УДК 004.421.5

ЗОХРЕ КАРИМ ЗАДЕ.,
МЕЛЬНИК А.П.,
САМКОВСКИЙ К.С.

МЕТОД ПОСТРОЕНИЯ НЕЛИНЕЙНОГО ГЕНЕРАТОРА ДВОИЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ НА СДВИГОВОМ РЕГИСТРЕ

Статья посвящена исследованию проблемы технологии проектирования сдвиговых регистров с нелинейными функциями обратной связи, которые гарантируют период повторения 2n для n-разрядного сдвигового регистра и обеспечивают улучшение характеристик двоичных последовательностей, которые важны для эффективности защиты информации. Разработан комбинаторный метод получения нелинейных булевых функций, обеспечивающий 2n для n-разрядного сдвигового регистра. Доказано, что предложенный метод позволяет на порядок увеличить количество нелинейных функций обратной связи по сравнению с известными методами.

Paper is dedicated to a problem of the design techniques of Nonlinear Feedback Shift Register (NFSR) with nonlinear feedback Boolean function, which ensure the repeat cycle 2n for n-bit register and improve the sequences quality characteristics which have an impact on data protection efficiency. The combinatorial method for obtaining nonlinear feedback functions which ensure the repeat cycle 2n for n-bit shift register has been worked out. It has been proved that proposed method allowed to increase the number of obtained feedback nonlinear function on one order in compare to known methods.

Введение

Псевдослучайные двоичные последовательности (ПСДП) активно применяются в информационных технологиях с середины 50-х годов. В последние годы технология получения таких последовательностей переживает этап динамичного развития в связи с расширением области их практического использования. Так, ПСДП широко используются в современных теле коммуникационных технологиях, в частности, в системах беспроводной передачи цифровых данных. Однако наиболее востребованным для ПСДП стало их использование в системах защиты информации. Они применяются в качестве базового элемента одного из трех типов алгоритмов защиты информации - потоковых алгоритмов. Кроме того, ПСДП широко используются для генерации ключей симметричных алгоритмов защиты данных и псевдослучайных двоичных строк протоколов аутентификации удаленных пользователей интегрированных систем [1].

В современных условиях роста производительности вычислительных систем и возможностей объединения значительного числа компьютеров в

сеть для нарушения защиты, актуальной становится проблема адекватного повышения надежности защиты информации, в том числе, за счет совершенствования средств получения и использования ПСДП.

Защитные свойства псевдослучайных последовательностей и функциональных преобразований, в теоретическом плане, определяются невозможностью аналитического решения систем нелинейных булевых уравнений [2]. Именно это свойство булевых преобразований лежит в основе использования ПСДП и булевых функций для защиты информации. Поэтому, уровень защищенности данных с использованием ПСДП прямо зависит от нелинейности булевых функций, которые используются для генерации ПСДП. Исходя из этого, важным резервом повышения эффективности широкого класса средств защиты информации, в которых применяются ПСДП, является усовершенствование средств формирования таких последовательностей в направлении увеличения нелинейности булевых преобразований, которые они используют.

Рост быстродействия вычислительных систем и скорости передачи цифровых данных выдвигает все более жесткие требования к производительности средств защиты информации, в том числе, к генераторам ПСДП: их реализация не должна заметно сказываться на основных процессах обработки или передачи информации. В этом плане, наиболее эффективными являются генераторы ПСДП на основе сдвиговых регистров. Именно поэтому, генераторы ПСДП на основе сдвиговых регистров с линейными функциями обратной связи получили наибольшее распространение в современных системах защиты данных [1]. Однако, использование сдвиговых регистров с линейной функцией обратной связи не обеспечивает необходимый уровень защиты: для этого в схему генераторов ПСДП вводятся дополнительные нелинейные преобразователи. Это усложняет средства получения ПСДП и снижает их быстродействие.

Таким образом, на современном этапе развития технологии защиты информации в системах и сетях вычислительной техники актуальной является задача повышения эффективности средств генерации ПСДП, в том, числе на основе сдвиговых регистров с нелинейными функциями обратной связи.

