5. *Stehlé D., Steinfeld R.* Faster fully homomorphic encryption, *Advances in Cryptology-ASIACRYPT 2010.* Springer Berlin Heidelberg, 2010, pp. 377-394.
6. *Gentry C., Halevi S.* Implementing gentry's fully-homomorphic encryption scheme, *Advances in Cryptology–EUROCRYPT 2011.* Springer Berlin Heidelberg, 2011, pp. 129-148.
7. *Zhirov A.O., Zhirova O.V., Krendelev S.F.* Bezopasnye oblachnye vychisleniya s pomoshch'yu gomomorfnoy kriptografii, *Bezopasnost' informatsionnykh tekhnologiy*. 2013, Vol. 1, pp. 6-12.
8. *Rostovtsev A., Bogdanov A., Mikhaylov M.* Secure evaluation of polynomial using privacy ring homomorphisms, *IACR Cryptology ePrint Archive*, 2011, Vol. 2011, pp. 24.
9. *Lidl R., Niderreiter H.* Finite Fields (Vol. 20, Encyclopedia of Math. and its Appl.), *Englewood Cliffs, NJ: Addison~ lVesley*. 1983, pp. 74-85.
10. *Klivans A.* Factoring polynomials modulo composites. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1997, No. CMU-CS-97-136.
11. *Benjamin A.T., Bennett C.D.* The probability of relatively prime polynomials, Mathematics *Magazine*, 2007, pp. 196-202.
12. *Shoup V.* NTL: A library for doing number theory, 2001.

Статью рекомендовал к опубликованию д.т.н., профессор Н.И. Витиска.

Трепачева Алина Викторовна – Южный федеральный университет; e-mail: alina1989malina@ya.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 89085196604; кафедра БИТ; аспирантка.

Trepacheva Alina Viktorovna – South Federal University; e-mail: alina1989malina@ya.ru; 2, Chehova street, Taganrog, 347928, Russia; phone: +79085196604; postgraduate student.

УДК 004.056.55: 003.26

Ф.Б. Буртыка

СИММЕТРИЧНОЕ ПОЛНОСТЬЮ ГОМОМОРФНОЕ ШИФРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ НЕПРИВОДИМЫХ МАТРИЧНЫХ ПОЛИНОМОВ

Представлена новая симметричная компактная полностью гомоморфная криптосхема, основанная на использовании матричных полиномов и производящая шифрование в два раунда: сначала открытые тексты, являющиеся элементами кольца вычетов, кодируются в матрицы с помощью секретного вектора $\vec{k}$, *а затем эти матрицы отображаются в матричные полиномы с использованием секретного неприводимого матричного полинома* $\mathbf{K}(X)$. *Расшифрование также происходит в два раунда: сначала осуществляется приведение по модулю* $\mathbf{K}(X)$, *а затем умножение полученной в результате матрицы на* $\vec{k}$. *Отображение расшифрования является гомоморфизмом колец. Время работы всех алгоритмов криптосхемы зависит полиномиально от параметра защищенности* λ. *Временные издержки при её использовании для вычисления над зашифрованными данными также полиномиальны от* λ. *Введение специального ключа перешифрования, зависящего от секретного ключа, позволило добиться того, что при вычислениях над шифровками их размеры всегда остаются ограниченными фиксированным полиномом от* λ. *При практической реализации возможно эффективное распараллеливание. Проводится анализ криптостойкости предложенной криптосхемы относительно атак на основе только шифротекстов, по известным открытым текстам и на ключ перешифрования. Продемонстрировано то, что все эти атаки могут быть сведены к решению системы полиномиальных уравнений от многих переменных над кольцом вычетов. Рассматривается вопрос о сложности решения этих систем существующими методами.*

Полностью гомоморфная криптосхема; матричные полиномы; системы полиномиальных уравнений; атака по известным открытым текстам; защищённые облачные вычисления.

Ph.B. Burtyka

SYMMETRIC FULLY HOMOMORPHIC ENCRYPTION USING IRREDUCIBLE MATRIX POLYNOMIALS

This paper presents a new symmetric compact fully homomorphic encryption scheme, based on usage of matrix polynomials. Its encryption algorithm proceeds in two steps: at the first step plaintexts being elements of residue class ring are encoded into matrices using a secret vector $\vec{k}$ *, then these matrices are mapped into matrix polynomials using secret irreducible matrix polynomial* $\mathbf{K}(X)$*. Decryption also proceeds in two steps: first reduction modulo* $\mathbf{K}(X)$*, and then obtained matrix is multiplied by* $\vec{k}$ *. Decryption function is a ring homomorphism. All algorithms of encryption scheme are polynomial in security parameter* λ *. Time overhead of homomorphic computations using this encryption scheme is also polynomial in* λ *. Special refresh key that depends on secret key allows keeping sizes of ciphertexts during computations over them within a fixed polynomial in* λ *bound. In real life implementation the cryptosystem enables effective parallelization. The paper analyzes the security of the proposed scheme against ciphertext only attack, known plaintext attack and refresh key attack. We demonstrate that all this attacks may be reduced to a problem of solving a system of polynomial equations over residue class ring. We discuss whether it is complex to solve it by existing methods.*

Fully homomorphic encryption scheme; matrix polynomials; system of polynomial equations; known plaintext attack; secure cloud computing.

Введение. В связи с распространением парадигмы облачных вычислений актуальной задачей становится организация гарантированно защищённых конфиденциальных вычислений. Еще Ривест, Эдлман и Дертузо описали [1] несколько основных вариантов решения этой задачи. Один из них, в оригинале называемый *гомоморфизмы конфиденциальности* (англ. *Privacy homomorphisms*), предполагал вычисление недоверенным исполнителем (сервером) некоторой функции над шифротекстами без их расшифрования. Однако, авторы концепции не надеялись увидеть работающее решение.

Среди криптосхем, построенных в рамках концепции гомоморфизмов конфиденциальности, можно выделить *аддитивно-гомоморфные* и *мультипликативно гомоморфные*. Аддитивно гомоморфные криптосхемы обладают следующим свойством: операция, проводимая над шифротекстами, соответствует сложению открытых текстов. Таковыми являются, например, криптосистема Пэйе [2]. У мультипликативно гомоморфных криптосхем операция, проводимая над шифротекстами, соответствует умножению открытых текстов. Пример мультипликативно-гомоморфной криптосистемы – RSA [3].

В 2009 г. исследователь из IBM, Крейг Джентри, предложил метод [4], который он назвал *полностью гомоморфным шифрованием* (англ. *Fully homomorphic encryption*) или сокращенно ПГШ (англ. вариант – FHE). Криптосхемы ПГШ являются одновременно аддитивно и мультипликативно гомоморфным. Если открытые тексты представляют собой биты, то логические операции AND (логическое умножение) и XOR (логическое сложение) составляют полный по Тьюрингу логический базис, т.е. через эти две операции может быть выражена любая функция[1]. Также криптосхемы ПГШ по Джентри обязаны обладать *свойством компактности*, т.е. не должно происходить неограниченного увеличения размеров шифртекстов в процессе гомоморфных вычислений.

[1] Любая булева функция может быть представлена так называемой схемой из функциональных элементов (СФЭ), т.е. в виде некоторого ориентированного графа, вершины которого помечены логическими операциями из базиса, а дуги представляют собой передачу результатов выполненной операции к следующей, при этом вершины, полустепень захода которых равна нулю (т.е. они не являются результатом какой-либо операции в рамках данной схемы), называются входами СФЭ, а вершины, полустепень исхода которых равна нулю, называются выходами СФЭ.

Построение криптосхем ПГШ открывает широкие возможности для безопасного делегирования вычислений. Однако к такому шифрованию предъявляются повышенные требования по криптостойкости. В своей работе Джентри предлагает строить ПГШ в три шага: построение гомоморфной криптосхемы для ограниченных вычислений, оптимизация алгоритма расшифрования, и наконец, применение оригинальной методики *самокоррекции* шифротекстов (англ. *bootstrapping*).

