УДК 004.04

Ю.В. Трифонова, Р.Ф. Жаринов

Возможности обезличивания персональных данных в системах, использующих реляционные базы данных

Рассматриваются вопросы обезличивания персональных данных, взгляд регулятора на требования к обезличиванию таких данных и проблемы, возникающие при применении методов обезличивания, предложенных Роскомнадзором. Вводится термин деперсонализации персональных данных. Предлагается кроссплатформенное решение для деперсонализации персональных данных в реляционных базах данных. Рассматриваются возможности использования инструментария CryptDB как надежного способа обезличивания персональных данных на стороне сервера.

Ключевые слова: обезличивание, деперсонализация, персональные данные, SQL, CryptDB.

Как только сложились требования к защите персональных данных в России, сразу же появились способы обхождения или уменьшения таких требований, и обезличивание персональных данных стало как раз одним из таких способов. Изначально обезличивание персональных данных позволяло самостоятельно оператору принимать решение о применяемых мерах и способах обеспечения безопасности персональных данных, после очередных изменений нормативных документов регуляторов обезличивание персональных данных стало значительно снижать требования к обработке таких данных, а как следствие – стоимость системы их защиты. До недавнего времени вопросы обезличивания постоянно обсуждались и являлись предметом ожесточенных споров, но в сентябре 2013 г. Роскомнадзор выпустил Приказ, который утвердил требования и методы по обезличиванию персональных данных, чем определил свою позицию в этом вопросе [1, 2]. Таким образом, Роскомнадзор предложил методику снижения обременений, позволяющую не применять в отношении обезличенных данных организационные и технические меры защиты, разработанные в свою очередь ФСТЭК и ФСБ. Хотя требования, предъявляемые сегодня к обработке персональных данных, позволяют достаточно гибко выбирать защитные мероприятия [3].

Обезличивание или деперсонализация. Итак, обезличивание персональных данных. Оказалось, что восприятие этого термина как представление персональных данных в виде, не позволяющем восстановить какую-либо информацию о субъекте персональных данных, является не совсем верным. Согласно ранее упомянутому Приказу Роскомнадзора обезличивание персональных данных должно обеспечивать не только защиту от несанкционированного использования, но и возможность их обработки, т.е. данные после обезличивания должны обладать рядом свойств, к которым относятся:

- Полнота – сохранение всей информации о конкретных субъектах или группах субъектов, которая имелась до обезличивания.
- Структурированность – сохранение структурных связей между обезличенными данными конкретного субъекта или группы субъектов, соответствующих связям, имеющимся до обезличивания.
- Релевантность – возможность обработки запросов по обработке персональных данных и получения ответов в одинаковой семантической форме.
- Семантическая целостность – сохранение семантики персональных данных при их обезличивании.
- Применимость – возможность решения задач обработки персональных данных, стоящих перед оператором, осуществляющим обезличивание персональных данных, обрабатываемых в информационных системах персональных данных, в том числе созданных и функционирующих в рамках реализации федеральных целевых программ (далее – оператор, операторы), без предварительного деобезличивания всего объема записей о субъектах.
- Анонимность – невозможность однозначной идентификации субъектов данных, полученных в результате обезличивания, без применения дополнительной информации.

Первым требованием к методу обезличивания Роскомнадзор поставил обратимость, т.е. возможность проведения деобезличивания. Таким образом, регулятор подтвердил возможность разбивания одной базы персональных данных на несколько с целью уменьшения требований к обработке части сведений, при этом используя возможность деобезличивания каждой конкретной записи для выполнения функций оператора. На наш взгляд, такой подход является неверным, поскольку данные обрабатываются в том же объеме у того же оператора. Если при этом снижаются требования к обработке, то возрастает вероятность атаки в момент деобезличивания персональных данных, и эта функция становится слабым звеном. Кроме того, уже неоднократно говорилось о том, что уникальность фамилии при определенных условиях является достаточным сведением для идентификации субъекта [4, 5], а по обезличенной базе данных при использовании простого метода перемешивания можно получить достаточно большое число сведений. Так, если в перемешанной базе данных будут храниться сведения о зарплате, то легко предположить, к кому относятся выбивающиеся из общего диапазона числа. Или же наличие известной фамилии в определенной базе данных может дать вам дополнительную информацию.