Генераторы ПСДП на сдвиговых регистрах

Как отмечалось выше, в подавляющем большинстве, генераторы ПСДП для систем защиты информации строятся на основе сдвиговых регистров с линейными функциями обратной связи (LFSR Linear Feedback Shift Register). Если линейная функция обратной связи соответствует простому полиному на полях Галуа, то период повторения кода на n-разрядном сдвиговом регистре равен 2^n-1 [1]. При аппаратной реализации LFSR обеспечивают высокий темп генерации ПСДП, которые хотя имеют хорошие статистические характеристики, но элементы которых линейно

зависимы. Соответственно, такие ПСДП могут быть легко предсказаны путем анализа выборки длиной $2 \cdot n$ бит. Поэтому реальные генераторы на основе LFSR имеют в своем составе нелинейные преобразователи. Это усложняет схему генератора и замедляет его работу.

Наиболее естественным решением проблемы повышение эффективности генерации ПСДП является использование для этой цели сдвигового регистра с нелинейной функцией обратной связи (NFSR-Nonlinear Feedback Shift Register). Имея простую структуру, NFSR могут быть эффективно реализованы аппаратно, обеспечивают высокий темп генерации ПСДП. Нелинейный характер функции обратной связи сводит задачу предсказания последовательности к решению системы нелинейных булевых уравнений, которое принципиально не может быть выполнено аналитически [2].

Основной трудностью практического применения NFSR является сложность получения функции $f(x_1,x_2,\ldots,x_n)$ обратной связи, определенной на компонентах вектора $X=\{x_1,x_2,\ldots,x_n\}$ значений разрядов сдвигового регистра и обеспечивающей период повторения 2^n. Общее количество N_f таких функций определяется формулой [2]:

$$N_f(n) = 2^{2^{n-1}-n} \qquad (1)$$

Хотя, определяемое приведенной формулой (1) число N_f для имеющих место на практике значений n достаточно велико и на несколько порядков превышает число подходящих функций для LSFR, получение самих функций, обеспечивающих период повторения 2^n представляет собой сложную задачу [2]. Так, уже при n=10 доля подходящих функций составляет $2^{-522} \approx 10^{-174}$ от общего числа функций.

Поэтому к настоящему времени предложено ряд методов получения функций рассматриваемого класса. Практически все из них используют те или иные специфические свойства функций, обеспечивающих в качестве обратной связи максимальный период повторения кода на сдвиговом регистре. Текущее состояние разрядов сдвигового регистра длиной n бит характеризуется двоичным вектором X_w значений его разрядов, который соответствует числу w:

$$X_w = \{x_1^w, x_2^w, \ldots, x_{n-1}^w, x_n^w\},$$

$$\forall j \in \{1,\ldots,n\} : x_j^w \in \{0,1\},$$

$$w = \sum_{j=1}^{n} x_j^w \cdot 2^{j-1}$$

При сдвиге регистра, его младшему разряду x_1 присваивается значение функции обратной связи $f(x_1,x_2,\ldots,x_n)$, соответственно, новое значение вектора битовых значений регистра становится равным X_v. Значение вектора X_v и соответствующего ему кода v определяется следующим образом:

$$X_v = \{ f(X_w), x_1^w, ..., x_{n-2}^w, x_{n-1}^w \},$$
$$v = (2 \cdot w) \bmod 2^n + f(X_W)$$

Важнейшим свойством функций $f(x_1,x_2,\ldots,x_n)$ обратной связи, обеспечивающих полный период повторения кода на сдвиговом регистре является условие единственности перехода в каждое из состояний регистра:

$$f(x_1, x_2, ..., x_n) = 1 \oplus f(x_1 \oplus 1, x_2, ..., x_n) \qquad (2)$$

Основными критериями оценки методов получения нелинейных функций $f(x_1,x_2,\ldots,x_n)$ обратной связи сдвигового регистра являются:

- объем вычислительных ресурсов, затрачиваемых на получение функций рассматриваемого класса;
- количество функций, которые могут быть получены с использованием метода;
- нелинейность формируемых функций.

Наиболее простым методом получения нелинейных функций обратной связи NFSR является модификация линейной функции LFSR [1]. Очевидными недостатками метода является малая нелинейность получаемых с его помощью функций и малое их количество.

Малое количество функций можно получить и с использованием итерационного метода построения NFSR [2], сущность которого состоит в специальном порядка заполнения таблицы истинности функции обратной связи. Такой же недостаток присущ и методам, основанным на модификации заданной нелинейной функции обратной связи [2].

Более эффективный подход к получению функций для NFSR основан на использовании концепции объединения колец – кодов, получаемых при циклическом сдвиге. Методы, основанные на этой концепции, позволяют получать функции с высокой нелинейностью и имеют вычислительную сложность $O(n^2)$ [3]. Основными недостатками является сложность процедуры генерации колец и весьма узкий класс формируемых функций по отношению к общему количеству (1).

