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. Тогда на помощь приходят криптографы, создавая новые и гибкие методы. Возможность их практического применения необходимо доказать. Этим я и решил заняться, создав собственный алгоритм шифрования, основанный на элементарных клеточных автоматах.

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.

В основном описанное преобразование является необратимым, за исключением случаев, когда используются ключи 15-85, 51-51, 85-15, 105, 150, 170-240, 204 и 240-170. Используя именно эти пары ключей при шифровании и дешифровании, конечный пользователь избавится от коллизий, допускаемых другими ключами.

При реализации мною был использован ОО стиль программирования, позволяющий наиболее наглядно описать алгоритм на C++. Язык выбран за его скорость и выразительность.

При взломе я считал, что криптоаналитику будет известен открытый текст, соответствующий ему шифротекст и реализация алгоритма. 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. Тем самым определяется ключ шифрования, необходимый для дешифровки или составления словаря.

Этот способ взламывает шифр за 1 секунду (среднее на 10 запусков для строки в 10 символов). Значит, до использования в реальном проекте ещё далеко. Нужно устранить главные причины провала:

1. Единый ключ
2. Посимвольное шифрование
3. Коллизии
4. Маленький диапазон значений ключа

Исходный код открыт и доступен любому желающему.