DOI: 10.15514/ISPRAS-2019-31(6)-8

О возможности стойкой обфускации программ в одной модели облачных вычислений

[1,2] А.В. Шокуров, ORCID: 0000-0002-6801-7728 <shok@ispras.ru>
[3] И.В. Абрамова, ORCID: 0000-0002-8421-4617<abramovairinacmcmsu@gmail.com>
[1,2,3] Н.П. Варновский, ORCID: 0000-0002-2363-0254 <barnaba.np@gmail.com>
[1,2,3,4] В.А. Захаров, ORCID: 0000-0002-3794-9565 <zakh@cs.msu.ru>

[1] Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25
[2] Московский физико-технический институт,
141700, Россия, Московская область, г. Долгопрудный, Институтский пер., 9
[3] Московский государственный университет имени М.В. Ломоносова,
119991, Россия, Москва, Ленинские горы, д. 1
[4] НИУ Высшая школа экономики,
101978, Россия, г. Москва, ул. Мясницкая, д. 20

Аннотация. В данной статье проведено исследование возможности применения одной модели облачных вычислений, использующей криптосерверы, для обфускации программ. Ранее эта модель облачных вычислений была предложена нами в связи с изучением задачи обеспечения информационной безопасности мультиклиентских распределенных вычислений над зашифрованными данными. На основе этой модели нами предложен новый подход, предусматривающий использование пороговых гомоморфных криптосистем для обфускации программ. Основным результатом статьи являются новое определение стойкости обфускации программ в модели облачных вычислений и теорема, доказывающая криптографическую стойкость предложенного алгоритма обфускации программ в предположении существования криптографически стойких пороговых гомоморфных систем шифрования.

Ключевые слова: обфускация программ; гомоморфное шифрование; стойкость; облачные вычисления

Для цитирования: Шокуров А.В., Абрамова И.В., Варновский Н.П., Захаров В.А. О возможности стойкой обфускации программ в одной модели облачных вычислений. Труды ИСП РАН, том 31, вып. 6, 2019 г., стр. 145–162. DOI: 10.15514/ISPRAS–2019–31(6)–8

Благодарности: Исследования, результаты которых представлены в этой статье, выполнены при поддержке Российского Фонда Фундаментальных Исследований (проект 19-01-00702).

On the possibility of secure program obfuscation in some model of cloud computing

[1,2] A.V. Shokurov, ORCID: 0000-0002-6801-7728 <shok@ispras.ru>
[3] I.V. Abramova ORCID: 0000-0002-8421-4617 <abramovairinacmcmsu@gmail.com>
[1,2,3] N.P. Varnovsky, ORCID: 0000-0002-2363-0254 <barnaba.np@gmail.com>
[1,2,3,4] V.A. Zakharov, ORCID: 0000-0002-3794-9565 <zakh@cs.msu.ru>

[1] Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia
[2] Moscow Institute of Physics and Technology (State University),
9 Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russia
[3] Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia
[4] National Research University, Higher School of Economics
20, Myasnitskaya Ulitsa, Moscow, 101978, Russia

Abstract. In this paper, we study the possibility of using a certain cloud computing model supplied with cryptoservers to obfuscate software programs. Earlier, we proposed this cloud computing model in our study of some information security problems for multi-client distributed computing over encrypted data. Based on this model, we proposed a new approach involving the use of threshold homomorphic cryptosystems for program obfuscation. The main result of this paper is a new definition of the resistance of obfuscation of programs in the cloud computing model and a theorem proving the cryptographic strength of the proposed algorithm of obfuscation of programs under the assumption of the existence of cryptographically strong threshold homomorphic encryption systems. The paper is organized as follows. In the introducing section we discuss the main aspects of the information security problems for cloud computing systems. Section 2 provides a description of the obfuscation program objectives, as well as a brief overview of the main achievements in its study. The next section provides general information about homomorphic cryptosystems. Section 4 describes a more special class of homomorphic cryptosystems - threshold homomorphic encryption systems. Section 5 introduces the cloud computing model, which is used as framework for our program obfuscation techniques. For this computing environment, in Section 6, the definition of the cryptographic strength of program obfuscation is formulated, a new method of program obfuscation using threshold homomorphic cryptosystems is described, and the cryptographic strength of the proposed obfuscation algorithm is proved.

Keywords: program obfuscation; homomorphic encryption; security; cloud computing

For citation: Shokurov A.V., Abramova I.V., Varnovsky N.P., Zakharov V.A. On the possibility of secure program obfuscation in some model of cloud computing. Trudy ISP RAN/Proc. ISP RAS, vol. 31, issue 6, 2019. pp. 145-162 (in Russian). DOI: 10.15514/ISPRAS-2019-31(6)-8

Acknowledgements. The studies, the results of which are presented in this article, were supported by the Russian Foundation for Basic Research (project 19-01-00702).

1. Введение

Современные прикладные информационные системы должны удовлетворять большому числу разнообразных требований, предъявляемых к их корректности, отказоустойчивости, совместимости с системным обеспечением, адаптируемости к вычислительному оборудованию и др. С появлением и развитием концепции облачных вычислений к числу этих требований было добавлено также и требование информационной безопасности; до недавнего времени оно предъявлялось лишь к телекоммуникационным протоколам и сравнительно узкому классу специализированных программ, реализующих криптографические алгоритмы и протоколы. Средства криптографии позволяли хорошо защищать данные при их хранении на общедоступных накопителях и передаче по открытым каналам связи. Но при вычислении (преобразовании) данных прикладными программами защите подлежали не сами

обрабатываемые данные или программы, работающие с этими данными, а вычислительные устройства (компьютеры, серверы, процессоры и пр.), на которых выполняются эти программы. Как правило, такое вычислительное устройство находилось в полном распоряжении пользователя, и он имел возможность применять для его защиты административные меры и физические средства.

Но при использовании систем облачных вычислений эти способы информационной защиты уже непригодны: в облачных системах вычислений данные (а также и программы) находятся в общедоступной среде, подобно тому, как это происходит при их хранении и передаче. Но тогда можно предполагать, что для их информационной защиты также можно использовать криптографические средства, например, шифрование. Главная трудность здесь состоит в том, что при хранении и передаче данные не подвергаются существенному изменению; они могут изменять форму их представления, но их семантическое содержание остается неизменным. Поэтому при хранении и передаче данные можно представлять в зашифрованном виде. Напротив, при проведении вычислений семантическое содержание данных очень важно, и поэтому для их защиты могут быть пригодны лишь такие системы шифрования, в которых эти изменения можно адекватно представить при помощи подходящих операций над шифрами данных.

Системы шифрования, обладающие указанным свойством, называются гомоморфными криптосистемами (гомоморфизм в алгебре означает свойство согласованности операций в двух различных алгебраических системах – в нашем случае, в пространстве данных и в пространстве их шифртекстов). Построение гомоморфных систем шифрования позволило бы значительно расширить область применения криптографических средств защиты информации. Поэтому задача построения гомоморфных криптосистем остается одной из центральных проблем современной криптографии.

Однако построение гомоморфной системы шифрование является лишь необходимым условием обеспечения надежной информационной защиты данных в процессе вычисления. Как известно, стойкость криптографических протоколов зависит не только от стойкости используемых в них криптографических примитивов (т.е. базовых функций, наподобие процедуры генерации ключей, шифрования, цифровой подписи и пр.), но также и от особенностей устройства самих протоколов и задач, для решения которых они предназначены. В связи с этим не меньшее значение имеют также и вопросы о том, для каких вычислительных схем и при каких условиях стойкость гомоморфного шифрования является достаточным условием стойкости информационной защиты этих схем.

