2008 Математические методы криптографии № 1(1)
DOI 10.17223/20710410/1/8
УДК 621.391

КРИПТОСИСТЕМЫ КЛЕТОЧНЫХ АВТОМАТОВ

С.К. Росошек, А.А. Боровков, О.О. Евсютин

Томский государственный университет,
Томский государственный университет систем управления и радиоэлектроники

E-mail: rososhek@list.ru

В работе вводится новый класс симметричных криптосистем, в которых криптографические преобразования реализуются посредством обратимых клеточных автоматов. Приведены два конкретных шифра клеточных автоматов: один с окрестностью Мура, другой – на разбиении.

Ключевые слова: *клеточный автомат*, *окрестность*, *разбиение*, *криптосистема.*

Идея клеточных автоматов была сформулирована независимо Дж. фон Нейманом и К. Цусе в конце 40-х годов.

Клеточные автоматы являются дискретными динамическими системами, поведение которых полностью определяется в терминах локальных зависимостей. Клеточный автомат может мыслиться как стилизованный мир. Пространство представлено равномерной сеткой, каждая ячейка которой, или клетка, содержит несколько битов данных (в простейшем случае может быть один бит); время идет вперед дискретными шагами, а законы мира выражаются единственным набором правил, по которым любая клетка на каждом шаге вычисляет свое новое состояние по состояниям ее близких соседей.

Чтобы синтезировать структуры значительной сложности, необходимо использовать большое количество клеток, а для того, чтобы эти структуры взаимодействовали и существенно эволюционировали, необходимо позволить автомату работать на протяжении большого количества шагов. Изменяя начальные условия, мы будем каждый раз получать совершенно иную картину результата.

В начальном состоянии двумерного клеточного автомата мы имеем двумерную плоскость (будем называть ее полем), разделенную на ячейки (клетки), каждая из которых содержит либо единицу, либо ноль. Задано правило, по которому каждая клетка будет вычислять свое будущее состояние по состояниям соседних клеток. Если предположить, что первоначальное состояние клеточного автомата – это исходный текст, подлежащий процедуре зашифрования, то его конечное состояние будет являться шифртекстом. Правило, которое определяет зависимость будущего состояния каждой клетки от текущего состояния соседних клеток, можно изменять. То есть можно задать несколько правил, которые будут меняться в процессе работы клеточного автомата под управлением ключа шифрования.

В результате можно получить следующие преимущества над классическими шифрами:

- ключевая последовательность не будет участвовать ни в каких преобразованиях с открытым текстом, что сделает невозможным нахождение ключа путем осуществления каких-либо преобразований над открытым текстом;
- каждый бит открытого текста будет зависеть от текущего состояния соседних битов, что сделает невозможным дешифрование части сообщения и значительно усложнит криптоанализ.

1. Шифры двумерных клеточных автоматов с окрестностью Мура

Начальное состояние

Получить начальное состояние клеточного автомата можно следующим образом. Если рассматривать исходный текст как последовательность бит, то заполняем им двумерный массив (лучше квадратный) последовательно способом, описанным ниже.

Задание правил

Так как будущее состояние каждой клетки должно зависеть от текущих состояний ее ближайших соседей, то необходимо определиться, какие именно соседние клетки будут входить в окрестность каждой рассматриваемой клетки. Поскольку в двумерном массиве каждая клетка имеет 8 соседей, за исключением клеток, находящихся на границе массива, то оптимальным выбором будем называть окрестностью клетки восемь ее непосредственных соседей, то есть тех клеток, с которыми она соприкасается в массиве, а не только имеет общую сторону. Такая окрестность в теории клеточных автоматов называется окрестностью Мура.

Используем в качестве ключа случайную последовательность из не менее 16 символов, каждому из которых в кодировке ASCII соответствует 8 бит.

