Известия Гомельского государственного университета
имени Ф. Скорины, № 3 (144), 2024

УДК 004.451.7:621.376.4 EDN: QVNMWR

# Система шифрования данных на основе устройств фазовой синхронизации

Л.Ю. Шилин, А.А. Навроцкий

В работе исследована возможность использования динамического хаоса при кодировании информации. Предложена система кодирования информации для передачи по открытым каналам связи на основе асинхронного ключа. В качестве гамма-ключа для обратимого кодирования файлов предлагается использовать последовательность чисел, получаемых при помощи генератора при хаотическом режиме работы. Учитываются значения фазы и частоты сигнала на выходе блока фильтров в режиме хаоса.
**Ключевые слова:** система фазовой синхронизации, режим детерминированного хаоса, генератор псевдослучайных чисел, криптографическая система.

The paper explores the possibility of using dynamic chaos when encoding information. An information coding system for transmission over open communication channels based on an asynchronous key is introduced. It is proposed to use a sequence of numbers obtained with a generator in a chaotic mode of operation as a gamma key for reversible file encoding. The values of the phase and frequency of the signal at the output of the filter block in chaos mode are taken into account.
**Keywords:** phase synchronization system, deterministic chaos mode, pseudo random number generator, cryptographic system.

**Введение.** Бурное развитие информатики, компьютерных сетей и беспроводных систем порождает большое количество задач, которые трудно, а иногда и практически невозможно решать, используя только традиционные подходы теории информации. Исследователи все чаще исследуют возможность применения альтернативных методов, используемых для решения задач в других областях науки и техники. Ярким примером является использование теории динамического хаоса для разработки новых принципов шифрования информации, так как интенсивный рост производительности процессоров сводит на нет многие традиционные криптографические решения.

**Генератор псевдослучайной последовательности на основе системы фазовой синхронизаци.** Генераторы псевдослучайных чисел находят широкое применение в криптографии: например, при поточном шифровании, когда необходимо генерировать совершенно непредсказуемые или попросту случайные числа. Использование динамического хаоса открывает перспективы создания на базе систем фазовой синхронизации высокоэффективных генераторов колебаний с частотной и фазовой хаотической модуляцией. Такие генераторы могут быть реализованы как аппаратно, так и программно. Данный аспект делает весьма привлекательной идею реализации систем шифрования информации на основе динамического хаоса.

Для реализации подобных способов криптозащиты необходимо исследовать класс систем фазовой синхронизации с целью определения параметров работы в различных режимах и исследования их свойств. Поскольку рассматривается программная реализация генератора псевдослучайных чисел, необходимо построить достаточно гибкую алгоритмическую модель. До настоящего времени программы расчета и методы моделирования импульсных систем с фазовым управлением представляли собой сложные аналого-цифровые устройства, что существенно затрудняло их анализ и построение. Кроме того, проектирование подобных комплексов требует учета ряда как внешних, так и внутренних детерминированных и случайных возмущений. Вследствие этого зачастую невозможно построить аналитическую модель: в системе учитывается время, нелинейности, стохастические переменные, а экспериментирование на реальных схемах требует значительных затрат.

Вышеназванные недостатки требуют иного подхода к моделированию подобных систем: целесообразно использовать имитационное моделирование, которое рассматривается как вычислительный эксперимент со сложной математической моделью, описывающей поведение реализуемой системы на компьютере. Наиболее популярным алгоритмическим подходом моделирования процессов в системах фазовой синхронизации является высокоточный подход. Он основывается на составлении математических моделей разной сложности.

Разбиение системы на простые блоки и использование принципа имитационного моделирования позволяет составить простую математическую модель системы. Полагаем, что на вход моделируемой системы ФАПЧ поступает напряжение, которое содержит две составляющие: напряжение опорного генератора и напряжение шумов, воздействующих на систему.

$$U_{oz}(t) = U_{on}(t) + \eta(t),$$

где $U_{on}(t)$ – опорный сигнал, $\eta(t)$ – напряжение шума.

Для непрерывной системы напряжение опорного генератора определяется следующим выражением:

$$u_{on}(t) = U_{фд} \cdot \sin \varphi_{on},$$

где $U_{фд}(t)$ – сигнал на выходе фазового детектора, $\varphi_{on}$ – мгновенное значение фазы опорного генератора:

$$\varphi_{on} = \varphi_{on0} + \int_0^t \omega_{on} \cdot dt,$$

где $\varphi_{on0}$ – начальная фаза сигнала, $\omega_{on}$ – частота опорного генератора.