В данной статье проведено исследование возможности применения одной модели облачных вычислений, использующей криптосерверы, для обфускации программ. Ранее эта модель облачных вычислений была предложена нами в связи с изучением задачи обеспечения информационной безопасности мультиклиентских распределенных вычислений над зашифрованными данными. На основе этой модели нами предложен новый подход, предусматривающий использование пороговых гомоморфных криптосистем для обфускации программ. Основным результатом статьи являются новое определение стойкости обфускации программ в модели облачных вычислений и теорема, доказывающая криптографическую стойкость предложенного алгоритма обфускации программ в предположении существования криптографически стойких пороговых гомоморфных систем шифрования. Подобный подход к обфускации программ восходит к статьям [1-3], в которых было показано, что информационная безопасность вычислений может быть гарантирована в том случае, если один из компонентов вычислительной системы (например, память компьютера) оказывается недоступным для противника.

Статья устроена следующим образом. В разд. 2 приводится описание задачи обфускации программ, а также краткий обзор основных достижений в ее исследовании. В следующем разделе приведены общие сведения о гомоморфных криптосистемах. В разд. 4 рассказано о более специальном классе гомоморфных криптосистем – пороговых гомоморфных

системах шифрования. В разд. 5 приведено описание модели облачных вычислений, в которой решается задача обфускации программ. Для этой вычислительной среды в разд. 6 сформулировано определение криптографической стойкости обфускации программ, описан новый метод обфускации программ с использованием пороговых гомоморфных криптосистем, и доказана криптографическая стойкость предложенного алгоритма обфускации.

2. Задача обфускации программ

Обфускацией программы называется любое ее преобразование, которое сохраняет функциональность программы, но при этом приводит программу в такую форму, из которой очень трудно извлечь полезную информацию об алгоритмах и структурах данных, реализованных в программе. Согласно определению, предложенному в статье [4], обфускатор программ – это такой вероятностный алгоритм O, который, получив на входе программу π, преобразует ее в программу $O(\pi)$, удовлетворяющую следующим трем требованиям.

- функциональность: программы π и $O(\pi)$ вычисляют одну и ту же функцию;
- эффективность: размер и быстродействие программы $O(\pi)$ ухудшаются незначительно по сравнению с программой π;
- стойкость: программа $O(\pi)$ трудна для понимания.

Задача обфускации программ была впервые упомянута в основополагающей работе [5]. В явном виде понятие обфускации программ было введено в статье [6]. Более точное определение требований эффективности и стойкости обфускации зависит от тех приложений, в которых обфускация программ планируется использоваться. С описанием различных применений обфускации программ для решения задач системного программирования и компьютерной безопасности можно ознакомиться в статьях [6-10]. Обфускирующие преобразования могут быть использованы для защиты интеллектуальной собственности в качестве средства, препятствующего восстановлению исходных алгоритмов на основе открытого программного кода и удалению из программ водяных знаков (*watermarkings*) и «отпечатков пальцев», для защиты программного обеспечения от атак со стороны вредоносных программ (компьютерных вирусов) и обеспечения безопасности мобильных агентов в информационных сетях, для проведения безопасного поиска в потоках данных, защиты баз данных, защиты проектных решений при проектировании микроэлектронных схем. Оборотной стороной полезных достоинств обфускации является возможность ее использования для затруднения обнаружения вредоносных программ, а также создания уязвимостей в системах защиты компьютеров.

Помимо системного программирования задача обфускации программ также исследовалась в криптографии. Уже в ранних работах [4,11-14] было отмечено, что обфускация программ позволяет преобразовывать криптосистемы с секретным ключом в криптосистемы с открытым ключом: в качестве открытого ключа выступает обфускированная процедура шифрования с вставленным в нее секретным ключом. При помощи обфускации программ можно также конструировать гомоморфные системы шифрования, функциональные системы шифрования, доверенные схемы перешифрования и электронно-цифровой подписи, избавляться от модели случайного оракула при доказательстве стойкости криптографических протоколов, осуществлять случайную перестановку зашифрованных сообщений в схемах тайного голосования, создавать схемы дезавуируемого (двусмысленного) шифрования и односторонние функции с секретом. Однако для того, чтобы каждое из перечисленных приложений обладало определенной криптографической стойкостью, используемая для его построения обфускация программ также должна удовлетворять некоторым требованиям

стойкости. Поэтому при исследовании проблемы обфускации программ с позиции математической криптографии требование стойкости выдвигается на первый план.

В 2001 г. строгое математическое определение стойкости обфускации было предложено в статье [4]: обфускация считается стойкой, если всякий противник может эффективно извлечь из анализа текста обфускированной программы не больше информации, нежели при проведении тестовых испытаний, имея к ней доступ как к «черному ящику». В этой же статье было показано, что существуют такие программы, для которых такая стойкость обфускации в принципе недостижима. Впоследствии в ряде работ [15-20] были предложены и другие, менее требовательные определения стойкости обфускации; однако и для этих определений была показана невозможность построения трансляторов, гарантирующих стойкую обфускацию произвольных программ. Обзор различных определений стойкости обфускации программ представлен в статье [21].

Вместе с тем, в работах [11,12,22-24] было показано, что для отдельных классов функций стойкая обфускация программ, вычисляющих эти функции, возможна при тех или иных криптографических предположениях. Наиболее значительное продвижение было достигнуто в работе [13]; ее авторы доказали осуществимость стойкой обфускации процедуры перешифрования сообщений. В целом, однако результаты исследований в этом направлении не дают больших оснований для оптимизма: до сих пор не удалось обнаружить достаточно сложной криптографической функции, процедуры вычисления которой допускают доказуемо стойкую обфускацию.

Существенное продвижение в изучении возможности построения криптографически стойких обфускаторов программ произошло с открытием методов построения стойких систем гомоморфного шифрования. Если в начале века бытовало мнение о том, что стойкие обфускаторы программ позволят создать гомоморфные системы шифрования, то после основополагающих результатов К. Джентри (C. Gentry) [25,26] исследователи задались противоположным вопросом: насколько полезным могут быть гомоморфные криптосистемы для построения стойких обфускаторов программ?

3. Гомоморфные криптосистемы

Система шифрования с открытым ключом состоит из следующих шести компонентов:

- пространство *сообщений* (открытых текстов) M; элементы множества M играют роль данных подлежащих информационной защите;
- пространство *шифртекстов* C; элементы множества C – это зашифрованные данные;
- пространство ключей шифрования (открытых ключей) K_p и расшифрования (секретных ключей) K_s;
- процедура *генерации ключей* Gen; это эффективный (полиномиальный по времени) вероятностный алгоритм (машина Тьюринга), который по заданному параметру стойкости λ вычисляет пару ключей (sk, pk) из множества $K_s \times K_p$ (секретный ключ расшифрования и соответствующий ему открытый ключ шифрования);
- процедура *шифрования* Enc; это эффективный вероятностный алгоритм, который по заданному открытому ключу pk и сообщению m вычисляет шифртекст $c = Enc(pk, m)$ этого сообщения;
- процедура *расшифрования* Dec; это эффективный детерминированный алгоритм, который по заданному секретному ключу sk и шифртексту c вычисляет сообщение $m = Dec(sk, c)$.

