Раздел I. Алгоритмы обработки информации

УДК 004.056.55 DOI 10.18522/2311-3103-2024-5-6-15

Л.К. Бабенко, В.С. Стародубцев

ОЦЕНКА ВРЕМЕНИ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ ШИФРОВАНИЯ, РАСШИФРОВАНИЯ, ГОМОМОРФНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ КРИПТОСИСТЕМЫ ДОМИНГО-ФЕРРЕРА

Рассматривается симметричная вероятностная гомоморфная криптосистема Доминго-Феррера, основанная на задаче факторизации чисел. В настоящее время актуальны гомоморфные криптосистемы двух типов: типа Джентри и основанные на задаче факторизации чисел. Отличительной особенностью последних по сравнению с криптосистемами типа Джентри является меньшая трудоёмкость выполнения гомоморфных операций, что значительно расширяет область их применения на практике. Однако, поскольку гомоморфные криптосистемы, основанные на задаче факторизации чисел, не получили широкого распространения и не были в достаточной мере проанализированы, в отличие от криптосистем типа Джентри, требуется их тщательное всестороннее исследование. Для рассматриваемой симметричной гомоморфной криптосистемы Доминго-Феррера приводятся описания операций генерации ключа, шифрования, расшифрования и выполнения гомоморфных вычислений. Для операций шифрования, расшифрования и выполнения гомоморфных вычислений приводится оценка сложности, выраженная в количестве базовых математических операций, а также графики, иллюстрирующие зависимости количества операций от выбранных параметров криптосистемы. Целью исследования является оценка сложности выполнения процессов шифрования, расшифрования и выполнения гомоморфных вычислений симметричной вероятностной гомоморфной криптосистемой Доминго-Феррера, основанной на задаче факторизации чисел. Основным результатом настоящей работы является оценка сложности и определение наиболее трудоёмких этапов шифрования, расшифрования и выполнения гомоморфных вычислений с помощью шифра Доминго-Феррера, подтвержденных рядом экспериментальных исследований. Проведенное исследование представляет собой важный шаг в развитии криптографической системы Доминго-Феррера, основанной на задаче факторизации чисел, имеет практическую значимость реализации алгоритмов с возможностью определения временных затрат шифрования, расшифрования и выполнения гомоморфных вычислений. Полученные результаты могут быть использованы исследователями и программистами при разработке реализаций криптосистемы Доминго-Феррера на языках программирования.

Информационная безопасность; конфиденциальная информация; гомоморфное шифрование; криптосистема Доминго-Феррера; криптоанализ; оценка сложности алгоритма шифрования.

L.K. Babenko, V.S. Starodubtsev

ESTIMATION OF THE EXECUTION TIME OF ENCRYPTION, DECRYPTION, AND HOMOMORPHIC CALCULATIONS USING THE DOMINGO-FERRER CRYPTOSYSTEM

This article considers a symmetric probabilistic homomorphic Domingo-Ferrer cryptosystem based on the problem of number factorization. Currently, homomorphic cryptosystems of two types are relevant: the Gentry type and those based on the problem of factorization of numbers. A distinctive feature of the latter, in comparison with Gentry-type cryptosystems, is the lower complexity of performing homomorphic operations, which significantly expands the scope of their application in practice. However, since homomorphic cryptosystems based on the number factorization problem have not been widely used and have not been sufficiently analyzed, unlike Gentry-type cryptosystems, their thorough comprehensive study is required. For the considered symmetric homomorphic Domingo-Ferrer cryptosystem, descriptions of

key generation, encryption, decryption, and homomorphic computing operations are given. For encryption, decryption, and homomorphic computing operations, a complexity estimate is given, expressed in the number of basic mathematical operations, as well as graphs illustrating the dependence of the number of operations on the selected parameters of the cryptosystem. The aim of the study is to assess the complexity of performing encryption, decryption and homomorphic calculations by a symmetric probabilistic homomorphic Domingo-Ferrer cryptosystem based on the number factorization problem. The main result of this work is an assessment of the complexity and determination of the most time-consuming stages of encryption, decryption and performing homomorphic calculations using the Domingo-Ferrer cipher, confirmed by a number of experimental studies. The conducted research represents an important step in the development of the Domingo-Ferrer cryptographic system based on the problem of factorization of numbers and has the practical significance of implementing algorithms with the ability to determine the time costs of encryption, decryption and performing homomorphic calculations. The results obtained can be used by researchers and programmers in the development of implementations of the Domingo-Ferrer cryptosystem in programming languages.

