Анализ корректирующих свойств кода системы остаточных классов для проектирования надежных облачных хранилищ

Н. И. Червяков[1], В. В. Бережной[2], М. А. Дерябин[3], Н. Н. Кучеров[4], Н. Н. Кучукова[5], И. В. Дворянинова[6]
Северо-Кавказский федеральный университет,
Кафедра прикладной математики и математического моделирования
[1]ncherviakov@ncfu.ru, [2]beregnoj@yandex.ru, [3]maderiabin@ncfu.ru, [4]nkucherov@ncfu.ru,
[5]nkucherov@ncfu.ru, [6]innadv99@mail.ru

***Аннотация*. Использование облачных технологий является экономически эффективным подходом к хранению и обработке данных. Однако вопросы обеспечения безопасности и надежности информации для такой модели информационных систем остаются актуальными в настоящее время. Один из перспективных и эффективных методов защиты данных с достижением необходимой надежности хранения информации предоставляет система остаточных классов (СОК). Использование СОК позволяет создавать надежные и безопасные мульти-облачные хранилища данных. Регулируя избыточность кода СОК, можно получать различные свойства корректирующего кода, сочетая эффективность реализации и надежность кодирования. В данной работе рассматриваются различные наборы модулей СОК и производится их оценка по различным показателям, позволяющим определить эффективность их применения для кодирования информации в мульти-облачных системах хранения и обработки данных.**

Ключевые слова: облачные хранилища; избыточная система остаточных классов; корректирущие способности кодов; мульти-облачные хранилища

I. Введение

Современные вычислительные средства и системы имеют тенденцию к увеличению параллелизма и распределённой структуре. Так появляются многопроцессорные системы и, одновременно с этим, широко используются распределенные вычисления на базе географически отдаленных вычислительных мощностей, объединенных в единый механизм. В качестве новой парадигмы хранения и обработки данных выступают мульти-облачные системы, которые позволяют объединить ресурсы сразу нескольких облачных провайдеров.

Технология облачных вычислений позволяет предоставлять услуги пользователям к различным ресурсам экономически эффективным способом. Несмотря на технические преимущества облачных вычислений, многие потенциальные пользователи не готовы их использовать и предоставлять доступ к личным данным ввиду потенциальных проблем конфиденциальности и надежности. Отсутствие безопасности при использовании такого метода предоставления услуг коллективного пользования ресурсами является одной из основных проблем облачных технологий.

В работе [1] обсуждается проблема сохранения конфиденциальности промежуточных наборов данных в облачных вычислениях. Авторы утверждают, что шифрование всех промежуточных наборов данных в облаке не является вычислительно и экономически эффективным, поскольку шифрование и дешифрование данных занимают много времени. Альянс Cloud Security Alliance (CSA) представил 10 наиболее важных проблем безопасности и конфиденциальности данных [2]. С целью решения указанных проблем в течение последнего десятилетия наблюдается повышение интереса исследователей к использованию системы остаточных классов (СОК) в облачных системах хранения и обработки больших информации [3], в которых, согласно CSA, обнаруживаются многочисленные преднамеренные и случайные угрозы безопасности [4]. Умышленные угрозы включают несанкционированный доступ к информации, перехват, фальсификацию, подделку, хакерские атаки и т.д. CSA заявляет, что в последние годы число несанкционированных доступов к информации, обрабатываемой и хранящейся в облаках, значительно увеличивается. Для снижения этого риска можно использовать криптографические протоколы и коды коррекции ошибок. Однако использование классических симметричных и асимметричных шифров требует большой вычислительной мощности и не применимо в некоторых случаях [5].

Случайные угрозы включают ошибки пользователя, небрежность, любопытство и т.д. В данном случае можно использовать систему защиты и управления информацией, основанную на проактивной концепции. Она включает одновременное использование взвешенной схемы разделения секрета на основе избыточной СОК (ИСОК), ключей шифрования и контрольных сумм для мониторинга полученных результатов. Альтернативным способом обеспечения конфиденциальности информации

Работа выполнена при финансовой поддержке РФФИ, проект № 18-07-00109, и при поддержке Гранта Президента Российской Федерации, проект МК-6294.2018.9