Обычно пространства сообщений, шифртекстов и ключей – это множества двоичных наборов. Параметр стойкости λ определяет длину ключей n, и $K = \{0,1\}^n$. В зависимости от выбранной длины ключей определяются размерности пространства сообщений $M = \{0,1\}^{q(n)}$ и пространства шифртекстов $C = \{0,1\}^{r(n)}$, где $q(n), r(n)$ – некоторые полиномы, зависящие от n. Для удобства описания и анализа некоторых систем шифрования двоичные наборы могут рассматриваться как записи натуральных чисел в двоичной системе счисления или представления элементов специальных групп в некоторой кодировке.

Три указанные процедуры системы шифрования – генерации ключей, шифрования и расшифрования – должны удовлетворять следующему требованию *корректности*:

- для любой пары ключей (sk,pk), порожденного процедурой генерации ключей Gen, и для любого сообщения m верно равенство $Dec\big(sk, Enc(pk, m)\big) = m$.

Пусть имеются некоторые множества S_1 и S_2, на которых определены функции $f_i^{(n_i)}(x_1, \dots, x_{n_i})$ и $g_i^{(n_i)}(y_1, \dots, y_{n_i}), 1 \le i \le m,$ соответственно. Отображение $\varphi: S_1 \to S_2$ называется гомоморфизмом из алгебраической системы $\left(S_1, f_1^{(n_1)}, \dots, f_m^{(n_m)}\right)$ в алгебраическую систему $\left(S_2, g_1^{(n_1)}, \dots, g_m^{(n_m)}\right)$, если для любого $i, 1 \le i \le m,$ и для любого набора $(d_1, \dots, d_{n_i})$ элементов из множества S_1 справедливо равенство

$$\varphi\left(f_i^{(n_i)}(d_1, \dots, d_{n_i})\right) = g_i^{(n_i)}$$

Предположим, что в пространстве сообщений M системы шифрования (M, C, K, Gen, Enc, Dec) определены эффективно вычислимые функции $f_i^{(n_i)}(x_1, \dots, x_{n_i})$ $, 1 \le i \le m.$ Эта система шифрования считается *гомоморфной* относительно перечисленного множества функций, если в пространстве шифртекстов существуют эффективно вычислимые функции $g_i^{(n_i)}(y_1, \dots, y_{n_i}), 1 \le i \le m,$ для которых при любом ключе, порождаемом процедурой генерации ключей Gen, отображение шифрования Enc является гомоморфизмом из алгебраической системы $\left(M, f_1^{(n_1)}, \dots, f_m^{(n_m)}\right)$ в алгебраическую систему $\left(C, g_1^{(n_1)}, \dots, g_m^{(n_m)}\right)$. Система шифрования считается *вполне гомоморфной*, если множество функций $f_1^{(n_1)}, \dots, f_m^{(n_m)}$ является полным относительно операции суперпозиции множеством функций на множестве M, т.е. любую эффективно вычислимую функцию на множестве M можно получить при помощи операции суперпозиции из функций указанного множества. Если $M = \{0,1\}$ (т.е. рассматриваются только булевы функции), то система шифрования является вполне гомоморфной в том и только том случае, когда она гомоморфна относительно полной системы операций булевой алгебры (например, относительно сложения и умножения). В общем случае система вполне гомоморфного шифрования снабжена эффективно вычислимой процедурой $Eval$, которая преобразует произвольное описание функции F на множестве сообщений (например, представленное схемой из функциональных элементов) в описание функции G на множестве шифртекстов, удовлетворяющей равенству

$$Enc(F(d_1, \dots, d_n), sk) = G$$

для любых наборов сообщений $(d_1, \dots, d_n)$и любых ключей sk, порождаемых процедурой генерации ключей Gen.

Понятие гомоморфного шифрования возникло в криптографии в 1978 году [27] сразу же после создания первой криптосистемы с открытым ключом. Однако все криптосистемы с открытым ключом, разработанные на рубеже веков, оказались гомоморфными относительно лишь одной из двух операций сложения и умножения битов (см., например, [28]). Наибольшим достижением в ранних работах по гомоморфному шифрованию

является результат статьи [29], авторы которой предложили криптосистему, которая на основе билинейных спариваний на эллиптических кривых была способна выполнять неограниченное число сложений и одно умножение над зашифрованными данными.

Проблема построения доказуемо стойких вполне гомоморфных криптосистем оставалась открытой до 2009 года, Джентри (C. Gentry) теоретически обосновал возможность построения такой системы в статьях [25,26]. Описанная система шифрования Джентри является семантически стойкой в предположении о неразрешимости за полиномиальное время некоторых вычислительных задач, наподобие наилучшей аппроксимации заданного вектора в векторном метрическом пространстве элементом заданной решетки, вложенной в это пространство или вычисления приближенного наибольшего общего кратного целых чисел, и она способна проводить гомоморфные вычисления над зашифрованными данными. Поэтому теоретически она может быть использована для информационной защиты данных в облачных вычислениях.

Однако возможности практического использования этой системы шифрования до сих пор остаются неочевидными. В 2010 г. Смарт (N.P. Smart) и Веркотерен (F. Vercauteren) модифицировали схему Джентри, уменьшив размер ключа, но ценой усложнения процедуры генерации ключей [30]. В том же году Джентри и Халеви (S. Halevi) представили реализацию системы шифрования [31,32], а Штеле (Stehle) и Штейн (Stein) упростили эту схему и одновременно с этим повысили ее эффективность [33]. В серии работ, начиная с 2011 г., Бракерски (Z. Brakerski), Джентри, Халеви, Смарт и Вайкунтанатхан (V. Vaikuntanathan) продолжали совершенствовать схемы гомоморфного шифрования с целью повышения их эффективности [34-42]. Им удалось разработать систему гомоморфного шифрования без использования процедуры скрытого перешифрования. Эта система основана на задаче выведывания на основе примеров с ошибками (LWE, learning with errors) [38]. В ней существенно улучшена эффективность гомоморфных вычислений. В 2012 г. Бракерски доказал, что стойкая система шифрования не может иметь простую процедуру расшифрования и быть при этом вполне гомоморфной [39]. Этот результат устанавливает определенные пределы эффективного практического применения гомоморфных систем шифрования в некоторых приложениях. В 2012-13 г. Халеви, Джентри и др. исследовали вопросы организации гомоморфных вычислений над зашифрованными данными и использования систем гомоморфного шифрования в базах данных [43,32].

Возможность использования гомоморфных криптосистем для обфускации программ впервые была отмечена в статье [44]. Было показано, что при наличии вполне гомоморфной криптосистемы с открытым ключом для построения стойкой обфускации произвольной программы достаточно добиться стойкой обфускации всего лишь одной процедуры расшифрования. Дальнейшие усилия были сосредоточены на решении этой задачи [45,46,33]. Однако окончательного решения эта проблема все еще не получила.

Долгое время бытовала уверенность в том, что с изобретением систем вполне гомоморфного шифрования, будут решены многие задачи криптографии. Однако после того как криптосистема Джентри была предложена, неожиданно было обнаружено, что в некоторых случаях безопасность вычислений все равно не может быть обеспечена, причем этот результат имеет абсолютный характер, т.е. он не зависит от стойкости систем шифрования. Впервые этот эффект был обнаружен в работе Ван Дайка (Van Dijk) и Джулса (Jules) [47]. В этой работе они рассмотрели три класса вычислений с секретными данными и сформулировали строгое определение стойкости схем вычислений над зашифрованными данными. Опираясь на это определение стойкости, они показали, что все вычисления первого класса можно сделать безопасными при помощи гомоморфных криптосистем. Однако для двух других классов это уже не так. Как было показано, вычислительные схемы второго класса можно использовать для того, чтобы построить стойкий обфускатор программ в модели «черного ящика». Поскольку ранее было

доказано, что такой обфускатор не существует, отсюда следует, что стойкой информационной защиты схем вычислений второго класса построить также невозможно.

