2. *Beatrice D. and Thomas H.* On-line Handwritten Signature Verification using Machine Learning Techniques with a Deep Learning Approach. Master's Theses in Math. Sciences, Lund University, 2015. 90 p.
3. *McCabe A., Trevathan J., and Read W.* Neural network-based handwritten signature verification // J. Computers. 2008. V. 3. No. 8. P. 9–22.

УДК 519.7 DOI 10.17223/2226308X/10/32

САМОПРОГРАММИРУЕМЫЕ КЛЕТОЧНЫЕ АВТОМАТЫ В КРИПТОГРАФИИ

А. А. Ефремова, А. Н. Гамова

Рассмотрены и реализованы различные виды самопрограммируемых клеточных автоматов. Проведено исследование возможности их применения в качестве генератора псевдослучайных чисел. В результате тестирования получено, что самопрограммируемые клеточные автоматы могут применяться в качестве генератора псевдослучайных чисел в криптографии. Для улучшения криптостойкости данного генератора могут быть предложены следующие методы: 1) учёт значения ячейки не в каждый момент времени, а через разные отрезки; 2) применение техники клеточного программирования для подбора используемых правил; 3) комбинирование одномерных и двумерных клеточных автоматов; 4) увеличение числа ячеек и радиуса окрестности.

Ключевые слова: *клеточный автомат, самопрограммируемый клеточный автомат, генератор псевдослучайных чисел, криптография.*

В задачах криптографии часто применяются псевдослучайные последовательности, которые должны быть неотличимы от истинно случайных по своим статистическим свойствам. Для выработки таких последовательностей используют специальные алгоритмы — генераторы псевдослучайных последовательностей. К настоящему времени разработано большое количество таких алгоритмов, основанных на использовании теории чисел, свойствах различных алгебраических систем, применении конечных (в том числе клеточных) автоматов и т. д.

Впервые клеточные автоматы (КлА) были применены в качестве генератора псевдослучайных чисел (ГПСЧ) С. Вольфрамом [1]. Он использовал однородные одномерные КлА с радиусом окрестности $r = 1$ по правилу 30.

Правило 30 задаётся формулой $s_i(t+1) = s_{i-1}(t) \oplus (s_i(t) \vee s_{i+1}(t))$, где $s_i(t)$ — состояние ячейки i в момент времени t. Правило 30 так называется потому, что 30 — десятичное представление вектора значений соответствующей булевой функции (табл. 1).

Таблица 1

Получение кода описания правила 30

$s_{i-1}(t)s_i(t)s_{i+1}(t)$	111	110	101	100	011	010	001	000
$s_i(t+1)$	0	0	0	1	1	1	1	0

С первого взгляда кажется, что восстановить ключ (начальное состояние автомата) по фрагменту последовательности сложно, но известно наличие успешных атак за приемлемое время. Атаки основываются на восстановлении правых (левых) смежных подпоследовательностей последовательности, полученной с помощью ячейки i.

В дальнейшем было обнаружено, что генераторы на основе неоднородных автоматов с применением правил 90 и 150 получаются более стойкими ГПСЧ и при использовании именно этих правил достигается максимальный период [2].

В [3] показано, что как ГПСЧ правило 30 плохо проходит тест на критерий согласия Пирсона (критерий χ^2) в сравнении с другими ГПСЧ, реализованными на основе неоднородных клеточных автоматов, и предложено использовать неоднородные одномерные КлА с $r = 1$ с применением четырёх правил (90, 150, 105 и 165), которые предоставляют хорошие псевдослучайные последовательности и огромное количество секретных ключей, сложных для криптоанализа.

С. Нанди предложил неоднородный автомат с применением правил 51, 153 и 195 с нулевыми граничными условиями для применения в криптографии [4], но такие криптосистемы были взломаны С. Блэкберном [5].

В [6] построен одномерный КлА, в котором правило перехода для каждой ячейки изменяется динамически в зависимости от состояний её соседних ячеек.

