УДК 004.021

Р.Ф. Жаринов

Исследование методов и средств решения задачи поиска вхождения символов в зашифрованные данные

Актуальность задачи поиска вхождений в зашифрованных данных обусловлена быстрым развитием рынка облачного хранения данных, тогда как на сегодняшний день не существует протоколов безопасной обработки данных на стороне облачного сервера, так как при обработке данных в зашифрованном виде либо необходимо передать серверу секретный ключ, либо каждый раз выкачивать всю базу данных. Решение задачи поиска вхождений над зашифрованными данными позволит безопасно хранить данные за пределами доверенной зоны. В настоящей статье рассмотрены общие методы безопасного поиска, а также приведен возможный инструментарий для создания протокола поиска вхождения в зашифрованных данных.
Ключевые слова: поиск в зашифрованных данных, гомоморфизм, метрики поиска вхождения.

В настоящее время среди компаний становится популярным строить информационную инфраструктуру с использованием облачных решений. Для малого и среднего бизнеса использование готовых аутсорсных решений по обработке данных позволяет уменьшить как штат сотрудников, обслуживающих информационные системы, так и сократить время внедрения и развертывания собственной инфраструктуры. Для крупных компаний использование облачных решений предоставляет возможность как горизонтального, так и вертикального масштабирования, в зависимости от изменяющихся условий рынка. Учитывая тот факт, что разглашение, потеря целостности и утечка данных могут разрушить основу для ведения бизнеса, они представляют большую ценность, но зачастую дублируются в открытом виде на многих внешних сервисах облачного хранения.

Компании, разрабатывающие инновационные решения или использующие конфиденциальную информацию, не могут позволить себе пользоваться преимуществами аутсорсных облачных решений ввиду возможности несанкционированного доступа со стороны сотрудников-администраторов внешних сервисов.

При выборе сервиса облачного хранения необходимо учитывать минимальный набор требований, предъявляемых к хранению и обработке информации, а именно:

- Авторизация – использование контроля доступа к собственным ресурсам.
- Безопасность на транспортном уровне – создание безопасного канала между пользовательским устройством и сервером.
- Схемы шифрования – позволяют хранить конфиденциальную информацию в виде, защищающем ее от несанкционированного доступа.
- Безопасный обмен файлами – возможность предоставления доступа к определенному множеству файлов сторонним пользователям, не являющимся клиентами данного сервиса.
- Дедубликация – возможность хранения уникальной информации на одной ноде облачного сервиса, исключая ее дублирование.

Существует достаточное количество сервисов, удовлетворяющих четырем из приведенных требований к хранению и обработке информации, но с использованием схем шифрования при хранении информации существует ряд проблем. В первую очередь это связано с обеспечением безопасной обработки данных в облачной структуре. Под безопасной обработкой понимается возможность производить операции над зашифрованными данными на стороне облачного сервера без передачи на его сторону дополнительной информации о хранимых данных, при этом также встает проблема комфортной работы с данными со стороны клиентов. Большинство предлагаемых решений для работы над зашифрованными данными значительно увеличивают время обработки запросов пользователя.

На сегодняшний день на рынке представлена только одна комплексная система обработки и хранения информации с использованием криптографических примитивов – система управления базой данных (СУБД) CryptDB [1]. Она позволяет производить конечный набор операций над зашиф-

рованными данными, без необходимости их расшифрования на стороне облачного сервера. К таким операциям относятся: сложение, объединение, сравнение, группировка, агрегирование данных, а также поиск по ключевым словам.

Как показывает практика, в современных автоматизированных системах неотъемлемой функцией является использование автодополнения, т.е. поиск вхождения по введенным данным в заданных полях базы данных (БД). В настоящее время не существует ни алгоритма, ни протокола, позволяющих производить операцию поиска вхождения в зашифрованных данных. Так, существует ряд решений для поиска ключевого слова, в случае идентичного совпадения введенного слова и слова в зашифрованном виде. Это не решает задачи успешного поиска, особенно в условии поиска в таких сложных языках, как русский, где у слов существует множество приставок, суффиксов и окончаний, что делает задачу корректной формулировки ключевого слова для поиска практически невыполнимой. Таким образом, настоящая статья будет посвящена поиску вхождений в зашифрованных данных.