Таким образом, для защиты данных в облачных системах вычислений необходимы дополнительные средства. В статье [48] был предложен один подход к организации безопасных вычислений в облачной среде с использованием пороговых гомоморфных систем шифрования.

4. Пороговые системы гомоморфного шифрования

Пороговая система гомоморфного шифрования с открытым ключом состоит из множества криптосерверов $S_1, S_2, \dots, S_l$, которые можно рассматривать как интерактивные вероятностные машины Тьюринга. Все криптосерверы попарно соединены друг с другом каналами связи и предназначены для выполнения следующих алгоритмов.

1. Протокол генерации ключей. При выполнении этого протокола на вход каждого криптосервера поступает натуральное число n в унарной форме записи, которое служит параметром стойкости системы шифрования. Получив это число, каждый криптосервер S_i вычисляет свою долю секретного ключа s_i, а затем все криптосерверы совместно в интерактивном режиме вычисляют открытый ключ pk, соответствующий сгенерированным долям $s_1, \dots, s_l$ секретного ключа.
2. Алгоритм шифрования Enc. На вход алгоритма поступает открытый ключ pk и открытый текст m; на выходе алгоритма вычисляется криптограмма $c = Enc(pk, m)$. Алгоритм шифрования может быть выполнен на любой вероятностной машине Тьюринга за полиномиальное время.
3. Протокол расшифрования, который включает процедуры расшифрования, выполняемые криптосерверами, и процедуру интеграции, которая выполняется пользователем криптосистемы. Каждый криптосервер S_i, получив на входе криптограмму c, вычисляет, используя свою долю секретного ключа s_i, фрагмент расшифрованного сообщения $Dec(s_i, c)$. Пользователь, располагая криптограммой c и всеми фрагментами расшифрованного сообщения $Dec(s_1, c), \dots, Dec(s_l, c)$, восстанавливает зашифрованное сообщение m при помощи процедуры интеграции. Для выполнения процедуры интеграции секретные ключи не нужны.

Примером пороговой системы гомоморфного шифрования может служить криптосистема, описанная в статье [7]. Организация вычислений над зашифрованными данными с применением подобной криптосистемы описана в статье [49].

Обычно рассматриваются два сценария использования пороговых криптосистем, в зависимости от того, кто является получателем информации после расшифрования.

В первом из них получателем открытого текста является каждый из криптосерверов. Основное требование к криптосистеме состоит в существовании такого порога $t, t < l$, что любая криптограмма может быть расшифрована только при согласии не менее чем t криптосерверов. Именно этот вариант обычно рассматривается в литературе.

Во втором сценарии получателем открытого текста является внешний участник, а функции криптосерверов полностью соответствуют их названию. В этой статье мы будем рассматривать пороговые криптосистемы с неинтерактивным протоколом расшифрования: крпитосервер S_i, получив криптограмму c, вычисляет некоторый фрагмент, обозначаемый записью $Dec(s_i, c)$, и посылает его получателю открытого текста по защищенному каналу связи. Существует эффективный алгоритм интеграции, который, получив на вход $c, Dec(s_1, c), \dots, Dec(s_l, c)$, вычисляет открытый текст m. В этом сценарии основное требование к криптосистеме формулируется так: существует такой порог $t, t < l$, что знание любых t долей секретного ключа не позволяет расшифровать криптограмму.

Мы приведем определение стойкости пороговой системы гомоморфного шифрования для второго варианта применения криптосистемы. Пороговая система шифрования с открытым ключом является пороговой не вполне гомоморфной криптосистемой (Threshold Somewhat Homomorphic Encryption, TSHE), если существуют такой эффективный алгоритм $Eval$ и такой параметр d, что для любой булевой схемы F глубины не более d с k входами, для любых $m_1, \dots, m_k \in \{0,1\}$ результат $Eval(pk, c_1, \dots, c_k, f)$ - это криптограмма открытого текста $F(m_1, \dots, m_k)$. Здесь $c_j = Enc(pk, m_j)$, а f – это битовая строка, описывающая схему F. Формально алгоритм $Eval$ определяется как полиномиальная вероятностная машина Тьюринга. Параметр d, вообще говоря, является функцией параметра стойкости n и других параметров TSHE.

В отличие от вполне гомоморфных, не вполне гомоморфные криптосистемы не используют процедуру перешифрования (*bootstrapping*) и потому могут проводить лишь ограниченные вычисления.

Для определения стойкости TSHE воспользуемся моделью противника, который представляет собой полиномиальную вероятностную машину Тьюринга Adv. Противнику доступна атака с известным открытым ключом pk и известными t долями секретного ключа. Все доли секретного ключа равноправны, и поэтому без ограничения общности далее мы будем считать, что это доли $s_1, \dots, s_t$. Угроза различения открытых текстов по их криптограммам определяется следующим образом: противник выбирает пару открытых текстов m^0, m^1 одинаковой длины, получает криптограмму $c = Enc(pk, m^\sigma)$, где $\sigma \in_R \{0,1\}$, и угадывает σ с вероятностью, существенно отличающейся от 1/2. Поскольку гомоморфные криптосистемы работают с однобитовыми открытыми текстами, здесь предполагается, что строки m^0и m^1 шифруются побитово и c - конкатенация соответствующих криптограмм.

Для формализации угрозы машине Adv предоставляется доступ к оракулу O, который, получив от противника Adv пару (m^0, m^1), выбирает случайный бит σ, вычисляет криптограмму $c = Enc(pk, m^\sigma)$ и возвращает c в качестве ответа на запрос.

Определение 1. TSHE называется семантически стойкой, если для любой полиномиальной вероятностной машины Тьюринга Adv имеет место равенство

$$Pr\left[Adv^O(pk, s_1, \dots, s_t) = \sigma\right] - 1/2 = \nu(n),$$

где $\nu(n)$ пренебрежимо малая функция, т.е. функция, удовлетворяющая соотношению $\nu(n) = o\big(1/P(n)\big)$ для любого полинома P.

5. Модель облачных вычислений

В исследовании задачи построения стойкой системы обфускации программ в облачных вычислениях рассматривается следующая модель облачной среды с использованием криптосерверов и гомоморфного шифрования. Эта модель включает следующие компоненты.

1. **Участники (агенты)**: облачный сервер, пользователи системы системы облачных вычислений $U_1, \dots, U_k$, клиенты системы облачных вычислений $C_1, \dots, C_r$, криптосерверы $S_1, \dots, S_l$.
2. **Телекоммуникационная сеть**. Каждый из пользователей, а также каждый из клиентов имеет канал связи с облачным сервером. Каналы связи с криптосерверами образуют полный граф с l вершинами. Каждый криптосервер, помимо этого, имеет канал связи с облачным сервером.
3. **Данные**. У каждого пользователя $U_i, i = 1, \dots, k$ имеется конфиденциальная информация m_i, которые должны храниться и обрабатываться на облачном сервере/
4. **Программа**. На облачном сервере также размещается код программы F, который может быть зашифрован $Enc(pk, F)$ (например, при помощи гомоморфной системы

шифрования). Каждый клиент $C_i, i = 1, \dots, r$ имеет право передать облачному серверу запрос на применение программ к данным пользователей этой системы. Этот запрос включает в себя идентификатор клиента C_i, открытой ключ p_i некоторой системы шифрования с открытым ключом.

