**Слайд 1**

Здравствуйте, я Чистобаев Андрей Андреевич, ученик 8 математического класса Гимназии №1538, хочу презентовать вам своё исследование на тему «Алгоритм шифрования на основе элементарных клеточных автоматов».

**Слайд 2**

С давних пор существует проблема сохранения конфиденциальности данных. Особо остро она встаёт перед разработчиками современности. Имеется множество способов защиты информации, но многие из них не предоставляют необходимых результатов – производительности и криптостойкости. Тогда на помощь приходят криптографы, создавая новые и ещё более гибкие методы. Возможность их практического применения необходимо доказать. Этим я и решил заняться, создав собственный алгоритм шифрования, основанный на элементарных клеточных автоматах.

Клеточные автоматы были изобретены Джоном фон Нейманом в середине XX века, группа элементарных была выделена из них Стивеном Вольфрамом в 90-ых годах прошлого столетия. В криптографии они ещё не нашли себе широкого применения.

**Слайд 3**

На вход поступает строка символов и номер элементарного клеточного автомата. Битами двоичного представления кода каждого символа строки инициализируется начальное состояние каждой клетки поля элементарного клеточного автомата. Из двоичного представления номера получается правило перехода состояний клеток поля. Посредством этого правила просчитывается следующее поколение, которое передаётся вызывающей стороне в виде строки.

**Слайд 4**

В соответствии с принятой терминологией номер клеточного автомата назовём ключом шифрования, входную строку – открытым текстом, а результат работы – шифротекстом.

**Слайд 5**

В основном описанное преобразование является необратимым, за исключением случаев, когда используются особые пары ключей: 15-85, 51-51, 85-15, 105, 150, 170-240, 204 и 240-170. Эти пары я нашёл опытным путём, применив ко всем символам кодировки Unicode все возможные ключи, сравнивая полученные результаты с ранее сохранёнными. Замечательно, что двоичное представление некоторых особых ключей – это инвертированное представление других особых ключей. Используя их при шифровании и дешифровании, конечный пользователь избавится от коллизий, допускаемых другими ключами. В пример можно привести ключ 110, при использовании которого возникает коллизия символов t и L, одинаково шифрующихся в Ü. Из-за этого шифротексты, соответствующие словам Lime и time будут равны. Заметьте, что данная коллизия не единственная.

**Слайд 6**

Дешифрование доступно только для ранее описанных пар ключей, несмотря на то, что и для других клеточных автоматов имеются обратные. Всё дело в коллизиях символов.

**Слайд 7**

При реализации мною был использован ОО стиль программирования, позволяющий наиболее наглядно описать алгоритм. C++ выбран за его скорость и выразительность, хотя он и не имеет полноценной поддержки кодировки Unicode. Поэтому в программе символьный тип заменён целочисленным, а преобразованием кода в символ занимается вызывающая сторона. Реализация получилась достаточно производительной – в последней версии 1 символ шифруется на моей системе за 0.000001 секунды.

**Слайд 8**

Но и этот шифр не сферический в вакууме, и нужно его проверить. Для этого мною был разработан и реализован алгоритм взлома. При взломе я считал, что криптоаналитику будет известен открытый текст, соответствующий ему шифротекст и реализация алгоритма, то есть максимально доступное количество данных. Эти данные позволяют вести взлом грубой силой. То есть я последовательно применяю каждый возможный ключ к открытому тексту, и сравниваю полученный результат с данным шифротекстом, определяя ключ шифрования, необходимый для дешифровки или составления словаря. Этот способ взламывает шифр за 1 секунду (среднее на 10 запусков для строки в 10 символов).

Причиной тому является маленький диапазон значений ключа. Всего существует 256 элементарных клеточных автоматов, следовательно, ключ может принимать значение от 0 до 255 включительно. Ключ является 8битным. В век, когда взламываются алгоритмы с 1024битными ключами такого размера недостаточно. Возможно, проблема решаема последовательным шифрованием различными ключами? Тогда мешает атака “meet-in-the-middle”, изобретённая Уитфилдом Диффи и Мартином Хэллман.

Открытый текст шифруется всеми возможными 8битными ключами и результаты сохраняются. Потом расшифровывается данный шифротекст всеми возможными 8битными ключами. Если полученное значение совпало с одним из сохранённых ранее, то получаются оба ключа. Эта атака позволяет не перебирать все возможные случаи, из-за чего выполняется меньше команд за счёт увеличения используемой памяти.

**Слайд 9**

В заключение могу точно сказать, что алгоритм в настоящем виде не имеет практической ценности. Он был объективно оценён, что значит моя работа полностью выполнена.