Хороший обзор достижений в области гомоморфного шифрования можно найти в работе [5].

Несмотря на большой успех новаторской работы Джентри и непрекращающиеся оптимизации [6–13], на текущий момент нет практически применимых криптосхем ПГШ. В данной статье делается попытка восполнить этот пробел.

Искусство построения криптосхем ПГШ состоит в том, что шифрование должно быть одновременно криптостойким, вычислительно эффективным и компактным. Все известные компактные криптосхемы ПГШ являются криптосхемами с открытым ключом и их криптостойкость основывается на каком-либо сложностном предположении. Однако последние исследования [14] показывают ограничения на производительность, присущие криптосхемам ПГШ с открытым ключом.

Некоторые работы, такие как [15], предлагают гомоморфизмы конфиденциальности, являющиеся симметричными криптосистемами. Однако они не могут обеспечить неограниченные гомоморфные вычисления из-за разрастания размеров шифротекстов. В работе [16] предлагается интересное решение этой проблемы: так называемый ключ умножения, который применяется после каждого умножения шифртекстов, чтобы ограничить их размер. Однако, хотя размер шифротекстов в общем и ограничен, но это ограничение не слишком существенно: оно допускает размеры шифротекста такими, что он может заполнить целиком всю оперативную память обычного настольного компьютера (!) при криптостойкости, позволяющей взломать шифр с помощью этого же компьютера.

Основные результаты. В данной работе предлагается *компактная симметричная полностью гомоморфная криптосхема* с малыми вычислительными издержками при проведении вычислений над шифротекстами, которая дает возможность простого и эффективного распараллеливания.

Шифротекстами являются *полиномы, коэффициентами которых являются матрицы (т.н. матричные полиномы)*. Секретный ключ состоит из матричного полинома $\mathbf{K}(X)$ и вектора $\vec{k}$. Идея построения криптосхемы достаточно проста: пусть есть m_1 и m_2 – открытые тексты, тогда шифротекстами будут матричные полиномы $\mathbf{C}_1(X)=\mathbf{R}_1(X)\cdot\mathbf{K}(X)+\mathbf{M}_1$ и $\mathbf{C}_2(X)=\mathbf{R}_2(X)\cdot\mathbf{K}(X)+\mathbf{M}_2$, такие что $\mathbf{M}_1\cdot\vec{k}=m_1\cdot\vec{k}$ и $\mathbf{M}_2\cdot\vec{k}=m_2\cdot\vec{k}$. Для расшифрования нужно сначала взять остаток от деления шифротекста на $\mathbf{K}(X)$, а затем извлечь открытый текст из полученной матрицы с помощью $\vec{k}$. Очевидно, что здесь выполняется аддитивный гомоморфизм. Немного сложнее обеспечить мультипликативный гомоморфизм. Хотя умножению матриц соответствует умножение их собственных значений, но при умножении матричных полиномов происходит рост их степени, поэтому необходимо её понижать. Один из способов сделать это – взятие остатка по модулю фиксированного матричного полинома.

Отметим, что поскольку кольцо матричных полиномов содержит делители нуля, необходимо установить некоторые условия, чтобы обеспечить корректность взятия остатка по модулю матричного полинома. Матричные полиномы и опера-

ции над ними вводятся чисто формально, при этом не все свойства, справедливые для обычных скалярных полиномов, справедливы и для матричных (по причине некоммутативности последних), однако при соблюдении определённых ограничений можно обеспечить выполнение тех свойств, которые необходимы для построения криптосхемы. Например, деление на матричный полином выполняется корректно при условии, что старший коэффициент полинома-делителя является единичной матрицей.

В данной статье теоретические оценки сложности всех алгоритмов даются как функции от некоторого параметра λ, называемого параметром уровня криптостойкости. Параметр защищённости λ – это целое число, управляющее соотношением между производительностью и криптостойкостью. Для того чтобы получить более защищённую (стойкую) криптосхему (например, в случае защиты особо ценных медицинских или финансовых облачных хранилищ данных), необходимо увеличить λ. Чем больше λ, тем выше криптостойкость.

Но если необходимо сделать более производительную систему (к примеру, в случае защиты мобильных приложений и сетей), следует взять меньшее λ (однако, тем самым, снижая криптостойкость). Использование такого параметра – распространённая практика при построении криптосхем ПГШ [4].

Общая архитектура предлагаемой организации вычислений и определения. Предлагаемая организация системы защищённых облачных вычислений (рис. 1.) включает двух участников: *клиент* и *сервер*. Протокол взаимодействия клиента и сервера выглядит следующим образом: сначала клиент генерирует секретный ключ (с уровнем криптостойкости, учитываемым с помощью параметра λ), позволяющий зашифровывать и расшифровывать сообщения. Также клиент генерирует некоторую дополнительную информацию (учитывая через параметр λ необходимый уровень криптостойкости), позволяющую ограничивать рост шифртекстов в процессе гомоморфных вычислений, но не позволяющую зашифровывать или расшифровывать. Назовем эту информацию *ключом перешифрования*. После отправки ключа перешифрования серверу он подготовлен к выполнению основной части работы – проведению гомоморфных вычислений над шифротекстами и взаимодействию с клиентом.

Формально, гомоморфная криптосхема $\mathcal{E}$ представляет собой четвёрку алгоритмов $\left(\mathrm{KeyGen}_\varepsilon, \mathrm{Encrypt}_\varepsilon, \mathrm{Decrypt}_\varepsilon, \mathrm{Evaluate}_\varepsilon\right)$. Вероятностный алгоритм $\mathrm{KeyGen}_\varepsilon$, принимающий на вход параметр уровня криптостойкости λ, и выдает в качестве результата пару ключей (sk, rk), где sk – секретный ключ, который хранится у клиента, а rk – ключ перешифрования, передаваемый серверу (он позволяет серверу сокращать размер шифртекстов в процессе вычислений, но не позволяет зашифровывать или расшифровывать). Алгоритмы $\mathrm{Encrypt}_\varepsilon$ и $\mathrm{Decrypt}_\varepsilon$ принимают на вход, соответственно, шифртекст или открытый текст вместе с секретным ключом sk. Алгоритм $\mathrm{Evaluate}_\varepsilon$ принимает на вход СФЭ F, набор шифротекстов $\langle m_1, ..., m_t \rangle$, ключ перешифрования rk, и выдает в качестве результата шифртекст *c*. Вычислительная сложность всех этих алгоритмов должна быть полиномиальна от параметра уровня криптостойкости λ и (в случае алгоритма $\mathrm{Evaluate}_\varepsilon$) количества схемных элементов F, а также они должны удовлетворять приведенным ниже требованиям корректности.

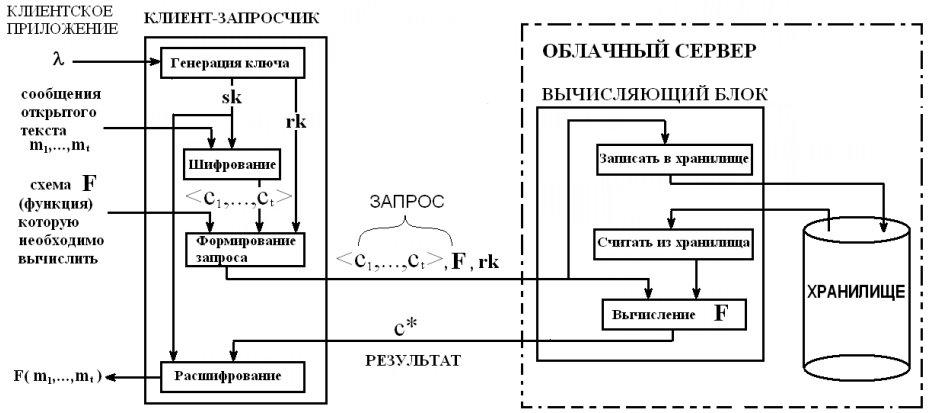

Рис. 1. Предлагаемая организация системы защищённых облачных вычислений.

