**Исследовательская работа**

**на тему:** Алгоритм шифрования на основе элементарного клеточного автомата

**Автор работы:**

**Обучающийся 8 класса**

**ГБОУ Гимназия №1538**

Чистобаев Андрей Андреевич

**Руководитель работы:**

Носкова Людмила Николаевна,

Преподаватель математики, высшая категория

**Консультант:**

Дергачёва Алевтина Николаевна

Преподаватель информатики, первая категория

**Москва**

**2017 год**

**Содержание**

**Обоснование темы**……………………………………………………………….**3**

**Цели и задачи**…………………………………………………………………….**4**

**Гипотеза**…………………………………………………………………………...**5**

**Методика**………………………………………………………………………….**6**

**Тестирование**………..……………………………………………………………**7**

**Критика взлома**………………………………………………………………….**9**

**Итоги**……………………………………………………………………………..**10**

**Литература**………………………………………………………………………**11**

**Приложение 1. Словарик**

**Приложение 2. О работе элементарных клеточных автоматов**

**Приложение 3. О работе алгоритма**

**Приложение 4. Сравнительная характеристика**

**Обоснование темы**

Сохранение конфиденциальности данных пользователей есть главная обязанность разработчиков современности. Для её выполнения созданы различные средства защиты информации – криптографические алгоритмы и хэш-функции. Конечно, не все из них применимы в реальных проектах. Такие бесполезные инструменты представляют собой лишь научный интерес.

При выборе метода защиты опытный криптограф перво-наперво обратит своё внимание на такое важное свойство, как криптостойкость. Уже потом он взглянет на производительность, новизну и другие качества. Криптостойкость определяется годами, поэтому не стоит рваться за последними трендами.

Для определения уровня защиты, предоставляемого тем или иным инструментом, нужно подвергнуть шифротекст, им созданный, криптоанализу. Этим я и решил заняться.

По крайней мере, с 2008 года в открытом доступе находится описание алгоритма, основанного на клеточном автомате, ссылку на которое вы найдёте в списке литературы. Я готов предложить свою наработку, которую создавал в полном неведении о предшественнике. Было решено использовать элементарные клеточные автоматы, более подходящие для данной задачи.

Но возможность использования модифицированной версии на практике ещё нужно доказать.

**Цели и задачи**

Передо мной стоит единственная цель – доказать пригодность моей версии.

Эту цель я собираюсь решить созданием и тестированием своего собственного алгоритма шифрования.

Это решение можно разбить на несколько более мелких задач:

1. **Изучение основ криптографии и криптоанализа.** Необходимо изучить основные термины и методы работы.
2. **Изучение основ работы элементарных клеточных автоматов.** Изучением данной темы я занимаюсь уже 2 года.
3. **Изучение основ одного из языков программирования.** У меня имеются достаточные знания по C++ для написания подобной программы.
4. **Создание реализации алгоритма.** Реализация уже имеется и выдаёт верные результаты.
5. **Тестирование алгоритма.** Тестирование будет производиться компьютерной программой.
6. **Доработки.** Конечно, многое придётся изменить.
7. **Выкладывание реализации в свободный доступ.** При помощи этого мне удастся получить мнение других людей.
8. **Использование реализации на практике.** Возможность практического применения обсуждается в приложении 3.

План логичен и последователен, следовательно, можно начинать работу.

**Гипотеза**

У данного алгоритма, подробно описанного в приложении 3 (+ листинги с подсветкой синтаксиса), имеется существенный недочёт – открытый текст шифруется необратимо, в отличии от аналога. То есть привычное равенство не выполняется:

***D( E( M ) ) ≠ M***

Формула взята из книги Брюса Шнайера «Прикладная криптография».

Но в современном мире уже существуют и активно используются подобные алгоритмы.

Данные алгоритмы применимы при авторизации пользователя в компьютерной сети. То есть пароль, введённый пользователем, сохраняется в базу данных в зашифрованном виде. Так как шифрование необратимое, открытый текст (пароль) известен только вводившему его пользователю. При авторизации, пароль, введённый в форму, необратимо шифруется и сравнивается с шифротекстом из базы данных, соответствующим введённому логину. При их совпадении доступ к сети открывается.

Но не всё так просто. Я считаю, что, несмотря на необратимость элементарных клеточных автоматов, при определённом ключе k равенство ***Dk( Ek( M ) ) = M*** становится верным.