Information security; confidential information; homomorphic encryption; Domingo-Ferrer cryptosystem; cryptanalysis; evaluation of the complexity of the encryption algorithm.

Введение. Гомоморфное шифрование является одной из важных техник в области криптографии, которая предоставляет возможность выполнять операции с зашифрованными данными без необходимости расшифровывать их. Это особенно полезно в контексте облачных вычислений и обработки данных, где часто возникает необходимость в анализе и использовании конфиденциальной информации.

Гомоморфное шифрование – современная техника, позволяющая значительно расширить область применения криптографии для защиты информации. Главной отличительной чертой гомоморфного шифрования является возможность выполнять различные операции над данными в зашифрованном виде без необходимости их предварительного расшифрования. Данное свойство крайне важно при использовании технологий облачных вычислений, поскольку позволяет применять данную технологию для обработки конфиденциальной информации.

В настоящее время существуют два основных типа гомоморфных криптосистем: типа Джентри и основанные на задаче факторизации чисел. Криптосистемы типа Джентри [1–4] получили широкое распространение и были всесторонне исследованы, для данных криптосистем доказана их высокая криптографическая стойкость. Однако гомоморфные операции в криптосистемах типа Джентри имеют высокую вычислительную сложность, ввиду чего выполняются крайне медленно и значительно сужают область их применения на практике. Криптосистемы, основанные на задаче факторизации чисел, обладают меньшей трудоёмкостью гомоморфных операций по сравнению с криптосистемами типа Джентри и имеют больший потенциал для применения на практике, однако на данный момент не обрели популярности и не были в достаточной мере проанализированы, и, как следствие, не была проведена оценка трудоёмкости. Поэтому оценка времени выполнения гомоморфных вычислений с помощью шифров, основанных на задаче факторизации чисел, является актуальной задачей.

Описание криптосистемы Доминго-Феррера. Эта криптосистема была создана в 1996 году и поддерживает гомоморфное сложение, вычитание и умножение [5–7]. Она является симметричной, то есть для шифрования и расшифрования используется один и тот же ключ [8]. Количество последовательных гомоморфных операций в данной системе не ограничено, однако размеры итоговых шифртекстов увеличиваются; например, при умножении размер шифртекста возрастает экспоненциально [9, 10]. Для инициализации криптосистемы Доминго-Феррера используется следующий набор чисел:

1) p и q – большие простые числа;

2) $n = p \times q$ – труднофакторизуемое число;

3) d – степень полиномов представления шифртекстов.

Алгоритмы генерации ключа, шифрования и расшифрования криптосистемой Доминго-Феррера приведены на рис. 1.

Рис. 1. Описание операций шифра Доминго-Феррера

Ключ в криптосистеме Доминго-Феррера представляется двумя числами – r_p и r_q, которые случайно выбираются из мультипликативных групп с соответствующим модулем: формулы (1) и (2) соответственно.

$$r_p \xleftarrow{\$} Z_p^*, \tag{1}$$

где Z_p^* – мультипликативная группа по модулю p, $\xleftarrow{\$}$ – операция выбора случайного элемента.

$$r_q \xleftarrow{\$} Z_q^*, \tag{2}$$

где Z_q^* – мультипликативная группа по модулю q, $\xleftarrow{\$}$ – операция выбора случайного элемента.

Количество операций выбора случайного элемента при шифровании. Для выполнения шифрования в криптосистеме Доминго-Феррера необходимо убедиться, что блок шифруемого сообщения $m \in Z_n$. Затем генерируется ряд случайных значений $a_1, \ldots, a_d$.Числам $a_2, \ldots, a_{d-1}$ присваивается значение по формуле (3).

$$a_i \xleftarrow{\$} Z_n, \tag{3}$$

где i – индекс числа в диапазоне [2; d-1], Z_n – кольцо по модулю n, $\xleftarrow{\$}$ – операция выбора случайного элемента.

Числу a_d аналогичным образом присваивается случайное значение из $Z_n \setminus \{0\}$ (любое ненулевое значение из Z_n) по формуле (4).

$$a_d \xleftarrow{\$} Z_n \setminus \{0\}. \tag{4}$$

Таким образом, из представленных формул (3) и (4) видно, что количество операций выбора случайного элемента зависит от d. Для ряда $a_2 \ldots a_d$ необходимо выбрать $d - 1$ случайных элементов.