Определение 1. (Корректность расшифрования после гомоморфного вычисления). Криптосхема $\varepsilon = (\mathrm{KeyGen}, \mathrm{Encrypt}, \mathrm{Decrypt}, \mathrm{Evaluate})$ корректна для СФЭ F, имеющей t входов, если для любой пары ключей (sk, rk), выданной алгоритмом $\mathrm{KeyGen}(\lambda)$, любых t открытых текстов m_i и соответствующих им шифртекстов $c_i \leftarrow \mathrm{Encrypt}(\mathbf{sk}, m_i)$ выполняется:

$$\mathrm{Decrypt}\left(\mathbf{sk}, \mathrm{Evaluate}(\mathbf{rk}, F, c)\right) = F(m_1, ..., m_t)\,.$$

Определение 2. Криптосхема $\varepsilon = (\mathrm{KeyGen}, \mathrm{Encrypt}, \mathrm{Decrypt}, \mathrm{Evaluate})$ полностью гомоморфна для класса СФЭ, если она корректна для всех СФЭ из этого класса.

Определение 3. Гомоморфная криптосхема называется *компактной*, если размер шифротекстов, получающихся в результате гомоморфного вычисления произвольной функции f над шифртекстами, не зависит от размера схемы из функциональных элементов, представляющей f, и ограничен полиномом $\beta(\lambda)$.

Матричные полиномы. Пусть $\mathbb{Z}_p^{N\times N}$ – кольцо $N \times N$ матриц с элементами из кольца $\mathbb{Z}_p$ целых чисел по модулю числа p. Рассмотрим множество последовательностей матриц из $\mathbb{Z}_p^{N\times N}$:

$$F = \{\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, ...\}, \mathbf{A}_i \in \mathbb{Z}_p^{N\times N},$$

таких, что все $\mathbf{A}_i$, кроме конечного их числа, равны нулевой матрице. Пусть $\mathbb{Z}_p^{N\times N}[X]$ обозначает множество всех таких последовательностей. Если $F, G \in \mathbb{Z}_p^{N\times N}[X]$, $G = \{\mathbf{B}_0, \mathbf{B}_1, \mathbf{B}_2, ...\}, \mathbf{B}_i \in \mathbb{Z}_p^{N\times N}$, то определим

$$\begin{aligned} F + G &= \{\mathbf{A}_0 + \mathbf{B}_0, \mathbf{A}_1 + \mathbf{B}_1, \mathbf{A}_2 + \mathbf{B}_2, ...\}, \\ F \cdot G &= \{\mathbf{A}_0 \cdot \mathbf{B}_0, \mathbf{A}_0\mathbf{B}_1 + \mathbf{A}_1\mathbf{B}_0, \mathbf{A}_0\mathbf{B}_2 + \mathbf{A}_1\mathbf{B}_1 + \mathbf{A}_2\mathbf{B}_0\} = \{\mathbf{C}_k\}, \end{aligned} \tag{1}$$

где $\mathbf{C}_k = \sum_{i+j=k} \mathbf{A}_i \cdot \mathbf{B}_j, k = 0,1,2,...$.

Можно показать, что при таких определениях сложения и умножения множество $\mathbb{Z}_p^{N\times N}[X]$ становится кольцом. Элементы этого кольца будем называть *матричными полиномами*.

Лемма 1. *Матричные полиномы образуют кольцо.*

Доказательство. Проверяется непосредственно выполнение свойств кольца.

Приведенный матричный полином – это такой полином, у которого коэффициент при старшей степени равен единичной матрице.

Лемма 2. (Корректность деления на приведенный матричный полином). Пусть $\mathbf{K}(X)\in\mathbb{Z}_n^{N\times N}[X]$ – приведённый матричный полином. Тогда для каждого матричного полинома $\mathbf{C}(X)\in\mathbb{Z}_n^{N\times N}[X]$ такого, что $\deg(\mathbf{K}(X))\le\deg(\mathbf{C}(X))$ представление в виде $\mathbf{C}(X)=\mathbf{Q}(X)\cdot\mathbf{K}(X)+\mathbf{R}(X)$, где $\deg(\mathbf{K}(X))>\deg(\mathbf{R}(X))$, существует и единственно.

Доказательство. Рассмотрим алгоритм деления полинома $\mathbf{C}(X)$ на полином $\mathbf{K}(X)$ «в столбик»:

1. Домножить $\mathbf{K}(X)$ на $X^{\deg(\mathbf{C}(X))-\deg(\mathbf{K}(X))}$ и на такое $\mathbf{A}\in\mathbb{Z}_n^{N\times N}$, чтобы старшие коэффициенты полиномов $\mathbf{C}(X)$ и $\mathbf{A}\cdot\mathbf{K}(X)\cdot X^{\deg(\mathbf{C}(X))-\deg(\mathbf{K}(X))}$ стали равными.
2. Вычесть $\mathbf{C}(X):=\mathbf{C}(X)-\mathbf{A}\cdot\mathbf{K}(X)\cdot X^{\deg(\mathbf{C}(X))-\deg(\mathbf{K}(X))}$.
3. Если $\deg(\mathbf{K}(X))>\deg(\mathbf{C}(X))$, то алгоритм возвращает в качестве результата текущее $\mathbf{C}(X)$, иначе переход к шагу 1.

Очевидно, что если $\mathbf{K}(X)$ является приведённым полиномом, то шаг 1 всегда может быть выполнен корректно. □

Каждому матричному полиному $\mathbf{P}(X)$ можно естественным образом сопоставить матричное уравнение $\mathbf{P}(X)=\mathbf{0}$. Интересно, что такое матричное уравнение может иметь корней больше, чем его степень [17], а может и не иметь корней совсем. В случае если такое уравнение не имеет корней, соответствующий матричный полином будем называть *неприводимым*.

Основное построение. Пусть $\lambda\in\mathbb{N}$ ($\mathbb{N}$ обозначает множество натуральных чисел) – параметр, обозначающий уровень криптостойкости, $\mathbb{Z}_p$ – пространство открытых текстов, $\mathbb{Z}_p^{N\times N}[X]$ – пространство шифротекстов, $\mathbb{Z}_p^{N\times N}[X]\times\mathbb{Z}_p^N$ – пространство секретных ключей, где $N=O(\lambda)$, p – простое число. Также для нашей криптосхемы кроме секретного ключа нужен так называемый *ключ перешифрования* $\mathbf{rk}$, который передается серверу для сокращения размеров шифротекстов в процессе вычислений. Он является элементом $\mathbb{Z}_p^{N\times N}[X]$.

Для удобства введем следующие обозначения: 1) $s\xleftarrow{\$}R$ означает, что s из R выбирается по равномерному распределению; 2) $D_{R,\mu,\sigma}$ – нормальное распределение над R с математическим ожиданием μ и дисперсией σ.

Теперь опишем алгоритмы нашей симметричной криптосхемы.

Генерация секретного ключа

1) Генерируется приведенный полином $\mathbf{K}(X) \in \mathbb{Z}_p^{N\times N}[X]$, не имеющий корней, такой, что $\deg(\mathbf{K}(X)) = O(\lambda)$ выбирается по распределению $D_{\mathbb{N},\mu,\sigma}$, $\mu = O(\lambda)$, $\sigma = O(\lambda)$, а его коэффициенты $\mathbf{K}_i \xleftarrow{\$} \mathbb{Z}_p^{N\times N}, i = 0,...,\deg(\mathbf{K}(X)) - 1$.
2) Генерируется вектор $\vec{k} \in \mathbb{Z}_p^N$, $k_i \xleftarrow{\$} \mathbb{Z}_p$ такой, что хотя бы одна координата вектора должна быть обратимой в $\mathbb{Z}_p$. Итого, на выходе алгоритма секретный ключ $\mathbf{sk} = \{\mathbf{K}(X), \vec{k}\}$.