5. **Контроль доступа**. Облако передает запрос клиента C_i в центр аутентификации, который проверяет полномочия клиента на применение программы F, и при наличии таковых, санкционирует выполнение запроса.
6. **Параметр стойкости**: натуральное число n; каждый из основных криптографических параметров модели (длина ключей и пр.) ограничен некоторым полиномом, зависящим от n.
7. **Вычислительные ресурсы**. Все участники рассматриваемой модели (включая противника) представляют собой вероятностные машины Тьюринга, работающие за полиномиальное от n время.

Основные допущения, в рамках которых рассматривается данная модель вычислений, таковы.

- Пользователи не доверяют облачному серверу, который может рассматриваться в качестве пассивного противника. Это означает, что все функции по хранению и обработке конфиденциальных данных выполняются облачным сервером корректно, но все сведения, доступные облачному серверу также считаются доступными противнику и могут быть использованы для компрометации как хранимых пользовательских данных, так и обрабатывающих эти данные программ.
- Криптосерверы связаны между собой защищенными каналами связи, и память каждого криптосервера также защищена. Это может быть достигнуто с применением аппаратных или программных средств информационной защиты, включающей криптосистемы шифрования, электронно-цифровой подписи и пр.
- Существует такое натуральное число t, которое называется порогом и ограничивает сверху количество скомпрометированных криптосерверов. Криптосервер считается скомпрометированным, если он проводит возложенные на него вычисления некорректно или допускает утечку доступных ему данных.

6. Обфускация программ в модели облачных вычислений

Интерес представляет следующая задача построения эффективной и стойкой системы обфускации программ в описанной модели облачных вычислений. Владелец программы зашифровывает ее посредством гомоморфной системы шифрования и передает для хранения на облачный сервер. Любой клиент системы, желающий воспользоваться программой, обращается с запросом. Если запрос удовлетворен, то клиент отправляет данные на криптосерверы, которые, используя пороговую гомоморфную систему шифрования, шифруют данные и передают их на облачный сервер. Облачный сервер проводит вычисление зашифрованной программы над зашифрованными данными и отправляет зашифрованный результат вычисления. Получив этот зашифрованный результат, криптосерверы, используя доли секретного ключа гомоморфной системы шифрования, расшифровывают его и отправляют клиенту. Обфускацией программы в данном случае является ее шифрование посредством гомоморфной криптосистемы, реализованной на криптосерверах.

Основной задачей здесь является исследование стойкости предложенного метода обфускации программ в рассматриваемой модели облачных вычислений.

Без ограничения общности можно предполагать, что в системе облачных вычислений имеется единственный клиент. Пусть p – открытый ключ некоторой системы шифрования РКС с открытым ключом и функцией шифрования E. Можно считать, что эта система шифрования удовлетворяет стандартному определению семантической стойкости.

Предполагается, что о противнике (атаке) известно следующее:

- открытые ключи пользователя и системы гомоморфного шифрования, которая реализована на криптосерверах;
- обфускированная программа $Enc(pk, F)$;
- доли секретного ключа $s_1, \dots, s_t$ гомоморфной системы шифрования, которыми владеют скомпрометированные криптосерверы.

Противник также имеет доступ к специальному оракулу S. Запросом к оракулу служит произвольная криптограмма c. В ответ оракул возвращает набор фрагментов, $Dec(s_1, c), \dots, Dec(s_t, c)$, расшифрованных при помощи тех долей секретного ключа, которыми располагают скомпрометированные криптосерверы. Оракул S моделирует возможность противника контролировать не более t криптосерверов.

Угроза состоит в различении обфускации в данной модели и идеальной обфускации, которая определяется как модель, в которой противнику доступны лишь открытый ключ pk и значения $F(m_1, \dots, m_k)$ всех возможных результатов применения программы к клиентским данным. Эти значения становятся доступны при обращении к оракулу F.

Определение 2. Обфускация в модели облачных вычислений с вспомогательными криптосерверами называется стойкой, если для любой полиномиальной вероятностной машины Тьюринга A существует такая полиномиальная вероятностная машина Тьюринга B, что для любой программы F, размер которой не зависит от параметра стойкости, и для любого набора пользовательский данных $m_1, \dots, m_k$ справедливо неравенство

$$|Pr[A^S(pk, p, Enc(pk, F), s_1, \dots, s_t) = 1] - P\,r[B^F(pk) = 1]\,| \le \nu(n),$$

где $\nu(n)$ – пренебрежимо малая функция (величина), и вероятность вычисляется относительно случайных величин, используемых в алгоритмах шифрования, а также относительно открытых ключей шифрования p, pk.

Нами была доказана следующая

Теорема 1. Если существует стойкая пороговая криптосистема гомоморфного шифрования, то существует стойкая обфускация в модели облачных вычислений с использованием криптосерверов.

Доказательство. Прежде всего, заметим, что из предположения о существовании стойкой пороговой системы гомоморфного шифрования следует существование семантически стойкой системы шифрования с открытым ключом $PKE(G, E, D)$.

Рассмотрим систему облачных вычислений с криптосерверами, на которых реализована пороговая гомоморфная криптосистема с открытым ключом $THSE(Gen, Enc, Dec, Eval)$. Обфускация программ в такой системе проводится следующим образом.

1. Вначале криптосерверы выполняют протокол генерации ключей Gen, в результате работы которого каждый криптосервер $S_i, 1 \le i \le l$, формирует свою долю секретного ключа s_i, и, кроме того, вычисляется и публикуется открытый ключ pk, соответствующий этим долям секретного ключа.
2. Владелец программы выполняет протокол загрузки программы, в результате которого вычисляется шифртекст $Ob(F) = Enc(pk, F)$ программы Fи размещается в памяти облачной системы вычислений.
3. Клиент системы облачных вычислений выполняет протокол подготовки данных для вычисления; в результате выполнения этого протокола вычисляются шифртексты $c_i = Enc(pk, m_i)$ данных $m_i, 1 \le i \le k$,и размещаются в памяти облачной системы вычислений. Кроме того, клиент, используя протокол генерации ключей G криптосистемы с открытым ключом, формирует секретный ключ s и открытый ключ p, вычисляет шифр открытого ключа $p' = Enc(pk, p)$ и размещает его в памяти облачной системы.

4. Облачная система, используя алгоритм вычислений над зашифрованными данными $Calc$, применяет зашифрованную программу $Ob(F)$ к зашифрованным данным $c_1, \dots, c_k, p'$, вычисляет дважды зашифрованный результат $c = Enc\left(pk, E\left(p, F(m_1, \dots, m_k)\right)\right)$, и отправляет его криптосерверам.
5. Криптосерверы $S_i, 1 \le i \le l$ выполняют протокол расшифрования Dec, используя доли секретного ключа s_i, и вычисляют, используя открытый ключ клиента p, зашифрованные фрагменты $d_i = E\left(p, Dec(s_i, c)\right)$ зашифрованного результата c и отправляют их клиенту.
6. Клиент получает от облачной системы и криптосерверов строки $c, d_1, \dots, d_l$. Используя секретный ключ s, клиент вначале расшифровывает криптограммы $d_1, \dots, d_l$ и извлекает фрагменты $Dec(s_1, c), \dots, Dec(s_l, c)$. Затем, применяя алгоритм интеграции расшифрованных фрагментов, клиент на основе строк $c, Dec(s_1, c), \dots, Dec(s_l, c)$ вычисляет зашифрованный результатат вычисления $E\left(p, F(m_1, \dots, m_k)\right)$. И, наконец, используя еще раз секретный ключ s, клиент извлекает результат вычисления $F(m_1, \dots, m_k)$.