Количество операций сложения и вычитания при шифровании. Когда определен набор случайных чисел $a_2 \ldots a_d$, значение a_1вычисляется по формуле (5).

$$a_1 = m - \left(\sum_{i=2}^{d} a_i\right) mod\ n, \tag{5}$$

где m – блок открытого текста, n – модуль, определенный в параметрах схемы шифрования.

Формула (5) содержит сложение сгенерированных случайных чисел из набора $a_2 \ldots a_d$. Всего в формуле (5) присутствует $d - 1$ слагаемых, следовательно для их сложения необходимо выполнить $d - 2$ операций сложения. Также в формуле (5) используется одна операция вычитания полученной суммы $\sum_{i=2}^{d} a_i$ из значения блока открытого текста m.

Сформированный набор чисел $a_1 \dots a_d$ представляет собой закодированный открытый текст и представляется в виде полинома по формуле (6). На данном этапе никаких математических операций не выполняется.

$$a(x) = a_d x^d + \cdots + a_1 x. \quad (6)$$

Количество операций умножения при шифровании. Когда открытый текст закодирован, происходит его шифрование с помощью составного ключа (r_p, r_q). Полином $\pi(x)$ формируется путём шифрования $a(x)$ на первой части ключа r_p по формуле (7).

$$\pi(x) = \left(a_d \times r_p^d x^d + \cdots + a_1 \times r_p x\right) mod\ p. \quad (7)$$

Полином $\rho(x)$ формируется аналогичным образом – путём шифрования $a(x)$ на второй части ключа r_q по формуле (8).

$$\rho(x) = \left(a_d \times r_q^d x^d + \cdots + a_1 \times r_q x\right) mod\ q. \quad (8)$$

Сформированная пара $(\pi(x), \rho(x))$ – зашифрованное сообщение m.

Из формул (7) и (8) следует, что для формирования каждого из полиномов шифртекста $(\pi(x), \rho(x))$ происходит умножение каждого коэффициента полинома $a(x)$ на соответствующую часть ключа, возведенную в степень полинома, перед которой данный коэффициент установлен [11–14]. Следовательно, количество операций умножения при формировании полиномов шифртекста $(\pi(x), \rho(x))$ определяется по формуле (9).

$$MulCount = 2\sum_{i=1}^{d} i, \quad (9)$$

где d – степень полинома представления шифртекста.

На рис. 2 приводится график зависимости количества операций умножения при шифровании криптосистемой Доминго-Феррера от выбранного значения d (степени полинома представления шифртекстов).

Рис. 2. Зависимость количества операций умножения от степени полинома представления шифртекста при шифровании

Как видно из рис. 2, график имеет вид, близкий к функции $y = x^2$, что показывает, что сложность операции шифрования по умножению является квадратичной.

Оценка количества операций получения остатка от деления при шифровании. В шифре Доминго-Феррера операция получения остатка от деления при шифровании используется на заключительном этапе формирования полинома $a(x)$ – при вычислении числа a_1 (формула (3)), а также в формулах (7) и (8) на этапе вычисления значений шифртекста $(\pi(x), \rho(x))$. При формировании числа a_1 операции умножения не выполняются, поэтому достаточно только одной операции $mod\ n$. При вычислении полиномов шифртекста $(\pi(x), \rho(x))$ выполняются операции умножения, поэтому необходимо приведение по модулю (для полинома $\pi(x)$ по модулю p, для полинома $\rho(x)$ – по модулю q) после каждой операции умножения и их общее количество равно количеству операций умножения – $2\sum_{i=1}^{d} i$. Таким образом, при шифровании в криптосистеме Доминго-Феррера общее количество операций получения остатка от деления $ModCount$ вычисляется по формуле (10).

$$ModCount = 2\sum_{i=1}^{d} i + 1, \quad (10)$$

где d – степень полинома представления шифртекста.

Таким образом, исходя из результатов оценки сложности процесса шифрования криптографической системы Доминго-Феррера видно, что в данном процессе выполняются:

- $d - 1$ операций выбора случайного элемента;
- $d - 2$ операций сложения;
- 1 операция вычитания;
- $2\sum_{i=1}^{d} i$ умножений;
- $2\sum_{i=1}^{d} i + 1$ операций получения остатка от деления.

