9. Данильченко А. Ф. Параметрически замкнутые классы функций трехзначной логики // Известия АН МССР. 1978. **2**. С. 13–20.
10. Danil'čenko A. F. On parametrical expressibility of the functions of k-valued logic // Colloq. Math. Soc. J. Bolyai. 1981. **28**. P. 147–159.
11. Barris S., Willard R. Finitely many primitive positive clones // Proc. Amer. Math. Soc. 1987. **101**. N 3. P. 427–430.
12. Марченков С. С. Задание позитивно замкнутых классов посредством полугрупп эндоморфизмов // Дискретная математика. 2012. **24**. № 4. С. 19–26.
13. Марченков С. С. Операторы замыкания логико-функционального типа. М.: МАКС Пресс, 2012.
14. Марченков С. С. Позитивно замкнутые классы трехзначной логики // Дискретный анализ и исследование операций. 2014. **21**. № 1. С. 67–83.

Поступила в редакцию
02.09.16

УДК 519.711.3

Н. П. Варновский[1], **В. А. Захаров**[2], **А. В. Шокуров**[3]

О ДЕДУКТИВНОЙ БЕЗОПАСНОСТИ ЗАПРОСОВ К БАЗАМ КОНФИДЕНЦИАЛЬНЫХ ДАННЫХ В СИСТЕМЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ*

Рассматриваются простейшие математические модели баз конфиденциальных данных в системах облачных вычислений. Для этих моделей предложено понятие дедуктивной безопасности запросов к базам данных. Установлены необходимые и достаточные условия дедуктивной безопасности и описаны некоторые классы запросов, удовлетворяющие этим условиям.

Ключевые слова: облачные вычисления, гомоморфная криптосистема, база данных, дедуктивная безопасность, симметрическая функция.

1. Введение. Открытие стойкого вполне гомоморфного шифрования [1] создало теоретические предпосылки решения задачи обеспечения информационной безопасности систем удаленных вычислений, включая системы облачных вычислений. За последние годы было предложено несколько подходов к построению гомоморфных криптосистем [2–4], проводятся исследования практических способов их реализации [5, 6], рассматриваются возможности применения гомоморфного шифрования для информационной защиты облачных вычислений [7] и баз данных [8, 9]. Но, как показано в работе [10], впечатление, что благодаря гомоморфным криптосистемам проблема защиты информации в облачных вычислениях может считаться решенной в принципе, неверно: даже в том случае, когда проводится лишь вычисление функций от хранящихся на облаке конфиденциальных значений аргументов, защита данных невозможна уже для системы с двумя пользователями. Чтобы избежать этого ограничения, в статье [11] была предложена специальная модель облачных вычислений, в состав которой помимо облачного сервера входят криптосерверы, на которых реализуется пороговая не вполне гомоморфная криптосистема с открытым ключом (Threshold Somewhat Homomorphic Encryption, TSHE). В статье [12] показано, что если TSHE является стойкой, и доля криптосерверов, контролируемых противником, не превосходит некоторого заданного порога, то предложенная система облачных вычислений является стойкой для вычислений ограниченной глубины.

Доказанная в [12] стойкость облачных вычислений не отменяет, тем не менее, возможности противника компрометировать конфиденциальные данные пользователей. Например, располагая

[1] Институт проблем информационной безопасности МГУ, ст. науч. сотр., e-mail: barnaba.np@gmail.com
[2] Факультет ВМК МГУ, проф., д.ф.-м.н., e-mail: zakh@cs.msu.su
[3] Институт системного программирования РАН, вед. науч. сотр., к.ф.-м.н., e-mail: shok@ispras.ru
* Работа выполнена при финансовой поддержке РФФИ, проект № 16–01–00714.

вычисленными значениями функций, зависящих от данных пользователя, противник может попытаться решить обратную задачу и определить значения аргументов этих функций. Для предотвращения этой возможности в системе облачных вычислений [11, 12] предусмотрен контроль доступа: облако передает запрос клиента в центр аутентификации, который проверяет полномочия клиента на вычисление запрашиваемой функции и, при наличии таковых, санкционирует выполнение запроса. Однако вопрос о том, какие запросы клиента считать допустимыми, оставался открытым. Данная статья инициирует исследование этого вопроса.