Покажем, что предложенная обфускация программ является стойкой.

Предположим противное. Тогда существует такие полиномиальная вероятностная машина Тьюринга A, набор входных данных $m_1, \dots, m_k$, и программа F, что для любой полиномиальной вероятностной машины Тьюринга B неравенство

$$|Pr[A^S(pk, p, Enc(pk, F), s_1, \dots, s_t) = 1] - Pr[B^F(pk) = 1]| > \varepsilon(n),$$

выполняется для бесконечно многих значений n. Здесь $\varepsilon(n)$ – функция, не являющаяся пренебрежимо малой, т.е. удовлетворяющая неравенству $\varepsilon(n) \ge 1/P\,(n)$ для некоторого полинома P.

Рассмотрим два сценария выполнения алгоритма A. В первом сценарии оракул S отвечает на запросы алгоритма в соответствии со своим предназначением, т.е. в ответ на запрос c возвращает набор криптограмм $Dec(s_1, c), \dots, Dec(s_t, c)$. Пусть

$$\delta_1^A(n) = Pr[A^S(pk, p, Enc(pk, F), s_1, \dots, s_t) = 1].$$

Во втором сценарии вместо оракула S противник использует оракул Q, который в ответ на каждый запрос c длины $|c| = r$ возвращает набор строк $Dec\left(s_1, Enc\left(pk, E(p, 0^r)\right)\right), \dots, Dec\left(s_t, Enc\left(pk, E(p, 0^r)\right)\right)$. Пусть

$$\delta_2^A(n) = Pr[A^Q(pk, p, Enc(pk, F), s_1, \dots, s_t) = 1].$$

Лемма 1. Для любой полиномиальной вероятностной машины Тьюринга верно соотношение

$$\delta_1^A(n) - \delta_2^A(n) = \nu(n)$$

Доказательство леммы.

Предположим противное: существует такая полиномиальная вероятностная машина Тьюринга A, для которой для бесконечно многих значений n неравенство $\delta_1^A(n) - \delta_2^A(n) \ge 1/Q\,(n)$ выполняется для некоторого полинома $Q(n)$. Покажем, что из этого предположения следует нестойкость системы шифрования РКС, которую использует клиент.

Построим полиномиальную вероятностную машину Тьюринга Adv_E, которая будет выступать в роли противника по отношению к РКС. Эта машина имеет возможность обращаться к случайному оракулу O. На входе p машина Adv_E создает описанную выше систему облачных вычислений и обращается к машине A, подавая ей на вход $pk, p, Enc(pk, F)$. Когда машина обращается к оракулу с запросом c, противник Adv_Eформирует пару $\left(\left(Dec(s_1, c), \dots, Dec(s_l, c)\right), 0^r\right)$, второй компонентой которой является нулевая строка той же длины, что и первый элемент пары, и передает эту пару

оракулу O. Оракул возвращает в ответ криптограмму c^*, и противник Adv_E передает эту криптограмму машине A. Выходом машины Adv_E служит выход машины A.

В том случае, когда c^* - это криптограмма первого элемента пары, все параметры и случайные величины, с которыми работает противник Adv_E, соответствуют работе машины A с оракулом S. Поэтому $Pr[Adv_E^Q(p) = 1] = \delta_1^A(n)$. А если c^* – это криптограмма второго элемента пары, то в этом случае вычисление противника Adv_E, соответствует работе машины A с оракулом Q. Поэтому $Pr[Adv_E^Q(p) = 0] = \delta_2^A(n)$. Коль скоро согласно предположению $\delta_1^A(n) - \delta_2^A(n) \geq 1/Q\,(n)$ для бесконечно многих значений n, получаем неравенство $|Pr[Adv_E^Q(p) = 1] - Pr[Adv_E^Q(p) = 0]| \geq 1/Q\,(n)$, противоречащее условию семантической стойкости системы шифрования РКС, которая используется в модели облачных вычислений.

Лемма доказана.

Располагая доказанной леммой, построим противника Adv для криптосистемы TSHE, используемой в модели облачных вычислений. Получив на входе открытый ключ pk, машина Adv формирует пару (m^0, m^1), где $m^1 = (m_1, \dots, m_k), m^0 = 0^r$, где $r = m^1 \vee$, и передает эту пару оракулу O, который возвращает в ответ криптограмму $c = Enc(pk, m^i)$, где $i \in \{0,1\}$. После этого противник Adv запускает машину A и подает на ее вход следующие данные: $pk, p, Enc(pk, F), c$.

Пусть $\sigma_1(n)$ – это вероятность совместного осуществления следующих двух событий: 1) оракул O вычислил криптограмму $c = Enc(pk, m^1)$ и 2) $Adv^O(pk) = 1$, а $\sigma_0(n)$ – это вероятность совместного осуществления следующих двух событий: 1) оракул O вычислил криптограмму $c = Enc(pk, m^0)$ и 2) $Adv^O(pk) = 0$.

Оценим величину $\sigma_1(n) + \sigma_0(n)$, которая равна вероятности того, что противник Adv принял правильное решение. Из описание противника Adv и определения вероятностей $\sigma_1(n)$ и $\sigma_0(n)$ следует, что их сумма равна величине

$$1/2\left(1 - Pr\left[A^Q\left(pk; p; Enc(pk, F); Enc(pk, m^0)\right) = 1\right]\right) +$$
$$1/2\left(Pr\left[A^Q\left(pk; p; Enc(pk, F); Enc(pk, m^1)\right) = 1\right]\right),$$

или, что то же самое

$$1/2 + 1/2(Pr\left[A^Q\left(pk; p; Enc(pk, F); Enc(pk, m^1)\right) = 1\right] -$$
$$Pr\left[A^Q\left(pk; p; Enc(pk, F); Enc(pk, m^0)\right) = 1\right].$$

Согласно лемме 1 верно, что

$$Pr\left[A^Q\left(pk; p; Enc(pk, F); Enc(pk, m^1)\right) = 1\right] -$$
$$Pr\left[A^S\left(pk; p; Enc(pk, F); Enc(pk, m^1)\right) = 1\right] = \nu(n).$$

Учитывая, что $m^0 = 0^r$, нетрудно заметить, что существует такая полиномиальная вероятностная машина Тьюринга B, для которой верно соотношение

$$Pr\left[A^Q\left(pk; p; Enc(pk, F); Enc(pk, m^0)\right) = 1\right] - Pr[B^F(pk)] = \nu(n).$$

И, наконец, как следует из предположения о нестойкости предложенной системы обфускации программ, соотношение

$$|Pr[A^S(pk, p, Enc(pk, F), s_1, \dots, s_t) = 1] - Pr[B^F(pk) = 1]| > \varepsilon(n)$$

верно для некоторой функции $\varepsilon(n)$, не являющейся пренебрежимо малой, для бесконечно многих значений n.

