3. Baumann, R., Cavin, S., & Schmid, S. (2006). Voice over IP – security and SPIT. Swiss Army FU Br 41, KryptDet Report, Zurich: University of Berne.
4. Sengar, H., Wijesekera, D., & Jajodia, S. (2008). Detecting VoIP floods using the Hellinger distance. IEEE Transactions on Parallel and Distributed Systems, Vol. 19, No. 6, pp. 794–805.
5. Ziemmermann, P. (2008). ZRTP: Media path key agreement for secure RTP draft-zimmermann-avt-zrtp-06. IETF draft. Алынған http://tools.ietf.org/html/draft-zimmermann-avt-zrtp-06
6. Thermos, P. & Takanen, A. (2008). Securing VoIP Networks. Pearson Education, Inc.

УДК 519.713

ОБ АЛГОРИТМАХ, ОПРЕДЕЛЯЮЩИХ ОБРАТИМЫЕ КОНЕЧНЫЕ АВТОМАТЫ С ПАМЯТЬЮ

Шахметова Гульмира Балтабаевна
sh_mira2004@mail.ru
Докторант 3 курса специальности «6D060200-Информатика»
Кафедра «Информатика и информационная безопасность»
Факультет «Информационные технологии»
ЕНУ им. Л.Н.Гумилева, Нур-Султан, Казахстан
Научный руководитель – Ж.С.Сауханова

На сегодняшний день, информационные технологии являются неотъемлемой частью всех сфер жизнедеятельности общества, например, таких как, электронное правительство, банковские транзакции, связь, телевидение и многое другое, что ведет за собой крайне важную необходимость в эффективных средствах обеспечения защиты информации и данных. Такие средства относятся к криптографии, которая отвечает за конфиденциальность, проверку подлинности, целостности и неотрицания авторства [1]. Следует отметить, что рост использования компьютеров и мобильных устройств во всех аспектах жизни человечества, особенно в коммуникации, привел к возникновению новых форм криптографии. Одним из таких направлений является конечно-автоматная криптография.

Идея конечно-автоматной криптографии заключается в интегрировании теории автоматов в классическую криптографию [2]. В работах [3, 4] обсуждены основные моменты применения конечных автоматов в качестве криптографических алгоритмов и их компонент. Как известно в криптографии можно использовать как автоматы без выхода [5], так и автоматы с выходом. В данной статье будут рассмотрены конечные автоматы с памятью. Авторами была поставлена цель продемонстрировать потенциальную возможность использования данного вида конечных автоматов в криптографических системах.

Конечный автомат (КА) – это абстрактное математическое устройство, работающее в дискретном времени. В формальном виде КА представлен как пятерка $M = < X, Y, S, \delta, \lambda >$, где X $= \{x_1, x_2, \dots, x_n\}$ - конечное множество входных символов, $Y = \{y_1, y_2, \dots, y_m\}$ – конечное множество выходных символов, $S = \{s_1, s_2, \dots, s_l\}$ – конечное множество внутренних состояний, δ: $S \times X \rightarrow S$– функция переходов; λ: $S \times X \rightarrow Y$– функция выходов.

Для заданий автоматов удобнее всего использовать табличное представление (см. таб 1), где в строках записываются текущее состояние (ТС), в столбцах указываются входные символы, а на пересечении строки и столбца – следующее состояние (СС), и через запятую выходной символ.

Таблица 1. Таблица состояний КА

ТС	**СС,** y_m		
	x_1	…	x_n

s_1	$\delta(s_1, x_1), \lambda(s_1, x_1)$	…	$\delta(s_1, x_n), \lambda(s_1, x_n)$
…	…	…	…
s_l	$\delta(s_l, x_1), \lambda(s_l, x_1)$	…	$\delta(s_l, x_n), \lambda(s_l, x_n)$

КА M определяется как *КА с памятью порядка* μ, если текущее состояние M может быть однозначно определено из знания последних входных символов μ и соответствующих выходных символов μ, где μ является наименьшим целым числом. В свою очередь, память КА может быть *входной* и *выходной.* В том случае, когда текущее состояние M может быть однозначно определено из знаний только последних входных символов μ, говорят, что M является *КА с входной памятью порядка* μ. В случае, когда знания о последних μ выходных символов достаточно для однозначного определения состояния M в некоторый момент времени за последние μ переходов, то КА M называется *КА с выходной памятью порядка* μ [6].