Вначале коротко описывается предложенная в статье [11] модель системы облачных вычислений с криптосерверами и объясняется, почему она нуждается в дополнительных мерах администрирования для защиты конфиденциальных данных пользователей. Далее рассматривается простейшая модель базы данных, формулируется требование дедуктивной безопасности запросов к этой базе данных, устанавливаются и обосновываются критерии дедуктивной безопасности запросов, приводится пример одного класса дедуктивно безопасных запросов к базе данных в рассматриваемой модели. В заключение показано, каким образом можно распространить полученные результаты на более сложные модели баз данных.

2. Модель системы облачных вычислений. В статьях [11, 12] была разработана и исследована модель облачных вычислений, состоящая из следующих компонентов.

Участники. К их числу относятся облачный сервер, пользователи $U_1, \ldots, U_k$, хранящие свои данные на облаке, клиенты $C_1, \ldots, C_r$, проводящие вычисления с этими данными, и криптосерверы $S_1, \ldots, S_\ell$.

Сеть связи. У каждого пользователя и клиента системы имеется канал связи с облачным сервером. Все криптосерверы связаны друг с другом. Кроме того, у каждого криптосервера есть канал связи с облачным сервером.

Данные. У каждого пользователя U_i, $i = 1, \ldots, k$, имеются конфиденциальные данные m_i, которые хранятся в зашифрованном виде на облаке. Для размещения данных на облаке пользователи применяют специальный протокол.

Вычисляемые функции. Имеется некоторое множество функций $F = \{f_1, \ldots, f_N\}$, которые разрешается вычислять над данными $m_1, \ldots, m_k$. Вычисление функций осуществляется алгоритмом Calc. Всякий клиент C_j, $j = 1, \ldots, r$, может передать облаку запрос на вычисление. В этом запросе указываются идентификатор клиента C_j, его открытый ключ p_j некоторой криптосистемы с открытым ключом и описание функции f из множества F, которую хочет вычислить клиент.

Контроль доступа. Запрос клиента C_j поступает в центр аутентификации, который проверяет полномочия этого клиента на вычисление запрашиваемой функции f и, при наличии таковых, санкционирует выполнение запроса.

Особая роль отводится криптосерверам $S_1, \ldots, S_l$, которые реализуют криптосистему TSHE, состоящую из следующих алгоритмов.

Распределенный протокол генерации ключей $KeyGen$. Получив на входе параметр стойкости, каждый сервер S_i генерирует свою долю s_i секретного ключа. Далее серверы совместно генерируют и публикуют открытый ключ pk.

Алгоритм шифрования Enc. Входами служат открытый ключ pk и открытый текст m, выходом — криптограмма $c = \mathrm{Enc}(pk, m)$. Процесс шифрования ничем не отличается от шифрования в обычной (непороговой) криптосистеме с открытым ключом.

Распределенный протокол расшифрования Dec. Входами для каждого криптосервера S_i служат соответствующая доля s_i секретного ключа и криптограмма c. Сервер S_i, получив криптограмму c, вычисляет некоторую информацию, обозначаемую через $\mathrm{Dec}(s_i, c)$, и посылает ее получателю открытого текста по защищенному каналу связи. Получатель располагает эффективным алгоритмом, который по полученным от криптосерверов данным c, $\mathrm{Dec}(s_1, c), \ldots, \mathrm{Dec}(s_l, c)$ вычисляет открытый текст m, соответствующий криптограмме c.

Алгоритм $Eval$ вычисления булевых схем глубины d. Для любой булевой схемы $\mathcal{F}$ с k входами, которая имеет глубину не более d и описывается битовой строкой f, и для любых криптограмм $c_1, \ldots, c_k$ входных данных $m_1, \ldots, m_k \in \{0, 1\}$, таких, что $c_j = \mathrm{Enc}(pk, m_j)$, $j = 1, \ldots, k$, результат $\mathrm{Eval}(pk, c_1, \ldots, c_k, f)$ — это криптограмма открытого текста $\mathcal{F}(m_1, \ldots, m_k)$ на открытом ключе pk.

