УДК 003.26:004.056.5 **Д.А. Ларченко, Ю.Г. Савченко**

УВЕЛИЧЕНИЕ ДЛИНЫ ПСЕВДОСЛУЧАЙНЫХ БИТОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЬЕЙ КОМБИНАТОРНЫМИ ПРИЕМАМИ

Рассмотрена задача построения генераторов псевдослучайных чисел на основе регистров сдвига с обратными связями по модулю 2. Показано, что при использовании не одного, а нескольких полиномов, разнообразие вариантов схем генераторов становится комбинаторно практически неисчерпаемым. Предлагается один из таких вариантов, который позволяет существенно увеличить длину последовательности и варьировать ее статистическими характеристиками.

Ключевые слова: регистр сдвига, скремблирование, генератор, криптоанализ, фильтр, полином.

Вступление

Регистры сдвига с обратными связями по модулю 2 (Linear feedback shift register − LFSR) нашли широкое применение для реализации разнообразных функций и процедур при помехозащищенном кодировании, сигнатурном диагностировании неисправностей в цифровых устройствах, скремблировании сообщений при информационном обмене и др. Основная функция (но не единственная) устройств этого класса – генерация псевдослучайных битовых последовательностей. Целью работы является нахождение способов увеличение длины неповторяющихся псевдослучайных битовых последовательностей.

Основная часть

Если оставить в стороне применение регистров с обратными связями в качестве кодирующих и декодирующих устройств [1], то требования к устройствам этого класса можно свести к таким двум.

1. Генерируемая битовая последовательность по своим статистическим характеристикам должна приближаться к случайной. Это, в частности, означает, что вероятности появления фрагментов любой фиксированной длины должны быть одинаковыми, т.е. 0 и 1 должны появляться с вероятностью $\frac{1}{2}$; $00, 01, 10, 11$ – с вероятностью $\frac{1}{4}$; 000, 001,…111 – $\frac{1}{8}$ и т.д. Кроме того, эти вероятности не должны зависеть от «предыстории», т.е. от того, какие битовые комбинации предшествовали появлению конкретного фрагмента.

2. Длина полного цикла генерируемой последовательности ***L*** должна быть максимально большой (чем больше, тем лучше). Особенно важно это для скремблеров, поскольку генерируемая последовательность – это, фактически, ключ шифрования, который определяет криптостойкость шифра. При сигнатурном диагностировании при увеличении тестовой последовательности уменьшается вероятность пропуска (невыявления) неисправности. Так, уже при $L = 10^9 \ldots 10^{10}$ вероятность того, что неисправность не проявится, становится пренебрежимо малой. А проведение диагностического эксперимента на реальных рабочих частотах в этом случае занимает всего несколько секунд.

Естественно, что при использовании любого детерминированного (регулярного) алгоритма генерации битовой последовательности добиться выполнения первого требования в части независимости от предыстории нереально, поскольку каждое последующее состояние регистра *однозначно* определяется предыдущим состоянием. Кроме того, во всех случаях практического применения генераторов на основе регистров сдвига с обратными связями требуется повторное воспроизведение одной и той же последовательности (для скремблеров при шифровании и дешифровании), при сигнатурном диагностировании – при построении словаря и при проведении диагностического эксперимента. По сути, реально можно лишь стремиться к ослаблению корреляционных связей между отдельными

фрагментами последовательности. Практически это означает, что задача предсказания конкретного вида фрагмента любой длины на основе знания значений предыдущих битов должна быть вычислительно достаточно трудоемкой (например, по временным затратам).

Традиционная схема LFSR с одним регистром (рис.1) легко реализуется как аппаратно, так и программно. Однако с точки зрения криптографических требований эта реализация достаточно просто может быть взломана при помощи известных алгоритмов, например, структура обратных связей регистра однозначно вычисляется, если известны 2n бит генерируемой последовательности [2] Известны и другие методы вскрытия, например, на основе так называемых алгебраических атак [3].