Генерация ключа перешифрования. Генерируется приведенный матричный полином $\mathbf{R}'(X) \in \mathbb{Z}_p^{N\times N}[X]$ такой, что $\deg(\mathbf{R}'(X)) = O(\lambda)$ выбирается по распределению $D_{\mathbb{N},\mu^*,\sigma^*}$, $\mu^* = O(\lambda)$, $\sigma^* = O(\lambda)$, $\mathbf{R}'_i \xleftarrow{\$} \mathbb{Z}_p^{N\times N}, i = 0,...,\deg(\mathbf{R}'(X)) - 1$. Тогда ключ перешифрования – это полином $\mathbf{rk}(X) = \mathbf{R}'(X) \cdot \mathbf{K}(X)$.

Шифрование

1) Открытому тексту $m \in \mathbb{Z}_p$ сопоставляется случайная матрица $\mathbf{M} \in \mathbb{Z}_p^{N\times N}$, такая, что $\mathbf{M} \cdot \vec{k} = m \cdot \vec{k}$ и $\mathbf{M} \cdot \mathbf{K}(X) = \mathbf{K}(X) \cdot \mathbf{M}$, т.е. она имеет собственный вектор $\vec{k}$ при собственном значении m и коммутирует с матричным полиномом $\mathbf{K}(X)$ (заметим, что такой выбор всегда возможен, например, в качестве матрицы $\mathbf{M}$ можно взять матрицу, кратную единичной).
2) Генерируется $\mathbf{R}(X) \in \mathbb{Z}_p^{N\times N}[X]$, где $\deg \mathbf{R}(X) = O(\lambda)$ выбирается по $D_{\mathbb{N},\mu^{**},\sigma^{**}}$, $\mu^{**} = O(\lambda)$, $\sigma^{**} = O(\lambda)$, так что $\deg(\mathbf{R}(X)) < \deg(\mathbf{R}'(X))$, $\mathbf{R}_i \xleftarrow{\$} \mathbb{Z}_p^{N\times N}, i = 0,...,\deg(\mathbf{R}(X))$.
3) Вычисляется шифртекст $\mathbf{C}(X) = \mathbf{R}(X) \cdot \mathbf{K}(X) + \mathbf{M}$.

Расшифрование

1) Вычисляется $\mathbf{M} = \mathbf{C}(X) \bmod \mathbf{K}(X)$.
2) Для обратимой координаты k_i вычисляется $m = k_i^{-1}\left(\mathbf{M} \cdot \vec{k}\right)$.

Гомоморфное вычисление. Сопоставление полиному $f(x_1,...,x_t)$ над $m_1,...,m_t \in \mathbb{Z}_p$ полинома $f^{\perp}(\mathrm{X}_1,...,\mathrm{X}_t)$ над соответствующими шифротекстами $\mathbf{C}_1,...,\mathbf{C}_t$ осуществляется простой заменой операций над $\mathbb{Z}_p$ на сложение и умножение полиномов в $\mathbb{Z}_p^{N\times N}[X]$. Для предотвращения роста степени матричных полиномов после их умножения осуществляется приведение по модулю $\mathbf{rk}(X)$.

Рассмотрим теперь вопрос о корректности построенной криптосхемы, т.е. о соответствии представленных алгоритмов определениям, данным выше.

Лемма 3. *Расшифрование вышеописанной криптосхемы корректно и является гомоморфизмом для всех арифметических схем, состоящих из сложений и умножений по модулю* p**.**

Доказательство: 1) Корректность расшифрования. Рассмотрим выражение $\left((\mathbf{R}(X) \cdot \mathbf{K}(X) + \mathbf{M}) \bmod (\mathbf{K}(X))\right) \cdot \vec{k}$, соответствующее расшифрованию $\mathbf{C}(X) = \mathbf{R}(X) \cdot \mathbf{K}(X) + \mathbf{M}$, где в $\mathbf{M}$ закодирован открытый текст m. Младший

коэффициент $\mathbf{C}(X)$ – это $\mathbf{R}_0 \cdot \mathbf{K}_0 + \mathbf{M}$ и алгоритм деления многочленов «в столбик» оставляет $\mathbf{M}$ при соблюдении порядка деления (при умножении матриц всегда слева или всегда справа). Итак, для любой обратимой координаты k_i справедливо $\left(((\mathbf{R}(X) \cdot \mathbf{K}(X) + \mathbf{M}) \bmod \mathbf{K}(X)) \cdot \vec{k}\right)_i \cdot k_i^{-1} = m$.

2) Аддитивный и мультипликативный гомоморфизмы. Пусть $\mathbf{C}_1(X) = \mathbf{R}_1(X) \cdot \mathbf{K}(X) + \mathbf{M}_1$ и $\mathbf{C}_2(X) = \mathbf{R}_2(X) \cdot \mathbf{K}(X) + \mathbf{M}_2$ – два шифротекста, шифрующих m_1 и m_2 соответственно. Их сумма $\mathbf{C}_1(X) + \mathbf{C}_2(X) = (\mathbf{R}_1(X) + \mathbf{R}_2(X)) \cdot \mathbf{K}(X) + \mathbf{M}_1 + \mathbf{M}_2$ является корректным шифртекстом (шифртекстом правильного вида) и после расшифрования даёт $(m_1 + m_2) \bmod p$, поскольку $(\mathbf{M}_1 + \mathbf{M}_2) \cdot \vec{k} = \mathbf{M}_1 \cdot \vec{k} + \mathbf{M}_2 \cdot \vec{k} = m_1 \cdot \vec{k} + m_2 \cdot \vec{k} = (m_1 + m_2) \cdot \vec{k}$.

Произведение шифротекстов

$$\mathbf{C}_1(X) \cdot \mathbf{C}_2(X) = \mathbf{R}_1(X) \cdot \mathbf{K}(X) \cdot \mathbf{R}_2(X) \cdot \mathbf{K}(X) +$$
$$+\mathbf{R}_1(X) \cdot \mathbf{K}(X) \cdot \mathbf{M}_2 + \mathbf{M}_1 \cdot \mathbf{R}_2(X) \cdot \mathbf{K}(X) + \mathbf{M}_1 \cdot \mathbf{M}_2 =$$
$$= (\mathbf{R}_1(X) \cdot \mathbf{K}(X) \cdot \mathbf{R}_2(X) + \mathbf{R}_1(X) \cdot \mathbf{M}_2 + \mathbf{R}_2(X) \cdot \mathbf{M}_1) \cdot \mathbf{K}(X) + \mathbf{M}_1 \cdot \mathbf{M}_2$$

также является корректным шифртекстом и после расшифрования даёт $(m_1 \cdot m_2) \bmod p$, поскольку

$$(\mathbf{M}_1 \cdot \mathbf{M}_2) \cdot \vec{k} = \mathbf{M}_1 \cdot (\mathbf{M}_2 \cdot \vec{k}) = \mathbf{M}_1 \cdot (m_2 \cdot \vec{k}) = m_2 \cdot (\mathbf{M}_1 \cdot \vec{k}) = m_1 \cdot m_2 \cdot \vec{k}.$$

И наконец осталось заметить, что остаток $\mathbf{C}(X)$ по модулю приведённого полинома $\mathbf{rk}$ является корректным шифртекстом, шифрующим тот же самый открытый текст m. Действительно, имеем:

$$\mathbf{C}(X) \bmod \mathbf{rk} = (\mathbf{R}(X) \cdot \mathbf{K}(X) + \mathbf{M}) \bmod (R'(X) \cdot \mathbf{K}(X)) =$$
$$= \mathbf{R}(X) \cdot \mathbf{K}(X) + \mathbf{M} - \mathbf{P}(X) \cdot R'(X) \cdot \mathbf{K}(X) = \mathbf{R}_{new}(X) \cdot \mathbf{K}(X) + \mathbf{M},$$

где $\deg(\mathbf{R}_{new}(X) \cdot \mathbf{K}(X) + \mathbf{M}) < \deg(\mathbf{rk})$. □

Лемма 4. *Вышеописанная криптосхема компактна*.