Если служба контроля доступа санкционирует выполнение запроса (C_j, p, f), то система облачных вычислений далее функционирует следующим образом:

— на этапе инициализации криптосерверы, выполняя распределенный протокол генерации ключей KeyGen, создают доли секретного ключа $s_1, \ldots, s_l$, а также вычисляют и публикуют открытый ключ pk;

— каждый пользователь $U_i, i = 1, \ldots, k$ вычисляет криптограмму $c_i = \mathrm{Enc}(pk, m_i)$ своих данных и размещает ее на облаке;

— алгоритм Calc, получив на входе открытые ключи p и pk, набор криптограмм $c_1, \ldots, c_k$ и описание схемы f, использует алгоритм Eval системы TSHE для вычисления криптограммы $c = \mathrm{Enc}(pk, E(p, f(m_1, \ldots, m_k)))$;

— каждый криптосервер S_i, получив на входе открытый ключ p и криптограмму c, вычисляет, используя распределенный алгоритм расшифрования, криптограмму $E(p, \mathrm{Dec}(s_i, c))$ и посылает ее клиенту;

— алгоритм клиента получает на входе строки $c, E(p, \mathrm{Dec}(s_1, c)), \ldots, E(p, \mathrm{Dec}(s_l, c))$ и секретный ключ, соответствующий открытому ключу p. Вначале клиент извлекает из криптограмм значения $\mathrm{Dec}(s_1, c), \ldots, \mathrm{Dec}(s_l, c)$, затем снимает внешнее шифрование в криптограмме c и после этого, еще раз применив свой секретный ключ, извлекает значение $\mathcal{F}(m_1, \ldots, m_k)$.

В статье [12] было сформулировано требование стойкости системы облачных вычислений и доказано, что описанная выше система облачных вычислений является стойкой при следующих предположениях:

1) облачный сервер может контролироваться лишь пассивным противником, который выполняет корректно все функции по хранению и обработке конфиденциальных данных;
2) используемая TSHE имеет порог t и является стойкой;
3) в системе облачных вычислений имеется не более t "нечестных" криптосерверов.

Сформулированное в [12] требование криптографической стойкости подразумевает, что пассивный противник, контролирующий клиентов системы, облачный сервер и не более t криптосерверов, способен получить лишь такую информацию о конфиденциальных данных пользователей, которую можно извлечь из значений функций, вычисляемых по запросам клиентов. Но легко видеть, что некоторые запросы клиентов позволяют скомпрометировать конфиденциальные данные пользователей в обход описанных выше средств криптографической защиты. Например, вычислив значение тождественной функции, зависящей только от данных m_i, противник может узнать данные пользователя U_i. Поэтому аутентификация запросов на вычисления, выдаваемых клиентами, которая осуществляется посредством выбора множества F, представляет собой важный этап функционирования системы облачных вычислений. В данной статье мы исследуем вопрос о том, какие множества функций F можно считать допустимыми для вычислений с точки зрения сохранения конфиденциальности данных пользователей.

3. Дедуктивная безопасность запросов: однобитовые базы данных. Описанную выше систему облачных вычислений можно использовать для работы с базами конфиденциальных данных. В простейшем случае каждый пользователь U_i, $1 \leqslant i \leqslant k$, хранит в базе данных строку из одного бита — конфиденциальное истинностное значение m_i некоторого базового предиката P_i, ассоциированного с этим пользователем. Такие базы данных будем называть однобитовыми. Предикат P_i, например, может представлять собой некоторый анкетный вопрос, а его истинностное значение m_i — это ответ пользователя U_i на данный вопрос. Этот ответ является конфиденциальной информацией. Клиент C_j, $j = 1, \ldots, r$, может обращаться к базе данных с запросами. Запросом является произвольная булева формула $\varphi(P_1, \ldots, P_k)$, зависящая от предикатов пользователей. Ответом на запрос является истинностное значение этой формулы $\varphi(m_1, \ldots, m_k)$. Клиент базы данных имеет полномочие обращаться с запросами из некоторого множества $\mathcal{Q} = \{\varphi_1, \ldots, \varphi_N\}$. Сформулируем требование безопасности для множества запросов.