**Методика**

Криптоанализ, его основы, изучался мною по книге Брюса Шнайера «Прикладная криптография». В ней автор выделил 4 основных метода криптоанализа:

1. **Вскрытие с использованием только шифротекста.**
2. **Вскрытие с использованием открытых текстов и соответствующих им шифротекстов.**
3. **Вскрытие с использованием выбранного открытого текста.**
4. **Адаптивное вскрытие с использованием выбранного открытого текста.**

Очевидно, что название метода содержит доступные криптоаналитику данные.

Во всех 4 случаях целью вскрытия является нахождение ключа шифрования, так как на этапе вскрытия допустимо использование моей реализации алгоритма, потому что считается, что на этапе взлома хакеру доступен алгоритм защиты. То есть определение алгоритма не может быть целью атаки.

Доступ к реализации позволяет подбирать ключ «грубой силой».

Примерная структура теста:

1. Сохранение текущего процессорного времени.
2. Попытка взлома «грубой силой» (перебор всех возможных ключей).
3. Сохранение текущего процессорного времени.
4. Вывод времени работы.
5. Вывод результатов.

**Тестирование**

Перед запуском теста стоит определиться с целью атаки и инструментом её проведения.

Почему целью выбран ключ?

Во-первых, алгоритм шифрования доступен с самого начала и активно используется во время вскрытия.

Во-вторых, ключ обеспечивает криптоаналитика новыми важными данными:

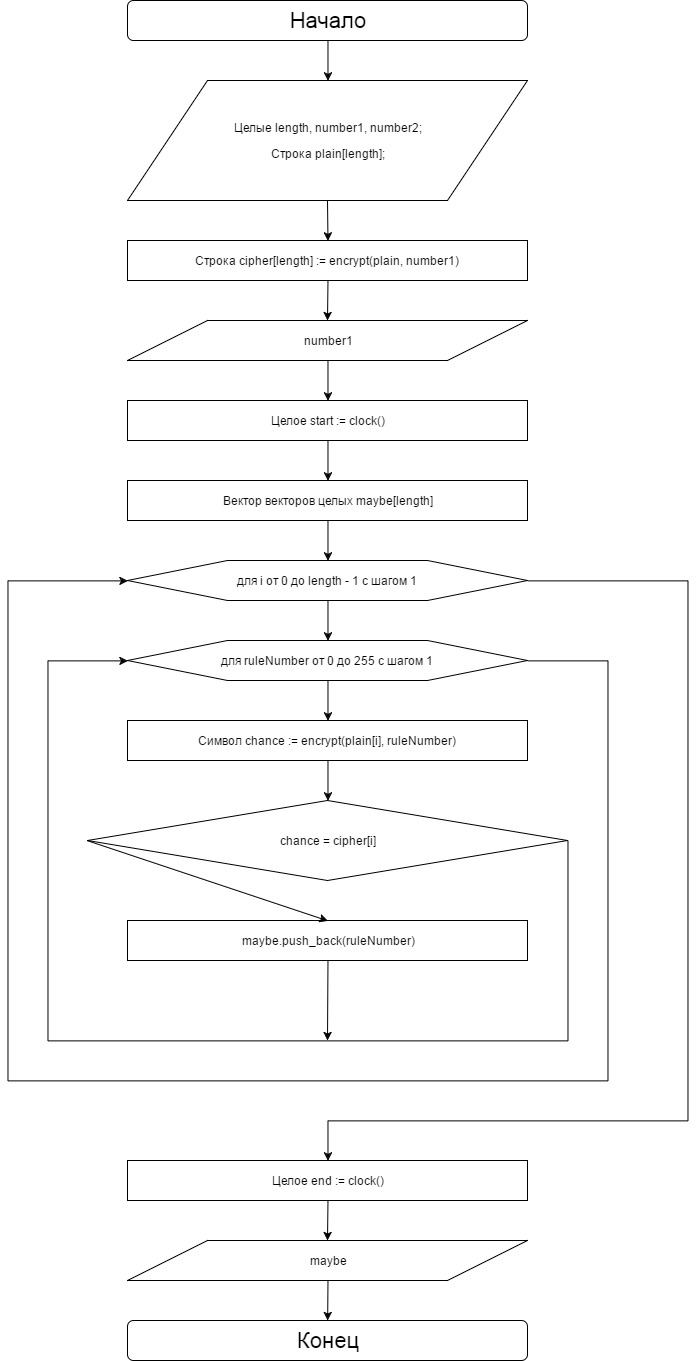
* **Список столкновений.** Список столкновений даёт взломщику возможность оптимизировать словарь (исходный символ -> шифрованный символ) для подбора открытого текста, исключив одинаково шифрующиеся символы.
* **Возможность подмены сообщения.** Так как, возможно, существуют обратимые элементарные клеточные автоматы, алгоритм может использоваться для передачи сообщений. Узнав ключ, взломщик сможет зашифровать своё сообщение и подменить им пойманное.