является использование гомоморфного шифрования на основе СОК [6,7]. Не менее важными факторами при обработке больших объемов данных являются производительность и масштабируемость. Распределенное хранилище может быть организовано на основе нескольких облачных хранилищ. Обычно в таких системах данные делятся на несколько частей, которые передаются различным облачным провайдерам для обеспечения доступности в случае сбоя. В таком случае должно обеспечиваться надежное пространство для хранения с мощным интерфейсом доступа для организации механизмов запроса и анализа [8]. Однако сбои распределенного хранилища могут вызывать несогласованность между разными копиями одних и тех же данных. Для решения данной проблемы могут использоваться большие базы данных. В этом случае для обеспечения высокой производительности обработка и анализ данных должны выполняться на основе принципов параллельных вычислений [9]. Чен и Хуан в работе [10] предложили модифицированную структуру MapReduce с использованием полного гомоморфного шифрования. Основными недостатками этой структуры являются высокая избыточность данных, вычислительная сложность алгоритмов шифрования данных и низкая надежность. Чтобы устранить эти недостатки, в [11] предложена надежная система облачного хранения на основе СОК, в которой операции могут выполняться в отдельно и одновременно, что упрощает и ускоряет вычисления с использованием облачных технологий. Избыточность остатков позволяет построить систему с обнаружением и коррекцией ошибок.

Однако, ни в одной из работ, посвященных использованию СОК в облачных технологиях, не проводится анализ использования конкретных величин модулей с точки зрения обеспечения надежности хранения данных и обеспечения восстановления информации при сбоях и отказах отдельных сервисов облачных хранилищ.

В рамках предлагаемой статьи проводится анализ систем модулей избыточной СОК для мульти-облачных хранилищ с целью обеспечения требуемых корректирующих способностей по исправлению ошибок в массивах данных.

II. Распределение данных в мультиоблачных хранилищах в СОК

Информационные последовательности, представляющие с точки зрения математической логики двоичные числа A, в СОК представляются неотрицательными остатками от деления на модули m_i, $i = 1,2,\dots,n$: $A = (a_1, a_2, \dots, a_n), a_i = A \bmod m_i = |A|_{m_i}$. Единственность представления числа в СОК в промежутке $[0, M-1]$, где $M = m_1 \cdot m_2 \cdot \dots \cdot m_n$ – рабочий диапазон представления чисел в СОК, обеспечивается при условии попарной взаимной простоты модулей m_i. Величина вычетов a_i по модулю m_i представляет собой целое число в интервале $[0, m_i - 1]$.

Возможности обнаружения и исправления ошибок в числах СОК проявляются при введении в нее контрольных (избыточных) модулей [12]. В качестве избыточных модулей берутся числа m_j, $j = n+1, n+2, \dots, n+r$, удовлетворяющие условиям взаимной простоты между собой и по всей совокупности исходных неизбыточных (рабочих) модулей. Число $R = M \cdot m_{n+1} \cdot \dots \cdot m_{n+r}$ будем называть полным диапазоном избыточной СОК (ИСОК).

Представление чисел в ИСОК осуществляется остатками по всем модулям $m_1, m_2, \dots, m_n, m_{n+1}, \dots, m_{n+r}$, $A = (a_1, a_2, \dots, a_n, a_{n+1}, \dots, a_{n+r})$. Каждый остаток ассоциируется с отдельным облачным провайдером и хранится или обрабатывается с использованием ресурсов этого провайдера. Диапазон представимых чисел не может превышать $M-1$, следовательно длина l кодируемой последовательности исходных данных определяется из выражения $l \le \log_2(M-1)$. Значение l определяет длину блока данных, на которые разбиваются передаваемые файлы.

Искажения в данных возникают при сбоях и неисправностях в процессе передачи, хранения и обработки их в облачных хранилищах. Так, по сведениям только из открытых источников, доступ к информации Amazon был ограничен в течение длительного времени из-за DDoS-атак в 2009 году. В 2013 году Amazon, Microsoft и Google объявили о серии сбоев в облаках. Технические сбои и потери данных из-за сбоев питания регулярно возникают у Amazon, Dropbox, Microsoft, Google и Yandex Disk. В первом квартале 2014 года Dropbox испытал перебои в обслуживании дважды. В 2013 году облачный провайдер Nirvanix был объявлен банкротом, что привело к прекращению работы всех его сервисов.

При хранении данных с использованием СОК можно восстанавливать искаженные данные отдельных хранилищ за счет избыточных данных по модулям $m_{n+1}, \dots, m_{n+r}$. Согласно [1] для гарантированного исправления ошибки по одному модулю необходимо наличие 2-х контрольных модулей m_{n+1}, m_{n+2} по абсолютной величине превышающих любой из рабочих модулей m_i. Для исправления двукратной ошибки – 4-х и т.п.