Если перевести все символы ключа в двоичную форму, можно создать следующую зависимость. Номер разряда элемента ключа будет соответствовать определенному соседу каждой клетки, а именно: первый разряд – «северо-западному» (для удобства, при указании сторон клетки будем пользоваться терминами сторон света) соседу, второй – «северному», третий – «северо-восточному», четвертый – «восточному» и т. д (пример изображен на рис. 1). Тогда будет производиться операция «исключающего ИЛИ» (XOR) между значением каждой клетки поля (0 или 1) и значением тех соседних ячеек, у которых соответствующий им бит в элементе (байте) ключа равен единице. То есть, если в первом разряде элемента ключа стоит единица, то операция будет произведена между значением текущей клетки и значением ее «северо-западного» соседа. Если, вдобавок, единица стоит еще и, например, в четвертом разряде элемента ключа, то полученный результат будет участвовать в операции «исключающего ИЛИ» со значением «восточного» соседа клетки. Можно сказать, что элемент (байт) ключа будет указывать каждой клетке, с какими соседями из ее окружения ей следует произвести операцию XOR.

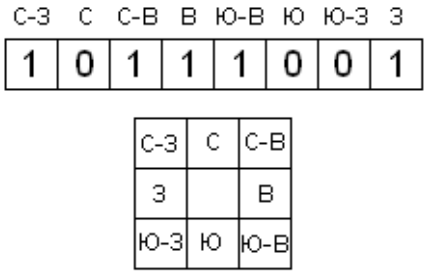

Рис. 1. Принцип соответствия номера бита в байте ключа – пространственному расположению соседей клетки

Таким образом, будет создан клеточный автомат, в котором, как и положено, будущее состояние каждой клетки будет зависеть от текущего состояния своего окружения, и на каждом шаге его работы правило, определяющее эту зависимость, будет меняться в соответствии с введенным ключом. Каждый шаг – новое правило.

Обратимость

Обратимость в теории клеточных автоматов достигается применением специальных обратимых правил.

Обратимости можно добиться, используя плоскость предыдущего состояния, значения клеток в которой учитываются при определении будущего состояния каждой клетки. Но использование этого метода на практике затрудняет необходимость хранения второй плоскости после процедуры зашифрования, так как без нее расшифрование невозможно. Но все же после проведения некоторых исследований стало ясно, что при соблюдении определенных правил работы клеточного автомата можно добиться обратимости и с использованием только одной плоскости.

Было выявлено, что при использовании данного алгоритма обратимость зависит от способа обработки граничных элементов массива, то есть тех элементов, у которых имеется меньше восьми соседей. Можно было просто заполнить их нулями, но в этом случае мы потеряли бы в криптостойкости, так как на обработку граничных элементов некоторые разные элементы ключа влияли бы одинаково. Поэтому нужно добавить недостающих соседей. Было решено заполнить их некоторой константой, генерируемой случайным образом. Чтобы добиться обратимости, эти клетки не должны изменяться на протяжении всего процесса шифрования. Если перед зашифрованием была введена некоторая константа, то перед расшифрованием должна быть введена та же константа, в противном случае расшифрование будет неверным. Независимо от константы ключ должен обеспечивать необходимый уровень защиты, но если у злоумышленника нет той константы, с использованием которой было зашифровано сообщение, это может послужить дополнительной защитой, так как ему придется ее подбирать. Учитывая большой размер блока, эта задача сравнима с перебором всех возможных ключей. Но в случае, если злоумышленник знает константу, которая использовалась при зашифровании сообщения, теоретически это может послужить уязвимостью при проведении криптоанализа. Плюс к этому константу необходимо где-то хранить для дополнительной защиты.

В ходе дальнейшего исследования был найден другой способ, с помощью которого можно добиться обратимости без потери в стойкости и необходимости хранения дополнительных данных. Это замыкание плоскости в торообразную поверхность. В этом случае недостающие соседи граничных клеток берутся с противоположной стороны массива. Но изменение этих «виртуальных» соседей должно осуществляться ди-

намически, то есть как только меняется состояние какой-либо граничной клетки, должно измениться и состояние «виртуального» соседа противоположной клетки на другой стороне массива.

Проблема полного шифрования

В любом блочном шифре существует проблема неполного последнего блока, для решения которой приходится добавлять некоторые лишние символы, чтобы заполнить блок полностью. Как правило, это создает некую уязвимость, которую можно использовать при криптоанализе.