Почему для взлома выбран язык программирования C++?

C++ очень практичный и универсальный инструмент. Поддерживая множество стилей программирования, он не принуждает пользователя следовать одному из них, предоставляя некоторую свободу действий. C++ быстр и выразителен. В этом можно убедиться, взглянув на листинги в приложении 3.

Цель и инструмент вскрытия определены, можно начинать. На следующей странице будет описан этап взлома с известными открытыми текстами и соответствующими им шифротекстами. Исходный код теста доступен в моём репозитории [github](https://github.com/Andrew15381/Cryptographic-algorithm-based-on-linear-Cellular-Automata).

**Атака на основе открытых текстов и соответствующих им шифротекстов.**

****

**Критика взлома**

Версия программы, изображённая в виде блок-схемы, немного упрощена для удобства пояснения.

Так как шифр является блочным (см. приложение 3), можно считать, что дано сразу несколько пар текст–шифротекст. Но по отдельности эти пары бесполезны, потому что для шифрования одного символа в другой пригодны сразу несколько различных ключей. При взломе предполагается, что все символы строки зашифрованы при помощи одного ключа. Поэтому, просчитав все возможные ключи для каждого отдельного символа исходной строки, мы сможем найти ключ, общий для всей строки.

При таком подходе результат неоднозначен. В различных элементарных клеточных автоматах при определённом начальном состоянии могут возникать одинаковые поколения. То есть, как описано выше, одному зашифрованному символу соответствуют сразу несколько различных ключей. Поэтому, если ошибочный ключ встретится у каждого символа, то он попадёт в ответ. В итоге получится 2 или более равновероятных ключа. Исключить какой-либо из них по известным данным невозможно.

Что касается моей гипотезы, то она подтвердилась. При помощи дополнительного теста мне удалось найти ключи, при использовании которых не возникает столкновений. Углубившись в тему, я узнал, как при помощи этих ключей шифруется символ. Закономерность была найдена во время просмотра кодов исходных и зашифрованных символов.

Клеточный автомат 204, к примеру, совершенно никак не изменяет исходное состояние. При использовании ключа 170 код символа уменьшается вдвое. А 240-ой, наоборот, увеличивает в 2 раза. То есть, зашифровав символ ключом 170, возможно вернуть его, используя ключ 240. Если не обращать внимания на надёжность такого метода, то можно найти и другой недостаток: 170 делит код символа на 2 нацело, не сохраняя дробной части. Очевидно, что информация, хранимая в младшем бите, необратимо теряется.

**Итоги**

Подводя итоги, можно сказать, что они лишь промежуточные. Я провёл полноценное исследование своего алгоритма и его реализации. Но множество свойств ещё не изучено до конца. Хотелось бы довести действо до логического завершения.

Передо мной уже стоят некоторые планы на будущее. Зачем нужна реализация, нигде не используемая? Надеюсь, вскоре ей найдётся практическое применение как в моих, так и в других проектах. Для этого нужно сделать множество доработок и новых тестов.

А сейчас можно лишь точно сообщить – алгоритм готов к использованию. Исходный код совершенно открытый, поэтому любой желающий может модифицировать его под свои нужды.

Любой комментарий, связанный с перспективами развития, принимается по моему адресу электронной почты – [odmen@itgnews.ga](mailto:odmen@itgnews.ga).

**Литература**

* «C++ для чайников» 2003, Стефан Рэнди Дэвис
* «Программирование. Принципы и практика с использованием C++» 2016, Бьярне Страуструп
* «Прикладная криптография» 2002, Брюс Шнайер
* «Криптосистемы клеточных автоматов» 2008, Росошек, Боровиков, Евсютин – <http://cyberleninka.ru/article/n/kriptosistemy-kletochnyh-avtomatov>