Самопрограммируемый клеточный автомат (СПКлА) — автомат, в котором правило перехода изменяется в зависимости от значений ячеек данного или смежного автомата. Использование динамически изменяемых правил позволяет избежать шаблонов, возникающих, когда ячейки имеют постоянные правила.

Несмотря на то, что было выявлено много неоднноднородных КлА (НКлА), которые являются качественными ГПСЧ и проходят большое количество тестов на случайность, НКлА менее пригоден для криптографических целей, чем СПКлА, так как каждая его ячейка имеет статическое правило перехода, что в некоторых случаях может упростить криптоаналитику задачу [7].

СПКлА может быть реализован как система двух связанных автоматов. Один из них (нижний) зависит от другого (верхнего) при выборе правила, которое нужно применить нижнему на следующем шаге. Каждая ячейка нижнего автомата ориентируется на значение соответствующей ячейки верхнего автомата при выборе правила, т. е. если значение ячейки верхнего автомата 0, то ячейка нижнего автомата переходит в следующее состояние по одному правилу, а если 1 — по другому. Использование верхнего автомата для изменения правил добавляет случайности автомату.

В данной работе реализовано несколько одномерных СПКлА(1), отличающихся только количеством ячеек, по правилам 165 и 150. В качестве автомата, определяющего правила перехода для данного автомата, используем КлА, развивающийся по правилам 90 и 150, которые применяются чередованием.

Правило 90: $s_i(t+1) = s_{i-1}(t) \oplus s_{i+1}(t)$.

Правило 150: $s_i(t+1) = s_{i-1}(t) \oplus s_i(t) \oplus s_{i+1}(t)$.

Правило 165: $s_i(t+1) = 1 \oplus s_{i-1}(t) \oplus s_{i+1}(t)$.

Оба автомата, определяющие СПКлА, имеют нулевые граничные условия и состоят из одинакового количества ячеек. В качестве выходных последовательностей берём три последовательности, образованные значениями соответственно одной, четырёх и восьми ячеек за один шаг работы.

Полученные последовательности подвергаются ряду тестов из пакета Diehard [8]. Результатом работы одного теста является одно или несколько p-значений; p-значение является значением статистики данного пакета и должно иметь равномерное распределение, если последовательность содержит независимые случайные биты. Тест считается пройденным успешно, если значение p лежит в пределах от нуля до единицы.

Для сравнения протестированы псевдослучайные последовательности, полученные с помощью СПКлА(2) без использования верхнего автомата по правилам 150/165, образованные значениями одной ячейки.

На основе полученных результатов можно сделать вывод, что качество последовательностей, выдаваемых автоматами, ухудшается с увеличением числа ячеек, участвующих в формировании выходной последовательности, а это значит, что время, за которое генерируется стойкая последовательность, будет возрастать.

По результатам тестирования, представленным в табл. 2 и 3, видно, что распределение p-значений СПКлА(1) при выходной последовательности, образованной значениями четырёх ячеек за каждый шаг, лучше, чем у СПКлА(2), а это значит, что СПКлА(1) более эффективен по времени работы.

Таблица 2

Результаты экспериментов для последовательностей, полученных с помощью СПКлА(1)

Количество ячеек автомата	20/20	20/20	20/20	21/21	21/21	21/21	22/22	22/22	22/22
Номер последовательности	1	2	3	1	2	3	1	2	3
Количество пройденных тестов	8	7	6	16	16	8	18	18	9

Таблица 3

Результаты экспериментов для последовательностей, полученных с помощью СПКлА(2)

Количество ячеек автомата	22	23	24	25	26	27
Количество пройденных тестов	3	13	12	3	18	6

Из последовательностей, полученных с помощью СПКлА(2), только автомат с количеством ячеек 26 смог пройти все тесты, а время работы такого автомата в 1,5 раза больше, чем у СПКлА(1) при использовании четырёх ячеек для формирования выходной последовательности.

Графические результаты анализа распределения p-значений сгенерированных последовательностей представлены на рис. 1–3. Они полностью соответствуют результатам в табл. 2 и 3. Видно, что распределение p-значений первой и второй последовательности автоматов, состоящих из 21 и 22 ячеек, близко к равномерному, то есть это стойкие последовательности; для третьей последовательности с увеличением количества ячеек результаты улучшаются незначительно.