Как известно, основная задача криптографического шифрования заключается в обратимых преобразованиях передаваемого текста с помощью специальных ключей. В конечно-автоматной криптографии, такими ключами являются КА. Отсюда следуют, что применяемые в шифровании/дешифровании КА должны восстанавливать входную последовательность символов, зная выходную последовательность, т.е. КА должны быть *обратимыми*.

Согласно [7] автоматы могут быть *обратимыми* с *нулевой задержкой*, *слабо* или *сильно обратимыми с конечной задержкой*. К первым относятся КА, функция выходов которых инъективна в каждом состоянии, ко вторым - КА, восстанавливающие входную последовательность по начальному состоянию и с задержкой по выходной последовательности, к третьим – КА, восстанавливающие входную последовательность только по выходной последовательности. Наибольший интерес в криптографии вызывают автоматы из второй группы. Данные автоматы так же называются *КА без потери информации*.

Стоит отметить, что не все КА могут быть применены в шифровании/дешифровании информации, поэтому существуют специальные алгоритмы, определяющие их принадлежность к нужной группе автоматов. К таким алгоритмам относятся: тест на проверку наличия памяти и задержки, определение наличия входной и выходной памяти, тест на обратимость [6].

Алгоритм 1. *Тест КА на наличие конечной памяти и задержки.*

1. Пусть дан КА $M =< X, Y, S, \delta, \lambda >$, представленный в виде таблицы состояний. В нашем случае X и $Y \in \{0,1\}$.

2. Строится тестирующая таблица (ТТаб), состоящая из 2 частей.

a. Верхняя часть ТТаб: Переписывается таблица состояний КА, в заголовках столбцов записываются всевозможные комбинации символов x_n/y_m (0/0, 0/1, 1/0, 1/1), в ячейках, следующие состояния, соответствующие этим комбинациям. В случае отсутствия перехода в следующее состояние, ставится прочерк " – ".

b. Нижняя часть ТТаб: Заголовки строк – это всевозможные неупорядоченные пары состояний, а в ячейках – соответствующие им следующие состояния. Если запись на пересечении строк s_i и s_j и столбцов x_n/y_m в верхней половине ТТаб равна s_p и s_q соответственно, то запись в ячейке строки $s_i s_j$ и столбца x_n/y_m нижней половинки ТТаб равна $s_p s_q$. Если для некоторой пары состояний s_i и s_j одна или обе соответствующие записи в некотором столбце x_n/y_m являются прочерк, то запись в строке $s_i s_j$ и столбца x_n/y_m, является прочерком.

3. Строится тестирующий ориентированный взвешенный граф G по ТТаб из шага 2.

a. Каждой совместимой паре состояний $s_i s_j$ соответствует вершина в G;

b. Дуга с надписью y_m выходит из вершины $s_i s_j$ в вершину $s_p s_q$, где p ≠ q, тогда и только тогда, когда совместимая пара $(s_p s_q)$ является следующим состоянием совместимой пары $(s_i s_j)$.

4. Определяют отношение исходного КА к группе КА с памятью и находят его задержку.

a. Граф G из шага 3 проверяется на наличие цикла. В случае если цикл имеется, то КА не является автоматом с конечной памятью, в случае отсутствия цикла – КА является автоматом с конечной памятью.

b. Находят задержку $\mu = l+1$, где l - самый длинный путь графа G из шага 3.

Алгоритм 2. *Тест КА на наличие входной конечной памяти.*

1. Пусть дан КА $M =< X, Y, S, \delta, \lambda >$, представленный в виде таблицы состояний. В нашем случае X и $Y \in \{0,1\}$.

2. Строится тестирующая таблица (ТТаб), состоящая из 2 частей.

a. Верхняя часть ТТаб: Переписывается таблица состояний КА, строки в верхней части таблицы соответствуют состояниям машины, столбцы – входным символам, а записи в ячейках таблицы являются следующими состояниями.

b. Нижняя часть ТТаб: Заголовки строк – это всевозможные неупорядоченные пары состояний, а в ячейках – соответствующие им следующие состояния. Если запись на пересечении строк s_i и s_j и столбцов x_n в верхней половине ТТаб равна s_p и s_q соответственно, то запись в ячейке строки $s_i s_j$ и столбца x_n нижней половинки ТТаб равна $s_p s_q$.