Доказательство: Существует полином, который ограничивает степень полинома шифртекста, поскольку $\deg\left(\mathbf{R}_{new}(X) \cdot \mathbf{K}(X) + \mathbf{M}\right) < \deg(\mathbf{rk}) = O(\lambda)$. Таким образом, полином шифротекста может быть записан с использованием числа битов, которое выражается полиномом от параметра защищённости λ.□

Вычислительные издержки. Вычисляемая над открытыми текстами арифметическая схема состоит из элементов сложения и умножения по модулю p. При замене каждой такой операции над открытыми текстами на операцию над шифротекстами происходит увеличение количества операций в $\mathbb{Z}_p$. Наибольшее увеличение происходит в случае операции умножения. Поэтому для получения верхней оценки на вычислительные издержки рассмотрим более подробно этот случай.

Сложность умножения матричных полиномов зависит от сложности двух алгоритмов: алгоритма умножения матриц и алгоритма умножения полиномов – умножение двух полиномов состоит в проведении определенного количества операций над матрицами, в свою очередь каждая операция над матрицами требует необходимого количества операций с их элементами.

Алгоритм умножения полиномов, который не требует соблюдения специфических условий (таких как отсутствие делителей нуля), имеет асимптотическую сложность $O(d^{1.5849\ldots})$ операций над коэффициентами полиномов, где d – наи-

большая из степеней полиномов [18]. Для вычисления последующего приведения произведения по модулю потребуется приблизительно столько же операций. Алгоритм умножения двух $N \times N$ матриц имеет асимптотическую сложность $O(N^{2.373..})$ элементарных операций [19].

Следовательно, при условии, что $N = O(\lambda)$ и степени полиномов шифртекстов равны $O(\lambda)$, асимптотическая оценка на общее число операций над элементами кольца открытых текстов, необходимых для гомоморфного сложения/умножения $\approx O(\lambda^{3.76})$.

Замечание. *Проведенный анализ последних достижений в области построения ПГШ показал, что на данный момент наилучшая оценка на вычислительные издержки при гомоморфном вычислении составляет* $g(\lambda) = \tilde{O}(\lambda^{3.5})$ *[12]. Лучшие оценки (* $g(\lambda) = O(\lambda^2)$ *и* $g(\lambda) = \tilde{O}(\lambda)$*) получены пока только для схем ПГШ для ограниченных вычислений [13]. Хотя в вышеописанной криптосхеме вычислительные издержки приблизительно такие же, как в работе [12], она имеет существенное преимущество: для ускорения работы криптосистемы можно естественно использовать всевозможные методы распараллеливания операций над матрицами [20].*

Анализ криптостойкости полученного шифрования. Для анализа криптостойкости сначала необходимо определить размер пространства ключей, поскольку если он будет небольшой, то такую криптосхему будет легко взломать полным перебором. В нашем случае секретными ключами являются примитивные матричные полиномы, имеющие нетривиальный коммутант. Несложно видеть, что общее количество примитивных матричных полиномов как функция имеет экспоненциальную зависимость от степени матричного полинома и от размерности используемых матриц, а компьютерные эксперименты показали, что и количество примитивных матричных полиномов с нетривиальным коммутантом также экспоненциально.

Атака на ключ перешифрования. Специфичной для нашей матричной криптосхемы является *атака на ключ перешифрования*. Этот ключ содержит в себе в неявном виде информацию о секретном ключе расшифрования, поэтому может стать слабым местом криптосхемы. Необходимо показать, чтоб раскрытие секретного ключа по ключу перешифрования эквивалентно решению некоторой известной сложной NP-полной задачи [21]. В данном случае, очевидно, что раскрытие **sk** по **rk** эквивалентно решению о факторизации матричных полиномов.

Определение 4. (*Задача факторизации матричных полиномов*).

Экземпляр (N, d, p, r)-задачи факторизации матричных полиномов состоит в том, чтобы по заданному матричному полиному $\mathbf{F}(X)$ степени d с коэффициентами из $\mathbb{Z}_p^{N \times N}$, ответить на вопрос «возможно ли разложение $\mathbf{F}(X) = \mathbf{F}_{left}(X) \cdot \mathbf{F}_{right}(X)$, такое что $\deg(\mathbf{F}_{right}(X)) = r$?», и если ответ положительный, то найти все такие $\mathbf{F}_{right}(X)$.

Для анализа сложности задачи факторизации матричных полиномов сведем её к задаче поиска решений некоторой системы полиномиальных уравнений над $\mathbb{Z}_p$.

Лемма 5. *Задача* (N, d, p, r) *поиска факторизации матричных полиномов эквивалентна решению системы из* $r \cdot N^2$ *алгебраических уравнений от* $(r+1) \cdot N^2$ *переменных.*

Доказательство. Пусть $\mathbf{C}(X)=\mathbf{R}(X)\cdot\mathbf{K}(X)$, $\deg\left(\mathbf{C}(X)\right)=d$, и мы ищем такое $\mathbf{K}(X)$, что $\deg\left(\mathbf{K}(X)\right)=r$. Тогда $\mathbf{C}_d\cdot X^d+\mathbf{C}_{d-1}\cdot X^{d-1}+...+\mathbf{C}_1\cdot X+\mathbf{C}_0=$

$=(\mathbf{R}_{d-r}\cdot X^{d-r}+...+\mathbf{R}_1\cdot X+\mathbf{R}_0)\cdot(\mathbf{K}_r\cdot X^r+\mathbf{K}_{r-1}\cdot X^{r-1}+...+\mathbf{K}_1\cdot X+\mathbf{K}_0)$.

Запишем формальные выражения для коэффициентов произведения в соответствии с формулами (1) формального определения операций. Получим систему матричных уравнений следующего вида:

$$\begin{cases}\mathbf{C}_d=\mathbf{R}_{d-r}\cdot\mathbf{K}_r,\\ \mathbf{C}_{d-1}=\mathbf{R}_{d-r}\cdot\mathbf{K}_{r-1}+\mathbf{R}_{d-r-1}\cdot\mathbf{K}_r,\\ \mathbf{C}_{d-2}=\mathbf{R}_{d-r}\cdot\mathbf{K}_{r-2}+\mathbf{R}_{d-r-1}\cdot\mathbf{K}_{r-1}+\mathbf{R}_{d-r-2}\cdot\mathbf{K}_r,\\ ...\\ \mathbf{C}_1=\mathbf{R}_0\cdot\mathbf{K}_1+\mathbf{R}_1\cdot\mathbf{K}_0,\\ \mathbf{C}_0=\mathbf{R}_0\cdot\mathbf{K}_0.\end{cases}$$

В ней можно перенести в правую часть первых $d-r+1$ уравнений неизвестные коэффициенты $\mathbf{R}_i$:

$$\begin{cases}\mathbf{C}_d\cdot\mathbf{K}_r^{-1}=\mathbf{R}_{d-r},\\ (\mathbf{C}_{d-1}-\mathbf{R}_{d-r}\cdot\mathbf{K}_{r-1})\cdot\mathbf{K}_r^{-1}=\mathbf{R}_{d-r-1},\\ (\mathbf{C}_{d-2}-\mathbf{R}_{d-r}\cdot\mathbf{K}_{r-2}-\mathbf{R}_{d-r-1}\cdot\mathbf{K}_{r-1})\cdot\mathbf{K}_r^{-1}=\mathbf{R}_{d-r-2},\\ ...\\ (\mathbf{C}_r-\mathbf{R}_1\cdot\mathbf{K}_{r-1}-...-\mathbf{R}_{\max(r,d-r)}\cdot\mathbf{K}_{\max(0,2r-d)})\cdot\mathbf{K}_r^{-1}=\mathbf{R}_0.\end{cases}$$