Для подтверждения полученных результатов разработана реализация шифра Доминго-Феррера на языке C++ и проведены экспериментальные исследования.

При выбранном значении степени полинома представления шифртекста $d = 2048$ оценочное число процессорных тактов составило 17037369, согласно [15]. На одном ядре процессора AMD Ryzen 5 3500U (2.1 ГГц) время шифрования одного блока – 34 мс.

Применение Китайской теоремы об остатках при расшифровании. В криптосистеме Доминго-Феррера на заключительном этапе расшифрования, в отличие от шифрования, происходит поиск открытого текста, как решение СЛАУ по модулям p и q с применением Китайской теоремы об остатках [16–18] по формуле (11).

$$m = CRT(\{M_p, M_q\}, \{p, q\}), \quad (11)$$

где CRT – функция, использующая Китайскую теорему об остатках для поиска значения открытого текста m.

Поиск решения системы линейных сравнений по модулю с применением доказательства Китайской теоремы об остатках [19] приведен в формуле (12).

$$x = \sum_{i=1}^{n} a_i \times M_i \times N_i \ mod\ M, \quad (12)$$

где a_i – целое число, $M_i = \frac{M}{p_i}$, $N_i = M_i^{-1}$, $M = p_1 \times p_2 \times \ldots \times p_n$.

Как видно из формулы (12), N_i определяется, как мультипликативное обратное M_i, которое вычисляется с помощью расширенного алгоритма Евклида [20]. Исходя из того, что на вход расширенного алгоритма Евклида подаются пары значений (p, q) и (q, p) количество выполняемых математических операций в процессе расшифрования не зависит от степени полинома представления шифртекста d, но зависит от выбранных параметров p и q. Поэтому сложность расширенного алгоритма Евклида рассматривается отдельно.

Оценка количества шагов расширенного алгоритма Евклида для поиска мультипликативного обратного. Описание расширенного алгоритма Евклида приведено на рис. 3. На вход алгоритма подается число a, для которого необходимо найти мультипликативное обратное в группе по модулю p.

$(x_1, x_2, x_3) = (1, 0, p)$;
$(y_1, y_2, y_3) = (0, 1, a)$;
Пока $y_3 \neq 0$ & $y_3 \neq 1$:
    Если $y_3 = 0$, то:
$a^{-1} \nexists$;
конец.
    Если $y_3 = 1$, то:
        $a^{-1} = y_2$;
        конец.
    $Q = \frac{x_3}{y_3}$;
    $(T_1, T_2, T_3) = (x_1 - Q \cdot y_1; x_2 - Q \cdot y_2; x_3 - Q \cdot y_3)$;
    $x = y; y = T$;

Рис. 3. Расширенный алгоритм Евклида

Для оценки количества шагов расширенного алгоритма Евклида на различных наборах данных, значения параметров p и q принимали значения простых чисел из диапазона (10; 10000). Результаты оценки приводятся на рис. 4. По оси абсцисс приведены значения числа q, по оси ординат – значения числа p. Чем светлее пиксель – тем больше шагов требуется для поиска мультипликативных обратных с помощью расширенного алгоритма Евклида.

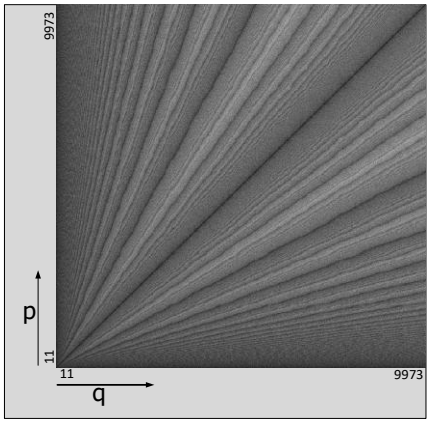

Рис. 4. Количество шагов расширенного алгоритма Евклида в зависимости от пары входных чисел (p, q)

В результате проведенного исследования минимальное число шагов расширенного алгоритма Евклида при расшифровании в криптосистеме Доминго-Феррера составило 2 (случаи, когда $p = q$), максимальное – 33 (например, при значениях $p = 3571, q = 9349$). В среднем криптосистеме Доминго-Феррера требовалось 14 шагов расширенного алгоритма Евклида для выполнения расшифрования. Как видно из рис. 3, на каждом шаге расширенного алгоритма Евклида [20] выполняется 1 деление, 3 вычитания и 3 умножения, то в среднем для выполнения расшифрования криптосистеме Доминго-Феррера требовалось 14 делений, 42 вычитания и 42 умножения.