Однако часто возникают ситуации, когда необходимо полностью исключить обратимость. Из четырех предложенных регулятором методов обезличивания только один отвечает этому требованию – метод изменения состава или семантики, который предполагает обезличивание персональных данных путем замены их результатами статистической обработки, обобщения или удаления части сведений. Но в некоторых случаях такие изменения в базе данных абсолютно недопустимы, так, например, при разработке или доработке конкретной системы структура базы данных и ее наполнение играют важную роль, однако разработчиками являются сторонние работники или работники оператора, в чьи функциональные обязанности обработка персональных данных не входит. Кроме того, такой метод идеален при презентации системы посторонним людям: например, в рамках продажи или при прохождении проверки, он обеспечит возможность демонстрации системы, при этом исключив возможность нарушения конфиденциальности персональных данных, обрабатываемых в ней. Кто-то, возможно, скажет, что это можно сделать, заполнив базу данных случайными данными, однако для проверки всех функций базы данных необходим большой объем различных записей, отвечающих определенным требованиям, что невозможно легко создать и заполнить без использования исходных данных.

Таким образом, появляется задача обезличивания базы персональных данных, с исключением возможности получения каких-либо сведений о субъектах персональных данных по косвенным признакам, но с сохранением полноты, структурированности, релевантности и семантической целостности, условно назовем такой метод деперсонализацией.

Кроссплатформенное решение для деперсонализации в реляционных базах данных. Сегодня сложно представить автоматизированную обработку персональных данных без использования базы данных. Большинство баз данных строится с использованием SQL. Так, по данным профессионального сообщества Wikibon, на их долю приходится более 80% рынка (рис. 1) [6]. К наиболее популярным системам управления базами данных (СУБД) на языке SQL среди бесплатных относятся: MySQL, MariaDB, MongoDB и PostgreSQL (рис. 2) [7].

Рис. 1. Доля рынка реляционных баз данных на 2012 по данным Wikibon [6]

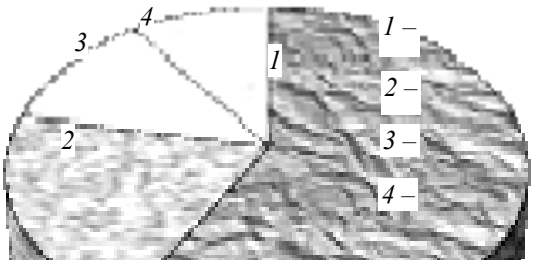

Рис. 2. Рынок реляционных баз дынных за 2013 г. по данным компании Infobox [7]

Выделяют два сегмента обезличивания персональных данных: статическое и динамическое. Статическое обезличивание применяется над копией промышленной версии базы данных и может беспрепятственно устанавливаться на внешние публичные носители информации без угрозы рас-

пространения реальных сведений о субъектах персональных данных. Такой подход мы и назвали деперсонализацией, поскольку он не предусматривает возможность обратного процесса – деобезличивания. Динамическое обезличивание работает в пределах защищенного периметра, используя дополнительные прокси-серверы, и работает путем перехвата и модификации ответов по заранее заданным алгоритмам. Объем передаваемых немодифицированных данных может зависеть от уровня полномочий запрашиваемого субъекта. В рамках настоящей работы подробно рассмотрим статистическое обезличивание, исходя из поставленной задачи.

Согласно магическому квадранту, приведенному в аналитическом отчете ведущей мировой исследовательской и консалтинговой компании Гартнер [8], на рынке присутствуют 3 лидирующих вендора: IBM с продуктом «InfoSphere Optim Data Privacy», Informatica с продуктом «Persistent Data Masking» и компания Oracle – «Data Masking Pack». Ввиду платности продуктов вышеуказанных вендоров оценить весь функционал не удалось. В табл. 1 приведено сравнение продуктов, использующих статическую деперсонализацию, по общедоступным параметрам.