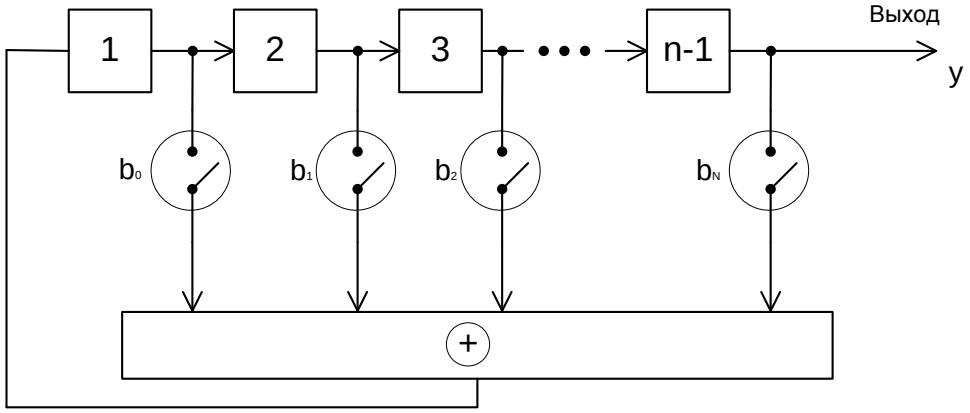

Рис. 1. Традиционная LFSR с одним регистром

Поэтому основным направлением в увеличении криптостойкости скремблирования остается усложнение ГПСЧ с целью ослабления корреляционных связей между отдельными фрагментами генерируемой последовательности.

Отметим здесь попутно, что требование равновероятности появления отдельных фрагментов в пределах длины регистра достаточно легко выполняется. Так, если в качестве генератора использовать простейший двоичный счетчик импульсов с возвратом в начальное состояние после заполнения, то это требование будет выполняться, хотя любое текущее состояние счетчика однозначно определено предыдущим состоянием.

Поэтому, исходя из приведенных соображений, остановимся, прежде всего, на возможностях увеличения длины цикла L генерируемой последовательности. Известно [1], что традиционная схема (рис.1) с одним регистром сдвига способна генерировать битовый цикл максимальной длины $2^n - 1$, где n – длина регистра. Это соотношение справедливо лишь при соответствующем выборе полинома, определяющем конкретный вид обратных связей в регистре. Полином

$$F(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \ldots + b_1 x + b_0$$

для получения последовательности максимальной длины должен быть неприводимым (примитивным) и быть делителем бинома $x^{2^n}+1$, т. е. всегда b_0=1.

Например, при $n = 15$ схема, приведення на рис.1, генерирует цикл из $(2^{15}-1)$ двоичных чисел разрядности 16 бит или $(2^{15}-1)\ 2^4 \approx 2^{19} \approx 10^6$ бит.

Как показала практика, этого вполне достаточно для сигнатурного диагностирования цифровых устройств. Однако для скремблирования этого может оказаться мало, поскольку реальная криптостойкость скремблирования в нашем случае определяется не длиной ключа (10^6 бит более чем достаточно), а возможностью вычислить структуру генератора и

стартовое слово, с которого начинается генерация. Исходя из приведеннях соображений, для повышения эффективности скремблирования следует не только удлиннять цикл, но и, прежде всего, усложнять структуру фильтра с тем, чтобы максимально усложнить криптоанализ.

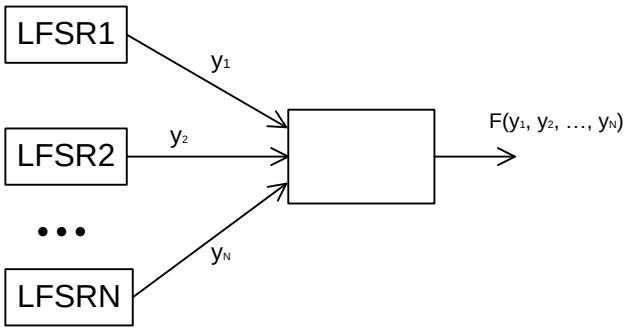

Рис. 2. Объединение генерируемых последовательностей

Очевидные варианты ослабления корреляционных связей базируются на использовании не одного, а нескольких LFSR и объединении генерируемых последовательностей. Объединение может быть осуществлено путем вычисления некоторой логической функции $f(y_1, y_2, \dots, y_N)$ от значений, генерируемых каждым LFSR в каждом такте. (рис.2). Однако здесь сразу же возникает вопрос: какой должна быть объединяющая функция? Можно показать, что если сохраняется требование равновероятности появления в последовательности 0 и 1, то в качестве $f(y_1, y_2, \dots, y_n)$ могут быть использованы только линейные функции. Однако строгое выполнение этого требования, по-видимому, не является обязательным. Действительно, если в генерируемой последовательности нарушается равномерное распределение 0 и 1 (возникает асимметрия), то это конечно же увеличивает корреляцию на битовом уровне, с одной стороны, но, с другой, − ввиду большого разнообразия функций, которые могут быть использованы для объединения, существенно усложняет криптоанализ. Одним из известных примеров [4] использования нелинейных функций являются пороговые функции, когда результирующий после объединения бит вычисляется по мажоритарному правилу