Для импульсной системы ФАПЧ опорный сигнал представляет собой последовательность коротких импульсов с частотой $\omega$.

Фазовый детектор (ФД) сравнивает частоты двух входных сигналов: сигнала опорного генератора и сигнала обратной связи; генерирует выходной сигнал, который является мерой их фазового рассогласования (например, если они различаются по частоте, то будет формироваться периодический выходной сигнал разностной частоты). Для рассматриваемой системы фазовый детектор будет определяет разность фаз входного сигнала и сигнала обратной связи.

$$\varphi = \varphi_{on} - \varphi_{oc},$$

где $\varphi_{on}$ – значение фазы опорного генератора, $\varphi_{oc}$ – значение фазы сигнала обратной связи.

В общем случае напряжение на выходе ФД можно представить выражением:

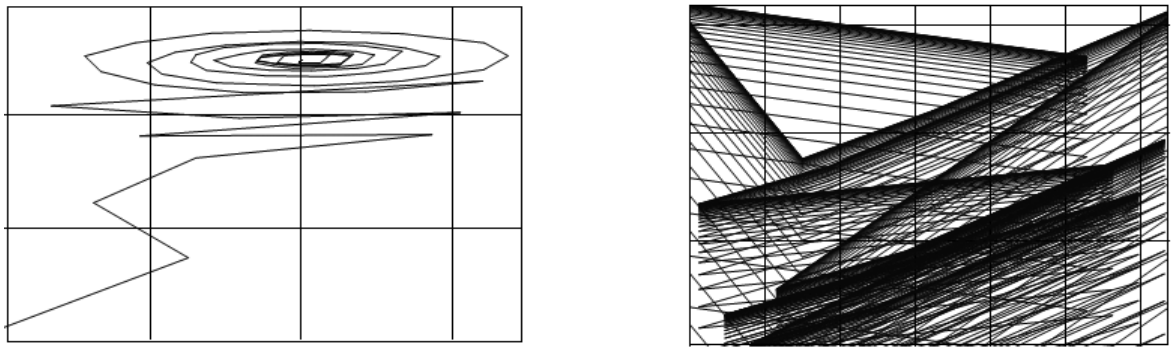
$$U_{д}(t) = M[U_{д}(t)] + \xi(t) = F(\varphi) + \xi(t),$$

где $F(\varphi) = M[U_{д}(t)]$ – дискриминационная характеристика, $\xi(t)$ – флуктуационная составляющая.

Данный подход характеризуется малыми затратами машинного времени, высокой точностью расчетов и при этом имеет жесткий алгоритм, который не позволяет при моделировании выйти за ограниченный круг исследуемых устройств. Следует отметить, что существующие пакеты прикладных программ для моделирования систем фазовой синхронизации (например, *Simulink*) во многих случаях представляют собой сложные и ресурсоемкие программные продукты, которые не обладают требуемой гибкостью и быстродействием. Кроме того, подобное программное обеспечение ориентировано на решение более общих задач, поэтому детальный анализ характеристик и режимов работы подобных систем зачастую не может быть реализован.

В связи с бурным развитием элементной базы, как следствие и мощностей вычислительной техники, наиболее привлекательными становятся упрощенные модели, которые основаны на модульном принципе и имитационном моделировании. Такой способ моделирования является более универсальным и позволяет разработчику всесторонне изменять структуру, включать и исключать дополнительные блоки.

Ранее была разработана математическая модель, применение которой позволило решить задачу моделирования и проанализировать работу аналоговых и импульсных систем фазовой синхронизации, а также оценить качество их работы. Были смоделированы характерные для подобных систем режимы работы. Традиционно в системе фазовой синхронизации в качестве основного динамического режима рассматривается режим синхронизации (установившийся режим) (рисунок 1, а), который характеризуется постоянством выходной координаты, неизменной длительностью импульсов, набега фазы. Он является основным рабочим режимом. Поэтому внимание исследователей было сосредоточено главным образом на изучении именно синхронного режима: точности синхронизации, областей удержания режима синхронизации в пространстве параметров и областей захвата в синхронный режим, времени вхождения системы в режим синхронизации и т. д.