Существующие решения. В 2000 г. была опубликована статья Сонга «Practical Techniques for Searches on Encrypted Data» [2], в которой авторы представили набор алгоритмов, позволяющих производить поиск в зашифрованных данных. Сложность поиска представленных алгоритмов была линейной $O(n)$ для каждого зашифрованного документа. Функции шифрования, поиска и расшифрования информации достаточно просты, поэтому будет описан лишь общий подход, используемый в алгоритме (рис. 1).

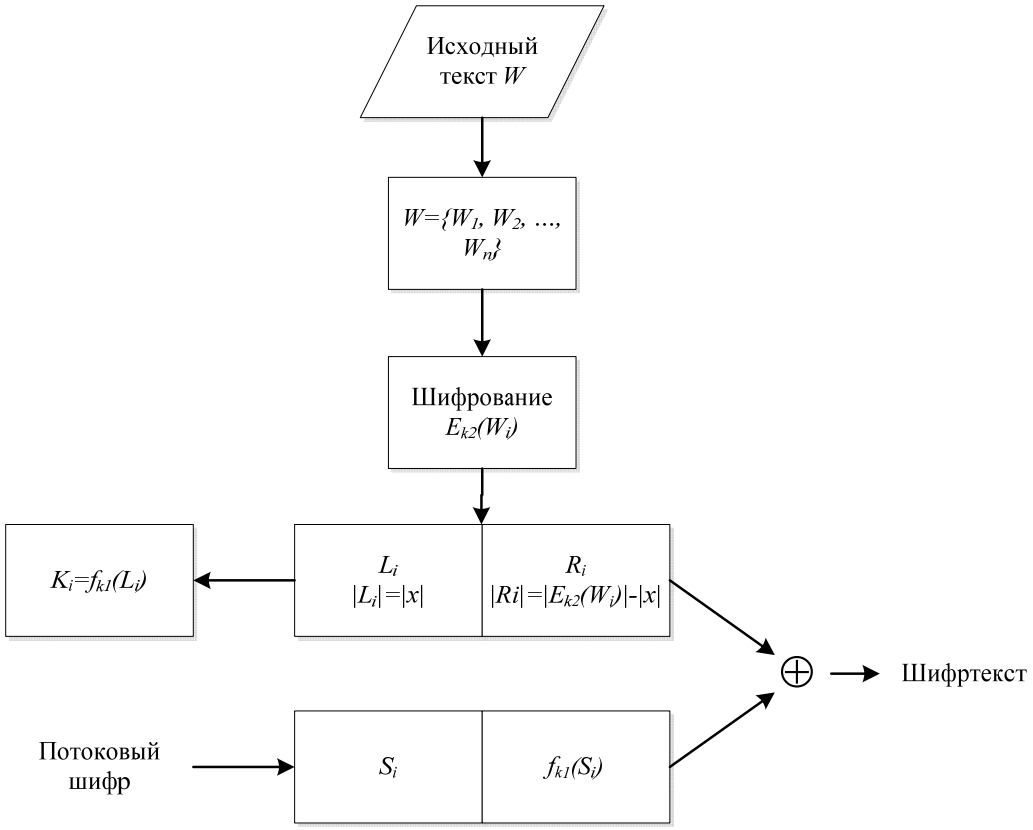

Рис. 1. Алгоритм шифрования в схеме Сонга

Алгоритм шифрования:

• На вход подается документ, который разбивается на слова по заранее заданному признаку (обычно выбирается разделение по расстоянию между словами), и удаляются все уникальные последовательности.

• Из одного мастер-ключа, предоставленного пользователем (MK), создается три подключа $KDF(MK)=<k_1,k_2,k_3>$, используемых в разных частях алгоритма и не позволяющих серверу расшифровать данные.