Количество операций умножения при расшифровании. В шифре Доминго-Феррера на начальном этапе расшифрования $\left(\pi(x), \rho(x)\right)$ снимаются значения ключа (r_p, r_q) путём умножения на их мультипликативные обратные (r_p^{-1}, r_q^{-1}), возведенные в соответствующие степени $A_p(x)$ и $A_q(x)$ по формулам (13) и (14) соответственно.

$$A_p(x) = \left(b_d \times (r_p^{-1})^d x^d + \cdots + b_1 \times (r_p^{-1})\, x\right) mod\ p, \qquad (13)$$

где b – коэффициенты полинома $\pi(x)$ шифртекста.

$$A_q(x) = \left(b_d \times (r_q^{-1})^d x^d + \cdots + b_1 \times (r_q^{-1})\, x\right) mod\ q, \qquad (14)$$

где b – коэффициенты полинома $\rho(x)$ шифртекста.

Так как в симметричной криптосистеме Доминго-Феррера ключ не изменяется, для его составляющих $\left(r_p, r_q\right)$ нет необходимости при каждом расшифровании вычислять мультипликативные обратные, поэтому принимается, что мультипликативные обратные $\left(r_p^{-1}, r_q^{-1}\right)$ вычисляются в процессе генерации ключа и хранятся в памяти.

Из формул (13) и (14) следует, что для формирования каждого из полиномов $\left(A_p(x)\,, A_q(x)\right)$ происходит умножение каждого коэффициента полиномов шифртекста $\left(\pi(x), \rho(x)\right)$ на мультипликативное обратное соответствующей части ключа, возведенное в степень полинома, перед которой данный коэффициент установлен [11–14]. Следовательно, количество операций умножения при формировании полиномов $\left(A_p(x)\,, A_q(x)\right)$ равно количеству проводимых умножений в операции шифрования.

Однако, в шифре Доминго-Феррера в отличие от шифрования, в операции расшифрования умножение также применяется в расширенном алгоритме Евклида (для значений p и q до 10000 – в среднем 42 раза) и в Китайской теореме об остатках 5 раз.

Следовательно, общее количество умножений $MulCount$, необходимых криптосистеме Доминго-Феррера с параметрами p и q до 10000 для расшифрования шифртекста определяется по формуле (15).

$$MulCount = 2\sum_{i=1}^{d} i + 5 + 42, \quad (15)$$

где d – степень полинома представления шифртекста.

На рис. 5 приводится график зависимости количества операций умножения при расшифровании криптосистемой Доминго-Феррера от выбранного значения d (степени полинома представления шифртекстов).

Рис. 5. Зависимость количества операций умножения от степени полинома представления шифртекста при расшифровании

Как видно из рис. 5, график имеет вид, близкий к функции $y = x^2$, что показывает, что сложность операции расшифрования по умножению является квадратичной.

Количество операций сложения и вычитания при расшифровании. После снятия значений ключа (r_p, r_q) для полиномов $\left(A_p(x), A_q(x)\right)$ по формулам (16) и (17) вычисляются суммы M_p и M_p соответственно.

$$M_p = \sum_{i=1}^{d} a_i \ mod\ p, \quad 16)$$

где a_i – коэффициенты полинома $A_p(x)$.

$$M_q = \sum_{i=1}^{d} b_i \ mod\ q, \quad (17)$$

где b_i – коэффициенты полинома $A_q(x)$.

Формулы (16) и (17) содержат сложение коэффициентов полиномов $\left(A_p(x), A_q(x)\right)$. Всего в формулах (16) и (17) присутствует по d слагаемых, следовательно для их сложения необходимо выполнить $2(d-1)$ операций сложения.

Важно отметить, что в шифре Доминго-Феррера в отличие от шифрования, в операции расшифрования операции сложения и вычитания также применяются в расширенном алгоритме Евклида (для значений p и q до 10000 – в среднем 42 вычитания) и в Китайской теореме об остатках (2 сложения). Следовательно, в процессе расшифрования требуется выполнить $2(d-1)+2$ операций сложения и 42 операции вычитания.

