УДК 004.056.5

МАРКОВСКИЙ А.П.,
ШАРШАКОВ А.С.

СПОСОБ УСКОРЕННОЙ ГЕНЕРАЦИИ ПСЕВДОСЛУЧАЙНЫХ ПЕРЕСТАНОВОК ЭЛЕМЕНТОВ В СИСТЕМАХ ЗАЩИТЫ ИНФОРМАЦИИ

В статье предложен новый способ генерации псевдослучайных перестановок элементов конечного множества, базирующийся на идее конгруэнтных генераторов. Предложенный способ позволяет ускорить формирование случайных перестановок без значительных затрат памяти. Показано, предложенный подход позволяет увеличить скорость программной реализации комбинаторной задачи формирования перестановок элементов в n раз по сравнению с известными методами. Предлагаемый способ может найти широкое применение в современных высокопроизводительных системах защиты информации.

In paper we propose the new approach to generation of pseudorandom finite set of n elements permutation based on the idea of congruent generators usage. The proposed approach allows to speed up the pseudorandom permutations generation without extra memory amounts. It has been shown that proposed approach allows to speed up software implementation of combinatory task permutations generation by n times in compare to known methods. The proposed approach could be used in modern high-performance data security systems.

Введение

Генерация случайных чисел, и их последовательностей играют важную роль в информационных технологиях. Они широко используются при статистическом моделировании, тестировании аппаратных и программных средств, в технологиях искусственного интеллекта. В последнее десятилетие быстрый прогресс информационной интеграции стимулирует развитие систем защиты данных, в которых широко используются случайные и псевдослучайные элементы [1].

Одной из важных для современных технологий защиты информации задач является генерация случайных перестановок элементов конечного множества (ГСПЭ). Эта задача лежит в основе перестановочных криптографических алгоритмов, процедур модификаций преобразований алгоритмов блокового шифрования, полиморфной реализации программных кодов для защиты данных от их реконструкции анализом динамики потребляемой мощности (АДПМ). При работе же с полиморфными программами реализация незаконного доступа к данным с использованием АДПМ значительно усложняется [2].

При всех указанных применениях ГСПЭ важным фактором эффективности является время решения этой задачи. Поскольку защита данных является, по своей сути, вспомогательной функцией, ее реализация не должна существенным образом сказываться на скорости обработки информации. Поэтому комбинаторная задача ГСПЭ должна решаться ДОСтаточно быстро и в качестве критерия здесь используется зависимость числа выполняемых операций от количества элементов. Предпочтение отдается тем методам, которые реализуют задачу не за счет дополнительных ресурсов, а за счет математического аппарата.

Другим важным критерием эффективности при решении задачи ГПСЭ количества вариантов перестановки, которые могут быть сгенерированы. Для некоторых применений важным является, чтобы это число вариантов было достаточно большим, для других играет роль, чтобы генерировались принципиально все возможные варианты перестановок.

Таким образом, научная задача создания эффективных способов и средств генерации случайных и псевдослучайных перестановок элементов конечного множества является актуальной и практически важной для современного этапа развития информационных технологий в целом и систем защиты информации в частности.

Анализ известных технологий ГСПЭ

Задача ГСПЭ может быть сведена к выбору последовательности номеров следования элементов конечного множества $\Omega=\{X_1,X_2,,X_n\}$. В таком ракурсе можно рассматривать элементы множества Ω как натуральные числа от 1 до n, т.е. $X_1=1$, $X_2=2,\ldots$, $X_n=n$.

Для задачи ГСПЭ основными критериями эффективности механизма ГСПЭ являются [3]:

– длительность цикла n повторения чисел в выборке;

– число вариантов h перестановок элементов множества Ω;

– время генерации t очередного элемента множества Ω;

– среднее время генерации T множества Ω;

Существует ряд известных технологий решения задачи ГСПЭ [1]. Их можно разделить на два класса, в зависимости от механизма, обеспечения неповторяемость элементов:

– методы, которые обеспечивают неповторяемость элементов Ω в генерируемой последовательности за счет обработки указанного множества (т.е. выполняя сравнение с ранее сформированными элементами указанного множества и исключение из множества);

– методы, в которых неповторяемость элементов множества Ω обеспечивается математическими процедурами генерации Ω (т.е. используется технология генерации последовательности, не требующей обработки).

Ниже будут проанализированы технологии обоих классов и даны их характеристики.

