Михаил А. Куприяшин[1], Регина Ф. Баязитова[2]
[1,2]Национальный исследовательский ядерный университет «МИФИ»,
Каширское ш., 31, Москва, 115409, Россия
[1]e-mail: kmickle@yandex.ru, https://orcid.org/0000-0001-6887-4810
[2]e-mail: rfbayazitova@mail.ru, https://orcid.org/0000-0003-0012-9309

СОКРЫТИЕ ДАННЫХ В БЛОКАХ ORACLE DATABASE

DOI: http://dx.doi.org/10.26583/bit.2022.3.05

Аннотация. Данная статья посвящена рассмотрению возможности сокрытия сообщения в структуре системы управления базами данных (СУБД) Oracle Database. В качестве контейнера для хранения секретного сообщения используются блоки данных: физические структуры, выделяемые СУБД для хранения данных таблиц и других объектов. В работе рассматривается возможность заполнения таблиц таким образом, чтобы характер фрагментации свободного места являлся носителем информации о секретном сообщении. В частности, фрагменты свободного места соответствуют нулевым битам сообщения, а заполненные фрагменты – единичным. Для формирования контейнера необходимы привилегии на чтение, удаление и добавление новых записей в таблицу. Для извлечения сообщения достаточно права на чтение таблицы. В работе исследуется методика встраивания сообщения, обсуждаются особенности применения методики. Отмечается, что операции удаления и последующего добавления строк приводят к повреждению стеганоконтейнера. Рассмотрены базы данных типов OLAP и OLTP. Подчёркивается, что базы данных, основанные на модели OLAP, существенно лучше подходят для встраивания секретных сообщений описанного вида из-за небольшого числа вышеуказанных операций.
Ключевые слова: стеганография, сокрытие данных, базы данных, системы управления базами данных, Oracle Database.
Для цитирования: КУПРИЯШИН, Михаил А.; БАЯЗИТОВА, Регина Ф. СОКРЫТИЕ ДАННЫХ В БЛОКАХ ORACLE DATABASE. Безопасность информационных технологий, [S.l.], т. 29, № 3, с. 62–71, 2022. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1433. DOI: http://dx.doi.org/10.26583/bit.2022.3.05.

Mikhail A. Kupriyashin[1], Regina F. Baiazitova[2]
[1,2]National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),
Kashirskaya shosse, 31, Moscow, 115409, Russia
[1]e-mail: kmickle@yandex.ru, https://orcid.org/0000-0001-6887-4810
[2]e-mail: rfbayazitova@mail.ru, https://orcid.org/0000-0003-0012-9309

Hiding data in blocks of Oracle Database

DOI: http://dx.doi.org/10.26583/bit.2022.3.05

Abstract. In this paper, we explore the possibility of storing a hidden message in the physical data structures of the Oracle Database Management System (DBMS). We consider data blocks, the fundamental Oracle Database storage units for table data and other objects. As a result of normal usage, free space in data blocks gets fragmented. We devised a technique to form the desired fragmentation pattern in a data block. Assuming fragments of free space correspond to “0” bits, and fragments of data corresponding to “1” bits, we may store a secret message inside a block. In order to form the container, we need the object privileges to SELECT, DELETE and INSERT data. The SELECT object privilege is required to retrieve the message at a later time. The paper covers the technique of message embedding in detail. In conclusion we discuss various use causes for the technique. It should be noted that the operations of deleting and then adding rows lead to eventual damage to the steganographic container. In this regard, we studied the OLAP and OLTP database models. We conclude that OLAP databases are better suited for embedding secret messages due to the small number of the operations mentioned before.
Keywords: steganography, data hiding, databases, database management systems, Oracle Database.

For citation: KUPRIYASHIN, Mikhail A.; BAIAZITOVA, Regina F. Hiding data in blocks of Oracle Database. IT Security (Russia), [S.l.], v. 29, no. 3, p. 62–71, 2022. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1433. DOI: http://dx.doi.org/10.26583/bit.2022.3.05.

Введение

В настоящее время существует большое количество методов цифровой стеганографии, использующих различные типы контейнеров: от изображений до заголовков пакетов TCP/IP [1, 2].

При этом большое количество таких методов основывается на особенностях структуры самих контейнеров: наличии избыточности, малозаметных областей метаданных. Авторами было замечено, что такими свойствами обладают и файлы систем управления базами данных (СУБД). В качестве объекта исследования была выбрана СУБД Oracle Database, являющаяся наиболее популярной среди всех решений.