Поэтому на основании приведенных неравенств приходим к заключению о том, что

$$Pr\left[A^Q\left(pk; p; Enc(pk, F); Enc(pk, m^1)\right) = 1\right] -$$
$$Pr\left[A^Q\left(pk; p; Enc(pk, F); Enc(pk, m^0)\right) = 1\right] > \varepsilon(n) - \nu(n),$$

а это означает, что $\sigma_1(n) + \sigma_0(n) > 1/2 + \varepsilon\,(n)/2$. Последнее неравенство означает, в частности, что противник Adv компрометирует криптосистему TSHE вопреки условию о том, что TSHE является семантически стойкой.

Полученное противоречие означает, что гипотеза о том, что предложенная система обфускации программ не является стойкой, несостоятельна.

Доказательство теоремы закончено.

7. Заключение

Основные выводы, которые могут быть извлечены из проведенных в настоящее время исследований применимости методов криптографии для информационной защиты облачных вычислений таковы.

1. Существует большое число эффективных и стойких систем шифрования, являющихся гомоморфными относительно отдельных алгебраических операций (сложения, умножения).
2. Разработано несколько систем вполне гомоморфного шифрования, стойкость которых доказана. Однако все эти разработки пока носят исключительно теоретический характер, и вычисления, которые можно проводить с использованием этих криптосистем, все еще чрезвычайно неэффективны.
3. Для разработки криптостойких схем облачных вычислений необходимы формальные математические модели. Отдельные модели были разработаны для этой цели. Однако разнообразие этих моделей все еще невелико, и они не охватывают многие аспекты облачных вычислений.
4. При помощи предложенных формальных моделей облачных вычислений удалось показать, что существование систем вполне гомоморфного шифрования еще не является достаточным условием решения задачи информационной защиты облачных вычислений. Более того, было показано, что некоторые схемы облачных вычислений принципиально не могут иметь стойкой информационной защиты.

Перспективным для дальнейших исследований представляется подход к разработке пороговых систем гомоморфного шифрования и использования системы распределенных доверенных серверов для выполнения критических по требованиям безопасности криптографических процедур.

Список литературы / References

[1]. Goldwasser S., Micali S. Probabilistic encryption and how to play mental poker keeping secret all partial information. In Proc. of the 14th ACM Symposium on the Theory of Computing, 1982, pp. 365–377.

[2]. Ostrovsky R. Efficient computation on oblivious RAMs. Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, 1990, p. 514-523.

[3]. Ostrovsky R., Skeith III W.E. Private searching on streaming data. Lecture Notes in Computer Science, vol. 3621, 2005, p. 223-240.

[4]. Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vedhan S., Yang K. On the (Im)possibility of obfuscating programs. Journal of the ACM, vol, 59, no, 2, 2012, article no. 6

[5]. Collberg C., Thomborson C., Low D. A taxonomy of obfuscating transformations. Technical Report, no. 148, Department of Computer Science, University. of Auckland, 1997.

[6]. Collberg C, Thomborson C. Watermarking, tamper-proofing, and obfuscation - tools for software protection. IEEE Transactions on Software Engineering, vol. 28, no. 6, 2002, pp. 735 - 746.

[7]. D'Anna L., Matt B., Reisse A., Van Vleck T., Schwab S., LeBlanc P. Self-protecting mobile agents obfuscation report. Report 03-015, Network Associates Laboratories, 2003.

[8]. Diffie W., Hellman M. New directions in cryptography. IEEE Transactions on Information Theory, vol. 22, issue 6, 1976, pp. 644–654.

[9]. Sahai A., Waters B. How to Use Indistinguishability Obfuscation: Deniable Encryption, and More. In Proc. of the 46th Annual ACM Symposium on Theory of Computing, 2014, pp. 475–484.

[10]. Varnovsky N.P. A note on the concept of obfuscation. Trudy ISP RAN/Proc. ISP RAS, vol. 6, 2004, pp. 127-136.
[11]. Garg S., Gentry C., Halevi S., Raykova M., Sahai M., Waters B. Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits. SIAM Journal on Computing, vol. 45, no. 3, 2016, pp. 882-929.
[12]. Hofheinz D., Malone-Lee J., Stam M. Obfuscation for cryptographic purposes. Lecture Notes in Computer Science, vol. 4392, 2007, pp. 214-232.
[13]. Hohenberger S., Rothblum G. N., Shelat A., Vaikuntanathan V. Securely obfuscating reencryption. Lecture Notes in Computer Science, vol 4392, 2007, pp. 233-252.
[14]. Pass K., Seth K., Telang S. Indistinguishability Obfuscation from Semantically-Secure Multilinear Encodings. Lecture Notes in Computer Science, vol. 8616, 2014, p. 500-517.
[15]. Goldwasser S., Rothblum G.N. On best possible obfuscation. Lecture Notes in Computer Science, vol 4392, 2007, pp. 194-213.
[16]. Hada S. Secure obfuscation for encrypted signatures. Lecture Notes in Computer Science, vol. 6110, 2010, pp. 92-112..
[17]. Lynn B., Prabhakaran M., Sahai A. Positive results and techniques for obfuscation. Lecture Notes in Computer Science, vol. 3027, 2004, pp. 20-39.
[18]. Varnovsky N.P., Zakharov V.A. On the possibility of provably secure obfuscating programs. Lecture Notes in Computer Science, vol. 2890, 2003, pp. 91-102.
[19]. Goldwasser S., Micali S. Probabilistic encryption and how to play mental poker keeping secret all partial information. In Proc. of the 14th ACM Symposium on the Theory of Computing, 1982, pp. 365–377.
[20]. Goldwasser S., Tauman Kalai Y. On the impossibility of obfuscation with auxiliary input. In Proc. of the 46th IEEE Symposium on Foundations of Computer Science, 2005, pp. 553-562..
[21]. Варновский Н.П., Захаров В.А., Кузюрин Н.Н., Шокуров А.В. Современное состояние исследований в области обфускации программ: определения стойкости обфускации. Труды ИСП РАН, том 26, вып. 3, 2014 г., стр. 167-198 / Varnovsky N.P., Zakharov V.A.., Kuzurin N.N., Shokurov A.V. The current state of art in program obfuscations:definitions of obfuscation security. . Trudy ISP RAN/Proc. ISP RAS, vol. 26, issue 3, 2014, pp. 167-198 (in Russian). DOI: 10.15514/ISPRAS-2014-26(3)-9.
[22]. Kuzurin N.N., Shokurov A.V., Varnovsky N.P., Zakharov V.A. On the concept of software obfuscation in computer security. Lecture Notes in Computer Science, vol.4779, 2008, pp. 281-298.
[23]. Min. Zao E, Yang Ceng. Homomorphic Encryption Technology for Cloud Computing. Procedia Computer Science, vol. 154, 2019, pp. 73-83.
[24]. Wee H. On obfuscating point functions. In Proc. of 37th ACM Symposium on Theory of Computing, 2005, p. 523-532.
[25]. Gentry C. Computing Arbitrary Functions of Encrypted Data. Communication of the ACM, vol. 53, no. 3, 2010. pp. 97-105.
[26]. Gentry C., Halevi S. Implementing Gentry's Fully-Homomorphic Encryption Scheme. Lecture Notes in Computer Science, vol. 6632, 2011, pp. 129-148.
[27]. Rivest R.L., Adleman L., Dertouzos M.L. On data banks and privacy homomorphisms. In DeMillo R.A., ed., Foundations on Secure Computation, Academia Press, 1978, pp. 169-179.
[28]. Paillier P. Public-key cryptosystems based on composite degree residuosity classes. Lecture Notes in Computer Science, vol. 1592, 1999, pp. 223–238.
[29]. Boneh D., Goh Eu-Jin, Nissim K. Evaluating 2-DNF formulas on ciphertexts. Lecture Notes in Computer Science, vol. 3378, 2005, pp. 325-341.
[30]. Smart N., Verkauteren F. Fully homomorphic encryption with relatively small ciphertext and key size. Lecture Notes in Computer Science, vol. 6056, 2010, pp. 420-443.
[31]. Gentry C., Halevi S., Smart N. Homomorphic Evaluation of the AES Circuit. Lecture Notes in Computer Science, vol. 7417. 2012, pp. 850-867.
[32]. Gentry C., Halevi S., Smart N. Better Bootstrapping in Fully Homomorphic Encryption. Lecture Notes in Computer Science, vol. 7293, 2012, pp. 1-16.
[33]. Stehle D., Steinfeld R. Faster Fully Homomorphic Encryption. Lecture Notes in Computer Science, vol. 6477, 2010, pp. 377-394.
[34]. Brakerski Z., Vaikuntanathan V. Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. Lecture Notes in Computer Science, vol. 6841, 2011, pp. 505-524.