• Затем каждое слово шифруется стандартным блочным алгоритмом $E_{k_2}(W_i)$, разбивается на две неравные части $<L_i,R_i>$ (зависит от пользовательского параметра длины левой части зашифрованного сообщения x, $x<|W_i|$), дополнительно обрабатываются $S_i = G(k_3)$, $F_{k_i} = (S_i)$, где $k_i = f_{k_i}(L_i)$ и полученные значения складываются между собой (выполняется операция, исключающая или), образуя тем самым шифртекст $C =< L_i,R_i > \oplus < S_i,F_{k_i}(S_i) >$.

Алгоритм поиска:

• Для операции поиска необходимо зашифровать ключевое слово, согласно алгоритму шифрования и передать на сторону сервера $< E_{k_2}(W), f_{k_1}(L) >$.

• После этого сервер начинает обрабатывать каждую хранимую зашифрованную запись $C_i \oplus E_{k_2}(W_i)$, получая на выходе пару значений $< S_i, F_k(S_i) >$.

• И, так как значение длины исходного потокового шифрования известно, а именно x, из полученной пары мы сможем получить строку S_i.

• В итоге мы должны сравнить результат функции $F_{f_{k_i}(L)}(S_i)$ с оставшимися битами $F_k(S_i)$.

Вычислительная сложность приведенного алгоритма детерминированного поиска ключевого слова в зашифрованных данных экспоненциально растет с увеличением входных данных, что объясняет его неприменимость на практике. Подробного описания последующих разработанных схем поиска в зашифрованных данных приводить не будем, поскольку ни одна из них не находит вхождения в зашифрованных данных, а выполняет только поиск ключевых слов при полном совпадении. Ниже приведены некоторые особенности таких решений и их сравнительные характеристики (табл. 1, 2).

Таблица 1

Ключевые особенности существующих схем поиска в зашифрованных данных

Схема	Сложность поиска	Тип поиска	Требуется ли перевычисление предыдущих записей после вставки новых?
Song [2]	$O(n)$	Линейный	Нет
Goh [3]	$O(d)$	Использование предвычисления	Нет
Improved Index [4]	$O(1)$	Использование предвычисления	Да
PEKS [5]	$O(n)$	Линейный	Нет
Ranked [6]	$O(d)$	Использование предвычисления	Да

Таблица 2

Возможности выполнения различного вида поисков в схемах поиска в зашифрованных данных

Схема	Точное соответствие	Поиск вхождения	Регистронезависимый поиск	Регулярные выражения	Поиск стеммы
Song	Да	Нет	Нет	Нет	Нет
Goh	Да	Возможно	Возможно	Нет	Возможно
Improved Index	Да	Возможно	Возможно	Нет	Возможно
PEKS	Да	Возможно	Возможно	Нет	Возможно
Ranked	Нет	Нет	Да	Нет	Да

Впоследствии появилось много статей, которые, используя разные техники, уменьшали вычислительную сложность предложенных схем. Но в итоге задача поиска остановилась на возможности определения вхождения ключевого слова в зашифрованном множестве данных.

Какие же можно придумать решения для организации как минимум регистронезависимого поиска, а как максимум поиска вхождения?

Универсальное решение, конечно же, будет вводить избыточность, т.е. помимо шифрования основного теста, необходимо произвести его модифицирование (а именно привести к нижнему регистру) и так же зашифровать. Данный способ увеличит место хранения информации и увеличит скорость поиска необходимого ключевого слова как минимум в 2 раза. Исходя из данного заключения, можно подытожить, что метод поиска вхождения если и будет работать, то очень медленно (поэтому в табл. 2 используется слово «возможно»).

Возможный инструментарий для поиска вхождения в зашифрованных данных. Какие же существующие криптографические и математические протоколы можно взять за основу для протокола поиска вхождения? К сожалению, такого инструментария оказалось немного. Наиболее хорошо подходят: метрики вычисления расстояния между строками и использование скрытых вычислений. Рассмотрим подробнее каждый из перечисленных инструментариев.

Опишем основные метрики вычисления расстояния между строками как разной, так и одинаковой длины. Расстояние Хемминга [7] относится к метрике вычисления расстояния между множеством одинаковой длины и определяет количество бит (если рассматривать двоичные векторы) которые необходимо изменить, чтобы превратить одну строку в другую: $D_h(x,y)=\sum_{i=1}^{n} x\oplus y$.