На основе анализа современных исследований в области цифровой стеганографии была найдена и разработана методика сокрытия сообщений в метаданных таблиц СУБД.

1. Особенности физической структуры данных

Архитектура СУБД Oracle Database имеет большое количество особенностей, представляющих потенциальную возможность для скрытого хранения данных. Одна из них основана на особенности фрагментации свободного пространства в этой СУБД.

Блок данных, являющийся наименьшей единицей хранения данных в архитектуре Oracle, состоит из заголовка и области для хранения данных. При добавлении данных в таблицу, содержимое строк последовательно добавляется в эту область. Как следствие, данные располагаются в нём в порядке своего появления. При удалении строки новые строки не будут занимать это место до достижения заполнения блока. Поэтому значения ROWID местоположения данных остаются без изменения (рис. 1).

Пользователи и администраторы СУБД могут определить физическое местоположение данных каждой строки, запросив при выборке значения псевдостолбца ROWID. Псевдостолбец ROWID существует во всех таблицах базы данных (БД), но значения ROWID для каждой строки не хранятся, а формируются на основе физического расположения данных этой строки в файлах и блоках данных [3].

Эти особенности можно использовать для встраивания в физическую структуру данных секретного сообщения. Если в последовательности возрастающих ROWID обнаруживаются пропуски, это свидетельствует об удалённых строках базы данных. Сопоставив удалённым строкам значение «0», а присутствующим – значение «1», можно закодировать секретное сообщение.

Предположим, что у нас есть пустой стеганоконтейнер, представляющий собой таблицу, и секретное сообщение длины N. Его можно заполнить следующим образом:

1. Удалить строки таблицы с помощью TRUNCATE или последовательных операций DELETE.
2. Последовательно рассмотреть биты секретного сообщения m1..mi..mn. В случае, если mi равен «1», в новую таблицу добавляется следующая строка из старой. В случае, если mi равен «0», в новую таблицу добавляются случайные данные.
3. Удалить из таблицы все строки, содержащие случайные данные.