$$f(y_1, y_2 \dots, y_n) = \begin{cases} 1, \textit{если } t > \dfrac{N+1}{2} & \textit{для нечетных } N; \\ 0, \textit{если } t < \dfrac{N+1}{2} & \textit{для четных } N; \end{cases}$$

где N – количество единиц в слове $(y_1, y_2 \dots, y_n)$.

Очевидно, что при использовании нескольких полиномов может быть предложено достаточно большое количество вариантов реализации ГПСЧ. Рассмотренный в [5] вариант предполагал последовательное переключение в фиксированном порядке полиномов из некоторой заданной совокупности. Ниже предлагается еще один достаточно очевидный подход, основанный на использовании для генерации битовой последовательности также не одного полинома, а некоторой их совокупности и механизма переключения в процессе генерации (создание «смеси» из последовательностей, соответствующих каждому из полиномов). Причем порядок их участия в процессе генерации комбинаторно изменяется.

Покажем, что возможности увеличения количества (разнообразия) различных последовательностей при таком подходе становится практически неограниченым.

Пусть для «микширования» в качестве исходных выбрано ***m*** неприводимых полиномов F_1(x), $F_2(x)$,…,F_m(x). Тогда путем простой перестановки уже можно образовать $m!$ различных последовательностей длиной

$$L = (2^n - 1)m$$

для полиномов одинаковой степени, равной n. Так, например, для $m = 3$ может быть выбрана любая из $m! = 3! = 6$ смесей (комбинаций)

$$F_1(x)F_2(x)F_3(x);\ F_1(x)F_3(x)F_2(x);\ F_2(x)F_1(x)F_3(x);\ F_2(x)F_3(x)F_1(x);\ F_3(x)F_2(x)F_1(x);$$
$$F_3(x)F_1(x)F_2(x);$$

А для $m = 10$ разнообразие смесей уже
$m! = 10! = 2728800$
при $L = 10(2^n - 1)$.

Можно пойти дальше, организовав «смесь смесей». Разнообразие последовательностей становится таким, что их перебор вычислительно нереализуем.

Таким образом, за счет использования не одного, а нескольких полиномов можно существенно увеличить длину генерируемой последовательности и, как можно предположить, усложнить криптоанализ. Оценить существенность этого фактора достаточно сложно. По сути, по сравнению с использованием одного полинома (m=1) при криптоанализе добавляется задача вычисления всего набора полиномов и последовательности их переключения. Кроме того, конкретный вид последовательности зависит еще и от стартовых комбинаций, с которых начинается генерация последовательности, соответствующей каждому полиному. Здесь также могут быть различные варианты. Например, переход от полинома $F_i(x)$ к $F_j(x)$ может осуществляться при использовании в качестве стартовой последнюю комбинацию последовательности, генерируемой с помощью $F_i(x)$. Можно такой переход проводить по временным меткам, не дожидаясь завершения цикла. Все такие и аналогичные им процедуры могут быть достаточно просто реализованы аппаратно или программно. На рис.3 приведена общая структурная схема ГПСЧ с переключением конкретного вида обратных связей [8].