[35]. Brakerski Z. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. Lecture Notes in Computer Science, vol. 7417, 2012, pp. 868-886.
[36]. Brakerski Z., Gentry C., Vaikuntanathan V. (Leveled) Fully Homomorphic Encryption without Bootstrapping. ACM Transactions on Computation Theory, vol. 18, no. 3, 2014, article no. 13.
[37]. Brakerski Z., Gentry C., Halevi S. Packed Ciphertexts in LWE-based Homomorphic Encryption.. Lecture Notes in Computer Science, vol 7778, 2013, pp. 1-13.
[38]. Brakerski Z. When Homomorphism Becomes a Liability. Lecture Notes in Computer Science, vol. 7785, 2013, pp. 143-161.
[39]. Brakerski Z., Vaikuntanathan V. Lattice-Based FHE as Secure as PKE. In Proc. of the 5th Conference on Innovations in Theoretical Computer Science, 2014, pp.1-12.
[40]. Brakerski Z., Rothblum G.N. Virtual Black-Box Obfuscation for All Circuits via Generic Graded Encoding. Lecture Notes in Computer Science, vol. 8349, 2014, pp. 1-25.
[41]. Gentry C., Halevi S., Smart N. Fully Homomorphic Encryption with Polylog Overhead. Lecture Notes in Computer Science, vol. 7237, 2012, pp. 465-482.
[42]. Gentry C., Sahai A., Waters B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. Lecture Notes in Computer Science, vol. 8042, 2013, pp. 75-92. DOI: 10.1007/978-3-642-40041-4_5.
[43]. Boneh B., Gentry C., Gorbunov S., Halevi S., Nikolaenko V., Segev G., Vaikuntanathan V., Vinayagamurthy D. Lecture Notes in Computer Science, vol. 8441, 2014, pp. 533-556.
[44]. Gentry C. Fully homomorphic encryption using ideal lattices. In Proc. of the 41st ACM Symposium on Theory of Computing, 2009, pp. 169-178.
[45]. Barak B., Garg S., Tauman Kalai Y., Paneth O., Sahai A. Protecting Obfuscation against Algebraic Attacks. Lecture Notes in Computer Science, vol. 8441, 2014, pp. 221–238.
[46]. Canetti R., Dwork C., Naor M., Ostrovsky R. Deniable encryption. Lecture Notes in Computer Science, vol. 1294, 1997, p. 90-104.
[47]. Van Dijk M., Juels A. On the impossibility of cryptography alone for privacy preserving cloud computing. On Proc. of the 5th USENIX Conference on Hot Topics in Security, 2010, pp. 1–8.
[48]. Варновский Н.П., Мартишин С.А., Храпченко М.В., Шокуров А.В. Методы пороговой криптографии для защиты облачных вычислений. Труды ИСП РАН, том 26, вып. 2, 2014, стр. 269-274 / Varnovskij N.P., Martishin S.A., Khrapchenko M.V., Shokurov A.V. A Threshold Cryptosystem in Secure Cloud Computations. Trudy ISP RAN/Proc. ISP RAS, vol. 26, issue 2, 2014, pp. 269-274 (in Russian). DOI: 10.15514/ISPRAS-2014-26(2)-12.
[49]. Варновский Н.П., Захаров В.А., Шокуров А.В. К вопросу о существовании доказуемо стойких систем облачных вычислений. Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика, 2016, no. 2, стр. 32-37 / Varnovsky N.P., Zakharov V.A., Shokurov A.V. On the Existence of Provably Secure Cloud Computing Systems. Moscow University Computational Mathematics and Cybernetics, vol. 40, no. 2, 2016, pp. 83–88.

Информация об авторах / Information about authors

Александр Владимирович ШОКУРОВ – кандидат физико-математических наук, доцент, ведущий научный сотрудник отдела прикладной математики и информатики ИСП РАН, доцент кафедры системного программирования МФТИ. Сфера научных интересов: гомоморфное шифрование, облачные вычисления, криптография на решетках, алгебра.

Alexander Vladimirovich SHOKUROV – Candidate of Physics and Mathematics, Associate Professor, Leading Researcher of the Department of Applied Mathematics and Computer Science, ISP RAS, Associate Professor of the Department of System Programming at MIPT. Research interests: homomorphic encryption, cloud computing, lattice cryptography, algebra.

Ирина Валерьевна АБРАМОВА – студентка магистратуры факультета вычислительной математики и кибернетики МГУ. Сфера научных интересов: математические методы криптоанализа, криптографические методы защиты информации.

Irina Valerievna ABRAMOVA – Master of Science, faculty of computational mathematics and cybernetics, Lomonosov Moscow State University. Research interests: mathematical methods of cryptoanalysis, cryptographic methods for information protection.

Николай Павлович ВАРНОВСКИЙ – старший научный сотрудник отдела прикладной математики и информатики ИСП РАН, старший научный сотрудник Института проблем информационной безопасности МГУ, преподаватель кафедры системного программирования МФТИ. Сфера научных интересов: теория сложности вычислений, математические методы криптоанализа, криптографические методы защиты информации.

Nikolay Pavlovich VARNOVSKY – Senior Researcher, Department of Applied Mathematics and Informatics, ISP RAS; Senior Researcher, Institute of Information Security Problems, Moscow State University; Lecturer, Department of System Programming, MIPT. Research interests: theory of computational complexity, mathematical methods of cryptanalysis, cryptographic methods for information protection.

Владимир Анатольевич ЗАХАРОВ – доктор физико-математических наук, профессор, старший научный сотрудник отдела прикладной математики и информатики ИСП РАН; профессор кафедры математической кибернетики МГУ; ведущий научный сотрудник лаборатории процессно-ориентированных информационных ВШЭ; доцент кафедры системного программирования МФТИ. Сфера научных интересов: теория сложности вычислений, математические методы криптоанализа, математическая логика, теория автоматов, методы верификации программ.

Vladimir Anatolyevich ZAKHAROV – Doctor of Physics and Mathematics, Professor, Senior Researcher, Department of Applied Mathematics and Computer Science, ISP RAS; Professor, Department of Mathematical Cybernetics, Moscow State University; Leading Researcher, HSE Laboratory of Process-Oriented Information Systems; Associate Professor, Department of System Programming, MIPT. Research interests: theory of computational complexity, mathematical methods of cryptanalysis, mathematical logic, theory of automata, methods of program verification.