Таблица 1

Сравнение продуктов, использующих статистическую деперсонализацию

Продукт / Характеристика	InfoSphere Optim Data Privacy [9]	Persistent Data Masking	Data Masking Pack
Предустановленные правила модификации данных	Да	Да	Да (алгоритм поиска отображения колонки к заданному правилу)
Возможность написание собственных правил-функций модификации данных	Да (C, C++, Lua, Assembler, VS COBOL II, PL/I, C)	Нет	Да (регулярные выражения)
Проверка целостности ссылок при модификации данных	Да	Да	Да
Возможность ускорения обработки при использовании кластеризации	Нет	Нет	Да (при использовании расширенной версии СУБД Oracle)

Как видно из отчета и сравнительной таблицы, продукты, способные провести как статическую, так и динамическую деперсонализацию на зарубежном рынке присутствуют в достаточном объеме. К сожалению, очень мало российских компаний из разряда малого бизнеса или государственных учреждений могут позволить себе столь дорогое решение. Поэтому большинство скриптов деперсонализации пишутся под себя без знания специфики, без анализа безопасности итогового решения (в качестве отрицательного примера можно привести скрипт использующий функцию реверсивности [10]) и как следствие на различных языках программирования.

Каким же образом можно унифицировать работу статической деперсонализации? Для начала необходимо выделить основные методы. К ним относятся:

- Перемешивание – перестановка значений в указанном множестве данных с удалением пиковой статистики данных.
- Обнуление/замыливание данных – возможность генерирования или установка одинаковых значений в поля с шаблонными данными (номер паспорта, пенсионного страхования и т.д.).
- Изменение семантики – удаление, замена или изменение части сведений какими-либо обобщенными значениями.
- Изменение итогового объема данных – увеличение объема модифицированной базы при помощи генерации или копирования данных либо удаление части зависимой информации.

В качестве универсального языка программирования будем использовать PL/SQL скрипты, предоставляющие мощный инструмент для обработки данных на сервере СУБД. В итоге алгоритм работы создания деперсонализированной базы данных будет выглядеть следующим образом:

- Для исходной базы данных необходимо настроить master-slave репликацию, тем самым при работе скрипта мы снимем нагрузку с основного сервера.
- Перед запуском скриптов необходимо остановить репликацию на вторичном сервере и произвести дублирование базы данных, тем самым актуализируя объем и сами данные БД.
- Завершающим шагом является запуск скриптов деперсонализации.

К основным недостаткам такого решения можно отнести необходимость адаптации хранимых процедур к БД организации. Связано это с отсутствием интуитивно понятного графического интерфейса, позволяющего соотносить поля с необходимыми методами деперсонализации данных.

Автоматизированная система обезличивания данных. Если подходить к вопросу обезличивания персональных данных по методике Роскомнадзора, т.е. сохраняя возможность обработки персональных данных в полном объеме, то можно предложить воспользоваться функционалом СУБД CryptDB, которая способна эффективно обслуживать запросы к реляционной базе данных – поиск, сортировка, математические функции и др. – без расшифровки записей. Таким образом, на стороне сервера база данных хранится в зашифрованном виде, что согласно Приказу регулятора [1] можно назвать обезличенным видом, поскольку это защищает данные от несанкционированного доступа и обеспечивает возможность их обработки. Обезличенные персональные данные за счет средств CryptDB позволяют сохранить такие свойства, как полнота, структурированность, релевантность, применимость и обратимость (описание свойств приведено выше), поскольку персональные данные в полном объеме могут обрабатываться легальным пользователем, а без дополнительной информации (секретного ключа) такая база данных будет представлять собой набор символов. Свойство семантической целостности в зашифрованной базе данных выполняться не будет, что полностью исключает возможность косвенного получения информации. Оценкой свойств такого способа обезличивания являются:

- Обратимость – позволяет провести процедуру деобезличивания.
- Вариативность – позволяет перейти от одной таблицы соответствия к другой без проведения процедуры деобезличивания.
- Изменяемость – позволяет вносить изменения в массив обезличенных персональных данных без предварительного деобезличивания.
- Стойкость – данный способ обезличивания является стойким к атакам на идентификацию субъекта персональных данных. Проведение такой атаки будет возможно только в случае раскрытия секретного ключа.
- Совместимость – возможно интегрирование записей, соответствующих отдельным атрибутам.
- Возможность косвенного деобезличивания – по зашифрованной базе данных невозможно провести косвенное деобезличивание персональных данных с использованием информации других операторов.
- Параметрический объем – объем зашифрованной базы данных в 4 раза больше объема исходной базы данных.
- Возможность оценки качества данных – проведение анализа качества обезличенных данных возможно.