Таким образом, существующие методы не позволяют в достаточной для практики степени решать задачу получения большого числа функций для NFSR. Это имеет следствием то, что при нарушении защиты имеется реальная возможность перебора всех функций, которые могут быть получены с использованием известных методов и, тем самым, снижается эффективность реализуемой с использованием ПСДП защиты данных.

Целью работы является разработка метода получения нелинейных функций обратной связи для NFSR, обеспечивающего увеличение числа формируемых функций указанного класса по сравнению с известными методами.

Метод получения нелинейной функции обратной связи для NFSR

В основе предлагаемого комбинаторного метода получения булевой функции обратной связи сдвигового регистра с максимальным периодом повторения лежат специфические свойства таких функций.

Если функция $f(x_1, x_2,\ldots,x_n)$ обратной связи удовлетворяет условию (2), то каждому коду v на р**Ошибка! Закладка не определена.**егистре предшествует только один код w.

Для каждого кода $w \in \{0,\ldots,2^n\text{-}1\}$ сдвигового регистра можно указать пару $< v_0^w, v_1^w >$ кодов, которые могут образовываться при сдвиге регистра после кода w : $v_0^w = (2 \cdot w) \bmod 2^n$ и $v_1^w = (2 \cdot w) \bmod 2^n + 1$.

Для каждого кода $w \in \{0,\ldots,2^n\text{-}1\}$ сдвигового регистра существует в точности один сопряженный ему код u такой, что: $v_0^W = v_0^u$ и $v_1^W = v_1^u$.

Векторы состояний сдвигового регистра X_w и X_u, которые соответствуют сопряженным кодам w и u отличаются значениями старшего бита: $\forall i \in \{1,..., n-1\} : x_i^w = x_i^u, x_n^w \neq x_n^u$. Согласно свойству (2) функция обратной связи на сопряженных векторах принимает противоположные значения: $f(X_w) \neq f(X_u)$. Обозначим вектор X_u, сопряженный X_w как X_w'.

Текущее состояние характеризуется множеством ϑ фрагментов: $\vartheta = \{P_1, P_2,\ldots,P_r\}$. Текущее значение количества фрагментов равно r. Каждый j-тый фрагмент характеризуется парой кодов $P_j = <X_j^s, X_j^f>$.

Предлагаемый алгоритм получения функции обратной связи, обеспечивающей максимальный период повторения кодов на сдвиговом регистре сводится к следующей последовательности операций:

1. Установить, что множество ϑ состоит из двух (r=2) фрагментов: $P_1 = <X_1^s, X_1^f>, P_2 = <X_2^s, X_2^f>$, причем $X_1^s = \{1,0,\ldots,0\}$, $X_1^f = \{0,\ldots,0,1\}$, $f(X_1^s) = 0$; $f(0,\ldots,0) = 1$, $X_2^s = \{0,1,\ldots,1\}$, $X_2^f = \{1,\ldots,1,0\}$, $f(X_2^s) = 1$, $f(1,...1,1) = 0$.

2. Произвольно выбирается один из r фрагментов множества ϑ. Пусть его номер равен j, $j \in \{1,\ldots,r\}$. Конечный код j-то фрагмента принимается в качестве текущего: $X_t = X_j^f$. Определяется код - X_t', сопряженный текущему коду X_t.

3. Выполняется поиск фрагмента, содержащего сопряженный код X_t' в качестве конечного. Если в результате поиска найден i-тый фрагмент, такой, что $X_i^f = X_t'$, то сопряженный фрагмент равен i и выполняется переход на пп.5.

4. В противном случае строится новый, (r+1)-й фрагмент, начальный и конечный код которого совпадает и состоит только из одного кода X_t' : $P_{r+1} = <X_t', X_t'>$; в этом случае – $i := r+1$. Счетчик числа фрагментов увеличивается на единицу: $r := r+1$.

5. Определяется пара кодов $<X_{n0}, X_{n1}>$, в которые возможен переход из текущего кода X_t и сопряженного ему кода X_t': $X_{n0} = X_t << 1 = X_t' << 1$, а X_{n1}

$=(X_t<<1)\oplus 1 = (X_t'<<1)\oplus 1$, где через $<<1$ обозначен логический сдвиг влево на один разряд с заполнением нулем.