В следующем разделе представлен анализ систем оснований, удовлетворяющих условиям исправления ошибок.

III. Анализ систем оснований избыточной СОК для проектирования облачных хранилищ

Для кодирования данных с использованием СОК файлы и блоки данных разбиваются на последовательности, длина которых k бит не должна превышать рабочий диапазон [11]: $2^k \le M-1$. Ориентируясь на разрядность современных процессоров, диапазон представления чисел в СОК M может быть ориентирован на величины 2^{32} и 2^{64} с целью повышения производительности перевода в Сок и восстановления. При этом величина M должна как можно меньше превышать значения $2^{32} \approx 4{,}29 \cdot 10^9$ или $2^{64} \approx 1{,}84 \cdot 10^{19}$ для минимизации избыточности.

Таким образом, целесообразно разбивать данные для кодирования в СОК на последовательности длиной $l_1 = 32$ бит или $l_2 = 64$ бит. С учетом таких небольших длин информационной части последовательностей особые

требования должны предъявляться к их избыточной части и тому, какое количество искажений позволит исправить тот или иной набор контрольных модулей m_j. Исходя из того принципа, что данные по каждому модулю размещаются в отдельных облачных хранилищах, и из гипотезы, что выход из строя оборудования сразу нескольких облачных провайдеров маловероятен, следует учитывать однократные ошибки (по одному модулю), как наиболее вероятные, а также двукратные (по двум модулям).

Приведем далее наборы оснований ССОК для диапазона $2^{32} \approx 4{,}29 \cdot 10^9$. В качестве критериев подбора модулей СОК определим следующие:

1. Незначительное превышение рабочим диапазоном СОК M величины кодируемых данных 2^{32}.

2. Минимизация суммарной разрядности как рабочего M, так и полного диапазона R СОК.

3. Увеличение процентного соотношения разрядности корректируемых ошибок, к общей разрядности представления данных.

Следует учитывать, что суммарное количество модулей не должно превышать число доступных провайдеров облачных хранилищ. Результаты подбора модулей представлены в табл. 1 – для локализации и исправления однократной ошибки и в табл. 2 – для двукратной ошибки и на рис. 1–2.

ТАБЛИЦА I НАБОРЫ МОДУЛЕЙ СОК ДЛЯ 32-Х РАЗРЯНОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ И ИСПРАВЛЕНИЯ ОШИБКИ ПО ОДНОМУ ОСНОВАНИЮ

№	Рабочий диапазон M		Избыточный диапазон		Исправление, *бит*
	модули m_i	*бит*	модули m_j	*бит*	
	7, 11, 13, 17, 19, 23, 25,29	36	31, 37	11	6
	31, 37, 41, 43, 47, 53	35	59, 61	12	6
	79, 83, 89, 97, 101	35	103, 107	14	7
	255, 256, 257, 263	34	269, 271	18	9
	5, 991, 997, 1009	33	1013, 1019	20	10
	1621, 1627, 1637	33	1657, 1663	22	11
	63, 8191, 8353	33	8363, 8369	28	14
	65536, 65537	33	65539, 65543	34	17

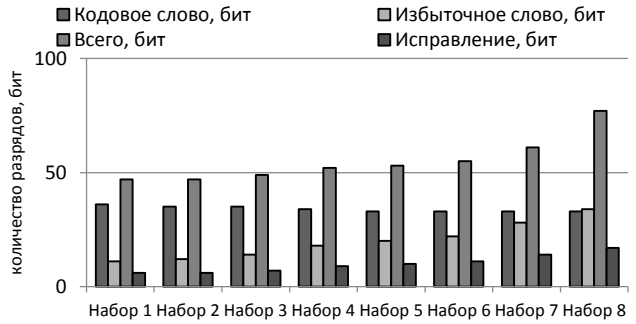

Рис. 1. Сравнительный анализ наборов модулей СОК для диапазона кодирования данных 2^{32} с исправлением ошибки по одному модулю

ТАБЛИЦА II НАБОРЫ МОДУЛЕЙ СОК ДЛЯ 32-Х РАЗРЯНОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ И ИСПРАВЛЕНИЯ ОШИБКИ ПО ДВУМ ОСНОВАНИЯМ