Расстояние Левенштейна [7, 8] определяется как минимальное число требуемых операций преобразования (вставка, удаление и замена) для преобразования одной строки в другую. В качестве решения подобного рода задач используется алгоритм динамического программирования, который хранит в матрице количество операций изменения во всех возможных суффиксах и префиксах в обеих строках. Сложность вычисления данного алгоритма – $O(|x|\times|y|)$, а объем памяти для хранения матрицы – $O(\max|x||y|)$. Данное расстояние определяется рекуррентной формулой

$$D_i(x,y)=d(a,b),\ \text{где}\ a=|x|,b=|y|,$$

$$d(i,j)=\begin{cases} \begin{matrix} 0,\ \text{if}\ i=0,\ j=0,\\ i,\ \text{if}\ j=0,\ i>0,\\ j,\ \text{if}\ i=0,\ j>0, \end{matrix} \\ \min\begin{pmatrix} d(i,j-1)+1 \\ d(i-1,j)+1 \\ d(i-1,j-1)+\begin{cases}1,\ \text{if}\ x[i]=y[i]\\ 0,\ \text{if}\ x[i]\neq y[i]\end{cases}\end{pmatrix}, \ \text{if}\ i>0,\ j>0. \end{cases}$$

К недостаткам рассмотренного алгоритма можно отнести необходимость запоминания проделанных операций преобразования ввиду того, что результатом поиска вхождения является только операция «удаление».

Рассмотрим более сложную гибридную функцию вычисления расстояния Монг-Элкан (Monge-Elkan) [8], в которой используется рекурсивная схема сравнения для двух строк. Для начала строка x разбивается на подмножество $\{a_1,a_2,\ldots,a_{|x|}\}$, и строка y разбивается на $\{b_1,b_2,\ldots,b_{|y|}\}$, затем функция сравнения принимает вид $D_{me}(x,y)=\frac{1}{|x|}\sum_{i=1}^{|x|}\max_{j=1} D'(A_i,B_i)$, где D' – это некоторая второстепенная функция вычисления расстояния. Если в качестве второстепенной функции взять модифицированную функцию (при условии того, что для поиска вхождения нам необходима только операция удаления) расчета расстояния Левенштейна, получим итоговую формулу поиска вхождения, при $D_{mel}(x,y)=\begin{cases}1, & \text{если } x \text{ является вхождением в } y,\\ <1, & \text{если } x \text{ не является вхождением в } y:\end{cases}$

$$D_{mel}(x,y)=\frac{1}{|x|}\sum_{i=1}^{|x|}\max_{j=1}\begin{cases} \begin{matrix} 0,\ \text{if}\quad i=0,\ j=0;\\ 0,\ \text{if}\quad j=0, i>0;\\ 0,\ \text{if}\quad i=0,\ j>0; \end{matrix} \\ d(i-1,j-1)+\begin{cases}1,\ \text{if}\quad x[i]=y[i],\\ 0,\ \text{if}\quad x[i]\neq y[i],\end{cases}\ \text{if}\quad i>0,\ j>0. \end{cases}$$

В табл. 3 приведены результаты сравнения работы некоторых метрик при поиске вхождения слов «ell» и «all» в слове «hello».

Как видно из таблицы, модифицированный алгоритм Монг-Элкан выдает нужный нам результат, но с сокращением сложности алгоритма.

Что же касается использования критографических примитивов в применении скрытых вычислений, представляется возможным использование только двух способов: забывчивая передача (OT) [9] и гомоморфная криптосистема. Под OT-протоколом понимается тип передачи, в котором отправитель не запоминает, что было передано получателю и было ли передано вообще. В большинстве

случаев сложность такой передачи является полиномиальной. Использование гомоморфной криптосистемы с аддитивными свойствами позволяет взаимодействовать между отправителем и получателем более эффективно и с меньшей сложностью.