Количество операций получения остатка от деления при расшифровании. Формулы (13) и (14) показывают, что для формирования полиномов $A_p(x)$ и $A_q(x)$, а также подсчёта сумм их коэффициентов M_p и M_q по формулам (16) и (17) соответственно требуются операции получения остатков от деления. Помимо этого, на заключительном

этапе расшифрования выполняется операция получения остатка от деления на n в Китайской теореме об остатках, что приводит к общему количеству получения остатка от деления $2\sum_{i=1}^{d} i + 3$.

Таким образом, исходя из результатов оценки сложности процесса расшифрования криптографической системы Доминго-Феррера видно, что в данном процессе выполняются:

- $2(d-1)+2$ операций сложения;
- 42 операции вычитания;
- $2\sum_{i=1}^{d} i + 5 + 42$ умножений;
- $2\sum_{i=1}^{d} i + 3$ операций получения остатка от деления.

Для подтверждения полученных результатов разработана реализация шифра Доминго-Феррера на языке C++ и проведены экспериментальные исследования.

При выбранном значении степени полинома представления шифртекста $d = 2048$ оценочное число процессорных тактов составило 17036373, согласно [15]. На одном ядре процессора AMD Ryzen 5 3500U (2.1 ГГц) время расшифрования одного блока – 32 мс.

После вычисления сумм M_p и M_q открытый текст исходного сообщения рассчитывается с помощью Китайской теоремы об остатках по формуле (18).

$$m = CRT\big(\{M_p, M_q\}, \{p, q\}\big), \tag{18}$$

где $CRT()$ – функция, использующая Китайскую теорему об остатках для поиска значения открытого текста m.

Оценка времени выполнения гомоморфных операций. Для выполнения операции (сложение, вычитание, умножение) над двумя шифртекстами в криптосистеме Доминго-Феррера нужно применить эту операцию к полиномам шифртекстов, как показано в формуле (19).

$$C_3 = \{(\pi(x)_1 \circ \pi(x)_2), (\rho(x)_1 \circ \rho(x)_2)\}, \tag{19}$$

где C_3 – результирующий шифртекст, $(\pi(x), \rho(x))_1$ – первый шифртекст, $(\pi(x), \rho(x))_2$ – второй шифртекст.

Важно, что выбранная математическая операция покоординатная, умножаются (складываются) коэффициенты полиномов только с одинаковой степенью.

Следовательно, выполнение гомоморфных операций в криптосистеме Доминго-Феррера обладает линейной сложностью $O(d)$ вне зависимости от того, какая именно гомоморфная операция выполняется.

Выводы. В данной работе проведена оценка сложности выполнения операций шифрования, расшифрования и гомоморфных вычислений симметричной вероятностной гомоморфной криптосистемой Доминго-Феррера, приведены формулы для расчёта количества математических операций в зависимости от выбранных параметров (степени полиномов представления шифртекста d, простых чисел p и q). Полученные оценки подтверждены экспериментальными исследованиями реализации шифра Доминго-Феррера на языке программирования C++. С использованием данной реализации время шифрования и расшифрования 1 блока размером 64 бита со значением модуля n до 10^8 и степенью полинома представления шифртекста $d = 2048$ на процессоре AMD Ryzen 5 3500U (2.1 ГГц) в однопоточном режиме оказалось примерно равным и составило 33 мс.