Для второго автомата графические результаты тоже соответствуют таблице. Как видно из рис. 3, автоматы с количеством ячеек 23, 24, 26 имеют распределение p-значений, близкое к равномерному, а автоматы с количеством ячеек 22, 25, 27 им уступают.

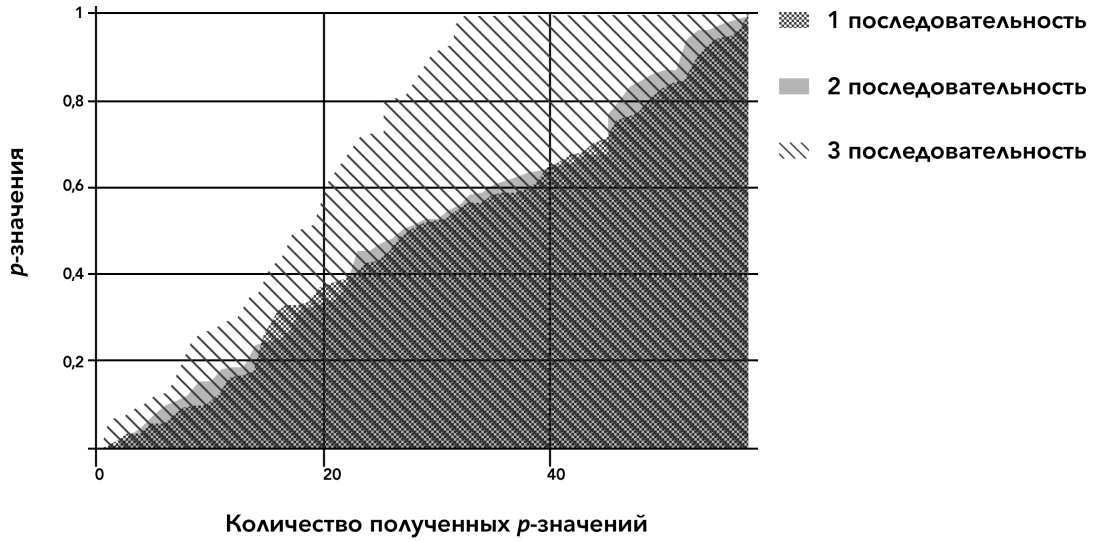

Рис. 1. Распределение p-значений для СПКлА(1) с количеством ячеек 21

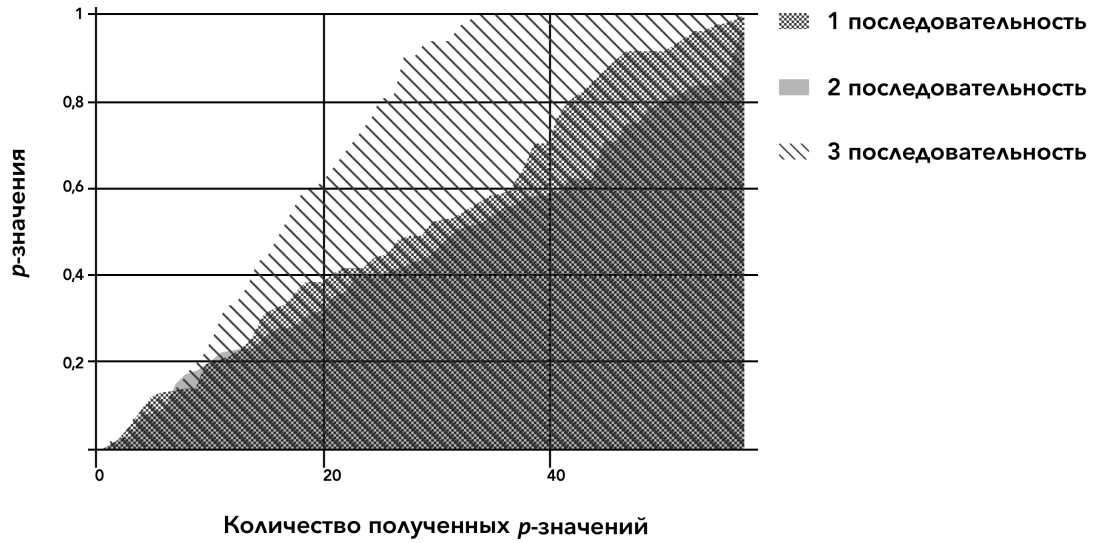

Рис. 2. Распределение p-значений для СПКлА(1) с количеством ячеек 22

Стоит отметить, что СПКлА(2) менее пригоден для криптографических целей, чем СПКлА(1), так как правило каждой его ячейки зависит только от её же значения.