Тривиальная технология, относящаяся к первому классу, состоит в том, что вначале множество Ω полагается пустым, а затем выполняется n циклов, в i-том (i=1,…,n) из которых генерируется случайное число $r\in\{1,\ldots,n\}$, которое сравнивается с ранее сформированными элементами множества Ω: если $r\notin\Omega$, то элемент r добавляется к множеству Ω: $\Omega = \Omega\cup r$; иначе генерируется новое число $r\in\{1,\ldots,n\}$ и снова осуществляется проверка, пока не будет сгенерировано такое r, которое можно добавить ко множеству Ω (т.е. $r\notin\Omega$). Пример процесса ГСПЭ для n = 8 представлен в таблице 1. Среднее число g_i циклов генерации i-го элемента множества Ω определяется формулой:

$$g_i = \sum_{j=1}^{\infty} j\cdot\frac{n-i+1}{n}\cdot\left(\frac{i-1}{n}\right)^{j-1} = \quad (1)$$

$$= \frac{n-i+1}{n}\cdot\sum_{j=1}^{\infty} j\cdot\left(\frac{i-1}{n}\right)^{j-1} =$$

$$= \frac{n-i+1}{n}\cdot\left(\frac{n}{n-i+1}\right)^{2} = \frac{n}{n-i+1}$$

Суммарное число g_Σ циклов генерации всех элементов множества Ω определяется формулой:

$$g_\Sigma = \sum_{i=1}^{n} g_i = n\cdot\sum_{i=1}^{n}\frac{1}{i} = n\cdot\ln(n) \quad (2)$$

Среднее время T_1 генерации всех n элементов множества Ω определяется выражением: $T_1 = g_\Sigma\cdot(\tau + 0.25\cdot t\cdot n) = n\cdot\ln(n)\cdot(\tau + 0.25\cdot t\cdot n)$, где τ – время генерации r, t –время выполнения команды процессора. Вычислительная сложность этого способа составляет $O(n^2\cdot\ln(n))$.

Табл. 1. Пример ГСПЭ базовым способом первого класса

№ цикла	*r*	Ω
		{}
0	2	{2}
1	7	{2,7}
2	2 5	{2,7} {2,7,5}
3	4	{2,7,5,4}
4	5 7 1	{2,7,5,4} {2,7,5,4} {2,7,5,4,1}
5	3	{2,7,5,4,1,3}
6	4 6	{2,7,5,4,1,3} {2,7,5,4,1,3,6}
7	3 1 0	{2,7,5,4,1,3,6} {2,7,5,4,1,3,6} {2,7,5,4,1,3,6,0}

Далее приведен анализ другой, наиболее известной из относящихся к первому классу, технологии ГСПЭ. Она сводится к выполнению n циклов, в i-том из которых (i=1,…,n) генерируется случайное число $r\in\{1,\ldots,n+1-i\}$ с исключением r-го элемента из копии S множества Ω. Реализация исключения выбранного элемента r решается путем сдвига части множества S. Среднее время T_2 ФСПЭ составляет $T_2=n\cdot(\tau+0.25\cdot t\cdot n)$, где τ – время генерации случайного числа r, t – время выполнения команды процессора.

Еще одна технология, основанная на идее предвычислений, состоит в предварительном формировании $G \le n!$ различных перестановок элементов Ω с сохранением их в памяти. Соответственно, при ГСПЭ генерируется случайное число $q\in\{1,..,G\}$ которое и определяет номер считываемой из памяти последовательности S. Время T_3 ГСПЭ состав-ляет $T_3 = \tau + t\cdot n$, а требуемый для хранения предвычисленных перестановок объем V_3 памяти: $V_3 = G\cdot n$. При очевидном выигрыше по времени ГСПЭ на устройстве, требуются значительные затраты па-

мяти и дополни-тельные меры по защите от утечки информации о таблице предвычислений.

Общим недостатком способов, относящихся к первому классу, является то, что время получения множества Ω имеет квадратичную зависимость от числа n элементов формируе-мого множества, то есть вычислительная сложность процедур генерации составляет $O(n^2)$. Достоинством технологий рассмат-риваемого класса является то, что они позволяют реализовать близкое к макси-мальному (т.е. $n!$) число вариантов h пере-становок элементов множества Ω.