Для всякой формулы φ и бита d обозначим через φ^d формулу $\neg\varphi$, если $d = 0$, и φ, если $d = 1$. Для набора формул $\mathcal{Q} = (\varphi_1, \ldots, \varphi_N)$ и двоичного набора $\Delta = (d_1, \ldots, d_N)$ записью $\mathcal{Q}^\Delta$ обозначим набор формул $(\varphi_1^{d_1}, \ldots, \varphi_N^{d_N})$. Для набора формул $\mathcal{Q}$, зависящих от предикатов $P_1, \ldots, P_k$, и набора $\Sigma = (m_1, \ldots, m_k)$ истинностных значений этих предикатов обозначим через $\mathcal{Q}(\Sigma)$ набор

$(\varphi_1(\Sigma),\ldots,\varphi_N(\Sigma))$ истинностных значений указанных формул. Символами **0** и **1** обозначим двоичные наборы, состоящие из одних нулей и одних единиц соответственно. Символом $\models$ обозначается отношение логического следования в классической логике высказываний.

Определение 1. Множество запросов $\mathcal{Q}=\{\varphi_1,\ldots,\varphi_N\}$ к однобитовой базе данных, состоящей из значений базовых предикатов $P_1,\ldots,P_k$, называется *дедуктивно безопасным*, если для любого набора $\Sigma=(m_1,\ldots,m_k)$ истинностных значений этих предикатов соотношение $\mathcal{Q}^{\mathcal{Q}(\Sigma)}\models P_i^{m_i}$ не выполняется ни для одного предиката $P_i,\ 1\leqslant i\leqslant k$.

Приведенное определение дедуктивной безопасности множества запросов $\mathcal{Q}$ подразумевает, что клиент базы данных, получив ответы на все доступные ему запросы, не может дедуктивно выведать (получить в виде логического следствия) истинностного значения предиката ни одного пользователя базы данных. Таким образом, контроль полномочий клиента состоит в проверке дедуктивной безопасности множества запросов, с которыми клиент вправе обращаться к базе данных.

Теорема 1. *Набор запросов $\mathcal{Q}=\{\varphi_1,\ldots,\varphi_N\}$ является дедуктивно безопасным тогда и только тогда, когда для любого базового предиката P_i, $1\leqslant i\leqslant k$, и для любого набора Σ' значений базовых предикатов существует такой набор Σ'' значений этих предикатов, для которого выполняются соотношения $P_i(\Sigma')\neq P_i(\Sigma'')$ и $\mathcal{Q}(\Sigma')=\mathcal{Q}(\Sigma'')$.*

Доказательство. Рассмотрим произвольный набор $\Sigma'=(m'_1,\ldots,m'_k)$ истинностных значений базовых предикатов и произвольный предикат P_i. Пусть $\mathcal{Q}(\Sigma')=(d_1,\ldots,d_N)$. Применительно к набору Σ' и предикату P_i свойство дедуктивной безопасности множества запросов $\mathcal{Q}$ равносильно требованию того, чтобы формула

$$\Phi(P_1,\ldots,P_k)=\varphi_1^{d_1}\wedge\varphi_2^{d_2}\wedge\ldots\wedge\varphi_N^{d_N}\to P_i^{m'_i}$$

не была тождественно истинной. Поскольку $\Phi(m'_1,\ldots,m'_k)=1$, последнее возможно тогда и только тогда, когда существует такой набор $\Sigma''=(m''_1,\ldots,m''_k)$ значений базовых предикатов, для которого $m'_i\neq m''_i$ и при этом $\varphi_j(\Sigma'')=d_j$ для каждого запроса φ_j из множества $\mathcal{Q}$. Теорема доказана.

Из теоремы 1 следует, что вычислительная сложность задачи проверки дедуктивной безопасности произвольного множества запросов $\mathcal{Q}$ к базам данных сравнительно велика.

Теорема 2. *Задача проверки дедуктивной безопасности наборов запросов к базам данных является Π_2^p-полной.*

Доказательство. Непосредственно из формулировки критерия дедуктивной безопасности, представленного в теореме 1, следует, что указанная задача принадлежит сложностному классу Π_2^p полиномиальной иерархии.