В данном алгоритме нельзя исправить ситуацию таким образом, потому что блок, как правило, имеет очень большой размер, что необходимо для достижения высокой криптостойкости, потому как чем больше размер поля клеточного автомата, тем шире распространяются взаимные зависимости клеток и тем более непредсказуем будет конечный результат. Но благодаря свойствам клеточных автоматов был разработан алгоритм, позволяющий полностью шифровать сообщение без потери в стойкости. Он заключается в том, что сначала определяется такое сочетание размера блока шифрования (размер блока не фиксирован) и их количества, чтобы оставалась незашифрованной часть сообщения, которая может поместиться в блок размером несколько меньше установленного. Эта часть сообщения помещается в дополнительный блок, а именно в центральную часть нового двумерного массива. Перед зашифрованием последнего блока, когда почти все сообщение уже зашифровано, вокруг этой части сообщения, расположенной в центре массива, записываются клетки из последнего шифруемого блока. То есть часть сообщения из последнего шифруемого блока как будто обволакивает ту остаточную часть сообщения, находящуюся в другом блоке. В результате этого граничные клетки дополнительного массива получают соседей из другого блока. Осуществляется один шаг шифрования. Затем, после каждого шага шифрования основного блока дополнительный массив «обволакивается» уже новыми полученными клетками и снова шифруется. Таким образом, на каждом шаге шифрования граничные клетки массива с остаточной частью сообщения будут получать новых соседей из другого блока и эти соседи никак не будут зависеть от значений ячеек данного массива. Благодаря этому, несмотря на меньший размер блока, что удобнее для криптоанализа, злоумышленнику придется подбирать значения соседних клеток для всех граничных клеток массива, да еще и на каждом шаге шифрования, что на практике осуществить невозможно.

Режимы шифрования

После создания первой программной реализации описанного алгоритма стало ясно, что алгоритм работает слишком медленно и не может конкурировать в скорости с лучшими современными криптографическими алгоритмами. Тогда было принято решение реализовать, помимо побитного шифрования, режим побайтного шифрования. В этом случае сам алгоритм не изменится, за исключением того, что каждая клетка двумерного массива (поля клеточного автомата) будет содержать байт, а не бит. Тогда операция XOR производится не с битами, а с байтами.

Описание алгоритма

Генерация ключей

1. Генерируется случайная последовательность символов (алфавит содержит 255 символов), но не менее 16.
2. Каждому символу введенной ключевой последовательности ставится в соответствие уникальное число.
3. Все полученные числа переписываются в двоичном представлении (результат – последовательность нулей и единиц длиной $8N$, где N – количество символов в ключе).

Алгоритм зашифрования

1. Вычисляется сочетание размера и количества блоков, удовлетворяющее следующим условиям:

• Для режима побитного шифрования –

$$(n-2)^2 / 8 \cdot c = f \tag{1}$$

или

$$(n-2)^2 / 8 \cdot c \geq f - \left((n-4)^2 / 8\right), \tag{2}$$

где n – длина блока; c – количество блоков; f – длина сообщения.

Условие (1) выполняется, когда можно подобрать такое сочетание длины и количества блоков, что сообщение можно полностью разбить на такие блоки без остатка. В противном случае должно выполняться условие (2).

• Для режима побайтного шифрования –

$$(n-2)^2 \cdot c = f \tag{3}$$

или

$$(n-2)^2 \cdot c \geq f - (n-4)^2 . \tag{4}$$

Число $(n-2)^2$ в режиме побайтного шифрования (для побитного – $(n-2)^2/8$) равняется количеству символов, помещающихся в один блок. При этом граничные клетки остаются пустыми для того, чтобы записать в них символы константы, или для имитации торообразной поверхности. Числа $(n-2)^2$ и $((n-2)^2/8)$ равняются максимальному количеству остаточных символов, помещающихся в дополнительный блок соответствующего размера. Здесь граничные клетки используются для записи в них значений из последнего основного блока.

2. Каждому символу сообщения ставится в соответствие уникальное число:

• для режима побитного шифрования: число, соответствующее текущему символу, приводится к двоичному виду и записывается в текущие 8 клеток поля, а граничные клетки массива остаются нетронутыми;

• для режима побайтного шифрования: число, соответствующее текущему символу, непосредственно записывается в текущую клетку массива, а граничные клетки массива остаются нетронутыми.