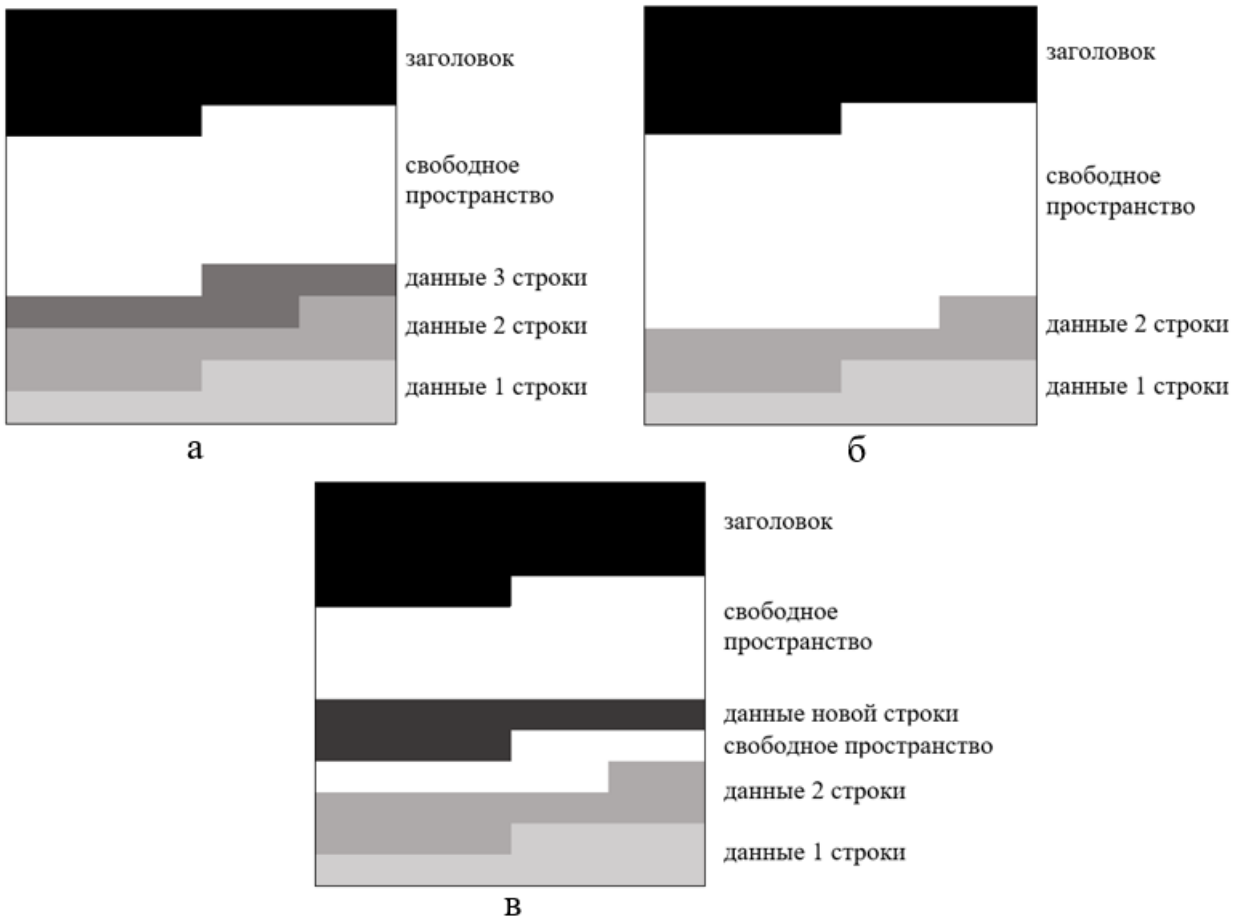

Рис. 1. Графическая иллюстрация заполнения блока строками: а – блок до удаления строки; б – блок после удаления строки; в – блок после добавления новой строки
Fig. 1. Graphic illustration of the process of filling a block with rows: a – the block before deleting; b – the block after deleting a row; c – the block after adding a new row

На шаге 2 всем строкам таблицы должны соответствовать последовательно возрастающие значения ROWID. После удаления данных на шаге 3, пропуски в последовательности ROWID будут соответствовать нулевым битам секретного сообщения. На рис. 2 представлена блок-схема встраивания. На шаге 1, если в таблице строки идут непрерывно в смысле последовательности их ROWID, то для встраивания достаточно вручную удалить каждую существующую строку с помощью DELETE. В противном случае следует очистить таблицу от всех строк с помощью команды TRUNCATE, которая помимо того, что удаляет все строки сразу, также освобождает всё пространство блоков, так, что новые строки вставляются с ROWID, начинающегося с первого в первом блоке объекта. В данном методе это необходимо во избежание встраивания каких-то строк в уже не используемое место в блоке.

На приведённой блок-схеме под удалением всех записей подразумевается использование TRUNCATE, а удалением каждой записи в отдельности — операции DELETE.

На рис. 3 приводится процесс встраивания сообщения в таблицу с данными.

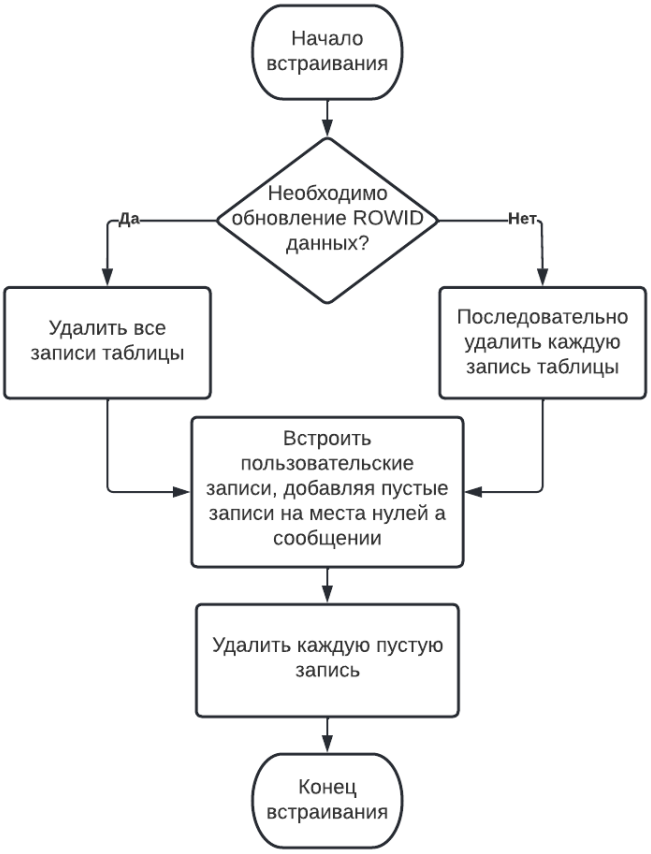

Рис. 2. Блок-схема для встраивания сообщения
Fig. 2. A block diagram for embedding a message