Таком образом, можно сделать вывод, что в криптографической системе Доминго-Феррера как при шифровании, так и при расшифровании наибольшее количество операций выполняется на этапах сопряжения с ключом, поскольку для каждого коэффициента полинома требуется возведение ключа в соответствующую степень. Это приводит к тому, что для шифрования или расшифрования требуется $2\sum_{i=1}^{d} i$ операций умножения, что является квадратичной сложностью алгоритма $O(d^2)$ и в контексте реализации на языках программирования и дальнейшего использования на практике требует оптимизации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Fan J., Vercauteren F.* Somewhat practical fully homomorphic encryption // Cryptology ePrint Archive. – 2012.
2. *Gentry C., Sahai A., Waters B.* Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based // Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. – Springer Berlin Heidelberg, 2013. – P. 75-92.
3. *Brakerski Z.* Fully homomorphic encryption without modulus switching from classical GapSVP // Annual cryptology conference. – Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. – P. 868-886.
4. *Brakerski Z., Gentry C., Vaikuntanathan V.* (Leveled) fully homomorphic encryption without bootstrapping // ACM Transactions on Computation Theory (TOCT). – 2014. – Vol. 6, No. 3. – P. 1-36.
5. *Domingo-Ferrer J.* A provably secure additive and multiplicative privacy homomorphism // International Conference on Information Security. – Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. – P. 471-483.
6. *Hariss K., Noura H., Samhat A. E.* An efficient fully homomorphic symmetric encryption algorithm // Multimedia Tools and Applications. – 2020. – Vol. 79, No. 17. – P. 12139-12164.
7. *Wang H., Wang Z., Domingo-Ferrer J.* Anonymous and secure aggregation scheme in fog-based public cloud computing // Future Generation Computer Systems. – 2018. – Vol. 78. – P. 712-719.
8. *Maqsood F. et al.* Cryptography: a comparative analysis for modern techniques // International Journal of Advanced Computer Science and Applications. – 2017. – Vol. 8, No. 6.
9. *Трепачева А. В.* Улучшенная атака по известным открытым текстам на гомоморфную криптосистему Доминго-Феррера // Тр. Института системного программирования РАН. – 2014. – Т. 26, № 5. – С. 83-98.
10. *Alabdulatif A., Kaosar M.* Privacy preserving cloud computation using Domingo-Ferrer scheme // Journal of King Saud University-Computer and Information Sciences. – 2016. – Vol. 28, No. 1. – P. 27-36.
11. *Merkel W. et al.* Factorization of numbers with physical systems // Fortschritte der Physik: Progress of Physics. – 2006. – Vol. 54, No. 8-10. – P. 856-865.
12. *Lenstra A. K. et al.* The factorization of the ninth Fermat number // Mathematics of Computation. – 1993. – Vol. 61, No. 203. – P. 319-349.
13. *Яковлев В.А., Шемякин С.Н., Таров Е.В.* Использование метода Монтгомери в алгоритме быстрого возведения в степень // I-methods. – 2023. – Vol. 15, No. 1. – P. 6.
14. *Hossain M.A. et al.* Performance analysis of different cryptography algorithms // International Journal of Advanced Research in Computer Science and Software Engineering. – 2016. – Vol. 6, No. 3.
15. *Agner F. Optimizing software in C++:* An optimization guide for Windows, Linux and Mac platforms. – 2020.
16. *Pei D., Salomaa A., Ding C.* Chinese remainder theorem: applications in computing, coding, cryptography. – World Scientific, 1996.
17. *Schindler W.* A timing attack against RSA with the chinese remainder theorem // Cryptographic Hardware and Embedded Systems—CHES 2000: Second International Workshop Worcester, MA, USA, August 17–18, 2000 Proceedings 2. – Springer Berlin Heidelberg, 2000. – P. 109-124.
18. *Wang W., Xia X. G.* A closed-form robust Chinese remainder theorem and its performance analysis // IEEE Transactions on Signal Processing. – 2010. – Vol. 58, No. 11. – P. 5655-5666.
19. *Iftene S.* General secret sharing based on the chinese remainder theorem with applications in e-voting // Electronic Notes in Theoretical Computer Science. – 2007. – Vol. 186. – P. 67-84.
20. *Iliev A., Kyurkchiev N.* The faster extended Euclidean algorithm // Collection of scientific works from conference. – 2018. – P. 21-26.

REFERENCES

1. *Fan J., Vercauteren F.* Somewhat practical fully homomorphic encryption, *Cryptology ePrint Archive,* 2012.
2. *Gentry C., Sahai A., Waters B.* Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based, *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*. Springer Berlin Heidelberg, 2013, pp. 75-92.
3. *Brakerski Z.* Fully homomorphic encryption without modulus switching from classical GapSVP, *Annual cryptology conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 868-886.
4. *Brakerski Z., Gentry C., Vaikuntanathan V.* (Leveled) fully homomorphic encryption without bootstrapping, *ACM Transactions on Computation Theory (TOCT),* 2014, Vol. 6, No. 3, pp. 1-36.