а) б)

Рисунок 1 – Фазовый портрет моделируемой системы в установившемся режиме (а), в режиме детерминированного хаоса (б)

Асинхронный режим, обусловленный наличием нелинейности в системе, был интересен только потому, что он нарушал устойчивую работу системы. Данный режим является нерегулярным. Причина нерегулярности определяется свойством нелинейных систем экспоненциально быстро разводить первоначально близкие траектории. Поэтому не представляется возможным предсказать поведение таких систем, так как начальные условия можно задавать лишь с конечной точностью, а ошибки экспоненциально возрастают. Данный режим характеризуется построением странных аттракторов в области фазового пространства (рисунок 1, б). Рюэль, Такенс и Ньюхауз показали, что уже после двух неустойчивостей на третьем шаге траектория начинает притягиваться к ограниченной области фазового пространства, в которой первоначально близкие траектории расходятся, так что движение становится хаотическим [1]. Несмотря на нехарактерность этого режима, в настоящей работе предлагается рассматривать систему фазовой синхронизации в режиме детерминированного хаоса для использования в дальнейшем в качестве генератора псевдослучайных чисел.

**Описание рассматриваемой криптосистемы.** В настоящее время активно ведутся исследования возможности использования динамического хаоса при кодировании информации [2]. Наибольшее развитие получила аппаратная реализация криптосистем на основе хаоса. Это объясняется простой реализации элементов системы фазовой синхронизации. Проблемы воспроизводимости характеристик генератора хаоса решаются с использованием различной элементной базы: интегральные схемы, цифровые сигнальные процессоры, программируемые логические интегральные схемы.

Программная реализация в большей степени базируется на схеме с нелинейным подмешиванием информационного сигнала в хаотический. При такой реализации информационный сигнал непосредственно участвует в формировании сложного хаотического поведения ведущей системы. Такой ввод информации нельзя назвать ни аддитивным наложением, ни обычной модуляцией. Данный выбор продиктован такими свойствами схемы, как точное извлечения информации из смеси с хаотическим сигналом, самосинхронизация передатчика с приемником, простота реализации. В качестве нелинейного элемента обычно используются псевдогенераторы хаоса, полученные на основе классических систем уравнений Лоренца (формула (1)) и Расслера, отображений Хенона и Икеды, формула, функции Вейерштрасса-Мандельброта, уравнения Меки-Гласса , а также двухмерный и трехмерный генераторы Ван дер Поля, генераторы на основе логистического отображения и несимметричного *TENT*-отображения [3].

Система уравнения Лоренца (Lorenz System):

$$\frac{dx}{dt} = -\sigma(x - y)$$

$$\frac{dy}{dt} = x(r - z) - y$$

$$\frac{dz}{dt} = xy - bz,$$

где $\sigma$, $r$ и $b$ – параметрические коэффициенты, определяющие динамику системы.

При этом необходимо оценить роль управляющих коэффициентов на работу системы. Это представляет собой достаточно трудоемкую задачу.

Исходя из вышеизложенного, предлагается на основе созданной имитационной модели создать систему кодирования информации. Последовательности чисел, получаемые при помощи генератора на основе системы при хаотическом режиме работы, предлагается использовать в качестве гамма-ключа для обратимого кодирования файлов. Учитываются значения фазы и частоты сигнала на выходе блока фильтров в режиме хаоса.

Предлагается на основе построенного генератора создать систему кодирования информации для передачи последней по открытым каналам связи. Будет использован симметричный алгоритм, в котором шифрование и дешифрование отличается только порядком выполнения и направлением некоторых шагов. В алгоритме будет использоваться один и тот же секретный ключ. Дешифрование будет осуществляться простым обращением шифрования. Каждый участник обмена данными может как расшифровать, так и зашифровать сообщение.

На передающей стороне имеются источник сообщений и источник ключей. Источник ключей выбирает конкретный ключ среди всех возможных ключей данной системы. Этот ключ передается некоторым способом принимающей стороне, причем предполагается, что его нельзя перехватить. Источник сообщений формирует некоторое сообщение, которое затем шифруется с использованием выбранного ключа. В результате процедуры шифрования получается засекреченное сообщение (криптограмма). Далее криптограмма передается по каналу связи. На принимающей стороне криптограмму с помощью ключа расшифровывают и получают исходное сообщение. Так как канал связи является открытым, незащищенным (примером может служить компьютерная сеть), то передаваемое сообщение может быть перехвачено. Для проектирования универсальной системы шифрования данных предлагается использовать гаммирование. Этот способ предполагает, что шифрование выполняется путем сложения символов исходного текста и ключа по модулю, равному числу букв в алфавите. Осуществляется побитовое сложение *n*-битового открытого текста и *n*-битового ключа:

$$y_i = x_i \oplus k_i,\ i = 1, \ldots, n,$$

где $x_1 \ldots, x_n$ – открытый текст, $k_1 \ldots, k_n$ – ключ, $y_1 \ldots, y_n$ – шифрованный текст.