а затем выразить в каждом уравнении $\mathbf{R}_i$ через $\mathbf{R}_{i+j}$ (более старшие, уже выраженные коэффициенты), т.е.

$$(\mathbf{C}_{d-1}-\mathbf{R}_{d-r}\cdot\mathbf{K}_{r-1})\cdot\mathbf{K}_r^{-1}=\mathbf{R}_{d-r-1}\Rightarrow(\mathbf{C}_{d-1}-(\mathbf{C}_d\cdot\mathbf{K}_r^{-1})\cdot\mathbf{K}_{r-1})\cdot\mathbf{K}_r^{-1}=\mathbf{R}_{d-r-1}$$

$$(\mathbf{C}_{d-2}-\mathbf{R}_{d-r}\cdot\mathbf{K}_{r-2}+\mathbf{R}_{d-r-1}\cdot\mathbf{K}_{r-1})\cdot\mathbf{K}_r^{-1}=\mathbf{R}_{d-r-2}\Rightarrow$$

$$\Rightarrow(\mathbf{C}_{d-2}-(\mathbf{C}_d\cdot\mathbf{K}_r^{-1})\cdot\mathbf{K}_{r-2}+(\mathbf{C}_{d-1}-(\mathbf{C}_d\cdot\mathbf{K}_r^{-1})\cdot\mathbf{K}_{r-1})\cdot\mathbf{K}_r^{-1}\cdot\mathbf{K}_{r-1})\cdot\mathbf{K}_r^{-1}=\mathbf{R}_{d-r-2}$$

и т.д. В результате все $\mathbf{R}_i$ будут выражены в виде полиномов от неизвестных $\mathbf{K}_j$ и известных матриц $\mathbf{C}_d,...,\mathbf{C}_r$. Если затем подставить эти выражения для $\mathbf{R}_i$ в уравнения для *r* младших коэффициентов $\mathbf{C}(X)$, то получится *r* матричных уравнений относительно *r+1* матричных неизвестных $\mathbf{K}_0,\mathbf{K}_1,...,\mathbf{K}_r$.

Полученную систему матричных уравнений можно преобразовать, пользуясь законом умножения матриц, в систему из $r\cdot N^2$ скалярных алгебраических уравнений над $\mathbb{Z}_p$ от $(r+1)\cdot N^2$ переменных. Согласно правилу умножения матриц общее количество различных мономов в системе будет значительно превосходить величину, равную $\sum_{i=0}^{d-2}N^i$.□

Замечание. Лемма 5 применима не только для матричных, но и для любых полиномов, однако для полиномов над конечными полями существуют другие, эффективные алгоритмы факторизации, использующие отсутствие делителей нуля.

Лемма 5 означает, что атака на ключ перешифрования $\mathbf{rk}(X) = \mathbf{R}'(X) \cdot \mathbf{K}(X)$ сводится к решению *d-2* системы полиномиальных уравнений (СПУ), соответствующих *d-2* гипотезам о степени ключа. Известно, что, в общем, решение СПУ над $\mathbb{Z}_p$ является трудной задачей. Тем не менее, также известно, что существуют классы системы полиномиальных уравнений, которые можно легко решить. В данной работе мы пока что не даем строгого доказательства того, что полученные в результате криптоанализа системы уравнений являются гарантированно сложными для решения экземплярами. Однако ниже мы проводим анализ того, насколько эти системы могут быть сложны для решения некоторыми стандартными методами.

a) Базисы Грёбнера. Уже при небольшом N (приблизительно $N > 6$) количество переменных $(r+1) \cdot N^2$ становится значительным и решение каждой такой системы с помощью базисов Грёбнера [22] становится неэффективным.

b) Линеаризация. Проанализируем теперь возможность решения данной системы с помощью метода линеаризации [22]. В предположении, что $\mathrm{d} = \deg(C) = O(\lambda),\ N = O(\lambda)$, количество уравнений $(r+1) \cdot N^2$ ограничено сверху величиной $O(\lambda^3)$. В свою очередь число различных мономов в системе ограничено снизу величиной $\sum_{i=0}^{d-2} N^i = \frac{N^{d-1}-1}{N-1} \approx O(\lambda^\lambda)$. Выбрав даже достаточно небольшое $\lambda \geq 16$, можно добиться того, что применение метода линеаризации не будет эффективным, поскольку разрыв между количеством мономов и уравнений будет значительным. Например, при $\lambda = 16$ уравнений будет $< 2^{12}$, а мономов $> 2^{32}$. Тогда после решения линеаризованной системы нужно будет опробовать на роль решения исходной системы $> 2^{20}$ векторов, что уже значительно.

c) XL метод. Применение XL метода в данном случае также не будет эффективным уже при $\lambda \geq 16$. Действительно, при $\lambda = 16$ в исходной линеаризованной системе будет $> 2^{32}$ переменных. После преобразования системы XL методом переменных станет еще больше, а также число уравнений возрастет до количества $> 2^{32}$. Решение получившейся СЛАУ в итоге займет больше, чем 2^{96} арифметических операций.

Атака на основе шифртекстов. По данной в открытом виде матрице $\mathbf{M} \in \mathbb{Z}_p^{N \times N}$ легко найти её собственные значения и собственные векторы. Однако если дан шифротекст вида $\mathbf{C}(X) = \mathbf{R}(X) \cdot \mathbf{K}(X) + \mathbf{M}$, то его свободный коэффициент не раскрывает информацию о спектре матрицы $\mathbf{M}$, поскольку $\mathbf{C}_0 = \mathbf{R}_0 \cdot \mathbf{K}_0 + \mathbf{M}$, где $\mathbf{R}_0$ – равномерно случайная матрица.

Предположим, что криптоаналитик перехватил последовательность шифротекстов $\{\mathbf{C}_i(X)\}_{i=1}^t$. Сделав предположение о степени секретного ключа $r = \deg(\mathbf{K}(X))$, он может записать систему матричных уравнений вида

$$\begin{cases} \mathbf{C}_i(X) = \mathbf{R}_i(X) \cdot \mathbf{K}(X) + \mathbf{M}_i, \\ \mathbf{K}(X) \cdot \mathbf{M}_i = \mathbf{M}_i \cdot \mathbf{K}(X), \\ \mathbf{rk}(X) = \mathbf{R}'(X) \cdot \mathbf{K}(X). \end{cases} \tag{2}$$

относительно матричных неизвестных

$$\{\mathbf{K}_i\}_{i=0}^r,\ \{\mathbf{M}_i\}_{i=1}^t,\ \{(\mathbf{R}_i)_j\}_{i=1,\ j=1}^{t,\ \deg(\mathbf{R}_j)},\ \{\mathbf{R}'_j\}_{j=1}^{\deg(\mathbf{R}')}.$$

Поступая таким же образом, как было описано в лемме 5, для каждого $\mathbf{C}_i(X)$ и можно выразить неизвестные $\{(\mathbf{R}_i)_j\}_{j=1}^{\deg(\mathbf{R}_j)}$ через $\{\mathbf{K}_j\}_{j=0}^{r}$, используя старшие $\deg(\mathbf{C}_i(X))-r$ коэффициентов $\mathbf{C}_i(X)$. Затем, подставляя полученные выражения в уравнения для r младших коэффициентов,

$$\begin{cases}(\mathbf{C}_i)_{r-1}=(\mathbf{R}_i)_0\cdot\mathbf{K}_{r-1}+\ldots+(\mathbf{R}_i)_{\max\{r-1,d-r\}}\cdot\mathbf{K}_{\min\{0,2r-d-1\}},\\ \ldots\\ (\mathbf{C}_i)_1=(\mathbf{R}_i)_0\cdot\mathbf{K}_1+(\mathbf{R}_i)_1\cdot\mathbf{K}_0,\\ (\mathbf{C}_i)_0=(\mathbf{R}_i)_0\cdot\mathbf{K}_0+\mathbf{M}_i.\end{cases}$$

криптоаналитик получит r матричных уравнений относительно $r+2$ неизвестных $\{\mathbf{K}_j\}_{j=0}^{r}$ и $\mathbf{M}_i$. Аналогично для $\mathbf{rk}(X)$ можно получить r уравнений от неизвестных $\{\mathbf{K}_j\}_{j=0}^{r}$.