Чтобы доказать Π_2^p-трудность задачи проверки дедуктивной безопасности, заметим, что согласно теореме 1 эта задача полиномиально взаимносводима к следующей задаче булевой алгебры: для двух произвольных заданных наборов булевых формул $\psi'_1(x'_1,\ldots,x'_k),\ldots,\psi'_N(x'_1,\ldots,x'_k)$ и $\psi''_1(x''_1,\ldots,x''_k),\ldots,\psi''_N(x''_1,\ldots,x''_k)$, зависящих от двух непересекающихся множеств переменных $X'=\{x'_1,\ldots,x'_k\}$ и $X''=\{x''_1,\ldots,x''_k\}$, выяснить, верно ли, что для любого набора Σ' значений переменных множества X' существует такой набор Σ'' значений переменных множества X'', для которого равенство $\psi'_i(\Sigma')=\psi''_i(\Sigma'')$ выполняется для всех пар формул ψ'_i, ψ''_i, $1\leqslant i\leqslant N$.

Покажем, что к последней задаче полиномиально сводима Π_2^p-полная задача проверки выполнимости квантифицированных булевых формул вида

$$\forall x'_1\ldots\forall x'_m\exists x''_1\ldots\exists x''_n\Psi(x'_1,\ldots,x'_m,x''_1,\ldots,x''_n). \tag{1}$$

Без ограничения общности можно считать, что бесквантороная часть этой формулы представлена в конъюнктивной нормальной форме $\Psi=\bigvee\limits_{\ell=1}^{M}D_\ell$. Каждому дизъюнкту

$$D_\ell=x'_{i_1}\vee\cdots\vee x'_{i_r}\vee x''_{j_1}\vee\cdots\vee x''_{j_s}$$

сопоставим две пары формул $f_{1\ell}=x'_{i_1}\vee\cdots\vee x'_{i_r}$, $g_{1\ell}=u_\ell$ и $f_{2\ell}=1$, $g_{2\ell}=u_\ell\vee x''_{j_1}\vee\cdots\vee x''_{j_s}$. Нетрудно видеть, что формула (1) выполнима тогда и только тогда, когда для любого набора Σ' значений переменных $x'_1,\ldots,x'_m$ существует такой набор Σ'' значений переменных $x''_1,\ldots,x''_n,u_1,\ldots,u_M$, что равенства $f_{1\ell}(\Sigma')=g_{1\ell}(\Sigma'')$ и $f_{2\ell}(\Sigma')=g_{2\ell}(\Sigma'')$ верны для всех $\ell=1,\ldots,M$. Теорема доказана.

4. Примеры дедуктивно безопасных запросов к базам данных. Поскольку задача проверки дедуктивной безопасности запросов к базам данных является вычислительно трудной,

целесообразно выделить некоторые практически значимые классы запросов, для которых эта задача может быть решена эффективно. Булева функция $f(y_1, \ldots, y_k)$ называется симметрической, если для любой перестановки $\theta : [1, k] \to [1, k]$ верно равенство $f(y_1, \ldots, y_k) = f(y_{\theta(1)}, \ldots, y_{\theta(k)})$; иными словами, значение симметрической функции определяется только количеством единиц в наборе значений аргументов. В сущности, симметрические запросы призваны собирать лишь статистические сведения о данных пользователей. Дедуктивная безопасность таких запросов означает, что статистические сведения, собранные клиентом базы данных, не позволяют ему получить сведения о данных какого-либо пользователя этой базы.

Теорема 3. *Пусть $\mathcal{Q}$ — некоторое множество симметрических запросов к базе данных. Тогда $\mathcal{Q}$ является дедуктивно безопасным в том и только том случае, когда существует такая пара наборов Σ_0 и Σ_1 истинностных значений базовых предикатов, для которых выполняются соотношения $\Sigma_0 \neq \mathbf{0}$, $\Sigma_1 \neq \mathbf{1}$, $\mathcal{Q}(\Sigma_0) = \mathcal{Q}(\mathbf{0})$ и $\mathcal{Q}(\Sigma_1) = \mathcal{Q}(\mathbf{1})$.*

Доказательство. Поскольку множество запросов $\mathcal{Q}$ симметрично, для любого набора Σ' истинностных значений базовых предикатов, отличного от наборов $\mathbf{0}$ и $\mathbf{1}$, и для любого базового предиката P_i, набор Σ'', обеспечивающий выполнимость критерия дедуктивной безопасности, сформулированного в теореме 1, получается за счет подходящей перестановки элементов набора Σ'. Если Σ' — это один из наборов $\mathbf{0}$ и $\mathbf{1}$, то для построения набора Σ'' нужно провести перестановку элементов соответственно в наборах Σ_0 и Σ_1, указанных в условии теоремы. Теорема доказана.