3. Строится тестирующий ориентированный взвешенный граф G по ТТаб из шага 2.

a. Каждой совместимой паре состояний $s_i s_j$ соответствует вершина в G;

b. Дуга с надписью x_n выходит из вершины $s_i s_j$ в вершину $s_p s_q$, где p ≠ q, тогда и только тогда, когда совместимая пара $(s_p s_q)$ является следующим состоянием совместимой пары $(s_i s_j)$.

4. Определяют отношение исходного КА к группе КА с входной памятью и находят его задержку.

a. Граф G из шага 3 проверяется на наличие цикла. В случае если цикл имеется, то КА не является автоматом с входной памятью, в случае отсутствия цикла – КА является автоматом с входной памятью.

b. Находят задержку $\mu = l+1$, где l - самый длинный путь графа G из шага 3.

Алгоритм 3. *Тест КА на наличие конечной выходной памяти и задержки.*

1. Пусть дан КА $M =< X, Y, S, \delta, \lambda >$, представленный в виде таблицы состояний. В нашем случае X и $Y \in \{0,1\}$.

2. Строится тестирующая таблица (ТТаб), состоящая из 2 частей.

a. Верхняя часть ТТаб: Переписывается таблица состояний КА, строки в верхней части таблицы соответствуют состояниям машины, столбцы – выходным символам. Запись на пересечении строки s_i и столбца y_m есть состояния, которые могут быть достигнуты из состояния s_i с помощью одного перехода, связанного с выходным символом y_m. Вся верхняя половина таблицы фактически представляет собой список следующих состояний автомата и поэтому называется *выходной таблицей*.

b. Нижняя часть ТТаб: Заголовки строк – это всевозможные неупорядоченные пары состояний, а в ячейках – соответствующие им следующие состояния. Если запись на пересечении строк s_i и s_j и столбцов y_m в верхней половине ТТаб равна s_p и s_q соответственно, то запись в ячейке строки $s_i s_j$ и столбца y_m нижней половинки ТТаб равна $s_p s_q$. Если запись на пересечении строк s_i и s_j и столбцов y_m в верхней половине ТТаб есть парные состояния $s_p s_q$ и $s_r s_t$ соответственно, то запись в ячейке строки $s_i s_j$ и столбца y_m нижней половинки ТТаб равна $s_p s_r, s_p s_t, s_q s_r, s_q s_t$. Если запись на пересечении строк s_i и s_j и столбцов y_m в верхней половине ТТаб равна прочерку, то запись в ячейке строки $s_i s_j$ и столбца y_m нижней половинки ТТаб тоже равна тире прочерку.

3. Строится тестирующий ориентированный взвешенный граф G по ТТаб из шага 2.

a. Каждой совместимой паре состояний $s_i s_j$ соответствует вершина в G;

b. Дуга с надписью y_m выходит из вершины $s_i s_j$ в вершину $s_p s_q$, где p ≠ q, тогда и только тогда, когда совместимая пара $(s_p s_q)$ является следующим состоянием совместимой пары $(s_i s_j)$.

4. Определяют отношение исходного КА к группе КА с выходной памятью и находят его задержку.

a. Граф G из шага 3 проверяется на наличие цикла. В случае если цикл имеется, то КА не является автоматом с выходной памятью, в случае отсутствия цикла – КА является автоматом с выходной памятью.

b. Находят задержку $\mu = l+1$, где l - самый длинный путь графа G из шага 3.

Алгоритм 4. *Тест КА на обратимость.*

1. Пусть дан КА $M =< X, Y, S, \delta, \lambda >$, представленный в виде таблицы состояний. В нашем случае X и $Y \in \{0,1\}$.

2. Строится тестирующая таблица (ТТаб), состоящая из 2 частей.

a. Верхняя часть ТТаб: Переписывается таблица состояний КА в виде выходной таблицы (см. алг. 3 пункт 2a).

b. Нижняя часть ТТаб: Каждая совместимая пара состояний, появляющаяся в верхней части, становится заголовком строки нижней части тестирующей таблицы. Следующие состояния этих пар находятся так: они состоят из всех подразумеваемых совместных пар. Любая подразумеваемая пара состояний, которая еще не была использована в качестве заголовка строки, становится заголовком строки, ее следующее состояние находится таким же способом. Процесс завершается, когда все совместимые пары состояний были использованы в качестве заголовков строк.

3. Строится тестирующий ориентированный взвешенный граф G по ТТаб из шага 2.

a. Каждой совместимой паре состояний $s_i s_j$ соответствует вершина в G;

b. Дуга с надписью y_m выходит из вершины $s_i s_j$ в вершину $s_p s_q$, где p ≠ q, тогда и только тогда, когда совместимая пара $(s_p s_q)$ является следующим состоянием совместимой пары $(s_i s_j)$.