Способ ГСПЭ конгруэнтного типа

Для реализации процедуры генерируется два случайных числа $1 \le \alpha$ и $1 \le \beta < n$, причем β и n должны быть взаимно простыми, то есть не иметь общих делителей. Каждое i-тое число X_i множества Ω формируется в соответствии со следующим выражением:

$$X_i = (\alpha + i \cdot \beta) \bmod n \qquad (3)$$

По сути, предлагаемый подход к генерации неповторяющейся последовательности из n натуральных чисел является модификацией конгруэнтных генераторов [1]. Достаточно просто показать, что ни одно из n генерируемых с использованием выражения (3) чисел не повторяется. Действительно, если пред-положить, что в процессе генерации формируются два равных между собой числа $X_i = X_j$ с порядковыми номерами i и j (причем $i, j \in \{1,..,n\}: j > i$), то это означает, что $i \cdot \beta \bmod n = = j \cdot \beta \bmod n$ или, что есть то же самое, $(j-i) \cdot \beta \bmod n = 0$. Из последнего следует, что суще-ствует целое d такое, что $(j-i) \cdot \beta = d \cdot n$, или:

$$\frac{(j-i) \cdot \beta}{n} = d \qquad (4)$$

Если $j-i$ и n не имеют общих делителей, то есть являются взаимно простыми, то левая часть выражения (4) не может быть целым числом. Это означает, что выражение (4) не выполняется при условии существования целого d. Пусть $j-i$ и n имеют наибольший общий целый делитель a, то есть $n = a \cdot s$ и $j-i = a \cdot c$. Тогда, в силу того, что $j-i < n$, то $c < s$, левая часть (4) имеет вид: $c \cdot \beta / s$. Поскольку ни c, ни β не имеют общих делителей с s, то выражение $c \cdot \beta / s$ не может быть целым числом. А это значит, что равенство (4) не может иметь места. Таким образом, доказано, что генерация элементов множества Ω в соответствии с выражением (3) обеспечивает отсутствие повторений.

Функционирование предложенного способа ГСПЭ иллюстрируется следующим примером. Например, при $\alpha = 3$, $\beta = 5$ и $n = 8$ генерируется следующая последовательность значений, составляющих множество Ω: {3,0,5,2,7,4,1,6}, как показано в таблице 2.

Табл. 2. Пример ГСПЭ предлагаемым способом

№ цикла i	Выражение для X_i	Ω
		{}
0	(3+0·5) mod 8	{3}
1	(3+1·5) mod 8	{3,0}
2	(3+2·5) mod 8	{3,0,5}
3	(3+3·5) mod 8	{3,0,5,2}
4	(3+4·5) mod 8	{3,0,5,2,7}
5	(3+5·5) mod 8	{3,0,5,2,7,4}
6	(3+6·5) mod 8	{3,0,5,2,7,4,1}
7	(3+7·5) mod 8	{3,0,5,2,7,4,1,6}

Очевидно, что порядок следования чисел в генерируемой последовательности не зависит от значения α: при изменении α меняется начальное значение. Например, при $\alpha = 1$, $\beta = 5$ и $n = 8$ генерируется следующая последовательность значений, составляющих множество Ω: {1,6,3,0,5,2,7,4}.

Ясно, что количество $N(\alpha)$ возможных значений α не превышает n: $N(\alpha) \le n$. В то же время количество $N(\beta)$ возможных значений β меньше n: $N(\beta) < n$. Общее количество возможных последовательностей чисел длиной n определяется произведением $N(\alpha) \cdot N(\beta) < n^2$.

Важно отметить, что на начальное значение генерируемой последовательности влияют как α, так и i. Соответственно, можно получать начальное значение α либо i, используя ГПСЧ любой разрядности, поскольку α и i не обязательно меньше n. Для получения следующих элементов множества Ω используется инкремент либо декремент i.

Однако стоит отметить следующую особенность: если есть такие $\beta_1 \ne \beta_2$, что $\beta_1 + \beta_2 = n$, то β_1 и β_2 генерируют обратные последовательности. Например, для n=8, $\alpha = 0$:

$\beta_1 = 1$, $\Omega = [0, 1, 2, 3, 4, 5, 6, 7]$.
$\beta_2 = 7$: $\Omega = [0, 7, 6, 5, 4, 3, 2, 1]$.
$\beta_3 = 3$, $\Omega = [0, 3, 6, 1, 4, 7, 2, 5]$.
$\beta_4 = 5$: $\Omega = [0, 5, 2, 7, 4, 1, 6, 3]$.

Исходя из этого, следует учесть закономерность, по которой определяется число вариантов h перестановок элементов множества Ω. При $n = 8$ есть 2 пары $\beta_1 = 1$, $\beta_2 = 7$ и $\beta_3 = 3$, $\beta_4 = 5$. При n вариантах выбора начального значения последовательности получим $h = 4 \cdot 8 = 32$.