6. Если $X_{n0} = X_j^s$ или $X_{n1} = X_i^s$, переход на пп.9.
7. Если $X_{n1} = X_j^s$ или $X_{n0} = X_i^s$, переход на пп.10.
8. Выполняется произвольно переход либо на пп.9, либо на пп.10.
9. Устанавливается $f(X_t) = 1$, $f(X_t')=0$. Если X_{n1} не совпадает с начальным кодом ни одного из фрагментов, то установить $X_j^f = X_{n1}$. Если X_{n1} совпадает с началом q-го фрагмента: $X_{n1} = X_q^s$, то отметить как удаленный q-тый фрагмент, заменив конечный код j-то фрагмента конечным кодом q-того фрагмента: $X_j^f = X_q^f$. Если X_{n0} не совпадает с начальным кодом ни одного из фрагментов, то установить $X_i^f = X_{n0}$. Если X_{n0} совпадает с началом g-го фрагмента: $X_{n0} = X_g^s$, то отметить как удаленный g-тый фрагмент, заменив конечный код i-то фрагмента конечным кодом g-того фрагмента: $X_i^f = X_g^f$. Переход на пп.11.
10. Устанавливается $f(X_t) = 0$, $f(X_t')=1$. Если X_{n0} не совпадает с начальным кодом ни одного из фрагментов, то установить $X_j^f = X_{n0}$. Если X_{n0} совпадает с началом q-го фрагмента: $X_{n0} = X_q^s$, то отметить как удаленный q-тый фрагмент, заменив конечный код j-то фрагмента конечным кодом q-го фрагмента: $X_j^f = X_q^f$. Если X_{n1} не совпадает с начальным кодом ни одного из фрагментов, то установить $X_i^f = X_{n1}$. Если X_{n1} совпадает с началом g-го фрагмента: $X_{n1} = X_g^s$, то отметить как удаленный g-тый фрагмент, заменив конечный код i-то фрагмента конечным кодом g-того фрагмента: $X_i^f = X_g^f$.
11. Удалить из ϑ все отмеченные к удалению фрагменты, изменив и[нумерацию и скорректировав значение r. Если $r>0$, то возврат на пп.2.

Предложенный алгоритм синтеза булевой нелинейной функции обратной связи, обеспечивающей максимальный период повторения иллюстрируется следующим примером.

Пусть $n=4$. В соответствии с пп.1 алгоритма, множество ϑ вначале состоит из двух фрагментов: $P_1 = <\{1000\},\{0001\}>$, $P_2 = <\{0111\},\{1110\}>$ значения функции обратной связи определяются на 4-х наборах: $f(1000)=0$; $f(0000)=1$; $f(1110)=1$; $f(1111)=0$.

В соответствии с пп.2 произвольно выбирается второй фрагмент множества ϑ, то есть j = 2. В качестве текущего принимается код $X_t = X_2^f = \{1110\}$. Сопряженный ему код $X_t' = \{0110\}$.

Код X_t' не совпадает с конечным кодом ни одного из 2-х фрагментов. Поэтому, в соответствии в пп.4 строится 3-й фрагмент $P_3 = <\{0110\},\{0110\}>$, $i=3$. Согласно пп.5 определяется пара кодов $X_{n0} = X_t<<1 = \{1100\}$, $X_{n1} = (X_t<<1)\oplus 1 = \{1101\}$. Коды X_{n0} , X_{n1} не совпадают с конечными кодами j-го и i-го фрагментов, поэтому, согласно пп. 8 осуществляется переход, например, на пп.9. Соответственно, устанавливаются значения функций для X_t и X_t': $f(1110) = 1$, $f(0110)=0$. Коды X_{n1} и X_{n0} не

совпадают с начальным кодом ни одного из фрагментов множества ϑ, поэтому конечный код фрагмента P_2 заменяется кодом X_{n1}: $X_2^f = X_{n1} = \{1101\}$, а конечный код фрагмента P_3 заменяется кодом X_{n0}: $X_3^f = X_{n0} = \{1100\}$.