№	Рабочий диапазон M		Избыточный диапазон		Исправление, *бит*
	модули m_i	*бит*	модули m_j	*бит*	
	7, 11, 13, 17, 19, 23, 25,29	36	31, 37, 41, 43	23	12
	31, 37, 41, 43, 47, 53	35	55, 59, 61, 63	24	12
	79, 83, 89, 97, 101	35	103, 107, 109, 113	28	14
	255, 256, 257, 263	34	269, 271, 277,281	36	18
	5, 991, 997, 1009	33	1013, 1019, 1021, 1023	40	20
	1621, 1627, 1637	33	1657, 1663, 1667, 1669	44	22
	63, 8191, 8353	33	8363, 8369, 8377, 8387	56	28
8	65536, 65537	33	65539, 65543, 65551, 65557	68	34

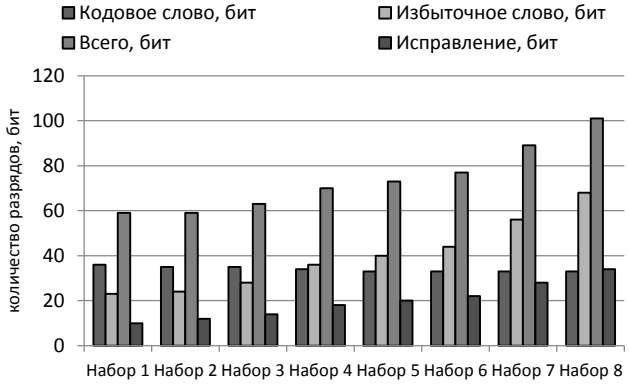

Рис. 2. Сравнительный анализ наборов модулей СОК для диапазона кодирования данных 2^{32} с исправлением ошибки по двум модулям

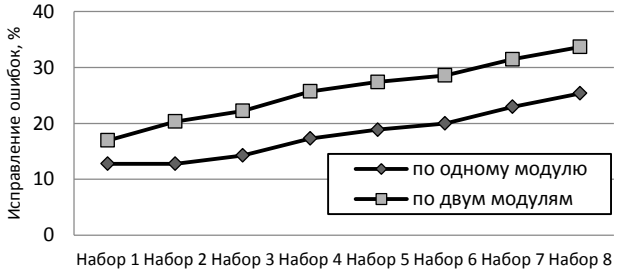

Рис. 3. Процент исправляемых ошибок от полного диапазона R для ошибок по одному и по двум модулям СОК

Проведем анализ наборов модулей на основе диаграмм рис. 1–3 для каждой таблицы, в которых представим следующие данные: разрядность представления рабочего диапазона СОК (кодовое слово, бит), разрядность представления избыточного диапазона (избыточное слово, бит), общая разрядность кодирования СОК (всего, бит), разрядность ошибочных данных, которые можно исправить с помощью соответствующего набора модулей (исправление, бит), отношение разрядности исправляемых

данных к общей разрядности кода СОК в процентах (процент обнаружения).

Из анализа таблиц и диаграмм следует, что при увеличении абсолютных значений рабочих модулей количество двоичных разрядов, необходимых для их представления уменьшается с 36 до 33. Величина 33 разряда является критической границей кодирования диапазона 2^{32}, которую уменьшить до 32 невозможно в условиях взаимной простоты значений величин модулей. Избыточные модули, в свою очередь, с увеличением их абсолютных значений требуют увеличения разрядности с экспоненциальной зависимостью. Наиболее прогрессивное увеличение разрядности наблюдается для наборов с четырьмя избыточными модулями. Так уже на 4 наборе модулей {255, 256, 257, 263} начинается превышение разрядности избыточного диапазона над рабочим. На 8 наборе это превышение становится более чем в 2 раза. Такое увеличение разрядности избыточного диапазона приводит к существенному увеличению длины всего кодового слова СОК. Так двоичное представление 1 набора требует 59 разрядов, а 8 набора – 101 разряд, что больше на 42%. В тоже время, несмотря на существенный рост разрядности, с увеличением значений модулей улучшаются корректирующие способности наборов. При обеспечении коррекции ошибки по одному модулю процент исправляемых данных увеличивается с 12,77% для набора 1, до 25,37% для набора 8. Для наборов модулей, обеспечивающих исправление ошибок по двум модулям эта же характеристика имеет значение 16,95% и 33,66% соответственно для этих наборов. Т.е., процент корректируемых данных имеет практически двукратное превышение, в то время, когда разрядность увеличивается только на 42%.