Нетрудно видеть, что для любого множества симметрических запросов к базе данных выполнимость предложенного в теореме 3 критерия дедуктивной безопасности можно проверить за полиномиальное время. В частности, дедуктивно безопасно всякое множество симметрических запросов $\mathcal{Q}$, удовлетворяющее равенству $\mathcal{Q}(\mathbf{0}) = \mathcal{Q}(\mathbf{1})$.

Помимо простоты проверки требования дедуктивной безопасности симметрические запросы обладают еще одним полезным свойством, которое облегчает их обработку в системах облачных вычислений, построенных на основе известных в настоящее время вполне гомоморфных криптосистем. Такие системы способны вычислять лишь булевы схемы ограниченной глубины. При вычислении более сложных схем происходит деградация зашифрованных данных, и для ее предотвращения нужно регулярно выполнять весьма дорогостоящий алгоритм перешифрования. Известно, что наименьшая глубина схем для почти всех булевых функций пропорциональна числу переменных, от которых зависят функции. Однако симметрические булевы функции принадлежат классу NC^1, т. е. их можно реализовать схемами логарифмической глубины.

5. Дедуктивная безопасность запросов: многобитовые базы данных. В общем случае данные пользователя не ограничиваются одним битом, а представляются более сложными структурами данных (списками, таблицами и др.). Однако при построении математической модели можно полагать, что данные каждого пользователя U_i задаются набором (строкой) $\mathbf{m}_i = (m_{i1}, \ldots, m_{in})$ истинностных значений базовых предикатов $P_i[1], \ldots, P_i[n]$. Порядковый номер j предиката $P_i[j]$ в этом списке будем называть уровнем предиката. Запрос клиента многобитовой базы данных — это произвольная булева формула $\varphi(P_1[1], \ldots, P_k[n])$, зависящая от базовых предикатов пользователей. Ответом на запрос является истинностное значение этой формулы $\varphi(\mathbf{m}_{11}, \ldots, \mathbf{m}_{kn})$.

Определение 2. Множество запросов $\mathcal{Q} = \{\varphi_1, \ldots, \varphi_N\}$ к многобитовой базе данных, состоящей из наборов значений базовых предикатов $\{P_i[j] : 1 \leqslant i \leqslant k, 1 \leqslant j \leqslant n\}$, называется дедуктивно безопасным, если для любого набора $\Sigma = (\mathbf{m}_1, \ldots, \mathbf{m}_k)$ истинностных значений этих предикатов, для любой булевой формулы $\psi(X_1, \ldots, X_n)$ и для любого пользователя U_i справедливо соотношение

$$\mathcal{Q}^{\mathcal{Q}(\Sigma)} \models \psi(P_i[1], \ldots, P_i[n]) \iff \models \psi(P_i[1], \ldots, P_i[n]).$$

В силу определения клиент многобитовой базы данных, воспользовавшись безопасным множеством запросов, не может дедуктивно выведать, исходя из ответов на эти запросы, никаких высказываний о данных каждого отдельного пользователя, кроме тавтологий. Для некоторых классов запросов к многобитовым базам данных необходимое и достаточное условие дедуктивной безопасности может быть получено на основании теоремы 1. Назовем ℓ-однородным всякий запрос к многобитовым базам данных, который зависит от базовых предикатов одного и того же уровня ℓ, т. е. задается булевой формулой вида $\varphi(P_1[\ell], \ldots, P_k[\ell])$.