5. *Domingo-Ferrer J.* A provably secure additive and multiplicative privacy homomorphism, *International Conference on Information Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 471-483.
6. *Hariss K., Noura H., Samhat A. E.* An efficient fully homomorphic symmetric encryption algorithm, *Multimedia Tools and Applications,* 2020, Vol. 79, No. 17, pp. 12139-12164.
7. *Wang H., Wang Z., Domingo-Ferrer J.* Anonymous and secure aggregation scheme in fog-based public cloud computing, *Future Generation Computer Systems,* 2018, Vol. 78, pp. 712-719.
8. *Maqsood F. et al.* Cryptography: a comparative analysis for modern techniques, *International Journal of Advanced Computer Science and Applications,* 2017, Vol. 8, No. 6.
9. *Trepacheva A.V.* Uluchshennaya ataka po izvestnym otkrytym tekstam na gomomorfnuyu kriptosistemu Domingo-Ferrera [An Improved Known-Plaintext Attack on the Domingo-Ferrer Homomorphic Cryptosystem], *Tr. Instituta sistemnogo programmirovaniya RAN* [Proceedings of the Institute for System Programming of the Russian Academy of Sciences], 2014, Vol. 26, No. 5, pp. 83-98.
10. *Alabdulatif A., Kaosar M.* Privacy preserving cloud computation using Domingo-Ferrer scheme, *Journal of King Saud University-Computer and Information Sciences,* 2016, Vol. 28, No. 1, pp. 27-36.
11. *Merkel W. et al.* Factorization of numbers with physical systems, *Fortschritte der Physik: Progress of Physics,* 2006, Vol. 54, No. 8-10, pp. 856-865.
12. *Lenstra A. K. et al.* The factorization of the ninth Fermat number, *Mathematics of Computation,* 1993, Vol. 61, No. 203, pp. 319-349.
13. *Yakovlev V.A., Shemyakin S.N., Tarov E.V.* Ispol'zovanie metoda Montgomeri v algoritme bystrogo vozvedeniya v stepen' [Using Montgomery's method in a fast exponentiation algorithm], *I-methods,* 2023, Vol. 15, No. 1, pp. 6.
14. *Hossain M. A. et al.* Performance analysis of different cryptography algorithms, *International Journal of Advanced Research in Computer Science and Software Engineering,* 2016, Vol. 6, No. 3.
15. *Agner F.* Optimizing software in C++: An optimization guide for Windows, Linux and Mac platforms, 2020.
16. *Pei D., Salomaa A., Ding C.* Chinese remainder theorem: applications in computing, coding, cryptography. World Scientific, 1996.
17. *Schindler W.* A timing attack against RSA with the chinese remainder theorem, *Cryptographic Hardware and Embedded Systems—CHES 2000: Second International Workshop Worcester, MA, USA, August 17–18, 2000 Proceedings 2*. Springer Berlin Heidelberg, 2000, pp. 109-124.
18. *Wang W., Xia X. G.* A closed-form robust Chinese remainder theorem and its performance analysis, *IEEE Transactions on Signal Processing,* 2010, Vol. 58, No. 11, pp. 5655-5666.
19. *Iftene S.* General secret sharing based on the chinese remainder theorem with applications in e-voting, *Electronic Notes in Theoretical Computer Science,* 2007, Vol. 186, pp. 67-84.
20. *Iliev A., Kyurkchiev N.* The faster extended Euclidean algorithm, *Collection of scientific works from conference,* 2018, pp. 21-26.

Статью рекомендовал к опубликованию д.ф.-м.н., профессор В.О. Осипян.

Бабенко Людмила Климентьевна – Южный федеральный университет; e-mail: lkbabenko@sfedu.ru; тел.: +79054530191; г. Таганрог, Россия; кафедра безопасности информационных технологий им. Макаревича О.Б.; д.т.н.; профессор.

Стародубцев Виталий Сергеевич – e-mail: vstarodubcev@sfedu.ru; тел.: +79996928150; кафедра безопасности информационных технологий им. Макаревича О.Б.; аспирант.

Babenko Lyudmila Kliment'evna – Southern Federal University; e-mail: lkbabenko@sfedu.ru; phone: +79054530191; Taganrog, Russia; the Department of Information Technology Security named after Makarevich O.B.; dr of eng. sc.; professor.

Starodubcev Vitalij Sergeevich – e-mail: vstarodubcev@sfedu.ru; phone: +79996928150; the Department of Information Technology Security named after Makarevich O.B.; post graduate student.