Следующим критерием оценки наборов модулей может служить удовлетворение их требованию отсутствия большого различия в величинах модулей. А именно, для обеспечения наилучшей криптографической стойкости схем разделения секрета, основанных на Китайской теореме об остатках [12] в облачных хранилищах рекомендуется использовать компактные последовательности взаимно простых чисел [13]. Последовательность взаимно простых чисел $m_0 < \cdots < m_n$, где $n \geq 1$, называется компактной последовательностью взаимно простых чисел, если $m_n < m_0 + m_0^{\theta}$ для некоторого действительного числа $\theta \in (0,1)$. При $\theta = 1$ верхняя граница последовательности равна удвоенному значению меньшего модуля. Из представленных 8-ми наборов модулей этому требованию удовлетворяют наборы с номерами 3, 4, 6, 8 в табл. 1 и 2. При обнаружении однократной ошибки можно добавить ещё набор 2 (табл. 1).

Чем меньше различие значений модулей в наборе (например, набор 8 является самым компактным), тем лучше криптографическая стойкость и данные поступающие в облачное хранилище распределяются более равномерно. В тоже время, различные провайдеры предлагают разные условия оплаты услуг по хранению и обработке данных, что может привести к некоторым финансовым издержкам. Для оптимизации этого случая следует использовать компактные последовательности с показателем θ, приближающимся к 1 (наборы 2 и 3).

IV. Выводы

Полученные результаты моделирования позволяют сделать следующие выводы: 1) для обеспечения высокой криптографической стойкости и наибольших корректирующих способностей в мульти-облачных хранилищах следует использовать наборы с максимально возможными значениями модулей СОК (наборы 8); 2) при учете экономической эффективности и ограниченности аппаратных ресурсов для хранения избыточных данных наилучшими являются наборы 2, 3; 3) наборы 1, 5, 7 не удовлетворяют требованиям криптографической стойкости и могут быть использованы для обеспечения надежности хранения в системах не использующих шифрование данных.

Список литературы

[1] Zhang X., Liu C., Nepal S., Pandey S., Chen J. A privacy leakage upper bound constraint-based approach for cost-effective privacy preserving of intermediate data sets in cloud. IEEE T. Parall. Distr., Vol. 24, 2013, pp. 1192–1202.

[2] Mora A. C., Chen Y., Fuchs A., Lane A., Lu R., Manadhata P. et al. Top ten big data security and privacy challenges. Cloud Security Alliance. https://www.isaca.org/Groups/Professional-English/big-data/GroupDocuments/Big_Data_Top_Ten_v1.pdf, 2012 (accessed 21.06.17)

[3] Chang C.H., Molahosseini A.S., Zarandi A.E., Tay T.F. Residue number systems: A new paradigm to datapath optimization for low-power and high-performance digital signal processing applications, IEEE Circ. Syst. Mag., Vol. 15, 2015, pp. 26-44.

[4] Hubbard D., Sutton M. Top threats to cloud computing v1. 0. Cloud Security Alliance. https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf, 2010 (accessed 21.06.17)

[5] Singh S., Sharma P.K., Moon S.Y., Park J.H. Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions, J. Amb. Intel. Hum. Comp., 2017, pp. 1-18.

[6] Cheon J.H., Kim J., Lee M.S., Yun A. CRT-based fully homomorphic encryption over the integers, Inform. Sciences, Vol. 310, 2015, pp. 149–162.

[7] Tchernykh A., Schwiegelsohn U., Talbi E., Babenko M. Towards Understanding Uncertainty in Cloud Computing with risks of Confidentiality, Integrity, and Availability, J. Comput. Sci.-Neth., 2016 (In press).

[8] Lynch C. Big data: How do your data grow? Nature, Vol. 455, 2008, pp. 28-29.

[9] Fernández A., S. del Río, López V., Bawakid A., M.J. del Jesus, Benítez J.M., Herrera F. Big Data with Cloud Computing: An insight on the computing environment, MapReduce, and programming frameworks. WIRES Data Min. Knowl., Vol. 4, No. 5, 2014, pp. 380–409.

[10] Chen X., Huang Q. The data protection of MapReduce using homomorphic encryption, in: Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, 2013, pp. 419–421.

[11] Celesti A., Fazio M., Villari M., Puliafito A. Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems, J. Netw. Comput., Vol. 59, 2016, pp. 208–218.

[12] Szabo N.S., Tanaka R.I. Residue arithmetic and its applications to computer technology. McGraw-Hill, 1967.

[13] Червяков Н.И., Дерябин М.А. Новый метод порогового разделения секрета, основанный на системе остаточных классов. Информационные технологии, том 22, №3, 2016, С.211-219.