При $n = 16$ есть 4 пары $\beta_1 = 1$, $\beta_2 = 15$, $\beta_3 = 3$, $\beta_4 = 13$, $\beta_5 = 5$, $\beta_6 = 11$ и $\beta_7 = 7$, $\beta_8 = 9$. При n ва-

риантах выбора начального значения последовательности получим: $h = 8 \cdot 16 = 128$.

То есть количество вариантов h перестановок элементов множества Ω пропорционально n и количеству таких чисел β ($1 \le \beta < n$), которые являются взаимно простыми с n. С учетом этого сложно оценить число вариантов h перестановок элементов множества Ω для данного способа в общем случае. Экспериментальные данные о числе вариантов h перестановок для $10 \le n \le 200$ представлены графически на рисунке 1. При составлении графика для демонстрации тенденции увеличения числа вариантов h перестановок с увеличением n была выполнена интраполяция. Фактически зависимость $h(n)$ связана с распределением взаимно простых с n чисел и сложно предсказуема.

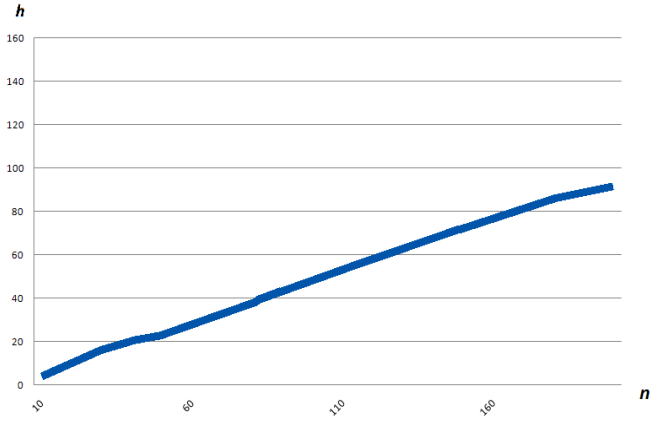

Рис. 1. График зависимости количества перестановок h от количества элементов n

Время T_4 генерации последовательности чисел, составляющих множество Ω определяется как: $T_4 = 2 \cdot \tau + 3 \cdot n \cdot t$, причем для генерации очередного элемента требуется только 3 команды: умножения, сложения и целочисленного деления для нахождения остатка, то есть минимум меньше по сравнению с рассмотренными выше способами. Затраты памяти при реализации предлагаемого способа также незначительны, что соответствует поставленной задаче.

Сравнительные характеристики предложенного способа ГППЭ представлены в таблице 3.

Табл. 3. Сравнительная характеристика параметров описанных механизмов ГСПЭ

Класс	Усл. обозн.	O	V	h
1	1Б	$O(n^2 \cdot \ln(n))$	$\approx n$	$\le n!$
	1И	$O(n^2)$	$\approx 2 \cdot n$	$\le n!$
	1П	$O(n^2)$	$\le n \cdot n!$	$\le n!$
2	2К	$O(n)$	$\approx n$	н/о

Условные обозначения, использованные в таблице 3: 1Б, 1И и 1П – базовый вариант, метод исключения и метод предвычислений соответственно как реализации методов первого класса, 2К – предложенный метод на основе конгруэнтного генератора.

Выводы

Разработан конгруэнтный способ ГСПЭ, обеспечивающих повышенную производительность. Доказано, что способ позволяет в n раз ускорить процедуру ГСПЭ при программной реализации за счет того, что для генерации очередного элемента последовательности требуется только 3 команды: умножения, сложения и целочисленного деления.

Предлагаемый способ ГСПЭ ориентирован на применение в современных высокопроизводительных системах защиты информации.

Список литературы

1. Иванов М.А., Чугунков И.В. Теория, применение и оценка качества генераторов псевдослучайных последовательностей. М.: КУДИЦ-ОБРАЗ. – 2003 – 260 с.
2. Марковский А.П., Зюзя А.А. Эффективная реализация псевдослучайного выбора элементов множества // Матеріали XII Міжнародної науково-технічної конференції ”Системний аналіз та інформаційні технології”. – К.: НТУУ ”КПІ”. ІПСА. – 2010. – с.458.
3. Самофалов К.Г., Марковский А.П., Зюзя А.А., Лёзин А.С. Стохастически полиморфная реализация алгоритма Rijndael на микроконтроллерах и смарт-картах // Проблеми інформатизації та управління. Збірник наукових праць. К.:НАУ. – 2010.- Вип. 1(29). – с.150-158.