3. Пока блок не будет заполнен полностью, выполняется п. 2.

4. Заполняются граничные клетки массива:

• используется константа:

- режим побитного шифрования: каждому символу сгенерированной константы ставится в соответствие уникальное число; это число приводится к двоичному виду и записывается в 8 граничных клеток;

- режим побайтного шифрования: каждому символу сгенерированной константы ставится в соответствие уникальное число и это число непосредственно записывается в текущую граничную клетку;

• осуществляется замыкание в торообразную поверхность: в каждую граничную клетку записывается значение клетки, расположенной на противоположной стороне двумерного массива (не являющаяся граничной клеткой).

5. Если верно условие (2) или (4) и выполняется зашифрование последнего блока сообщения, то заполняется дополнительный массив для остатка сообщения, который не войдет в основной цикл шифрования.

Выполняются п. 2 – 3 применительно к дополнительному блоку и остатку сообщения.

6. Если справедливо условие (2) или (4) и выполняется зашифрование последнего блока сообщения, то заполняются граничные клетки вокруг заполненных ячеек дополнительного массива, содержащего остаток сообщения, который не вошел в основной цикл шифрования.

Вокруг заполненных ячеек дополнительного массива клетки заполняются значениями соответствующих ячеек из основного блока шифрования.

7. Выполняется преобразование значения текущей клетки (граничные клетки не преобразуются).

Производится операция XOR между значением текущей клетки поля и значениями тех соседних ячеек, у которых соответствующий им бит в элементе ключа равен единице.

Соответствие номера разряда элемента ключа следующее: первый разряд соответствует «северо-западному» соседу, второй – «северному», третий – «северо-восточному», четвертый – «восточному», пятый – «юго-восточному», шестой – «южному», седьмой – «юго-западному», восьмой – «западному».

8. Если шифрование выполняется с замыканием плоскости в торообрзаную поверхность, то после преобразования клетки, имеющей общую сторону с граничной клеткой, полученное значение записывается в граничную клетку на противоположной стороне массива.

9. Пока все клетки массива не прошли преобразование, выполняются п. 7 – 8.

10. Если имеет место условие (2) или (4) и выполняется зашифрование последнего блока сообщения, то снова выполняется п. 6.

11. Если выполняется условие (2) или (4) и производится зашифрование последнего блока сообщения, то выполняется п. 7 применительно к дополнительному массиву, пока все клетки дополнительного массива не пройдут процедуру преобразования.

12. Берется следующий элемент ключевой последовательности.

13. Пока ключевая последовательность не кончится, выполняются п. 7 – 12.

14. Вывод шифртекста:

• для режима побитного шифрования: из массива берутся значения восьми текущих клеток, затем эта последовательность приводится к десятичной форме, а полученное число заменяется символом, который соответствуют этому числу, и записывается в шифртекст, причем значения граничных клеток не берутся;

• для режима побайтного шифрования: из массива берется значение текущей клетки, полученное число заменяется символом, который соответствует этому числу, и записывается в шифртекст, причем значения граничных клеток не берутся.

15. Пока массив не кончится, выполняется п. 14.

16. Если сообщение зашифровано не полностью, осуществляется зашифрование следующего блока сообщения, то есть выполняются п. 2 – 16.

Алгоритм расшифрования

Алгоритм расшифрования полностью идентичен алгоритму зашифрования, за исключением следующих моментов:

- элементы ключа берутся в обратном порядке;
- элементы массива обрабатываются в обратном порядке;
- п. 6 алгоритма зашифрования переносится на место п. 10, то есть первое заполнение граничных клеток дополнительного массива происходит после первого раунда расшифрования основного блока, а не до него.

Очень важным свойством данного криптографического алгоритма является неучастие ключа в преобразованиях сообщения. Обычно ключ участвует в некоторых операциях преобразования с битами шифруемого сообщения, что является потенциальной уязвимостью. На этом основан один из самых успешных методов криптоанализа – линейный криптоанализ. Благодаря популярному в наши дни методу отказа оборудования удается взламывать даже самые стойкие шифры. С его помощью удается выявить следы взаимодействия битов ключа с битами шифруемого сообщения на каком-либо шаге работы алгоритма. Но здесь ключ не участвует в операциях преобразования с битами шифруемого сообщения, он только указывает каждому биту (байту) сообщения, с какими именно соседями производить операцию XOR. Вследствие этого линейный криптоанализ к данному шифру просто неприменим, а метод отказа оборудования, скорее всего, не сможет обнаружить никаких следов взаимодействия битов ключа с битами сообщения, потому как их там просто не будет.