ROWID	SOME_USER_DATA
AAATZEAAMAAAACTAAA	Ufa
AAATZEAAMAAAACTAAB	Kazan
AAATZEAAMAAAACTAAC	Moscow
AAATZEAAMAAAACTAAD	Orel
AAATZEAAMAAAACTAAE	Izhevsk
AAATZEAAMAAAACTAAF	Suzdal

а

ROWID	SOME_USER_DATA
AAATZEAAMAAAACTAAA	Ufa
AAATZEAAMAAAACTAAD	Kazan
AAATZEAAMAAAACTAAF	Moscow
AAATZEAAMAAAACTAAG	Orel
AAATZEAAMAAAACTAAI	Izhevsk
AAATZEAAMAAAACTAAJ	Suzdal

в

ROWID	SOME_USER_DATA
AAATZEAAMAAAACTAAA	Ufa
AAATZEAAMAAAACTAAB	<empty>
AAATZEAAMAAAACTAAC	<empty>
AAATZEAAMAAAACTAAD	Kazan
AAATZEAAMAAAACTAAE	<empty>
AAATZEAAMAAAACTAAF	Moscow
AAATZEAAMAAAACTAAG	Orel
AAATZEAAMAAAACTAAH	<empty>
AAATZEAAMAAAACTAAI	Izhevsk
AAATZEAAMAAAACTAAJ	Suzdal

б

Рис. 3. Изменение вида таблицы и адресов строк в процессе встраивания сообщения: а – строки таблицы перед встраиванием; б – строки таблицы после удаления и встраивания вместе со строками для удаления; в – строки таблицы по окончании встраивания, после удаления соответствующих строк
Fig. 3. A change of the table and addresses of rows during the embedding of the message: a – rows of the table before embedding; b – rows of the table after deleting and adding them with rows for later deletion; c – rows of the table at the end of embedding after deletion of some rows

Для извлечения сообщения необходимо:

1. Выполнить выборку из заполненного стеганоконтейнера.
2. Определить нормальный шаг ROWID на основе размера каждой строки.
3. Просканировать последовательность ROWID по возрастанию. В каждой следующей строке, изменение ROWID на один шаг соответствует биту «1», а на n шагов – (n–1) последовательному биту «0» секретного сообщения.

В табл. 1 приводится описание необходимых привилегий для извлечения сообщения.

Таблица 1. Обязательные привилегии на этапе извлечения сообщения

SQL операция	Описание применения
SELECT	Операция получения содержимого таблиц. Для реализации метода на принимающей стороне требуется наличие этой привилегии на таблицу, выбранную в качестве контейнера.

2. Выбор потенциального контейнера

В рассматриваемой методике ключевыми параметрами для выбора подходящего контейнера являются размер таблицы и имеющиеся у пользователя привилегии на таблицу.

Размер таблицы ограничивает размер сообщения, которое можно в неё встроить. Каждая строка представляет бит данных. Количество строк таблицы должно соответствовать размеру сообщения следующим образом: их должно быть не меньше, чем вес Хэмминга встраиваемого сообщения.

После выбора таблицы с учётом фактора размера, следует изучить текущее положение строк таблицы. Если в таблице строки идут непрерывно в смысле последовательности их ROWID, то для встраивания достаточно вручную удалить каждую существующую строку с помощью DELETE, а затем перезаписать их с соответствующими «пустыми» строками в качестве нулей. Необходимые для встраивания в этом случае привилегии приведены в табл. 2.

Таблица 2. Обязательные привилегии для встраивания сообщения в первом случае

SQL-инструкция	Описание применения
DELETE	Используется во время подготовки таблицы к встраиванию, то есть очищению от строк, а затем при удалении строк, которые выступают в роли нулей во встраиваемом сообщении.
INSERT	Используется при обратном встраивании строк вместе со строками-нулями.
SELECT	Используется на этапе подготовки к встраиванию сообщения для получения текущих значений ROWID строк.

В противном случае следует очистить таблицу от всех строк с помощью команды TRUNCATE, которая помимо того, что удаляет все строки сразу, освобождает блок таким образом, что после этого новые строки вставляются с ROWID, начинающегося с первого в блоке (последний символ в значении ROWID будет снова «A»). В методике это необходимо во избежание встраивания каких-то строк в освобождённое посредством операций удаления место в блоке. Тогда список необходимых привилегий в первом случае пополнится привилегией DROP, которая необходима для выполнения TRUNCATE, согласно [3]. Полный список приведён в табл. 3.