Рассмотрим более подробно принципы работы CryptDB (схема работы с CryptDB представлена на рис. 3). Прокси-сервер хранит у себя мастер-ключ и схему базы данных. Сторонний сервер хранит у себя зашифрованную базу данных, хранимые процедуры и функции для работы CryptDB, а также служебные таблицы. Запросы с данными шифруются только от прокси-сервера и обратно. А все пользовательские запросы и ответы передаются в незашифрованном виде, поэтому прокси-сервер должен находиться в доверенной зоне [11–13].

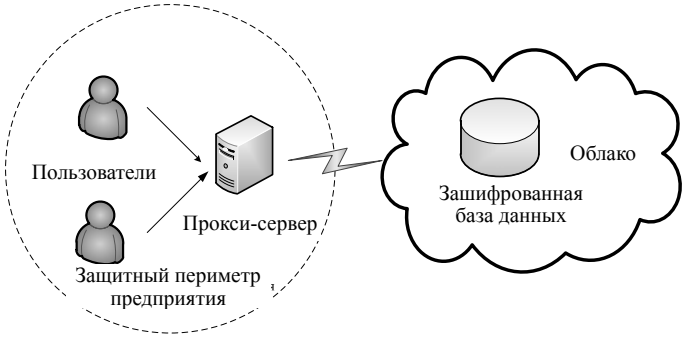

Рис. 3. Общая схема работы CryptDB

Из вышепредставленных криптографических алгоритмов детального рассмотрения заслуживает функция объединения данных разных таблиц по ключевым полям без необходимости их повторного шифрования. Итак, рассмотрим работу данного алгоритма на примере собственного приложения (табл. 2).

Таблица 2

Алгоритмы шифрования, функции и операции, используемые CryptDB

Схема	Алгоритм	Функция	Операции
RND	AES in CBC	нет	
DET	AES in CMC	Equality	=, !=, IN, COUNT, GROUP BY
OPE	OPSE [14]	Order	>, >=, <, <=, SORT, MAX, MIN
HOM	Paillier cryptosystem [15]	+,*	SUM
SEARCH	Song et al. [16]	Поиск по словам	LIKE
JOIN	ECC [17]	Join	LEFT JOIN, INNER JOIN, RIGHT JOIN, JOIN

Для реализации данного алгоритма необходимо реализовать следующие функции:

1. Генерация ключей.

а. Расчет параметров эллиптической кривой (генерация ключей, выбор кривой и фиксированной точки над нею).

б. Генерация секретных ключей для обращения к столбцу Sk_{col}, Sk_{msg}.

в. При генерации ключей используется алгоритм генерации псевдослучайного числа $PRP_{key}(\arg)$.

2. Шифрование сообщения m в виде точки C_i при обращении к столбцу i.

а. Рассчитываются секретные ключи для столбца: $csk_i = PRP_{Sk_{col}}(i)$, $csk_j = PRP_{Sk_{col}}(j)$.

б. Точка C_i вычисляется как $C_i = G \cdot csk_i \cdot PRP_{Sk_{msg}}(m)$, где G – фиксированная точка.

3. Вычисление токена $t_{i\to j}$ для операции обращения к столбцам выполняется следующим образом:

а. Используя значения из 2, *а* наш токен примет вид $t_{i\to j} = \frac{csk_j}{csk_i} \bmod n$, где n – это порядок эллиптической кривой.

4. Как результат мы можем продолжать выполнять операции шифрования на другой таблице без необходимости перешифрования:

а. Вычислим точку $C_{new} = C_i \cdot t_{i\to j}$.