Учитывая все вышесказанное, можно утверждать, что криптосистема, разработанная на базе клеточных автоматов, может получить серьезные преимущества в криптостойкости и, возможно, потребуется разработка новых методов криптоанализа для подобных шифров.

2. Двумерные клеточные автоматы на разбиении

Клеточные автоматы на разбиении характеризуются следующим образом.

1. Решетка клеточного автомата разбита на множество конечных однородных частей – *блоков*.

2. *Правило клеточного автомата* задается таким образом, что оно *рассматривает и изменяет содержимое всего блока*, а не отдельной клетки. Одно и то же правило применяется ко всем блокам. Блоки не зависят друг от друга.

3. *Разбиение меняется на каждом шаге*, чтобы разные блоки могли взаимодействовать между собой [1].

Простейшая схема разбиения, когда блоки имеют размер 2 × 2 клетки, а шаги, в которых блоки, соответствующие четной решетке, чередуются с шагами, использующими нечетную решетку, называется окрестностью Марголуса.

В данной работе рассматривается вариант такой схемы разбиения, когда разбиение ограничивается четной и нечетной решетками, но блоки могут иметь больший размер.

Клеточный автомат на разбиении будет обратимым в том случае, если правило клеточного автомата устанавливает взаимно однозначное соответствие между старым и новым состояниями блока. То есть правило представлено в виде таблицы, состоящей из двух столбцов, в первом из которых записаны все возможные текущие состояния блока, а во втором – соответствующие им новые состояния, при этом второй столбец представляет собой перестановку первого столбца. Тогда для того, чтобы развернуть систему в обратном направлении, необходимо поменять столбцы в таблице местами, учитывая при этом разбиение решетки клеточного автомата.

Таким образом, был получен класс обратимых правил, на основе которых разрабатывается новый шифр.

В ходе работы над созданием шифра были написаны несколько программ, некоторые из них представлены ниже.

В программе «Машина клеточных автоматов» реализованы обратимые клеточные автоматы на разбиении, т.е. тот класс клеточных автоматов, о котором говорилось выше.

Клеточные автоматы, реализованные в программе, имеют следующие характеристики:

- решетка клеточного автомата имеет размер 60 × 60 клеток;
- так же как и в окрестности Марголуса, для разбиения решетки клеточного автомата на блоки используются четная и нечетная решетки;
- блоки имеют размер от 2 × 2 до 5 × 5 клеток.

Содержимое одного блока решетки представляется в виде двоичного числа. Каждая клетка решетки клеточного автомата может принимать два состояния, то есть представляет собой бит данных. Чтобы получить значение содержимого блока, необходимо просмотреть значения всех его клеток слева направо и сверху вниз. Таким образом, определяются все биты искомого двоичного числа от самого старшего до самого младшего.

Правило клеточного автомата задается в виде таблицы, состоящей из двух столбцов. В первом столбце перечислены все возможные состояния блока, числа от 0 до $2^{ab} - 1$ в двоичном представлении (a – высота

блока, b – ширина). Второй столбец является перестановкой первого, причем перестановка эта задается простым циклическим сдвигом первого столбца таблицы правил на определенное число позиций.

Программа «Машина клеточных автоматов» дает возможность пользователю, работающему с ней, проследить развитие конкретного клеточного автомата во времени в течение заданного числа шагов.

В программе «Криптосистема клеточных автоматов на разбиении» реализован уже непосредственно сам шифр на основе клеточных автоматов на разбиении.

Развитие клеточного автомата происходит под управлением следующих параметров:

- высота блоков решетки клеточного автомата (a);
- ширина блоков решетки клеточного автомата (b);
- сдвиг в таблице правил (p);
- число шагов развития клеточного автомата во времени (n).