Таблица 3

Вычисления расстояний с использованием различных метрик

Метрика	Вычисленное расстояние Hello и ell	Вычисленное расстояние Hello и all
Levenshtein	2,0	3
MongeElkan	1,0	0,8(6)
SmithWaterman [7]	6,0	4,0
JaroWinkler [7]	0,8(6)	0,6(8)
Модифицированный MongeElkan	1,0	0,(6)

На сегодняшний момент криптосистема Пэйе [10] является одной из систем вероятностного шифрования, обладающая гомоморфным свойством аддитивности. Использование такой криптосистемы является наиболее перспективным с точки зрения решения задачи скрытых вычислений при поиске вхождения в зашифрованных данных.

Заключение. В настоящей статье приведен обзор существующих решений поиска в зашифрованных данных. На сегодняшний день не существует алгоритма, способного на практике продемонстрировать возможность поиска вхождения в зашифрованном тексте. В работе приведен обзор метрик, позволяющих вычислить минимальное расстояние для преобразования одной строки в другую, что необходимо для дальнейшей разработки двухстороннего протокола поиска вхождения в зашифрованных данных. В метрике было предложено использование гибридной функции подсчета расстояния, а в качестве второстепенной функции вычисления расстояния использовать модифицированное расстояние Левенштейна только с операцией удаления префиксов и суффиксов. В качестве основы безопасности дальнейшего протокола планируется использовать гомоморфную криптосистему Пэйе, которая поддерживает свойство аддитивности.

Литература

1. CryptDB: Protecting Confidentiality with Encrypted Query Processing / R.A. Popa, C.M.S. Redfield, N. Zeldovich, H. Balakrishnan // Proceedings of the 23-rd ACM Symposium on Operating Systems Principles. – Cascais, Portugal, 2011. – P. 85–100.

2. Song D.X. Practical Techniques for Searches on Encrypted Data / D.X. Song, D. Wagner, A. Perrig // Proceedings of IEEE Symposium on Security and Privacy, S&P 2000. – Berkeley, USA, 2000. – P. 44–55.

3. Secure indexes [Электронный ресурс]. – Режим доступа: http://crypto.stanford.edu/~eujin/papers/secureindex/secureindex.pdf, свободный (дата обращения: 20.05.2014).

4. Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions / R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky // Proceedings of the 13th ACM conference on Computer and communications security. – Alexandria, USA. – 2006. – P. 79–88.

5. Public Key Encryption with Keyword Search / D. Boneh, G.D. Crescenzo, R. Ostrovsky, G. Persiano // Proceedings of Eurocrypt 2004. – Interlaken, Switzerland, 2004. – P. 506–522.

6. Confidentiality-preserving rank-ordered search / A. Swaminathan, Y. Mao, G.M. Su et al. // Proceedings of the 2007 ACM workshop on Storage security and survivability. – N.Y., USA, 2007. – P. 7–12.

7. String Similarity Metrics for Information Integration [Электронный ресурс]. – Режим доступа: http://www.coli.uni-saarland.de/courses/LT1/2011/slides/stringmetrics.pdf, свободный (дата обращения: 25.04.2014).

8. Cohen W.W. A comparison of string distance metrics forname-matching tasks / W.W. Cohen, P. Ravikumar, S.E. Fienberg // Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web. – Acapulco, Mexico, 2003. – P. 73–78.

9. Naor M. Oblivious transfer with adaptive queries / M. Naor, B. Pinkas // Proceedings of 19-th Annual International Cryptology Conference. – Santa Barbara, California, USA, 1999. – P. 573–590.

10. Paillier P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes // Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques. – Prague, Czech Republic, 1999. – P. 223–238.

Жаринов Роман Феликсович
Аспирант каф. технологий защиты информации
Санкт-Петербургского государственного университета аэрокосмического приборостроения
Тел.: 8 (812) 494-70-77
Эл. почта: roman@vu.spb.ru

Zharinov R.F.
Research of methods and instruments for searching entry in encrypted data

Problem of searching entry in encrypted data is relevant whereas the rapid development of cloud storage market. While today there is no security protocol processing on the server side of the cloud. For processing data in encrypted form it's need to pass secret key to server or download full database each time. Solution of the problem of searching entry in encrypted data will allow to store data outside the trusted zone in secure form. This article reviews the general methods of secure search, as well as instruments to create a protocol of searching entry in encrypted data.
Keywords: searching entry in encrypted data, homomorphism, string methods distance.