Чтобы проверить правильность расчетов, необходимо доказать правильность работы алгоритма. Проверим, является ли точка C_{new} зашифрованной точкой другой таблицы:

$$C_i \cdot t_{i\to j} = C_i \cdot \frac{csk_j}{csk_i} \bmod n = G \cdot csk_i \cdot PRP_{Sk_{msg}}(m) \cdot \frac{csk_j}{csk_i} \bmod n = G \cdot PRP_{Sk_{col}}(j) \cdot PRP_{Sk_{msg}}(m).$$

Согласно алгоритму (п. 2, *а*), данное вычисление является зашифрованным обращением к столбцу j, из чего следует, что при использовании токена $t_{i\to j}$ можно обращаться к другой таблице зашифрованной на одном ключе Sk_{col}.

Дополнительно стоит отметить, что CryptDB шифрует все данные различными алгоритмами для обеспечения возможности выполнения операций над ними без расшифрования. Каждый столбец строки в зашифрованной базе данных представлен с 4-кратной избыточностью. Например, для хранения значения столбца ID, в зашифрованной базе данных хранится 4 столбца значений: вектор инициализации (IV) и 3 слоя (det, hom, ope) (рис. 4).

ID	NAME
1	Sasha

C1-IV	C1-Eq	C1-Ord	C1-Add	C2-IV	C2-Eq	C2-Ord	C2-Add
X27c3	X2b82	Xcb94	Xc2e4	X8a13	Xd1e3	X7eb1	X29b0

Рис. 4. Пример хранения исходной базы данных в зашифрованном виде CryptDB

Разработчики приложений, работающих с CryptDB, имеют возможность указывать для конкретных столбцов минимальный слой, в котором тот может находиться, таким образом, данные этого столбца не смогут находиться в слое, менее защищенном, чем установленный разработчиком ми-

нимальный слой. Если специфика работы с конкретными данными известна заранее, то для уменьшения времени обработки запросов можно удалить те слои, в которых нет необходимости.

Заключение. В настоящей статье рассмотрены различные подходы к обезличиванию персональных данных. Приведено кроссплатформенное решение для деперсонализации в различных реляционных базах данных, что позволит избежать необходимости написания скрипта для каждой СУБД и ошибок при их самостоятельном написании.

СУБД CryptDB рассмотрена с точки зрения надежного инструмента обезличивания персональных данных, иллюстрация свойств и характеристик такого обезличивания в рамках методических рекомендаций Роскомнадзора показывает возможность такого применения. Конечно, CryptDB является достаточно новой технологией, которая имеет ряд недостатков, в основном связанных со скоростью работы.

Какое-то время развитие CryptDB остановилось последняя версия проекта вышла в 2011 г., однако в 2013 году вышла статья об использовании описываемой СУБД [18]. Исходя из быстрого развития сервисов, предоставляющих облачное хранение данных, такой подход к обработке данных может стать лучшим решением для компаний малого бизнеса за счет низкой стоимости и универсальности такого решения.

Литература

1. Приказ Роскомнадзора от 05.09.2013 № 996 «Об утверждении требований и методов по обезличиванию персональных данных» (вместе с «Требованиями и методами по обезличиванию персональных данных, обрабатываемых в информационных системах персональных данных, в том числе созданных и функционирующих в рамках реализации федеральных целевых программ») [Электронный ресурс]. – Режим доступа: http://base.consultant.ru/cons/cgi/online.cgi?req=doc;base=LAW;n=151882, свободный (дата обращения: 21.05.2014).

2. Методические рекомендации по применению приказа Роскомнадзора от 5 сентября 2013 г. № 996 «Об утверждении требований и методов по обезличиванию персональных данных» (утв. Роскомнадзором 13.12.2013) [Электронный ресурс]. – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_157082/, свободный (дата обращения: 21.05.2014).

3. Лукацкий А.В. Роскомнадзор выпускает неплохую методику по обезличиванию персональных данных // Бизнес без опасности [Электронный ресурс]. – Режим доступа: http://lukatsky.blogspot.ru/2013/12/blog-post_19.html, свободный (дата обращения: 21.04.2014).