Сдвиг в таблице правил напрямую зависит от размеров блока. Это число, которое выбирается в интервале $1 \le p \le 2^{a \cdot b} - 1$. Число шагов n не взаимосвязано ни с какими другими параметрами.

Ключ разрабатываемого шифра необходимо выбирать таким образом, чтобы все указанные параметры клеточного автомата входили в его состав.

Конечным результатом работы является блочный симметричный шифр на основе клеточных автоматов на разбиении. Шифрование происходит следующим образом: открытый текст разбивается на блоки одинакового размера, соответствующего размеру решетки клеточного автомата, каждый блок используется в качестве начального состояния решетки клеточного автомата, предварительно он должен быть представлен в двоичном виде, затем происходит развитие клеточного автомата во времени под управлением заданного ключа, шифртекстом будет являться конечное состояние решетки клеточного автомата. Расшифрование происходит аналогичным образом: в качестве начального состояния решетки клеточного автомата используется шифртекст, открытым текстом соответственно будет являться конечное состояние решетки клеточного автомата.

На данный момент были разработаны два шифра на основе клеточных автоматов на разбиении. В качестве примера приведем один из них.

Шифр является блочным симметричным шифром, который построен на основе клеточного автомата на разбиении с размерами блока 3×3 клетки (решетка клеточного автомата имеет размер 60×60 клеток) и имеет следующее описание.

1. *Генерация ключей.*

Ключом является случайно сгенерированная битовая строка размером 256 бит.

2. *Алгоритм зашифрования.*

1) Открытый текст разбивается на отдельные блоки по 450 символов, полученные блоки последовательно зашифровываются с помощью одного и того же ключа.

2) Текущий блок открытого текста представляется в двоичном виде, для этого берется ASCII-код каждого символа блока и переводится из десятичной системы в двоичную.

3) Полученной битовой строкой заполняется решетка клеточного автомата, это будет ее начальным состоянием.

4) Клеточный автомат развивается в течение 256 шагов (число шагов равно длине ключа).

5) На каждом шаге развития клеточного автомата разбиение определяется из ключа: если на данном шаге соответствующий бит ключа равен 0, то разбиение на данном шаге соответствует четной решетке, если 1 – разбиение соответствует нечетной решетке. Сдвиг в таблице правил (параметр клеточного автомата p) в двоичном представлении имеет размер 9 бит и определяется из ключа следующим образом: ключ разбивается на две половины, из 4-х байт слева от условной границы берутся младшие биты, из 4-х справа – старшие биты, таким образом определены 8 бит из 9, а последний 9-й бит определяется с помощью операции XOR между ними.

6) Конечное состояние решетки клеточного автомата является блоком шифртекста, соответствующим текущему блоку открытого текста.

7) Полученный блок шифртекста переводится из двоичного в символьное представление.

8) Шаги 2 – 7 повторяются для всех блоков, на которые разбит открытый текст.

3. *Алгоритм расшифрования.*

1) Шифртекст разбивается на отдельные блоки по 450 символов, полученные блоки последовательно расшифровываются с помощью одного и того же ключа.

2) Текущий блок шифртекста представляется в двоичном виде, для этого берется ASCII-код каждого символа блока текста и переводится из десятичной системы в двоичную.

3) Полученной битовой строкой заполняется решетка клеточного автомата, это будет ее начальным состоянием.

4) Клеточный автомат развивается в течение 256 шагов (число шагов равно длине ключа).

5) Данный этап отличается от соответствующего этапа в алгоритме зашифрования только тем, что ключ необходимо просматривать, начиная с конца, а не с начала.

6) Конечное состояние решетки клеточного автомата является блоком исходного текста, соответствующим текущему блоку шифртекста.

7) Полученный блок исходного текста переводится из двоичного в символьное представление.

8) Шаги 2 – 7 повторяются для всех блоков, на которые разбит шифртекст.

Описанный шифр реализован программно в новой версии компьютерной программы «Криптосистема клеточных автоматов на разбиении».

В заключение заметим, что разработанные нами криптосистемы на базе трехмерных клеточных автоматов не вошли в данную статью из-за ограниченности ее размера.

ЛИТЕРАТУРА

1. *Тоффоли Т.*, *Марголус Н.* Машины клеточных автоматов. М.: Мир, 1991. 280 с.