Собрав все вместе, криптоаналитик получит систему из $r\cdot t+r+t$ матричных уравнений от $r+t+1$ неизвестных $\{\mathbf{K}_j\}_{j=0}^{r}$ и $\{\mathbf{M}_i\}_{i=1}^{t}$. Воспользовавшись правилом умножения матриц, криптоаналитик получит соответствующую ей систему из $N^2\cdot(r\cdot t+r+t)$ скалярных полиномиальных уравнений от $N^2\cdot(r+t)$ неизвестных.

Ясно, что так же, как и в случае атаки на ключ перешифрования, при небольшом N количество переменных становится значительным. Тогда решение системы с помощью базисов Грёбнера становится неэффективным.

Теперь рассмотрим применимость линеаризации. Предположим, как и ранее, что $\deg(\mathbf{C}_i)=O(\lambda),\ \deg(\mathbf{rk})=O(\lambda),\ N=O(\lambda)$. А также будем считать, что $t=O(\lambda^{\delta}),\ \delta<<\lambda$ (это стандартное предположение при криптоанализе). В этом случае количество уравнений в системе ограничено сверху величиной $O(\lambda^{\beta}),\ \beta=\delta+3,\ \beta<<\lambda$. Нижняя же оценка (недостижимая) на количество различных мономов будет следующей:

$$t\cdot N^2+\sum_{i=0}^{\max\{\deg(\mathbf{C}_i),\mathbf{rk}\}-2}N^i\approx O(\lambda^{\lambda}).$$

Ясно, что при таких условиях решение системы методом линеаризации будет неэффективным при $\lambda\geq 16$ (по тем же соображениям, что и выше).

Рассмотрим случай, когда $\beta\approx\lambda$. Количества уравнений и переменных в линеаризованной системе могут оказаться близкими, однако размеры системы окажутся слишком большими. В частности, при $\lambda>8$ придется решать линеаризованную систему уравнений из 2^{24} уравнений от более чем 2^{24} неизвестных. Это уже является трудной задачей.

По аналогичным соображениям применение XL метода также не будет эффективным.

Замечание. Криптоаналитик не знает степень ключа, поэтому он вынужден перебирать r и для каждого $r\in\{2,\min\{\deg(\mathbf{C}_i),\deg(\mathbf{rk})\}-1\}$ составлять и решать полиномиальную систему.

Атака по известным открытым текстам. Предположим криптоаналитик перехватил пары (шифртекст, открытый текст) – $\{\mathbf{C}_i(X),m_i\}_{i=1}^{t}$. Тогда к системе (2) добавятся соотношения вида $\mathbf{M}_i\cdot\vec{k}=m_i\cdot\vec{k},\ i=\overline{1,t}$. В скалярном виде каждое из них может быть переписано как

$$\begin{cases} \sum_{l=1}^{N} (\mathbf{M}_i)_{1,l} \cdot k_l = k_1 \cdot m_i, \\ \sum_{l=1}^{N} (\mathbf{M}_i)_{2,l} \cdot k_l = k_2 \cdot m_i, \\ \ldots \\ \sum_{l=1}^{N} (\mathbf{M}_i)_{N,l} \cdot k_l = k_N \cdot m_i. \end{cases}$$

Полученные $N \cdot t$ уравнений необходимо добавить к $N^2 \cdot (r \cdot t + r + t)$ уравнениям, которые были получены при анализе по шифртекстам. Количество неизвестных теперь будет равным $N^2 \cdot (r+t) + N$. А количество мономов будет ограничено снизу величиной $2 \cdot t \cdot N^2 + \sum_{i=0}^{\max\{\deg(\mathbf{C}_i),\mathbf{rk}\}-2} N^i$. Рассуждая так же, как и раньше, можно прийти к выводу, что методы линеаризации, XL и вычисления базисов Грёбнера окажутся неэффективными при решении полиномиальной системы, составленной относительно коэффициентов ключа.

Экспериментальная программная реализация криптосхемы. С целью экспериментальной оценки производительности описанной криптосхемы были реализованы программные блоки операций с матричными полиномами, а также некоторый «тестовый» вариант самой криптосхемы.

В таблице представлены замеры времени работы алгоритмов на компьютере с процессором Quad Core Celeron 1.7 ГГц и 4 Гб оперативной памяти.

Таблица 1

Производительность алгоритмов криптосхемы

Значение λ	KeyGen	Encrypt	Умножение с приведением по модулю	Decrypt
8	19 мс	2 мс	6 мс	22 мс
12	830 мс	100 мс	25 мс	51 мс
16	50 с	4 с	96 мс	64 мс
24	2 мин	12 сек	619 мс	160 мс
32	6 мин	14 сек	2 сек	352 мс
48	16 мин	2 мин	8 сек	1.5 сек
64	56 мин	5 мин	1 мин	4 сек

Заключение. Была предложена компактная симметричная схема шифрования, основанная на матричных полиномах. Временные издержки данной схемы при гомоморфном вычислении составляют $\approx O(\lambda^{3.76})$, что немного превышает полученную в работе [12] оценку $\approx \tilde{O}(\lambda^{3.5})$. Однако, в отличие от криптосхемы из работы [12], действия с представленной матричной криптосистемой могут быть очень легко и эффективно распараллелены, поскольку основными действиями в ней являются сложение и умножение матриц. В частности, при наличии достаточного количества процессоров гомоморфное сложение шифртекстов может быть осуществлено за то же время, что и сложение открытых текстов.

Также были проанализированы возможные атаки на матричную криптосистему. Было показано, что криптоанализ по ключу перешифрования, по шифртекстам и по известным открытым текстам описанной матричной криптосистемы может быть сведен к решению систем полиномиальных уравнений. Надлежащий вы-

бор параметра λ может сделать эти системы сложными для решения классическим XL методом, а также методами линеаризации, вычисления базисов Грёбнера. В дальнейшем планируется более подробно исследовать получаемые при криптоанализе системы уравнений, в частности, возможность решения методом треугольной декомпозиции [23].