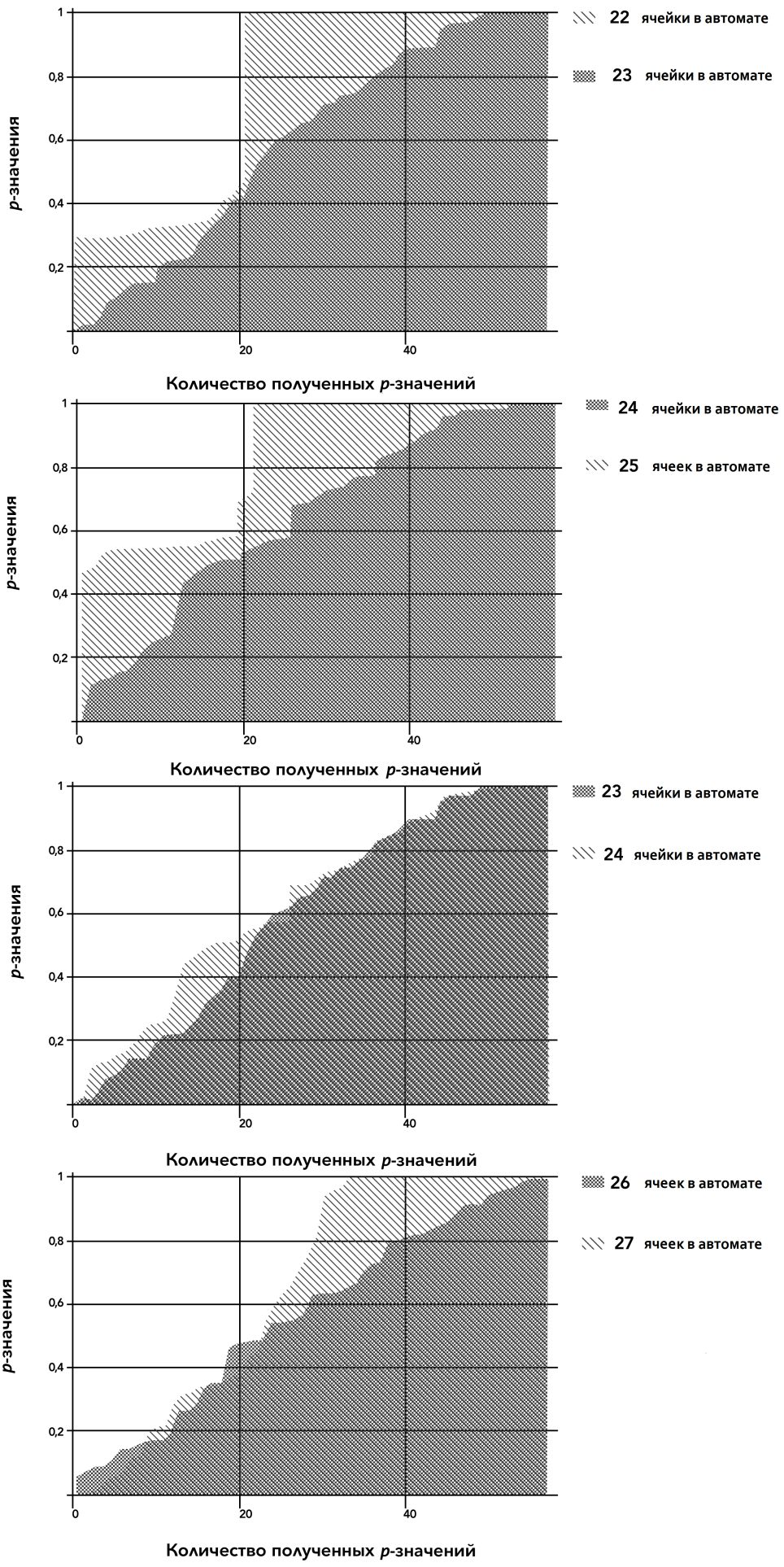

Рис. 3. Распределение p-значений для СПКлА(2) с разным количеством ячеек

ЛИТЕРАТУРА

1. *Wolfram S.* A New Kind of Science. Wolfram Media, 2002. 1192 p.
2. *Seredinski F., Bouvry P., and Zomaya A. Y.* Cellular automata computations and secret key cryptography // Parallel Computing. 2004. V. 30. P. 753–766.
3. *Tomassini M. and Perrenoud M.* Nonuniform cellular automata for cryptography // Complex Systems. 2000. No. 12. P. 71–81.
4. *Nandi S., Kar B., and Chaudhuri P.* Theory and applications of cellular automata in cryptography // IEEE Trans. Comput. 1994. V. 43. No. 12. P. 1346–1357.
5. *Blackburn S. R., Murphy S., and Paterson K. G.* Comments on "Theory and Applications of Cellular Automata in Cryptography" // IEEE Trans. Comput. 1997. V. 46. No. 5. P. 637–638.
6. *Guan S. U. and Tan S. K.* Pseudorandom number generation with self-programmable cellular automata // IEEE Trans. Computer Aided Design of Integrated Circuits and Systems. 2004. V. 23. No. 7. P. 1095–1101.
7. *Ефремова А. А., Гамова А. Н.* Генератор псевдослучайных чисел на основе клеточных автоматов // Материалы Междунар. науч. конф. «Компьютерные науки и информационные технологии». Саратов: СГУ, 2016. С. 131–134.
8. *Marsaglia G.* Diehard: A Battery of Tests of Randomness. 1995. `http://stat.fsu.edu/pub/diehard/`

УДК 517.19 DOI 10.17223/2226308X/10/33