Таблица 3. Обязательный список привилегий для встраивания во втором случае

SQL-инструкция	Описание применения
DROP	Привилегия на удаление одного объекта. Используется для подготовки таблицы к встраиванию сообщения.
INSERT	Используется при обратном встраивании строк вместе со строками-нулями.
DELETE	Используется для удаления строк-нулей при встраивании.
SELECT	Как и в предыдущем случае, необходимо для получения текущих значений ROWID строк при подготовке к встраиванию, а затем при удалении – для выяснения ROWID, по которым требуется его произвести.

3. Оценка пропускной способности канала

Важнейшим аспектом для обеспечения возможности эксплуатации разработанных методик является оценка их характеристик, на которых будут строиться рекомендации по их использованию, на основе которых пользователи смогут принять ряд необходимых решений по выбору контейнера и другим моментам перед непосредственным встраиванием и хранением сообщений. На данный момент существует множество различных характеристик, как общих, так и основывающихся на спецификах контейнеров в конкретных методах. Для осуществления оценки методики можно провести анализ пропускной способности и продолжительности жизни контейнера с сообщением.

Под пропускной способностью в стеганографии понимается максимальный размер секретного сообщения, которое пересылается в одном контейнере – такое описание приводится в [4]. Оценка вместительности встречается в большом количестве современных исследований. В [5] дано аналогичное определение коэффициента использования контейнера или коэффициента эффективности, который определяется как отношение размера наибольшего сообщения к размеру.

В стеганографии сложилось несколько общепринятых подходов к оценке пропускной способности [6]. В данной работе, по аналогии с [7], оценим пропускную способность как количество бит информации в строке таблицы.

На вход перед встраиванием подаются данные в двоичном представлении. При этом на этапе встраивания каждая строка соответствует одному двоичному символу, затем строки, кодирующие нули, удаляются, однако место, выделенное под них в блоке, остаётся. При оценке пропускной способности необходимо учитывать это. Тогда пропускная способность для методики – 1 бит/строка. Приведём пример. Пусть размер пространства, выделяемого в блоке под пользовательские данные, составляет 4096 байт, количество строк пользовательских данных в выбранной таблице равно 100. Средний размер строк таблицы равен 15 байтам. Тогда максимальное количество строк, помещающихся в эту таблицу, будет примерно равно 273. Соответственно, если встраиваемое сообщение будет не соответствовать хотя бы одному критерию: вес Хэмминга не более 100 и размер сообщения не более 273 бит, то встраивание станет невозможным.

После осуществления оценки пропускной способности канала в выбранной методике следует оценить объём пространства, доступного для встраивания, в конкретном выбранном контейнере.

Для методики этот объём будет зависеть, как отмечено выше, от размера блока, значения резервируемого места в блоке, которое будет использовано в случае изменений размеров строк, а также количества строк в таблице. Размер пространства в блоке, которое

выделяется для пользовательских строк, определяется значением PCTFREE. Это параметр, определяемый при создании таблицы, показывает, какой максимальный процент блока может быть заполнен до момента, когда он будет отмечен системой как занятый. Таким образом, для получения оценки необходимо вычислить средний размер строки выбранной таблицы, получить значение PCTFREE для неё и размер блока в табличном пространстве. Окончательное значение будет вычисляться как отношение произведения PCTFREE и размера к средней длине строк.

4. Условия повреждения контейнера

В течение жизненного цикла данные в таблице, как правило, подвергаются различного рода модификациям, которые могут привести к разрушению скрытого сообщения. Такими событиями могут быть следующие:

1. Удаление строк, участвующих в хранении сообщения.
2. Добавление строк в случаях, когда блок помечается системой, как свободный.
3. Удаление таблицы.
4. Перемещение таблицы.

В общем случае, невозможно оценить вероятность наступления вышеперечисленных событий, так как она существенно зависит от режима использования конкретной базы данных. Тем не менее, значительное число баз данных в этом отношении описывается одной из двух широко известных моделей: OLAP или OLTP. Обе основываются на реляционной теории, но используют отличающиеся технические подходы [8]

Первой будет рассмотрена модель OLAP. Хранилища данных преимущественно работают в режиме чтения; данные на запись поступают редко и большими порциями [8]. Ценность в таких СУБД представляет накопленная за достаточно большой промежуток времени историческая информация [9]. Записи, ранее помещённые в хранилище, редко подвергаются модификациям. Из этого следует вывод о том, что при использовании описываемой в этой работе методики сообщение не подвергнется разрушениям в течение длительного времени.