Также планируется разработать полноценную оптимизированную реализацию матричной криптосхемы (в том числе и параллельную версию). Ожидается, что предложенная криптосхема ПГШ значительно превзойдёт по производительности известные на сегодняшний день полностью гомоморфные криптосхемы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Rivest R.L., Adleman L., Dertouzos M. L.* On data banks and privacy homomorphisms // Foundations of secure computation. – 1978. – Vol. 4, № 11. – P. 169-180.
2. *Paillier P*. Public-key cryptosystems based on composite degree residuosity classes // Advances in cryptology-EUROCRYPT'99. – Springer Berlin Heidelberg, 1999. – P. 223-238.
3. *Rivest R. L., Shamir A., Adleman L.* A method for obtaining digital signatures and public-key cryptosystems // Communications of the ACM. – 1983. – Vol. 26, No. 1. – P. 96-99.
4. *Gentry C*. Fully homomorphic encryption using ideal lattices // STOC. – 2009. – Vol. 9. – P. 169-178.
5. *Vaikuntanathan V*. Computing blindfolded: New developments in fully homomorphic encryption // Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on IEEE. – 2011. – P. 5-16.
6. *Gentry C., Halevi S*. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits // Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on IEEE, 2011. – P. 107-109.
7. *Jing-Li H., Ming Y., Zhao-Li W*. Fully homomorphic encryption scheme extended to large message space // Instrumentation, Measurement, Computer, Communication and Control, International Conference on IEEE, 2011. – P. 533-536.
8. *Alperin-Sheriff J., Peikert C*. Practical bootstrapping in quasilinear time // Advances in Cryptology–CRYPTO 2013. – Springer Berlin Heidelberg, 2013. – P. 1-20.
9. *Alperin-Sheriff J., Peikert C*. Faster Bootstrapping with Polynomial Error // IACR Cryptology ePrint Archive. – 2014. – Vol. 2014. – P. 94.
10. *Orsini E., van de Pol J., Smart N.P*. Bootstrapping BGV Ciphertexts With A Wider Choice of p and q.
11. *Smart N.P., Vercauteren F*. Fully homomorphic SIMD operations // Designs, codes and cryptography. – 2014. – Vol. 71, No. 1. – P. 57-81.
12. *Stehlé D., Steinfeld R*. Faster fully homomorphic encryption // Advances in Cryptology-ASIACRYPT 2010. – Springer Berlin Heidelberg, 2010. – P. 377-394.
13. *Brakerski Z., Gentry C., Vaikuntanathan V*. (Leveled) fully homomorphic encryption without bootstrapping // Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. – ACM, 2012. – P. 309-325.
14. *Bogdanov A., Lee C. H*. Limits of provable security for homomorphic encryption // Advances in Cryptology–CRYPTO 2013. – Springer Berlin Heidelberg, 2013. – P. 111-128.
15. *Domingo-Ferrer J*. A Provably Secure Additive and Multiplicative Privacy Homomorphism // Information Security. – Springer Berlin Heidelberg, 2002. – P. 471-483.
16. *Hojsík M., Půlpánová V*. A fully homomorphic cryptosystem with approximate perfect secrecy // Topics in Cryptology–CT-RSA 2013. – Springer Berlin Heidelberg, 2013. – P. 375-388.
17. *Гельфанд С.И.* О числе решений квадратного уравнения // Издание осуществлено при поддержке РФФИ (издательский проект № 01-01-14022). – 2004. – С. 124.
18. *Кнут Д.* Искусство программирования. Т. 2. Получисленные алгоритмы. – СПб.: Вильямс, 2007. – 788 с.
19. *Williams V.V*. Multiplying matrices faster than Coppersmith-Winograd // Proceedings of the forty-fourth annual ACM symposium on Theory of computing. – ACM, 2012. – P. 887-898.

20. *Olsson R.A., Keen A.W.* Parallel Matrix Multiplication // The JR Programming Language: Concurrent Programming in an Extended Java. – 2004. – P. 211-225.
21. *Schaefer T.J.* The complexity of satisfiability problems // Proceedings of the tenth annual ACM symposium on Theory of computing. – ACM, 1978. – P. 216-226.
22. *Bard G.* Algebraic cryptanalysis. – Springer, 2009.
23. *Gao X. S., Huang Z.* Characteristic set algorithms for equation solving in finite fields // Journal of Symbolic Computation. – 2012. – Vol. 47, No. 6. – P. 655-679.

REFERENCES

1. *Rivest R.L., Adleman L., Dertouzos M. L.* On data banks and privacy homomorphisms, *Foundations of secure computation*, 1978, Vol. 4, № 11, pp. 169-180.
2. *Paillier P.* Public-key cryptosystems based on composite degree residuosity classes, *Advances in cryptology-EUROCRYPT'99*. Springer Berlin Heidelberg, 1999, pp. 223-238.
3. *Rivest R. L., Shamir A., Adleman L.* A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, 1983, Vol. 26, No. 1, pp. 96-99.
4. *Gentry C.* Fully homomorphic encryption using ideal lattices, *STOC*, 2009, Vol. 9, pp. 169-178.
5. *Vaikuntanathan V.* Computing blindfolded: New developments in fully homomorphic encryption, Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on IEEE, 2011, pp. 5-16.
6. *Gentry C., Halevi S.* Fully homomorphic encryption without squashing using depth-3 arithmetic circuits, *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on IEEE*, 2011, pp. 107-109.
7. *Jing-Li H., Ming Y., Zhao-Li W.* Fully homomorphic encryption scheme extended to large message space, *Instrumentation, Measurement, Computer, Communication and Control, International Conference on IEEE*, 2011. pp. 533-536.
8. *Alperin-Sheriff J., Peikert C.* Practical bootstrapping in quasilinear time, *Advances in Cryptology–CRYPTO 2013.* Springer Berlin Heidelberg, 2013, pp. 1-20.
9. *Alperin-Sheriff J., Peikert C.* Faster Bootstrapping with Polynomial Error, *IACR Cryptology ePrint Archive*, 2014, Vol. 2014, pp. 94.
10. *Orsini E., van de Pol J., Smart N.P.* Bootstrapping BGV Ciphertexts With A Wider Choice of p and q.
11. *Smart N.P., Vercauteren F.* Fully homomorphic SIMD operations, *Designs, codes and cryptography*, 2014, Vol. 71, No. 1, pp. 57-81.
12. *Stehlé D., Steinfeld R.* Faster fully homomorphic encryption, *Advances in Cryptology-ASIACRYPT 2010*. Springer Berlin Heidelberg, 2010, pp. 377-394.
13. *Brakerski Z., Gentry C., Vaikuntanathan V.* (Leveled) fully homomorphic encryption without bootstrapping, *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference.* ACM, 2012, pp. 309-325.
14. *Bogdanov A., Lee C. H.* Limits of provable security for homomorphic encryption, Advances in Cryptology–CRYPTO 2013. Springer Berlin Heidelberg, 2013, pp. 111-128.
15. *Domingo-Ferrer J.* A Provably Secure Additive and Multiplicative Privacy Homomorphism, *Information Security.* Springer Berlin Heidelberg, 2002, pp. 471-483.
16. *Hojsík M., Půlpánová V.* A fully homomorphic cryptosystem with approximate perfect secrecy, *Topics in Cryptology–CT-RSA 2013*. Springer Berlin Heidelberg, 2013, pp. 375-388.
17. *Gel'fand S.I.* O chisle resheniy kvadratnogo uravneniya [The number of solutions of a quadratic equation], Izdanie osushchestvleno pri podderzhke RFFI (izdatel'skiy proekt № 01-01-14022). 2004, pp. 124.
18. *Knut D.* Iskusstvo programmirovaniya [The art of computer programming]. T. 2. Poluchislennye algoritmy [Poluchyennyye algorithms]. St. Petersburg: Vil'yams, 2007, 788 p.
19. *Williams V.V.* Multiplying matrices faster than Coppersmith-Winograd, *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012, pp. 887-898.
20. *Olsson R.A., Keen A.W.* Parallel Matrix Multiplication, *The JR Programming Language: Concurrent Programming in an Extended Java*. 2004, pp. 211-225.

21. *Schaefer T.J.* The complexity of satisfiability problems, *Proceedings of the tenth annual ACM symposium on Theory of computing*. ACM, 1978, pp. 216-226.
22. *Bard G.* Algebraic cryptanalysis. Springer, 2009.
23. *Gao X.S., Huang Z.* Characteristic set algorithms for equation solving in finite fields, *Journal of Symbolic Computation*, 2012, Vol. 47, No. 6, pp. 655-679.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ромм.

Буртыка Филипп Борисович – Южный федеральный университет; e-mail: bbfilipp@ya.ru; 347928, г. Таганрог, ул. Чехова, 2, корпус "И"; тел.: +79081948371; кафедра безопасности информационных технологий; аспирант.

Burtyka Philipp Borisovich – Southern Federal University; e-mail: bbfilipp@ya.ru; Block “I”, 2, Chekhov street, Taganrog, 347928, Russia; phone: +79081948371; the department of information technologies security; postgraduate student.