4. Определяют отношение исходного КА к группе обратимых КА с памятью и находят его задержку.

a. Граф G из шага 3 проверяется на наличие цикла. В случае если цикл имеется, то КА не является обратимым автоматом с конечной памятью, в случае отсутствия цикла – КА является обратимым автоматом с конечной памятью.

b. Находят задержку $\mu = l+2$, где l - самый длинный путь графа G из шага 3.

В статье были рассмотрены конечные автоматы с памятью общего вида, как известно не все автоматы этого класса могут быть обратимыми, поэтому для определения принадлежности КА к классу обратимых КА с памятью существуют описанные выше основные алгоритмы. Данные алгоритмы позволяют расширить область применения КА с памятью в криптографии. Данные автоматы могут использоваться в качестве ключей криптографических алгоритмов, а так же могут быть компонентами какой-либо комбинированной криптографической системы, состоящей из нескольких алгоритмов шифрования и дешифрования информации.

Список использованных источников

1. Шнайер Б. Прикладная криптография: протоколы, алгоритмы и исходные тексты на языке С. 2-е издание. – М.: Триумф, 2002, 816 с.

2. Шарипбай А.А., Сауханова Ж.С., Шахметова Г.Б., Сауханова М.С. Онтология конечно-автоматной криптографии // Онтология проектирования, Т.9, №1(31), 2019. С.36-49. – DOI: 10.18287/2223-9537-2019-9-1-36-49.

3. АгибаловГ.П. Конечные автоматы в криптографии // Прикладная дискретная математика. Приложение Математические методы криптографии, №2, 2009. С 45-73.
4. Богаченко Н.Ф. Применение теоретико-автоматных моделей в криптографии //Математические структуры и моделирование/Омск, 2007. С. 112-120.
5. Dömösi.P. A novel cryptosystem based on finite automata without outputs. Automata, Formal Languages and Algebraic Systems // Proceedings of AFLAS, 2008. P. 23-32 .
6. Kohavi Z., Niraj K. Switching and Finite Automata Theory. 3-е издание. – NY.: Cambridge University Press, 2010, p. 617
7. Катеринский Д.А. Об обратимости конечных автоматов с конечной задержкой // Прикладная дискретная математика, Приложение, №5, 2012. С. 43-44.

UDC 004

DEVELOPMENT OF A MOBILE APPLICATION TO IMPROVE THE SPEED OF FIRST AID

Daribay Omirzak
daribai.omirzak.od@gmail.com
Scientific adviser – Yermaganbetova M.A.

The mobile client being developed must provide the patient with deferred telemedicine consultation services. Because the main one the disadvantage of the previously considered solutions was the lack of possibility integration into a ready-made TMS, it was decided to develop its own mobile client for integration into the TMS of Akmola region.

For the patient, the application being developed must provide following function:

- possibility to register in TMS via the mobile app;
- organization of the patient's personal account;
- possibility of consulting a patient with a doctor by deferred medical consultations;
- the choice of the appropriate medical category, specialty or name;
- tracking the status of a completed consultation;
- view the response from a specialist;
- storing the consultation history with the ability to filter and search.

To integrate a mobile client into an existing TMS, it is necessary to highlight the main technology stack on which the system is built information system, and consider its architecture.

Main components of TMS:

- DB server;
- BY an Expert;
- AT the employee's FAP;
- mobile client.

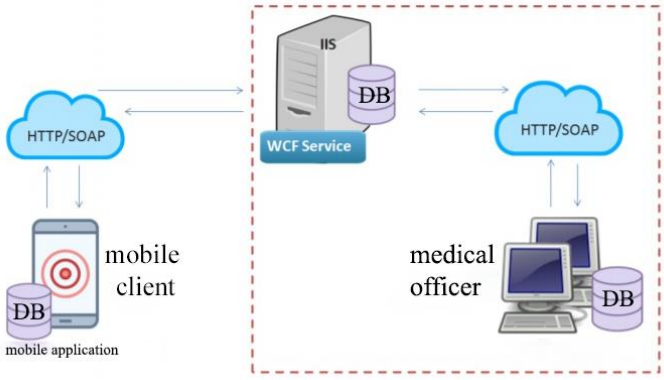

Figure 1. Scheme of interaction of a mobile client in TMS