Поскольку $r = 3 > 0$, то выполняется возврат на пп.2. Из множества $\vartheta = \{<\{1000\},\{0001\}>,<\{0111\},\{1101\}>,<\{0110\},\{1100\}>\}$ произвольно выбирается, например, 3-й фрагмент, то есть $j=3$. Соответственно, $X_t = X_3^f = \{1100\}$, $X_t' = \{0100\}$. Код X_t' не совпадает с конечным кодом ни одного из фрагментов, поэтому, согласно пп. 4 множество дополняется новым фрагментом $P_4 = <\{0100\},\{0100\}>$ и $i=3$. Вычисляются коды $X_{n0} = \{1000\}$, $X_{n1} = \{1001\}$. Поскольку они не совпадают с начальными кодами фрагментов P_3 и P_4, то, согласно пп.8 выбор значения функций образной связи на наборе X_t осуществляется произвольно: например, по пп.10 $f(1100)=0$, $f(0100)=1$. Код X_{n0} совпадает с начальным кодом 1-го фрагмента ($q=1$). Согласно пп.10 указанный фрагмент помечается к удалению, а конечный код 3-го фрагмента замещается конечным кодом 1-го фрагмента: $X_3^f = X_1^f = \{0001\}$. Код X_{n1} не совпадает с начальным кодом ни одного из 4-х фрагментов множества ϑ, поэтому X_{n1} замещает конечный код фрагмента P_4: $X_4^f = X_{n1} = \{1001\}$. Из 4-х фрагментов пп.11 удаляется ранее отмеченный фрагмент P_1. Оставшиеся перенумеровываются, так, что после выполнения пп.11 множество ϑ состоит из 3-х фрагментов: $\vartheta = \{P_1, P_2, P_3\}$, $P_1 = <\{0111\},\{1101\}>$, $P_2 = <\{0110\},\{0001\}>$, $P_3 = <\{0100\},\{1001\}>$. Поскольку $r = 3 > 0$, то осуществляется возврат на пп.2.

На третьем цикле в качестве текущего выбирается код $X_t = \{1101\}$, конечный в первом фрагменте P_1: $j=1$. Сопряженный код $X_t' = \{0101\}$ не равен ни одному из конечных, поэтому множество ϑ дополняется новым фрагментом $P_4 = <\{0101\},\{0101\}>$, $i=4$. Образуемые коды $X_{n0} = \{1010\}$, $X_{n1} = \{1011\}$ не совпадают с начальными кодами фрагментов P_1 и P_3, что определяет произвольный выбор значения функции на наборе X_t : $f(1101)=1$, $f(0101)=0$. Так как коды X_{n0}, X_{n1} не совпадают с начальным кодом ни одного из 4-х фрагментов множества ϑ, то X_{n1} заменяет конечный код фрагмента 1, а X_{n0} заменяет конечный код фрагмента 4. После этого $P_1 = <\{0111\},\{1011\}>$, $P_2 = <\{0110\},\{0001\}>$, $P_3 = <\{0100\},\{1001\}>$, $P_4 = <\{0101\},\{1010\}>$. Поскольку $r = 4 > 0$, то осуществляется возврат на пп.2.

На четвертом цикле в качестве текущего выбирается код $X_t = \{0001\}$, конечный во втором фрагменте P_2: $j=2$. Сопряженный код $X_t' = \{1001\}$ совпадает с конечным кодом фрагмента P_3, поэтому $i=4$. Образуемые коды $X_{n0} = \{0010\}$, $X_{n1} = \{0011\}$ не совпадают с начальными кодами фрагментов P_2 и P_3, что определяет произвольный выбор значения функции на наборе X_t: например в соответствии с пп.10: $f(0001)=0$, $f(1001)=1$. Так как коды X_{n0}, X_{n1} не совпадают с начальным кодом ни одного из 4-х фрагментов множества ϑ, то X_{n1} заменяет конечный код P_2, а X_{n0} заменяет

конечный код фрагмента P_3. В конце 4-го цикла ϑ состоит из: P_1=<{0111}, {1011}>, P_2= <{0110},{0010}>, P_3=<{0100},{0011}>, P_4=<{0101}, {1010}>. Так как r=4>0, то осуществляется возврат на пп.2.