ПОСТРОЕНИЕ $(4, 8)$-СХЕМЫ ВИЗУАЛЬНОЙ КРИПТОГРАФИИ НА ОСНОВЕ КЛАССА ЛИНЕЙНЫХ ХЭШ-ФУНКЦИЙ

Н. А. Зорина, Ю. В. Косолапов

С использованием визуальной криптографии строится $(4, 8)$-схема разделения секрета, представляющего собой чёрно-белое изображение. Для её построения применяется $(4, 4)$-схема визуальной криптографии и класс линейных хэш-функций. Показано, что использование этого класса позволяет построить стойкую схему с приемлемой относительной контрастностью восстанавливаемого секретного чёрно-белого изображения.

Ключевые слова: *визуальная криптография, линейные хэш-функции.*

В [1] М. Наором и А. Шамиром предложена (k, n)-схема разделения секрета, представляющего собой чёрно-белое изображение. В основе этой схемы лежит построение на основе исходного изображения n таких чёрно-белых («теневых») изображений, что при совмещении любых k из них (и более) можно восстановить исходное секретное изображение. При этом «совмещение» теневых изображений следует представлять как наложение этих изображений, нанесённых на прозрачную пленку, а «восстановление» — просмотр совмещённых теневых изображений на свет. Свою схему М. Наор и А. Шамир назвали схемой визуальной криптографии. В [1] сначала строятся (k, k)-схемы, а затем на их основе — (k, n)-схема $(n > k)$, в частности, с использованием k-универсальных хэш-функций. В настоящей работе ставится задача построения $(4, 8)$-схемы на основе класса линейных хэш-функций [2].