Теорема 4. *Пусть* $\mathcal{Q} = \bigcup\limits_{\ell=1}^{n} \mathcal{Q}_\ell$, *причем каждое подмножество* $\mathcal{Q}_\ell$, $1 \leqslant \ell \leqslant n$, *состоит только из* ℓ*-однородных запросов. Тогда множество запросов* $\mathcal{Q}$ *является дедуктивно безопасным в том и только том случае, когда каждое подмножество запросов* $\mathcal{Q}_\ell$ *удовлетворяет условиям, сформулированным в теореме* 1, *т. е. для любого предиката* $P_i[\ell]$, $1 \leqslant i \leqslant k$, *и для любого набора* Σ' *истинностных значений базовых предикатов уровня* ℓ *существует такой набор* Σ'' *значений этих предикатов, для которого выполняются соотношения* $P_i[\ell](\Sigma') \neq P_i[\ell](\Sigma'')$ *и* $\mathcal{Q}_\ell(\Sigma') = \mathcal{Q}_\ell(\Sigma'')$.

Доказательство теоремы аналогично доказательству теоремы 1. Из теоремы 4 следует, что в качестве дедуктивно безопасных множеств запросов к многобитовым базам данных можно использовать ℓ-однородные симметрические запросы, удовлетворяющие условиям теоремы 3. Такие запросы позволяют вычислять некоторую статистическую информацию (например, сумму, медиану, среднеквадратическое отклонение и др.) о количественных данных, содержащихся в базе, но не раскрывают никакой специальной информации (например, имен), относящейся к отдельным пользователям базы данных.

В данной статье мы оставляем открытым вопрос о более общих критериях дедуктивной безопасности запросов к многобитовым базам данных.

6. Заключение. Один из важных аспектов безопасности облачных вычислений связан с проверкой полномочий клиента на вычисление запрашиваемых функций. Вопросы такого рода изучались ранее (см., например, [13]), и для их решения был предложен метод управления доступом на основе ролей (role-based access control) [14]. Суть его состоит в том, что контролирующая программа (монитор) проверяет в процессе вычисления соответствие действий клиента той роли (полномочиям), которая ему назначена. Однако для системы облачных вычислений, работающей с зашифрованными данными, такой мониторинг осуществить затруднительно или иногда просто невозможно. Поэтому при использовании гомоморфных криптосистем для информационной защиты облачных баз данных целесообразно уметь заранее оценивать последствия выполнения того или иного запроса с точки зрения сохранения безопасности данных пользователей.

В данной статье рассмотрена простейшая модель облачной базы данных и лишь некоторые требования дедуктивной безопасности запросов к базам данных. Эти требования, с одной стороны, являются слишком жесткими, поскольку не принимают во внимание роли клиентов, которые могут иметь разные права доступа к данным. С другой стороны, в рассмотренной модели не учтены возможности противника контролировать некоторых пользователей базы данных и вносить в нее изменения. Поэтому представляет интерес задача уточнения понятия дедуктивной безопасности запросов с учетом ролей, которые могут быть отведены клиентам, и изучение условий, которым должны удовлетворять дедуктивно безопасные запросы, соответствующие заданным ролям.

Авторы статьи выражают признательность А. А. Вороненко и А. А. Татузову за ценные советы, данные в процессе работы над задачами, исследованными в статье.

СПИСОК ЛИТЕРАТУРЫ

1. Gentry C. Fully homomorphic encryption using ideal lattices // Proc. of the 41-st Annual ACM Symp. on Theory of Computing. New York: ACM, 2009. P. 169–178.
2. Van Dijk M., Gentry C., Halevi S., Vaikuntanathan V. Fully homomorphic encryption over the integers // Proc. of the 29-th Intern. Conf. “Advances in Cryptology — EUROCRYPT 2010”. Lecture Notes in Computer Science. Vol. 6110. Berlin–Heidelberg: Springer, 2010. P. 24–43.
3. Brakerski Z., Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE // Proc. of the 52-nd Symp. on Foundations of Computer Science. Washington: IEEE Computing Society, 2011. P. 97–106.
4. Gentry C., Sahai A., Waters B. Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based // Proc. of the 32-nd Intern. Conf. “Advances in Cryptology — EUROCRYPT 2013”. Lecture Notes in Computer Science. Vol. 8042. Berlin–Heidelberg: Springer, 2013. P. 75–92.
5. Gentry C., Halevi S. Implementing Gentry’s fully-homomorphic encryption scheme // Proc. of the 30-st Intern. Conf. “Advances in Cryptology — EUROCRYPT 2011”. Lecture Notes in Computer Science. Vol. 6632. Berlin–Heidelberg: Springer, 2011. P. 129–148.