На пятом цикле в качестве произвольно выбираемого текущего кода X_t выделяется конечный код P_4: X_t = {1010}, j=4. Сопряженный код X_t'={0010} совпадает с конечным кодом фрагмента P_2, поэтому i=2. Формируются коды X_{n0} = {0100} и X_{n1} = {0101}. Второй из них совпадает с начальным кодом фрагмента P_4, X_{n1}= X_i^s, то есть выполняется условие пп.7, соответственно осуществляется переход на пп.10, в рамках которого устанавливается f(1010)=0, f(0010)=1. Код X_{n0} совпадает с началом 3-го фрагмента (q=3): X_{n0} = X_3^s, поэтому фрагмент P_3 отмечается к удалению, конечный код P_4 заменяется конечным кодом 3-го фрагмента - {0011}: P_4=<{0101},{0011}>. Код X_{n1} совпадает с началом 4-го фрагмента (g=4) : X_{n1} = X_4^s ={0101}. Поэтому, в соответствием с пп.10, фрагмент P_4 отмечается к удалению, а конечный код 2-го фрагмента заменяется конечным кодом 4-го фрагмента, так, что: P_2= <{0110},{0011}>. При выполнении пп.11 удаляются отмеченные 3-й и 4-й фрагменты, после чего множество ϑ состоит только из 2 фрагментов: P_1=<{0111},{1011}> и P_2= <{0110}, {0011}>. Так как r = 2>0, то осуществляется возврат на пп.2.

На шестом цикле в качестве текущего выбирается код X_t = {1011}, конечный во втором фрагменте P_1: j=1. Сопряженный код X_t'={0011} совпадает с конечным кодом фрагмента P_2, поэтому i=2. Формируются коды X_{n0} = {0110} и X_{n1} = {0111}. Второй из них совпадает с начальным кодом фрагмента P_1, X_{n1}= X_i^s, то есть выполняется условие пп.7, соответственно осуществляется переход на пп.10, устанавливается f(1011)=0, f(0111)=1. Код X_{n0} совпадает с началом 2-го фрагмента (q=2): X_{n0} = X_2^s, поэтому фрагмент P_2 отмечается к удалению, конечный код P_1 заменяется конечным кодом 2-го фрагмента – {0011}: P_1=<{0111},{0011}>. Код X_{n1} совпадает с началом 1-го фрагмента (g=1): X_{n1} = X_1^s ={0111}. Поэтому, в соответствием с пп.10, фрагмент P_1 отмечается к удалению. При выполнении пп.11 удаляются отмеченные 2-й и 1-й фрагменты, после чего множество ϑ становится пустым. Так как r = 0, то работа алгоритма закончена. Построенная функция в результате функция обратной связи f(X) приведена в таблице.

Таблица истинности функции f(X) обратной связи, обеспечивающей период 2^4 повторения кода на сдвиговом регистре

$x_1x_2x_3x_4$	$f(X)$	$x_1x_2x_3x_4$	$f(X)$	$x_1x_2x_3x_4$	$f(X)$	$x_1x_2x_3x_4$	$f(X)$
0 0 0 0	1	0 1 0 0	1	1 0 0 0	0	1 1 0 0	0
0 0 0 1	0	0 1 0 1	0	1 0 0 1	1	1 1 0 1	1
0 0 1 0	1	0 1 1 0	0	1 0 1 0	0	1 1 1 0	1
0 0 1 1	1	0 1 1 1	1	1 0 1 1	0	1 1 1 1	0

Выводы

В результате проведенных исследований предложен метод построения генераторов ПСДП на NFSR, основой которого является комбинаторный подход к получению нелинейной функции обратной связи, обеспечивающий максимальный период повторения кода на сдвиговом регистре. Фактически разработанный подход представляет собой способ уменьшения перебора всех функций, указанного класса на основе их специфических свойств.

Проведенные экспериментальные исследования показали, что предложенный метод эффективен для относительно небольшой длины n сдвигового регистра: $n \leq 20$. Основным достоинством комбинаторного подхода, лежащего в основе разработанного метода является значительное расширение числа получаемых функций обратной связи, обеспечивающих максимальный период повторения ПСДП. Метод может быть использован при проектировании высокоскоростных средств защиты данных при их передаче по беспроводным линиям.

Список использованной литературы

1. Иванов М.А., Чугунков И.В. Теория, применение и оценка качества генераторов псевдослучайных последовательностей. М.: КУДИЦ-ОБРАЗ.- 2003 – 260 с.
2. Golomb S.W. Shift register sequences. Laguna Hill. California: Aegean Park Press.- 1982.-324 p.
3. Марковський О.П., Зохре Карім Заде Сейфолах, Гурін В.Є. Ефективний метод побудови нелінійних генераторів для телекомунікаційних систем //Электроника и связь. Тематический выпуск ”Проблемы электроники” ч.3. – ПЦ ”Аверс” - 2007.- С.87-89.