Базы данных OLTP служат источником информации в хранилищах данных. Данные в них обновляются и меняются регулярно, поэтому скорость обработки запросов является важнейшим показателем эффективности работы системы, для обеспечения которой служат транзакции. Они обрабатываются в течение нескольких миллисекунд и нет ничего, что требовало бы больше секунды [10].

Операция чтения, очевидно, не может повлиять на контейнер в разработанной методике, так как не меняют данные, содержащие само сообщение. Изменение строки влияет только на её содержимое, при этом положение строки не меняется (возможно, за исключением случаев так называемых переездов строки: в случае, когда она не помещается в установленные пределы на уровне блоков данных в ней помещается адрес её нового расположения, однако при этом сама строка не перемещается в таблице и продолжает занимать своё место в блоке, поэтому это не повлияет на значение ROWID и не стоит дальнейшего рассмотрения). Так как сообщение встраивается в метаданные таблицы, то эта операция не несёт угрозы его целостности. Наконец удаление строки по описанной выше логике приведёт к потере одного бита сообщения.

В связи с тем, что при встраивании сообщения предполагается использование разности адресов соседних строк для кодирования соответствующего количества нулей между единицами, в случае удаления строки пользователем, не участвующим в секретном общении, единица сообщения будет заменена нулём. Добавление легитимным пользователем строки в случае, если блок помечен системой как свободный, может привести к замене нуля сообщения на единицу.

5. Рекомендации по применению

Для применения методики первым шагом необходимо оценить БД на наличие подходящей таблицы для встраивания. Этот процесс представлен в виде блок-схемы на рис. 4. При наличии возможности выбора между различными БД в данной работе выбрана модель OLAP, в связи с описанными преимуществами.

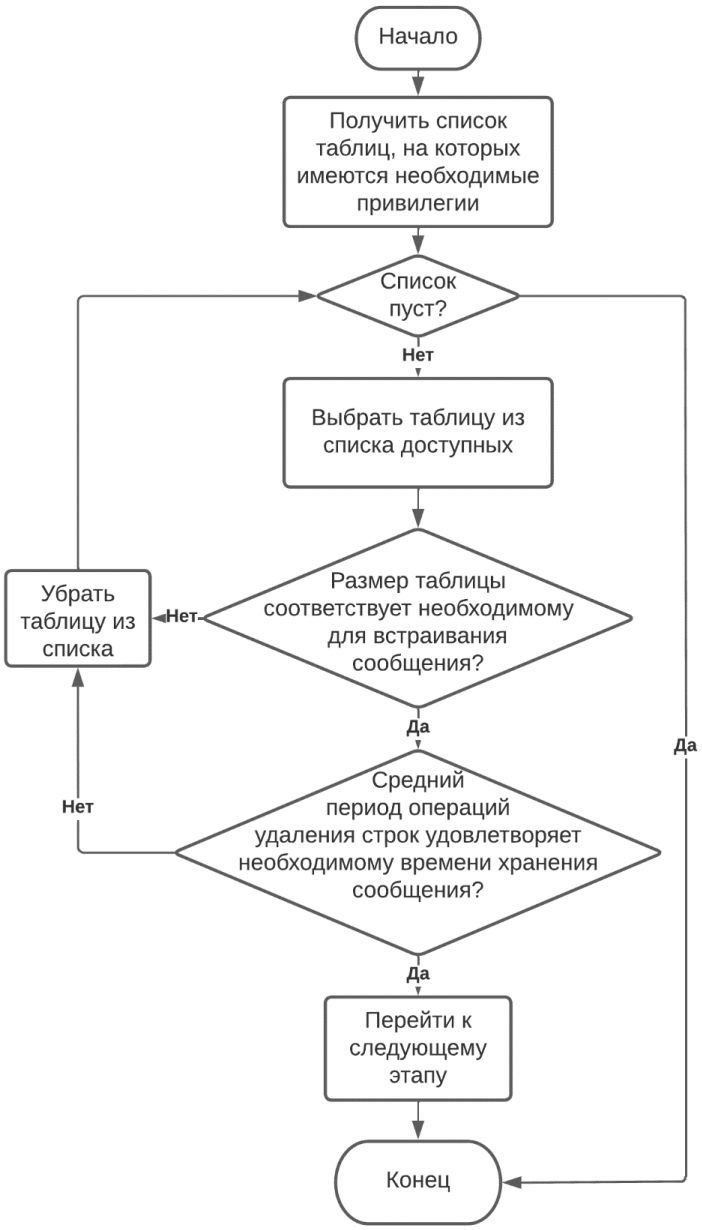

Рис. 4. Блок-схема поиска подходящего контейнера для реализации методики
Fig. 4. A block diagram of the search for eligible container for the technique