С точки зрения простоты реализации наиболее привлекательным является двоичное (битовое) гаммирование, так как при гаммировании по модулю два можно использовать одну и ту же операцию как для шифрования, так и для дешифрования. Операция сложения по модулю два очень быстро выполняется на компьютере (в отличие от многих других арифметических операций), поэтому наложение гаммы на текст большого объема не требует больших затрат времени. Недостатком такого подхода является невысокая скорость обработки больших объемов информации, например, видеофайлов. В этом случае при шифровании возможно изменение длины основания (например, не побитовое, а побайтовое шифрование сообщения).

**Программная реализация.** При проектировании эмулятора системы фазовой системы синхронизации было принято решение о разделении логики составных блоков. Принимая во внимание дальнейшее возможное использование и расширение моделируемых блоков, функциональные принципы работы составных блоков системы были реализованы в отдельном загружаемом модуле (динамически подключаемой библиотеке).

При разработке программного обеспечения была использована повторимая архитектурная конструкция – паттерн «компоновщик», структурирующий объекты. Этот паттерн позволяет компоновать составные части моделируемой системы в древовидные структуры для представления иерархии часть-целое, что позволяет единообразно трактовать индивидуальные и составные блоки системы. Архитектура приложения, согласно выбранному паттерну, будет иметь вид, представленный на рисунке 2.

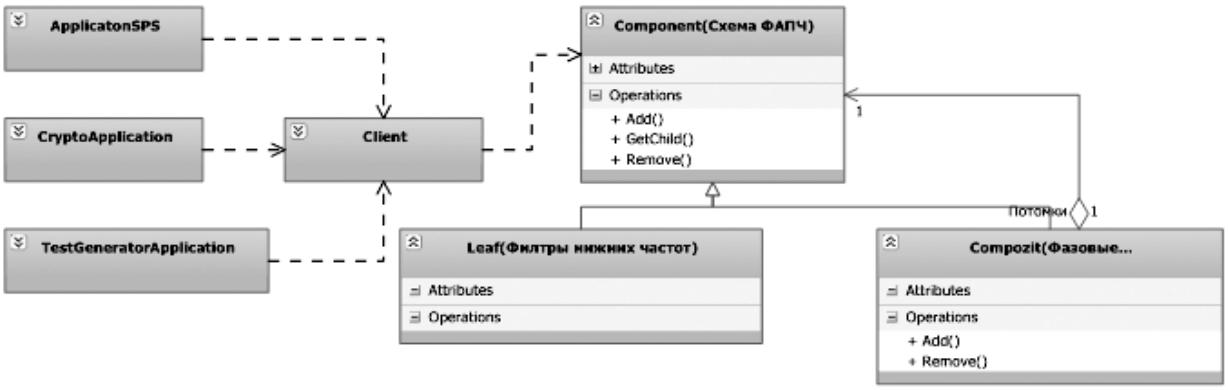



Рисунок 2 – Диаграмма архитектуры подключаемой библиотеки

*Component* (система ФАПЧ) – класс моделируемой системы. В нем объявляются интерфейсы для компонуемых объектов системы. Так же здесь также представлена реализация логики, общей для всех остальных классов. Этот класс инкапсулирует интерфейс для доступа к потомкам (составным элементам) и управления ними.

*Leaf* (фильтры нижних частот) – класс, объекты которого представляют логику поведения блоков фильтра нижних частот. Данные класс не имеет потомков, поэтому определяет поведение примитивных объектов в общей композиции.

