяется соотношением скоростей шифрования, пересылки данных и обращения к памяти координатора;
3) при использовании РВ с 2^p участниками и подсистемы КВ координатора с числом процессоров 2^k, где $k \leqslant p$ — от 2^k до 2^p раз, сокращение определяется соотношением скоростей шифрования, пересылки данных и обращения к памяти кластерных вычислителей.

При использовании КВ память распределяется по вычислителям кластерной системы.

ЛИТЕРАТУРА

1. *Фомичёв В. М.* Методы дискретной математики в криптологии. М.: ДИАЛОГ-МИФИ, 2010.
2. *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: ТРИУМФ, 2002.
3. Словарь криптографических терминов / под ред. Б. А. Погорелова и В. Н. Сачкова. М.: МЦНМО, 2006. 94 с.
4. *Брассар Ж.* Современная криптология: пер. с англ. М.: Полимед, 1999. 173 с.
5. *Грушо А. А., Тимонина Е. Е., Применко Э. А.* Анализ и синтез криптоалгоритмов. Курс лекций. Йошкар-Ола: МФ МОСУ, 2000.