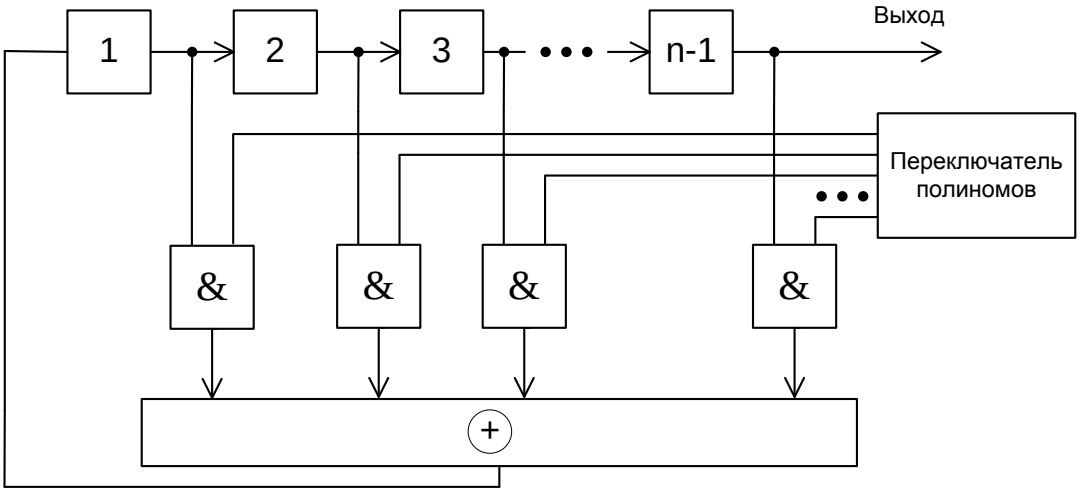

Рис. 3. ГПСЧ с переключением конкретного вида обратных связей

Конкретная реализация генератора при использовании полиномов

$$g_1(x) = x^{16} \oplus x^{12} \oplus x^9 \oplus x^7 \oplus 1; \quad g_4(x) = x^{16} \oplus x^9 \oplus x^5 \oplus x^3 \oplus 1;$$
$$g_2(x) = x^{16} \oplus x^{12} \oplus x^7 \oplus x \oplus 1; \quad g_5(x) = x^{16} \oplus x^{10} \oplus x^7 \oplus x^6 \oplus 1;$$
$$g_3(x) = x^{16} \oplus x^{12} \oplus x^9 \oplus x^6 \oplus 1; \quad g_6(x) = x^{16} \oplus x^{15} \oplus x^4 \oplus x^2 \oplus 1;$$
$$g_7(x) = x^{16} \oplus x^{10} \oplus x^5 \oplus x^3 \oplus 1;$$

позволяет создать $(2^{16} - 1)7! = 33029640$ различных последовательностей длиной 458745 бит. Эти цифры получены без учета возможности управления стартовыми словами при переключении полиномов. Если учесть и эту возможность, то приведенные цифры увеличиваются комбинаторно [6].

Выводы

Следует отметить, что, используя для генерации несколько полиномов, можно на порядки увеличить длину генерируемой последовательности и, как можно ожидать, в широких пределах варьировать статистические характеристики псевдослучайной последовательности. При любой реализации механизма переключения полиномов (программной или аппаратной) окончательный выбор того или иного алгоритма и последовательности переключения производится на основе результатов тестирования, например, с помощью статистических тестов NIST.

ЛИТЕРАТУРА

1. Берлекэмп Э. Алгебраическая теория кодирования. – М.: Мир, 1971, с.477.

2. Исагулиев К.П., Справочник по криптологии. Изд. Новое знание, 2004, 237 с.

3. Пометун С.О., Алгебраїчні атаки на потокові шифратори як узагальнення кореляційних атак. – Системні дослідження та інформаційні технології, №2, 2009, с.29-40.

4. Bruce Schneier Applied Cryptography: Protocols, Algorithms, and Source Code in C; John Wiley & Sons, 1996, 784 p.

5. Малогулко Р.В., Савченко Ю.Г.,Вдосконалення генераторів ПВП та їх застосування в системах скремблер-дескремблер телекомунікаційних пристроїв, Наукові записки УНДІЗ, №4, 2008, с.51-56.

6. Иванов М.А., Чугунков И.В. Теория, применение и оценка качества генераторов псевдослучайных последовательностей. — М.: КУДИЦ-ОБРАЗ, 2003. — 240 с

7. Розоринов Г.Н.,Толюпа С.В., Контроль как механизм обеспечения безопасности информационного обмена // "Правове, нормативне та метрологічне забезпечення захисту інформації в Україні, №1 (23), 2012, с.65-70.

8. Alan G. Konheim, Computer Security and Cryptography, John Wiley & Sons, Inc., 2009, 542 p.

Надійшла: 24.02.2014 р. Рецензент: д.т.н., проф. Розорінов Г.М.