6. Gentry C., Halevi S., Smart N. Fully homomorphic encryption with polylog overhead // Proc. of the 31-st Intern. Conf. "Advances in Cryptology — EUROCRYPT 2012". Lecture Notes in Computer Science. Vol. 7237. Berlin–Heidelberg: Springer, 2012. P. 465–482.
7. Lopez-Alt A., Tromer E., Vaikuntanathan V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption // Proc. of the 44-th Annual ACM Symp. on Theory of Computing. New York: ACM, 2012. P. 1219–1234.
8. Gahi Y., Guennoun M., El-Khatib K. A secure database system using homomorphic encryption schemes // Proc. of the 3-d Intern. Conf. on Advances in Databases, Knowledge, and Data Applications. Red Hook: IARIA, 2011. P. 54–58.
9. Boneh D., Gentry C., Halevi S., Wang F., Wu D.J. Private database queries using somewhat homomorphic encryption // Proc. of ACNS. Lecture Notes in Computer Science. Vol. 7954. Berlin–Heidelberg: Springer, 2013. P. 129–148.
10. Van Dijk M., Juels A. On the impossibility of cryptography alone for privacy-preserving cloud computing // Proc. of the 5-th USENIX Conf. on Hot Topics in Security. Berkeley: USENIX Association, 2010. P. 1–8.
11. Варновский Н.П., Мартишин С.А., Храпченко М.В., Шокуров А.В. Пороговые системы гомоморфного шифрования и защита информации в облачных вычислениях // Программирование. 2015. № 4. С. 47–51. (Varnovskiy N.P., Martishin S.A., Khrapchenko M.V., Shokurov A.V. Secure cloud computing based threshold homomorphic encryption // Programming and Computer Software. 2015. **41**. N 4. P. 215–218.)
12. Варновский Н.П., Захаров В.А., Шокуров А.В. К вопросу о существовании доказуемо стойких систем облачных вычислений // Вестн. Моск. ун-та. Сер. 15. Вычисл. матем. и киберн. 2016. № 2. С. 32–38. (Varnovskiy N.P., Zakharov V.A., Shokurov A.V. On the existence of provably secure cloud computing systems // Moscow Univ. Comput. Math. and Cybern. 2016. **36**. N 2. P. 83–88.)
13. Barker S. Deductive database security // Research Directions in Data and Applications Security. IFIP Advances in Information and Communication Technology. Vol. 128. New York: Springer, 2003. P. 103–114.
14. Sandhu R., Coyne E.J., Feinstein H.L., Youman C.E. Role-based access control models // IEEE Computer. 1996. **29**. N 2. P. 38–47.

Поступила в редакцию
18.05.16

УДК 004.031.43

В. А. Костенко[1], **А. В. Плакунов**[2]

МУРАВЬИНЫЕ АЛГОРИТМЫ ДЛЯ ПЛАНИРОВАНИЯ ВЫЧИСЛЕНИЙ В ЦЕНТРАХ ОБРАБОТКИ ДАННЫХ*

Рассмотрен основанный на схеме муравьиных колоний алгоритм для решения задачи отображения запросов на физические ресурсы центров обработки данных. Приведены результаты экспериментального исследования свойств алгоритма и его сравнение с алгоритмами, сочетающими жадные стратегии и ограниченный перебор.

Ключевые слова: муравьиные алгоритмы, центр обработки данных.

1. Введение. В работе рассматривается задача планирования вычислений в центрах обработки данных (ЦОД), работающих в режиме "инфраструктура как услуга" (Infrastructure-as-a-Service, IaaS). Задача заключается в построении отображения запросов создания виртуальных машин и виртуальных систем хранения данных, соединенных виртуальными каналами, на вычислительные серверы и серверы хранения данных, и построении маршрутов для виртуальных

[1] Факультет ВМК МГУ, вед. науч. сотр., к.ф.-м.н., e-mail: kost@cs.msu.su
[2] Факультет ВМК МГУ, асп., e-mail: artacc@lvk.cs.msu.su

* Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации, уникальный номер (ID) RFMEFI60714X0070, соглашение № 14.607.21.0070.