Второй этап подготовки к реализации стеганографического канала заключается в модификации сообщения в соответствии с потребностями в достижении максимальной практической устойчивости к внешним факторам. Это особенно важно в случае, если база данных работает в соответствии с моделью OLTP. Для снижения риска повреждения сообщения следует рассмотреть возможность применения корректирующих кодов, например, кодов Хэмминга. На данный момент существует большое множество различных вариантов. Порядок действий при этом представлен на рис. 5.

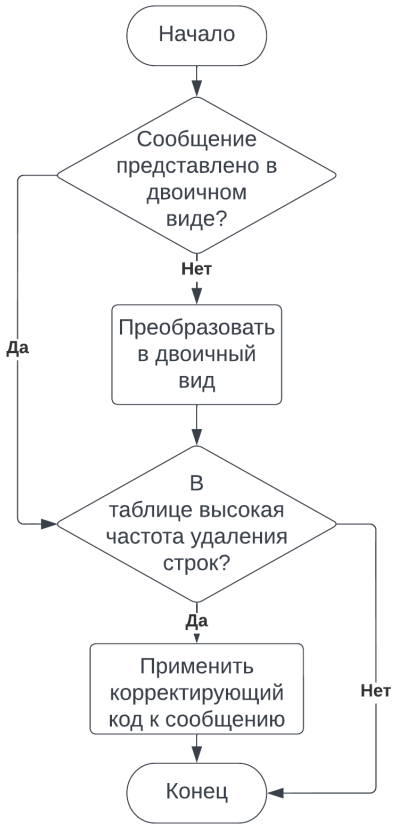

Рис. 5. Блок-схема второго этапа подготовки к встраиванию
Fig. 5. A block diagram of the second part of preparations for the embedding

Заключение

В настоящей работе предложена и исследована методика сокрытия сообщения в структуру СУБД Oracle Database на основе особенности фрагментации свободного места. При встраивании происходит добавление дополнительных строк к пользовательским, таким образом, что первые соответствуют нулям, а вторые – единицам сообщения.

Применение данной методики возможно при соблюдении определённых условий, связанных с привилегиями на выполнение необходимых операций, а также наличием достаточного места в таблице.

Для практического применения методики приведён анализ важных стеганографических характеристик – пропускная способность канала и продолжительность жизни заполненного контейнера. На основе полученных характеристик и описания необходимых предварительных условий предложены рекомендации по эффективной эксплуатации методики на практике.

СПИСОК ЛИТЕРАТУРЫ:

1. Dhobale D.D., Ghorpade V.R., Patil B.S., Patil S.B. Steganography by hiding data in TCP/IP headers. 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE). Chengdu: IEEE. 2010, p. V4-61-V4-65. DOI: http://dx.doi.org/10.1109/ICACTE.2010.5579643.
2. Sergeev A.V., Khorev P.B. Overview on hiding data in microsoft office documents. 3rd International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2021. M.: IEEE. 2021, p.1–4. DOI: http://dx.doi.org/10.1109/REEPE51337.2021.9388038.
3. Ashdown L., McCormack J., Kyte T. Oracle Database Database Concepts, 18 c. – Oracle Database Documentation. URL: https://docs.oracle.com/en/database/oracle/oracle-database/18/cncpt/ (дата обращения: 5.04.2022).