4. Петрыкина Н.И. Правовое регулирование оборота персональных данных. Теория и практика. – М.: Статут, 2011. – 134 с.

5. Является ли ФИО персональными данными в контексте ФЗ-152? // Форум информационной безопасности [Электронный ресурс]. – Режим доступа: http://www.itsecurity.groteck.ru/forum.php?sub=6788&from=0&format=printer-friendly, свободный (дата обращения: 21.04.2014).

6. Big Data Database Revenue and Market Forecast 2012–2017 / D. Floyer, J. Kelly, D. Vellante, S. Miniman // Professional community Wikibon [Электронный ресурс]. – Режим доступа: http://wikibon.org/wiki/v/Big_Data_Database_Revenue_and_Market_Forecast_2012–2017, свободный (дата обращения: 21.04.2014).

7. Какие стеки технологий используют чаще на платформе Jelastic? // Блог компании Infobox на Хабрахабр [Электронный ресурс]. – Режим доступа: http://habrahabr.ru/company/infobox/blog/209792/, свободный (дата обращения: 21.04.2014).

8. Magic Quadrant for Data Masking Technology / Gartner. – 2013 [Электронный ресурс]. – Режим доступа: https://www.gartner.com/doc/2636081, свободный (дата обращения: 21.04.2014).

9. Compare IBM data masking solutions: InfoSphere Optim and DataStage [Электронный ресурс]. – Режим доступа: http://www.ibm.com/developerworks/data/library/techarticle/dm-1211maskingsolution/dm-1211maskingsolution-pdf.pdf, свободный (дата обращения: 21.04.2014).

10. On REVERSing comma-separated set of words [Электронный ресурс]. – Режим доступа: http://vbegun.blogspot.ru/2008/01/on-reversing-coma-separated-set-of.html, свободный (дата обращения: 21.04.2014).

11. CryptDB : HOWTO Compile on Ubuntu Linux 12.04 [Электронный ресурс]. – Режим доступа: http://whitehatty.wordpress.com/2012/09/30/cryptdb-howto-compile-on-ubuntu-linux-12-04/, свободный (дата обращения: 21.04.2014).

12. Документация CryptDB [Электронный ресурс]. – Режим доступа: http://people.csail.mit.edu/nickolai/papers/raluca-cryptdb.pdf, свободный (дата обращения: 21.04.2014).

13. Redfield C.M.S. Practical security for multi-user web application databases: partial fulfillment of the requirements for the degree of Master of engineering in electrical engineering and computer science / Massachusetts Institute of Technology. – 2012. – 68 p.

14. Orderpreserving symmetric encryption / A. Boldyreva, N. Chenette, Y. Lee, A. O'Neill // Proceedings of the 28-th Annual international conference on the theory and applications of cryptographic techniques. – Cologne, Germany, 2009. – P. 224–241.

15. 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.

16. 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.

17. 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.

18. Darrow B. You want to crunch top-secret data securely? CryptDB may be the app for that. – 2013. [Электронный ресурс]. – Режим доступа: http://gigaom.com/2013/04/05/you-want-to-crunch-top-secret-data-securely-cryptdb-may-be-the-app-for-that/, свободный (дата обращения: 21.04.2014).

Трифонова Юлия Викторовна
Ассистент каф. технологий защиты информации
Санкт-Петербургского государственного университета аэрокосмического приборостроения (ГУАП)
Тел.: 8 (812) 494-70-77
Эл. почта: ulia@guap.ru

Жаринов Роман Феликсович
Аспирант каф. технологий защиты информации ГУАП
Тел.: 8 (812) 494-70-77
Эл. почта: roman@vu.spb.ru

Trifonova U.V., Zharinov R.F.

Opportunities of depersonalization personal data in systems using relational databases

This article discusses depersonalization of personal data, Federal service for supervision of communications, information technology, and mass media (Roscomnadzor) point of view and the using problems of depersonalization methods. Cross-platform solution of depersonalization personal data in relational databases is offered. The possibilities of using CryptDB tools, as a reliable method of anonymization of personal data on the server side.

Keywords: depersonalization, personal data, SQL, CryptDB.