*Compozit* (фазовые детекторы, объекты управления) – объекты этого класса определяют поведение компонентов, у которых есть потомки. В нем делегируется компоненты-потомки, и реализуются интерфейсы, относящееся к управлению объектов класса Leaf. Клиенты используют интерфейс класса Component для взаимодействия с объектами в составной структуре. Данный подход позволяет из примитивных объектов составлять сложные композиции. Любая схема может работать как с примитивным объектом, так и со сложным. Пользователь может единообразно работать с различными составными структурами. Используемый ключ представляет собой параметры моделируемой электрической схемы, которые можно условно разделить на две группы: «физические» и «архитектурные». К «физическим» относятся электрические величины элементов цепи: значения амплитуды и частоты опорного генератора, сопротивлений, индуктивностей, емкостей. Под «архитектурными» параметрами понимаются сами элементы электрической схемы (например, количество резистивностей или индуктивностей в фильтре, способ их соединения, используемый в цепи фазовый дискриминатор). Функциональность динамически загружаемой библиотеки, эмулирующей работу системы фазовой синхронизации, была расширена за чет добавления возможности сериализации объектов класса, эмулирующих логику работы ФАПЧ. Сериализация представляет собой процесс преобразования объекта в поток байтов с целью сохранения его в памяти, в базе данных или в файле. Ее основное назначение – сохранение состояния объекта для обеспечения возможности его восстановления. Предлагаемый способ работы с ключами преследует две цели. Во-первых, так значительно легче и быстрее обрабатывать данные, чем читать их из текстового файла. Во-вторых, преобразованные в последовательность байты не несут смысловую и статистическую нагрузку. Иными словами, без знания структуры объекта ключ становиться практически бесполезным. Как уже отмечалось ранее, должна существовать возможность изменения параметров кодирования в зависимости от размера файла.

При кодировании файла целиком (без учета структуры) снижается криптостойкость шифра. Это объясняется тем, что многие файлы помимо основных данных хранят однородные данные о формате. Поэтому для некоторых форматов файлов целесообразно шифровать только основные данные. При шифровании текстовых файлов преобразовывать символы в коды таблицы соответствующей кодировки (например, *ANSI, UNICODE*). Далее производить преобразование над кольцом, мощность которого соответствует размеру таблицы кодировки. Для повышения криптостойкости возможно использование обратного отображения кодов в

символы. При шифровании изображений необходимо выделить каждый пиксель изображения и выполнить гаммирование отдельно для каждого цветового канала. Результат шифрования рисунка в формате *.bmp представлен на рисунке 3.

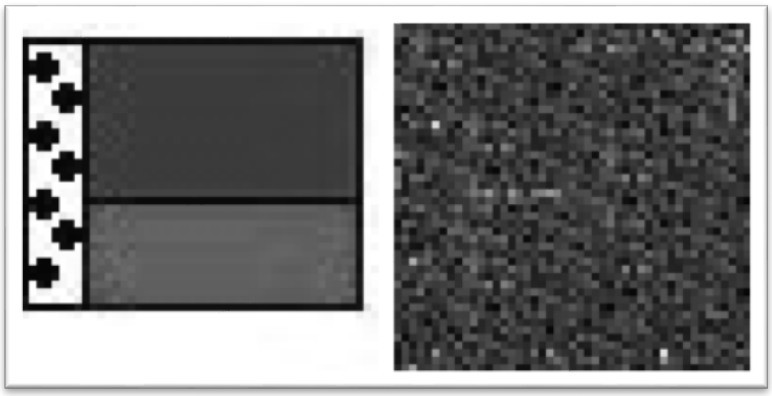


Рисунок 3 – Зашифрованное изображение

**Заключение.** Предложенная в работе криптосистема обладает рядом неоспоримых преимуществ. Для используемого ключа присуща случайность (равновероятность), ключ нельзя вырабатывать с помощью какого-либо детерминированного устройства. При шифровании коротких сообщений можно добиться равенства длины ключа и открытого текста. Важной особенностью системы является то, что при передаче ключа достаточно указать параметры работы системы в хаотическом режиме работы. Это уменьшает объем передаваемой информации и повышает стойкость шифра. Фактически происходит передача параметров генератора случайных чисел. Следует отметить высокую криптографическую стойкость и простую реализацию алгоритма.

## Литература

1. Шустер, Г. Детерминированный хаос. Введение / Г. Шустер. – М. : Мир, 1988. – 240 с.

2. Дмитриев, А. С. Динамический хаос : новые носители информации для систем связи / А. С. Дмитриев, А. И. Панас. – М. : Физматлит, 2002. – 251 с.

3. Кузнецов, С. П. Динамический хаос : курс лекций : учеб. пособ. для вузов / С. П. Кузнецов. – М. : Физматлит, 2001. – 295 с.

Белорусский государственный
университет информатики и радиоэлектроники

Поступила в редакцию 05.02.2024