4. Варновский Н.П., Голубев Е.А., Логачев О.А. Современные направления стеганографии. Материалы конференции "Математика и безопасность информационных технологий" в МГУ 28-29 октября 2004 г. М.: МЦНМО. 2005, с. 32–64. URL: http://www.iisi.msu.ru/UserFiles/File/publications/mabit04.pdf (дата обращения: 30.08.22).
5. Аграновский А.В., Балакин А.В., Грибунин В.Г., Сапожников С.А. Стеганография, цифровые водяные знаки и стеганоанализ: Монография. М.: Вузовская книга, 2009. – 220 с. URL: https://www.elibrary.ru/item.asp?id=20312731 (дата обращения: 30.08.22) – EDN RCNCEV.
6. Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. Киев: МК-Пресс, 2006. – 288 с. URL: https://studfile.net/preview/7379018/ (дата обращения: 30.08.22).
7. Kadhim I.J., Premaratne P., Vial P.J., Halloran B. Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research. Neurocomputing. 2019, vol. 335, p. 299–326. DOI: http://dx.doi.org/10.1016/j.neucom.2018.06.075.
8. Plattner H. A common database approach for OLTP and OLAP using an in-memory column database. Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. 2009, p. 1–2. DOI: http://dx.doi.org/10.1145/1559845.1559846.
9. Inmon, W.H. Building the Data Warehouse, Fourth Edition. – 4th ed. м Indianapolis: Wiley Publishing, Inc., 2005. – 576 p. URL: https://ia800202.us.archive.org/9/items/2005BuildingTheDataWarehouse4thEditionWilliamH.Inmon/2005%20-%20Building%20The%20Data%20Warehouse%20%284th%20Edition%29%20%28William%20H.%20Inmon%29.pdf (дата обращения: 30.08.22).
10. Кайт Т. Oracle для профессионалов: архитектура и методики программирования. Т. Кайт, Д. Кун. 3-е изд. М.: И.Д. Вильямс, 2016. – 960 с.

REFERENCES:

[1] Dhobale D.D., Ghorpade V.R., Patil B.S., Patil S.B. Steganography by hiding data in TCP/IP headers. 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE). Chengdu: IEEE. 2010, p. V4-61-V4-65. DOI: http://dx.doi.org/10.1109/ICACTE.2010.5579643.
[2] Sergeev A.V., Khorev P.B. Overview on hiding data in microsoft office documents. 3rd International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2021. M.: IEEE. 2021, p.1–4. DOI: http://dx.doi.org/10.1109/REEPE51337.2021.9388038.
[3] Ashdown L., McCormack J., Kyte T. Oracle Database Database Concepts, 18c – Oracle Database Documentation. URL: https://docs.oracle.com/en/database/oracle/oracle-database/18/cncpt/ (accessed: 5.04.2022).
[4] Varnovskij N. P., Golubev E. A., Logachev O. A. Sovremennye napravleniya steganografii [Modern Modern directions of steganography]. Materialy konferencii "Matematika i bezopasnost' informacionnyx texnologij" v MGU 28-29 oktyabrya 2004 g. M.: MCNMO. 2005, s. 32–64. URL: http://www.iisi.msu.ru/UserFiles/File/publications/mabit04.pdf (accessed: 30.08.22) (in Russian).
[5] Agranovskij A.V., Balakin A.V., Gribunin V.G., Sapozhnikov S.A. Steganografiya, cifrovye vodyanye znaki i steganoanaliz: Monografiya [Steganography, digital watermarks and steganoanalysis: Monograph]. M.: Vuzovskaya kniga, 2009. – 220 s. URL: https://www.elibrary.ru/item.asp?id=20312731 (accessed: 30.08.22) – EDN RCNCEV (in Russian).
[6] Konaxovich G.F., Puzyrenko A.Yu. Komp'yuternaya steganografiya. Teoriya i praktika. [Computer steganography. Theory and practice]. Kiev: MK-Press, 2006. – 288 s. URL: https://studfile.net/preview/7379018/ (accessed: 30.08.22) (in Russian).
[7] Kadhim I.J., Premaratne P., Vial P.J., Halloran B. Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research. Neurocomputing. 2019, vol. 335, p. 299–326. DOI: http://dx.doi.org/10.1016/j.neucom.2018.06.075.
[8] Plattner, H. A common database approach for OLTP and OLAP using an in-memory column database / Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. 2009, p. 1–2. DOI: http://dx.doi.org/10.1145/1559845.1559846.
[9] Inmon, W.H. Building the Data Warehouse, Fourth Edition. – 4th ed. м Indianapolis: Wiley Publishing, Inc., 2005. – 576 p. URL: https://ia800202.us.archive.org/9/items/2005BuildingTheDataWarehouse4thEditionWilliamH.Inmon/2005%20-%20Building%20The%20Data%20Warehouse%20%284th%20Edition%29%20%28William%20H.%20Inmon%29.pdf (дата обращения: 30.08.22).
[10] Kajt T. Oracle dlya professionalov: arxitektura i metodiki programmirovaniya [Expert Oracle, Signature Edition Programming Techniques]. T. Kajt, D. Kun. 3-e izd. M.: I.D. Vil'yams, 2016. – 960 s. (in Russian).

Поступила в редакцию – 01 июля 2022 г. Окончательный вариант – 31 августа 2022 г.
Received – July 01, 2022. The final version – August 31, 2022.